NEC

Preliminary User's Manual

V850E/PHO3

32-bit Single-Chip Microcontroller

Hardware

µPD70F3483
µPD70F3441

Document No. U17754EE4V0UM00
Date Published April 2008
© NEC Electronics 2007
Printed in Germany

Notes for CMOS Devices

(1) Precaution against ESD for semiconductors

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

(2) Handling of unused input pins for CMOS

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

(3) Status before initialization of MOS devices

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

Legal Notes

- The information in this document is current as of April 2008. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.
- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such NEC Electronics products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: “Standard”, “Special” and “Specific”.

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated “quality assurance program” for a specific application. The recommended applications of NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

"Standard":	Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
"Special":	Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
"Specific":	Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

Note

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

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.

FlexRay™

FlexRay™ is a trademark of DaimlerChrysler AG.

FlexRay™ IP-Module

The FlexRay IP-Module is licensed from the Robert Bosch GmbH.

Regional Information

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

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

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

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668, Japan
Tel: 044 4355111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554,|
U.S.A.
Tel: 408 5886000
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
0472 Düsseldorf, Germany
Tel: 0211 65030
http://www.eu.necel.com/

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908 691133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01 30675800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091 5042787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 6387200

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02 667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven,
The Netherlands
Tel: 040 2654010

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27
ZhiChunLu Haidian District,
Beijing 100083, P.R.China
Tel: 010 82351155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2511-2512, Bank of China
Tower,
200 Yincheng Road Central,
Pudong New Area,
Shanghai 200120, P.R. China
Tel: 021 58885400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886 9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R.O.C.
Tel: 02 27192377

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253 8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku, Seoul,
135-080, Korea Tel: 02-558-3737
http://www.kr.necel.com/

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: $\overline{\text{xxx}}$ (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.ee.nec.de.

Documentation Feedback

NEC Electronics appreciates your feedback to this document.
Please submit your suggestions for corrections or improvements to

DOC_PHO3@eu.necel.com

Thank you

Table of Contents

Chapter 1 Introduction 23

1.1 General 23

1.2 Features Summary 24

1.3 Description 27

1.3.1 Internal units 28

1.3.2 Structure of the manual 29

Chapter 2 Pin Functions 31

2.1 Overview 31

2.1.1 Description 32

2.1.2 Terms 35

2.1.3 Noise elimination 35

2.2 Port Group Configuration Registers 36

2.2.1 Overview 36

2.2.2 Pin function configuration 37

2.2.3 Pin data input/output 41

2.2.4 Emergency shut-off configuration 43

2.2.5 Write enable register 47

2.3 Port Group Configuration 49

2.3.1 Port group configuration lists 49

2.3.2 Non-port pins 53

2.3.3 Port Types Diagrams 55

2.3.4 Alphabetic pin function list 56

2.3.5 Port group 0 64

2.3.6 Port group 1 65

2.3.7 Port group 2 66

2.3.8 Port group 3 67

2.3.9 Port group 4 68

2.3.10 Port group 5 69

2.3.11 Port group 6 70

2.3.12 Port group 7 71

2.3.13 Port group 8 72

2.3.14 Port group 9 73

2.3.15 Port group 10 74

2.3.16 Port group 11 75

2.3.17 Port group AL 76

2.3.18 Port group AH 78

2.3.19 Port group CS 79

2.3.20 Port group CT 80

2.3.21 Port group CM 81

2.3.22 Port group CD 82

2.3.23 Port group DL 83

2.3.24 Port group DH 85

2.4 Noise Elimination 87

2.4.1 Analog filtered inputs 87

2.4.2 Digitally filtered inputs 88

2.5 Pin Functions during and after Reset 91

2.6 Recommended Connection of unused Pins . . . 93
2.7 Package Pins Assignment . . . 94

Chapter 3 CPU System Functions . . . 97

3.1 Overview . . . 97
3.1.1 Description . . . 98
3.2 CPU Register Set . . . 99
3.2.1 General purpose registers (r0 to r31) . . . 100
3.2.2 System register set . . . 101
3.2.3 Floating-point arithmetic registers . . . 107
3.3 Operation Modes . . . 110
3.3.1 Normal operation mode . . . 110
3.3.2 Flash programming mode . . . 110
3.3.3 Boundary scan mode . . . 111
3.4 Address Space . . . 111
3.4.1 CPU address space and physical address space . . . 111
3.4.2 Program and data space . . . 113
3.5 Memory . . . 115
3.5.1 Memory areas . . . 115
3.6 RAM ECC error detection . . . 117
3.6.1 RAM ECC initialization . . . 117
3.6.2 RAM ECC registers . . . 118
3.6.3 RAM ECC function check . . . 121
3.6.4 Recommended use of data address space . . . 122
3.7 Write Protected Registers . . . 123

Chapter 4 Interrupt Controller (INTC) . . . 125

4.1 Features . . . 125
4.2 Non-Maskable Interrupts . . . 132
4.2.1 Operation . . . 133
4.2.2 Restore . . . 134
4.2.3 Non-maskable interrupt status flag (NP) . . . 135
4.2.4 NMI control . . . 135
4.3 Maskable Interrupts . . . 136
4.3.1 Operation . . . 136
4.3.2 Restore . . . 138
4.3.3 Priorities of maskable interrupts . . . 139
4.3.4 xxICn - Maskable interrupts control register . . . 143
4.3.5 IMR0 to IMR7 - Interrupt mask registers . . . 147
4.3.6 ISPR - In-service priority register . . . 148
4.3.7 Maskable interrupt status flag (ID) . . . 149
4.3.8 External maskable interrupts . . . 149
4.4 Edge and Level Detection Configuration . . . 150
4.5 Interrupt Sharing . . . 153
4.6 Software Exception . . . 157
4.6.1 Operation . . . 157
4.6.2 Restore . . . 158
4.6.3 Exception status flag (EP) . . . 159
4.7 Exception Trap . . . 160

4.7.1 Illegal opcode definition . . . 160
4.7.2 Debug trap . . . 162
4.8 Multiple Interrupt Processing Control . . . 164
4.9 Interrupt Response Time . . . 166
4.10 Periods in Which Interrupts Are Not Acknowledged . . . 167

Chapter 5 CRC Function (CRC) . . . 169
5.1 Features . . . 169
5.1.1 CPU-CRC . . . 169
5.1.2 DATA-CRC . . . 169
5.2 Configuration . . . 170
5.3 CRC Registers . . . 171
5.4 Operation . . . 176
5.4.1 CPU-CRC . . . 176
5.4.2 DATA-CRC . . . 178

Chapter 6 Clock Generator . . . 179
6.1 Overview . . . 179
6.1.1 Features . . . 179
6.1.2 Description . . . 180
6.2 Control Registers . . . 182
6.3 Clock Monitor . . . 183
6.3.1 Clock Monitor registers . . . 183
6.3.2 Operation of the Clock Monitor . . . 185
6.4 Power Save Control . . . 187
6.4.1 HALT mode . . . 188

Chapter 7 Flash Memory . . . 191
7.1 Code Flash Memory Overview . . . 192
7.1.1 Code flash memory features . . . 192
7.1.2 Code flash memory mapping . . . 192
7.1.3 Code flash memory functional outline . . . 194
7.1.4 Code flash memory erasure and rewrite . . . 197
7.1.5 Code flash memory error detection . . . 198
7.2 Data Flash Memory . . . 200
7.2.1 Data flash memory features . . . 200
7.2.2 Data flash memory map . . . 200
7.2.3 Data flash control register . . . 201
7.2.4 Data flash reading . . . 201
7.2.5 Data flash writing . . . 201
7.3 Flash Programming with Flash Programmer . . . 202
7.3.1 Programming environment . . . 202
7.3.2 Communication mode . . . 203
7.3.3 Pin connection with flash programmer PG-FP4 . . . 205
7.3.4 Flash memory programming control . . . 207
7.4 Code Flash Self-Programming . . . 213
7.4.1 Self-programming enable . . . 214
7.4.2 Self-programming library functions . . . 214
7.4.3 Secure self-programming (boot cluster swapping) . . . 215

7.4.4 Interrupt handling during flash self-programming 218

Chapter 8 Data Protection and Security 219
8.1 Overview 219
8.2 N-Wire Debug Interface Protection 219
8.3 Flash Programmer and Self-Programming Protection 221

Chapter 9 Bus and Memory Control (BCU, MEMC) 225
9.1 Overview 225
9.1.1 Description 226
9.1.2 Memory areas/banks and chip select signals 228
9.1.3 Peripheral I/O areas 230
9.1.4 Data flash access 232
9.1.5 FlexRay access 233
9.1.6 CRC module access 234
9.1.7 Chip select priority control 235
9.1.8 Bus properties 236
9.1.9 Boundary operation conditions 237
9.1.10 Initialization for access to external devices 238
9.2 Registers 239
9.2.1 BCU registers 240
9.2.2 Memory Controller registers 247
9.3 Configuration of Memory Access 256
9.3.1 Endian format 256
9.3.2 Wait functions 257
9.3.3 External wait function 258
9.4 External Devices Interface Timing 259
9.4.1 SRAM, external ROM, external I/O access 259
9.5 Data Access Order 267
9.5.1 Access to 8-bit data busses 267
9.5.2 Access to 16-bit data busses 273

Chapter 10 DMA Functions (DMA Controller) 279
10.1 Overview 279
10.1.1 Principle of operation 279
10.1.2 Forcible termination of DMA transfer 280
10.1.3 DMA interrupt function 281
10.2 Registers 282
10.3 Transfer Details 290
10.3.1 DMA transfer from A/D Converter 290
10.3.2 DMA transfer to Timer S 293
10.3.3 DMA transfer to/from serial interfaces 296
10.3.4 Example of forcible termination of DMA transfer 300

Chapter 11 Asynchronous Serial Interface (UARTC) 301
11.1 Features 301
11.2 Configuration 302
11.3 UARTC Registers 304
11.4 Interrupt Request Signals 314

11.5 Operation . . . 315
11.5.1 Data format . . . 315
11.5.2 SBF transmission/reception format . . . 317
11.5.3 SBF transmission . . . 319
11.5.4 SBF reception . . . 319
11.5.5 UART transmission . . . 321
11.5.6 Continuous transmission procedure . . . 322
11.5.7 UART reception . . . 324
11.5.8 Reception errors . . . 326
11.5.9 Parity types and operations . . . 326
11.5.10 Receive data noise filter . . . 328
11.6 Baud Rate Generator . . . 329
11.7 Cautions . . . 335

Chapter 12 Clocked Serial Interface (CSIB) . . . 337

12.1 Features . . . 337
12.2 Configuration . . . 338
12.3 CSIB Control Registers . . . 340
12.4 Operation . . . 346
12.4.1 Single transfer mode (master mode, transmission/reception mode) . . . 346
12.4.2 Single transfer mode (master mode, reception mode) . . . 348
12.4.3 Continuous mode (master mode, transmission/reception mode) . . . 349
12.4.4 Continuous mode (master mode, reception mode) . . . 350
12.4.5 Continuous reception mode (error) . . . 351
12.4.6 Continuous mode (slave mode, transmission/reception mode) . . . 352
12.4.7 Continuous mode (slave mode, reception mode) . . . 354
12.4.8 Clock timing . . . 355
12.5 Output Pins . . . 357
12.6 Operation Flow . . . 358
12.7 Baud Rate Generator . . . 364
12.7.1 Overview . . . 364
12.7.2 Baud rate generator registers . . . 365
12.7.3 Baud rate generation . . . 366
12.8 Cautions . . . 366
12.8.1 CSIB operation stop . . . 366

Chapter 13 Enhanced Queued Clocked Serial Interface (CSIE) . . . 369

13.1 Features . . . 369
13.1.1 Queued CSI block diagram . . . 370
13.1.2 Input/Output pins . . . 371
13.2 Queued CSI Control Registers . . . 372
13.3 Explanation of Queued CSI Functions . . . 393
13.3.1 Transmit buffer . . . 393
13.3.2 Serial data direction select function . . . 395
13.3.3 Data length select function . . . 396
13.3.4 Slave mode . . . 396
13.3.5 Master mode . . . 397
13.3.6 Transmission clock select function . . . 398

13.3.7 Description of the single transfer mode . . . 399
13.3.8 Description of the block transfer mode . . . 401
13.3.9 Description of the operation modes . . . 403
13.3.10 Additional timing and delay selections . . . 403
13.3.11 Default pin levels . . . 406
13.3.12 Transmit buffer overflow interrupt signal (INTCEnOF) . . . 407
13.3.13 Enhanced chip select timing . . . 408
13.4 Operating Procedure . . . 416
13.4.1 Single transfer mode (master mode, transmit only mode) . . . 416
13.4.2 Single transfer mode (master mode, receive only mode) . . . 417
13.4.3 Single transfer mode (master mode, transmit/receive mode) . . . 418
13.4.4 Single transfer mode (slave mode, transmit only mode) . . . 419
13.4.5 Single transfer mode (slave mode, receive only mode) . . . 420
13.4.6 Single transfer mode (slave mode, transmit/receive mode) . . . 421
13.4.7 Block transfer mode (master mode, transmit only mode) . . . 422
13.4.8 Block transfer mode (master mode, receive only mode) . . . 423
13.4.9 Block transfer mode (master mode, transmit/receive mode) . . . 424
13.4.10 Block transfer mode (slave mode, transmit only mode) . . . 425
13.4.11 Block transfer mode (slave mode, receive only mode) . . . 426
13.4.12 Block transfer mode (slave mode, transmit/receive mode) . . . 427
13.5 Cautions . . . 428
13.5.1 Inconsistent read data in master mode . . . 428
13.5.2 Transmit/receive enable after CEnTX0 write . . . 428

Chapter 14 CAN Controller (CAN) . . . 429

14.1 Features . . . 429
14.1.1 Overview of functions . . . 430
14.1.2 Configuration . . . 431
14.2 CAN Protocol . . . 432
14.2.1 Frame format . . . 432
14.2.2 Frame types . . . 433
14.2.3 Data frame and remote frame . . . 433
14.2.4 Error frame . . . 440
14.2.5 Overload frame . . . 441
14.3 Functions . . . 442
14.3.1 Determining bus priority . . . 442
14.3.2 Bit stuffing . . . 442
14.3.3 Multi masters . . . 443
14.3.4 Multi cast . . . 443
14.3.5 CAN sleep mode/CAN stop mode function . . . 443
14.3.6 Error control function . . . 443
14.3.7 Baud rate control function . . . 450
14.4 Connection with Target System . . . 453
14.5 Internal Registers of CAN Controller . . . 454
14.5.1 CAN module register and message buffer addresses . . . 454
14.5.2 CAN Controller configuration . . . 455
14.5.3 CAN registers overview . . . 456
14.5.4 Register bit configuration . . . 458
14.6 Bit Set/Clear Function . . . 461
14.7 Control Registers . . . 463

14.8 CAN Controller Initialization . . . 499
14.8.1 Initialization of CAN module . . . 499
14.8.2 Initialization of message buffer . . . 499
14.8.3 Redefinition of message buffer . . . 499
14.8.4 Transition from initialization mode to operation mode . . . 501
14.8.5 Resetting error counter CnERC of CAN module . . . 502
14.9 Message Reception . . . 503
14.9.1 Message reception . . . 503
14.9.2 Receive data read . . . 504
14.9.3 Receive history list function . . . 505
14.9.4 Mask function . . . 507
14.9.5 Multi buffer receive block function . . . 509
14.9.6 Remote frame reception . . . 510
14.10 Message Transmission . . . 511
14.10.1 Message transmission . . . 511
14.10.2 Transmit history list function . . . 513
14.10.3 Automatic block transmission (ABT) . . . 515
14.10.4 Transmission abort process . . . 517
14.10.5 Remote frame transmission . . . 518
14.11 Power Saving Modes . . . 519
14.11.1 CAN sleep mode . . . 519
14.11.2 CAN stop mode . . . 522
14.11.3 Example of using power saving modes . . . 523
14.12 Interrupt Function . . . 524
14.13 Diagnosis Functions and Special Operational Modes . . . 525
14.13.1 Receive-only mode . . . 525
14.13.2 Single-shot mode . . . 526
14.13.3 Self-test mode . . . 527
14.13.4 Receive/transmit operation in each operation mode . . . 528
14.14 Time Stamp Function . . . 529
14.14.1 Time stamp function . . . 529
14.15 Baud Rate Settings . . . 530
14.15.1 Baud rate setting conditions . . . 530
14.15.2 Representative examples of baud rate settings . . . 534
14.16 Operation of CAN Controller . . . 538

Chapter 15 A/D Converter (ADC) . . . 567
15.1 Features . . . 567
15.2 Configuration . . . 568
15.3 Control Registers . . . 571
15.4 Operation . . . 580
15.4.1 Basic operation . . . 580
15.4.2 Operation mode and trigger mode . . . 581
15.5 Operation in A/D Trigger Mode . . . 586
15.5.1 Select mode operation . . . 586
15.5.2 Scan mode operation . . . 588
15.6 Operation in Timer Trigger Mode . . . 590
15.6.1 Select mode operation . . . 590
15.6.2 Scan mode operation . . . 594
15.7 Operation in External Trigger Mode . . . 596

15.7.1 Select mode operation 596
15.7.2 Scan mode operation 600
15.8 Precautions 602

Chapter 16 16-Bit Timer/Event Counter AA (TAA) 605
16.1 Features 605
16.2 Function Outline 605
16.3 Configuration 606
16.4 Input Selection Registers 613
16.5 Control Registers 616
16.6 Operation 628
16.6.1 Anytime write and reload 629
16.6.2 Interval timer mode (TAAnMD2 to TAAnMD0 = 000_B) 633
16.6.3 External event counter mode (TAAnMD2 to TAAnMD0 = 001_B) 636
16.6.4 External trigger pulse mode (TAAnMD2 to TAAnMD0 = 010_B) 640
16.6.5 One-shot pulse mode (TAAnMD2 to TAAnMD0 = 011_B) 643
16.6.6 PWM mode (TAAnMD2 to TAAnMD0 = 100_B) 646
16.6.7 Free-running mode (TAAnMD2 to TAAnMD0 = 101_B) 651
16.6.8 Pulse width measurement mode (TAAnMD2 to TAAnMD0 = 110B)657
16.6.9 32-bit Capture in free-running cascade mode 664
16.6.10 Specific TAAnCCRm overflow value write 669
16.7 Timer Synchronization Operation Function 674

Chapter 17 16-Bit Timer/Event Counter S (TMS) 677
17.1 Features 677
17.2 Configuration 679
17.3 Control Register 694
17.4 Basic Operation 759
17.4.1 16-bit counter basic operation 759
17.4.2 Compare register rewriting 761
17.4.3 Output list in each mode 775
17.5 Match Interrupt 783
17.5.1 Caution on compare match interrupt 785
17.6 Flags 787
17.6.1 Count-up flags (TSnCUF, TSnSUF) 788
17.6.2 Positive/negative-phase simultaneous active detection flag (TSnTBF) . . . 790
17.6.3 Reload request flag 791
17.6.4 Noise detection flag (TSnNDF) 792
17.6.5 Pattern order detection flag (TSnTSF) 793
17.6.6 Pattern error detection flag (TSnPEF) 795
17.6.7 Pattern reverse detection flag (TSnPRF) 796
17.6.8 TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF) 798
17.6.9 TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) . 800
17.6.10 Pattern phase difference detection flag (TSnPPF) 801
17.6.11 Overflow flag (TSnOVF) 802
17.6.12 Timer output pattern flags (TSnOPF2 to TSnOPF0) 803
17.6.13 Pattern switch detection flag (TSnESG) 804

17.7 Interrupt Thinning out Function . . . 805
17.7.1 Interrupt thinning out function operation . . . 806
17.7.2 Operation example when peak and trough interrupts occur alternatively (in high-accuracy T-PWM mode)808
17.7.3 Operation example when only peak interrupt occurs (in high-accuracy T-PWM mode) 810
17.7.4 Operation example when only trough interrupt occurs (in high-accuracy T-PWM mode) 812
17.7.5 Operation example when only peak interrupt occurs (in PWM mode) . . . 814
17.8 A/D Conversion Trigger Function . . . 816
17.8.1 A/D conversion trigger operation . . . 817
17.9 Error/warning Interrupt . . . 823
17.9.1 Error interrupt function . . . 823
17.9.2 Warning interrupt function . . . 826
17.10 Operation in Each Mode . . . 827
17.10.1 Interval timer mode . . . 828
17.10.2 External event count mode . . . 832
17.10.3 External trigger pulse output mode . . . 840
17.10.4 One-shot pulse mode . . . 844
17.10.5 PWM mode . . . 848
17.10.6 Free-running mode . . . 858
17.10.7 Triangular-wave PWM mode . . . 862
17.10.8 High-accuracy T-PWM mode . . . 867
17.10.9 PWM mode with dead time . . . 914
17.10.10 120° excitation mode . . . 928
17.10.11 Special 120° excitation mode . . . 969
17.10.12 Special pattern output mode . . . 999
17.10.13 Software output function . . . 1014
17.10.14 180° excitation control function . . . 1016
17.11 Timer Synchronization . . . 1017

Chapter 18 16-bit Timer/Event Counter T (TMT) . . . 1019
18.1 Features . . . 1019
18.2 Function Outline . . . 1019
18.3 Configuration . . . 1020
18.4 Control Registers . . . 1026
18.5 Basic Operation . . . 1042
18.5.1 Basic counter operation . . . 1042
18.5.2 Method for writing to compare register . . . 1046
18.6 Operation in Each Mode . . . 1050
18.6.1 Interval timer mode . . . 1050
18.6.2 External event count mode . . . 1053
18.6.3 External trigger pulse output mode . . . 1058
18.6.4 One-shot pulse mode . . . 1061
18.6.5 PWM mode . . . 1064
18.6.6 Free-running mode . . . 1069
18.6.7 Pulse width measurement mode . . . 1074
18.6.8 Triangular wave PWM mode . . . 1075
18.6.9 Encoder count function . . . 1076
18.6.10 Offset trigger generation mode . . . 1102

18.7 Cautions 1103
18.7.1 TMT 'Encoder load enable'-mode limitation 1103

Chapter 19 Timer Interconnection 1105

19.1 Control register 1108
19.2 2-Phase Encoder Control Function 1110
19.2.1 Function overview 1110
19.2.2 Configuration 1111
19.3 3-Phase Pulse Input Control Function 1113
19.3.1 Function overview 1113
19.3.2 Configuration 1114
19.4 3-Phase Encoder Function 1116
19.4.1 Function overview 1116
19.4.2 Configuration 1116
19.5 TMT as Filter for TMAA 1117

Chapter 20 On-Chip Debug Unit 1119

20.1 Functional Outline 1119
20.1.1 Debug functions 1119
20.1.2 Security function 1121
20.2 Connection to N-Wire Emulator 1122
20.2.1 KEL connector 1122
20.3 Restrictions and Cautions on On-Chip Debug Function 1126

Chapter 21 Random Number Generator (RNG) 1127

21.1 Features 1127
21.2 Configuration 1127
21.3 Operation 1128
21.3.1 Access timing 1128

Chapter 22 Auxiliary Frequency Output Function (AFO) 1129

22.1 Features 1129
22.2 Configuration 1129
22.3 Control Registers 1130
22.4 Operation 1132
22.4.1 Auxiliary frequency generation 1132
22.4.2 Interval timer function 1132

Chapter 23 Non Break Debug Unit (NBD) 1133

23.1 Functional Outline 1133
23.1.1 Debug functions 1133
23.2 Connection to RTE-NBD2 Evaluator 1135
23.3 Restrictions and Cautions on NBD Functions 1137
23.4 NBD Registers 1139

Chapter 24 FlexRay™ (FR) 1145

24.1 About this Document 1145
24.1.1 Conventions 1145

24.1.2 Definition 1145
24.1.3 Scope 1145
24.1.4 References 1145
24.1.5 Terms and abbreviations 1146
24.2 Overview 1147
24.2.1 Block diagram 1148
24.3 Host CPU Interface 1151
24.3.1 Host CPU interface block diagram 1151
24.3.2 Signal description 1152
24.3.3 Host CPU interface timing 1154
24.3.4 CPU, MEMC, Port and FlexRay IP register settings 1157
24.4 Programmer's Model 1159
24.4.1 Register map 1159
24.4.2 V850 E-Ray registers 1164
24.4.3 Special registers 1168
24.4.4 Interrupt registers 1173
24.4.5 CC control registers 1194
24.4.6 CC status registers 1214
24.4.7 Message buffer control registers 1229
24.4.8 Message buffer status registers 1233
24.4.9 Identification Registers 1244
24.4.10 Input buffer 1246
24.4.11 Output buffer 1253
24.5 Functional Description 1263
24.5.1 Communication cycle 1263
24.5.2 Communication modes 1265
24.5.3 Clock synchronization 1266
24.5.4 Error handling 1268
24.5.5 Communication controller states 1270
24.5.6 Network management 1284
24.5.7 Filtering and masking 1284
24.5.8 Transmit process 1287
24.5.9 Receive process 1290
24.5.10 FIFO function 1292
24.5.11 Message handling 1294
24.5.12 Message RAM 1303
24.5.13 Module interrupt 1311
24.6 Appendix 1313
24.6.1 Register bit overview 1313
24.6.2 Assignment of FlexRay Configuration Parameters 1324
24.7 Cautions 1326
24.7.1 Missing cycle start interrupt in startup phase 1326
24.7.2 Update of status registers CCEV and CCSV delayed with respect to interrupt flags EIR.PEMC, SIR.WST, SIR.SUCS, SIR.WUPA, SIR.WUPB1326
24.7.3 Clearing of interrupt flags EIR.PEMC, SIR.WST, SIR.WUPA, SIR.WUPB, SIR.CAS, SIR.SUCS not accepted1327
24.7.4 Content error detected in sync frame at slot boundary 1327
24.7.5 Loop back mode operates only at 10 MBit/s 1328
24.7.6 False clock correction value possible in case action point coincides with end of received syncframe1328

24.7.7 Noise following a dynamic frame that delays idle detection may fail to stop slot counting for the remainder of the dynamic segment1329
24.7.8 Mismatch between macrotick value and cycle counter value at cycle boundary 1329
24.7.9 Wrong payload data transmitted due to disturbed channel-idle sequence 1330
24.7.10 Status of Network Management Vector NMVn after the CC has left NORMAL_ACTIVE or NORMAL_PASSIVE state by application of commands READY, HALT, or FREEZE1330
24.7.11 Reception of wakeup pattern signalled for channel disconnected by SUCC1.CCHA/B = ’0’1331
24.7.12 Clock correction value calculated with wrong deviation value 1331
24.7.13 Wrong payload data written to receive buffer 1332
24.7.14 In case of a faulty configuration of pLatestTx and transmission across dynamic segment boundary error flags EIR.LTVA/B may not be set when a latest transmit violation occurs1333
24.7.15 In case eray_bclk is below eray_sclk/2, TEST1.CERA/B may fail to report a detected coding error1333
24.7.16 Generating EIR.SFO considers number of sync frames in the last cycle only, double cycle not evaluated1334
24.7.17 CAS collision in case of macrotick length > CAS 1334
24.7.18 Reception of more than gSyncNodeMax different sync frames per double cycle may lead to incorrect offset correction value1335
24.7.19 Time stamp of the wrong channel may be used for offset correction term calculation if reception time stamps of a sync frame are different for channel A and B1336
24.7.20 Sync frame reception after noise or aborted frame before action point . . 1336
24.7.21 Additional SIR.WST events. 1337
24.7.22 Update of Aggregated Channel Status ACS in dynamic segment in minislots following ID 20471337
24.7.23 Faulty update of FRLDTS.LDTA,B[10:0] due to ECC error. 1338
24.7.24 Improper resolution of startup collision 1338
24.7.25 ’integration successful on X’ and ’integration abort on Y’ at the same point in time leads to inconsistent states of SUC and GTU1339
24.7.26 Detection of parity errors outside immediate scope 1340
24.7.27 FRCCSV.SLM [1:0] delayed to FRCCSV.POCS[5:0] on transitions between states WAKEUP and READY1340
24.7.28 Wakeup listen counter started one bit time early 1340
24.7.29 Payload corruption after reception of valid frame followed by slot boundary crossing frame1341
24.7.30 FRCCSV.RCA[4:0] and FRCCSV.PSL[5:0] inconsistent after FREEZE command1341
24.7.31 CLEAR_RAMS command does not clear the 1st RAM word 1342
24.7.32 Cycle filtering in slot 1 1342
24.7.33 Corruption of frame received in slot N by second frame reception before action point of transmit slot N+11343
24.7.34 Corruption of frame received in slot N by second frame reception before action point of transmit slot N+11343
24.7.35 Corruption of frame received in slot N by second frame reception before action point of transmit slot N+11344

Chapter 25 Reset Function 1345
25.1 Features 1345
25.2 Configuration. 1345

25.3 Operation 1347

Chapter 26 Boundary Scan 1349
26.1 Boundary Scan Pins 1349
26.2 Boundary Scan Functions 1350

Appendix A Special Function Registers 1351
A.1 CAN Registers 1351
A.2 FlexRay Registers 1354
A.3 Other Special Function Registers 1357

Revision History 1375

Index 1377

Chapter 1 Introduction

The V850E/PHO3 is a product in NEC Electronics' V850 family of single-chip microcontrollers designed for automotive applications.

1.1 General

The V850E/PHO3 single-chip microcontroller devices make the performance gains attainable with 32-bit RISC-based controllers available for embedded control applications.

The V850E/PHO3 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. The integrated FlexRay™ interface implements the FlexRay™ network protocol.

Thus equipped, the V850E/PHO3 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) V850E CPU

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

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

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

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

(2) On-chip flash memory

The V850E/PHO3 microcontroller has on-chip flash memory. It is possible to program the controller directly in the target environment where it is mounted.

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

(3) Reliability

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

Writing to the flash is protected by various measures.

The CPU core is subject to a multi-channel cyclic redundancy check (CRC). This allows comprehensive self-diagnosis of the CPU.

(4) A full range of software development tools

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

1.2 Features Summary

The V850E/PHO3 series consists of the μPD70F3441 and μPD70F3483 microcontrollers.

The CPU core provides:

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

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

Table 1-1 V850E/PHO3 features (1/2)

Features		V850E/PHO3
CPU		V850E (32-bit RISC)
Internal memory	Code ROM	• µPD70F3483: 768 KB • µPD70F3441: 992 KB
	Mask ROM	-
	RAM	60 KB
	Data Flash	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. 128 MHz
	MainOSC	operates on 16 MHz crystal
	PLL ratio	x 8, x 5
	Internal oscillator	6.8 MHz typ.
	FlexRay	80 MHz
Interrupts	Non-maskable	1 ch
	Maskable int.	101 ch
	Maskable ext.	14 ch
I/O lines	I/O ports	143
	Input ports	5
Timers	TAA	10 ch (16-bit general purpose timer/counter, cascadable for 32-bit operations)
	TMS	2 ch (16-bit timer/counter with Motor Control Functions)
	TMT	2 ch (16-bit general purpose timer/counter with PWM functions)
A/D converters	Analog inputs	20
	Resolution	10 bit
	Type	Successive approximation
Serial interfaces	UARTC	3 ch
	CSIB	2 ch Clocked Serial Interface
	CSIE	2 ch queued CSI
	CAN	2 ch CAN (32 message buffers for each channel)
FlexRay		2 ch (Protocol specification V2.1)
DMA		8 ch
Boundary scan		provided

Table 1-1 V850E/PHO3 features (2/2)

Features		V850E/PHO3
Other functions	Power save modes	HALT
	RNG	Random Number Generator provided
	Clock Monitor	Main oscillator monitor provided
	Aux. frequency output	Programmable baud rate generator provided
	On-chip debug	• Connection of an external N-Wire emulator provided • NBD (Non Break Debug) interface with 4 KB of tuning RAM (overlay RAM) for RAM monitoring, data tuning, calibration and bypassing
Power supply		• 3.3 V ± 0.3 V • 1.5V ± 0.15 V (refer to Data Sheet)
Package		357 pin BGA (0.8 mm ball pitch)

1.3 Description

The following figure provides a functional block diagram of the V850E/PHO3 microcontroller.

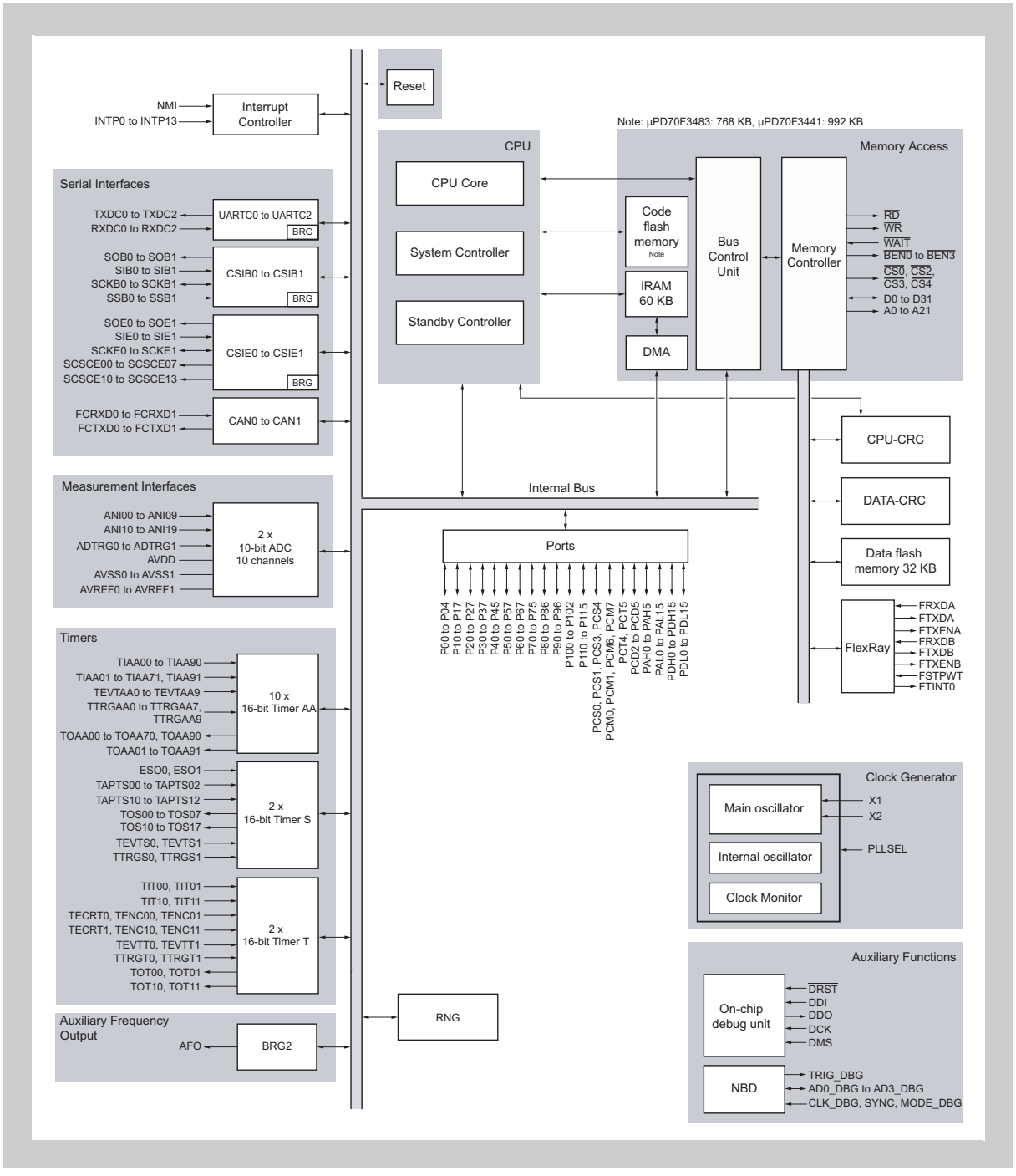


Figure 1-1 V850E/PHO3 block diagram

Note The CAN Controller of this device fulfills the requirements according ISO 11898. Additionally, the CAN Controller was tested according to the test procedures required by ISO 16845. The CAN Controller has successfully passed all test patterns. Beyond these test patterns, other tests like robustness tests and processor interface tests as recommended by C&S/FH Wolfenbuettel have been performed with success.

1.3.1 Internal units

CPU The CPU can execute almost all instruction processing, such as address calculation, arithmetic and logic operations, and data transfer, in one clock cycle under control of a five-stage pipeline.

Dedicated hardware units such as a multiplier and a 32-bit barrel shifter are provided to speed up complicated instruction processing.

Code flash This is a built-in flash memory of 992 KB (768 KB) mapped to addresses 000 0000_H to 00F $7FFF_H$ (00B $FFFF_H$). The CPU can access this memory in one clock when it fetches an instruction.

Data flash A 32 KB data flash is provided, accessible via the Memory Controller.

RAM This is a RAM of 60 KB mapped to addresses 3FF 0000_H to 3FF $EFFF_H$. The CPU can access this RAM in one clock when it accesses data.

DMA Controller The V850E/PHO3 has an eight-channel DMA Controller that transfers data between the internal RAM, serial interfaces, Timer S and external memory - in response to interrupt requests from the on-chip peripherals.

BCU / MEMC The Bus Control Unit (BCU) and Memory Controller (MEMC) control the access to the on-chip peripherals, to the data flash, FlexRay, and to external memory.

Clock Generator The Clock Generator generates the system clocks. It includes the crystal-controlled 16 MHz main oscillator and an internal 8 MHz internal oscillator.

Clock Monitor The Clock Monitor monitors the main oscillator. In case of failure, it can switch the CPU system and the Memory Controller to the internal oscillator.

On-chip Debug Unit An on-chip debug function that uses an N-Wire interface is provided.

NBD The Non Break Debug unit allows to access code flash, RAM, tuning RAM and peripherals in the background while the CPU is executing program code.

INTC The Interrupt Controller (INTC) processes non-maskable and maskable interrupt requests from the on-chip peripheral hardware and external sources. Eight levels of priorities can be specified for these interrupt requests, and multiple servicing control can be performed on interrupt sources.

CPU-CRC The CPU-CRC (CPU Cyclic Redundency Check) function can be used to support user self test software designed for testing the CPU core function.

DATA-CRC The DATA-CRC (DATA Cyclic Redundency Check) can be used verify or generate CRC protected data streams of arbitrary length and different bit widths.

UARTC The UARTs provide 2-wire asynchronous serial interfaces.

CSIB The Clocked Serial Interfaces are 3-wire variable-length serial interfaces.

CSIE The Enhanced Queued Clocked Serial Interfaces CSIE provide an internal FIFO buffer for queued operation.

CAN The CAN Controllers are small-scale digital data transmission systems that transfer data between units.

A/D Converter These are two high-speed, high-resolution 10-bit A/D Converters, each with 10 analog input pins. They are of successive approximation type.

Timers/counters Ten 16-bit timers/event counters TAA, two 16-bit timers/event counters TMS with motor control capability, and two general purpose 16-bit timers TMT are provided.

RNG The hardware Random Number Generator (RNG) generates 16-bit random numbers. The seed is generated by hardware, and the sequence of numbers passes the FIPS tests.

AFO The Auxiliary Frequency Output (AFO) provides a clock signal generated by a separate baud rate generator (BRG2).

NBD The Non-Breaking Debug unit (NBD) supports real-time debugging while the controller is active.

1.3.2 Structure of the manual

This manual explains how to use the V850E/PHO3 microcontroller devices. It provides comprehensive information about the building blocks, their features, and how to set registers in order to enable or disable specific functions.

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

- Core functions

  “Pin Functions” on page 31
  “CPU System Functions” on page 97
  “Interrupt Controller (INTC)” on page 125
  “CRC Function (CRC)” on page 169
  “Clock Generator” on page 179

- Memory access

  “Flash Memory” on page 191
  “Bus and Memory Control (BCU, MEMC)” on page 225
  “DMA Functions (DMA Controller)” on page 279

- Serial interfaces

  “Asynchronous Serial Interface (UARTC)” on page 301
  “Clocked Serial Interface (CSIB)” on page 337
  “Enhanced Queued Clocked Serial Interface (CSIE)” on page 369
  “CAN Controller (CAN)” on page 429

- Measurement interfaces

  “A/D Converter (ADC)” on page 567

- Timers

  “16-Bit Timer/Event Counter AA (TAA)” on page 605
  “16-Bit Timer/Event Counter S (TMS)” on page 677
  “16-bit Timer/Event Counter T (TMT)” on page 1019
  “Timer Interconnection” on page 1105

- Auxiliary functions

  “On-Chip Debug Unit” on page 1119
  “Random Number Generator (RNG)” on page 1127
  “Auxiliary Frequency Output Function (AFO)” on page 1129
  “Non Break Debug Unit (NBD)” on page 1133
  “FlexRayTM” on page 1147

- Power and reset

  “Reset Function” on page 1345

Chapter 2 Pin Functions

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

2.1 Overview

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

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

For a description of the terms pin, port or port group, see *"Terms" on page 35*.

Features summary

- Number of ports and port groups:
  - Input/Output ports:143
  - Input ports: 5
  - Port groups: 20
- 3.3 V I/O:
  Please refer to the Data Sheet.
- Configuration possible for individual pins.
- Emergency shut-off configuration (only for dedicated ports 5 and 6)

2.1.1 Description

This microcontroller has the port groups shown below.

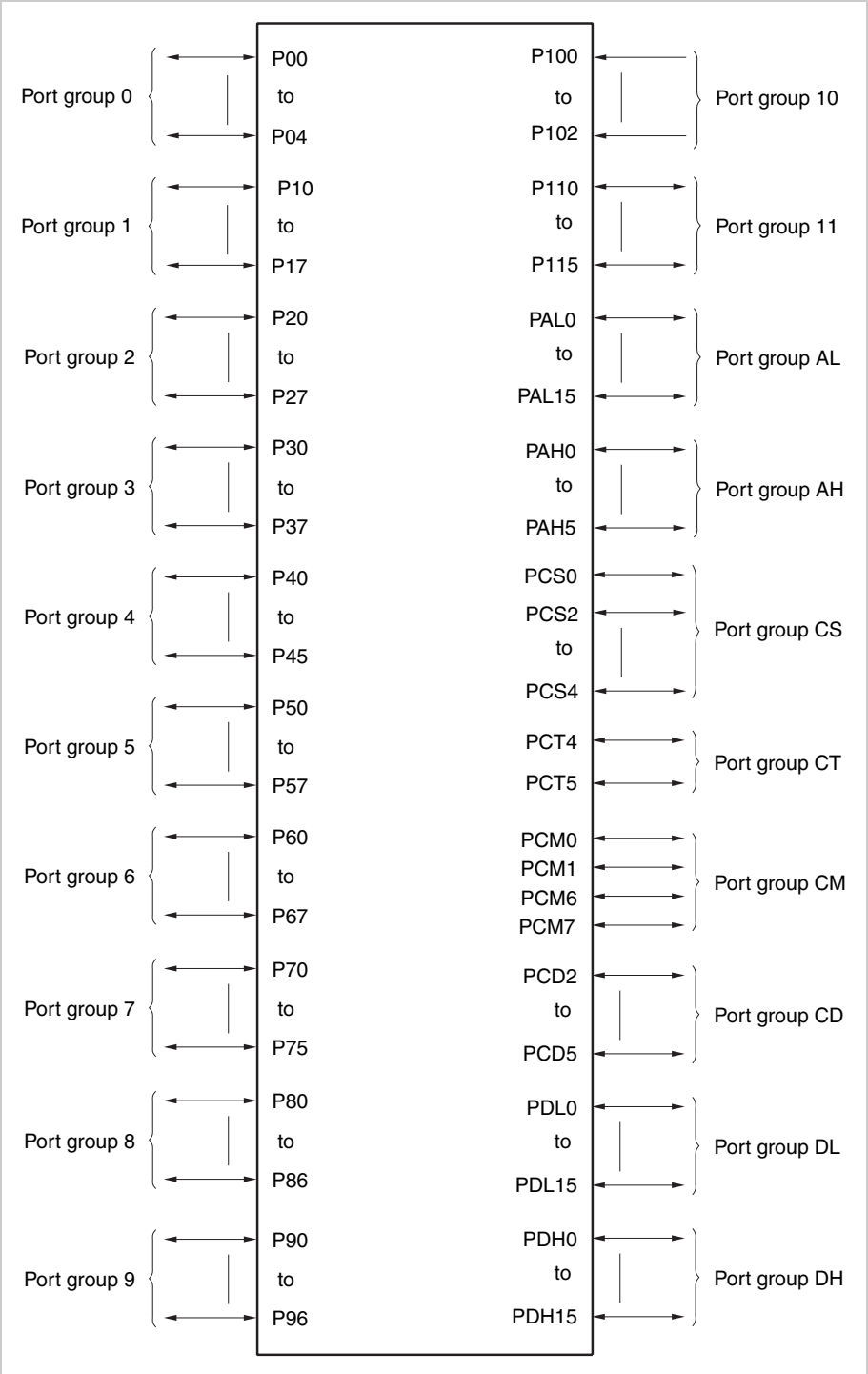


Figure 2-1 Port groups

Port group overview

Table 2-1 gives an overview of the port groups. For each port group it shows the supported functions in port mode and in alternative mode. Any port group can operate in 8-bit or 1-bit units. Port groups AL, DL and DH can additionally operate in 16-bit units.

Table 2-1 Functions of each port group (1/2)

Port group name	Function	
	Port mode	Alternative mode
0	5-bit input	• External interrupt 0 to 3 • Non maskable interrupt • Emergency shut-off control of Timers TMS0/TMS1 • ADC trigger input
1	8-bit input/output	• Timer TAA0 to TAA3 channels
2	8-bit input/output	• Timer TAA4 to TAA7 channels
3	8-bit input/output	• External interrupt 4 and 5 • UARTC0 receive/transmit data, • UARTC1 receive/transmit data • CAN0 receive/transmit data, • CAN1 receive/transmit data
4	6-bit input/output	• CSIB0 data/clock line • CSIB1 data/clock line • UARTC2 receive/transmit data • External interrupt 13
5	8-bit input/output	• Timer TMS0 pulse signal output
6	8-bit input/output	• Timer TMS1 pulse signal output
7	6-bit input/output	• Timer TMT0 channels • Timer TMT1 channels • External interrupt 12 • Auxiliary frequency output
8	7-bit input/output	• CSIE0 data/clock line • CSIE0 chip select areas 0 to 3 • CSIB0 serial slave select input • External interrupts 6 to 8
9	7-bit input/output	• CSIE0 chip select areas 4 to 7 • CSIE1 data/clock line • CSIE1 chip select areas • CSIB1 serial slave select input • External interrupts 9 to 11 • Interrupt signal from external/internal FlexRay
10	3-bit input/output	• Timer TAA8 channels • Timer TAA9 channels
11	6-bit input/output	• Timer TMS0 input channels • Timer TMS1 input channels
AL	16-bit input/output	• External memory interface address lines 0 to 15
AH	6-bit input/output	• External memory interface address lines 16 to 21
CS	4-bit input/output	• External memory interface chip select signals
CT	2-bit input/output	• External memory interface read/write strobe
CM	4-bit input/output	• Bus clock output from MEMC • External memory interface data wait request

Table 2-1 Functions of each port group (2/2)

Port group name	Function	
	Port mode	Alternative mode
CD	4-bit input/output	• Byte enable output of external data bus
DL	16-bit input/output	• External memory interface data lines 0 to 15
DH	16-bit input/output	• External memory interface data lines 16 to 31

Pin configuration

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

- For a general description of the registers, see *"Port Group Configuration Registers" on page 36*.
- For every port, detailed information on the configuration registers is given in *"Port Group Configuration" on page 49*.

There are several types of control circuits, defined as port types. For a description of the port types, see *"Port Types Diagrams" on page 55*.

2.1.2 Terms

In this section, the following terms are used:

- **Pin**

  Denotes the physical pin. Every pin is uniquely denoted by its pin number.
  A pin can be used in several modes. Depending on the selected mode, a pin name is allocated to the pin.

- **Port group**

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

- **Port mode / Port**

  A pin in port mode works as a general purpose input/output pin. It is then called “port”.
  The corresponding name is Pnm. For example, P07 denotes port 7 of port group 0. It is referenced as “port P07”.

- **Alternative mode**

  In alternative mode, a pin can work in various non-general purpose input/output functions, for example, as the input/output pin of on-chip peripherals.
  The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.
  Note that for example P01 and INTP0 denote the same physical pin. The different names indicate the function in which the pin is being operated.

- **Port type**

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

2.1.3 Noise elimination

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

See *“Noise Elimination” on page 87* for a detailed description.

2.2 Port Group Configuration Registers

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

- *“Pin function configuration” on page 37*
- *“Pin data input/output” on page 41*
- *“Emergency shut-off configuration” on page 43*
- *“Write enable register” on page 47*

2.2.1 Overview

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

Table 2-2 Registers for port group configuration

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

n = 0 to 11, AL, AH, DL, DH, CS, CT, CM and CD

2.2.2 Pin function configuration

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

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

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

Table 2-3 Pin function configuration (overview)

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

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

(1) PMCn - Port mode control register

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

For port groups with up to eight ports, this is an 8-bit register. For port groups with up to 16 ports, this is a 16-bit register.

Access This register can be read/written in 8-bit and 1-bit units.
16-bit registers can also be read/written in 16-bit units.

Address see *"Port Group Configuration" on page 49*

Initial Value 00_H or 0000_H. This register is initialized by $\overline{\text{RESET}}$.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCn15	PMCn14	PMCn13	PMCn12	PMCn11	PMCn10	PMCn9	PMCn8	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0
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 2-4 PMCn register contents

Bit position	Bit name	Function
7 to 0 or 15 to 0	PMCn[7:0] or PMC[15:0]	Specifies the operation mode of the corresponding pin 0: Port mode 1: Alternative mode

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

Therefore, it is recommended to follow the below procedure:

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

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

(2) PMn - Port mode register

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

For port groups with up to eight ports, this is an 8-bit register. For port groups with up to 16 ports, this is a 16-bit register.

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

Access This register can be read/written in 8-bit and 1-bit units.
16-bit registers can also be read/written in 16-bit units.

Address see *"Port Group Configuration" on page 49*

Initial Value FF_H or $FFFF_H$. This register is initialized by $\overline{RESET}$.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMn15	PMn14	PMn13	PMn12	PMn11	PMn10	PMn9	PMn8	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0
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 2-5 PMn register contents

Bit position	Bit name	Function
7 to 0 or 15 to 0	PMn[7:0] or PMn[15:0]	Specifies input/output mode of the corresponding pin in port mode 0: Output mode (output enabled) 1: Input mode (output disabled)

(3) PFCn - Port function control register

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

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

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

Address see *"Port Group Configuration" on page 49*

Initial Value 00_H

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

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

Table 2-6 PFCn register contents

Bit position	Bit name	Function
7 to 0	PFCn[7:0]	See *Table 2-3 on page 37* for details

(4) PFCEn - Port function control expansion register

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

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

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

Address see *"Port Group Configuration" on page 49*

Initial Value 00_H

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

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

Table 2-7 PFCEn register contents

Bit position	Bit name	Function
7 to 0	PFCn[7:0]	See *Table 2-3 on page 37* for details

2.2.3 Pin data input/output

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

(1) Pn - Port register

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

For port groups with up to eight ports, this is an 8-bit register. For port groups with up to 16 ports, this is a 16-bit register.

Access This register can be read/written in 8-bit and 1-bit units.
16-bit registers can also be read/written in 16-bit units.

Address see *"Port Group Configuration" on page 49*

Initial Value Undefined.

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pn15	Pn14	Pn13	Pn12	Pn11	Pn10	Pn9	Pn8	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0
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 2-8 Pn register contents

Bit position	Bit name	Function
7 to 0 or 15 to 0	Pn[7:0] or Pn[15:0]	Data, see *Table 2-9 on page 41* and *Table 2-10 on page 42* for details.

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

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

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

Function	PM	I/O
Write to Pn…		
…and output contents of Pn to pins	0	O
…without affecting the pin status	1	I
Read from Pn…		
…and thus read the pin status	1	I
…and disregard the pin status	0	O

Alternative mode

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

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

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

Function	PM	I/O
Write to Pn without affecting the pin status	X	–
Read from Pn…		
…and read the value of the alternative output function (for pins in alternative output function)	0	–
…and disregard the pin status (for pins in alternative input function)		
…and thus read the pin status	1	I

Caution

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

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

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

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

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

Example:

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

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

2.2.4 Emergency shut-off configuration

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

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

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

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

(1) PESCn - Emergency shut-off control register

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

Writing to this register is protected by a special sequence of instructions. To enable write access to PESCn, first write to PRCMD. Please refer to *"Write Protected Registers" on page 123* for details.

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

Address see *"Port Group Configuration" on page 49*

Initial Value 00_H. This register is initialized by $\overline{\text{RESET}}$.

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

Table 2-11 PESCn register contents

Bit Position	Bit Name	Function
3	ESOmEN	Enables/disables the emergency shut-off control by ESOm input 0: disabled 1: enabled
1 to 0	ESOmED [1:0]	Sets the edge or level of input signal ESOm for triggering an emergency shut-off.

ESOmED1	ESOmED0	Selected edge/level
0	0	falling edge
0	1	rising edge
1	0	low level
1	1	high level

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

Note

1. The emergency shut-off function is not dedicated to a single port of a port group, but acts on the *entire* port group. However, note that not all ports of a port group may be subject to the emergency shut-off function. Refer to the description of the concerned port group.
2. If the emergency shut-off condition is fulfilled, the concerned pin is also switched to high impedance state if the port is configured as port output (PMCnm = 0, PMnm = 0).
3. The output buffer of the concerned ports Pnm are forcibly disabled (high impedance output) as long as PESCn.ESOmEN and ESOSTn.ESOmST are set to 1.

Setting up an emergency shut-off

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

1. Power on (all registers are reset)
2. PRCMD write (write protect released)
3. Clear ESOmEN bit to 0
4. PRCMD write (write protect released)
5. Set ESOmED0, ESOmED1 bits
6. PRCMD write (write protect released)
7. Clear ESOmST bit of ESOSTn register to 0
8. PRCMD write (write protect released)
9. Set PESMKnm bits to 0 respectively 1
10. PRCMD write (write protect released)
11. Set ESOmEN bit to 1

(2) ESOSTn - Emergency shut-off status register

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

Writing to this register is protected by a special sequence of instructions. To enable write access to ESOSTn, first write to PRCMD. Please refer to *"Write Protected Registers" on page 123* for details.

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

Address see *"Port Group Configuration" on page 49*

Initial Value 00_H. This register is initialized by $\overline{\text{RESET}}$.

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

Table 2-12 ESOSTn register contents

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

Note

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

(3) PESMKn - Emergency shut-off mask register

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

Writing to this register is protected by a special sequence of instructions. To enable write access to PESMKn, first write to PRCMD. Please refer to *"Write Protected Registers" on page 123* for details.

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

Address see *"Port Group Configuration" on page 49*

Initial Value 00_H. This register is initialized by $\overline{\text{RESET}}$.

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

Table 2-13 PESMKn register contents

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

2.2.5 Write enable register

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

(1) PRCMD - Command register

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

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

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

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

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

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

Address FFFF F1FC$_H$.

Initial Value The contents of this register is undefined.

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

(2) PHS - Peripheral status register

The 8-bit PHS register indicates the status of a write attempt to a register protected by PHCMD (see also *"PRCMD - Command register" on page 47*).

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

Address FFFF $F802_H$.

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

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

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

Table 2-14 PHS register contents

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

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

2.3 Port Group Configuration

This section provides an overview of the port groups (*Table 2-15*) and of the pin functions (*Table 2-17 on page 56*). In *Table 2-63 on page 91* it is listed how the pin functions change if the microcontroller is reset or if it is in one of the standby modes.

In the subsections, for every port group the settings of the configuration registers is listed. Further, the addresses and initial values of the configuration registers are given. See *"Port group 0" on page 64* to *"Port group DH" on page 85*.

For the port type diagrams refer to *"Port Types Diagrams" on page 55*.

2.3.1 Port group configuration lists

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

Table 2-15 Port group list (1/5)

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin location
0	P00	–	NMI	2	J-23
	P01	–	ESO0/INTP0	2	J-24
	P02	–	ESO1/INTP1	2	H-24
	P03	–	ADTRG0/INTP2	2	H-23
	P04	–	ADTRG1/INTP3	2	G-23
1	P10	TOAA00	TIAA00/TEVTAA1	5-K	AC-11
	P11	TOAA01	TIAA01/TTRGAA1	5-K	AD-11
	P12	TOAA10	TIAA10/TTRGAA0	5-K	AD-12
	P13	TOAA11	TIAA11/TEVTAA0	5-K	AC12
	P14	TOAA20	TIAA20/TEVTAA3	5-K	AB-14
	P15	TOAA21	TIAA21/TTRGAA3	5-K	AC-13
	P16	TOAA30	TIAA30/TTRGAA2	5-K	AD-13
	P17	TOAA31	TIAA31/TEVTAA2	5-K	AA-14
2	P20	TOAA40	TIAA40/TEVTAA5	5-K	AC-14
	P21	TOAA41	TIAA41/TTRGAA5	5-K	AD-14
	P22	TOAA50	TIAA50/TTRGAA4	5-K	AA-15
	P23	TOAA51	TIAA51/TEVTAA4	5-K	AB-15
	P24	TOAA60	TIAA60/TEVTAA7	5-K	AC-15
	P25	TOAA61	TIAA61/TTRGAA7	5-K	AD-15
	P26	TOAA70	TIAA70/TTRGAA6	5-K	AC-16
	P27	TOAA71	TIAA71/TEVTAA6	5-K	AD-16

Table 2-15 Port group list (2/5)

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin location
3	P30	–	RXDC0/INTP4	5-K	AC-20
	P31	TXDC0	–	5-K	AD-20
	P32	–	RXDC1/INTP5	5-K	AC-21
	P33	TXDC1	–	5-K	AD-21
	P34	–	FCRXD0	5-K	V-23
	P35	FCTXD0	–	5-K	V-24
	P36	–	FCRXD1	5-K	U-23
	P37	FCTXD1	–	5-K	U-24
4	P40	–	SIB0/RXDC2/INTP13	5-K	M-23
	P41	SOB0/TXDC2	–	5-K	M-24
	P42	SCKB0	SCKB0	5-K	L-23
	P43	–	SIB1	5-K	L-24
	P44	SOB1	–	5-K	K-23
	P45	SCKB1	SCKB1	5-K	K-24
5	P50	TOS00	–	5-K	T-23
	P51	TOS01	–	5-K	T-24
	P52	TOS02	–	5-K	R-23
	P53	TOS03	–	5-K	R-24
	P54	TOS04	–	5-K	P-23
	P55	TOS05	–	5-K	P-24
	P56	TOS06	–	5-K	N-23
	P57	TOS07	–	5-K	N-24
6	P60	TOS10	–	5-K	AC-17
	P61	TOS11	–	5-K	AD-17
	P62	TOS12	–	5-K	AD-18
	P63	TOS13	–	5-K	AC-18
	P64	TOS14	–	5-K	AB-18
	P65	TOS15	–	5-K	AA-18
	P66	TOS16	–	5-K	AC-19
	P67	TOS17	–	5-K	AD-19
7	P70	TOT00	TIT00/TEVTT1 TENCT00	5-K	A-15
	P71	TOT01	TIT01/TTRGT1 TENCT01	5-K	B-14
	P72	–	TECRT0/INTP12	5-K	A-14
	P73	TOT10	TIT10/TTRGT0 TENCT11	5-K	C-13
	P74	TOT11	TIT11/TEVTT0 TENCT10	5-K	D-13
	P75	AFO	TECRT1	5-K	B-13

Table 2-15 Port group list (3/5)

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin location
8	P80	–	SIE0	5-K	G-24
	P81	SOE0	–	5-K	F-23
	P82	SCKE0	SCKE0	5-K	F-24
	P83	SCSE00	INTP6	5-K	E-23
	P84	SCSE01	INTP7	5-K	E-24
	P85	SCSE02	INTP8	5-K	D-23
	P86	SCSE03	SSB0	5-K	D-24
9	P90	–	SIE1	5-K	B-18
	P91	SOE1	–	5-K	A-18
	P92	SCKE1	SCKE1	5-K	A-17
	P93	SCSE10/SCSE04	INTP9	5-K	B-17
	P94	SCSE11/SCSE05/ FTINT0	INTP10	5-K	A-16
	P95	SCSE12/SCSE06	INTP11/FSTPWT	5-K	B-16
	P96	SCSE13/SCSE07	SSB1	5-K	B-15
10	P100	TOAA81	TIAA80/TEVTAA8	5-K	A-13
	P101	TOAA90	TIAA90/TTRGAA9	5-K	B-12
	P102	TOAA91	TIAA91/TEVTAA9	5-K	A-12
11	P110	–	TAPTS00/TEVTS0	5-K	B-11
	P111	–	TAPTS01/TTRGS 0	5-K	A-11
	P112	–	TAPTS02	5-K	B-10
	P113	–	TAPTS10/TEVTS1	5-K	A-10
	P114	–	TAPTS11/TTRGS1	5-K	A-9
	P115	–	TAPTS12	5-K	B-9
AL	PAL0	A0	–	5	C-2
	PAL1	A1	–	5	C-1
	PAL2	A2	–	5	D-2
	PAL3	A3	–	5	D-1
	PAL4	A4	–	5	E-2
	PAL5	A5	–	5	E-1
	PAL6	A6	–	5	F-2
	PAL7	A7	–	5	G-3
	PAL8	A8	–	5	F-1
	PAL9	A9	–	5	G-4
	PAL10	A10	–	5	G-2
	PAL11	A11	–	5	G-1
	PAL12	A12	–	5	H-3
	PAL13	A13	–	5	H-4
	PAL14	A14	–	5	H-2
	PAL15	A15	–	5	H-1

Table 2-15 Port group list (4/5)

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin location
AH	PAH0	A16	–	5	J-2
	PAH1	A17	–	5	J1
	PAH2	A18	–	5	K-2
	PAH3	A19	–	5	K-1
	PAH4	A20	–	5	L-4
	PAH5	A21	–	5	L-3
CS	PCS0	$\overline{CS0}$	–	5	B-8
	PCS2	$\overline{CS2}$	–	5	C-6
	PCS3	$\overline{CS3}$	–	5	A-8
	PCS4	$\overline{CS4}$	–	5	D-7
CT	PCT4	$\overline{RD}$	–	5	B-7
	PCT5	$\overline{WR}$	–	5	A-7
CM	PCM0	–	$\overline{WAIT}$	5	B-6
	PCM1	–	–	5	A-6
	PCM6	–	–	5	A-5
	PCM7	–	–	5	B-5
CD	PCD2	$\overline{BEN0}$	–	5	B-4
	PCD3	$\overline{BEN1}$	–	5	A-4
	PCD4	$\overline{BEN2}$	–	5	B-3
	PCD5	$\overline{BEN3}$	–	5	A-3
DL	PDL0	D0	D0	5	L-2
	PDL1	D1	D1	5	L-1
	PDL2	D2	D2	5	M-4
	PDL3	D3	D3	5	M-3
	PDL4	D4	D4	5	M-2
	PDL5	D5	D5	5	M-1
	PDL6	D6	D6	5	R-3
	PDL7	D7	D7	5	N-2
	PDL8	D8	D8	5	N-1
	PDL9	D9	D9	5	T-3
	PDL10	D10	D10	5	P-2
	PDL11	D11	D11	5	P-1
	PDL12	D12	D12	5	R-2
	PDL13	D13	D13	5	R-1
	PDL14	D14	D14	5	T-1
	PDL15	D15	D15	5	T-2

Table 2-15 Port group list (5/5)

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin location
DH	PDH0	D16	D16	5	U-4
	PDH1	D17	D17	5	U-3
	PDH2	D18	D18	5	U-1
	PDH3	D19	D19	5	U-2
	PDH4	D20	D20	5	V-1
	PDH5	D21	D21	5	V-2
	PDH6	D22	D22	5	W-1
	PDH7	D23	D23	5	W-2
	PDH8	D24	D24	5	Y-1
	PDH9	D25	D25	5	W-3
	PDH10	D26	D26	5	Y-2
	PDH11	D27	D27	5	W-4
	PDH12	D28	D28	5	AA-1
	PDH13	D29	D29	5	AA-2
	PDH14	D30	D30	5	AB-1
	PDH15	D31	D31	5	AB-2

2.3.2 Non-port pins

Table 2-16 lists all pins which do not have a general purpose I/O port function.

Table 2-16 Non-port pins

Name	Port type	Pin location
AD0_DBG	5-K	D-14
AD1_DBG	5-K	C-14
AD2_DBG	5-K	C-7
AD3_DBG	5-K	D-6
ANI00	7	AD-3
ANI01	7	AC-3
ANI02	7	AD-4
ANI03	7	AC-4
ANI04	7	AD-5
ANI05	7	AC-5
ANI06	7	AB-6
ANI07	7	AA-6
ANI08	7	AB-7
ANI09	7	AA-7
ANI10	7	AA-10
ANI11	7	AB-10
ANI12	7	AA-9
ANI13	7	AB-9

Table 2-16 Non-port pins

Name	Port type	Pin location
ANI14	7	AC-9
ANI15	7	AD-9
ANI16	7	AA-8
ANI17	7	AB-8
ANI18	7	AC-8
ANI19	7	AD-8
CLK_DBG	2	D-11
DCK	2	M-22
DDI	2	L-22
DDO	5	P-22
DMS	2	N-21
$\overline{\text{DRTS}}$	2-I	L-21
FRXDA	2	B-22
FRXDB	2	B-21
FTXDA	3	A-22
FTXDB	3	A-21
FTXENA	3	A20
FTXENB	3	A-19
MODE_DBG	2-I	C-11
MODE0/FLMD1	2	H-21
MODE1/FLMD0	2	G-22
MODE2	2	G-21
MODE3	2	H-22
PLLSEL	2	U-22
$\overline{\text{RESET}}$	2	N-22
SYNC	2	C-17
TRIG_DBG	4	C-16

2.3.3 Port Types Diagrams

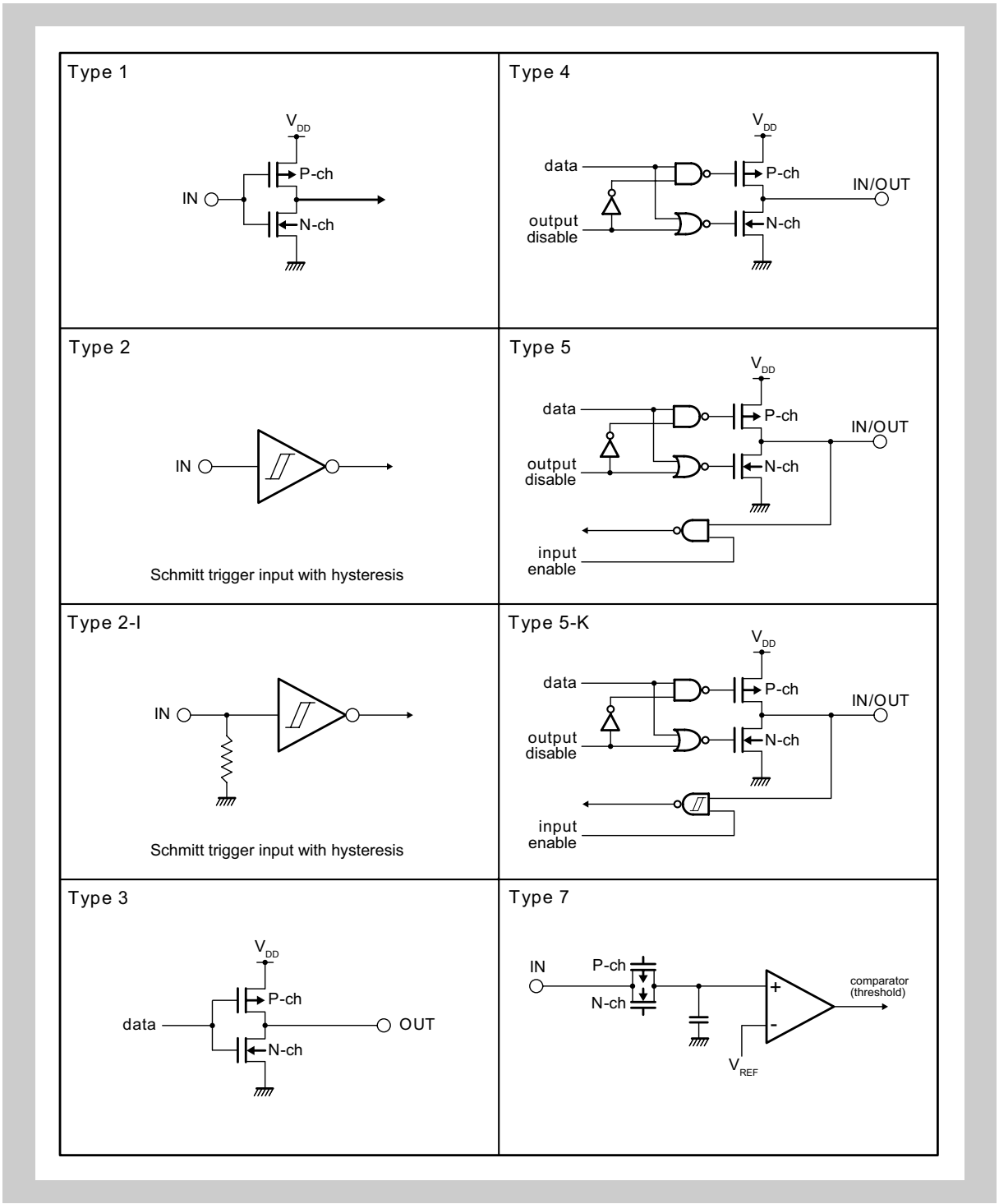


Figure 2-2 Port type diagrams

2.3.4 Alphabetic pin function list

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

Table 2-17 Alphabetic pin functions list (1/8)

Pin name	I/O	Pin function	Port	Pin location
A0	O	External memory interface address bus	PAL0	C-2
A1			PAL1	C-1
A2			PAL2	D-2
A3			PAL3	D-1
A4			PAL4	E-2
A5			PAL5	E-1
A6			PAL6	F-2
A7			PAL7	G-3
A8			PAL8	F-1
A9			PAL9	G-4
A10			PAL10	G-2
A11			PAL11	G-1
A12			PAL12	H-3
A13			PAL13	H-4
A14			PAL14	H-2
A15			PAL15	H-1
A16			PAH0	J-2
A17			PAH1	J1
A18			PAH2	K-2
A19			PAH3	K-1
A20			PAH4	L-4
A21			PAH5	L-3
AD0_DBG	I/O	NBD command/data	–	D-14
AD1_DBG			–	C-14
AD2_DBG			–	C-7
AD3_DBG			–	D-6
ADTRG0	I	ADC0 trigger input	P03	H-23
ADTRG1	I	ADC1 trigger input	P04	G-23
AFO	O	Auxiliary frequency output	P75	B-13

Table 2-17 Alphabetic pin functions list (2/8)

Pin name	I/O	Pin function	Port	Pin location
ANI00	I	ADC inputs	–	AD-3
ANI01			–	AC-3
ANI02			–	AD-4
ANI03			–	AC-4
ANI04			–	AD-5
ANI05			–	AC-5
ANI06			–	AB-6
ANI07			–	AA-6
ANI08			–	AB-7
ANI09			–	AA-7
ANI10			–	AA-10
ANI11			–	AB-10
ANI12			–	AA-9
ANI13			–	AB-9
ANI14			–	AC-9
ANI15			–	AD-9
ANI16			–	AA-8
ANI17			–	AB-8
ANI18			–	AC-8
ANI19			–	AD-8
AVDD	–	ADC supply voltage (3.3 V)	–	AC-6, AD-6
AVREF0	–	ADC0 reference voltage input	–	AD-7
AVREF1	–	ADC1 reference voltage input	–	AC-7
AVSS0	–	ADC0 ground	–	AA-5, AB-5
AVSS1	–	ADC1 ground	–	AC-10, AD-10
$\overline{\text{BEN0}}$	O	Byte enable output of external data bus (D0-D7)	PCD2	B-4
$\overline{\text{BEN1}}$	O	Byte enable output of external data bus (D8-D15)	PCD3	A-4
$\overline{\text{BEN2}}$	O	Byte enable output of external data bus (D16-D23)	PCD4	B-3
$\overline{\text{BEN3}}$	O	Byte enable output of external data bus (D24-D31)	PCD5	A-3
CLK_DBG	I	NBD clock signal	–	D-11
$\overline{\text{CS0}}$	O	External memory interface chip select signals	PCS0	B-8
$\overline{\text{CS2}}$			PCS2	C-6
$\overline{\text{CS3}}$			PCS3	A-8
$\overline{\text{CS4}}$			PCS4	D-7
CVDD15	–	Oscillator supply voltage (1.5 V)	–	W-24, AA-24
CVSS15		Oscillator supply ground	–	Y-24, AB-24

Table 2-17 Alphabetic pin functions list (3/8)

Pin name	I/O	Pin function	Port	Pin location
D0	I/O	External memory interface data bus	PDL0	L-2
D1			PDL1	L-1
D2			PDL2	M-4
D3			PDL3	M-3
D4			PDL4	M-2
D5			PDL5	M-1
D6			PDL6	R-3
D7			PDL7	N-2
D8			PDL8	N-1
D9			PDL9	T-3
D10			PDL10	P-2
D11			PDL11	P-1
D12			PDL12	R-2
D13			PDL13	R-1
D14			PDL14	T-1
D15			PDL15	T-2
D16			PDH0	U-4
D17			PDH1	U-3
D18			PDH2	U-1
D19			PDH3	U-2
D20			PDH4	V-1
D21			PDH5	V-2
D22			PDH6	W-1
D23			PDH7	W-2
D24			PDH8	Y-1
D25			PDH9	W-3
D26			PDH10	Y-2
D27			PDH11	W-4
D28			PDH12	AA-1
D29			PDH13	AA-2
D30			PDH14	AB-1
D31			PDH15	AB-2
DCK	I	N-Wire interface clock	–	M-22
DDI	I	N-Wire interface debug data input	–	L-22
DDO	O	N-Wire interface debug data output	–	P-22
DMS	I	N-Wire interface debug mode select input	–	N-21
$\overline{\text{DRST}}$	I	N-Wire debug interface reset	–	L-21
ESO0	I	Emergency shut-off control of Timer TMS0	P01	J-24
ESO1	I	Emergency shut-off control of Timer TMS1	P02	H-24
FCRXD0	I	CAN0 receive data	P34	V-23
FCRXD1	I	CAN1 receive data	P36	U-23

Table 2-17 Alphabetic pin functions list (4/8)

Pin name	I/O	Pin function	Port	Pin location
FCTXD0	O	CAN0 transmit data	P35	V-24
FCTXD1	O	CAN1 transmit data	P37	U-24
FLMD0 (= MODE1)	I	Flash writing control	–	G-22
FLMD1 (= MODE0)	I	Flash writing control	–	H-21
FRXDA	I	FlexRay channel A receive data	–	B-22
FRXDB	I	FlexRay channel B receive data	–	B-21
FSTPWT	I	Interrupt signal from external FlexRay	P95	B-16
FTINT0	O	Interrupt signal from internal FlexRay	P94	A-16
FTXDA	O	FlexRay channel A transmit data	–	A-22
FTXDB	O	FlexRay channel B transmit data	–	A-21
FTXENA	O	FlexRay channel A transmit permission	–	A-20
FTXENB	O	FlexRay channel B transmit permission	–	A-19
INTP0	I	External interrupts	P01	J-24
INTP1			P02	H-24
INTP2			P03	H-23
INTP3			P04	G-23
INTP4			P30	AC-20
INTP5			P32	AD-20
INTP6			P83	E-23
INTP7			P84	E-24
INTP8			P85	D-23
INTP9			P93	B-17
INTP10			P94	A-16
INTP11			P95	B-16
INTP12			P72	A-14
INTP13			P40	M-23
MODE_DBG	I	NBD execution permission	–	C-11
MODE0	I	Operating mode select pins	–	H-21
MODE1			–	G-22
MODE2			–	G-21
MODE3			–	H-22
NMI	I	Non-maskable interrupt	P00	J-23
PLLSEL		PLL selection (80 MHz or 128 MHz)	–	U-22
$\overline{RD}$	O	External memory interface read strobe	PCT4	B-7
$\overline{RESET}$	I	Reset input	–	N-22
RXDC0	I	UARTC0 receive data	P30	AC-20
RXDC1	I	UARTC1 receive data	P32	AC-21
RXDC2	I	UARTC2 receive data	P40	M-23
SCKB0	I/O	Clocked Serial Interface CSIB0 clock line	P42	L-23
SCKB1	I/O	Clocked Serial Interface CSIB1 clock line	P45	K-24

Table 2-17 Alphabetic pin functions list (5/8)

Pin name	I/O	Pin function	Port	Pin location
SCKE0	I/O	CSIE0 clock line	P82	F-24
SCKE1	I/O	CSIE1 clock line	P92	A-17
SCSE00	O	CSIE0 chip selects	P83	E-23
SCSE01			P84	E-24
SCSE02			P85	D-23
SCSE03			P86	D-24
SCSE04			P93	B-17
SCSE05			P94	A-16
SCSE06			P95	B-16
SCSE07			P96	B-15
SCSE10	O	CSIE1 chip selects	P93	B-17
SCSE11			P94	A-16
SCSE12			P95	B-16
SCSE13			P96	B-15
SIB0	I	CSIB0 data input	P40	M-23
SIB1	I	CSIB1 data input	P43	L-24
SIE0	I	CSIE0 data input	P80	G-24
SIE1	I	CSIE1 data input	P90	B-18
SOB0	O	CSIB0 data output	P41	M-24
SOB1	O	CSIB1 data output	P44	K-23
SOE0	O	CSIE0 data output	P81	F-23
SOE1	O	CSIE1 data output	P91	A-18
SSB0	I	CSIB0 serial slave select input	P86	D-24
SSB1	I	CSIB1 serial slave select input	P96	B-15
SYNC	I	NBD synchronizing signal	–	C-17
TAPTS00	I	Timer TMS0 trigger signals	P110	B-11
TAPTS01			P111	A-11
TAPTS02			P112	B-10
TAPTS10	I	Timer TMS1 trigger signals	P113	A-10
TAPTS11			P114	A-9
TAPTS12			P115	B-9
TECRT0	I	Timer TMT0 clear input signal	P72	A-14
TECRT1	I	Timer TMT1 clear input signal	P75	B-13
TENCT00	I	Timer TMT0 encoder input	P70	A-15
TENCT01			P71	B-14
TENCT10	I	Timer TMT1 encoder input	P74	D-13
TENCT11			P73	C-13
TEVTAA0	I	Timer TAA0 event input	P13	AC-12
TEVTAA1	I	Timer TAA1 event input	P10	AC-11
TEVTAA2	I	Timer TAA2 event input	P17	AA-14
TEVTAA3	I	Timer TAA3 event input	P14	AB-14

Table 2-17 Alphabetic pin functions list (6/8)

Pin name	I/O	Pin function	Port	Pin location
TEVTAA4	I	Timer TAA4 event input	P23	AB-15
TEVTAA5	I	Timer TAA5 event input	P20	AC-14
TEVTAA6	I	Timer TAA6 event input	P27	AD-16
TEVTAA7	I	Timer TAA7 event input	P24	AC-15
TEVTAA8	I	Timer TAA8 event input	P100	A-13
TEVTAA9	I	Timer TAA9 event input	P102	A-12
TEVTS0	I	Timer TMS0 event input	P110	B-11
TEVTS1	I	Timer TMS1 event input	P113	A-10
TEVTT0	I	Timer TMT0 event input	P74	D-13
TEVTT1	I	Timer TMT1 event input	P70	A-15
TIAA00	I	Timer TAA0 capture trigger input	P10	AC-11
TIAA01			P11	AD-11
TIAA10	I	Timer TAA1 capture trigger input	P12	AD-12
TIAA11			P13	AC-12
TIAA20	I	Timer TAA2 capture trigger input	P14	AB-14
TIAA21			P15	AC-13
TIAA30	I	Timer TAA3 capture trigger input	P16	AD-13
TIAA31			P17	AA-14
TIAA40	I	Timer TAA4 capture trigger input	P20	AC-14
TIAA41			P21	AD-14
TIAA50	I	Timer TAA5 capture trigger input	P22	AA-15
TIAA51			P23	AB-15
TIAA60	I	Timer TAA6 capture trigger input	P24	AC-15
TIAA61			P25	AD-15
TIAA70	I	Timer TAA7 capture trigger input	P26	AC-16
TIAA71			P27	AD-16
TIAA80	I	Timer TAA8 capture trigger input	P100	A-13
TIAA90	I	Timer TAA9 capture trigger input	P101	B-12
TIAA91			P102	A-12
TIT00	I	Timer TMT0 capture trigger input	P70	A-15
TIT01			P71	B-14
TIT10	I	Timer TMT1 capture trigger input	P73	C-13
TIT11			P74	D-13
TOAA00	O	Timer TAA0 pulse signal output	P10	AC-11
TOAA01			P11	AD-11
TOAA10	O	Timer TAA1 pulse signal output	P12	AD-12
TOAA11			P13	AC-12
TOAA20	O	Timer TAA2 pulse signal output	P14	AB-14
TOAA21			P15	AC-13
TOAA30	O	Timer TAA3 pulse signal output	P16	AD-13
TOAA31			P17	AA-14

Table 2-17 Alphabetic pin functions list (7/8)

Pin name	I/O	Pin function	Port	Pin location
TOAA40	O	Timer TAA4 pulse signal output	P20	AC-14
TOAA41			P21	AD-14
TOAA50	O	Timer TAA5 pulse signal output	P22	AA-15
TOAA51			P23	AB-15
TOAA60	O	Timer TAA6 pulse signal output	P24	AC-15
TOAA61			P25	AD-15
TOAA70	O	Timer TAA7 pulse signal output	P26	AC-16
TOAA71			P27	AD-16
TOAA81	O	Timer TAA8 pulse signal output	P100	A-13
TOAA90	O	Timer TAA9 pulse signal output	P101	B-12
TOAA91			P102	A-12
TOS00	O	Timer TMS0 pulse signal outputs Outputs TOS01 to TOS06 can be controlled by an emergency shut-off. See *"Emergency shut-off configuration" on page 43*.	P50	T-23
TOS01			P51	T-24
TOS02			P52	R-23
TOS03			P53	R-24
TOS04			P54	P-23
TOS05			P55	P-24
TOS06			P56	N-23
TOS07			P57	N-24
TOS10	O	Timer TMS1 pulse signal outputs Outputs TOS11 to TOS16 can be controlled by an emergency shut-off. See *"Emergency shut-off configuration" on page 43*.	P60	AC-17
TOS11			P61	AD-17
TOS12			P62	AD-18
TOS13			P63	AC-18
TOS14			P64	AB-18
TOS15			P65	AA-18
TOS16			P66	AC-19
TOS17			P67	AD-19
TOT00	O	Timer TMT0 pulse signal output	P70	A-15
TOT01			P71	B-14
TOT10	O	Timer TMT1 pulse signal output	P73	C-13
TOT11			P74	D-13
TRIG_DBG	O	NBD event trigger output	–	C-16
TTRGAA0	I	Timer TAA0 trigger input	P12	AD-12
TTRGAA1	I	Timer TAA1 trigger input	P11	AD-11
TTRGAA2	I	Timer TAA2 trigger input	P16	AD-13
TTRGAA3	I	Timer TAA3 trigger input	P15	AC-13
TTRGAA4	I	Timer TAA4 trigger input	P22	AA-15
TTRGAA5	I	Timer TAA5 trigger input	P21	AD-14
TTRGAA6	I	Timer TAA6 trigger input	P26	AC-16
TTRGAA7	I	Timer TAA7 trigger input	P25	AD-15
TTRGAA9	I	Timer TAA9 trigger input	P101	B-12

Table 2-17 Alphabetic pin functions list (8/8)

Pin name	I/O	Pin function	Port	Pin location
TTRGS0	I	Timer TMS0 trigger input	P111	A-11
TTRGS1	I	Timer TMS1 trigger input	P114	A-9
TTRGT0	I	Timer TMT0 trigger input	P73	C-13
TTRGT1	I	Timer TMT1 trigger input	P71	B-14
TXDC0	O	UARTC0 transmit data	P31	AD-20
TXDC1	O	UARTC1 transmit data	P33	AD-21
TXDC2	O	UARTC2 transmit data	P41	M-24
VDD15x	–	Core supply voltage	–	refer to *Figure 2-4 on page 95*
VDD3x	–	I/O supply voltage	–	
VSS15x	–	Core supply ground	–	
VSS3x	–	I/O supply ground	–	
$\overline{\text{WAIT}}$	I	External memory interface data wait request	PCM0	B-6
$\overline{\text{WR}}$	O	External memory interface write strobe	PCT5	A-7
X1	–	Main oscillator terminals Crystal connection/external clock input for system clock oscillation (X2 opens when the external clock is input)	–	W-23
X2			–	Y-23

2.3.5 Port group 0

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

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

Port group 0 includes the following pins:

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

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

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

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

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

Table 2-19 Port group 0: configuration registers

Register	Address	Initial value	Used bits							
P0[a]	FFFF F400$_H$	–	0	0	0	P04	P03	P02	P01	P00

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

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

2.3.6 Port group 1

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

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

Port group 1 includes the following pins:

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

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

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

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

Table 2-21 Port group 1: configuration registers

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

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

2.3.7 Port group 2

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

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

Port group 2 includes the following pins:

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

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

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

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

Table 2-23 Port group 2: configuration registers

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

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

2.3.8 Port group 3

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

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

Port group 3 includes the following pins:

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

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

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

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

Table 2-25 Port group 3: configuration registers

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

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

2.3.9 Port group 4

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

- External interrupt (INTP13)
- UARTC2 receive/transmit data (RXDC2, TXDC2)
- Clocked Serial Interface CSIB0 data/clock line (SIB0, SOB0, SCKB0)
- Clocked Serial Interface CSIB1 data/clock line (SIB1, SOB1, SCKB1)

Port group 4 includes the following pins:

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

Pin functions in different modes			Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P40 (I/O)	SIB0 (I)	RXDC2/INTP13 (I)	P40 (I)	5-K	D	S
P41 (I/O)	SOB0 (O)	TXDC2 (O)	P41 (I)	5-K	–	S
P42 (I/O)	SCKB0 (I/O)	prohibited	P42 (I)	5-K	–	S
P43 (I/O)	SIB1 (I)	prohibited	P43 (I)	5-K	–	S
P44 (I/O)	SOB1 (O)	prohibited	P44 (I)	5-K	–	S
P45 (I/O)	SCKB1 (I/O)	prohibited	P45 (I)	5-K	–	S

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

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

Table 2-27 Port group 4: configuration registers

Register	Address	Initial value	Used bits							
PMC4	FFFF F448$_H$	00$_H$	0	0	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40
PM4	FFFF F428$_H$	FF$_H$	1	1	PM45	PM44	PM43	PM42	PM41	PM40
PFC4	FFFF F468$_H$	00$_H$	0	0	0	0	0	0	PFC41	PFC40
P4	FFFF F408$_H$	00$_H$	0	0	P45	P44	P43	P42	P41	P40

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

2.3.10 Port group 5

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

- Timer TMS0 pulse signal output (TOS00 to TOS07)

Note The outputs TOS01 to TOS06 of timer TMS0 can be controlled by an emergency shut-off. See *"Emergency shut-off configuration" on page 43* for details.

Port group 5 includes the following pins:

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

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

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

Table 2-29 Port group 5: configuration registers

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

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

Note Writing to registers PMC5, PM5, P5, PESC5, ESOST5 and RESMK5 is protected by a special sequence of instructions. Please refer to *"Write enable register" on page 47* for details.

2.3.11 Port group 6

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

- Timer TMS1 pulse signal output (TOS10 to TOS17)

Note The outputs TOS11 to TOS16 of timer TMS1 can be controlled by an emergency shut-off. See *"Emergency shut-off configuration" on page 43* for details.

Port group 6 includes the following pins:

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

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

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

Table 2-31 Port group 6: configuration registers

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

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

Note Writing to registers PMC6, PM6, P6, PESC6, ESOST6 and and RESMK6 is protected by a special sequence of instructions. Please refer to *"Write enable register" on page 47* for details.

2.3.12 Port group 7

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

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

Port group 7 includes the following pins:

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

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

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

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

Table 2-33 Port group 7: configuration registers

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

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

2.3.13 Port group 8

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

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

Port group 8 includes the following pins:

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

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

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

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

Table 2-35 Port group 8: configuration registers

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

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

2.3.14 Port group 9

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

- Clocked Serial Interface CSIE0 chip select signal (SCSE04 to SCSE07)
- Clocked Serial Interface CSIE1 data/clock line (SIE1, SOE1, SCKE1)
- Clocked Serial Interface CSIE1 chip select signal (SCSE10 to SCSE13)
- Clocked Serial Interface CSIB1 serial slave select input (SSB1)
- External interrupt (INTP9 to INTP11)
- FlexRay external/internal interrupt (FSTPWT, FTINT0)

Port group 9 includes the following pins:

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

Pin functions in different modes					Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)							
	PFCE = 0		PFCE = 1					
	Function 1 PFC = 0	Function 2 PFC = 1	Function 3 PFC = 0	Function 4 PFC = 1				
P90 (I/O)	SIE1 (I)	prohibited	prohibited	prohibited	P90 (I)	5-K	–	S
P91 (I/O)	SOE1 (O)	prohibited	prohibited	prohibited	P91 (I)	5-K	–	S
P92 (I/O)	SCKE1 (I/O)	prohibited	prohibited	prohibited	P92 (I)	5-K	–	S
P93 (I/O)	SCSE10 (O)	INTP9 (I)	SCSE04 (O)	prohibited	P93 (I)	5-K	D	S
P94 (I/O)	SCSE11 (O)	INTP10 (I)	SCSE05 (O)	FTINT0 (O)	P94 (I)	5-K	D	S
P95 (I/O)	SCSE12 (O)	INTP11 (I)/ FSTPWT (I)	SCSE06 (O)	prohibited	P95 (I)	5-K	D	S
P96 (I/O)	SCSE13 (O)	SSB1 (I)	SCSE07 (O)	prohibited	P96 (I)	5-K	–	S

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

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

Table 2-37 Port group 9: configuration registers

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

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

2.3.15 Port group 10

Port group 10 is a 3-bit port group. In alternative mode, it comprises pins for the following functions:

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

Port group 10 includes the following pins:

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

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

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

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

Table 2-39 Port group 10: configuration registers

Register	Address	Initial value	Used bits							
PMC10	FFFF F454$_H$	00$_H$	0	0	0	0	0	PMC102	PMC101	PMC100
PM10	FFFF F434$_H$	FF$_H$	1	1	1	1	1	PM102	PM101	PM100
PFC10	FFFF F474$_H$	00$_H$	0	0	0	0	0	PFC102	PFC101	PFC100
P10	FFFF F414$_H$	00$_H$	0	0	0	0	0	P102	P101	P100

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

2.3.16 Port group 11

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

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

Port group 11 includes the following pins:

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

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

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

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

Table 2-41 Port group 11: configuration registers

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

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

2.3.17 Port group AL

Port group AL is a 16-bit port group. In alternative mode, it comprises pins for the external memory interface address lines 0 to 15.

Port group AL includes the following pins:

Table 2-42 Port group AL: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PAL0 (I/O)	A0 (O)	PAL0 (I)	5	–	nS
PAL1 (I/O)	A1 (O)	PAL1 (I)	5	–	nS
PAL2 (I/O)	A2 (O)	PAL2 (I)	5	–	nS
PAL3 (I/O)	A3 (O)	PAL3 (I)	5	–	nS
PAL4 (I/O)	A4 (O)	PAL4 (I)	5	–	nS
PAL5 (I/O)	A5 (O)	PAL5 (I)	5	–	nS
PAL6 (I/O)	A6 (O)	PAL6 (I)	5	–	nS
PAL7 (I/O)	A7 (O)	PAL7 (I)	5	–	nS
PAL8 (I/O)	A8 (O)	PAL8 (I)	5	–	nS
PAL9 (I/O)	A9 (O)	PAL9 (I)	5	–	nS
PAL10 (I/O)	A10 (O)	PAL10 (I)	5	–	nS
PAL11 (I/O)	A11 (O)	PAL11 (I)	5	–	nS
PAL12 (I/O)	A12 (O)	PAL12 (I)	5	–	nS
PAL13 (I/O)	A13 (O)	PAL13 (I)	5	–	nS
PAL14 (I/O)	A14 (O)	PAL14 (I)	5	–	nS
PAL15 (I/O)	A15 (O)	PAL15 (I)	5	–	nS

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

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

Table 2-43 Port group AL: configuration registers

Register	Address	Initial value	Used bits							
PMCALL	FFFF F040$_H$	00$_H$	PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0
PMCALH	FFFF F041$_H$	00$_H$	PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8
PMCAL (16 bit)	FFFF F040$_H$	0000$_H$	PMCAL15 to PMCAL8 (PMCALH)				PMCAL7 to PMCAL0 (PMCALL)			
PMALL	FFFF F020$_H$	FF$_H$	PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0
PMALH	FFFF F021$_H$	FF$_H$	PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8
PMAL (16 bit)	FFFF F020$_H$	FFFF$_H$	PMAL15 to PMAL8 (PMALH)				PMAL7 to PMAL0 (PMALL)			
PALL	FFFF F000$_H$	00$_H$	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
PALH	FFFF F001$_H$	00$_H$	PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8
PAL (16 bit)	FFFF F000$_H$	0000$_H$	PAL15 to PAL8 (PALH)				PAL7 to PAL0 (PALL)			

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

All 16-bit registers can be accessed in 16-bit units.

2.3.18 Port group AH

Port group AH is a 6-bit port group. In alternative mode, it comprises pins for the external memory interface address lines 16 to 21:

Port group AH includes the following pins:

Table 2-44 Port group AH: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PAH0 (I/O)	A16 (O)	PAH0 (I)	5	–	nS
PAH1 (I/O)	A17 (O)	PAH1 (I)	5	–	nS
PAH2 (I/O)	A18 (O)	PAH2 (I)	5	–	nS
PAH3 (I/O)	A19 (O)	PAH3 (I)	5	–	nS
PAH4 (I/O)	A20 (O)	PAH4 (I)	5	–	nS
PAH5 (I/O)	A21 (O)	PAH5 (I)	5	–	nS

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

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

Table 2-45 Port group AH: configuration registers

Register	Address	Initial value	Used bits							
PMCAH	FFFF F042$_H$	00$_H$	0	0	PMCAH5	PMCAH4	PMCAH3	PMCAH2	PMCAH1	PMCAH0
PMAH	FFFF F022$_H$	FF$_H$	1	1	PMAH5	PMAH4	PMAH3	PMAH2	PMAH1	PMAH0
PAH	FFFF F002$_H$	00$_H$	0	0	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0

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

2.3.19 Port group CS

Port group CS is a 4-bit port group. In alternative mode, it comprises pins for the external memory interface chip select signals ($\overline{CS0}$, $\overline{CS2}$, $\overline{CS3}$, $\overline{CS4}$)

Port group CS includes the following pins:

Table 2-46 Port group CS: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PCS0 (I/O)	$\overline{CS0}$ (O)	PCS0 (I)	5	–	nS
PCS2 (I/O)	$\overline{CS2}$ (O)	PCS2 (I)	5	–	nS
PCS3 (I/O)	$\overline{CS3}$ (O)	PCS3 (I)	5	–	nS
PCS4 (I/O)	$\overline{CS4}$ (O)	PCS4 (I)	5	–	nS

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

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

Table 2-47 Port group CS: configuration registers

Register	Address	Initial value	Used bits							
PMCCS	FFFF F048$_H$	00$_H$	0	0	0	PMCCS4	PMCCS3	PMCCS2	0	PMCCS0
PMCS	FFFF F028$_H$	FF$_H$	1	1	1	PMCS4	PMCS3	PMCS2	1	PMCS0
PCS	FFFF F008$_H$	00$_H$	0	0	0	PCS4	PCS3	PCS2	0	PCS0

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

2.3.20 Port group CT

Port group CT is a 2-bit port group. In alternative mode, it comprises pins for the external memory interface read/write strobe.

Port group CT includes the following pins:

Table 2-48 Port group CT: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PCT4 (I/O)	$\overline{RD}$ (O)	PCT4 (I)	5	–	nS
PCT5 (I/O)	$\overline{WR}$ (O)	PCT5 (I)	5	–	nS

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

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

Table 2-49 Port group CT: configuration registers

Register	Address	Initial value	Used bits							
PMCCT	FFFF F04A$_H$	00$_H$	0	0	PMCCT5	PMCCT4	0	0	0	0
PMCT	FFFF F02A$_H$	FF$_H$	1	1	PMCT5	PMCT4	1	1	1	1
PCT	FFFF F00A$_H$	00$_H$	0	0	PCT5	PCT4	0	0	0	0

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

2.3.21 Port group CM

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

- External memory interface data wait request ($\overline{WAIT}$)

Port group CM includes the following pins:

Table 2-50 Port group CM: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PCM0 (I/O)	$\overline{WAIT}$ (I)	PCM0 (I)	5	–	nS
PCM1 (I/O)	–	PCM1 (I)	5	–	nS
PCM6 (I/O)	–	PCM6 (I)	5	–	nS
PCM7 (I/O)	–	PCM7 (I)	5	–	nS

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

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

Table 2-51 Port group CM: configuration registers

Register	Address	Initial value	Used bits							
PMCCM	FFFF F04C$_H$	00$_H$	0[a]	0[a]	0	0	0	0	0[a]	PMCCM0
PMCM	FFFF F02C$_H$	FF$_H$	PMCM7	PMCM6	1	1	1	1	PMCM1	PMCM0
PCM	FFFF F00C$_H$	00$_H$	PCM7	PCM6	0	0	0	0	PCM1	PCM0

a) These bits must remain in their default state 0.

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

2.3.22 Port group CD

Port group CD is a 4-bit port group. In alternative mode, it comprises pins for the byte enable output of the external data bus.

Port group CD includes the following pins:

Table 2-52 Port group CD: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PCD2 (I/O)	$\overline{\text{BEN0}}$ (O)	PCD2 (I)	5	–	nS
PCD3 (I/O)	$\overline{\text{BEN1}}$ (O)	PCD3 (I)	5	–	nS
PCD4 (I/O)	$\overline{\text{BEN2}}$ (O)	PCD4 (I)	5	–	nS
PCD5 (I/O)	$\overline{\text{BEN3}}$ (O)	PCD5 (I)	5	–	nS

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

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

Table 2-53 Port group CD: configuration registers

Register	Address	Initial value	Used bits							
PMCCD	FFFF F04E$_H$	00$_H$	0	0	PMCCD5	PMCCD4	PMCCD3	PMCCD2	0	0
PMCD	FFFF F02E$_H$	FF$_H$	1	1	PMCD5	PMCD4	PMCD3	PMCD2	1	1
PCD	FFFF F00E$_H$	00$_H$	0	0	PCD5	PCD4	PCD3	PCD2	0	0

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

2.3.23 Port group DL

Port group DL is a 16-bit port group. In alternative mode, it comprises pins for the external memory interface data lines 0 to 15.

Port group DL includes the following pins:

Table 2-54 Port group DL: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PDL0 (I/O)	D0 (I/O)	PDL0 (I)	5	–	nS
PDL1 (I/O)	D1 (I/O)	PDL1 (I)	5	–	nS
PDL2 (I/O)	D2 (I/O)	PDL2 (I)	5	–	nS
PDL3 (I/O)	D3 (I/O)	PDL3 (I)	5	–	nS
PDL4 (I/O)	D4 (I/O)	PDL4 (I)	5	–	nS
PDL5 (I/O)	D5 (I/O)	PDL5 (I)	5	–	nS
PDL6 (I/O)	D6 (I/O)	PDL6 (I)	5	–	nS
PDL7 (I/O)	D7 (I/O)	PDL7 (I)	5	–	nS
PDL8 (I/O)	D8 (I/O)	PDL8 (I)	5	–	nS
PDL9 (I/O)	D9 (I/O)	PDL9 (I)	5	–	nS
PDL10 (I/O)	D10 (I/O)	PDL10 (I)	5	–	nS
PDL11 (I/O)	D11 (I/O)	PDL11 (I)	5	–	nS
PDL12 (I/O)	D12 (I/O)	PDL12 (I)	5	–	nS
PDL13 (I/O)	D13 (I/O)	PDL13 (I)	5	–	nS
PDL14 (I/O)	D14 (I/O)	PDL14 (I)	5	–	nS
PDL15 (I/O)	D15 (I/O)	PDL15 (I)	5	–	nS

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

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

Table 2-55 Port group DL: configuration registers

Register	Address	Initial value	Used bits							
PMCDLL	FFFF F044$_H$	00$_H$	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0
PMCDLH	FFFF F045$_H$	00$_H$	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8
PMCDL (16 bit)	FFFF F044$_H$	0000$_H$	PMCDL15 to PMCDL8 (PMCDLH)				PMCDL7 to PMCDL0 (PMCDLL)			
PMDLL	FFFF F024$_H$	FF$_H$	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
PMDLH	FFFF F025$_H$	FF$_H$	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
PMDL (16 bit)	FFFF F024$_H$	FFFF$_H$	PMDL15 to PMDL8 (PMDLH)				PMDL7 to PMDL0 (PMDLL)			
PDLL	FFFF F004$_H$	00$_H$	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
PDLH	FFFF F005$_H$	00$_H$	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
PDL (16 bit)	FFFF F004$_H$	0000$_H$	PDL15 to PDL8 (PDLH)				PDL7 to PDL0 (PDLL)			

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

All 16-bit registers can be accessed in 16-bit units.

2.3.24 Port group DH

Port group DH is a 16-bit port group. In alternative mode, it comprises pins for the external memory interface data lines 16 to 31.

Port group DH includes the following pins:

Table 2-56 Port group DH: pin functions and port types

Pin functions in different modes		Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
PDH0 (I/O)	D16 (I/O)	PDH0 (I)	5	–	nS
PDH1 (I/O)	D17 (I/O)	PDH1 (I)	5	–	nS
PDH2 (I/O)	D18 (I/O)	PDH2 (I)	5	–	nS
PDH3 (I/O)	D19 (I/O)	PDH3 (I)	5	–	nS
PDH4 (I/O)	D20 (I/O)	PDH4 (I)	5	–	nS
PDH5 (I/O)	D21 (I/O)	PDH5 (I)	5	–	nS
PDH6 (I/O)	D22 (I/O)	PDH6 (I)	5	–	nS
PDH7 (I/O)	D23 (I/O)	PDH7 (I)	5	–	nS
PDH8 (I/O)	D24 (I/O)	PDH8 (I)	5	–	nS
PDH9 (I/O)	D25 (I/O)	PDH9 (I)	5	–	nS
PDH10 (I/O)	D26 (I/O)	PDH10 (I)	5	–	nS
PDH11 (I/O)	D27 (I/O)	PDH11 (I)	5	–	nS
PDH12 (I/O)	D28 (I/O)	PDH12 (I)	5	–	nS
PDH13 (I/O)	D29 (I/O)	PDH13 (I)	5	–	nS
PDH14 (I/O)	D30 (I/O)	PDH14 (I)	5	–	nS
PDH15 (I/O)	D31 (I/O)	PDH15 (I)	5	–	nS

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

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

Table 2-57 Port group DH: configuration registers

Register	Address	Initial value	Used bits							
PMCDHL	FFFF F046$_H$	00$_H$	PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0
PMCDHH	FFFF F047$_H$	00$_H$	PMCDH15	PMCDH14	PMCDH13	PMCDH12	PMCDH11	PMCDH10	PMCDH9	PMCDH8
PMCDH (16 bit)	FFFF F046$_H$	0000$_H$	PMCDH15 to PMCDH8 (PMCDHH)				PMCDH7 to PMCDH0 (PMCDHL)			
PMDHL	FFFF F026$_H$	FF$_H$	PMDH7	PMDH6	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0
PMDHH	FFFF F027$_H$	FF$_H$	PMDH15	PMDH14	PMDH13	PMDH12	PMDH11	PMDH10	PMDH9	PMDH8
PMDH (16 bit)	FFFF F026$_H$	FFFF$_H$	PMDH15 to PMDH8 (PMDHH)				PMDH7 to PMDH0 (PMDHL)			
PDHL	FFFF F006$_H$	00$_H$	PDH7	PDH6	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0
PDHH	FFFF F007$_H$	00$_H$	PDH15	PDH14	PDH13	PDH12	PDH11	PDH10	PDH9	PDH8
PDH (16 bit)	FFFF F006$_H$	0000$_H$	PDH15 to PDH8 (PDHH)				PDH7 to PDH0 (PDHL)			

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

All 16-bit registers can be accessed in 16-bit units.

2.4 Noise Elimination

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

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

Following alternative function signals are not filtered at the ports:

- UARTCn receive data signals RXDCn, but UARTCn has built-in noise rejection filters
- FlexRay receive data signals FRXDA, FRXDB and interrupt input FSTPWD, but FlexRay has built-in noise rejection filters
- CANn receive data signals FCRXDn

In the following all signals are listed, which are filtered.
The port group functions tables in section *"Port group 0" on page 64* and all following also indicate filtered signals.

2.4.1 Analog filtered inputs

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

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

Table 2-58 lists all signals which are passed through an analog filter to remove noise and glitches.

Table 2-58 Input signals with analog filters

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

2.4.2 Digitally filtered inputs

The inputs of the peripherals listed in *Table 2-59* are passed through a digital filter to remove noise and glitches.

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

Table 2-59 Input signals with digital filters (1/2)

Function	Pin name	Inputs with digital filter
Interrupts	P03/ADTRG0/INTP2	ADTRG0/INTP2
	P04/ADTRG1/INTP3	ADTRG1/INTP2
	P30/RXDC0/INTP4	INTP4
	P32/RXDC1/INTP5	INTP5
	P83/SCSE00/INTP6/FCRXD2	INTP6
	P84/SCSE01/INTP7/FCTXD2	INTP7
	P85/SCSE02/INTP8/FCRXD3	INTP8
	P93/SCSE10/INTP9/SCSE04	INTP9
	P94/SCSE11/INTP10/SCSE05/FTINT0	INTP10
	P95/SCSE12/INTP11/SCSE06/FSTPWT	INTP11
	P72/TECRT0/INTP12	TECRT0/INTP12
	P40/SIB0/RXDC2/INTP13	INTP13
Timers A	P10/TIAA00/TEVTAA1/TOAA00	TIAA00/TEVTAA1
	P11/TIAA01/TTRGAA1/TOAA01	TIAA01/TTRGAA1/
	P12/TIAA10/TTRGAA0/TOAA10	TIAA10/TTRGAA0/
	P13/TIAA11/TEVTAA0/TOAA11	TIAA11/TEVTAA0/
	P14/TIAA20/TEVTAA3/TOAA20	TIAA11/TEVTAA0/
	P15/TIAA21/TTRGAA3/TOAA21	TIAA21/TTRGAA3/
	P16/TIAA30/TTRGAA2/TOAA30	TIAA30/TTRGAA2/
	P17/TIAA31/TEVTAA2/TOAA31	TIAA31/TEVTAA2
	P20/TIAA40/TEVTAA5/TOAA40	TIAA40/TEVTAA5
	P21/TIAA41/TTRGAA5/TOAA41	TIAA41/TTRGAA5
	P22/TIAA50/TTRGAA4/TOAA50	TIAA41/TTRGAA5
	P23/TIAA51/TEVTAA4/TOAA51	TIAA51/TEVTAA4
	P24/TIAA60/TEVTAA7/TOAA60	TIAA60/TEVTAA7
	P25/TIAA61/TTRGAA7/TOAA61	TIAA61/TTRGAA7
	P26/TIAA70/TTRGAA6/TOAA70	TIAA70/TTRGAA6
	P27/TIAA71/TEVTAA6/TOAA71	TIAA71/TEVTAA6
	P100/TIAA80/TEVTAA8/TOAA81	TIAA80/TEVTAA8
	P101/TIAA90/TTRGAA9/TOAA90	TIAA90/TTRGAA9
	P102/TIAA91/TEVTAA9/TOAA91	TIAA91/TEVTAA9

Table 2-59 Input signals with digital filters (2/2)

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

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

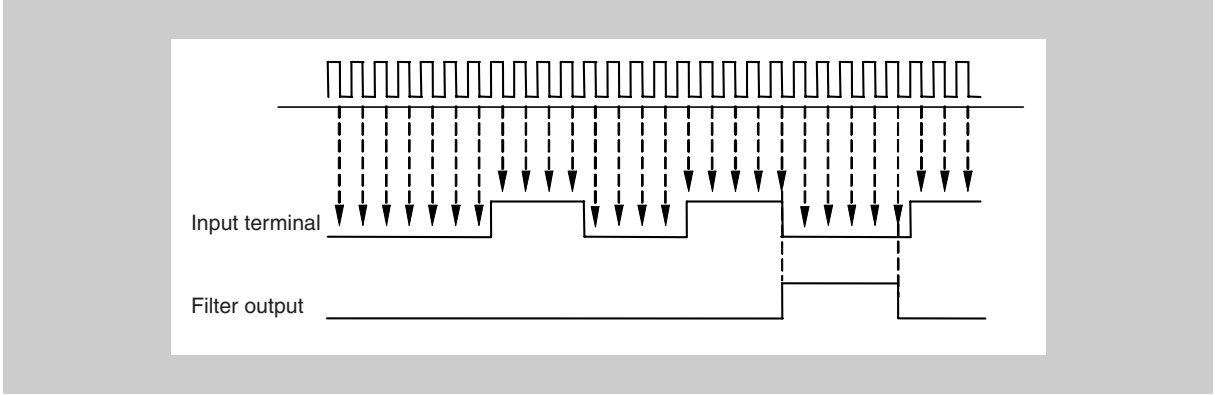


Figure 2-3 Digital noise removal example

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

Table 2-60 Digital noise removal features

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

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

(1) NRC - Noise rejection time control register

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

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

Address FFFF F7A0$_H$

Initial Value 00$_H$. This register is cleared by $\overline{\text{RESET}}$.

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

Table 2-61 NRC register contents

Bit position	Bit name	Function
7 to 0	NRC[7:0]	Sets the sampling clock for the noise elimination circuit for the corresponding input signal: 0: 32 MHz 1: 4 MHz For an assignment of bit positions to input signals see *Table 2-62 on page 90*.

Table 2-62 Assignment of input signals to bit positions for register NRC

Bit position	Bit name	Input signals	Description
7	NRC7	• TAPTS10 to TAPTS12, TEVTS1, TTRGS1 • TAPTS00 to TAPTS02, TEVTS0, TTRGS0	• Timer TMS1 input channels • Timer TMS0 input channels
6	NRC6	• TIAA90,TIAA91, TTRGAA9, TEVTA9 • TIAA80, TEVTAA8	• Timer TAA9 input channels • Timer TAA8 input channels
5	NRC5	• TIAA70,TIAA71, TEVTAA7, TTRGAA7 • TIAA60,TIAA61, TEVTAA6, TTRGAA6	• Timer TAA7 input channels • Timer TAA6 input channels
4	NRC4	• TIAA50,TIAA51, TEVTAA5, TTRGAA5 • TIAA40,TIAA41, TEVTAA4, TTRGAA4	• Timer TAA5 input channels • Timer TAA4 input channels
3	NRC3	• TIAA30,TIAA31, TEVTAA3, TTRGAA3 • TIAA20,TIAA21, TEVTAA2, TTRGAA2	• Timer TAA3 input channels • Timer TAA2 input channels
2	NRC2	• TIAA10,TIAA11, TEVTAA1, TTRGAA1 • TIAA00,TIAA01, TEVTAA0, TTRGAA0	• Timer TAA1 input channels • Timer TAA0 input channels
1	NRC1	• TIT10, TIT11, TEVTT1, TTRGT1, TECRT1, TENCT11, TENCT10 • TIT00, TIT01, TEVTT0, TTRGT0, TECRT0, TENCT01, TENCT00	• Timer TMT1 input channels • Timer TMT0 input channels
0	NRC0	• ADTRG0, ADTRG1 • INTP2 to INTP13	• ADC0 and ADC1 trigger input • External interrupts 2 to 13

2.5 Pin Functions during and after Reset

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

Table 2-63 Pin functions during and after $\overline{\text{RESET}}$ (1/2)

Pin name		$\overline{\text{RESET}}$ = 0	$\overline{\text{RESET}}$ = 1		
			Normal operation	CPU in HALT mode, DMA operating	No access to external memory
External memory interface					
A[21:0] (PAL[15:0], PAH[5:0])		Hi-Z	operating	operating	retains status of last external memory access
D[31:0] (PDL[15:0], PDH[15:0])		Hi-Z	operating	operating	Hi-Z
$\overline{\text{CS0}}$, $\overline{\text{CS[4:2]}}$ (PCS0, PCS[4:2])		Hi-Z	operating	operating	H
$\overline{\text{BEN[3:0]}}$ ($\overline{\text{PCD[5:2]}}$)		Hi-Z	operating	operating	H
$\overline{\text{RD}}$ (PCT4)		Hi-Z	operating	operating	H
$\overline{\text{WR}}$ (PCT5)		Hi-Z	operating	operating	H
$\overline{\text{WAIT}}$ (PCM0)		Hi-Z	operating	operating	–
N-Wire interface					
DCK	$\overline{\text{DRST}}$ = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	$\overline{\text{DRST}}$ = H	operating	operating	operating	operating
DDI	$\overline{\text{DRST}}$ = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	$\overline{\text{DRST}}$ = H	operating	operating	operating	operating
DDO	$\overline{\text{DRST}}$ = L	Hi-Z	L	L	L
	$\overline{\text{DRST}}$ = H	operating	operating	operating	operating
DMS	$\overline{\text{DRST}}$ = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	$\overline{\text{DRST}}$ = H	operating	operating	operating	operating
Non-break debug interface					
CLK_DBG	MODE_DBG = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	MODE_DBG = H	Hi-Z	operating	operating	operating
SYNC	MODE_DBG = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	MODE_DBG = H	Hi-Z	operating	operating	operating
TRIG_DBG	MODE_DBG = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	MODE_DBG = H	Hi-Z	operating	operating	operating
AD0_DBG, AD1_DBG, AD2_DBG, AD3_DBG	MODE_DBG = L	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	MODE_DBG = H	Hi-Z	operating	operating	operating
Flexray interface					
FRXDA/FRXDB		Hi-Z	operating	operating	operating
FTXDA/FTXDB		Hi-Z	operating	operating	operating
FTXENA/FTXENB		Hi-Z	operating	operating	operating
FTINT0 (P94)		Hi-Z	operating	operating	operating
FSTPWT (P95)		Hi-Z	operating	operating	operating

Table 2-63 Pin functions during and after $\overline{\text{RESET}}$ (2/2)

Pin name	$\overline{\text{RESET}}$ = 0	$\overline{\text{RESET}}$ = 1		
		Normal operation	CPU in HALT mode, DMA operating	No access to external memory
Pins in alternative mode, which are not mentioned above				
Alternative inputs	–	operating	operating	operating
Alternative outputs	–	operating	operating	operating
Pins in port mode, which are not mentioned above				
Port inputs	Hi-Z	operating	–	operating
Port outputs	–	operating	retains the last status	operating

2.6 Recommended Connection of unused Pins

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

Table 2-64 Recommended connection of unused pins

Pin	Recommended connection
Port pins	
Pins of port group 0: P0[4:0]	connect to VSS3x
Pins of all other port groups: P1[7:0], P2[7:0], P3[7:0], P4[5:0], P5[7:0], P6[7:0], P7[5:0], P8[6:0], P9[6:0], P10[2:0], P11[5:0], PAL[15:0], PAH[5:0], PCS0, PCS[4:2], PCD[5:2], PDL[15:0], PAH[15:0]	• output ports: leave open • input ports: connect to VDD3x or VSS3x via resistors
Non-port pins	
AD0_DBG to AD3_DBG	connect to VSS3x via resistors
ANI00 to ANI09	connect to AVDD or AVSS0
ANI10 to ANI19	connect to AVDD or AVSS1
CLK_DBG	connect to VSS3x via resistor
DCK, DDI, DMS	connect to VDD3x via resistors
DDO	leave open
$\overline{\text{DRST}}$	connect to VSS3x
FRXDA/FRXDB	connect to VDD3x or VSS3x via resistors
FTXDA/FTXDB/FTXENA/FTXENB	leave open
MODE_DBG	connect to VSS3x
PLLSEL	• 128 MHz: connect to VSS3x via resistor • 80 MHz: connect to VDD3x via resistor
SYNC	connect to VSS3x via resistor
TRIG_DBG	leave open

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

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

Pins with fixed level Beside the power supply pins VDD30/VSS30 and VDD15/VSS15 several other pins have to be set to a fixed level, respectively have to be left open:

- pins to be connected to VSS3x or VSS5x, marked as IC_VSSxx in *Figure 2-4*)
- pins to be connected to VSS3x, marked as IC_VSS3x in *Figure 2-4*)

- pins to be connected to VDD3x, marked as IC_VDD3x in *Figure 2-4*)
- pins to be left open, marked as NC in *Figure 2-4*)

2.7 Package Pins Assignment

The following figure shows the location of pins in top view. Every pin is labelled with its pin name. For port pins, only the pin name in port mode is given. For a list of all alternative pin names, refer to *Table 2-15 on page 49*.

Note When connecting the pins, consider the following:

- IC_VSSxx: Connect to VSS3x or VSS15x
- IC_VSS3x: Connect to VSS3x
- IC_VDD3x: Connect to VDD3x
- NC: Don't connect, leave open
- Index: Connect to VSS3x or VSS15x or leave open
- AVSSx: Avoid injecting noise from VSS30 or VSS15 by providing a low impedance connection of AVSSx to a location with low noise ground potential.
- Thermal: Connect to VSS3x or VSS15x.

	A	B	C	D	E	F	G	H	J	K	L	M	
24	IC_VSSxx	IC_VSSxx	NC	P86	P84	P82	P80	P02	P01	P45	P43	P41	24
23	IC_VSSxx	IC_VSSxx	IC_VSSxx	P85	P83	P81	P04	P03	P00	P44	P42	P40	23
22	FTXDA	FRXDA	IC_VSSxx	VSS30	VSS15	VDD15	MODE1	MODE3	VSS30	VDD30	DDI	DCK	22
21	FTXDB	FRXDB	VSS30	IC_VSSxx	IC_VSSxx	VDD15	MODE2	MODE0	VSS15	VDD30	$\overline{\text{DRST}}$	VSS30	21
20	FTXENA	NC	VSS15	IC_VSSxx									20
19	FTXENB	NC	VDD30	VDD30									19
18	P91	P90	IC_VDD3x	IC_VDD3x									18
17	P92	P93	SYNC	VDD15									17
16	P94	P95	TRIG_DBG	VDD15									16
15	P70	P96	VSS15	VSS30						Thermal	Thermal	Thermal	15
14	P72	P71	AD1_DBG	AD0_DBG						Thermal	Thermal	Thermal	14
13	P100	P75	P73	P74						Thermal	Thermal	Thermal	13
12	P102	P101	VSS30	VDD30						Thermal	Thermal	Thermal	12
11	P111	P110	MODE_DBG	CLK_DBG						Thermal	Thermal	Thermal	11
10	P113	P112	VSS15	VDD15						Thermal	Thermal	Thermal	10
9	P114	P115	VSS30	VDD15									9
8	PCS3	PCS0	VSS15	VDD30									8
7	PCT5	PCT4	AD2_DBG	PCS4									7
6	PCM1	PCM0	PCS2	AD3_DBG									6
5	PCM6	PCM7	VDD30	IC_VSSxx	INDEX								5
4	PCD3	PCD2	VSS30	IC_VSSxx	IC_VSSxx	VDD30	PAL9	PAL13	VDD30	VSS15	PAH4	PDL2	4
3	PCD5	PCD4	IC_VSSxx	VSS15	VSS30	VDD15	PAL7	PAL12	VDD15	VSS30	PAH5	PDL3	3
2	IC_VSSxx	IC_VSSxx	PAL0	PAL2	PAL4	PAL6	PAL10	PAL14	PAH0	PAH2	PDL0	PDL4	2
1	IC_VSSxx	IC_VSSxx	PAL1	PAL3	PAL5	PAL8	PAL11	PAL15	PAH1	PAH3	PDL1	PDL5	1
	A	B	C	D	E	F	G	H	J	K	L	M	

	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	
24	P57	P55	P53	P51	P37	P35	CVDD15	CVSS15	CVDD15	CVSS15	IC_VSSxx	IC_VSSxx	24
23	P56	P54	P52	P50	P36	P34	X1	X2	IC_VSS3xx	IC_VSS3xx	IC_VSS3xx	IC_VSSxx	23
22	$\overline{\text{RESET}}$	DDO	VDD15	VSS15	PLLSEL	IC_VSS3x	VSS15	VSS15	VSS30	IC_VSSxx	IC_VSS3xx	IC_VSS3xx	22
21	DMS	VDD30	VDD30	VSS30	IC_VSS3x	IC_VSS3x	VDD15	IC_VSSxx	IC_VSSxx	VSS30	P32	P33	21
20									IC_VSSxx	VDD30	P30	P31	20
19									VDD15	VDD15	P66	P67	19
18									P65	P64	P63	P62	18
17									VSS15	VSS15	P60	P61	17
16									VSS30	VSS30	P26	P27	16
15	Thermal	Thermal	Thermal						P22	P23	P24	P25	15
14	Thermal	Thermal	Thermal						P17	P14	P20	P21	14
13	Thermal	Thermal	Thermal						VDD30	VDD30	P15	P16	13
12	Thermal	Thermal	Thermal						VDD15	VDD30	P13	P12	12
11	Thermal	Thermal	Thermal						VSS15	VSS30	P10	P11	11
10	Thermal	Thermal	Thermal						ANI10	ANI11	AVSS1	AVSS1	10
9									ANI12	ANI13	ANI14	ANI15	9
8									ANI16	ANI17	ANI18	ANI19	8
7									ANI09	ANI08	AVREF1	AVREF0	7
6									ANI07	ANI06	AVDD	AVDD	6
5									AVSS0	AVSS0	ANI05	ANI04	5
4	VSS15	VDD30	VDD30	VSS30	PDH0	VDD30	PDH11	IC_VSSxx	IC_VSSxx	IC_VSSxx	ANI03	ANI02	4
3	VSS30	VDD15	PDL6	PDL9	PDH1	VDD15	PDH9	VSS15	VSS30	IC_VSSxx	ANI01	ANI00	3
2	PDL7	PDL10	PDL12	PDL15	PDH3	PDH5	PDH7	PDH10	PDH13	PDH15	IC_VSSxx	IC_VSSxx	2
1	PDL8	PDL11	PDL13	PDL14	PDH2	PDH4	PDH6	PDH8	PDH12	PDH14	IC_VSSxx	IC_VSSxx	1
	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	

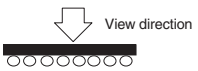


Figure 2-4 Pin overview

Chapter 3 CPU System Functions

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

3.1 Overview

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

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

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

Features summary

The CPU has the following special features:

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

3.1.1 Description

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

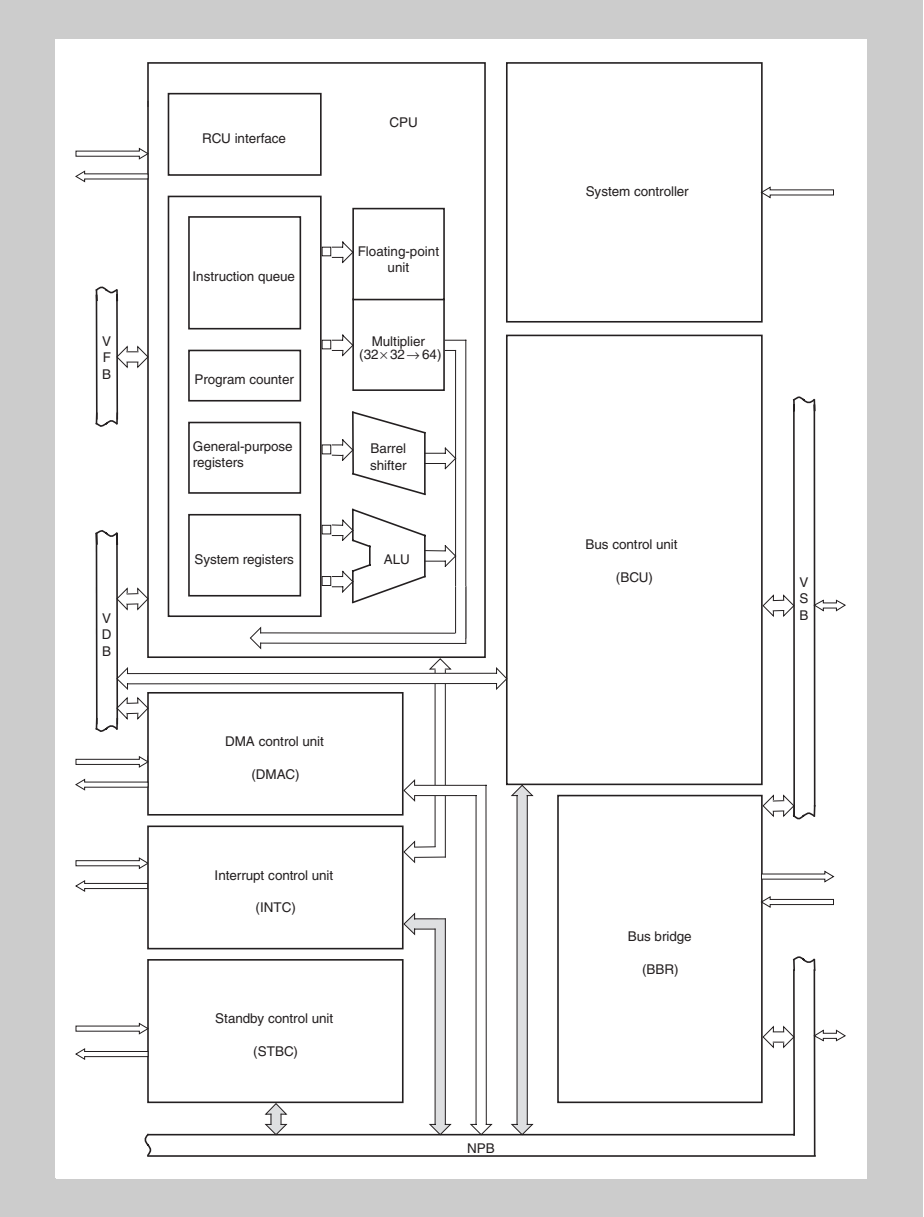


Figure 3-1 CPU system

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

Table 3-1 Bus types

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

3.2 CPU Register Set

There are three categories of registers:

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

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

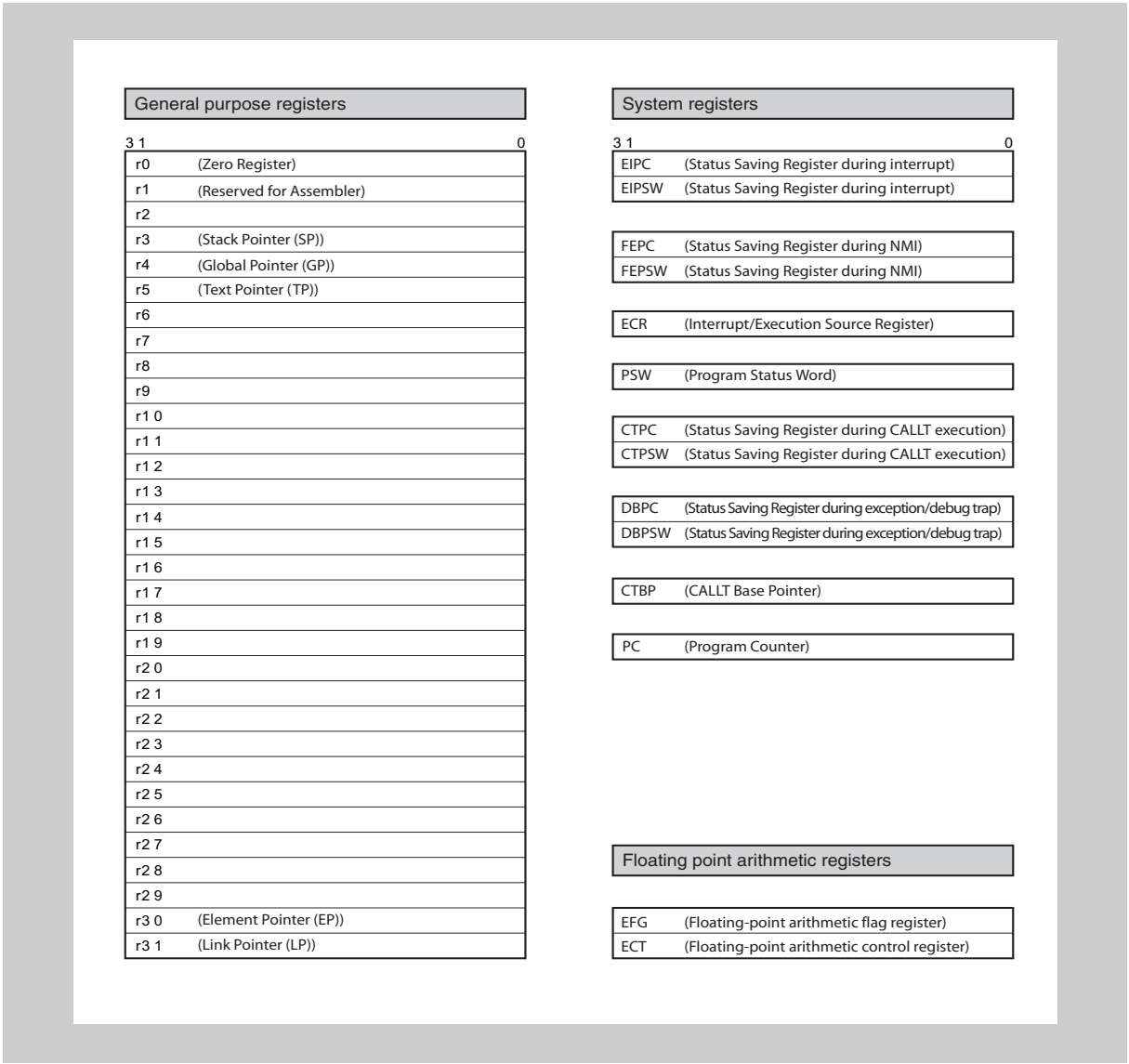


Figure 3-2 CPU register set

Some registers are write protected. That means, writing to those registers is protected by a special sequence of instructions. Refer to *"Write Protected Registers" on page 123* for more details.

3.2.1 General purpose registers (r0 to r31)

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

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

Table 3-2 General purpose registers

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

a) Registers r0 and r30 are used by dedicated instructions.
b) Registers r1, r3, r4, r5, and r31 may be used by the assembler/compiler.

Caution Before using registers r1, r3 to r5, r30, and r31, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used.

3.2.2 System register set

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

To read/write the system registers, use instructions LDSR (load to system register) or STSR (store contents of system register), respectively, with a specific system register number (RegID) indicated below.
The program counter states an exception. It cannot be accessed via LDSR or STSR instructions. No regID is allocated to the program counter.

Example STSR 0, r2

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

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

Table 3-3 System register numbers

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

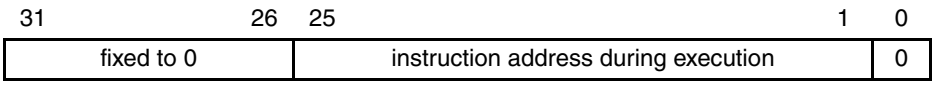
a) Reading from this register is only enabled between a DBTRAP exception (exception handler address $0000\ 0060_H$) and the exception handler terminating DBRET instruction. DBTRAP exceptions are generated upon ILGOP detections (refer to *"Interrupt Controller (INTC)" on page 125*).

(1) PC - Program counter

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

Access This register can not be accessed by any instruction.

Initial Value $0000\ 0000_H$. The program counter is cleared by $\overline{RESET}$.



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

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

When one of the occasions listed in *Table 3-4* occurs, except for some instructions, the address of the instruction following the one being executed is saved to the saving registers.
For more details refer to *Table 3-9 on page 106* and to the *"Interrupt Controller (INTC)" on page 125*.

All PC saving registers are built up as the PC, with the initial value $0xxx\ xxxx_H$ (x = undefined).

Table 3-4 PC saving registers

Register	Shortcut	Saves contents of PC in case of
Status saving register during interrupt	EIPC	• software exception • maskable interrupt
Status saving register during non-maskable interrupts	FEPC	• non-maskable interrupt
Status saving register during exception/debug trap	DBPC[a]	• exception trap • debug trap • debug break • during a single-step operation
Status saving register during CALLT execution	CTPC	• execution of CALLT instruction

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

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

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

(3) PSW - Program status word

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

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

Initial Value 0000 0020$_H$. The program status is initialized by $\overline{\text{RESET}}$.

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

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

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

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

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

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

Saturated operation instructions

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

Table 3-6 Saturation-processed operation result

Status of operation result	Flag status			Saturation-processed operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFF FFFF$_H$
Maximum negative value exceeded	1	1	1	8000 0000$_H$
Positive (maximum not exceeded)	x[a]	0	0	Operation result itself
Negative (maximum not exceeded)			1	

a) Retains the value before operation.

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

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

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

All PSW saving registers are built up as the PSW, with the initial value 0000 $0xxx_H$ (x = undefined).

Table 3-7 PSW saving registers

Register	Shortcut	Saves contents of PSW in case of
Status saving register during interrupt	EIPSW	• software exception • maskable interrupt
Status saving register during non-maskable interrupts	FEPSW	• non-maskable interrupt
Status saving register during exception/debug trap	DBPSW[a]	• exception trap • debug trap • debug break • during a single-step operation
Status saving register during CALLT execution	CTPSW	• execution of CALLT instruction

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

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

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

(5) ECR - Interrupt/exception source register

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

For a list of interrupts/exceptions and corresponding exception codes, see *Table 3-9 on page 106*.

Initial Value $0000\ 0000_H$. This register is cleared by $\overline{RESET}$.

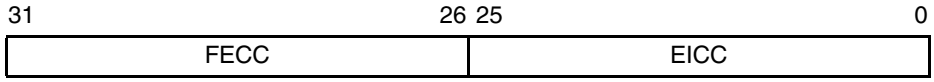


Table 3-8 ECR register contents

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

The following table lists the exception codes.

Table 3-9 Interrupt/execution codes

Interrupt/Exception Source			Classification	Exception Code	Handler Address	Value restored to EIPC/FEPC
Name		Trigger				
Non-maskable interrupt (NMI)		refer to *"Interrupt Controller (INTC)" on page 125*	Interrupt	0010_H	$0000\ 0010_H$	next PC (see Note)
Maskable interrupt			Interrupt	refer to *"Interrupt Controller (INTC)" on page 125*	• higher 16 bits: 0000_H • lower 16 bits: exception code	next PC (see Note)
Software exception	TRAP0n (n = 0 to F_H)	TRAP instruction	Exception	$004n_H$	$0000\ 0040_H$	next PC
	TRAP1n (n = 0 to F_H)	TRAP instruction	Exception	$005n_H$	$0000\ 0050_H$	next PC
Exception trap (ILGOP)		Illegal instruction code	Exception	0060_H	$0000\ 0060_H$	next PC
Debug trap		DBTRAP instruction	Exception	0060_H	$0000\ 0060_H$	next PC

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

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

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

(6) CTBP - CALLT base pointer

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

Initial Value Undefined

31	30	29	28	27	26	25 ... 1	0
0	0	0	0	0	0	base address	0
R^a	R^a	R^a	R^a	R^a	R^a	R/W	R

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

3.2.3 Floating-point arithmetic registers

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

Table 3-10 FPU control registers

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

(1) EFG - Floating-point arithmetic flag register

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

Initial Value $0000\ 0000_H$.

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

Table 3-11 EFG register contents

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

(2) ECT - Floating-point arithmetic control register

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

Initial Value $0000\ 0000_H$.

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

Table 3-12 ECT register contents

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

3.3 Operation Modes

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

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

- Normal operation mode
- Flash programming mode

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

Table 3-13 Selection of operation modes

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

Note

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

3.3.1 Normal operation mode

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

After reset release program execution is started at address 0000 0000_H.

The lower 1 MB of the memory area is mapped to internal resources, for instance to the internal flash memory. Thus, external memory mapped to this area can not be addressed in normal operation mode. See also *"Bus and Memory Control (BCU, MEMC)" on page 225*.

3.3.2 Flash programming mode

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

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

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

For more information see section *"Flash Memory" on page 191*.

3.3.3 Boundary scan mode

This mode enables boundary tests according to the IEEE1149.1 specification via a JTAG interface.

For more information see section *“Boundary Scan” on page 1349*.

3.4 Address Space

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

3.4.1 CPU address space and physical address space

The CPU supports the following address space:

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

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

The 256 MB physical address space is seen as 16 images in the 4 GB CPU address space:

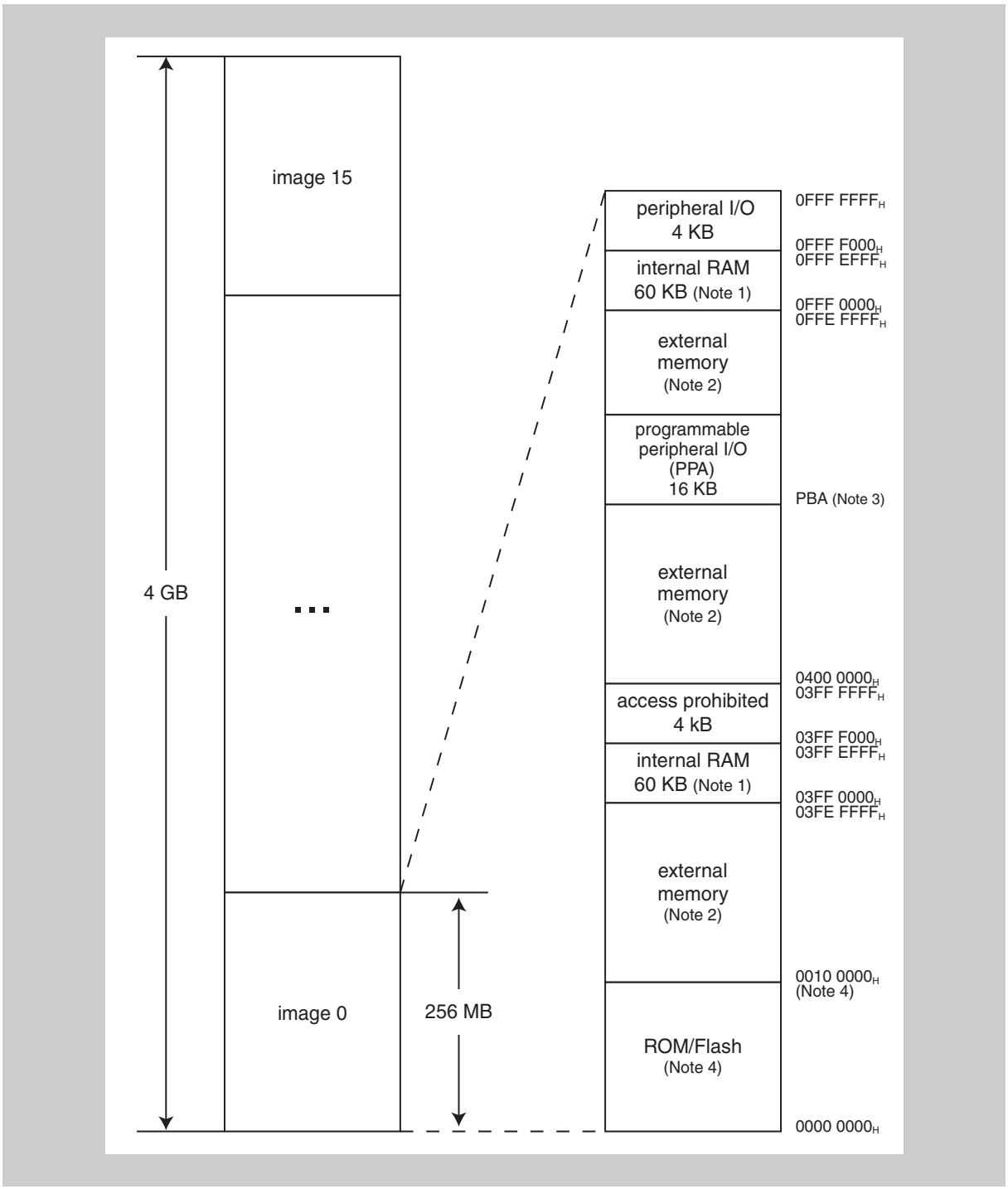


Figure 3-3 Images in the CPU address space

Note

1. The internal RAM area (03FF 0000$_H$ and 03FF EFFF$_H$) is mirrored to the area 0FFF 0000$_H$ to 0FFF EFFF$_H$. If data is written in one area, it appears also in the other area.
2. Note that other internal resources may be mapped to the external memory areas. Refer to *"Bus and Memory Control (BCU, MEMC)" on page 225*.
3. The programmable peripheral base address is defined by the BPC register.
4. μPD70F3483: 768 KB up to 000B FFFF$_H$;
   μPD70F3441: 1 MB up to 000F 7FFF$_H$

3.4.2 Program and data space

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

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

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

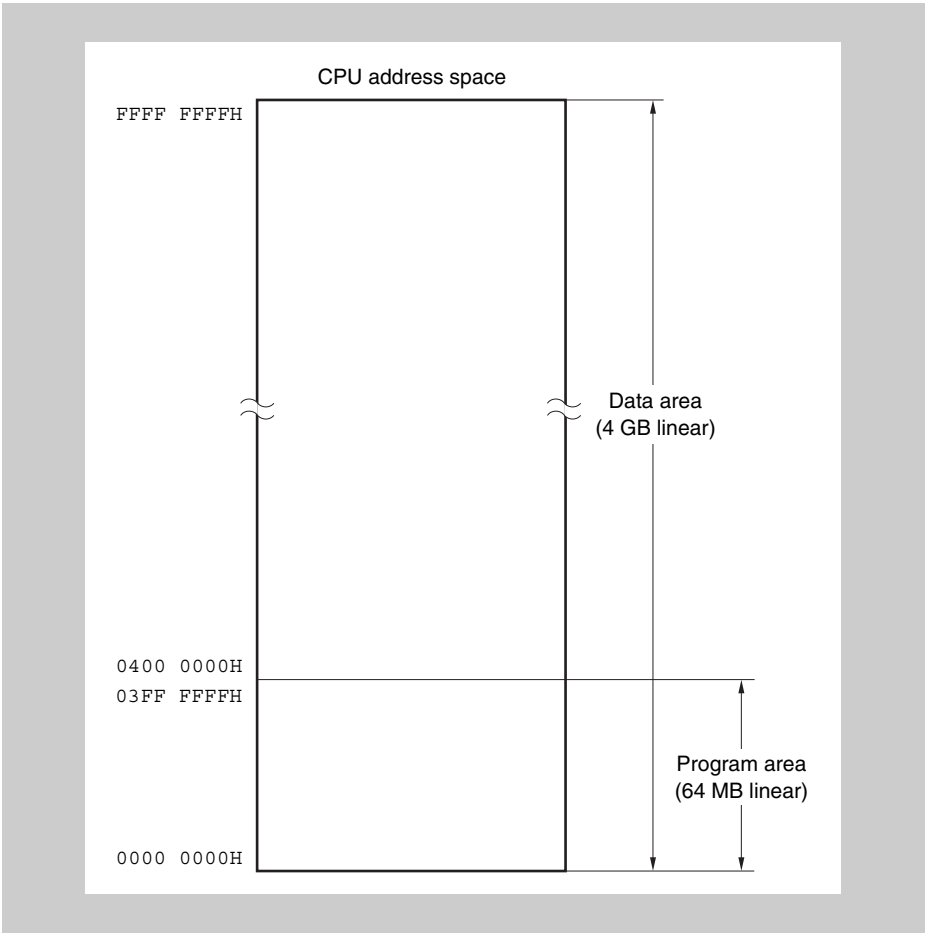


Figure 3-4 CPU address space

(1) Wrap-around of data space

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

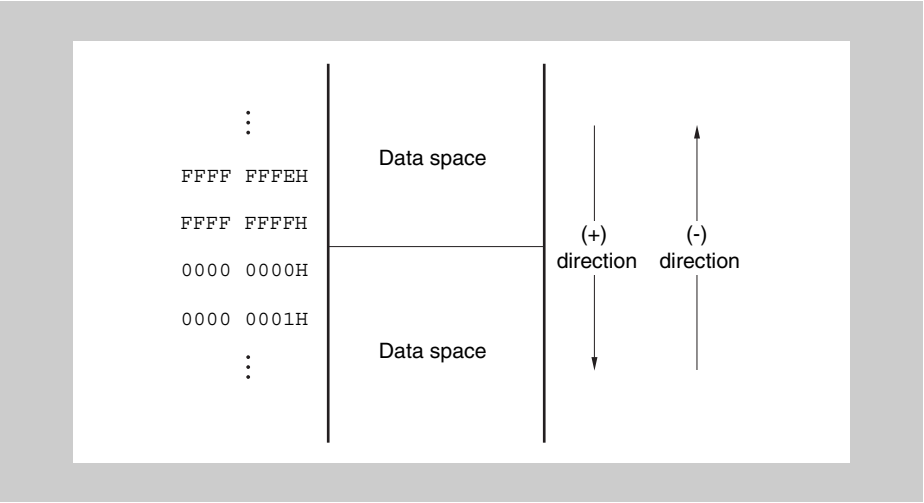


Figure 3-5 Wrap-around of data space

(2) Wrap-around of program space

If an instruction address calculation exceeds 26 bits, only the lower 26 bits of the result are considered. Therefore, the addresses 0000 0000_H and 03FF $FFFF_H$ are contiguous addresses. This results in a wrap-around of the program space:

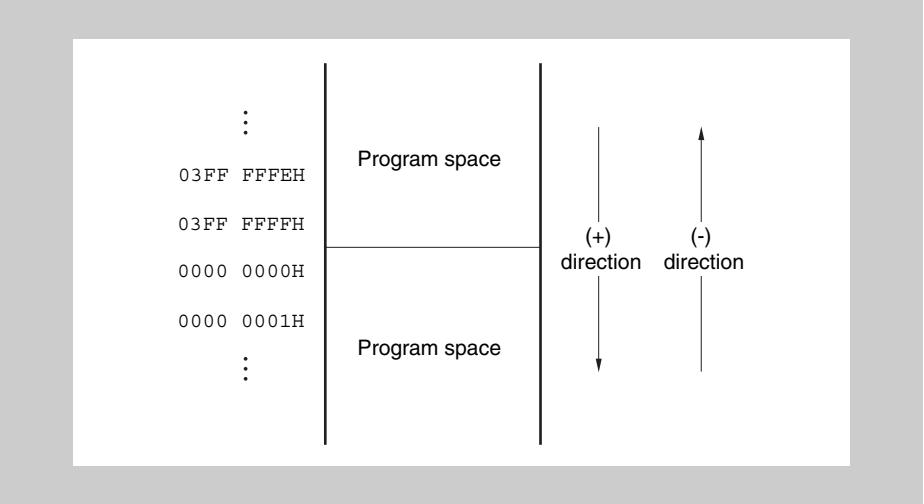


Figure 3-6 Wrap-around of program space

Caution No instruction can be fetched from the 4 KB area of 0FFF $F000_H$ to 0FFF $FFFF_H$ because this area is defined as peripheral I/O area. Therefore, do not execute any branch to this area.

3.5 Memory

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

3.5.1 Memory areas

The internal memory of the CPU provides several areas:

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

The areas are briefly described below.

(1) Internal flash area

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

Table 3-14 Internal flash areas

Device	Internal flash size	Address range
µPD70F3483	768 KB	0000 0000_H to 000B $FFFF_H$
µPD70F3441	992 KB	0000 0000_H to 000F $7FFF_H$

(2) Internal RAM area

The 60 KB area between addresses 03FF 0000_H and 03FF $EFFF_H$ is provided as the internal RAM area.

Note The internal RAM area is mirrored to the area 0FFF 0000_H to 0FFF $EFFF_H$. If data is written in one area, it appears also in the other area.

(3) Fixed peripheral I/O area

The 4 KB area between addresses 0FFF $F000_H$ and 0FFF $FFFF_H$ is provided as the fixed peripheral I/O area. Accesses to these addresses are passed over to the NPB bus (internal bus).

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

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

For a list of all peripheral I/O registers, see *“Special Function Registers” on page 1351*.

Note

1. Because the physical address space covers 256 MB, the address bits A[31:28] are not considered. Thus, this address space can also be addressed via the area FFFF $F000_H$ to FFFF $FFFF_H$. This has the advantage that the area can be indirectly addressed by an offset and the zero base r0.
   Therefore, in this manual, all addresses of peripheral I/O registers in the 4 KB peripheral I/O area are given in the range FFFF $F000_H$ to FFFF $FFFF_H$ instead of 0FFF $F000_H$ to 0FFF $FFFF_H$.
2. The *fixed* peripheral I/O area is mirrored to the upper 4 KB of the *programmable* peripheral I/O area PPA—regardless of the base address of the PPA. If data is written to one area, it appears also in the other area.
3. Program fetches cannot be executed from any peripheral I/O area.
4. Word registers, that means 32-bit registers, are accessed in two half word accesses. The lower two address bits are ignored.
5. For registers in which byte access is possible, if half word access is executed:
   - During read operation: The higher 8 bits become undefined.
   - During write operation: The lower 8 bits of data are written to the register.

Caution

1. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.
2. For DMA transfer, the fixed peripheral I/O area (0FFF $F000_H$ to 0FFF $FFFF_H$) cannot be specified as the source/destination address. Be sure to use the RAM area (0FFF 0000_H to 0FFF $EFFF_H$ or 03FF 0000_H to 03FF $EFFF_H$) for source/destination address of DMA transfer.

(4) Programmable peripheral I/O area

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

See *"Bus and Memory Control (BCU, MEMC)" on page 225* for details.

(5) External memory area

All address areas that do not address any internal memory or peripheral I/O registers can be used as external memory area. Note that other internal resources may be mapped to the external memory areas.

Access to the external memory area uses the chip select (CS) signals assigned to each memory area.

For access to external memory, see *"Bus and Memory Control (BCU, MEMC)" on page 225*.

3.6 RAM ECC error detection

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

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

INTDEDR can generate two kinds of interrupts:

- an NMI via the NMI sharing function
- a maskable interrupt (exception code 0080_H), that is shared with double error detections of the flash memory and FlexRay error detection functions

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

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

3.6.1 RAM ECC initialization

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

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

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

3.6.2 RAM ECC registers

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

Table 3-15 RAM ECC registers overview

Register name	Shortcut	Address
RAM ECC error address register	RAMEAD	FFFF F8B0$_H$
RAM ECC error data location register	RAMEDLR	FFFF F8B2$_H$
RAM ECC control register	RAMECC	FFFF F8B4$_H$

(1) RAMEAD - RAM ECC error address register

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

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

Address FFFF F8B0$_H$

Initial Value 0000$_H$

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

Table 3-16 RAMEAD register contents

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

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

This register is not overwritten by a new error detection until

- the INTERRF.INTERR1 is cleared to 0. Thus no new error is signaled by interrupt INTDEDR until the INTERRF.INTERR1 is cleared to 0.
- RAMEAD is cleared to 0000$_H$ by any reset.

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

(2) RAMEDLR - RAM ECC error data location register

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

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

Address FFFF F8B2$_H$

Initial Value 00$_H$

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

Table 3-17 RAMEDLR register contents

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

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

(3) RAMECC - RAM ECC control register

RAMECC control the operation of the RAM ECC function.

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

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

Address FFFF F8B4$_H$

Initial Value 00$_H$

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

Table 3-18 RAMECC register contents

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

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

3.6.3 RAM ECC function check

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

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

3.6.4 Recommended use of data address space

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

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

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

For efficient use of the relative addressing feature, the data segments should be located in the address range FFFF $F800_H$ to 0000 0000_H and 0000 0000_H to 0000 $7FFF_H$. The peripheral I/O registers and the internal RAM is aligned to the upper bound, thus the registers and a part of the RAM can be addressed via relative addressing, with base address 0 (r0).

It is recommended to locate flash memory data segments in the area up to 0000 $7FFF_H$, so access to these constant data can utilize also relative addressing.

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

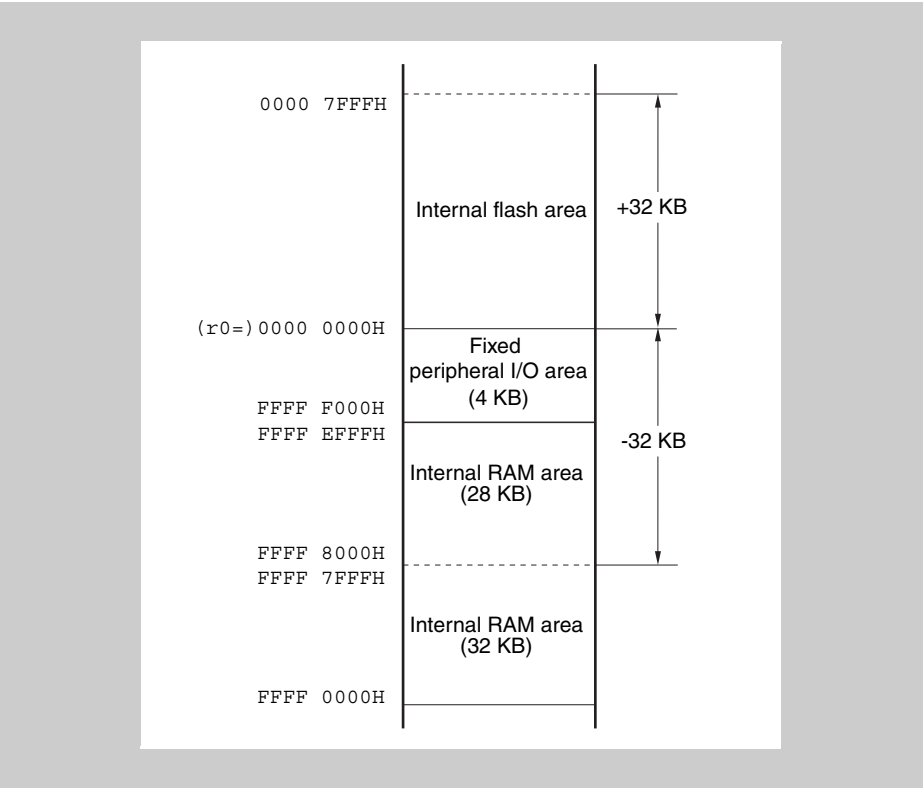


Figure 3-7 Example application of wrap-around

3.7 Write Protected Registers

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

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

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

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

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

Table 3-19 Overview of write protected registers

Write protected register	Shortcut	Corresponding write enable register	Shortcut	For details see
Port group control registers of port group 5	P5, PM5, PMC5, PESC5, ESOST5, PESMK5	Command register	PRCMD	*"Pin Functions" on page 31*
Port group control registers of port group 6	P6, PM6, PMC6, PESC6, ESOST6, PESMK6			
RAM ECC control register	RAMECC	Command register	PRCMD	*"RAM ECC error detection" on page 117*
Clock Monitor mode register	CLM	Command register	PRCMD	*"Clock Generator" on page 179*
Data flash control register	DFLCTL	Command register	PRCMD	*"Flash Memory" on page 191*

Note The FlexRay module also contains write protected registers. See *"FlexRayTM" on page 1147* for details.

Example Write to port register P5:

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

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

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

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

However, any non-maskable interrupt can still be acknowledged.

- DMA:
  In the above example, DMA transfers can still take place. They may destroy the sequence.

  If appropriate, you may disable DMA transfers in advance. Otherwise you must check whether writing to the protected register was successful. To do so, check the status via the status register, if available, or by reading back the protected register.

Chapter 4 Interrupt Controller (INTC)

This controller is provided with a dedicated Interrupt Controller (INTC) for interrupt servicing and can process a large amount of maskable and up to five non-maskable interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution. Generally, an exception takes precedence over an interrupt.

This controller can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

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

4.1 Features

- Interrupts
  - Non-maskable interrupts: up to 5 sources
    (depending on whether interrupt sharing is enabled or not)
  - Maskable interrupts:
    - internal peripherals: 101 sources
    - external: 14 sources
  - 8 levels of programmable priorities (maskable interrupts)
  - Multiple interrupt control according to priority
  - Masks can be specified for each maskable interrupt request
  - Noise elimination, edge detection and valid edge specification, level detection for external interrupt request signals
  - Wake-up capable
    (analogue noise elimination for external interrupt request signals)
- Exceptions
  - Software exceptions: 2 channels with each 16 sources
  - Exception traps: 1 source (illegal opcode exception)

Table 4-1 Interrupt/exception source list (1/6)

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

Table 4-1 Interrupt/exception source list (2/6)

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

Table 4-1 Interrupt/exception source list (3/6)

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

Table 4-1 Interrupt/exception source list (4/6)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTTAA7OV	TAA7OVIC	TAA7 overflow interrupt	TAA7	66	04A0H	000004A0H	nextPC
	INTTAA7CC0	TAA7CCIC0	TAA7 capture compare channel 0	TAA7	67	04B0H	000004B0H	nextPC
	INTTAA7CC1	TAA7CCIC1	TAA7 capture compare channel 1	TAA7	68	04C0H	000004C0H	nextPC
	INTTAA8OV	TAA8OVIC	TAA8 overflow interrupt	TAA8	69	04D0H	000004D0H	nextPC
	INTTAA8CC0	TAA8CCIC0	TAA8 capture compare channel 0	TAA8	70	04E0H	000004E0H	nextPC
	INTTAA8CC1	TAA8CCIC1	TAA8 capture compare channel 1	TAA8	71	04F0H	000004F0H	nextPC
	INTTAA9OV	TAA9OVIC	TAA9 overflow interrupt	TAA9	72	0500H	00000500H	nextPC
	INTTAA9CC0	TAA9CCIC0	TAA9 capture compare channel 0	TAA9	73	0510H	00000510H	nextPC
	INTTAA9CC1	TAA9CCIC1	TAA9 capture compare channel 1	TAA9	74	0520H	00000520H	nextPC
	INTBRG0[b]	BRGIC	BRG0 match	BRG0	75	0530H	00000530H	nextPC
	INTBRG1[b]		BRG1 match	BRG1				
	INTBRG2	BRGIC2	BRG2 match	BRG2	76	0540H	00000540H	nextPC
	INTFR0	FRIC0	FlexRay error/status interrupt line 0	FlexRay	77	0550H	00000550H	nextPC
	INTFR1	FRIC1	FlexRay error/status interrupt line 1	FlexRay	78	0560H	00000560H	nextPC
	INTFR2	FRIC2	FlexRay timer 0 interrupt	FlexRay	79	0570H	00000570H	nextPC
	INTFR3	FRIC3	FlexRay timer 1 interrupt	FlexRay	80	0580H	00000580H	nextPC
	INTFRIB	FRIBIC	FlexRay transfer IBF to MBF busy	FlexRay	81	0590H	00000590H	nextPC
	INTFROB	FROBIC	FlexRay transfer MBF to OBF busy	FlexRay	82	05A0H	000005A0H	nextPC
	INTC0ERR	C0ERRIC	CAN0 error interrupt	CAN0	83	05B0H	000005B0H	nextPC
	INTC0WUP	C0WUPIC	CAN0 wake up interrupt	CAN0	84	05C0H	000005C0H	nextPC
	INTC0REC	C0RECIC	CAN0 receive interrupt	CAN0	85	05D0H	000005D0H	nextPC
	INTC0TRX	C0TRXIC	CAN0 transmit interrupt	CAN0	86	05E0H	000005E0H	nextPC
	INTC1ERR	C1ERRIC	CAN1 error interrupt	CAN1	87	05F0H	000005F0H	nextPC
	INTC1WUP	C1WUPIC	CAN1 wake up interrupt	CAN1	88	0600H	00000600H	nextPC
	INTC1REC	C1RECIC	CAN1 receive interrupt	CAN1	89	0610H	00000610H	nextPC
	INTC1TRX	C1TRXIC	CAN1 transmit interrupt	CAN1	90	0620H	00000620H	nextPC

Table 4-1 Interrupt/exception source list (5/6)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTTS0CC1B	C2ERRIC	TMS0 capture compare channel 1	TMS0	91	0630H	00000630H	nextPC
	INTTS0CC2B	C2WUPIC	TMS0 capture compare channel 2	TMS0	92	0640H	00000640H	nextPC
	INTTS0CC3B	C2RECIC	TMS0 capture compare channel 3	TMS0	93	0650H	00000650H	nextPC
	Reserved	–	Reserved	–	94	0660H	00000660H	nextPC
	INTTS1CC1B	C3ERRIC	TMS1capture compare channel 1	TMS1	95	0670H	00000670H	nextPC
	INTTS0CC2B	C3WUPIC	TMS1 capture compare channel 2	TMS1	96	0680H	00000680H	nextPC
	INTTS0CC3B	C3RECIC	TMS1 capture compare channel 3	TMS1	97	0690H	00000690H	nextPC
	Reserved	–	Reserved	–	98	06A0H	000006A0H	nextPC
	INTCB0T[b]	CB0TIC	CSIB0 transmit interrupt	CSIB0 (DMAC)[c]	99	06B0H	000006B0H	nextPC
	INTUC2T[b]		UARTC2 transmit interrupt	UARTC2 (DMAC)[c]				
	INTCB0R[b]	CB0RIC	CSIB0 receive complete interrupt	CSIB0 (DMAC)[c]	100	06C0H	000006C0H	nextPC
	INTUC2R[b]		UARTC2 receive complete interrupt	UARTC2 (DMAC)[c]				
	INTP13[b]		External interrupt 13	Pin				
	INTCB0RE[b]	CB0REIC	CSIB0 receive error interrupt	CSIB0	101	06D0H	000006D0H	nextPC
	INTUC2RE[b]		UART2 receive error interrupt	UART2				
	INTCB1T	CB1TIC	CSIB1 transmit interrupt	CSIB1 (DMAC)[c]	102	06E0H	000006E0H	nextPC
	INTCB1R	CB1RIC	CSIB1 receive complete interrupt	CSIB1 (DMAC)[c]	103	06F0H	000006F0H	nextPC
	INTCB1RE	CB1REIC	CSIB1 receive error interrupt	CSIB1	104	0700H	00000700H	nextPC
	INTCE0OF	CE0OFIC	CSIE0 overflow interrupt	CSIE0	105	0710H	00000710H	nextPC
	INTCE0C	CE0CIC	CSIE0 transfer end interrupt	CSIE0 (DMAC)[c]	106	0720H	00000720H	nextPC
	INTCE1OF	CE1OFIC	CSIE1 overflow interrupt	CSIE1	107	0730H	00000730H	nextPC
	INTCE1C	CE1CIC	CSIE1 transfer end interrupt	CSIE1 (DMAC)[c]	108	0740H	00000740H	nextPC
	INTUC0RE	UC0REIC	UARTC0 receive error interrupt	UARTC0	109	0750H	00000750H	nextPC
	INTUC0T	UC0TIC	UARTC0 transmit interrupt	UARTC0 (DMAC)[c]	110	0760H	00000760H	nextPC
	INTUC1RE	UC1REIC	UARTC1 receive error interrupt	UARTC1	111	0770H	00000770H	nextPC
	INTUC1T	UC1TIC	UARTC1 transmit interrupt	UARTC1 (DMAC)[c]	112	0780H	00000780H	nextPC

Table 4-1 Interrupt/exception source list (6/6)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTAD0	ADIC0	ADC0 end of conversion	ADC0 (DMAC)[c]	113	0790H	00000790H	nextPC
	INTAD1	ADIC1	ADC1 end of conversion	ADC1 (DMAC)[c]	114	07A0H	000007A0H	nextPC
	INTDMA2	DMAIC2	DMA2 transmission end	DMAC	115	07B0H	000007B0H	nextPC
	INTDMA3[b]	DMAIC3	DMA3 transmission end	DMAC	116	07C0H	000007C0H	nextPC
	INTFL[b]		Sequencer interrupt	Flash memory				

a) Depending on the setting of INTSEL.ISR, this interrupt source generates an NMI or a maskable interrupt. For more information see *"Interrupt Sharing" on page 153*. For details see *"Interrupt Sharing" on page 153*.

b) This is a shared interrupt. For details see *"Interrupt Sharing" on page 153*.

c) If this interrupt is defined as a DMA trigger, it is generated by the corresponding DMA completion interrupt INTDMAn instead of the original interrupt source. For further details refer to *"DMA Functions (DMA Controller)" on page 279*

Note

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

4.2 Non-Maskable Interrupts

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

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

Table 4-2 Non-maskable interrupts

Interrupt Source	Generating Unit	Comment
NMI	Pin	When the valid edge specified by bits INTM0.ESN[1:0] is detected on the NMI pin, the interrupt occurs.
INTDEDF	Flash memory	These interrupt sources only generate a *non-maskable* interrupt if interrupt sharing is enabled for non-maskable interrupts. See *"Interrupt Sharing" on page 153*.
INTDEDR	RAM	
INTDEDFR	FlexRay	
INTOSD	Clock Monitor	

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

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

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

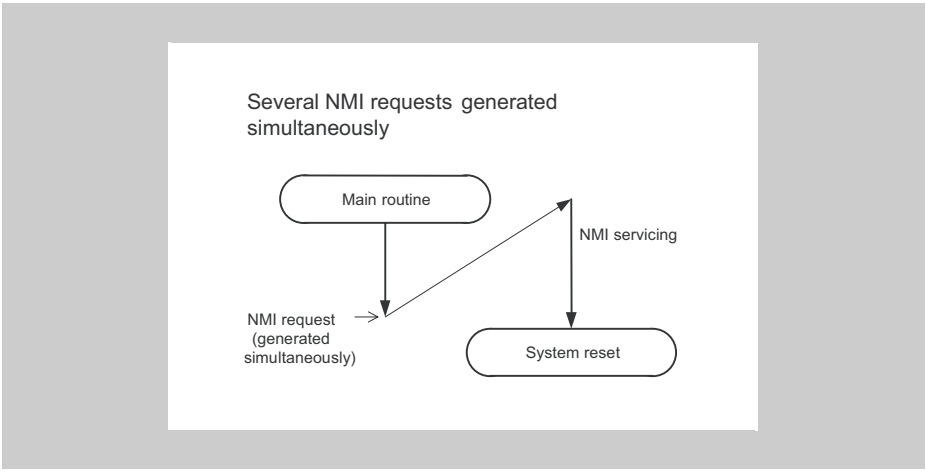


Figure 4-1 Example of non-maskable interrupt request acknowledgement operation: multiple NMI requests generated at the same time

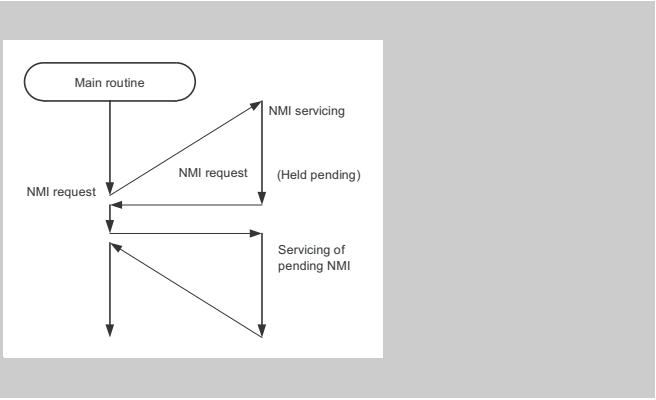


Figure 4-2 Example of non-maskable interrupt request acknowledgement operation: NMI request generated during NMI servicing

4.2.1 Operation

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

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

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

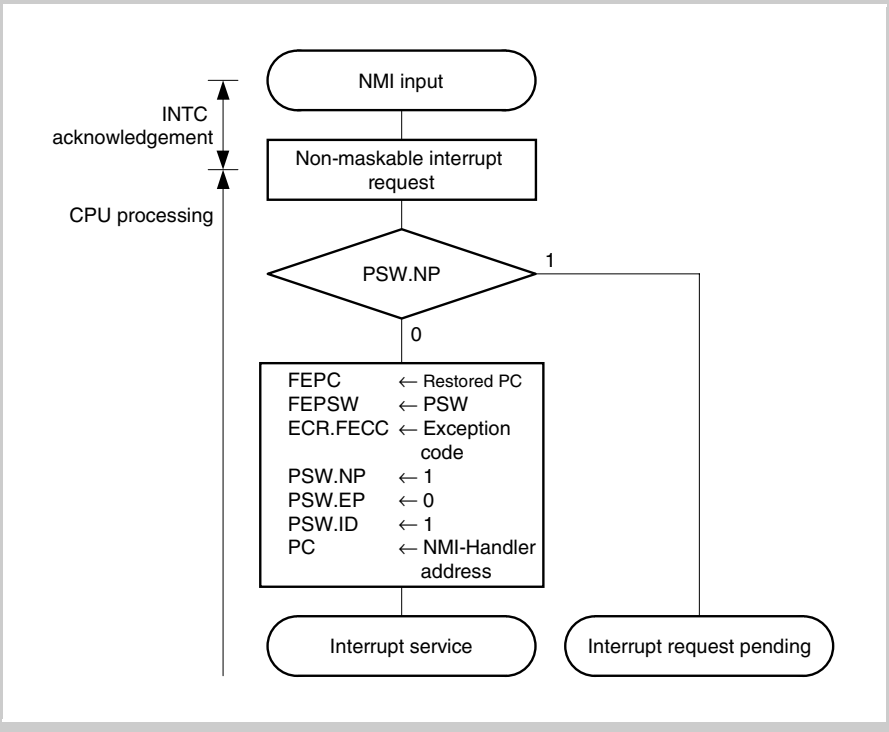


Figure 4-3 Processing configuration of non-maskable interrupt

4.2.2 Restore

Execution is restored from the non-maskable interrupt (NMI) processing by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

1. Restores the values of the PC and the PSW from FEPC and FEPSW, respectively, because the EP bit of the PSW is 0 and the NP bit of the PSW is 1.
2. Transfers control back to the address of the restored PC and PSW.

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

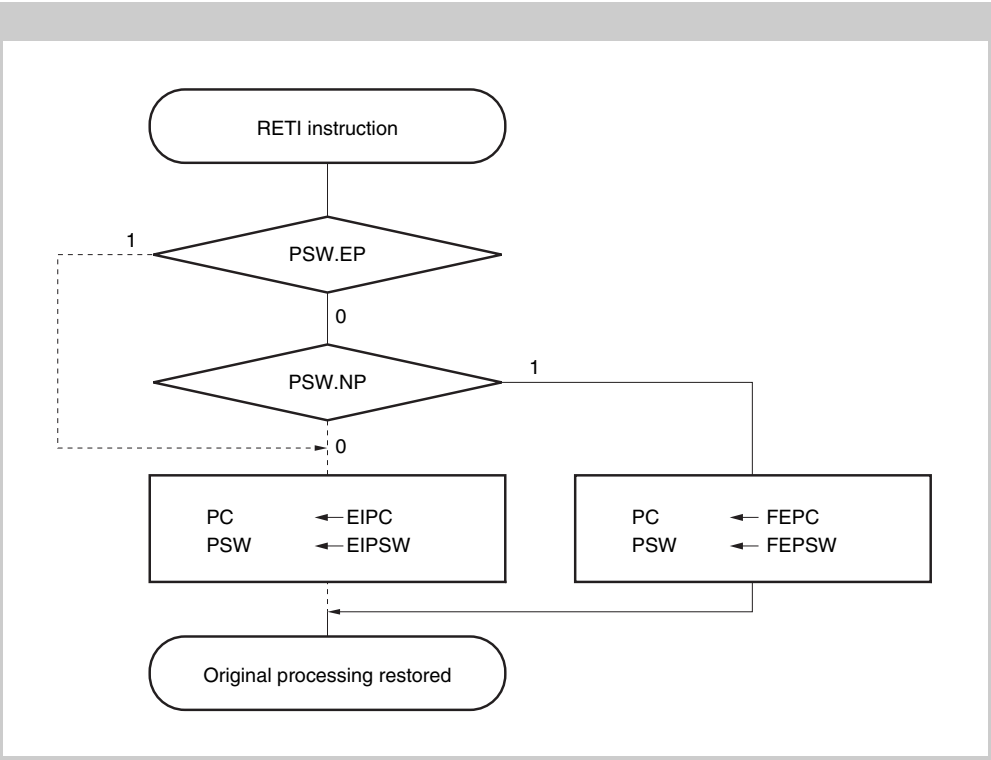


Figure 4-4 RETI instruction processing

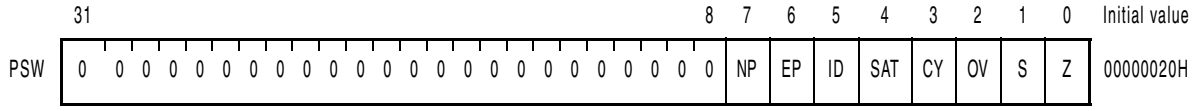
Caution When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately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

Note The solid line indicates the CPU processing flow.

4.2.3 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 an NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.



Bit position	Bit name	Function
7	NP	Indicates whether NMI interrupt processing is in progress. 0: No NMI interrupt processing 1: NMI interrupt currently being processed

4.2.4 NMI control

The NMI can be configured to generate an NMI upon a rising, falling or both edges at the NMI pin. To enable respectively disable the NMI and to configure the edge refer to *“Edge and Level Detection Configuration” on page 150*.

4.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers.

If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request has been acknowledged, the acknowledgement of other maskable interrupt requests is disabled and the interrupt disabled (DI) status is set.

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

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

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

4.3.1 Operation

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

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

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

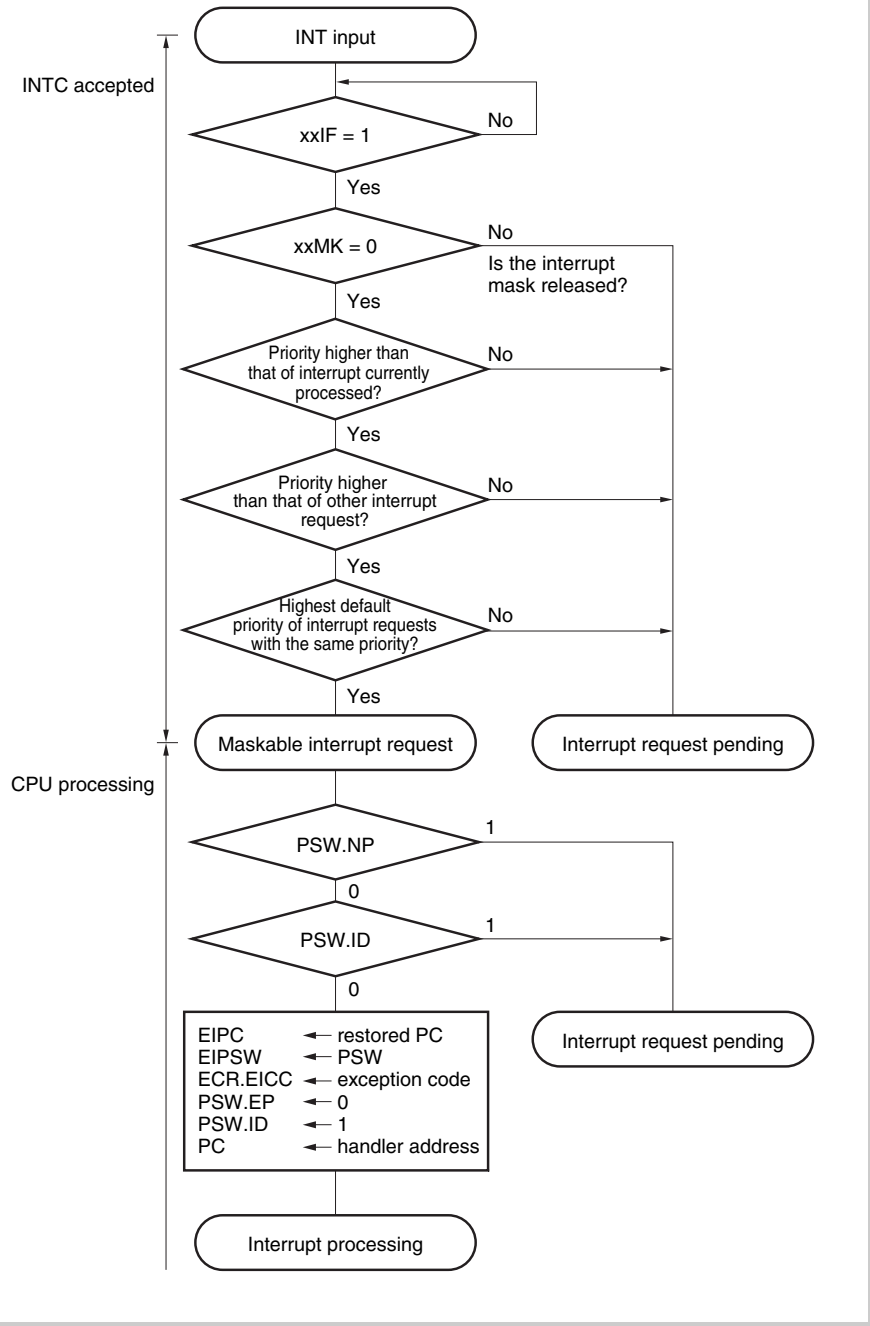


Figure 4-5 Maskable interrupt processing

Note For the ISPR register, see *"ISPR - In-service priority register" on page 148*.

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

4.3.2 Restore

Recovery from maskable interrupt processing is carried out by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

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

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

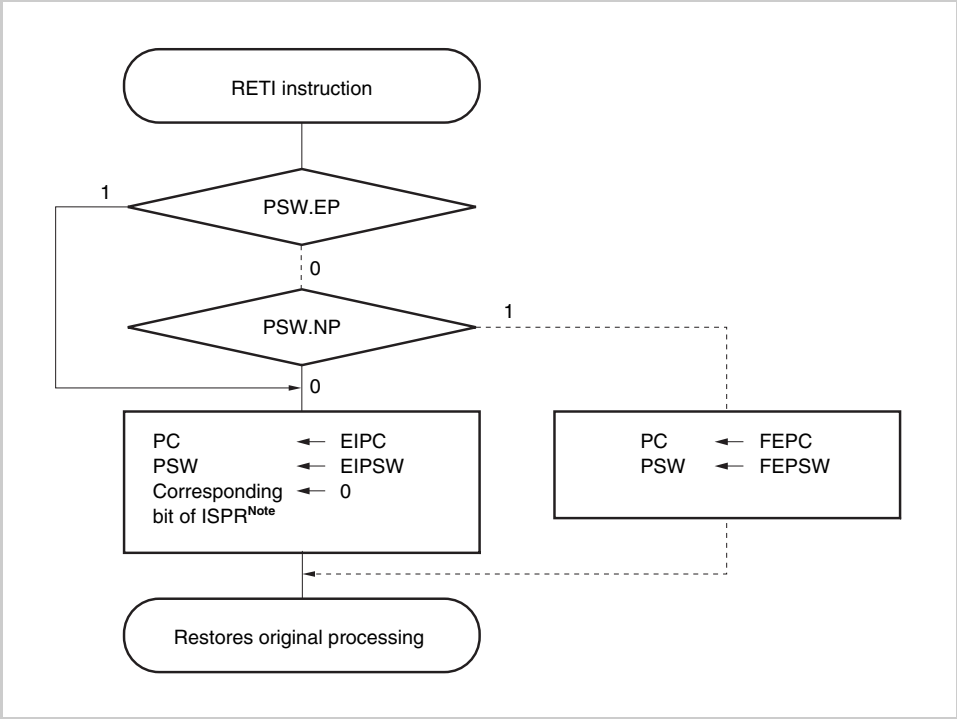


Figure 4-6 RETI instruction processing

Note 1. For the ISPR register, see *"ISPR - In-service priority register" on page 148*.

2. The solid lines show the CPU processing flow.

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

4.3.3 Priorities of maskable interrupts

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

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to the interrupt/exception source list table. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

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

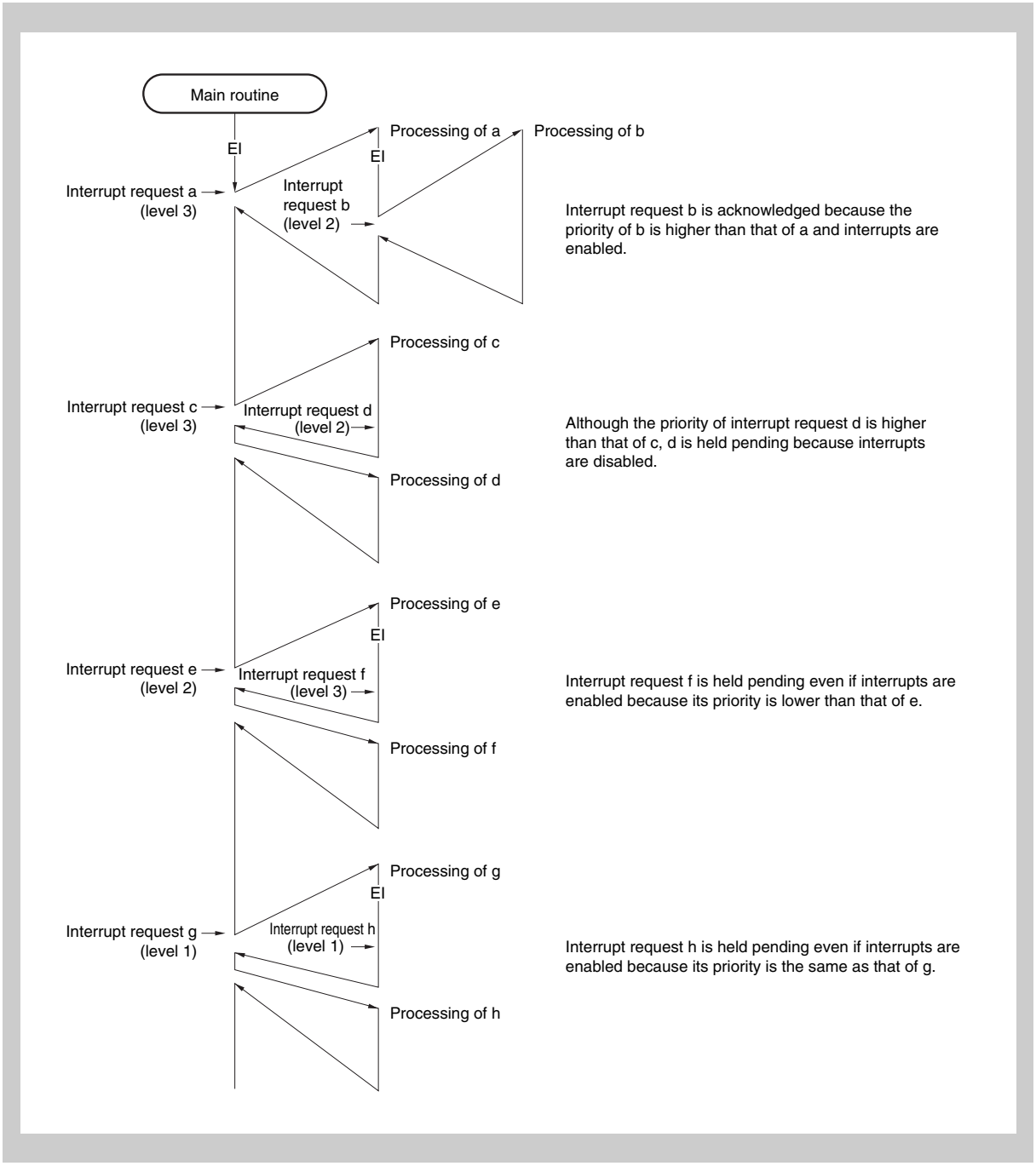


Figure 4-7 Example of processing in which another interrupt request is issued while an interrupt is being processed (1/2)

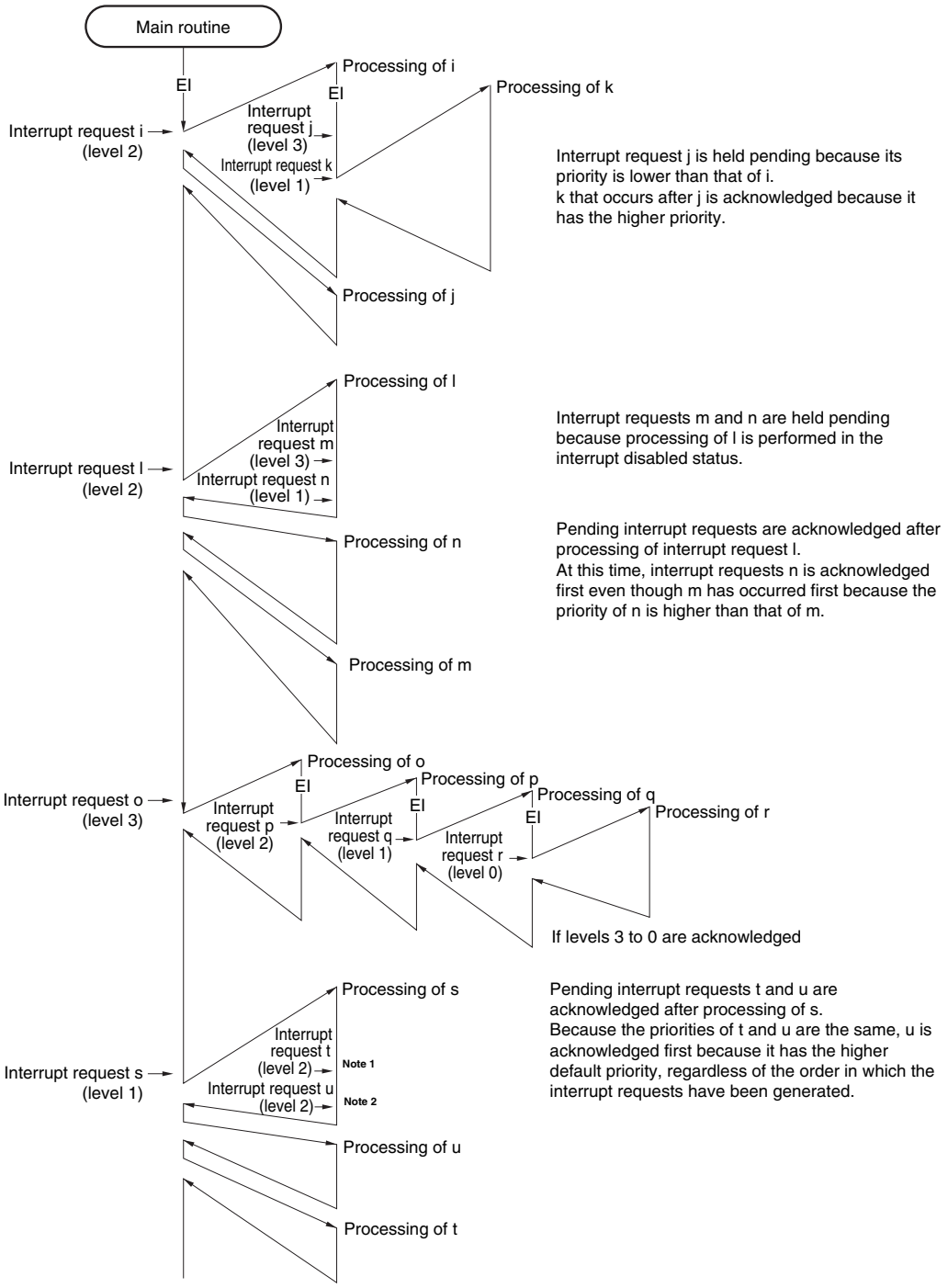


Figure 4-8 Example of processing in which another interrupt request is issued while an interrupt is being processed (2/2)

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

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

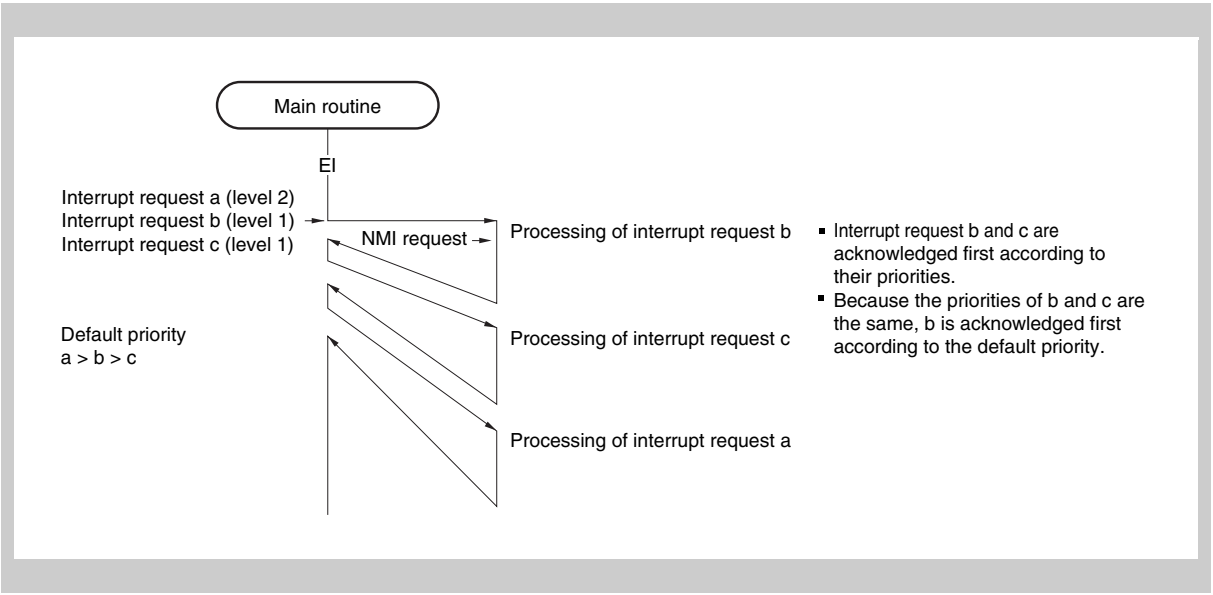


Figure 4-9 Example of processing interrupt requests simultaneously generated

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

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

4.3.4 xxICn - Maskable interrupts control register

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

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

	7	6	5	4	3	2	1	0	Address	Initial value
xxICn	xxIFn	xxMKn	0	0	0	xxPR2	xxPR1	xxPR0	FFFFF110H to FFFFF1F8H	47H

Bit position	Bit name	Function
7	xxIFn	This is an interrupt request flag. 0: Interrupt request not issued 1: Interrupt request issued The flag xxIFn is reset automatically by the hardware if an interrupt request is acknowledged.
6	xxMKn	This is an interrupt mask flag. 0: Enables interrupt processing 1: Disables interrupt processing (pending)
2 to 0	xxPR2 to xxPR0	8 levels of priority order are specified for each interrupt.

xxPR2	xxPR1	xxPR0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest)
0	0	1	Specifies level 1
0	1	0	Specifies level 2
0	1	1	Specifies level 3
1	0	0	Specifies level 4
1	0	1	Specifies level 5
1	1	0	Specifies level 6
1	1	1	Specifies level 7 (lowest)

Note xx: identification name of each peripheral unit. For a complete list of all interrupt control registers see *Table 4-3 on page 144*.

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

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

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

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

Address	Register	Bit						Associated interrupt
		7	6	5 - 3	2	1	0	
FFFFF15CH	TT0CCIC0	TT0CCIF0	TT0CCMK0	0	TT0CCPR02	TT0CCPR01	TT0CCPR00	INTTT0CC0
FFFFF15EH	TT0CCIC1	TT0CCIF1	TT0CCMK1	0	TT0CCPR12	TT0CCPR11	TT0CCPR10	INTTT0CC1
FFFFF160H	TT0ECIC	TT0ECIF	TT0ECMK	0	TT0ECPR2	TT0ECPR1	TT0ECPR0	INTTT0EC
FFFFF162H	TT1OVIC	TT1OVIF	TT1OVMK	0	TT1OVPR2	TT1OVPR1	TT1OVPR0	INTTT1OV
FFFFF164H	TT1CCIC0	TT1CCIF0	TT1CCMK0	0	TT1CCPR02	TT1CCPR01	TT1CCPR00	INTTT1CC0
FFFFF166H	TT1CCIC1	TT1CCIF1	TT1CCMK1	0	TT1CCPR12	TT1CCPR11	TT1CCPR10	INTTT1CC1
FFFFF168H	TT1ECIC	TT1ECIF	TT1ECMK	0	TT1ECPR2	TT1ECPR1	TT1ECPR0	INTTT1EC
FFFFF16AH	TAA0OVIC	TAA0OVIF	TAA0OVMK	0	TAA0OVPR2	TAA0OVPR1	TAA0OVPR0	INTTAA0OV
FFFFF16CH	TAA0CCIC0	TAA0CCIF0	TAA0CCMK0	0	TAA0CCPR02	TAA0CCPR01	TAA0CCPR00	INTTAA0CC0
FFFFF16EH	TAA0CCIC1	TAA0CCIF1	TAA0CCMK1	0	TAA0CCPR12	TAA0CCPR11	TAA0CCPR10	INTTAA0CC1
FFFFF170H	TAA1OVIC	TAA1OVIF	TAA1OVMK	0	TAA1OVPR2	TAA1OVPR1	TAA1OVPR0	INTTAA1OV
FFFFF172H	TAA1CCIC0	TAA1CCIF0	TAA1CCMK0	0	TAA1CCPR02	TAA1CCPR01	TAA1CCPR00	INTTAA1CC0
FFFFF174H	TAA1CCIC1	TAA1CCIF1	TAA1CCMK1	0	TAA1CCPR12	TAA1CCPR11	TAA1CCPR10	INTTAA1CC1
FFFFF176H	TAA2OVIC	TAA2OVIF	TAA2OVMK	0	TAA2OVPR2	TAA2OVPR1	TAA2OVPR0	INTTAA2OV
FFFFF178H	TAA2CCIC0	TAA2CCIF0	TAA2CCMK0	0	TAA2CCPR02	TAA2CCPR01	TAA2CCPR00	INTTAA2CC0
FFFFF17AH	TAA2CCIC1	TAA2CCIF1	TAA2CCMK1	0	TAA2CCPR12	TAA2CCPR11	TAA2CCPR10	INTTAA2CC1
FFFFF17CH	TAA3OVIC	TAA3OVIF	TAA3OVMK	0	TAA3OVPR2	TAA3OVPR1	TAA3OVPR0	INTTAA3OV
FFFFF17EH	TAA3CCIC0	TAA3CCIF0	TAA3CCMK0	0	TAA3CCPR02	TAA3CCPR01	TAA3CCPR00	INTTAA3CC0
FFFFF180H	TAA3CCIC1	TAA3CCIF1	TAA3CCMK1	0	TAA3CCPR12	TAA3CCPR11	TAA3CCPR10	INTTAA3CC1
FFFFF182H	TAA4OVIC	TAA4OVIF	TAA4OVMK	0	TAA4OVPR2	TAA4OVPR1	TAA4OVPR0	INTTAA4OV
FFFFF184H	TAA4CCIC0	TAA4CCIF0	TAA4CCMK0	0	TAA4CCPR02	TAA4CCPR01	TAA4CCPR00	INTTAA4CC0
FFFFF186H	TAA4CCIC1	TAA4CCIF1	TAA4CCMK1	0	TAA4CCPR12	TAA4CCPR11	TAA4CCPR10	INTTAA4CC1
FFFFF188H	TAA5OVIC	TAA5OVIF	TAA5OVMK	0	TAA5OVPR2	TAA5OVPR1	TAA5OVPR0	INTTAA5OV
FFFFF18AH	TAA5CCIC0	TAA5CCIF0	TAA5CCMK0	0	TAA5CCPR02	TAA5CCPR01	TAA5CCPR00	INTTAA5CC0
FFFFF18CH	TAA5CCIC1	TAA5CCIF1	TAA5CCMK1	0	TAA5CCPR12	TAA5CCPR11	TAA5CCPR10	INTTAA5CC1
FFFFF18EH	TAA6OVIC	TAA6OVIF	TAA6OVMK	0	TAA6OVPR2	TAA6OVPR1	TAA6OVPR0	INTTAA6OV
FFFFF190H	TAA6CCIC0	TAA6CCIF0	TAA6CCMK0	0	TAA6CCPR02	TAA6CCPR01	TAA6CCPR00	INTTAA6CC0
FFFFF192H	TAA6CCIC1	TAA6CCIF1	TAA6CCMK1	0	TAA6CCPR12	TAA6CCPR11	TAA6CCPR10	INTTAA6CC1
FFFFF194H	TAA7OVIC	TAA7OVIF	TAA7OVMK	0	TAA7OVPR2	TAA7OVPR1	TAA7OVPR0	INTTAA7OV
FFFFF196H	TAA7CCIC0	TAA7CCIF0	TAA7CCMK0	0	TAA7CCPR02	TAA7CCPR01	TAA7CCPR00	INTTAA7CC0
FFFFF198H	TAA7CCIC1	TAA7CCIF1	TAA7CCMK1	0	TAA7CCPR12	TAA7CCPR11	TAA7CCPR10	INTTAA7CC1
FFFFF19AH	TAA8OVIC	TAA8OVIF	TAA8OVMK	0	TAA8OVPR2	TAA8OVPR1	TAA8OVPR0	INTTAA8OV
FFFFF19CH	TAA8CCIC0	TAA8CCIF0	TAA8CCMK0	0	TAA8CCPR02	TAA8CCPR01	TAA8CCPR00	INTTAA8CC0
FFFFF19EH	TAA8CCIC1	TAA8CCIF1	TAA8CCMK1	0	TAA8CCPR12	TAA8CCPR11	TAA8CCPR10	INTTAA8CC1
FFFFF1A0H	TAA9OVIC	TAA9OVIF	TAA9OVMK	0	TAA9OVPR2	TAA9OVPR1	TAA9OVPR0	INTTAA9OV
FFFFF1A2H	TAA9CCIC0	TAA9CCIF0	TAA9CCMK0	0	TAA9CCPR02	TAA9CCPR01	TAA9CCPR00	INTTAA9CC0
FFFFF1A4H	TAA9CCIC1	TAA9CCIF1	TAA9CCMK1	0	TAA9CCPR12	TAA9CCPR11	TAA9CCPR10	INTTAA9CC1
FFFFF1A6H	BRGIC	BRGIF	BRGMK	0	BRGPR2	BRGPR1	BRGPR0	INTBRG0 INTBRG1
FFFFF1A8H	BRGIC2	BRGIF2	BRGMK2	0	BRGPR22	BRGPR21	BRGPR20	INTBRG2
FFFFF1AAH	FRIC0	FRIF0	FRMK0	0	FRPR02	FRPR01	FRPR00	INTFR0

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

Address	Register	Bit						Associated interrupt
		7	6	5 - 3	2	1	0	
FFFFF1ACH	FRIC1	FRIF1	FRMK1	0	FRPR12	FRPR11	FRPR10	INTFR1
FFFFF1AEH	FRIC2	FRIF2	FRMK2	0	FRPR22	FRPR21	FRPR20	INTFR2
FFFFF1B0H	FRIC3	FRIF3	FRMK3	0	FRPR32	FRPR31	FRPR30	INTFR3
FFFFF1B2H	FRIBIC	FRIBIF	FRIBMK	0	FRIBPR2	FRIBPR1	FRIBPR0	INTFRIB
FFFFF1B4H	FROBIC	FROBIF	FROBMK	0	FROBPR2	FROBPR1	FROBPR0	INTFROB
FFFFF1B6H	C0ERRIC	C0ERRIF	C0ERRMK	0	C0ERRPR2	C0ERRPR1	C0ERRPR0	INTC0ERR
FFFFF1B8H	C0WUPIC	C0WUPIF	C0WUPMK	0	C0WUPPR2	C0WUPPR1	C0WUPPR0	INTC0WUP
FFFFF1BAH	C0RECIC	C0RECIF	C0RECMK	0	C0RECPR2	C0RECPR1	C0RECPR0	INTC0REC
FFFFF1BCH	C0TRXIC	C0TRXIF	C0TRXMK	0	C0TRXPR2	C0TRXPR1	C0TRXPR0	INTC0TRX
FFFFF1BEH	C1ERRIC	C1ERRIF	C1ERRMK	0	C1ERRPR2	C1ERRPR1	C1ERRPR0	INTC1ERR
FFFFF1C0H	C1WUPIC	C1WUPIF	C1WUPMK	0	C1WUPPR2	C1WUPPR1	C1WUPPR0	INTC1WUP
FFFFF1C2H	C1RECIC	C1RECIF	C1RECMK	0	C1RECPR2	C1RECPR1	C1RECPR0	INTC1REC
FFFFF1C4H	C1TRXIC	C1TRXIF	C1TRXMK	0	C1TRXPR2	C1TRXPR1	C1TRXPR0	INTC1TRX
FFFFF1C6H	C2ERRIC	C2ERRIF	C2ERRMK	0	C2ERRPR2	C2ERRPR1	C2ERRPR0	INTC2ERR
FFFFF1C8H	C2WUPIC	C2WUPIF	C2WUPMK	0	C2WUPPR2	C2WUPPR1	C2WUPPR0	INTC2WUP
FFFFF1CAH	C2RECIC	C2RECIF	C2RECMK	0	C2RECPR2	C2RECPR1	C2RECPR0	INTC2REC
FFFFF1CEH	C3ERRIC	C3ERRIF	C3ERRMK	0	C3ERRPR2	C3ERRPR1	C3ERRPR0	INTC3ERR
FFFFF1D0H	C3WUPIC	C3WUPIF	C3WUPMK	0	C3WUPPR2	C3WUPPR1	C3WUPPR0	INTC3WUP
FFFFF1D2H	C3RECIC	C3RECIF	C3RECMK	0	C3RECPR2	C3RECPR1	C3RECPR0	INTC3REC
FFFFF1D6H	CB0TIC	CB0TIF	CB0TMK	0	CB0TPR2	CB0TPR1	CB0TPR0	INTCB0T INTUC2T
FFFFF1D8H	CB0RIC	CB0RIF	CB0RMK	0	CB0RPR2	CB0RPR1	CB0RPR0	INTCB0R INTUC2R INTP13
FFFFF1DAH	CB0REIC	CB0REIF	CB0REMK	0	CB0REPR2	CB0REPR1	CB0REPR0	INTCB0RE INTUC2RE
FFFFF1DCH	CB1TIC	CB1TIF	CB1TMK	0	CB1TPR2	CB1TPR1	CB1TPR0	INTCB1T
FFFFF1DEH	CB1RIC	CB1RIF	CB1RMK	0	CB1RPR2	CB1RPR1	CB1RPR0	INTCB1R
FFFFF1E0H	CB1REIC	CB1REIF	CB1REMK	0	CB1REPR2	CB1REPR1	CB1REPR0	INTCB1RE
FFFFF1E2H	CE0OFIC	CE0OFIF	CE0OFMK	0	CE0OFPR2	CE0OFPR1	CE0OFPR0	INTCE0OF
FFFFF1E4H	CE0CIC	CE0CIF	CE0CMK	0	CE0CPR2	CE0CPR1	CE0CPR0	INTCE0C
FFFFF1E6H	CE1OFIC	CE1OFIF	CE1OFMK	0	CE1OFPR2	CE1OFPR1	CE1OFPR0	INTCE1OF
FFFFF1E8H	CE1CIC	CE1CIF	CE1CMK	0	CE1CPR2	CE1CPR1	CE1CPR0	INTCE1C
FFFFF1EAH	UC0REIC	UC0REIF	UC0REMK	0	UC0REPR2	UC0REPR1	UC0REPR0	INTUC0RE
FFFFF1ECH	UC0TIC	UC0TIF	UC0TMK	0	UC0TPR2	UC0TPR1	UC0TPR0	INTUC0T
FFFFF1EEH	UC1REIC	UC1REIF	UC1REMK	0	UC1REPR2	UC1REPR1	UC1REPR0	INTUC1RE
FFFFF1F0H	UC1TIC	UC1TIF	UC1TMK	0	UC1TPR2	UC1TPR1	UC1TPR0	INTUC1T
FFFFF1F2H	ADIC0	ADIF0	ADMK0	0	ADPR02	ADPR01	ADPR00	INTAD0
FFFFF1F4H	ADIC1	ADIF1	ADMK1	0	ADPR12	ADPR11	ADPR10	INTAD1
FFFFF1F6H	DMAIC2	DMAIF2	DMAMK2	0	DMAPR22	DMAPR21	DMAPR20	INTDMA2
FFFFF1F8H	DMAIC3	DMAIF3	DMAMK3	0	DMAPR32	DMAPR31	DMAPR30	INTDMA3 INTFL

4.3.5 IMR0 to IMR7 - Interrupt mask registers

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

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

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

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

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

	15	14	13	12	11	10	9	8	Address	Initial value
IMR0	TS0OVMK	PMK12	PMK11	PMK10	PMK9	PMK8	PMK7	PMK6	FFFFF100H	FFFFH
	7	6	5	4	3	2	1	0		
	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	OSDMK	ERRMK		

	15	14	13	12	11	10	9	8	Address	Initial value
IMR1	TS1CCMK4	TS1CCMK3	TS1CCMK2	TS1CCMK1	TS1CCMK0	TS1OVMK	TS0WNMK	TS0ERMK	FFFFF102H	FFFFH
	7	6	5	4	3	2	1	0		
	TS0ODMK	TS0CDMK0	TS0CCMK5	TS0CCMK4	TS0CCMK3	TS0CCMK2	TS0CCMK1	TS0CCMK0		

	15	14	13	12	11	10	9	8	Address	Initial value
IMR2	TTA0CCMK1	TTA0CCMK0	TTA0OVMK	TT1ECMK	TT1CCMK1	TT1CCMK0	TT1OVMK	TT0ECMK	FFFFF104H	FFFFH
	7	6	5	4	3	2	1	0		
	TT0CCMK1	TT0CCMK0	TT0OVMK	TS1WNMK	TS1ERMK	TS1ODMK	TS1CDMK0	TS1CCMK5		

	15	14	13	12	11	10	9	8	Address	Initial value
IMR3	TTA6OVMK	TTA5CCMK1	TTA5CCMK0	TTA5OVMK	TTA4CCMK1	TTA4CCMK0	TTA4OVMK	TTA3CCMK1	FFFFF106H	FFFFH
	7	6	5	4	3	2	1	0		
	TTA3CCMK0	TTA3OVMK	TTA2CCMK1	TTA2CCMK0	TTA2OVMK	TTA1CCMK1	TTA1CCMK0	TTA1OVMK		

	15	14	13	12	11	10	9	8	Address	Initial value
IMR4	FRMK2	FRMK1	FRMK0	BRGMK2	BRGMK	TTA9CCMK1	TTA9CCMK0	TTA9OVMK	FFFFF108H	FFFFH
	7	6	5	4	3	2	1	0		
	TTA8CCMK1	TTA8CCMK0	TTA8OVMK	TTA7CCMK1	TTA7CCMK0	TTA7OVMK	TTA6CCMK1	TTA6CCMK0		

	15	14	13	12	11	10	9	8	Address	Initial value
IMR5	1	1	1	1	1	C1TRXMK	C1RECMK	C1WUPMK	FFFFF10AH	FFFFH
	7	6	5	4	3	2	1	0		

	C1ERRMK	C0TRXMK	C0RECMK	C0WUPMK	C0ERRMK	FROBMK	FRIBMK	FRMK3

	15	14	13	12	11	10	9	8	Address	Initial value
IMR6	UC1REMK	UC0TMK	UC0REMK	CE1CMK	CE1OFMK	CE0CMK	CE0OFMK	CB1REMK	FFFFF10CH	FFFFH

7	6	5	4	3	2	1	0
CB1RMK	CB1TMK	CB0REMK	CB0RMK	CB0TMK	1	1	1

	15	14	13	12	11	10	9	8	Address	Initial value
IMR7	0	0	0	0	0	0	0	0	FFFFF10EH	001FH

7	6	5	4	3	2	1	0
0	0	0	DMAMK3	DMAMK2	ADMK1	ADMK0	UC1TMK

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

4.3.6 ISPR - In-service priority register

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt request is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request having the highest priority is automatically reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

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

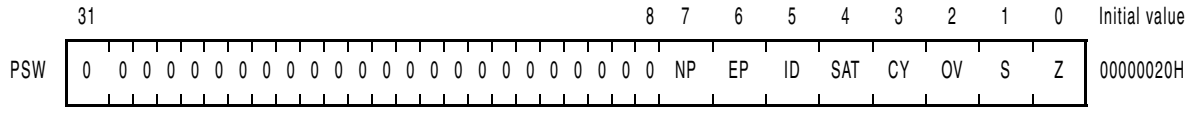
	7	6	5	4	3	2	1	0	Address	Initial value
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	FFFFF1FAH	00H

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

Note n = 0 to 7 (priority level)

4.3.7 Maskable interrupt status flag (ID)

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



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

4.3.8 External maskable interrupts

This microcontroller provides 14 maskable external interrupts INTP0 to INTP13 with the following features:

- Analog input filter (refer to *"Analog filtered inputs" on page 87*)
- Interrupt detection selectable for each interrupt input:
  - Rising edge
  - Falling edge
  - Both edges: rising and falling edge
- Wakeup capability from stand-by mode of INTPn upon
  - Rising edge
  - Falling edge
  - Both edges: rising and falling edge

For configuration of the external interrupt events refer to *"Edge and Level Detection Configuration" on page 150*.

4.4 Edge and Level Detection Configuration

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

Table 4-4 Edge specification of interrupts

ESN1	ESN0	Function	
		INTP0 to INTP13	NMI
0	0	falling edge	falling edge
0	1	rising edge	rising edge
1	0	disabled (no NMI generated)	Setting prohibited
1	1	falling and rising edge	falling and rising edge

(1) INTM0 - External interrupt edge specification register

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

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

Address FFFF F880$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

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

Table 4-5 INTM0 register contents

Bit position	Bit name	Function
7	ES21	Specifies the edge detection at terminal INTP2. See *Table 4-4 on page 150*.
6	ES20	
5	ES11	Specifies the edge detection at terminal INTP1. See *Table 4-4 on page 150*.
4	ES10	
3	ES01	Specifies the edge detection at terminal INTP0. See *Table 4-4 on page 150*.
2	ES00	
1	ESN1	Specifies the edge detection at terminal NMI. See *Table 4-4 on page 150*.
0	ESN0	

(2) INTM1 - External interrupt edge specification register

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

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

Address FFFF F882$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
ES61	ES60	ES51	ES50	ES41	ES40	ES31	ES30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
INTP6		INTP5		INTP4		INT3	

Table 4-6 INTM1 register contents

Bit position	Bit name	Function
7	ES61	Specifies the edge detection at terminal INTP6. See *Table 4-4 on page 150*.
6	ES60	
5	ES51	Specifies the edge detection at terminal INTP5. See *Table 4-4 on page 150*.
4	ES50	
3	ES41	Specifies the edge detection at terminal INTP4. See *Table 4-4 on page 150*.
2	ES40	
1	ES31	Specifies the edge detection at terminal INTP3. See *Table 4-4 on page 150*.
0	ES30	

(3) INTM2 - External interrupt edge specification register

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

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

Address FFFF F884$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

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

Table 4-7 INTM2 register contents

Bit position	Bit name	Function
7	ES101	Specifies the edge detection at terminal INTP10. See *Table 4-4 on page 150*.
6	ES100	
5	ES91	Specifies the edge detection at terminal INTP9. See *Table 4-4 on page 150*.
4	ES90	
3	ES81	Specifies the edge detection at terminal INTP8. See *Table 4-4 on page 150*.
2	ES80	
1	ES71	Specifies the edge detection at terminal INTP7. See *Table 4-4 on page 150*.
0	ES70	

(4) INTM3 - External interrupt edge specification register

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

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

Address FFFF F886$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
0	0	ES131	ES130	ES121	ES120	ES111	ES110
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		INTP13		INTP12		INTP11	

Table 4-8 INTM3 register contents

Bit position	Bit name	Function
5	ES131	Specifies the edge detection at terminal INTP13. See *Table 4-4 on page 150*.
4	ES130	
3	ES121	Specifies the edge detection at terminal INTP12. See *Table 4-4 on page 150*.
2	ES120	
1	ES111	Specifies the edge detection at terminal INTP11. See *Table 4-4 on page 150*.
0	ES110	

4.5 Interrupt Sharing

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

Non-maskable interrupts

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

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

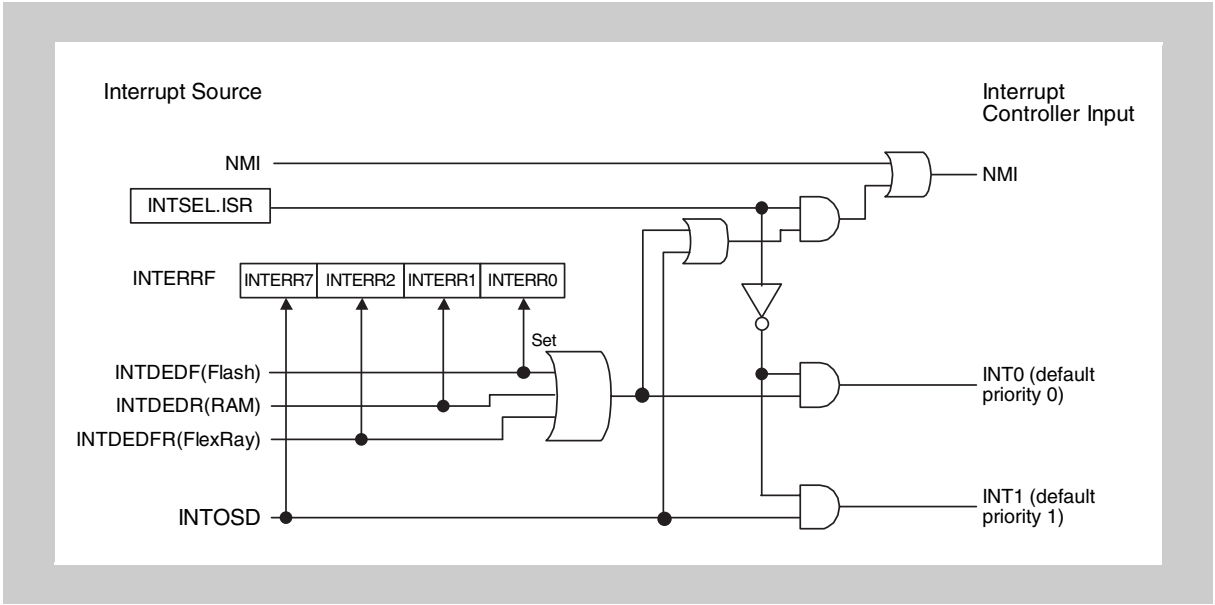


Figure 4-10 Interrupt sharing for NMI

Maskable interrupts

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

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

Table 4-9 on page 154 lists the shared maskable interrupts and, if available, their corresponding interrupt selection bit. For more details on the interrupt sources see the interrupt/exception source list in *Table 4-1 on page 126*.

Table 4-9 Shared maskable interrupts

Shared interrupts	Interrupt selection		Valid interrupt sources	Interrupt source identification	Default priority
INTDEDF / INTDEDR / INTDEFR	logical “or” combination		INTDEDF / INTDEDR / INTDEFR	status flags in the interrupt source module	0
INTP4 / INTUC0R	logical “or” combination		INTP4 / INTUC0R	INTUC0R identified by status flags in UARTC0. If only INTUC0R interrupts shall be served, INTP4 interrupts can be suppressed by setting INTM1.ES4[1:0] = 10_B.	6
INTP5 / INTUC1R	logical “or” combination		INTP5 / INTUC1R	INTUC1R identified by status flags in UARTC1. If only INTUC1R interrupts shall be served, INTP5 interrupts can be suppressed by setting INTM1.ES5[1:0] = 10_B.	7
INTBRG0 / INTBRG1	INTSEL.BRGSSR	0	INTBRG0	determined by INTSEL.BRGSSR	75
		1	INTBRG1		
INTCB0T / INTUC2T	PFC4.PFC40[a]	0	INTCB0T	determined by PFC4.PFC40	99
		1	INTUC2T		
INTCB0R / INTUC2R / INTP13	PFC4.PFC40[a]	0	INTCB0R	determined by PFC4.PFC40	100
		1	INTP13 / INTUC2R logical “or” combination	determined by PFC4.PFC40 and status flags in UARTC2 If only INTUC2R interrupts shall be served, INTP13 interrupts can be suppressed by setting INTM3.ES13[1:0] = 10_B.	
INTCB0RE / INTUC2RE	PFC4.PFC41[a]	0	INTCB0RE	determined by PFC4.PFC41	101
		1	INTUC2RE		
INTDMA3 / INTFL	INTSEL.INTSEL4	0	INTDMA3	–	116
		1	INTFL	–	

a) For a description of registers PFCn see *“Pin Functions” on page 31*

(1) INTSEL - Interrupt share enable register

The 8-bit INTSEL register specifies

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

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

Address FFFF F6D0$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

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

Caution
1. The bits INTSEL.bit3 and INTSEL.bit2 must be set to “1” after reset release and must not be changed afterwards.
2. The bit INTSEL1.bit1 must not be changed from its default value “0”.

Table 4-10 INTSEL register contents

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

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

(2) INTERRF - Interrupt source flag register

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

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

Address FFFF F6D2$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
INTERR7	0	0	0	0	INTERR2	INTERR1	INTERR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 4-11 INTTERF register contents

Bit position	Bit name	Function
7	INTERR7	Displays the status of interrupt INTOSD (Clock Monitor) 0: Interrupt not generated 1: Interrupt generated
2	INTERR2	Displays the status of interrupt INTDEDFR (FlexRay) 0: Interrupt not generated 1: Interrupt generated
1	INTERR1	Displays the status of interrupt INTDEDR (RAM) 0: Interrupt not generated 1: Interrupt generated
0	INTERR0	Displays the status of interrupt INTDEDF (Flash) 0: Interrupt not generated 1: Interrupt generated

Note

1. After the CPU has processed the associated interrupt routine, the related flag in the INTERRF register must be reset to '0' by a bit manipulation instruction. If a further interrupt occurs while the associated bit in the register is reset, the new interrupt request will also be discarded.
2. Writing to ROMEAD register is prohibited while INTERR0 bit is "1". Writing to RAMEAD register is prohibited while INTERR1 bit is "1".
3. After occurrence of an INTDEDFR (FlexRay double error detection) event, the bit FRMHDS.DMR in the FlexRay Controller must be reset to '0' additionally.

4.6 Software Exception

4.6.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

1. Saves the restored PC to EIPC.
2. Saves the current PSW to EIPSW.
3. Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
4. Sets the EP and ID bits of the PSW.
5. Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

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

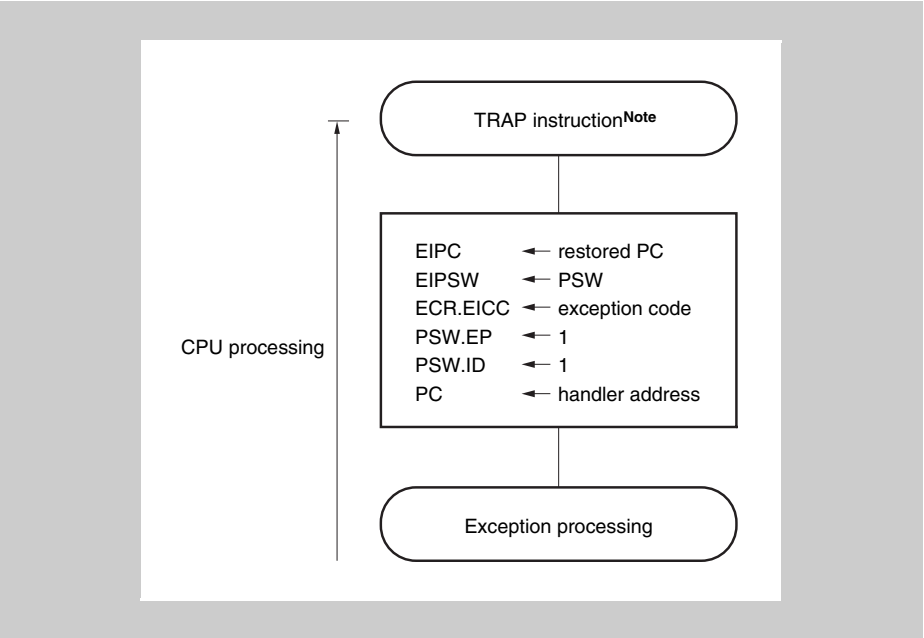


Figure 4-11 Software exception processing

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

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

4.6.2 Restore

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

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

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

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

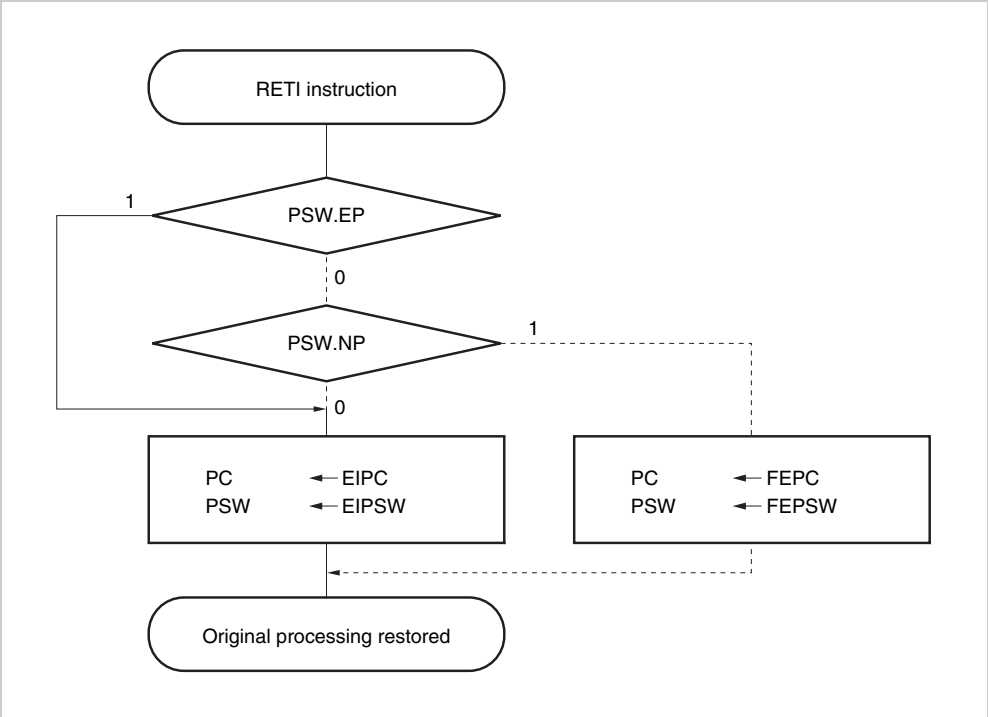


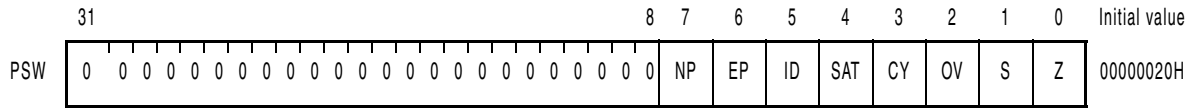
Figure 4-12 RETI instruction processing

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

Note The solid lines show the CPU processing flow.

4.6.3 Exception status flag (EP)

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



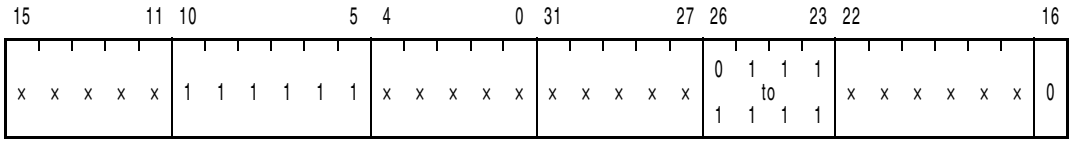
Bit position	Bit name	Function
6	EP	Shows that exception processing is in progress. 0: Exception processing not in progress. 1: Exception processing in progress.

4.7 Exception Trap

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

4.7.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 23 to 26) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.



Note ×: Arbitrary

(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine:

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

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

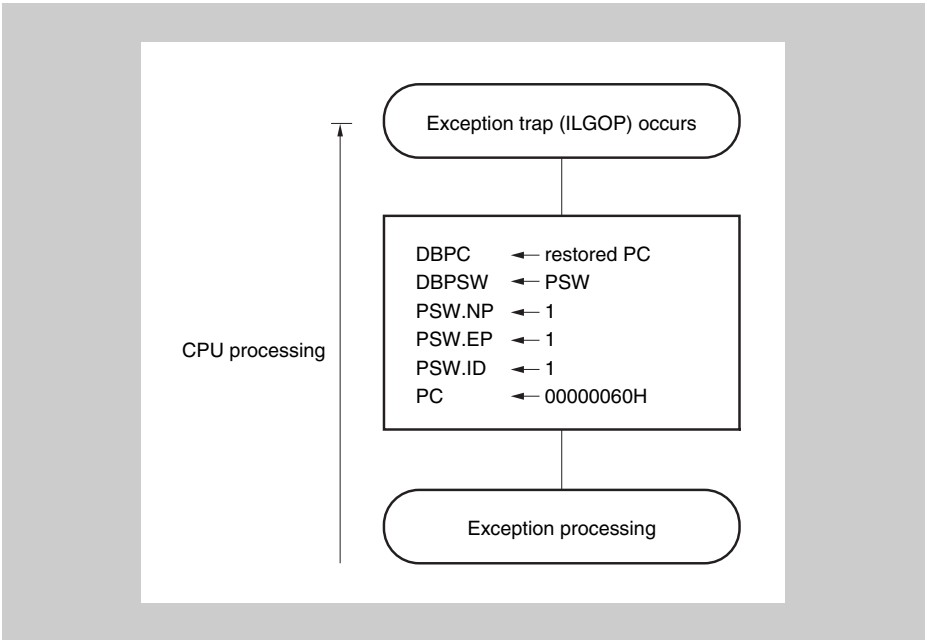


Figure 4-13 Exception trap processing

(2) Restore

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

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

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

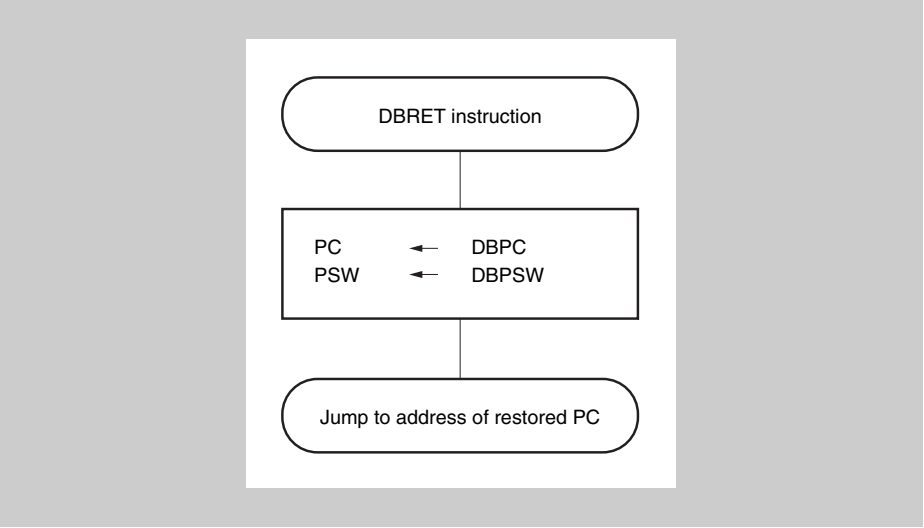


Figure 4-14 Restore processing from exception trap

4.7.2 Debug trap

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

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

(1) Operation

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

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

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

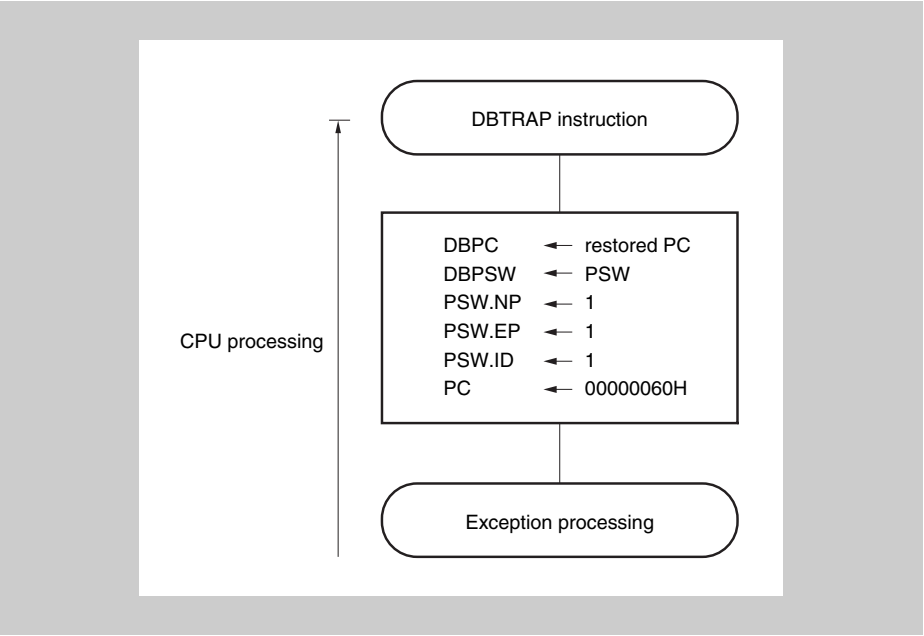


Figure 4-15 Debug trap processing

(2) Restore

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

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

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

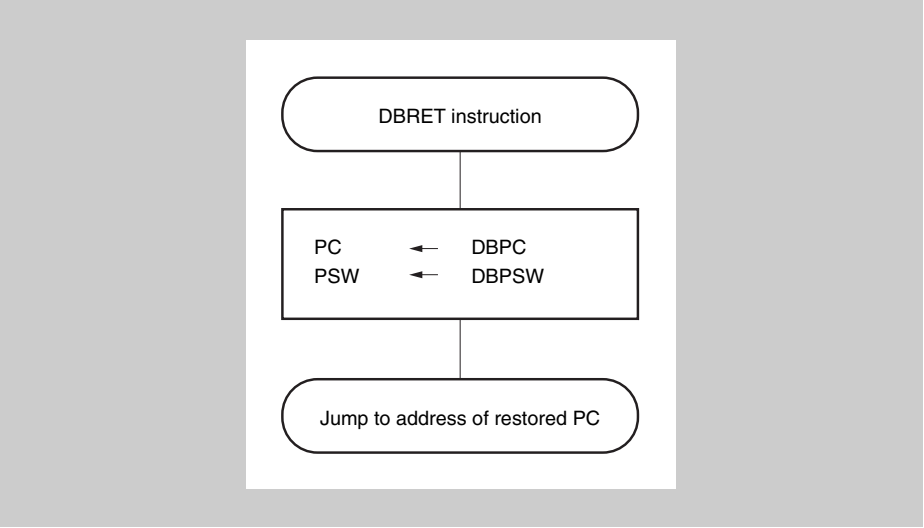


Figure 4-16 Restore processing from debug trap

4.8 Multiple Interrupt Processing Control

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

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

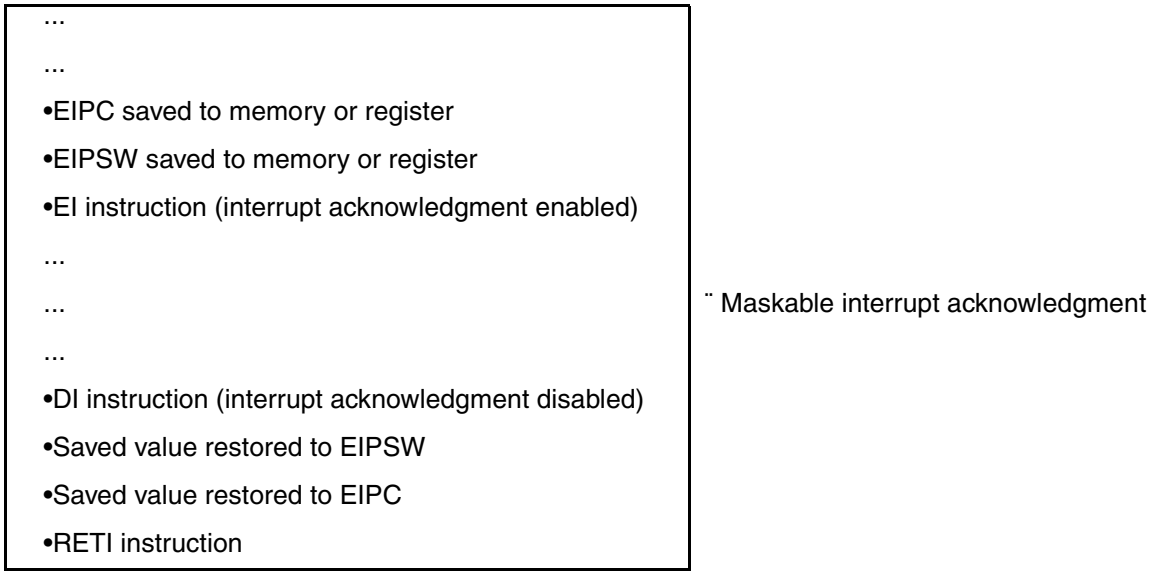
Maskable interrupt multiple processing control is executed when an interrupt has an enable status (ID = 0). Thus, if multiple interrupts are executed, it is necessary to have an interrupt enable status (ID = 0) even for an interrupt processing routine.

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

This is accomplished by the following procedure.

(1) Acknowledgment of maskable interrupts in service program

Service program of maskable interrupt or exception



(2) Generation of exception in service program

Service program of maskable interrupt or exception

... ... •EIPC saved to memory or register •EIPSW saved to memory or register ... •TRAP instruction ... •Saved value restored to EIPSW •Saved value restored to EIPC •RETI instruction	¨ Exception such as TRAP instruction acknowledged.

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

The priority order of maskable interrupts is as follows.

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 >
Level 5 > Level 6 > Level 7 (Low)

Interrupt processing that has been suspended as a result of multiple processing control is resumed after the processing of the higher priority interrupt has been completed and the RETI instruction has been executed.

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

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

4.9 Interrupt Response Time

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

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

- During software or hardware STOP mode
- When an external bus is accessed
- When there are two or more successive interrupt request non-sampling instructions (see *"Periods in Which Interrupts Are Not Acknowledged" on page 167*).
- When the interrupt control register is accessed

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

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

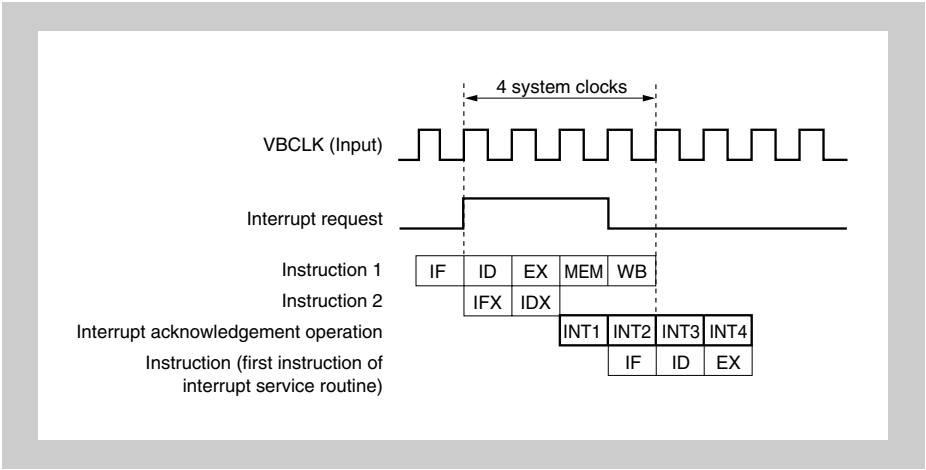


Figure 4-17 Pipeline operation at interrupt request acknowledgment (outline)

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

Table 4-12 Interrupt response time

Interrupt response time (internal system clocks)			Condition
	Internal interrupt	External interrupt	
Minimum	4	4 + digital noise filter delay	The following cases are exceptions: • In IDLE/software STOP mode • External bit access • Two or more interrupt request non-sample instructions are executed • Access to interrupt control register
Maximum	8	8 + digital noise filter delay	

4.10 Periods in Which Interrupts Are Not Acknowledged

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

The interrupt request non-sampling instructions are as follows:

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

Chapter 5 CRC Function (CRC)

The microcontroller offers two different instances for Cyclic Redundancy Check (CRC) generation, CPU-CRC and DATA-CRC.

5.1 Features

5.1.1 CPU-CRC

The CPU-CRC function can be used to support user self test software designed for testing the CPU core function. By observing dedicated CPU internal states and automatically accumulating a CRC signature by hardware corresponding routines for signature generation formerly realized in software may become obsolete. Proper function of such self test software in conjunction with the CRC unit and impact on the achieved fault coverage still needed to be carefully checked by the user.

- 32-bit Ethernet CRC ($04C11DB7_H$)
  ($X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X^1+1$)
- CRC generation to an arbitrary data block length
- 5 fix assigned channels for different observation points
  - Channel 0: synchronous register file write data
  - Channel 1: the register number of PSWSTATE and register file
  - Channel 2: IRAM address (only at write instructions)
  - Channel 3: IRAM write data (only at write instruction)
  - Channel 4: asynchronous register file write data

  Throughout this chapter the channel numbers are indicated by “n” (n = 0 to 4)
- CPU internal states are automatically stored in the CPU-CRC input register on each trigger event and the generated CRC is stored in the CPU-CRC data register.

5.1.2 DATA-CRC

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

- 32-bit Ethernet CRC ($04C11DB7_H$)
  ($X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X^1+1$)
  Input data reflected
  Output data reflected
  Final XOR with $FFFF\ FFFF_H$
- 16-bit CCITT CRC (1021_H)
  ($X^{16}+X^{12}+X^5+1$)
  Input data not reflected
  Output data not reflected
  Final XOR with 0000_H

- CRC generation to an arbitrary data block length
- One channel for DATA-CRC generation
- After initialization of the DATA-CRC input register every write access to the DATA-CRC input register generates a new CRC according to the chosen polynomial and the result is stored in the CPU-CRC data register.

5.2 Configuration

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

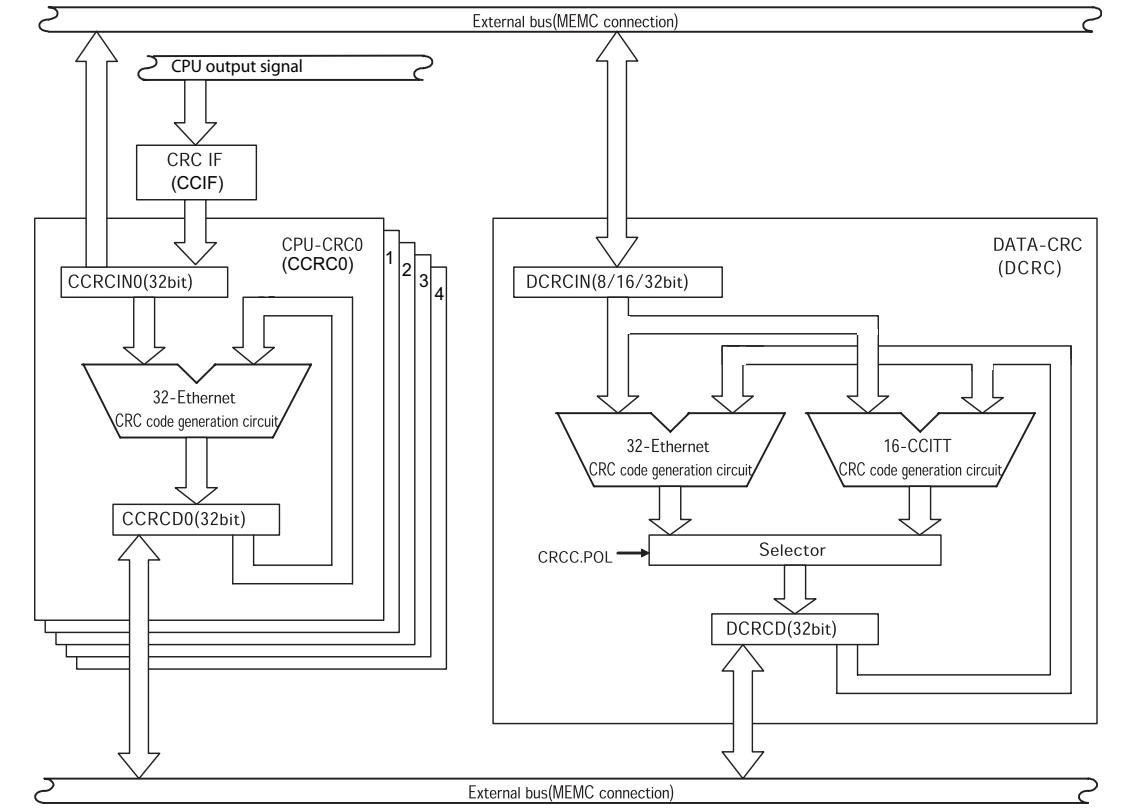


Figure 5-1 Block diagram of CPU-CRC and DATA-CRC

5.3 CRC Registers

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

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

Table 5-1 CRC registers overview

Register name	Shortcut	Address
CPU-CRC input register 0	CCRCIN0	<base>
CPU-CRC input register 1	CCRCIN1	<base> + 08_H
CPU-CRC input register 2	CCRCIN2	<base> + 10_H
CPU-CRC input register 3	CCRCIN3	<base> + 18_H
CPU-CRC input register 4	CCRCIN4	<base> + 20_H
CPU-CRC data register 0	CCRCD0	<base> + 04_H
CPU-CRC data register 1	CCRCD1	<base> + $0C_H$
CPU-CRC data register 2	CCRCD2	<base> + 14_H
CPU-CRC data register 3	CCRCD3	<base> + $1C_H$
CPU-CRC data register 4	CCRCD4	<base> + 24_H
DATA-CRC input register	DCRCIN	<base> + 30_H
DATA-CRC data register	DCRCD	<base> + 34_H
CRC control register	CRCC	<base> + 40_H

Base address The base address of the CRC module is 0FE0 0000_H.

(1) CCRCINn - CPU-CRC input register n (n = 0 - 4)

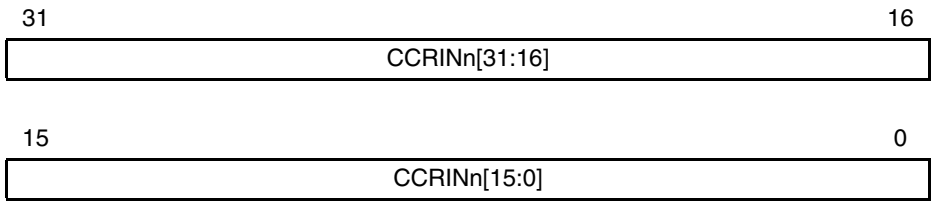
The CCRCINn register is a 32-bit data input register that holds the input data for CRC calculation.

The register is latched by actions generated by CPU (refer to below).

Access This register is read-only in 32-bit units.

Address CCRCIN0: <base>, CCRCIN1: <base> + 08_H
CCRCIN2: <base> + 10_H, CCRCIN3: <base> + 18_H
CCRCIN4: <base> + 20_H

Initial Value 0000 0000_H. The register is cleared by any reset.



Output signal latch The CPU output signals are latched by the following action:

- CCRCIN0: synchronous register file write data
- CCRCIN1: PSWSTATE only bits 9-0, bits14 to 0 register number of register file and bits 31 to 15 are set to "0"
- CCRCIN2: at IRAM write access only address bits 27 to 2, the remaining bits are set to “0”
- CCRCIN3: IRAM data, only at write access. Always 32 bit are used for input, at byte or half word accesses the invalid bytes are set to “0”
- CCRCIN4: asynchronous register file write data

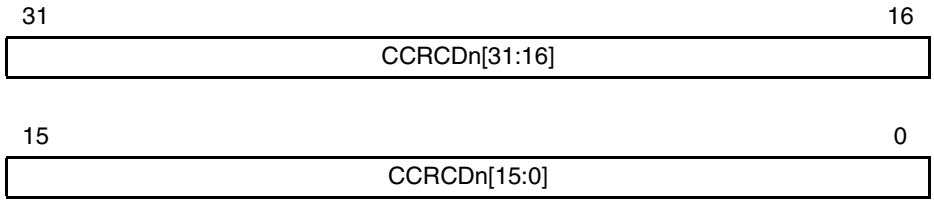
(2) CCRCDn - CPU-CRC data register n (n = 0 - 4)

This register is a 32-bit register for the result storage of the CPU-CRC generated by the 32-bit Ethernet polynomial.

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

Address CCRCD0: <base> + 04_H, CCRCD1: <base> + $0C_H$
CCRCD2: <base> + 14_H, CCRCD3: <base> + $1C_H$
CCRCD4: <base> + 24_H

Initial Value $0000\ 0000_H$. The register is cleared by any reset.



(3) DCRCIN - DATA-CRC input register

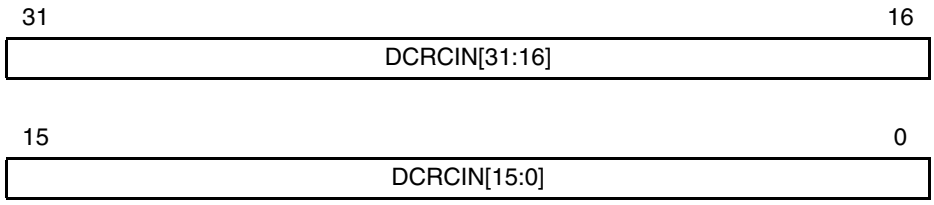
The DCRCIN register is a 32-bit data input register that holds the input data for DATA-CRC calculation. The effective bit width used for CRC calculation must be set at CRCC.ISZn bits of the CRC control register.

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

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

Address <base> + 30_H

Initial Value $0000\ 0000_H$. The register is cleared by any reset.



(4) DCRCD - DATA-CRC data register

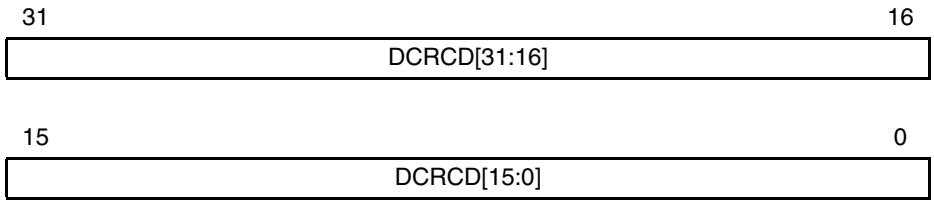
This register is a 32-bit register for the result storage of the DATA-CRC generated by the 32bit Ethernet or 16-bit CCITT polynomial.

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

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

Address <base> + 34_H

Initial Value $0000\ 0000_H$. The register is cleared by any reset.



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

(5) CRCC - CRC control register

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

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

Address <base> + 40_H

Initial Value 0000_H. The register is cleared by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8
0	0	0	CRCEN4	CRCEN3	CRCEN2	CRCEN1	CRCEN0

7	6	5	4	3	2	1	0
0	0	0	0	0	ISZ1	ISZ0	POL

CRCENn	Enable / disable CPU-CRC generation for related channel n
0	CPU-CRC generation disabled
1	CPU-CRC generation enabled

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

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

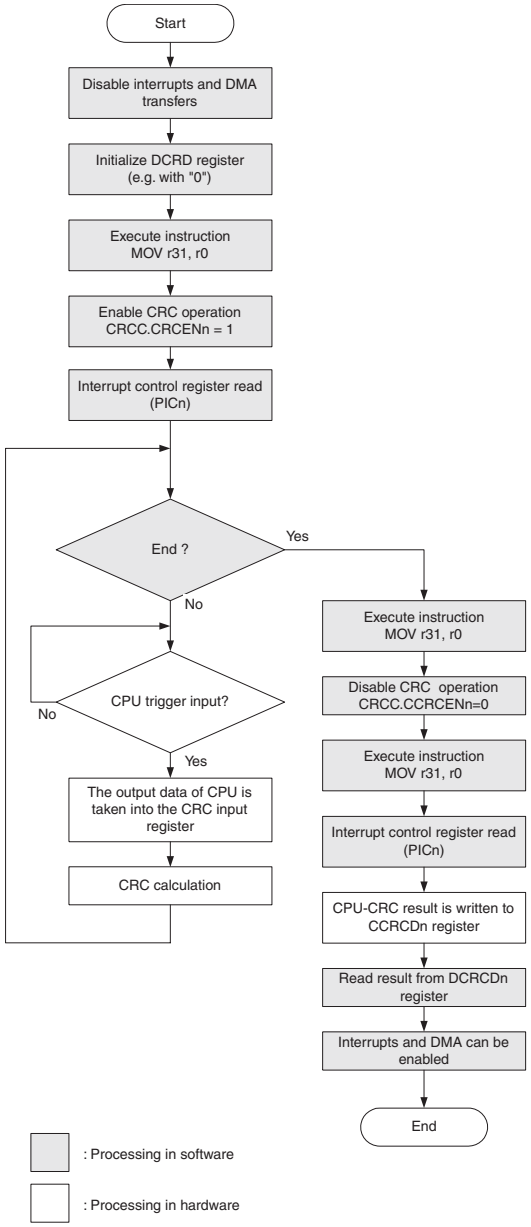
Note After changing the DATA-CRC generating function (CRCC.POL) the DCRCD register must be initialized.

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

5.4 Operation

5.4.1 CPU-CRC

The following flow chart shows the CPU-CRC generating sequence.



Note 1: Read one interrupt control register (there is no dedicated interrupt control register any interrupt control register can be used) and ensure that always the same value is read, otherwise the CPU-CRC generation result differs.

Note 2: After the 1st CPU trigger input, the CPU-CRC operational result is not taken into the CRC data register. But on read access of the CCRCD register the actual CRC value is forwarded to the register and the actual CRC value is read.

Figure 5-2 CPU-CRC flow diagram

Note

1. During CPU-CRC generation, read access to all peripheral I/O registers including the registers accessible via external memory bus ($\overline{CS1}$, $\overline{CS5}$, $\overline{CS6}$ and $\overline{CS7}$) is prohibited.
2. Do not change the address position of the CPU-CRC program.
3. Interrupts are prohibited during CPU-CRC generating procedure including maskable and non maskable interrupts.
4. DMA transfers are prohibited during CPU-CRC generating procedure. Therefore all DMA transfers must be terminated beforehand.
5. The move instruction “mov r31, r0” must be executed before writing the CRCC register.
6. Read the interrupt control register and ensure that always the same value is read, otherwise the CPU-CRC generation result differs.
7. Channel 4 asynchronously monitors write data to the CPU's internal register file. Accordingly the operation result may differ with the test program's execution timing, even though the same program code was executed. Therefore please make sure that the execution timing of any selftest program using channel 4 is absolutely reproducible and does not vary e.g. due to interrupt occurrence. Please consider that if this signature is used to compute one final signature of several tests, the final signature will also differ if channel 4 was operated with a different timing.

5.4.2 DATA-CRC

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

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

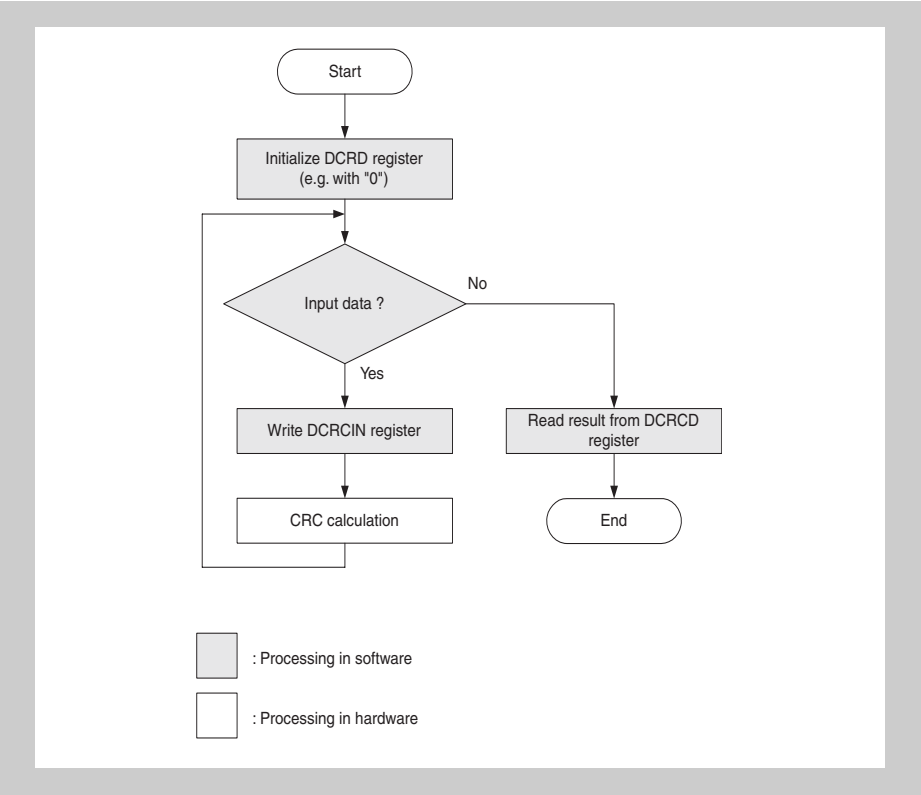


Figure 5-3 DATA-CRC flow diagram

Note 1. After the first data write to the input register DCRCIN, the DATA-CRC operational result is not taken into the CRC data register (DCRCD). But on read access of the DCRCD register the actual CRC value is forwarded to the register and the actual CRC value is read.

2. The CRC data register DCRCD must be initialized if the polynomial is changed by changing bit CRCC.POL.

Chapter 6 Clock Generator

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

6.1 Overview

The Clock Generator has two oscillators:

- The main oscillator generates the system clock from an external 16 MHz crystal or resonator.
- An internal oscillator generates a clock that is used to monitor the main oscillator. In case of a main oscillator malfunction detection, an interrupt or a reset can be generated, while the micocontroller is operating with the free-running PLL output clock.

6.1.1 Features

- Crystal frequency: f_X = 16 MHz
- Multiplier function using two phase locked loop (PLL) synthesizers:
  - A times 8 multiplier generates f_{XX} = 128 MHz, the high-speed internal system clock.
  - A times 5 multiplier generates f_{XX}' = 80 MHz for FlexRay. This clock can also be applied to the CPU.
- 6.8 MHz internal oscillator
- Built-in Clock Monitor
- Automatic change of CPU clock source if main oscillator fails

6.1.2 Description

The following figure shows a simplified block diagram of the Clock Generator:

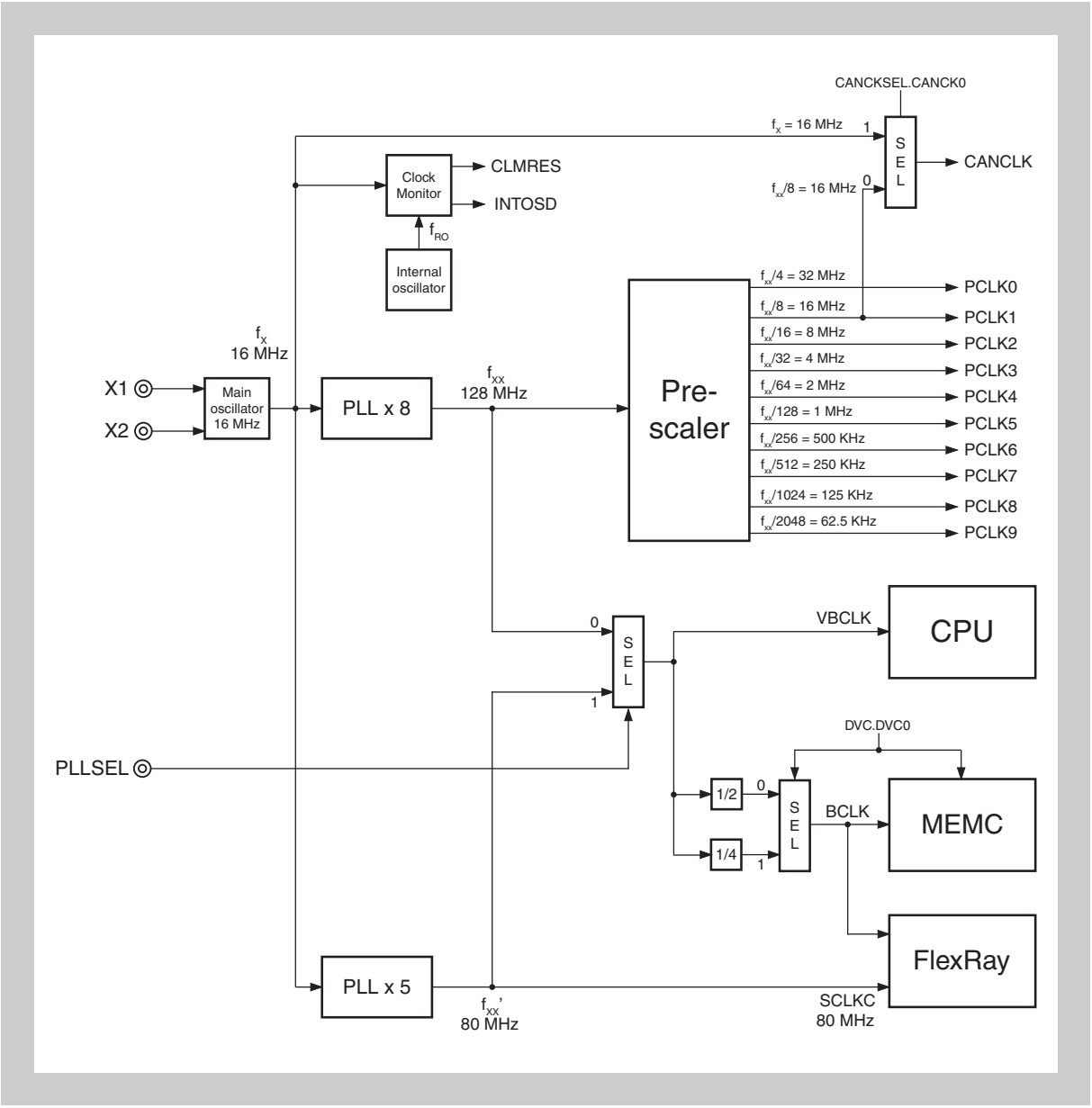


Figure 6-1 Clock Generator block diagram

Note f_X: Main oscillator output clock (16 MHz)

f_{XX}: Internal system clock (PLL × 8 output, 128 MHz)

f_{XX}': Internal system clock (PLL × 5 output, 80 MHz)

f_{RO}: Internal oscillator clock

VBCLK: Clock to CPU (128 MHz or 80 MHz)

BCLK: Clock to Memory Controller

SCLKC: Clock to FlexRay (= f_{XX}': PLL × 5 output, 80 MHz)

f_{XX} PLL An external resonator or crystal has to be connected to the X1 and X2 pins. Its frequency is multiplied by the PLL synthesizer. By this an internal system clock f_{XX} is generated that has 8 times the frequency (f_X) of the external resonator or crystal.

The clock controller enables the PLL automatically and starts clock supply to the system after oscillation stabilization time has passed.

f_{XX}' PLL An additional PLL, fed with the main oscillator clock f_X, with a multiplication factor of 5 generates the 80 MHz clock f_{XX}', which is used to supply the FlexRay module, also named SCLKC. f_{XX}' can also be used as the internal system clock instead of f_{XX}.

VBCLK The CPU clock VBCLK can be switched from 128 MHz to 80 MHz by a high level at pin PLLSEL (ground = 128 MHz, high level = 80 MHz). This is a preselection that must not be changed when the controller is active.

CANCLK The clock CANCLK for the CAN controllers can be chosen from the peripheral clock PCLK1 or the main oscillator clock f_x, both having a frequency of 16 MHz. The CANCLK choice is made by CANCKSEL.CANCK0.

PCLK The peripheral modules are supplied with a set of 10 peripheral clock PCLK0 to PCLK9, derived from the PLL output clock f_{xx} by a prescaler. The maximum frequency delivers PCLK0 with 32 MHz, the other PCLKn are the half of its predecessor PCLK(n-1).

6.2 Control Registers

The Clock Generator is controlled and operated by means of the following register:

Table 6-1 Clock Generator registers overview

Register name	Shortcut	Address
CAN clock selection register	CANCKSEL	FFFF F860$_H$

(1) CANCKSEL - CAN clock selection register

The CANCKSEL register determines the clock source of the CAN clock CANCLK.

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

Address FFFF F860$_H$

Initial Value 00$_H$. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CANCK0
R	R	R	R	R	R	R	R/W

Table 6-2 CANCKSEL register contents

Bit position	Bit name	Function
0	CANCK0	16 MHz CAN clock CANCLK supply selection 0: peripheral clock PCLK1 1: main oscillator clock f_x

6.3 Clock Monitor

The Clock Monitor monitors the operation of the main oscillator.

In case of malfunction of the main oscillator, it can generate the reset CLMRES or the interrupt INTOSD.

If the main oscillator has stopped to operate, the PLLs output their free-running clock frequency at f_{XX} respectively f_{XX}'. Thus the CPU system clock VBCLK and the base clock for all peripheral clocks PCLKn is derived from the free-running PLL output clocks.

Caution In case of a main oscillator stop, be aware of the following:

1. The CPU can operate on the free-running PLL output clock.
2. DMA transfers may fail to operate.
3. Peripheral modules, supplied by PCLKn, may fail to operate.
4. In on-chip debug mode the Clock Monitor is automatically stopped. Read/write of the Clock Monitor's registers is still possible.

Emergency shut-off The Clock Monitor interrupt INTOSD is capable to initiate an emergency shut-off (immediate change into high impedance mode) of certain output pins. For further information refer to *"Pin Functions" on page 31*.

6.3.1 Clock Monitor registers

The Clock Monitor is controlled and operated by means of the following register:

Table 6-3 Clock Monitor registers overview

Register name	Shortcut	Address
Clock Monitor mode register	CLM	FFFF FCA0$_H$

(1) CLM - Clock Monitor mode register

The 8-bit CLM register controls the operation of the Clock Monitor.

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

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

Address FFFF FCA0$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CLMD	CLME
R	R	R	R	R	R	R	R/W

Table 6-4 CLM register contents

Bit position	Bit name	Function
1	CLMD	Defines the action if the main oscillator frequency is too low: 0: Generate internal reset CLMRES. 1: Generate interrupt INTOSD.
0	CLME	Clock Monitor enable: 0: Disable the Clock Monitor. 1: Enable the Clock Monitor.

Note Once CLM.CLME has been set, it cannot be cleared by writing to the register. This bit can only be cleared by $\overline{\text{RESET}}$.

Caution CLMD must not be changed if the Clock Monitor is enabled (CLME = 1).

Since disabling of the Clock Monitor (CLME = 0) is only carried out by a $\overline{\text{RESET}}$, it is not possible to change CLMD after the Clock Monitor has been started.

6.3.2 Operation of the Clock Monitor

The Clock Monitor counts the falling edges of the main oscillator clock f_X during a period of 3 clock cycles of the half of the internal oscillator clock f_{RO} (= $1/T_{RO}$), i.e. within 3 x $2T_{RO}$.

In case the number of counted f_X is less than two, the Clock Monitor issues the reset CLMRES or the interrupt INTOSD, depending on CLM.CLMD.

As the frequency of the internal oscillator is heavily temperature depend, f_{RO} varies between its minimum f_{ROmin} (= $1/T_{ROmin}$) and maximum f_{ROmax} (= $1/T_{ROmax}$) values.

Main oscillator fail detection

Figure 6-2 illustrates the calculation of the minimum frequency f_{Xmin1}, that the Clock Monitor always detects as a main oscillator fail condition.

$$T_{Xmax1} = 3 \cdot 2 \cdot T_{ROmax}$$

and thus

$$f_{Xmin1} = f_{ROmin} / 6$$

All main oscillator frequencies below f_{Xmin1} are reported as main oscillator failure.

The main oscillator frequency f_X (= $1/T_X$) in *Figure 6-2* falls below this limit f_{Xmin1} and the Clock Monitor reports oscillator failure by generating CLMRES respectively INTOSD.

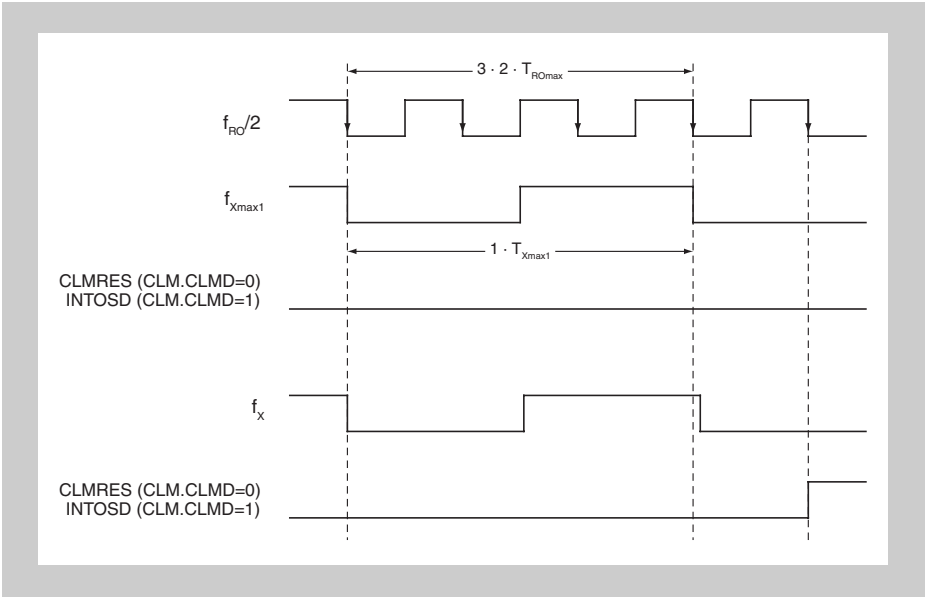


Figure 6-2 Main oscillator fail detection

Main oscillator no-fail detection

Figure 6-3 illustrates the calculation of the minimum frequency f_{Xmin2}, that the Clock Monitor always detects as a main oscillator no-fail condition.

$$2 \cdot T_{Xmax2} = 3 \cdot 2 \cdot T_{ROmin}$$

and thus

$f_{Xmin2} = f_{ROmax} / 3$

All main oscillator frequencies above f_{Xmin2} are judged as main oscillator no-fail.

The main oscillator frequency f_X (= $1/T_X$) in *Figure 6-3* falls below this limit f_{Xmin2} and the Clock Monitor reports oscillator failure by generating CLMRES respectively INTOSD.

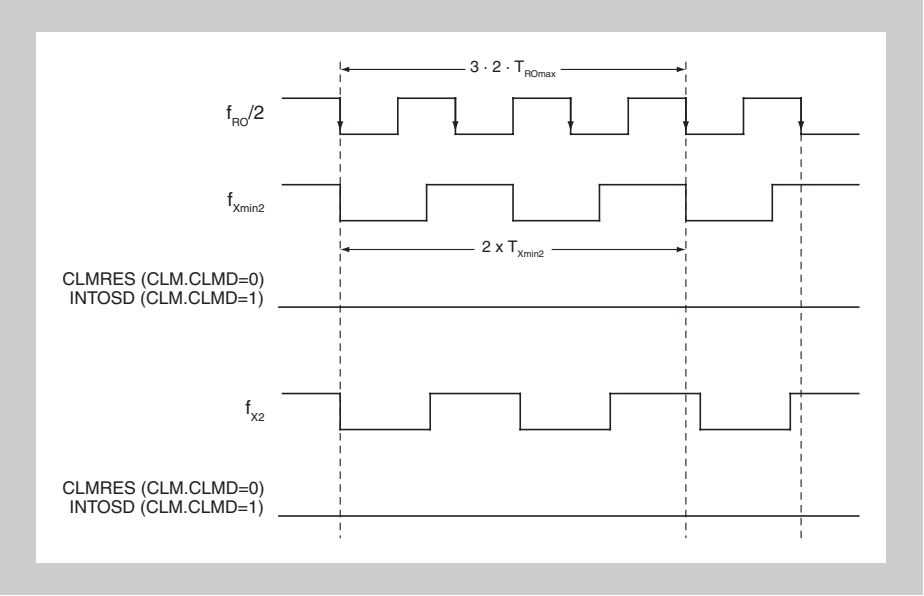


Figure 6-3 Main oscillator “no fail” detection

All main oscillator frequencies between f_{Xmin1} and f_{Xmin2} may be detected as oscillator failure.

Table 6-5 summarizes the main oscillator fail/no-fail detections and gives an example with following internal oscillator frequencies:

- f_{ROmin} = 5,88 MHz $\Rightarrow$ T_{ROmax} = 170 ns
- f_{ROmax} = 8,00 MHz $\Rightarrow$ T_{ROmin} = 125 ns

For information concerning the real internal oscillator frequencies refer to the Data Sheet.

Table 6-5 Main oscillator fail detection

f_X	Clock Monitor detection	Example
$f_X < f_{Xmin1} = f_{ROmin / 6}$	fail	f_X < 0,98 MHz
$f_{Xmin1} < f_X < f_{Xmin2}$	fail or no-fail	0,98 MHz < f_X < 2,67 MHz
$f_X > f_{Xmin2} = f_{ROmax / 3}$	no-fail	f_X > 2,67 MHz

Note Main oscillator frequencies above its nominal value f_X will not be detected as failure.

6.4 Power Save Control

The power save function of V850E/PHO3 supports the HALT mode. In this mode, the Clock Generator (oscillator and PLL synthesizers) continues to operate, but the CPU clock and hence program execution stops. The built-in RAM keeps the state it had before the HALT mode was set.

The clock supply to the on-chip peripherals continues. Peripheral devices that do not require instruction processing remain operating.

The HALT mode can reduce the average power consumption of the entire system by intermittent operation with the normal operation mode.

The system is switched to HALT mode by a specific instruction (the HALT instruction).

Figure 6-4 shows the transitions between normal operation mode and HALT mode.

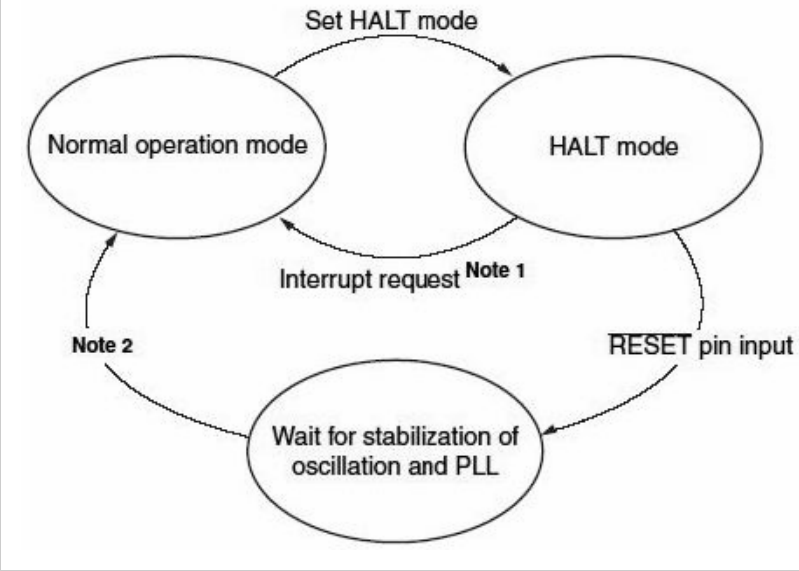


Figure 6-4 Power Save Mode State Transition Diagram

Note 1. Non-maskable interrupt request signal (NMI) or unmasked maskable interrupt request signal.

2. The oscillation stabilization time is necessary after release of reset because the PLL is initialized by a reset. The stabilization time is determined by hardware.

6.4.1 HALT mode

(1) Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

When HALT mode is set, clock supply is stopped to the CPU only. The Clock Generator and PLL continue operating. Clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 6-6 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Caution

1. Insert five or more NOP instructions after the HALT instruction.
2. If the HALT instruction is executed while an interrupt request is being held pending, the HALT mode is set but is released immediately by the pending interrupt request.

The following table shows the operation status in the HALT mode.

Table 6-6 Operation Status in HALT Mode

Function	Operation Status
Clock Generator	Operating
Internal system clock (f_{XX})	Supplied
CPU	Stopped
DMA	Operating
Interrupt controller	Operating
Ports	Maintained
On-chip peripheral I/O (excluding ports)	Operating
Internal data	All internal data such as CPU registers, states, data, and the contents of internal RAM are retained in the state they were before HALT mode was set.
A0 to A21	Holds last status
D0 to D31	Hi-Z
$\overline{RD}$	H
$\overline{WR}$	H
$\overline{BEN0}$ to $\overline{BEN3}$	H
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$	H
$\overline{WAIT}$	Invalid

(2) Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI), an unmasked maskable interrupt request signal, or any reset.

After the HALT mode has been released, the normal operation mode is restored.

(a) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

- If an interrupt request signal with a priority lower than or same as the interrupt currently being serviced is generated, the HALT mode is released, but the newly generated interrupt request signal is not acknowledged. The interrupt request signal itself is retained.
- If an interrupt request signal with a priority higher than that of the interrupt currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged.

Table 6-7 Operation after releasing HALT mode by interrupt request signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address	
Unmasked maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(b) Releasing HALT mode by reset

The same operation as the normal reset operation is performed.

Chapter 7 Flash Memory

The V850E/PHO3 microcontroller is equipped with following internal flash memory:

- code flash:
  - μPD70F3483: 768 KB
  - μPD70F3441: 992 KB
- data flash: 32 KB

The code flash memory is attached to the dedicated fetch bus interface of the V850 CPU core. It is used for non-volatile storage of program code and constant data.

The data flash memory is accessible via the memory interface bus. It holds nonvolatile user's data, which are subject to be altered during normal program operation.

Flash memory is commonly used in the following development environments and applications:

- For altering software after solder-mounting of the V850 microcontroller on the target system.
- For differentiating software in small-scale production of various models.
- For data adjustment when starting mass production.
- For facilitating inventory management.
- For updating software after shipment.

The flash memory can be written in different ways:

- by a flash programmer equipped with a suitable adapter (off-board write)
- mounted on the target board by connecting a dedicated flash programmer to the target system (on-board write)
- by the V850 microcontroller's application software (self-programming)

7.1 Code Flash Memory Overview

7.1.1 Code flash memory features

- Internal code flash memory:
  - μPD70F3483: 768 KB
  - μPD70F3441: 992 KB
- Operation speed: up to 128 MHz by 2-way interleaved access
  - 4-byte/1 CPU clock cycle access for consecutive instruction fetches
  - 4-byte/5 CPU clock cycles access for random instruction and data fetches
- All-blocks batch erase or single block erase
- Erase/write with single power supply
- Communication with dedicated flash programmer via two serial interfaces
- On-board and off-board programming
- Flash memory programming by self-programming
- 7-bit ECC (error correction code) for every 4 bytes
  - 1 bit error: automatic correction
  - 2 bit errors: detection
  - Specific interrupt generation in case of bit errors

7.1.2 Code flash memory mapping

The V850 microcontroller's internal code flash memory area is divided into blocks of 4 KB blocks and can be programmed/erased in block units. All or some of the blocks can also be erased at once.

Following figures list the block structures and address assignments of the code flash memory.

Additional information comprise:

- Boot swap cluster size
  Configurable size of boot cluster for secure self-programming, refer to *"Secure self-programming (boot cluster swapping)" on page 215*.
- Interleave
  Interleave configuration of the flash memory blocks.
- CPU branch latency
  Number of additional CPU clock cycles during instruction fetches of non-linear code.

	Block 247 (4 KB)	000F 7FFFH	Address
		000F 7000H	
	...		
		...	
	Block 192 (4 KB)	000C 0FFFH	
		000C 0000H	
Block 191 (4 KB)	Block 191 (4 KB)	000B FFFFH	
		000B F000H	
...	...		
		...	
Block 128 (4 KB)	Block 128 (4 KB)	0008 0FFFH	
		0008 0000H	
Block 127 (4 KB)	Block 127 (4 KB)	0007 FFFFH	
		0007 F000H	
...	...		
		...	
Block 96 (4 KB)	Block 96 (4 KB)	0006 0FFFH	
		0006 0000H	
Block 95 (4 KB)	Block 95 (4 KB)	0005 FFFFH	
		0005 F000H	
...	...		
		...	
Block 1 (4 KB)	Block 1 (4 KB)	0000 1FFFH	
		0000 1000H	
Block 0 (4 KB)	Block 0 (4 KB)	0000 0FFFH	
		0000 0000H	
768 KB	992 KB	Code flash size	
64 KB		Boot swap cluster sizes	
2-way		Interleave	
4 cycles		CPU branch latency	
µPD70F3483	**µPD70F3441**	Products	

Figure 7-1 Code flash memory configuration

7.1.3 Code flash memory functional outline

Serial programming

The internal flash memory of the V850 microcontroller can be rewritten by using the rewrite function of a dedicated flash programmer, regardless of whether the V850 microcontroller has already been mounted on the target system or the device is not mounted (off-board/on-board programming).

Since there is no functional difference between on-board and off-board programming by an external flash programmer, both will be gathered as “serial programming” - in contrast ”to self-programming”.

Self-programming

The self-programming facility, which facilitates rewriting of the flash memory by the user program, is ideal for program updates after production and shipment, since no additional programming equipment is required. During self-programming some software services as well as interrupt serving can still be in operation, e.g to sustain communication with other devices.

While the self-programming mode can be initiated from the normal operation mode the external flash programmer mode is entered immediately after release of a system reset.
Refer to *“Flash memory programming control” on page 207* for details on how to enter normal operation or serial flash programming mode.

Extra area

The flash memory contains an extra area, used to store the settings of security and protection functions and other flash relevant information.
The extra area is not mapped into the CPU’s address space, thus is not directly accessible by the user’s program. The extra area’s settings can only be read and modified by an external programmer or by the self-programming library.

Boot swap

A boot swap function makes safe re-programming of the flash memory possible and is used to maintain an operable software version, even if re-programming fails for any reason, e.g. in a power fail situation.
For further information concerning boot swapping refer to *“Secure self-programming (boot cluster swapping)” on page 215*.

Protection

A set of protection flags can be specified during flash memory programming to prohibit access the flash memory in different ways, implying read-out, rewrite and erase protections. By these means the code flash memory can be protected against read-out and rewrite of the flash memory content by unauthorized persons.
For further information concerning data protection refer to *“Data Protection and Security” on page 219*.

Table 7-1 Flash memory write methods

Environment	Interface	Outline	Operation Mode
Serial programming	Serial I/F (UART, CSI)	Flash memory programming is done by an external flash programmer. The device may be mounted on the target system (on-board) or unmounted (off-board) by using a suitable programming adapter board. In either case the communication between the device and the flash progammer is using a serial interface. For details refer to *"Flash Programming with Flash Programmer" on page 202*.	Flash memory programming mode
Self-programming	Self-programming library	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of off-board/on-board programming. The self-programming library provides all necessary functions to be called by the user's software. For details refer to *"Code Flash Self-Programming" on page 213*.	Normal operation mode

Table 7-2 on page 196 summarizes the functions used to modify flash memory content.

Table 7-2 Basic functions for flash memory modifications

Function	Functional outline	Support (√: Supported, ×: Not supported)	
		Serial programming	Self-programming
Block erasure	The contents of specified memory blocks are erased.	√	√
Multiple block erasure	The contents of the specified successive multiple blocks are erased.	√	√
Chip erasure	The contents of the entire memory area is erased all at once. The extra area - except the boot block cluster protection flag - is also erased. **Caution:** The chip erase function erases also the data flash memory.	√	x [a]
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	√	√
Verify	Data read from the flash memory is compared with data transferred from the flash programmer.	√	x [b]
Checksum	Microcontroller internally calculated checksum over the entire flash memory content is compared with the checksum calculated by the external programmer	√	x
Blank check	The erasure status of the entire memory is checked.	√	√
Protection settings	Following functions can be prohibited: • chip erase • block erase • write • read • rewriting of the boot block cluster	√	√ [c]

a) In self-programming mode all blocks can be specified to be erased at once by block erasure. Note that the extra area is not erased in this case.

b) Can be carried out by the user's program.

c) Except protection against rewriting of the boot block cluster all other protections have no effect in self-programming mode.
Protection settings can be activated in self-programming mode. Already activate protection settings can not be deactivated.

The following table lists the available flash memory protection functions.

For details refer to *"Data Protection and Security" on page 219*.

Table 7-3 Protection functions

Function	Functional outline	Applicable (√: applies, ×: doesn't apply)	
		Serial programming	Self-programming
Chip erase command prohibit	Erasure of the entire flash (including the extra area[a] and the data flash) or single blocks impossible.	√	×
Block erase command prohibit	Erasure of single blocks impossible.	√	×
Program command prohibit	Erasure and rewrite of single blocks impossible.	√	×
Read command prohibit	Read-out of any flash content impossible.	√	×
Rewriting boot area prohibit	Erasure (by block or chip erase) or writing of the boot block cluster impossible.	√	√

a) The boot block cluster protection flag is not erased.

7.1.4 Code flash memory erasure and rewrite

Erasure According to its block structure the flash memory can be erased in two different modes.

- All-blocks batch erasure
  All blocks are erased all together.
- Block erasure
  Each 4 KB flash memory block can be erased separately.
  In self-programming mode any number of contiguous flash memory blocks can be erased all together.

Rewrite In self-programming and serial programming mode it is possible to rewrite the flash memory in smaller units than one block. Once a complete block has been erased it can be rewritten in units of 8 byte, starting at 8 byte aligned addresses, i.e. at 000_B as the lower 3 address bits. Each unit can be rewritten only once after erasure of the complete block.

7.1.5 Code flash memory error detection

The code flash memory control circuit is equipped with an error detection function. A 7-bit ECC (Error Correction Coding) code is dedicated to each 32-bit word in the flash memory.

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

INTDEDF can generate two kinds of interrupts:

- an NMI via the NMI sharing function
- a maskable interrupt (exception code 0080_H), that is shared with double error detections of the RAM and FlexRay error detection functions

Besides generating an non-maskable or maskable interrupt, the address of the erroneous data is saved a register.

Note As error detection is also performed on instructions, which are prefetched by the CPU's execution pipeline, an error may be detected and reported, even if the instruction is not attempted to be executed because of a program branch.

One register is provided for the error correction function.

(1) ROMEAD - Flash ROM ECC error register

The 32-bit ROMEAD register holds the address of the flash memory location, where an error was detected first.

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

Address FFFF F8A0$_H$

Initial Value 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	ROMEADDR			

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

Table 7-4 ROMEAD register contents

Bit Position	Bit Name	Function
19 - 2	ROMEADDR	20-bit address of first detection of a flash ROM error.

The address of the first error detection is stored in ROMEAD. At the same time the interrupt flag INTERRF.INTERR0 is set to 1.

This register is not overwritten by a new error detection until

- the INTERRF.INTERR0 is cleared to 0. Thus no new error is signaled by interrupt INTDEDF until the INTERRF.INTERR0 is cleared to 0.
- ROMEAD is cleared to 0000 0000$_H$ by any reset.

Afterwards a new ECC detection address can be stored.

(2) Flash memory error detection check

For checking the functionality of the flash memory error detection, a dedicated address in the flash memory needs to be reserved for generating a mismatch between the flash memory data word an its associated ECC.

This flash memory address needs to be programmed twice:

1. The first data written to the address generates the correct ECC
2. The second data written to the same address must differ to the first data. Consider that during re-programming only “1” bits can be changed to “0”, but not vice versa. Further the correct ECC of the second data must differ to the correct ECC of the first data.

Afterwards any read of the data (either as an instruction fetch or as a read of a constant) should generate a maskable or non-maskable flash memory error interrupt INTDEDF.

7.2 Data Flash Memory

The V850E/PHO3 microcontroller contains a 32 KB data flash in addition to the code flash. The data flash is on-chip connected to the external memory bus.

7.2.1 Data flash memory features

The data flash has the following features:

- 32 KB of data flash memory in 2 KB blocks
- Write access in 32-bit steps
- Erase in 2 KB blocks
- Write, erase operations to the data flash while user's code can be executed from code flash

Note NEC provides sample code for a library enabling the flexible emulation of an EEPROM. By means of this library the emulated number of write/erase cycles can be extended far beyond the number of maximum write/erase cycles specified in the Data Sheet.

7.2.2 Data flash memory map

The data flash is connected to the memory bus, but is built into the V850 microcontroller.

The data flash is mapped to the $\overline{CS1}$ area to address range 3E0 0000_H to 3E0 $7FFF_H$.

The memory interface has to be set up as follows:

Table 7-5 BCU/MEMC register settings for data flash access

Control bit	Required setting	Comment
CSC0.CS1[3:0]	0001_B	• area 0 assigned to $\overline{CS1}$ • default, don't change
BSC.BSC1[1:0]	10_B	• 32 bit bus width • default, don't change
BEC.BE10	0	• little endian • default, don't change
BCT0.ME1	1	• enable $\overline{CS1}$ • no default, must be changed
AWC.AHW1 AWC.ASW1	00_B	• no address setup/hold waits • no default, must be changed
DWC0.DWC1[2:0]	001_B	• 1 data wait state • no default, must be changed
BCC.BC11	0	• no idle states • no default, must be changed

For further information about the memory interface configuration refer to

"Bus and Memory Control (BCU, MEMC)" on page 225.

7.2.3 Data flash control register

(1) DFLCTL - Data flash control register

The data flash is controlled with the data flash control register DFLCTL to enable the access to the data flash and to define the memory address location.

Writing to this register is protected by a special sequence of instructions. Please refer to *"Write Protected Registers" on page 123*.

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

Address FFFF FCF8$_H$

Initial value 00$_H$. This register is cleared by any reset.

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

Table 7-6 DFLCTL register contents

Bit position	Bit name	Function
7	DFLEN	Read access control of data flash: 0: Disable 1: Enable

7.2.4 Data flash reading

The data flash can be read via the external memory bus.

Reading of the data flash is performed with the following procedure:

1. Enable the read access to the data flash by setting the DFLCTL.DFLEN = 1.
2. Execute read operation

7.2.5 Data flash writing

The data flash can be written by using the data flash library or serial programming with an external flash programmer tool.

Programming during normal operation is achieved by using the data flash access layer software library. The data flash access layer is described in a separate User's Manual.

Note The chip erase command of an external programmer erases also the data flash.

7.3 Flash Programming with Flash Programmer

A dedicated flash programmer can be used for external writing of the flash memory.

- On-board programming
  The contents of the flash memory can be rewritten with the V850 microcontroller mounted on the target system. Mount a connector that connects the flash programmer on the target system.
- Off-board programming
  The flash memory of the V850 microcontroller can be written before the device is mounted on the target system, by using a dedicated programming adapter.

7.3.1 Programming environment

The necessary environment to write a program to the flash memory of the V850 microcontroller is shown below.

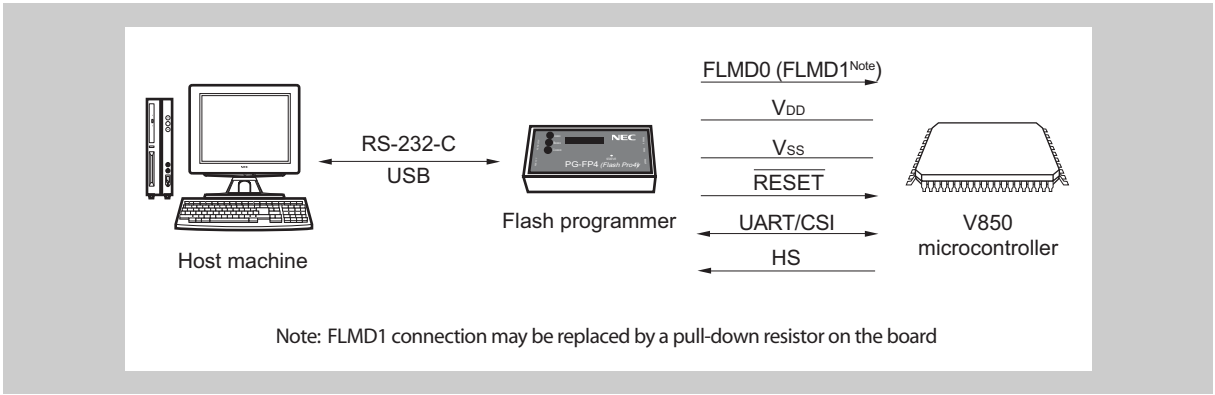


Figure 7-2 Environment to write program to flash memory

A host machine is required for controlling the flash programmer.

Following V850 microcontroller serial interfaces can be used as the interface between the flash programmer and the V850 microcontroller:

- asynchroneous serial interface UART
- clocked serial interface CSI

If a CSI interface is used with handshake, the flash programmer's HS signal is connected to a certain V850 port, in the following generally named as HSPORT. The port used as HSPORT for this product is given in *Table 7-8*.

Flash memory programming off-board requires a dedicated programming adapter.

In this chapter the terms UART and CSI may be used generically for the dedicated interface types and channels the V850 microcontroller provides. UART and CSI signal names are used accordingly.

7.3.2 Communication mode

The communication between the flash programmer and the V850 microcontroller utilizes the asynchronous serial interface UART or optionally the synchronous serial interface CSI.

For programming via the synchronous serial interface CSI without handshake and with handshake modes are supported. In the latter mode the port pin HSPORT is used for the programmer's handshake signal HS.

(1) UART

The external flash programmer offers various choices of available baud rates.

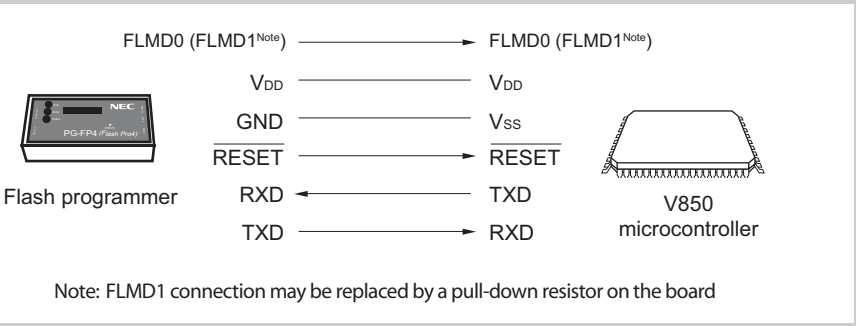


Figure 7-3 Communication with flash programmer via UART

(2) CSI without handshake

The external flash programmer offers various choices of available clock rates.

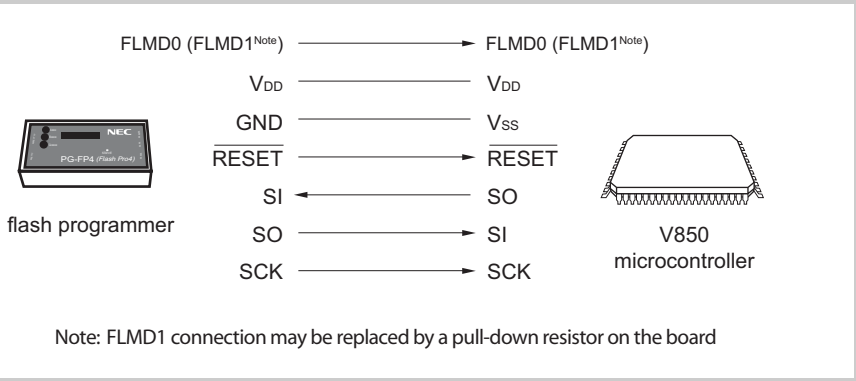


Figure 7-4 Communication with flash programmer via CSI without handshake

The flash programmer outputs a transfer clock and the V850 microcontroller operates as a slave.

(3) CSI with handshake (CSI + HS)

The external flash programmer offers various choices of available clock rates.

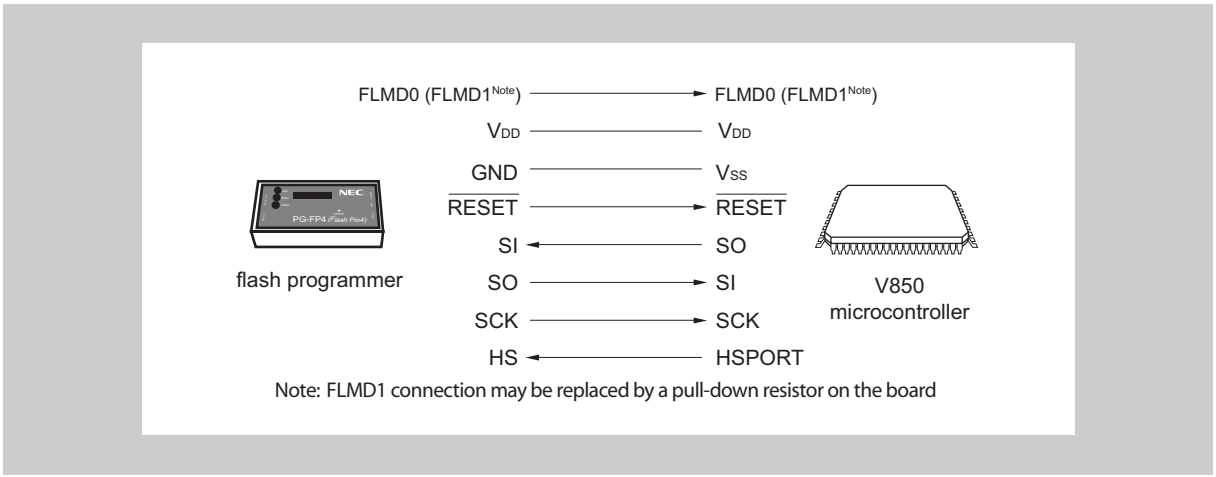


Figure 7-5 Communication with flash programmer via CSI with handshake

The flash programmer outputs a transfer clock and the V850 microcontroller operates as a slave.

HSPORT The microcontroller port used for the handshake signal HS is specified in *Table 7-8 on page 206*.

7.3.3 Pin connection with flash programmer PG-FP4

A connector must be mounted on the target system to connect the flash programmer for on-board writing. In addition, functions to switch between the normal operation mode and flash memory programming mode and to control the V850 microcontroller's reset pin must be provided on the board.

When the flash memory programming mode is set, all the pins not used for flash memory programming are in the same status as immediately after reset.

If the PG-FP4 is used as the flash programmer, it generates the signals listed in *Table 7-7* for the V850 microcontroller. For details, refer to the PG-FP4 User's Manual (U15260E).

Table 7-7 Signals generated by flash programmer PG-FP4

PG-FP4			Controller	Connection		
Signal name	**I/O**	**Pin function**	**Pin name**	**UART**	**CSI**	**CSI + HS**
FLMD0	Output	Write enable/disable	FLMD0	√	√	√
FLMD1	Output	Write enable/disable	FLMD1	×	×	×
V_{DD}	I/O	V_{DD} voltage generation/voltage monitor	V_{DD}	√	√	√
GND	–	Ground	V_{SS}	√	√	√
CLK	Output	Clock output to the controller	X1	×	×	×
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	√	√	√
SI/RXD	Input	Receive signal	SO/TXD	√	√	√
SO/TXD	Output	Transmit signal	SI/RXD	√	√	√
SCK	Output	Transfer clock	SCK	×	√	√
HS	Input	Handshake signal for CSI + HS communication	HSPORT[a]	×	×	√

a) The microcontroller port used for the handshake signal HS is specified in *Table 7-8 on page 206*.

Note √: Must be connected.
×: Does not have to be connected.

Table 7-8 Wiring of V850E/PHO3 flash writing adapters for CSIB

Flash programmer (FG-FP4) connection pin			Name of FA board pin	UARTC0	CSIB0 + HS	CSIB0
Signal name	I/O	Pin function		Pin name	Pin name	Pin name
SI/RxD	I	Receive signal	SI	TXDC0	SOB0	
SO/TxD	O	Transmit signal	SO	RXDC0	SIB0	
SCK	O	Transfer clock	SCK	Not needed	SCKB0	
CLK	O	Clock to V850 microcontroller	X1	Leave open		
			X2	Leave open		
$\overline{\text{RESET}}$	O	Reset signal	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$		
FLMD0	I	Write voltage	FLMD0	FLMD0		
FLMD1	I	Write voltage	FLMD1	FLMD1		
HS	I	Handshake signal for CSI + HS	RESERVE/ HS	Not needed	HSPORT = PDH0	Not needed
VDD	–	VDD voltage generation/ voltage monitor	VDD	V_{DD}		
				V_{DD1}		
				BV_{DD}		
				EV_{DD}		
				AV_{REF0}		
				AV_{REF1}		
GND	–	Ground	GND	V_{SS}		
				V_{SS1}		
				BV_{SS}		
				EV_{SS}		
				AV_{SS}		
				AV_{SS1}		

Table 7-9 V850E/PHO3 pin numbers for serial programming

Pin name	Port	V850E/PHO3 pin nr.
TXDC0	P31	AD-20
RXDC0	P30	AC-20
SIB0	P40	M-23
SOB0	P41	M-24
SCKB0	P42	L-23
$\overline{\text{RESET}}$	–	N-22
FLMD0	–	G-22
FLMD1	–	H-21
PDH0	PDH0	U-4

7.3.4 Flash memory programming control

The procedure to program the flash memory is illustrated below.

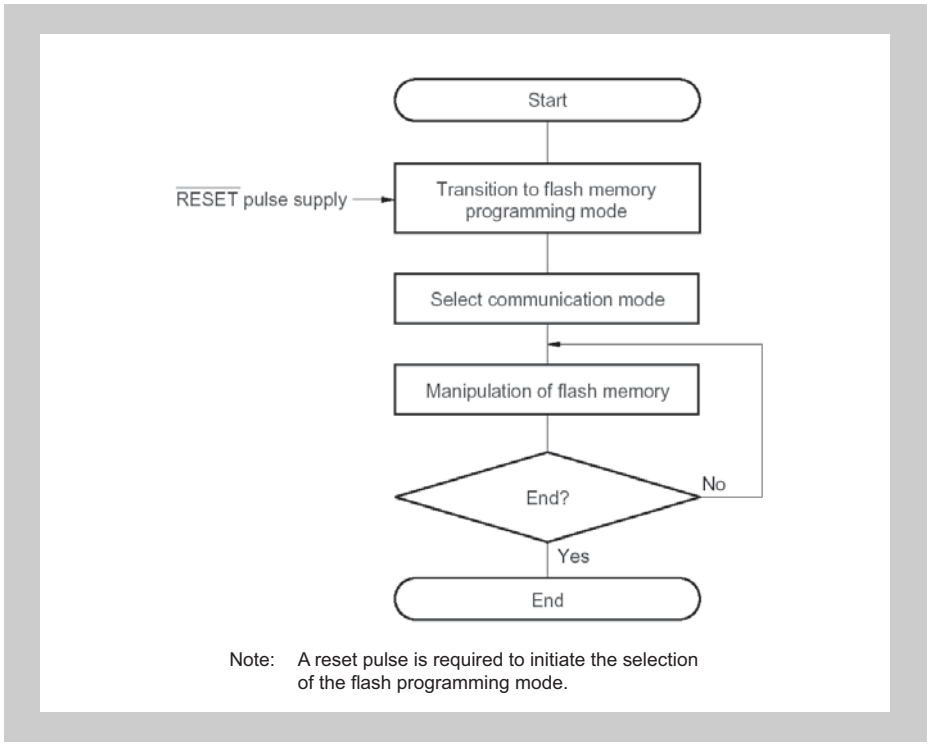


Figure 7-6 Flash memory programming procedure

(1) Operation mode control

To rewrite the contents of the flash memory by using the flash programmer, set the V850 microcontroller in the flash memory programming mode.

To set this mode, set the FLMD0 and FLMD1 pins as shown in *Table 7-10 on page 208* and release $\overline{\text{RESET}}$.

In the normal operation mode, 0 V is input to the FLMD0 pin. A pull-down resistor at the FLMD0 pin ensures normal operation mode if no flash programmer is connected. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. Additionally the FLMD1 pin has to hold 0 V level.

Table 7-10 Operation mode setting

Pins		Operation mode
FLMD0	**FLMD1**	
V_{SS}	X	Normal operation mode
V_{DD}	V_{SS}	Flash programming mode
	V_{DD}	Setting prohibited

An example of connection of the FLMD0 and FLMD1 pins is shown below. FLMD1 can be connected to ground via a resistor. Alternatively the FLMD1 pin may also be connected directly to the FLMD1 signal of the flash programmer.

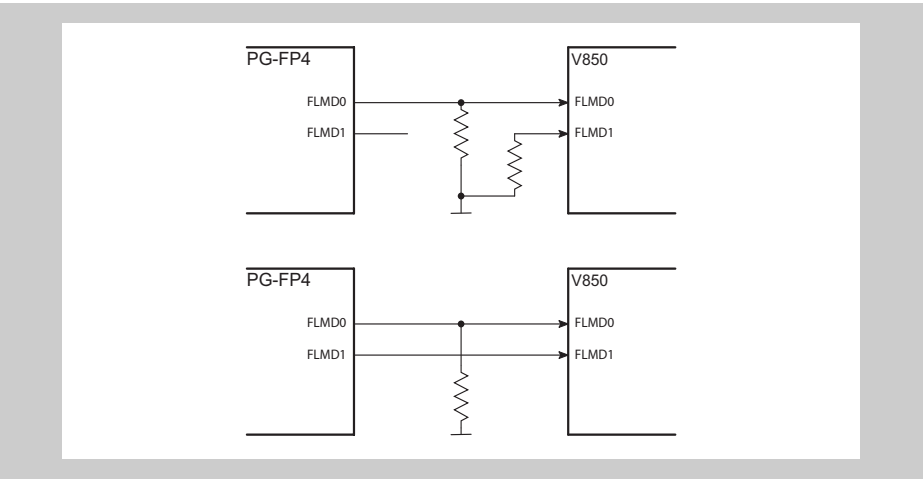


Figure 7-7 Example of connection to flash programmer PG-FP4

Once started in normal operation mode (FLMD0 = 0), FLMD0 pin is used for enabling self-programming. Refer also to *"Code Flash Self-Programming" on page 213*.

(2) Potential conflicts with on-board signal connections

Serial I/O signals

If other devices are connected to the serial interface pins in use for flash memory programming in on-board programming mode take care that the concerned signals do not conflict with the signals of the flash programmer and the V850 microcontroller. Output pins of the other devices must be isolated or set in high impedance state. Ensure that the other devices do not malfunction because of flash programmer signals.

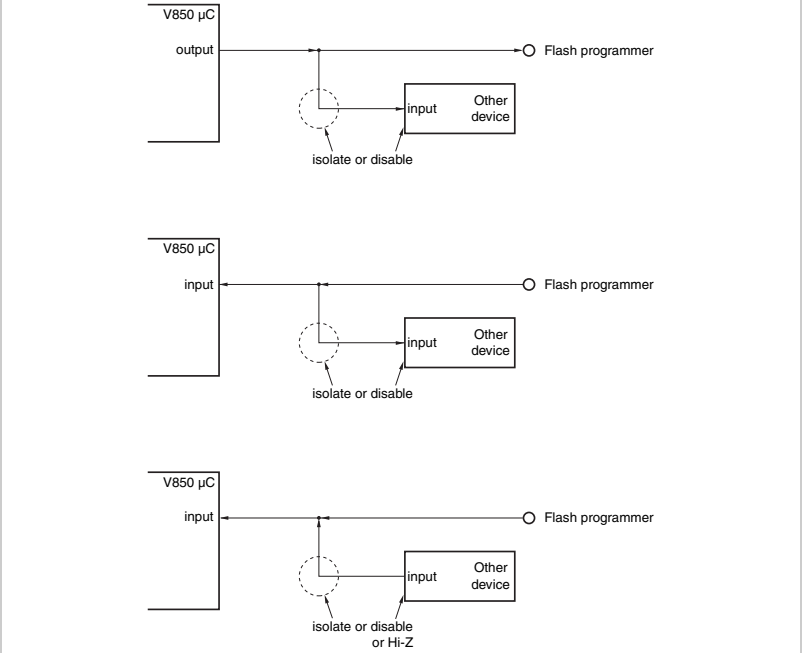


Figure 7-8 Potential conflicts with serial interfaces signals

$\overline{\text{RESET}}$

Pay attention in particular if the flash programmer's $\overline{\text{RESET}}$ signal is connected also to an on-board reset generation circuit. The reset output of the reset generator may ruin the flash programming process and may need to be isolated or disabled.

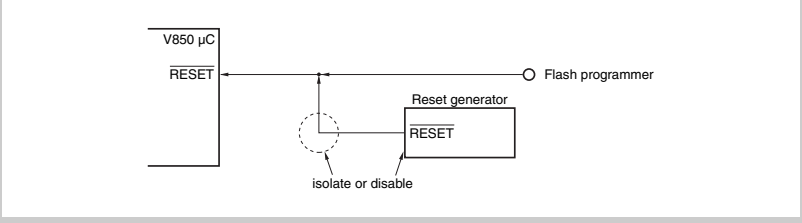


Figure 7-9 Potential conflict with $\overline{\text{RESET}}$

Ports The V850 port pins adopts following status during serial programming:

Ports used for programming are configured as UART respectively CSI pins.

All other pins remain in their default state after reset release.

In case the default state after reset of the pins not used for programming is inport port or high -impedance output port, pay attention to other devices connected to these pins. If these devices require defined levels at the pins, the ports may have to be connected to V_{DD} or V_{SS} via a resistors.

Oscillators Connect all oscillator pins in the same way as in the normal operation mode.

$\overline{\text{DRST}}$ During flash memory programming, input a low level to $\overline{\text{DRST}}$ or leave it open. Do not input a high level.

Power supply Supply the same power to all power supply pins, including reference voltages, power regulator pins, etc., as in the normal operation mode.

(3) Selection of the communication mode

The communication interface is chosen by applying a specified number of pulses to the FLMD0 pin after reset release. Note that this is handled by the flash programmer.

Figure 7-10 on page 211 gives an example how the UART is established for the communication between the flash programmer and the V850 microcontroller.

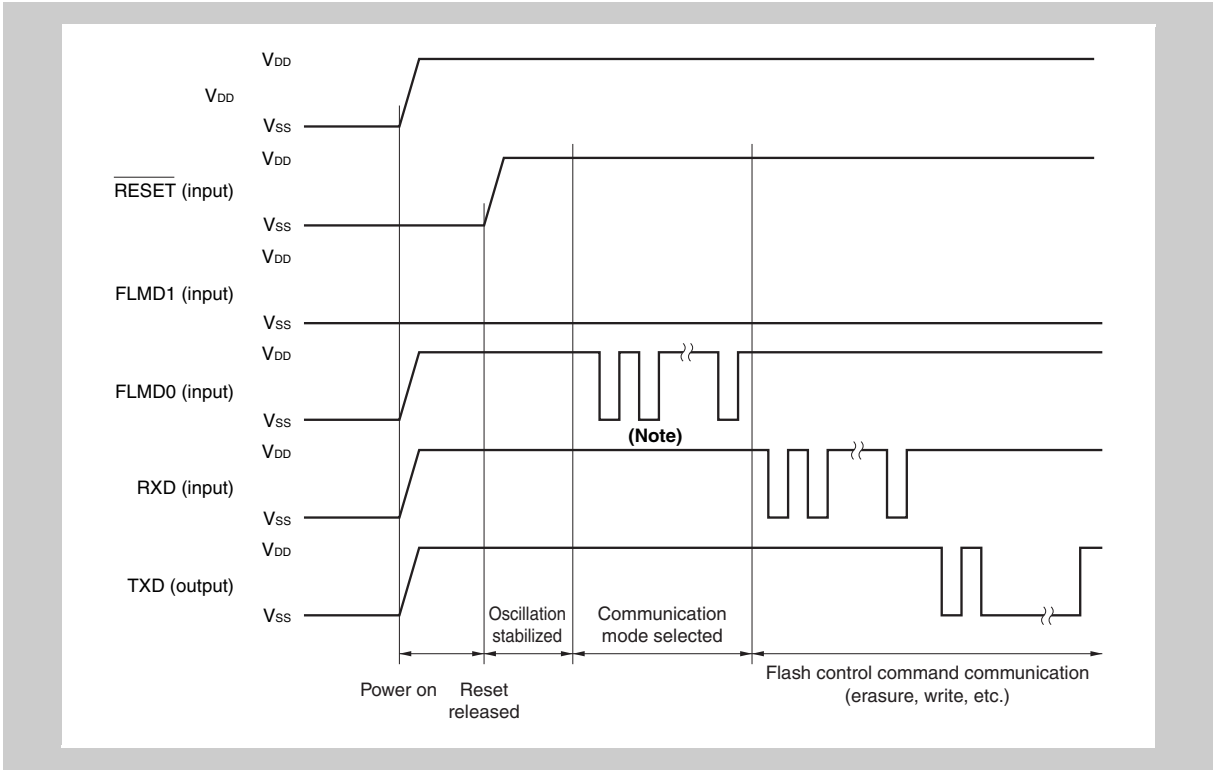


Figure 7-10 Selection of communication mode

Note The number of clocks to be inserted differs depending on the chosen communication mode. For details, refer to *Table 7-11 on page 211*.

Table 7-11 FLMD0 pulses for communication mode setting

FLMD0 pulses	Communication Mode	Remarks
0	UART	Communication rate: 9600 bps (after reset), LSB first
8	CSI	V850E/PHO3 performs slave operation, MSB first
11	CSI + HS	V850E/PHO3 performs slave operation, MSB first
Other	–	Setting prohibited

When UART has been selected after reception of the FLMD0 pulses with 9600 bps, the flash programmer changes the baud rate according to the user's choice via the flash programmer's user interface.

At first the programmer sends two 00_H bytes, which are used by the microcontoller to measure the baud rate and to set up it's own baud rate accordingly.

(4) Communication commands

The flash programmer sends commands to the V850 microcontroller. Depending on the commands, the V850 microcontroller returns status information or the requested data.

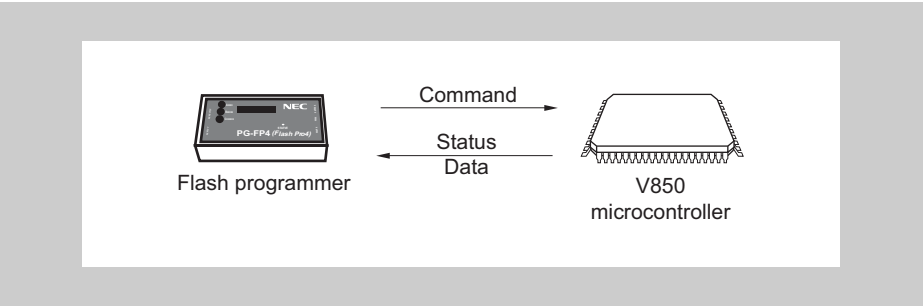


Figure 7-11 Communication commands exchange

The following table lists the flash memory control commands of the V850 microcontroller. All these commands are issued by the flash programmer, and the V850 microcontroller performs the corresponding processing.

Table 7-12 Flash memory control commands

Classification	Command name	Support			Function
		CSIB	CSIB + HS	UARTA	
Blank check	Block blank check command	√	√	√	Checks erasure status of entire memory.
Erase	Chip erase command	√	√	√	Erases all memory contents.
	Block erase command	√	√	√	Erases memory contents of specified block.
Write	Write command	√	√	√	Writes data by specifying write address and number of bytes to be written, and executes verify check.
Verify	Verify command	√	√	√	Compares input data with all memory contents.
System setting and control	Reset command	√	√	√	Escapes from each status.
	Oscillation frequency setting command	√	√	√	Sets oscillation frequency.
	Baud rate setting command	–	–	√	Sets baud rate when UART is used.
	Silicon signature command	√	√	√	Reads silicon signature information.
	Version acquisition command	√	√	√	Reads version information of device.
	Status command	√	√	–	Acquires operation status.
	Protection setting command	√	√	√	Sets protection against chip erasure, block erasure, and writing.

7.4 Code Flash Self-Programming

This V850 microcontroller supports a flash macro service that allows the user program to rewrite the internal flash memory by itself.

By using this flash macro service and a self-programming library, provided by NEC, the user's program is able to rewrite the flash memory with data, transferred in advance to the internal RAM or the external memory.

Thus the user program can be upgraded and constant data can be rewritten in the field.

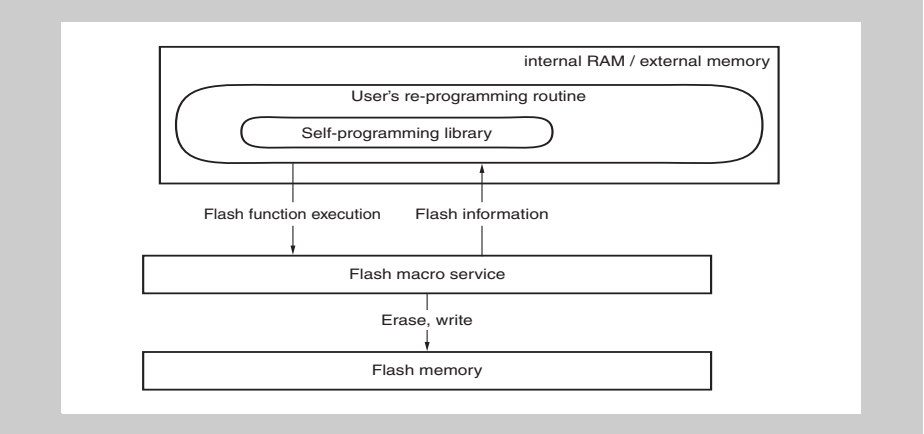


Figure 7-12 Concept of self-programming

During self-programming access to the flash memory is not possible. Thus program execution is only possible by instruction fetching from internal RAM or external memory.

Consequently the instructions of user re-programming software routines, which shall remain in operation during the self-programming procedure, must be copied from the flash memory to the internal RAM or external memory prior to activating the self-programming. Since interrupt processing by using the interrupt vectors in the flash memory is also impossible during self-programming, a special feature is provided to re-route interrupt acknowledges to the internal RAM (refer to *"Interrupt handling during flash self-programming" on page 218*).

It is recommended to refer to the Application Note "Self-Programming" (document nr. U16929EE) for comprehensive information concerning flash self-programming. This document explains also the functions of the self-programming library. The latest version of this document and the library can be loaded via the URL

http://www.eu.necel.com

Caution The self-programming operation employs also the internal Firmware, which makes use of 9 KB of the internal RAM in the address range FFFF $CC00_H$ to FFFF $EFFF_H$.

Thus this RAM area is overwritten and must be re-initialized after completion of the self-programming.

Consider also that the stack occupies additionally up to 1 KB of the RAM.

7.4.1 Self-programming enable

The self-programming functions can be started out of the normal user mode of the V850 microcontroller.

The V850 microcontroller must be set into self-programming mode via the self-programming library.

For security reasons writing and erasing of the flash memory must be additionally permitted by setting the external FLMD0 pin to high level. Note that FLMD0 holds low level in normal operation mode after reset release.

This requires some external components or wiring, e.g. connecting an output port to FLMD0.

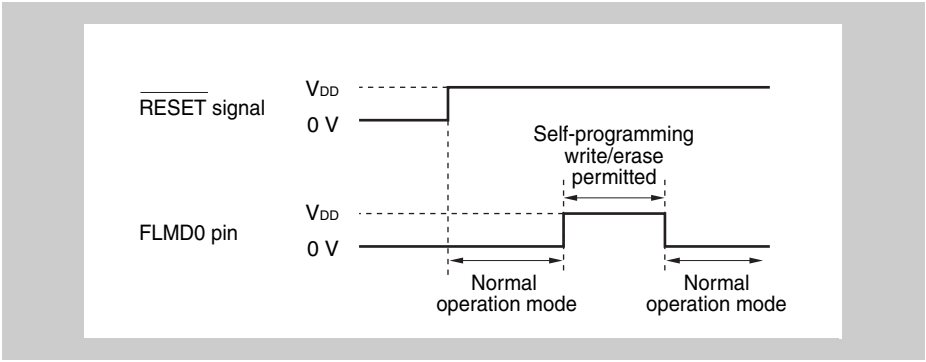


Figure 7-13 Self-programming enable

When self-programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

7.4.2 Self-programming library functions

Code flash memory self-programming by the user's program is supported by the self-programming library.

This library provides a set of C function calls to carry out basic functions like

- blank-check/erase/rewrite/verify of the flash
- boot cluster swapping, including definition of boot block clusters
- setting of protection flags
- obtain various information concerning the code flash memory

Detailed information how to use the library functions is given in the Application Note: "Self-Programming Library for embedded Single Voltage FLASH" (document no. U16929EE).

The up-to-date version of the self-programming library and the above mentioned Application Note can be obtained from *http://www.eu.necel.com*.

7.4.3 Secure self-programming (boot cluster swapping)

The V850 flash microcontrollers support a mechanism to swap a cluster of code flash memory blocks, starting from address 0000 0000_H, with another cluster of the same size, located immediately above the first one.

Boot swap cluster

The cluster of blocks starting at address 0000 0000_H is named active boot swap cluster, since it contains the entry point of the user's program at the default reset vector 0000 0000_H.

Boot swap flag

Which of the two clusters is the active boot block cluster is controlled by the boot swap flag, that can be defined during flash programming via the self-programming library.
The boot swap flag is stored in the flash memory extra area.

Figure 7-14 on page 215 shows an example of the boot block swapping function with a cluster size of 4 flash memory blocks. After inverting the boot_flag - it becomes not(boot_flag) - blocks 4 to 7 become the active boot block cluster. Thus after next reset release the user's program starts from the new boot swap cluster.

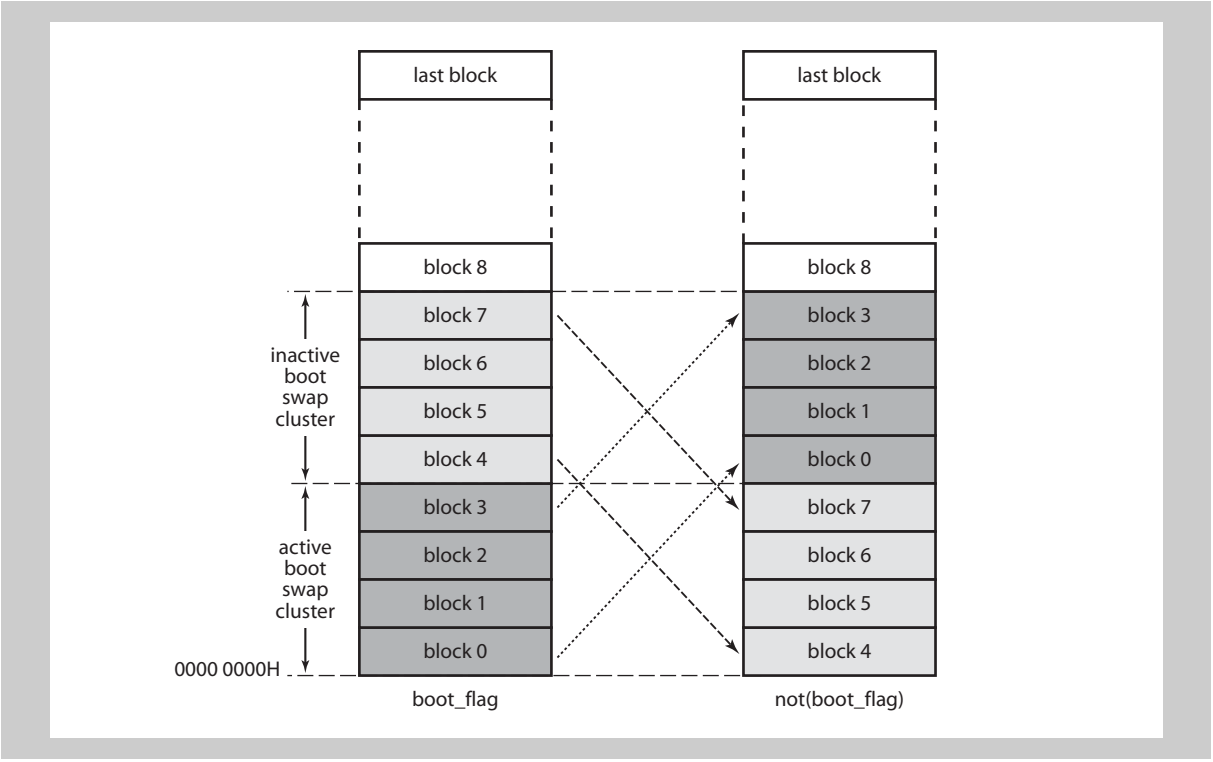


Figure 7-14 Boot swap cluster swapping function

Secure self-programming

The boot cluster swapping function enables secure self-programming. In case the boot code shall be rewritten, the new code can be written to the inactive boot block cluster, while the boot_flag remains in its previous state.
If rewriting of the boot block cluster has been completed successfully, the boot_flag can be inverted, making the new boot code active.
If rewriting of the new boot code fails for any reason, e.g. power fail or unintended reset, the old boot code still remains active and rewriting can be started again.

Boot block cluster

The boot code size itself may be smaller than the boot swap cluster size.

The number of flash memory blocks, which are part of the boot code, are named boot block cluster. The number of boot blocks, which are member of the cluster, can be defined during self-programming via the self-programming library.

The boot block cluster size of the V850E/PHO3 is fixed to 64 KB, i.e. it comprises 16 blocks.

Table 7-13 on page 216 shows the relation between the number of boot blocks, the boot block cluster size and the boot swap cluster.

Table 7-13 Relation between boot block and boot swap cluster

Number of boot blocks	Boot block cluster	Boot swap cluster
00_H	$0000\ 0000_H$ - $0000\ 0FFF_H$ (4 KB)	$0000\ 0000_H$ - $0000\ FFFF_H$ (64 KB)
01_H	$0000\ 0000_H$ - $0000\ 1FFF_H$ (8 KB)	
02_H	$0000\ 0000_H$ - $0000\ 2FFF_H$ (12 KB)	
03_H	$0000\ 0000_H$ - $0000\ 3FFF_H$ (16 KB)	
04_H	$0000\ 0000_H$ - $0000\ 4FFF_H$ (20 KB)	
...	...	
07_H	$0000\ 0000_H$ - $0000\ 7FFF_H$ (32 KB)	
08_H	0000 0000H - $0000\ 8FFF_H$ (36 KB)	
...	...	
$0F_H$	$0000\ 0000_H$ - $0000\ FFFF_H$ (64 KB)	
10_H	Setting prohibited	
...		
FF_H		

Number of boot blocks

The number of boot blocks has to be defined by the user during self-programming. It determines the blocks, which are subject to the boot block cluster protection, that allows to protect the boot blocks from any erase or write process.

Figure 7-15 on page 217 illustrates an example with following settings:

- number of boot blocks: 2 (boot block cluster contains 2 blocks), thus the active boot block cluster comprises
  - if boot_flag: blocks 0 and 1
  - if not(boot_flag): blocks 4 and 5
- active boot swap clusters comprises
  - if boot_flag: blocks 0 to 3
  - if not(boot_flag): blocks 4 to 7

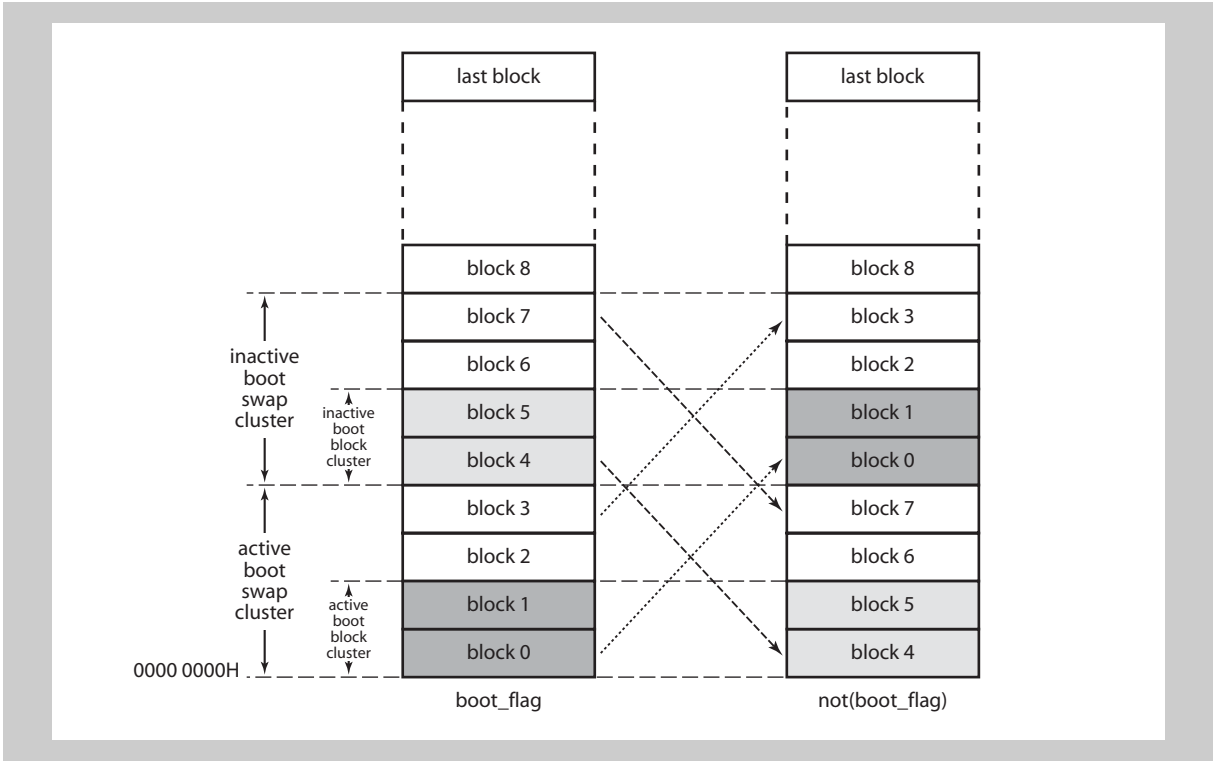


Figure 7-15 Boot cluster swapping function

Boot block protection

To prohibit rewriting of the boot blocks, the boot block cluster protection flag can be set during flash memory programming. When this flag is set, the blocks of the active boot block cluster can neither be erased nor written. Boot cluster swapping is impossible as well.
Note that only the blocks of the active boot block cluster are protected. In the example according to *Figure 7-15 on page 217*, for instance, blocks 0 and 1 would be prohibited, while blocks 2 and 3 could still be erased and written.

Caution

Once the boot block cluster protection has been activated, it can never be deactivated again.

For further information concerning flash memory protection flags refer to *"Data Protection and Security" on page 219*.

7.4.4 Interrupt handling during flash self-programming

This V850 microcontroller provides functions to maintain interrupt servicing during the self-programming procedure.

Since neither the interrupt vector table nor the interrupt handler routines, which are normally located in the code flash memory, are accessible while self-programming is active, interrupt acknowledges have to be re-routed to non-flash memory, i.e. to the internal RAM.

Therefore two prerequisites are necessary to enable interrupt servicing during self-programming:

- The concerned interrupt handler routine needs to be copied to the internal RAM, respectively external memory. The user has to initiate this copy process.
- The concerned interrupt acknowledge has to be re-routed to that handler. Re-routing to the handler is done by the internal firmware. Thus the user doesn't have to care about.

The internal firmware and the self-programming library provide functions to initialize and process such interrupts.

The interrupt handler routines can be copied from flash to the internal RAM by use of self-programming library functions.

The addresses of the interrupt handler routines are set up via the self-programming library as well.

Note 1. Note that this special interrupt handling adds some interrupt latency time.

2. Special interrupt handling is done only during the flash programming environment is activated. If self-programming is deactivated, the normal interrupt vector table in the flash memory is used.

All interrupt vectors are relocated to one entry point in the internal RAM:

- New entry point of *all* maskable interrupts is the 1st address of the internal RAM. A handler routine must check the interrupt source. The interrupt request source can be identified via the interrupt/exception source register ECR.EICC (refer to *"System register set" on page 101*)
- New entry point of *all* non maskable interrupts is the word address following the maskable interrupt entry, i.e. the second address of the internal RAM. The interrupt request source can be identified via the interrupt/exception source register ECR.FECC (refer to*"System register set" on page 101*).

In general a jump to a special handler routine will be placed at the 1st and 2nd internal RAM address, which identifies the interrupt sources and branches to the correct interrupt service routine.

The function serving the interrupt needs to be compiled as an interrupt function (i.e. terminate with a RETI instruction, save/restore all used registers, etc.).

It is recommended to refer to the Application Note "Self-Programming" (document nr. U16929EE) for comprehensive information concerning flash self-programming. This document explains also the functions of the self-programming library. The latest version of this document can be loaded via the URL

http://www.eu.necel.com

Chapter 8 Data Protection and Security

8.1 Overview

The microcontroller supports various methods for securing safe (re-)programming of the internal flash memory and protecting of the flash memory data against undesired access, such as illegal read-out or illegal reprogramming.

Security functions

Security functions enhance the reliability of the flash operation during normal operation and support countermeasures against unexpected failures during reprogramming processes. These are basically:

- Secure self-programming
- Secure bootloader update
- Boot block cluster protection

These functions are described in detail in *"Flash Memory" on page 191*.

Protection functions

Protection functions provide a set of mechanisms to protect the internal flash memory data from being read, erased or altered by unauthorized persons. These are basically:

- On-chip (N-Wire) debug interface protection
- Flash memory erase/write/read protection via the serial programming interface

Some interfaces offer in general access to the internal flash memory: N-Wire debug interface, external flash programmer interfaces and self-programming facilities. All of these interfaces need to be considered for a proper protection concept.

The following sections give an overview about supported protection methods.

8.2 N-Wire Debug Interface Protection

In general, read-out of the flash memory contents is possible via the N-Wire debug interface, but protection against illegal read-out can be enabled. For protection of the flash memory, the usage of the debug interface can be protected and it can be disabled. The debug interface is protected via a 10-byte ID code and an internal flag (N-Wire use enable flag).

When the debugger is started, the status of a flag is queried (N-Wire use enable flag). Set this flag to zero to disable the use of the N-Wire in-circuit emulator.

When debugging is enabled (N-Wire use enable flag is set), you have to enter a 10-byte ID code via the debugger. The code is compared with the ID code stored in the internal flash memory. If the codes do not match, debugging is not possible.

The N-Wire use enable flag can be set or reset while reprogramming the flash by an external flash writer or with the self-programming feature. The flag is located at bit 7 at address 0000 0079_H.

You can specify your own 10-byte ID code and program it to the internal flash memory by an external flash writer or with the self-programming feature. The ID code is located in the address range 0000 0070_H to 0000 0079_H.

The protection levels are summarized in *Table 8-1*

Table 8-1 Possible results of ID code comparison

N-Wire use enable flag	ID code	Protection Level
0	X[a]	Level 2: Full protection N-Wire debug interface cannot be used.[b]
1	user-specific ID code	Level 1: ID code protection user ID code N-Wire debug interface can only be used if the user enters the correct ID code.
	ID code is all ones[c]	Level 0: ID code protection with default ID code N-Wire debug interface can be used if the user enters the default ID code FF_H for all ID bytes.

a) Codes are not compared

b) Once the N-Wire debug interface has been set as “use-prohibited”, it cannot be used until the flash memory is re-programmed.

c) This is the default state after the flash memory has been erased.

Note

1. After you have set protection levels 1 or 2, set the “block erase disable flag” in the flash extra area. Otherwise, an unauthorized person could erase the block that contains the ID code or the “N-Wire use enable flag”, respectively, and thus suspend the protection.
2. If an unauthorized user tries to find out the 10-byte ID by comparing all possible ID codes, this will take up to 3.83 x 10^8 years at 100 MHz.

8.3 Flash Programmer and Self-Programming Protection

In general, illegal read-out and re-programming of the flash memory contents is possible via the flash writer interface and the self-programming feature. For protection of the flash memory, the set of flags, which can be set by the user, provide various protection levels.

Serial programming

It is possible to prohibit any access from external via the serial programming interface, e.g. by an external flash programmer. With maximum protection the internal flash memory can not be erased, read-out or written at all, neither in block units nor the entire flash memory.

Self-programming

During self-programming all operations to erase, read or program the flash memory is under control of the user's program. Thus no further protection functions in self-programming mode are considered. One exception is the boot block protection, which applies also in self-programming mode.

Protection flags

The protection flags can be set respectively reset by an external flash programmer, provided the effective protection level allows to do so.
In self-programming mode the effective protection flags can not be reset, but other ones can be set to enhance the protection level.
The protection flags are stored in the flash memory extra area.

Each protection function can be used in combination with the others at the same time.

(1) Write protection flag

Set this flag to disable the write function via external flash programmer interfaces.
No flash memory content can be written from external, if this flag is set. Erasure of single blocks is prohibited as well.

This flag does not affect the self-programming interface.
In self-programming mode writing of the flash memory is further on possible.

(2) Chip erase protection flag

Set this flag to disable the chip erase function via external flash programmer interfaces.
No flash memory content can be erased - neither in single blocks nor the entire flash memory - from external, if this flag is set.

Chip erase is not available in self-programming mode, though it is possible to erase the entire flash memory content by block erase of all blocks all together. Note that the contents of the extra area is not erased by this means. I.e. protection flags, etc. are still valid.

(3) Block erase protection flag

Set this flag to disable the feature to erase single blocks via external flash programmer interfaces.
Single blocks can not be erased. Chip erase is still possible, provided the chip erase protection flag is not set.

This flag does not affect the self-programming interface.
In self-programming mode erasure of single blocks or sets of contiguous blocks of the flash memory is further on possible.

(4) Read-out protection flag

Set this flag to disable the feature that allows reading back the flash memory via the serial programming interfaces.
No flash content can be read out.

This flag does not affect the self-programming interface.
In self-programming mode read-out of flash memory content is further on possible.

(5) Boot block cluster protection flag

Set this flag to disable erasure and rewrite of the boot block cluster.
The boot block cluster can not be manipulated in any way (no erase/write).

This applies in serial and self-programming mode.

Once this flag is set, it is impossible to reset this flag. Thus the boot block cluster content can not be changed any more.

For the explanation of the boot block cluster refer to *"Secure self-programming (boot cluster swapping)" on page 215*.

All protection flags are reset after shipment of the device, thus no protection is enabled at all.
Once a protection flag has been set, i.e. the protection is effective, it can not be reset by any means, except after a chip erase, which erases the entire flash memory including the extra area.
Consequently without prior chip erase the protection level can only be increased, but not decreased.

Table 8-2 Protection functions overview

Function	Functional outline	Applicable (√: applies, ×: doesn't apply)	
		Serial programming	Self-programming
Block erase command prohibit	Erasure of single blocks impossible. Once block erase protection is enabled, disable is only possible after chip erase.	√	×
Chip erase command prohibit	Erasure of the entire flash (including the extra area) or single blocks impossible. Once chip erase protection is enabled, all protection flag settings can not be changed any more.	√	×
Program command prohibit	Erasure and rewrite of single blocks impossible. Once write protection is enabled, disable is only possible after chip erase.	√	×
Read command prohibit	Read-out of any flash content impossible. Once read protection is enabled, disable is only possible after chip erase.	√	×
Rewriting boot block cluster prohibit	Erasure (by block or chip erase) or writing of the boot block cluster impossible. Once rewrite protection of the boot block cluster is enabled, it can not be disabled any more.	√	√

Table 8-3 Rewriting operation when erasing/writing is enabled/prohibited

Prohibition state		Programming mode	Block erasure		Chip erasure	Write	
			Boot area	None boot area		Boot area	None boot area
Rewriting boot area enabled	All enabled	Self-programming	yes		–	yes	
		Serial programming	yes		yes	yes	
	Block erase command prohibited	Self-programming	yes		–	yes	
		Serial programming	no		yes	yes	
	Chip erase command prohibited	Self-programming	yes		–	yes	
		Serial programming	no		no	yes	
	Write command prohibited	Self-programming	yes		–	yes	
		Serial programming	no		yes	no	
Rewriting boot area prohibited	All enabled	Self-programming	no	yes	–	no	yes
		Serial programming		yes	no		yes
	Block erase command prohibited	Self-programming		yes	–		yes
		Serial programming		no	yes		yes
	Chip erase command prohibited	Self-programming		yes	–		yes
		Serial programming		no	no		yes
	Write command prohibited	Self-programming		yes	–		yes
		Serial programming		no	yes		no

Note –: not supported

Table 8-4 Read operation when reading is enabled/prohibited

Prohibition State	Programming mode	Read
Read command enabled	Self-programming	√
	Serial programming	√
Read command prohibited	Self-programming	√
	Serial programming	×

Note √: execution enabled, ×: execution disabled, –: not supported

Chapter 9 Bus and Memory Control (BCU, MEMC)

Besides providing access to on-chip peripheral I/Os, the microcontroller device supports access to external memory devices (such as external ROM and RAM) and external I/O. The Bus Control Unit BCU and Memory Controller MEMC control the access to on-chip peripheral I/Os and to external devices.

9.1 Overview

The following external devices can be connected to the microcontroller device:

- SRAM / RAM
- ROM
- External I/O

Features summary

The bus and memory control of the microcontroller device provides:

- 22 address signals (A0 to A21)
- Selectable data bus width for each chip select area (8 bits, 16 bits and 32 bits)
- 4 chip select signals externally available ($\overline{CS0}$, $\overline{CS2}$, $\overline{CS3}$ and $\overline{CS4}$) to access external memory
- Access to memory takes a minimum of two CPU clock cycles
- An address setup wait state and an address hold state can be inserted for each chip select area
- Up to 7 data wait states can be inserted for each chip select area (programmable wait)
- External data wait function through $\overline{WAIT}$ pin
- Idle state can be inserted after a read/write cycle

9.1.1 Description

The figure below shows a block diagram of the modules that are necessary for accessing on-chip peripherals, external memory, or external I/O.

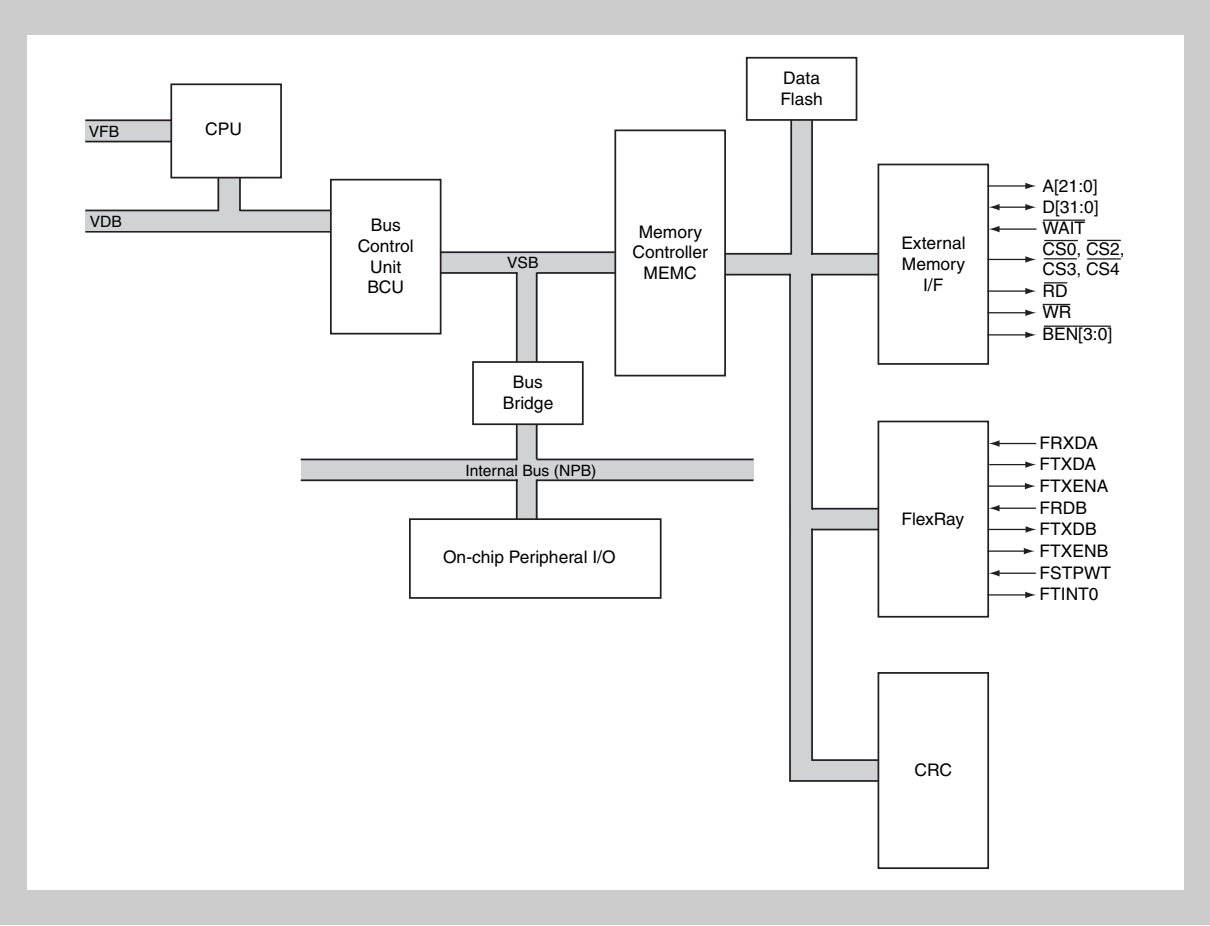


Figure 9-1 Bus and Memory Control block diagram

Busses The busses are abbreviated as follows:

- NPB: NEC peripheral bus
- VSB: V850 system bus
- VDB: V850 data bus
- VFB: V850 fetch bus

BCU The Bus Control Unit (BCU) controls the access to on-chip peripherals, to external memory, and to external I/O.

For access to external devices, the BCU generates the necessary control signals (chip select signals) for the Memory Controller.

Memory Controller The 256 MB address range is divided into 2-MB memory banks and 64-MB memory areas. Each of the memory areas/banks can be assigned to an external device via the chip select area control registers CSC0 and CSC1.

If an instruction uses such an address, a chip select signal is generated. The device supports four chip select signals ($\overline{CS0}$, $\overline{CS2}$, $\overline{CS3}$ and $\overline{CS4}$). Each chip select signal covers a certain address range, also called “chip select area”. For details see *“Memory areas/banks and chip select signals” on page 228*.

The Memory Controller generates the control signals for access to the external devices, the FlexRay module, the CPU- and DATA-CRC units and the data

flash. For example, it generates the read strobe ($\overline{RD}$) and the write strobe ($\overline{WR}$). From the 28 bit address of the CPU, the lower 22 bits are passed to the external device.

If two chip select signals are specified in the CSCn registers for a single memory area/bank, the priority control selects one of the chip select signals. The priority order is given in *"CSCn - Chip select area control registers" on page 242*.

The external signals of the Memory Controller are listed in the following table:

Table 9-1 Memory Controller external connections

Signal name	I/O	Active level	Pins	Function
$\overline{CS0}$	O	L	$\overline{CS0}$	Chip select signal
$\overline{CS2}$	O	L	$\overline{CS2}$	Chip select signal
$\overline{CS3}$	O	L	$\overline{CS3}$	Chip select signal
$\overline{CS4}$	O	L	$\overline{CS4}$	Chip select signal
A[0:21]	O	–	A0 to A21	Address bus
D[0:31]	I/O	–	D0 to D31	Data bus
$\overline{WAIT}$	I	L	$\overline{WAIT}$	Data wait
$\overline{WR}$	O	L	$\overline{WR}$	Write strobe
$\overline{RD}$	O	L	$\overline{RD}$	Read strobe
$\overline{BEN0}$	O	L	$\overline{BEN0}$	Byte enable output (D0-D7)
$\overline{BEN1}$	O	L	$\overline{BEN1}$	Byte enable output (D8-D15)
$\overline{BEN2}$	O	L	$\overline{BEN2}$	Byte enable output (D16-D23)
$\overline{BEN3}$	O	L	$\overline{BEN3}$	Byte enable output (D24-D31)

All pins are in input port mode after reset. Refer to *"Pin Functions" on page 1*.

Configuration

The microcontroller device supports interfacing with various memory devices. To make the bus and Memory Controller suitable for the connected device, the endian format, wait functions and idle state insertions can be configured.

For a detailed description, see *"Configuration of Memory Access" on page 256*.

9.1.2 Memory areas/banks and chip select signals

The 256 MB address range is divided into the 64 MB areas 0 to 3 (see *Figure 9-2 on page 229*).

The chip select signals $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$, $\overline{CS6}$ are fixed assigned to these areas.

Memory areas 0 and 3 are further subdivided into

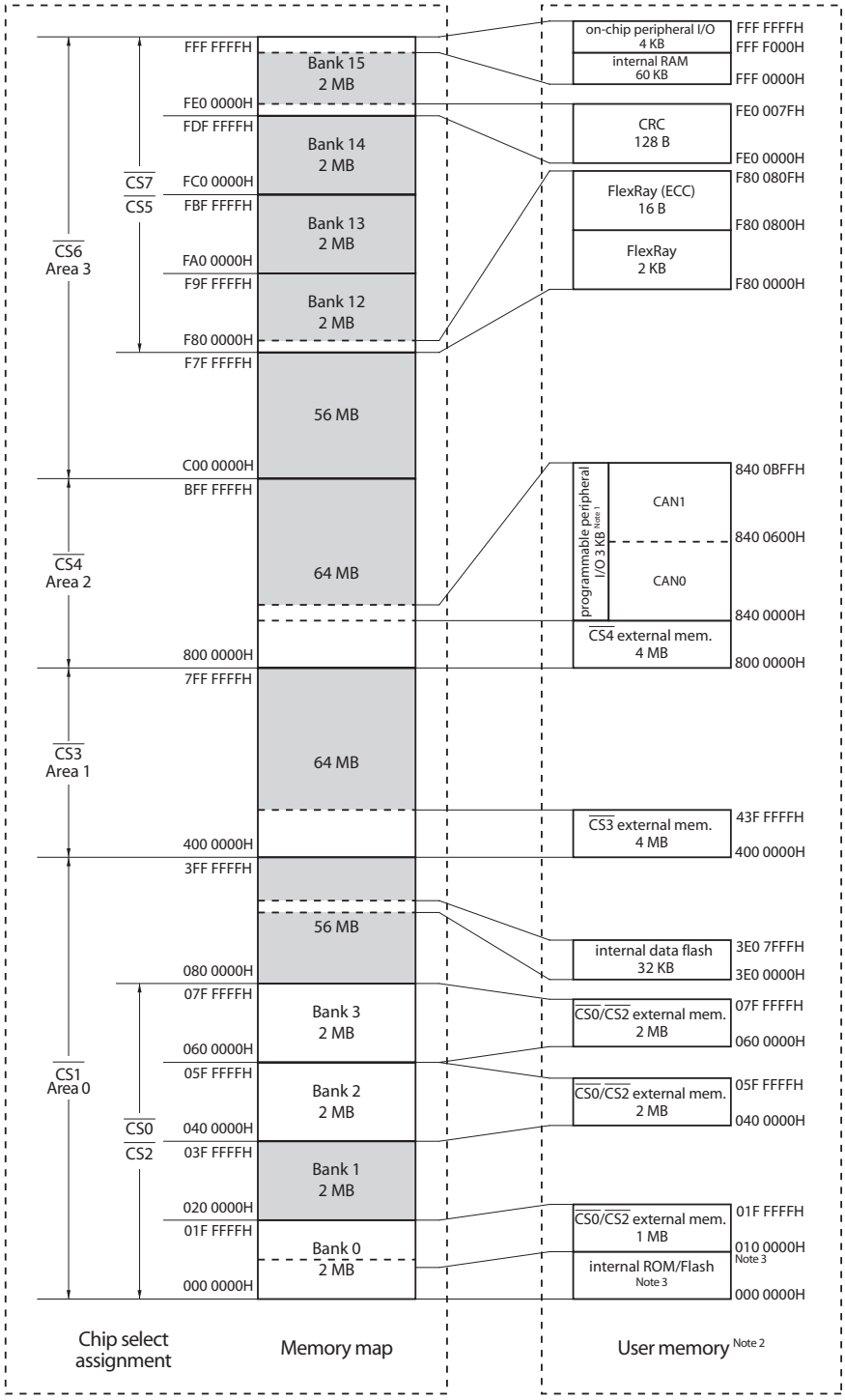
- four 2 MB memory banks (banks 0 to 3, respectively 12 to 15)
- the remaining 56 MB

Each address in the address range of the memory banks 0 to 3 and 12 to 15 is assigned to one or more chip select ($\overline{CS}$) signals by configuring the chip select area control registers CSC0 and CSC1:

- For addresses that lie within one of the memory banks, the corresponding chip select signal can be configured.
  If a memory bank is configured for external access, access to that memory bank generates the corresponding chip select signals.
- For addresses which do not lie within any of the memory banks, the corresponding chip select signal of the memory area is allocated by default.

Chip select area The memory area that activates the same chip select signal is called chip select area.

Note Throughout this chapter, the individual chip select signals are identified by “k” (k = 0 to 7), for example $\overline{CSk}$ for the chip select signal k or BSC.BSCk[1:0] for setting the data bus width of the memory banks corresponding to chip select signal k.



Note 1: The shown base address 0840 0000H of the programmable peripheral area assumes the Peripheral area selection control register BPC = A100H.

Note 2: Access to memory areas others than the shown user memory areas is forbidden.

Note 3: μPD70F3483: 768 KB up to 000B FFFFH; μPD70F3441: 1 MB up to 000F 7FFFH

The access to the grey shaded areas is prohibited!

Figure 9-2 Memory areas/banks and chip select signals

9.1.3 Peripheral I/O areas

Two areas of the address range are reserved for the registers of the on-chip peripheral functions. These areas are called “peripheral I/O areas”:

Table 9-2 Peripheral I/O areas

Name	Address range	Size
Fixed peripheral I/O area	0FFF $F000_H$ to 0FFF $FFFF_H$	4 KB
Programmable peripheral I/O area (PPA)	Can be allocated at arbitrary addresses. Base address is defined in the BPC register.	16 KB

(1) Fixed peripheral I/O area

The fixed peripheral I/O area holds the registers of the on-chip peripheral I/O functions.

Access timing During a read or write access the CPU operation stops until the access via the NPB is completed.

Note Because the address space covers 256 MB, the address bits A[31:28] are not considered. Therefore, in this manual, all addresses of peripheral I/O registers in the 4 KB peripheral I/O area are given in the range FFFF $F000_H$ to FFFF $FFFF_H$ instead of 0FFF $F000_H$ to 0FFF $FFFF_H$.

(2) Programmable peripheral I/O area (PPA)

The usage and the address range of the PPA is configurable. The PPA extends the fixed peripheral I/O area and assigns an additional 12 KB address space for accessing on-chip peripherals.

The figure below illustrates the programmable peripheral I/O area (PPA).

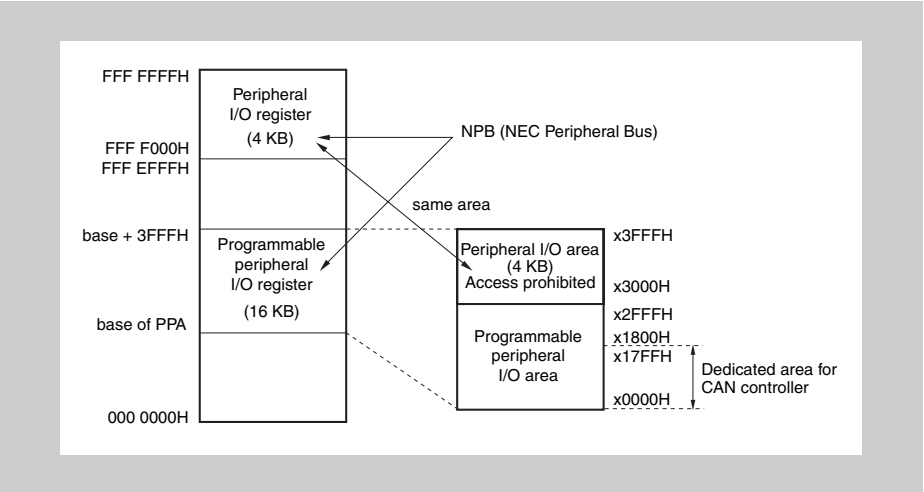


Figure 9-3 Programmable peripheral I/O area

The CAN modules registers and message buffers are allocated to the PPA. Refer to *“CAN module register and message buffer addresses” on page 454* for information how to calculate the register and message buffer addresses of the CAN modules.

Access timing During a read access the CPU operation stops until the read access via the NPB is completed.

During a write access the CPU operation continues operation, provided any preceded NPB access is already finished. If a preceded NPB access is still ongoing the CPU stops until this access is finished and the NPB is cleared.

Caution

1. If the programmable peripheral I/O area overlaps one of the following areas, the programmable peripheral I/O area becomes ineffective:
   - Peripheral I/O area
   - ROM area
   - RAM area
2. The *fixed* peripheral I/O area is mirrored to the upper 4 KB of the *programmable* peripheral I/O area – regardless of the base address of the PPA. Access to this mirror of the *fixed* peripheral I/O area is prohibited.

Note All address definitions in this manual that refer to the programmable peripheral area assume that the base address of the PPA is 840 0000_H, that means BPC = $A100_H$.

9.1.4 Data flash access

The data flash is connected to the memory bus, but is built into the microcontroller. The data flash is activated by chip select signal $\overline{CS1}$. The data flash can be accessed in the same way as external devices that are attached to the external memory bus. Thus all options to configure access via the external memory bus must be set up for the data flash as well.

Table 9-4 shows all required register settings for the data flash's chip select signal $\overline{CS1}$. Some settings are appropriate for the data flash per default, others must be changed before the first access to the data flash.

For details about the control settings refer to the description of the registers.

Table 9-3 Register settings for data flash access

Control bit	Required setting	Comment
CSC0.CS1[3:0]	0001_B	• area 0 assigned to $\overline{CS1}$ • default, don't change
BSC.BSC1[1:0]	10_B	• 32 bit bus width • default, don't change
BEC.BE10	0	• little endian • default, don't change
BCT0.ME1	1	• enable $\overline{CS1}$ • no default, must be changed
AWC.AHW1 AWC.ASW1	00_B	• no address setup/hold waits • no default, must be changed
DWC0.DWC1[2:0]	001_B	• 1 data wait state • no default, must be changed
BCC.BC11	0	• no idle states • no default, must be changed

Base address The base address of the data flash is 03E0 0000_H.

9.1.5 FlexRay access

The FlexRay module is connected to the memory bus, but is built into the microcontroller. The module is activated by chip select signal $\overline{CS5}$. The FlexRay module can be accessed in the same way as external devices that are attached to the external memory bus. Thus all options to configure access via the external memory bus must be set up for the FlexRay module as well.

Table 9-4 shows all required register settings for the FlexRay module's chip select signal $\overline{CS5}$. Some settings are appropriate for the FlexRay module per default, others must be changed before the first access to the FlexRay module.

For details about the control settings refer to the description of the registers.

Table 9-4 Register settings for FlexRay module access

Control bit	Required setting	Comment
CSC1.CS5[3:0]	1000_B	• bank 12 assigned to $\overline{CS5}$ • no default, must be changed
BSC.BSC5[1:0]	10_B	• 32 bit bus width • default, don't change
BEC.BE50	0	• little endian • default, don't change
BCT1.ME5	1	• enable $\overline{CS5}$ • no default, must be changed
AWC.AHW5 AWC.ASW5	00_B	• no address setup/hold waits • no default, must be changed
DWC1.DWC5[2:0]	001_B	• 1 data wait state • no default, must be changed
BCC.BC51	0	• no idle states • no default, must be changed

Base address The base address of the FlexRay is 0F80 0000_H.

9.1.6 CRC module access

The CRC module is connected to the memory bus, but is built into the microcontroller. The module is activated by chip select signal $\overline{CS7}$. The CRC module can be accessed in the same way as external devices that are attached to the external memory bus. Thus all options to configure access via the external memory bus must be set up for the CRC module as well.

Table 9-4 shows all required register settings for the CRC module's chip select signal $\overline{CS7}$. Some settings are appropriate for the CRC module per default, others must be changed before the first access to the CRC module.

For details about the control settings refer to the description of the registers.

Table 9-5 Register settings for CRC module access

Control bit	Required setting	Comment
CSC1.CS7[3:0]	0001_B	• bank 15 assigned to $\overline{CS7}$ • default, don't change
BSC.BSC7[1:0]	10_B	• 32 bit bus width • default, don't change
BEC.BE70	0	• little endian • default, don't change
BCT1.ME7	1	• enable $\overline{CS7}$ • no default, must be changed
AWC.AHW7 AWC.ASW7	00_B	• no address setup/hold waits • no default, must be changed
DWC1.DWC7[2:0]	000_B	• no data wait states • no default, must be changed
BCC.BC71	0	• no idle states • no default, must be changed

Base address The base address of the CRC module is 0FE0 0000_H.

9.1.7 Chip select priority control

The chip select signals $\overline{CS0}$ to $\overline{CS7}$ can be assigned to overlapping memory areas by setting the chip select area control registers CSC0 and CSC1. The chip select priority control rules the generation of chip select signals in this case.

Access to internal resources, which are mapped to a memory bank respectively area overrules the external access. As a consequence, the assigned $\overline{CSn}$ signal is not generated externally.

If different chip select signals are set (CSC0.CSCkm = 1) for the same memory bank or area, the priority order is as follows:

- internal resource > $\overline{CS0}$ > $\overline{CS2}$ > $\overline{CS1}$
- internal resource > $\overline{CS7}$ > $\overline{CS5}$ > $\overline{CS6}$
- internal resource > $\overline{CS4}$
- internal resource > $\overline{CS3}$

Examples:

- If both chip select signal $\overline{CS0}$ and $\overline{CS1}$ are set for memory bank 1, only the chip select signal $\overline{CS0}$ will be generated.
- If during access to bank 2 $\overline{CS2}$ should *not* be active, activate $\overline{CS0}$ for this bank (CSC0.CS02 = 1). Due to the priority order, only chip select signal $\overline{CS0}$ will be active for bank 2.

9.1.8 Bus properties

This section summarizes the properties of the external bus.

(1) Bus width

The microcontroller device accesses external memory and external I/O in 8-bit, 16-bit, or 32-bit units.

The data bus size for each chip select area is specified in the bus size configuration register (BSC).

The operation for each type of access is given in *"Access to 8-bit data busses" on page 267* and in *"Access to 16-bit data busses" on page 273*.

(2) Bus priority order

There are two kinds of external bus cycles as shown below. A CPU operand data access has higher priority than an instruction fetch.

Table 9-6 Bus priority order

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

(3) Bus access

The number of CPU (VBCLK) clocks necessary for accessing each resource - independent of the bus width - is as follows:

Table 9-7 Number of bus access clocks

Bus cycle configuration		Internal code flash	Internal RAM	External memory
Instruction fetch	Consecutive addresses access	1	1[a]	(2 + n) * m
	Branch	7	1[a]	(2 + n) * m
Operand data access		7	1	(2 + n) * m

a) In case of contention with data access, the instruction fetch from internal RAM takes 2 clocks.

Note n: number of data wait states, defined in registers DWC0 and DWC1

m: memory controller clock division factor, defined in register DVC0:
– DVC0.DVC0 = 0 (BCLK = VBCLK/2):m = 2
– DVC0.DVC0 = 1 (BCLK = VBCLK/4):m = 4

9.1.9 Boundary operation conditions

The microcontroller device has the following boundary operation conditions:

(1) Program space

Instruction fetches from the internal peripheral I/O area are inhibited and yield NOP operations.

If a branch instruction exists at the upper limit of the internal RAM area, a pre-fetch operation (invalid fetch) that straddles over the internal peripheral I/O area does not occur.

(2) Data space

The microcontroller device is provided with an address misalign function.

By this function, data of any format (word: 32 bit, halfword: 16 bit, byte: 8 bit) can be placed to any address in memory, even though the address is not aligned to the data format (that means address 4n for words, address 2n for halfwords).

- Unaligned halfword data access
  When the LSB of the address is A0 =1, two byte accesses are performed.
- Unaligned word data access
  When the LSB of the address is A0 =1, two byte and one halfword accesses are performed. In total it takes 3 bus cycles.
  – When the LSBs of the address are A[1:0] =10_B, two halfword accesses are performed.

Note Accessing data on misaligned addresses takes more than one bus cycle to complete data read/write. Consequently, the bus efficiency will drop.

9.1.10 Initialization for access to external devices

To enable access to external devices as well as the internal FlexRay communication controller, initialize the following registers after any reset.

1. Chip select area control registers CSCn
   Define the memory banks that are allocated to external devices. Memory banks that are not allocated to external devices, must be deactivated.
2. Bus cycle type configuration registers BCTn
   Specify the external devices that are connected to the microcontroller device. For memory banks that are not allocated to external devices, the corresponding bits in registers BCT0 and BCT1 should be reset.
3. Bus size configuration register BSC
   Set the data bus width for the active chip select areas.
4. Data wait control registers DWCn
   Set the number of data wait states with respect to the starting bus cycle.
5. Bus cycle control register BCC
   Specify insertion of an idle state after read.
6. Endian configuration register BEC
   Set the endian format for each chip select area.
7. Address setup wait control register AWC
   Set the number of address setup wait and address hold wait states for each chip select area.
8. Control register DVC
   Specify insertion of an idle state after write.

Caution
1. Do not change these registers after initialization.
2. Do not access external devices before initialization is finished.

9.2 Registers

Access to on-chip peripherals, to external memory, and to external I/O is controlled and operated by registers of the Bus Control Unit (BCU) and of the Memory Controller:

Table 9-8 Bus and memory control register overview

Module	Register name	Shortcut	Address
Bus Control Unit (BCU)	Peripheral area selection control register	BPC	FFFF $F064_H$
	Internal peripheral function wait control register	VSWC	FFFF $F06E_H$
	Chip select area control registers	CSC0	FFFF $F060_H$
		CSC1	FFFF $F062_H$
	Bus size configuration register	BSC	FFFF $F066_H$
	Endian configuration register	BEC	FFFF $F068_H$
Memory Controller (MEMC)	Bus cycle configuration registers	BCT0	FFFF $F480_H$
		BCT1	FFFF $F482_H$
	Address wait control register	AWC	FFFF $F488_H$
	Data wait control registers	DWC0	FFFF $F484_H$
		DWC1	FFFF $F486_H$
	Bus cycle control register	BCC	FFFF $F48A_H$
	Bus clock dividing frequency control register	DVC	FFFF $F48E_H$

9.2.1 BCU registers

The following registers are part of the BCU. They define the data bus width, the endian format of word data, and they control access to external devices.

(1) BPC - Peripheral area selection control register

The 16-bit BPC register defines whether the programmable peripheral I/O area (PPA) is used or not and determines the starting address of the PPA.

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

Address FFFF F064$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA15	0	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

Table 9-9 BPC register contents

Bit Position	Bit Name	Function
15	PA15	Select usage of programmable peripheral I/O area (PPA). 0: PPA disabled 1: PPA enabled
13 to 0	PA[13:0]	Bits PA[13:0] specify bits 27 to 14 of the starting address of the PPA. The other bits of the address are fixed to 0.

Caution The bits marked with 0 must always be 0.
The base address PBA of the programmable peripheral area sets the start address of the 16 KB PPA in a range of 256 MB. The 256 MB page is mirrored 16 times to the entire 32-bit address range.

The base address PBA is calculated by

$$PBA = BPC.PA[13:0] \times 2^{14}$$

Table 9-10 shows how the addresses of the programmable peripheral area are assembled. The base address PBA is highlighted.

Table 9-10 Address range of programmable peripheral area (16 KB)

31	…	28	27	…	14	13	…	1	0	bit
0	…	0	BPC.PA[13:0]			1	…	1	1	
	…		…				…			
0	…	0	BPC.PA[13:0]			0	…	0	1	
0	…	0	BPC.PA[13:0]			0	…	0	0	PBA

Note The recommended setting for the BPC register is A100$_H$. With this configuration the programmable peripheral area is mapped to the address range 0840 0000$_H$ to 0840 3FFF$_H$. With this setting the CAN message buffer registers are accessible via the addresses given in *"CAN Controller (CAN)" on page 429*.

(2) VSWC - Internal peripheral function wait control register

The 8-bit VSWC register controls the bus access wait for the on-chip peripheral I/O registers.

Access to on-chip peripheral I/O registers is made in 3 clocks (without wait), however, waits may be required depending on the operation frequency. Set the values described below to the VSWC register in accordance with the operation frequency used.

For DMA transfers, the bus access wait and signal timing is controlled by registers DMAWC0 and DMAWC1. For a description of these registers see *"DMA Functions (DMA Controller)" on page 279*.

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

Address FFFF F06E$_H$

Initial Value 77$_H$

VSWC must be initialized after $\overline{\text{RESET}}$ according to below table:

Table 9-11 Initialization of VSWC

System clock	128 MHz	80 MHz
VSWC	37$_H$	24$_H$

Note The bits marked with 0 must always be 0.

(3) CSCn - Chip select area control registers

The 16-bit registers CSC0 and CSC1 assign the chip select signals $\overline{CS0}$ to $\overline{CS3}$ and $\overline{CS4}$ to $\overline{CS7}$ to memory banks respectively areas (see also *"Memory areas/banks and chip select signals" on page 228*). If a bit in CSCn is set, access to the corresponding memory bank or area will generate the corresponding chip select signal and activate the Memory Controller.

$\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$ and $\overline{CS6}$ are fixed assigned to complete 64 MB memory areas.

$\overline{CS0}$, $\overline{CS2}$, $\overline{CS5}$ and $\overline{CS7}$ can be assigned independently to several 2 MB memory banks.

If several chip select signals are assigned to identical memory areas, a priority control rules the generation of the signals (refer to *"Chip select priority control" on page 235*).

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

Address CSC0: FFFF $F060_H$
CSC1: FFFF $F062_H$

Initial Value $2C11_H$
These registers must be initialized as described in *Table 9-14 on page 244*.

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
$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$			

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
$\overline{CS4}$				$\overline{CS5}$				$\overline{CS6}$				$\overline{CS7}$			

Caution To initialize an external memory area after a reset, registers CSCn have to be set. Do not change these registers after initialization. Do not access external devices before initialization is finished.

The register contents in *Table 9-12* and *Table 9-13* read as follows:

- CSkm = 0: Corresponding chip select signal is *not* active during access to memory bank/area.
- CSkm = 1: Corresponding chip select signal is active during access to memory bank/area.

Table 9-12 CSC0 register contents

Bit Position	Bit Name	Access to memory area or bank	Chip select signal
15 to 12	CS3[3:0]	area 1, independent of bit 15 to 12 setting	$\overline{CS3}$
11	CS23	bank 3	$\overline{CS2}$
10	CS22	bank 2	
9	CS21	bank 1	
8	CS20	bank 0	
7 to 4	CS1[3:0]	area 0, independent of bit 7 to 4 setting	$\overline{CS1}$
3	CS03	bank 3	$\overline{CS0}$
2	CS02	bank 2	
1	CS01	bank 1	
0	CS00	bank 0	

Table 9-13 CSC1 register contents

Bit Position	Bit Name	Access to memory area or bank	Chip select signal
15 to 12	CS4[3:0]	area 2, independent of bit 15 to 12 setting	$\overline{CS4}$
11	CS53	bank 12	$\overline{CS5}$
10	CS52	bank 13	
9	CS51	bank 14	
8	CS50	bank 15	
7 to 4	CS6[3:0]	area 3, independent of bit 7 to 4 setting	$\overline{CS6}$
3	CS73	bank 12	$\overline{CS7}$
2	CS72	bank 13	
1	CS71	bank 14	
0	CS70	bank 15	

Initialization Initialize the registers as follows:

Table 9-14 Initialization of CSCn

Bits	Set to value	Comment
CSC0.CS0[3:0]	$xx0x_B$	Set as required to assign $\overline{CS0}$ to bank 0, 2, 3[a]: • 0000_B: no assignment • 0001_B: bank 0 • 0100_B: bank 2 • 0101_B: bank 0, 2 • 1000_B: bank 3 • 1001_B: bank 0, 3 • 1100_B: bank 2, 3 • 1101_B: bank 0, 2, 3
CSC0.CS1[3:0]	0001_B	Default, don't change $\overline{CS1}$ assigned to area 0 for internal data flash
CSC0.CS2[3:0]	$xx0x_B$	Set as required to assign $\overline{CS2}$ to bank 0, 2, 3[a]: • 0000_B: no assignment • 0001_B: bank 0 • 0100_B: bank 2 • 0101_B: bank 0, 2 • 1000_B: bank 3 • 1001_B: bank 0, 3 • 1100_B: bank 2, 3 • 1101_B: bank 0, 2, 3
CSC0.CS3[3:0]	0010_B	Default, don't change $\overline{CS3}$ is assigned to external memory in area 1 400 0000_H - 7FF $FFFF_H$
CSC1.CS4[3:0]	0010_B	Default, don't change $\overline{CS4}$ is assigned to external memory in area 2 800 0000_H - BFF $FFFF_H$
CSC1.CS5[3:0]	1000_B	$\overline{CS5}$ assigned to bank 12 for internal FlexRay **Caution:** These bits must be changed to 1000_B (default value is 1100_B).
CSC1.CS6[3:0]	0001_B	Default, don't change
CSC1.CS7[3:0]	0001_B	Default, don't change $\overline{CS7}$ assigned to bank 15 for internal CRC

a) - bank 0 external memory address rage: 010 0000_H - 01F $FFFF_H$
- bank 2 external memory address rage: 040 0000_H - 05F $FFFF_H$
- bank 3 external memory address rage: 060 0000_H - 07F $FFFF_H$

(4) BSC - Bus size configuration register

The 16-bit BSC register controls the data bus width for each chip select area.

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

Address FFFF F066$_H$

Initial Value AAAA$_H$
This register must be initialized as described in *Table 9-16*.

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
$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

Table 9-15 BSC register contents

Bit position	Bit name	Function
15 to 0	BSk[1:0	Sets the data bus width for each chip select area.

BSk1	BSk0	Data bus size
0	0	8 bit
0	1	16 bit
1	0	32 bit
1	1	prohibited

Caution To initialize an external memory area after a reset, register BSC has to be set. Do not change this register after initialization. Do not access external devices before initialization is finished.

Initialization Initialize the register as follows:

Table 9-16 Initialization of BSC

Bits	Set to value	Comment
BSC.BS0[1:0]	xx$_B$	Set as required
BSC.BS1[1:0]	10$_B$	Default, don't change Data bus width for internal data flash: 32 bit
BSC.BS2[1:0]	xx$_B$	Set as required
BSC.BS3[1:0]	xx$_B$	Set as required
BSC.BS4[1:0]	xx$_B$	Set as required
BSC.BS5[1:0]	10$_B$	Default, don't change Data bus width for internal FlexRay: 32 bit
BSC.BS6[1:0]	10$_B$	Default, don't change
BSC.BS7[1:0]	10$_B$	Default, don't change Data bus width for internal CRC: 32 bit

(5) BEC - Endian configuration register

The 16-bit BEC register defines the endian format in which word data in the memory is processed. Each chip select area is controlled separately.

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

Address FFFF F068$_H$

Initial Value 0000$_H$
This register must be initialized as described in *Table 9-18*

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
$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

Table 9-17 BEC register contents

Bit Position	Bit Name	Function
14, 12, 10, 8, 6, 4, 0	BEk0	Sets the endian format for processing of word data for each chip select area. 0: Little endian 1: Big endian

Caution
1. The bits marked with 0 must always be 0.
2. To initialize an external memory area after a reset, register BEC has to be set. Do not change this register after initialization. Do not access external devices before initialization is finished.

Initialization Initialize the register as follows:

Table 9-18 Initialization of BEC

Bits	Set to value	Comment
BEC.BE00	x_B	Set as required
BEC.BE10	0_B	Default, don't change Endian format for internal data flash: little endian
BEC.BE20	x_B	Set as required
BEC.BE30	x_B	Set as required
BEC.BE40	x_B	Set as required
BEC.BE50	0_B	Default, don't change Endian format for internal FlexRay: little endian
BEC.BE60	0_B	Default, don't change
BEC.BE70	0_B	Default, don't change Endian format for internal CRC: little endian

Note Set the chip select area which is specified as the programmable peripheral I/O area to little endian format.

9.2.2 Memory Controller registers

The following registers are part of the Memory Controller. They specify the number of data wait states, the number of address wait states, and the number of idle states.

(1) BCTn - Bus cycle configuration registers

The 16-bit BCTn registers enable the operation of the Memory Controller for each chip select signal.

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

Address BCT0: FFFF $F480_H$
BCT1: FFFF $F482_H$

Initial Value 4444_H
These registers must be initialized as described in *Table 9-20 on page 248*.

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
$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$			

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
$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$			

Table 9-19 BCTn register contents

Bit Position	Bit Name	Function
15, 11, 7, 3	MEk	Enables/disables Memory Controller operation for chip select area k. 0: Operation disabled 1: Operation enabled

Caution

1. The bits marked with 0 must always be 0.
2. The bits marked with 1 must always be 1.
3. To initialize an external memory area after a reset, registers BCTn have to be set. Do not change this register after initialization. Do not access external devices before initialization is finished.

Initialization Initialize the registers as follows:

Table 9-20 Initialization of BCTn

Bits	Set to value	Comment
BCT0.ME0	x_B	Set as required
BCT0.ME1	1_B	Enable internal data flash. **Caution:** This bit must be changed to 1 (default value is 0).
BCT0.ME2	x_B	Set as required
BCT0.ME3	x_B	Set as required
BCT1.ME4	x_B	Set as required
BCT1.ME5	1_B	Enable internal FlexRay. **Caution:** This bit must be changed to 1 (default value is 0).
BCT1.ME6	0_B	Default, don't change
BCT1.ME7	1_B	Enable internal CRC. **Caution:** This bit must be changed to 1 (default value is 0).

(2) AWC - Address wait control register

The 16-bit AWC register controls the insertion of an address setup wait and address hold wait state before and after the T1 cycle. The address setup wait and address hold wait state can be enabled for each chip select area.

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

Address FFFF F488$_H$

Initial Value FFFF$_H$: After system setup, by default, address hold and wait states insertion is enabled for each chip select area.
This register must be initialized as described in *Table 9-22 on page 250*

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
$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

Table 9-21 AWC register contents

Bit position	Bit name	Function
1, 3, 5, 7, 9, 11, 13, 15	AHWk	Enables/disables the address hold wait insertion for chip select areas 0: No wait state inserted 1: Address hold wait state inserted after T1 bus cycle
0, 2, 4, 6, 8, 10, 12, 14	ASWk	Enables/disables the address setup wait insertion for chip select areas 0: No wait state inserted 1: Address setup wait state inserted before T1 bus cycle

Caution To initialize an external memory area after a reset, register AWC has to be set. Do not change this register after initialization. Do not access external devices before initialization is finished.

Note For access to internal memory and peripheral I/O areas, programmable waits are *not* carried out.

Initialization Initialize the registers as follows:

Table 9-22 Initialization of AWC

Bits	Set to value	Comment
AWC.AHW0 AWC.ASW0	x x	Set as required
AWC.AHW1 AWC.ASW1	0 0	No insertions for internal data flash. **Caution:** These bits must be changed to 00_B (default value is 11_B).
AWC.AHW2 AWC.ASW2	x x	Set as required
AWC.AHW3 AWC.ASW3	x x	Set as required
AWC.AHW4 AWC.ASW4	x x	Set as required
AWC.AHW5 AWC.ASW5	0 0	No insertions for internal FlexRay. **Caution:** These bits must be changed to 00_B (default value is 11_B).
AWC.AHW6 AWC.ASW6	1 1	Default, don't change
AWC.AHW7 AWC.ASW7	0 0	No insertions for internal CRC. **Caution:** These bits must be changed to 00_B (default value is 11_B).

(3) DWCn - Data wait control registers

The 16-bit DWCn registers control the number of data wait states after the first access cycle (T1). Each chip select area is controlled separately. A maximum of seven data wait states is possible.

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

Address DWC0: FFFF F484$_H$
DWC1: FFFF F486$_H$

Initial Value 7777$_H$: After system setup, by default, seven data wait states are inserted for each chip select area.
These registers must be initialized as described in *Table 9-24 on page 252*.

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
$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$			

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
$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$			

Table 9-23 DWCn register contents

Bit position	Bit name	Function
14 to 12, 10 to 8, 6 to 4 2 to 0	DWk[2:0]	Sets the number of wait states after the first access cycle (T1) for chip select areas.

DWk[2:0]	Number of inserted wait states
000_B	No wait state inserted
001_B	1 wait state
010_B	2 wait states
011_B	3 wait states
...	...
111_B	7 wait states

Caution
1. The bits marked with 0 must always be 0.
2. To initialize an external memory area after a reset, registers DWCn have to be set. Do not change these registers after initialization. Do not access external devices before initialization is finished.

Note For access to internal memory and peripheral I/O areas, programmable waits are *not* carried out.

Initialization Initialize the registers as follows:

Table 9-24 Initialization of DWCn

Bits	Set to value	Comment
DWC0.DWC0[2:0]	xxx_B	Set as required (refer to below caution)
DWC0.DWC1[2:0]	001_B	1 wait for Internal data flash. **Caution:** These bits must be changed to 001_B (default value is 111_B).
DWC0.DWC2[2:0]	xxx_B	Set as required (refer to below caution)
DWC0.DWC3[2:0]	xxx_B	Set as required (refer to below caution)
DWC1.DWC4[2:0]	xxx_B	Set as required (refer to below caution)
DWC1.DWC5[2:0]	001_B	1 wait for internal FlexRay. **Caution:** These bits must be changed to 001_B (default value is 111_B).
DWC1.DWC6[2:0]	111_B	Default, don't change
DWC1.DWC7[2:0]	000_B	No wait for internal CRC. **Caution:** These bits must be changed to 000_B (default value is 111_B).

Caution If the CPU is operated with 128 MHz (pin PLLSEL = 0) and the memory controller's clock is chosen as BCLK = VBCLK/2 (DVC.DVC0 = 0), at least one data wait state is required for access to the external memory.
Thus minimum values in this case are

- DWC0.DWC0[2:0] ≥ 001B
- DWC0.DWC2[2:0] ≥ 001B
- DWC0.DWC3[2:0] ≥ 001B
- DWC0.DWC4[2:0] ≥ 001B

(4) BCC - Bus cycle control register

The 16-bit BCC register controls for each chip select area

- insertion of an idle state after any *read* access

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

Address FFFF F48A$_H$

Initial Value AAAA$_H$: After system reset, an idle state is inserted.
These registers must be initialized as described below.

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
$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

Note The default value 0 of bits 14, 12, 10, 8, 6, 4, 0 and 0 must not be changed.

Table 9-25 BCC register contents

Bit position	Bit name	Function
15, 13, 11, 9, 7, 5, 3, 1	BCk1	Enables/disables the idle state insertion after any read access for chip select areas. 0: No idle state inserted 1: Idle state inserted after any read access **Note:** For inserting an idle state also after any write access DVC.BCWI has to be set to 1 additionally..

Caution To initialize an external memory area after a reset, register BCC has to be set. Do not change this register after initialization. Do not access external devices before initialization is finished.

Note For access to internal memory and peripheral I/O areas, *no* idle states are inserted.

Initialization Initialize the registers as follows

Table 9-26 Initialization of BCC

Bits	Set to value	Comment
BCC.BC01	x	Set as required
BCC.BC11	0	Internal data flash: • no idle states BCC.BC11 = 0 **Caution:** BCC.BC11 must be changed to 0 (default value is 1).
BCC.BC21	x	Set as required
BCC.BC31	x	Set as required
BCC.BC41	x	Set as required
BCC.BC51	0	Internal FlexRay: • no idle states BCC.BC51 = 0 **Caution:** BCC.BC51 must be changed to 0 (default value is 1).
BCC.BC61	1	Default, don't change
BCC.BC71	0	Internal CRC: • no idle states BCC.BC71 = 0 **Caution:** BCC.BC71 must be changed to 0 (default value is 1).

(5) DVC - Bus clock dividing frequency control register

The 8-bit DVC register controls

- idle state inserted after any write access
- the frequency of the bus clock BCLK

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

Address FFFF F48E$_H$

Initial Value 00$_H$.

7	6	5	4	3	2	1	0
BCWI	0	0	0	0	0	0	DVC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9-27 DVC register contents

Bit position	Bit name	Function
7	BCWI	Enables/disables the idle state insertion after any write access (only for chip select areas with BCC.BCk1 = 1, i.e. idle state insertion after read access must be enabled as well). 0: No idle state inserted 1: Idle state inserted after any write access
0	DVC0	Specifies the frequency of the bus clock BCLK. 0: BCLK = f_{xx}/2 1: BCLK = f_{xx}/4

Caution

1. The bits marked with 0 must always be 0.
2. To initialize an external memory area after a reset, register DVC has to be set. Do not change this register after initialization. Do not access external devices before initialization is finished.

Note For access to internal memory, *no* idle states are inserted.

9.3 Configuration of Memory Access

The microcontroller device supports interfacing with various memory devices. Therefore, the endian format, wait functions and idle state insertions can be configured.

9.3.1 Endian format

The endian format is specified with the endian configuration register (BEC). It defines the byte order in which word data is stored.

"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:

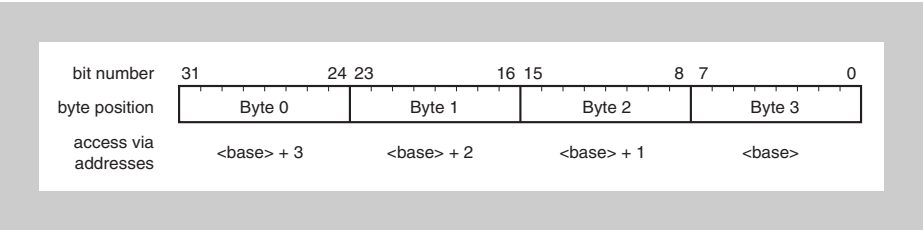


Figure 9-4 Big endian addresses within a word

"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:

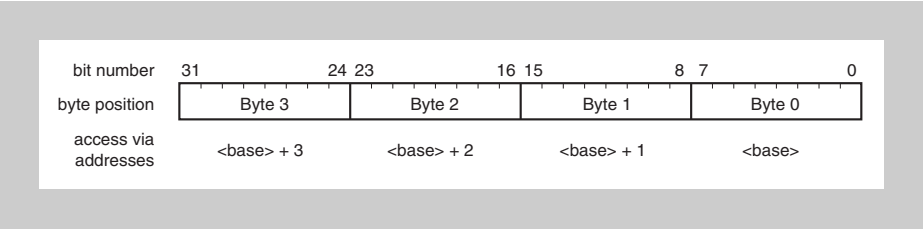


Figure 9-5 Little endian addresses within a word

9.3.2 Wait functions

Several wait functions are supported:

(1) Address setup wait

The microcontroller device allows insertion of an address setup wait state before the first access cycle (T1 state).

The address setup wait state can be enabled by AWC.ASWk = 1 individually for each chip select area.

(2) Address hold wait

The microcontroller device allows insertion of an address hold wait state after the first access cycle (T1 state).

The address hold wait state can be enabled by AWC.AHWk = 1 individually for each chip select area

(3) Programmable data wait function

With the purpose of realizing easy interfacing with low-speed memory or with I/Os, it is possible to insert up to seven data wait states after the first access cycle (T1 state).

The number of wait states can be specified by data wait control registers DWC0 and DWC1.

(4) Idle state insertion function

To facilitate interfacing with low-speed memory devices, an idle state (TI) can be inserted between two bus cycle, that means after the T2 state. Idle states are inserted to meet the data output flow delay on memory read or write accesses for each chip select space. The next bus cycle is started after the idle state.

The idle state after read access is specified by BCC.BCk1 = 1.
The idle state after write access is specified by DVC.BCWI = 1.

9.3.3 External wait function

Each read or write operation takes at least two cycles (T1 and T2). To stretch the access cycle for accessing slow external devices, any number of wait states (TW) can be inserted under external control of the $\overline{\text{WAIT}}$ signal.

The $\overline{\text{WAIT}}$ signal can be set asynchronously from the system clock. The $\overline{\text{WAIT}}$ signal is sampled at the rising edge of the clock in the T1 and TW states. Depending on the level of the $\overline{\text{WAIT}}$ signal at sampling timing, a wait state is inserted or not.

(1) Relationship between programmable wait and external wait

If both programmable wait and external wait ($\overline{\text{WAIT}}$) are applied, an OR relation gives the resulting number of wait cycles. *Figure 9-6* shows that as long as any of the two waits is active, a wait cycle will be performed.

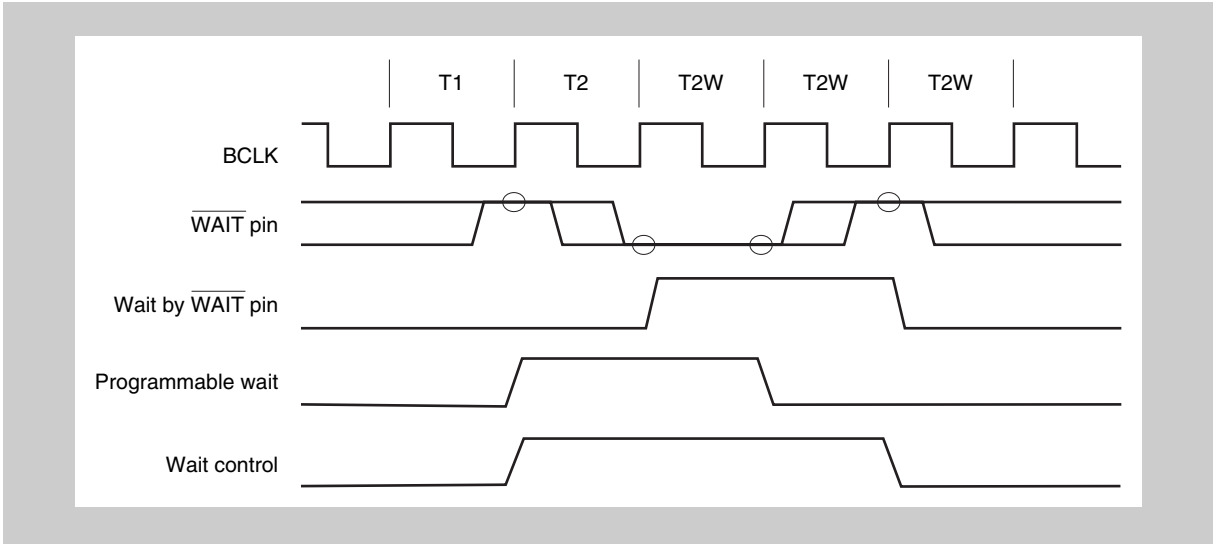


Figure 9-6 Example of wait insertion

Note The circles indicate the sampling timing.

9.4 External Devices Interface Timing

This section presents examples of write and read operations. The states are abbreviated as:

- T1 state: The state the address is issued
- T2 state: The state for the data access.
- T2W state: Wait state that is inserted according to the DWC0 and DWC1 register settings and according to the $\overline{\text{WAIT}}$ input.
- T1S state: Address setup wait state that is inserted according to the AWC.ASWk settings.
- T1H state: Address hold wait state that is inserted according to the AWC.AHWk settings.
- TI state: Idle state that is inserted according to the BCC register settings.

9.4.1 SRAM, external ROM, external I/O access

(1) Read (data wait inserted)

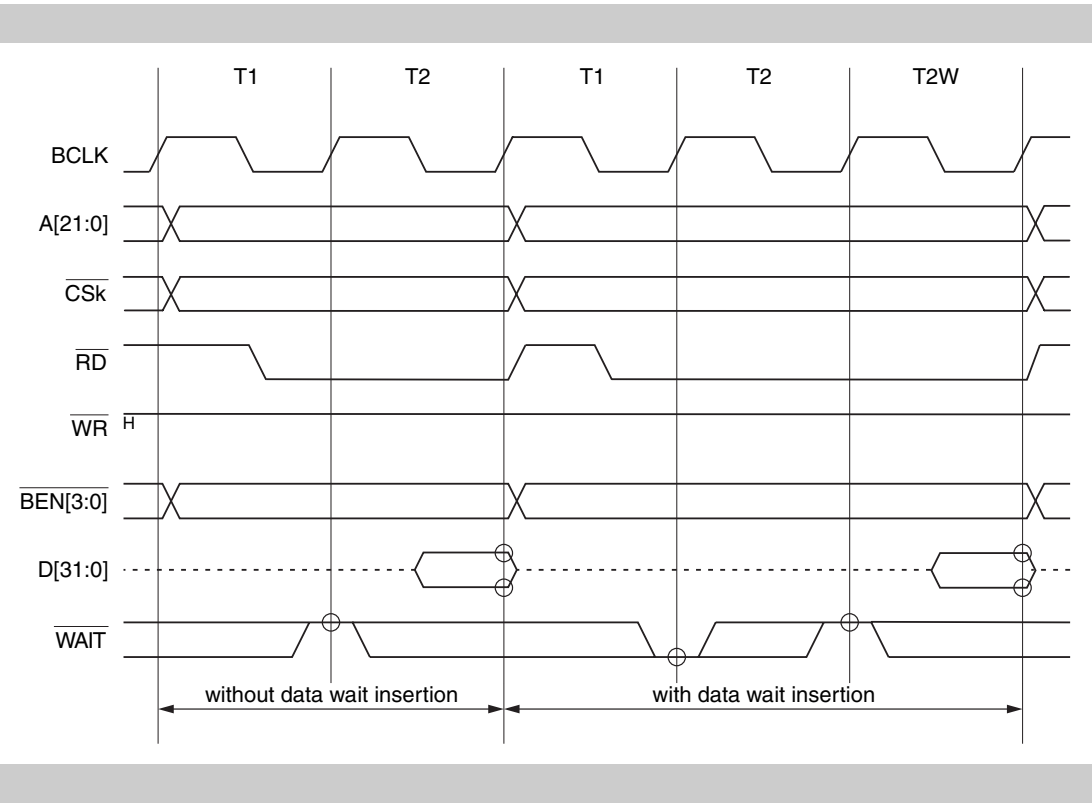


Figure 9-7 Read timing (data wait inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(2) Read (idle state inserted)

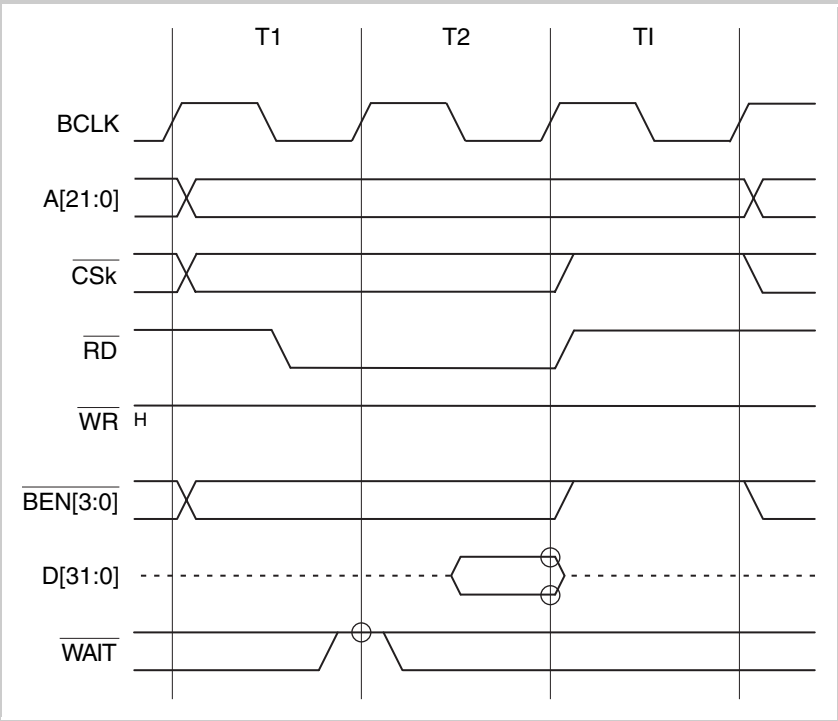


Figure 9-8 Read timing (idle state inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(3) Read (data wait, idle state inserted)

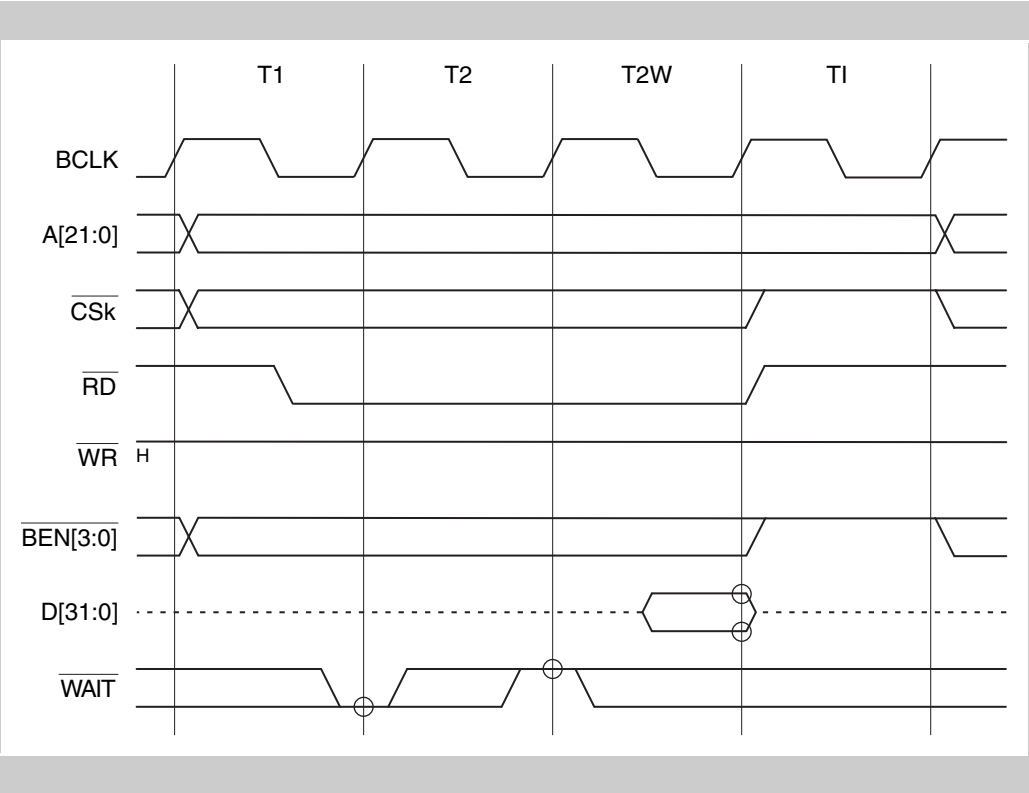


Figure 9-9 Read timing (data wait, idle state inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(4) Read (address setup wait and address hold wait state inserted)

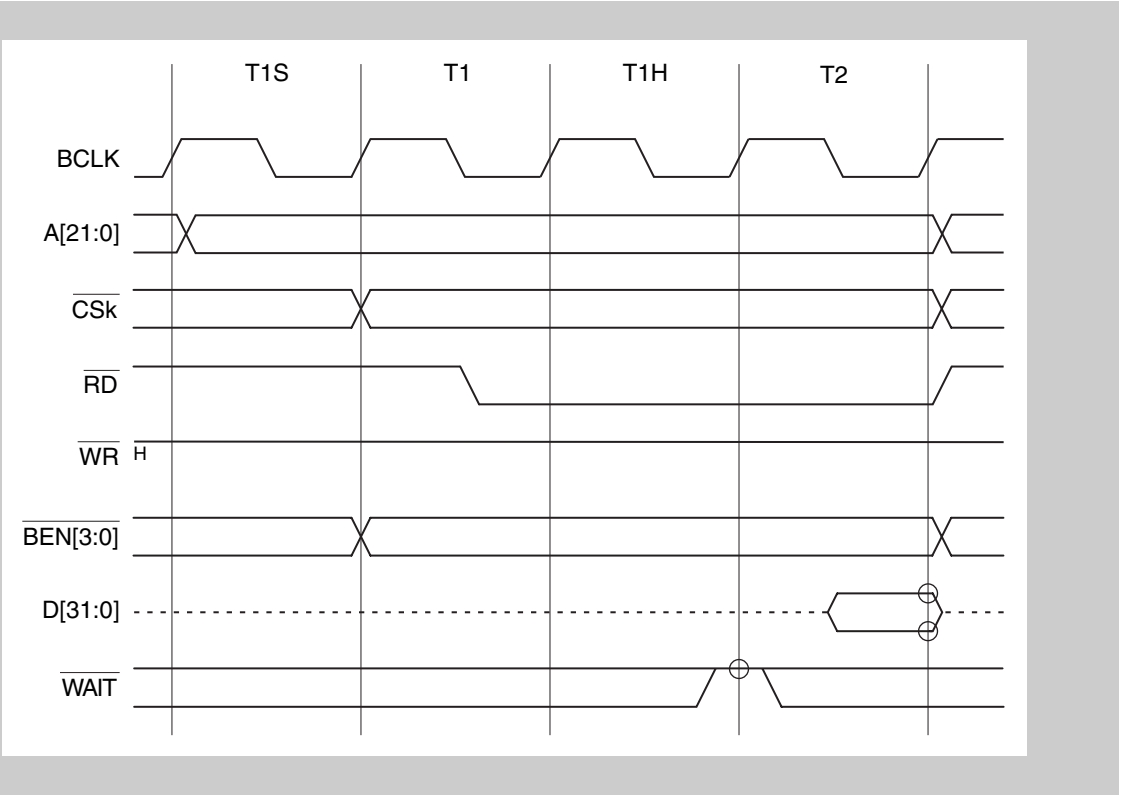


Figure 9-10 Read timing (address setup wait and address hold wait state inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(5) Write (data wait inserted)

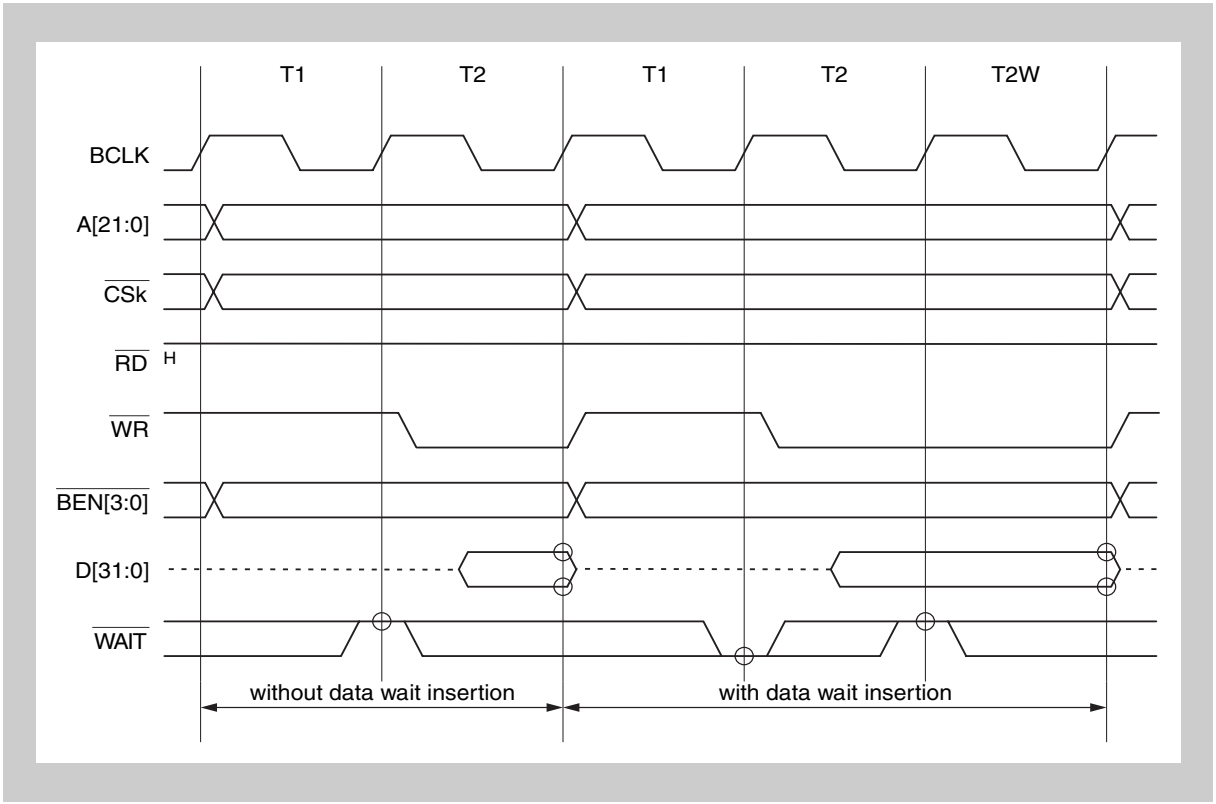


Figure 9-11 Write timing (data wait inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(6) Write (idle state inserted)

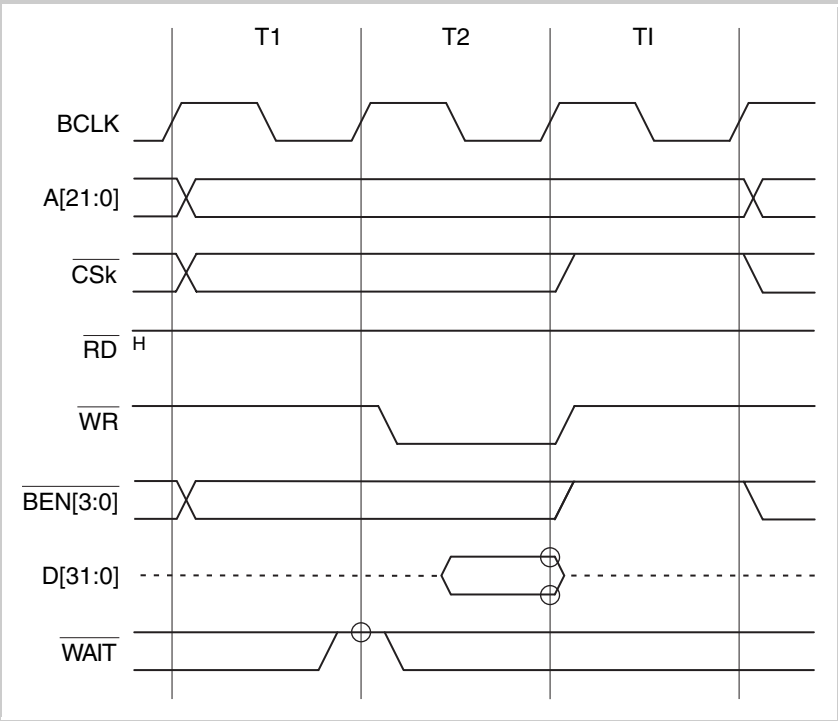


Figure 9-12 Write timing (idle state inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(7) Write (data wait, idle state inserted)

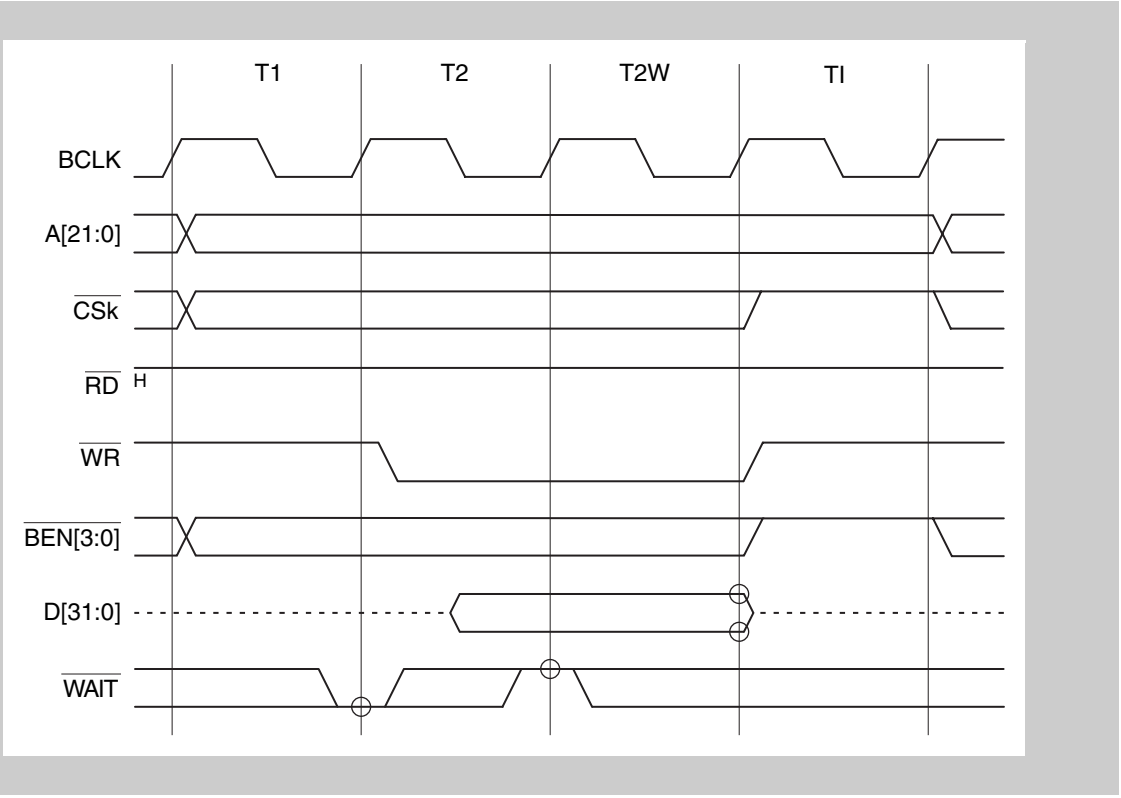


Figure 9-13 Write Timing (data wait, idle state inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

(8) Write (address setup wait and address hold wait state inserted)

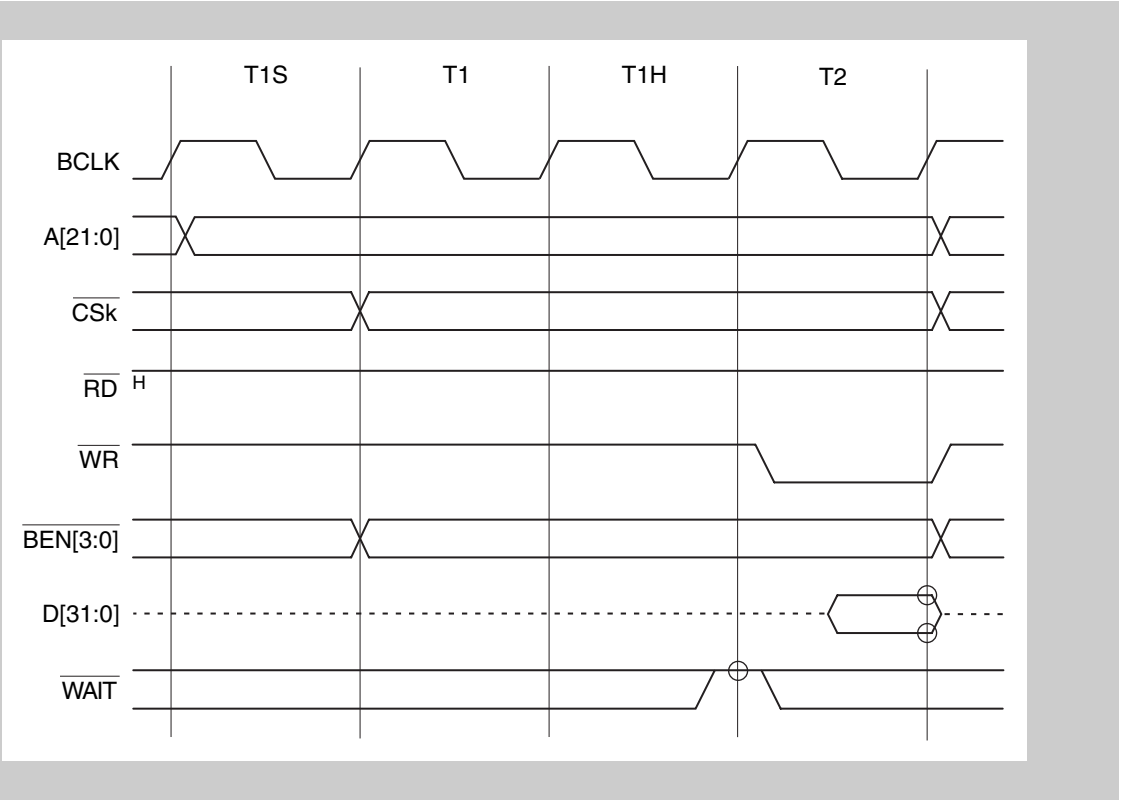


Figure 9-14 Write timing (address setup wait and address hold wait state inserted)

Note 1. The circle indicates the sampling timing.

2. The dashed line indicates the high impedance state respectively undefined data.

9.5 Data Access Order

9.5.1 Access to 8-bit data busses

This section shows how byte, half word and word accesses are performed for an 8-bit data bus.

(1) Byte access (8 bits)

(a) Little endian

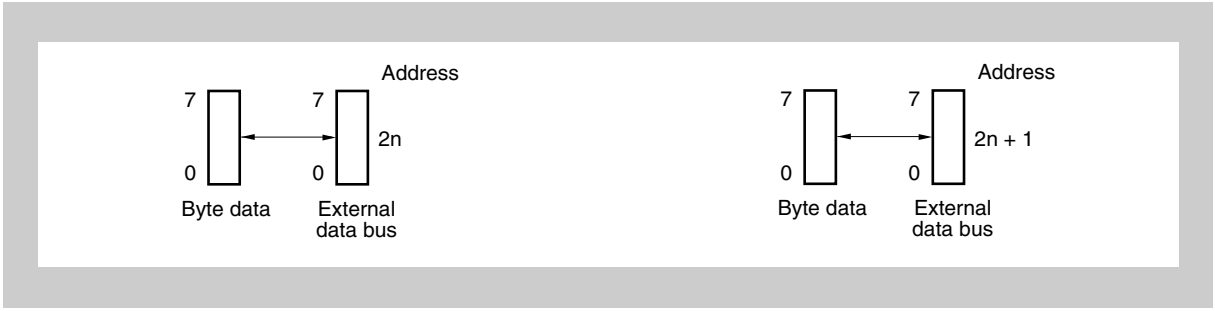


**Figure 9-15 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(b) Big endian

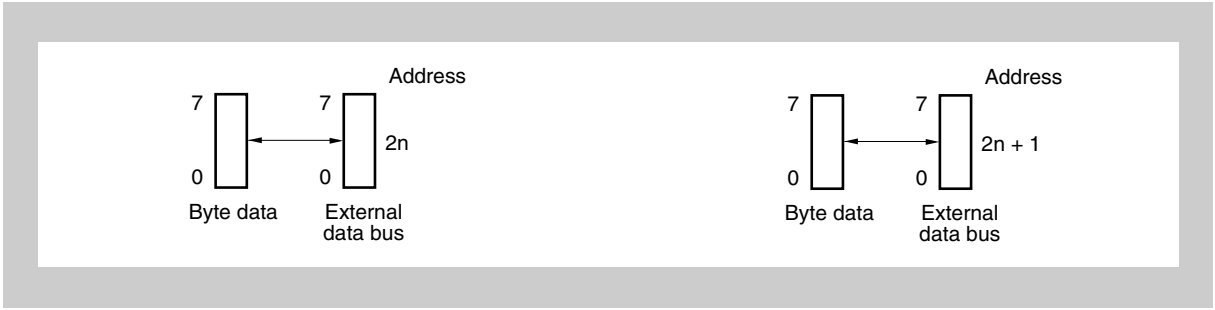


**Figure 9-16 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(2) Halfword Access (16 bits)

(a) Little endian

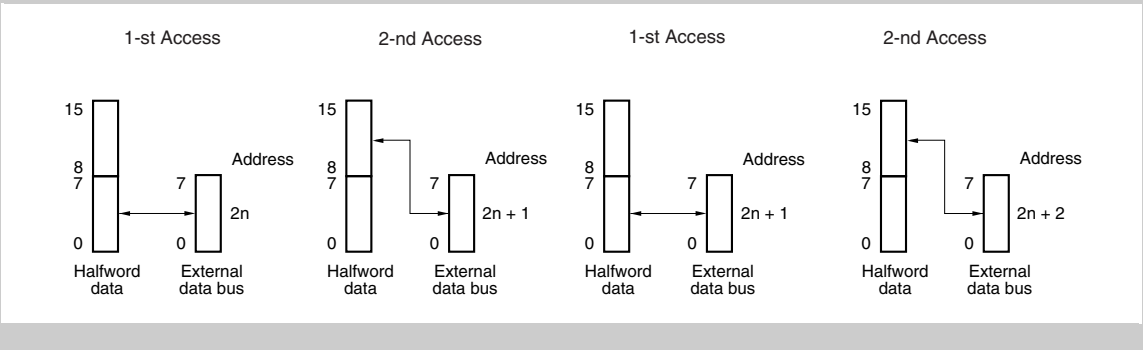


**Figure 9-17 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(b) Big endian

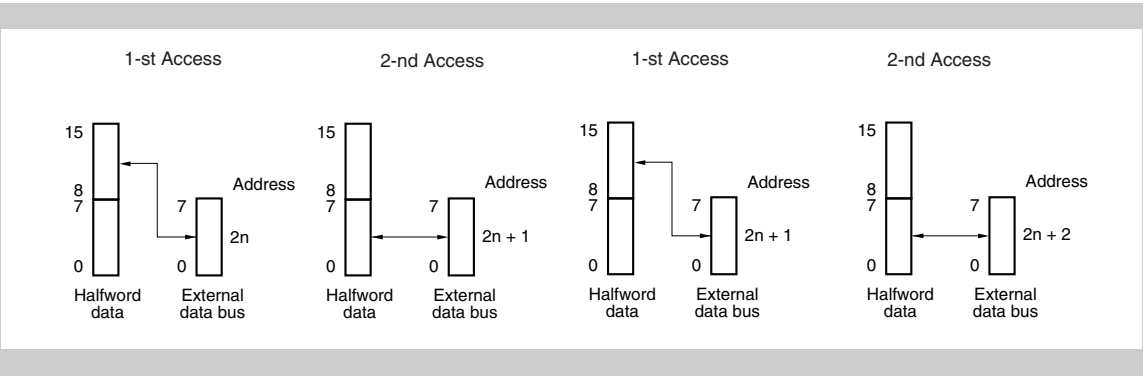


**Figure 9-18 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(3) Word Access (32 bits)

(a) Little endian

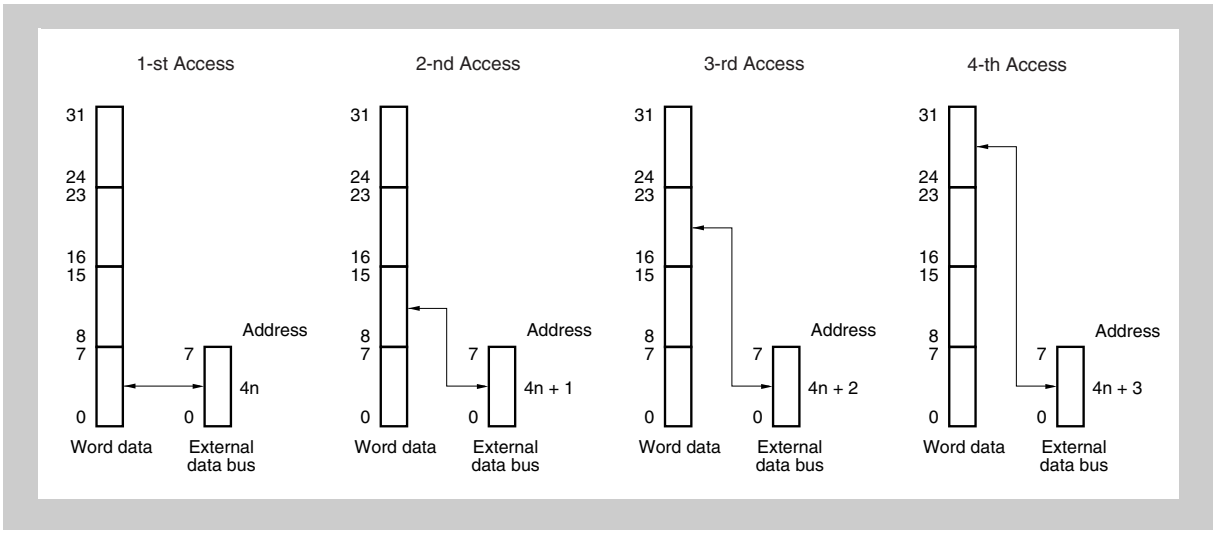


Figure 9-19 Access to address 4n

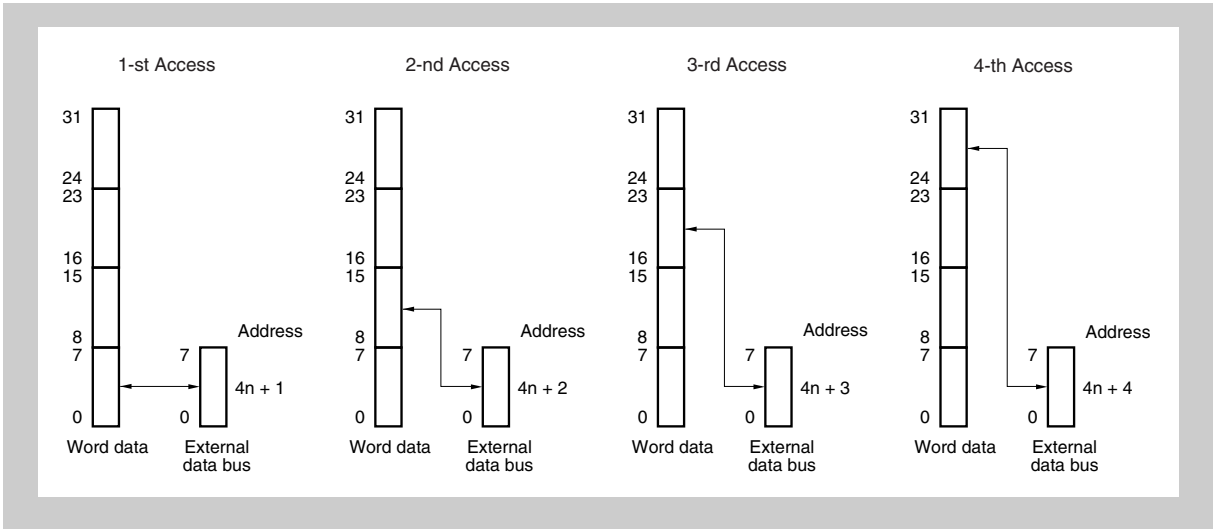


Figure 9-20 Access to address 4n + 1

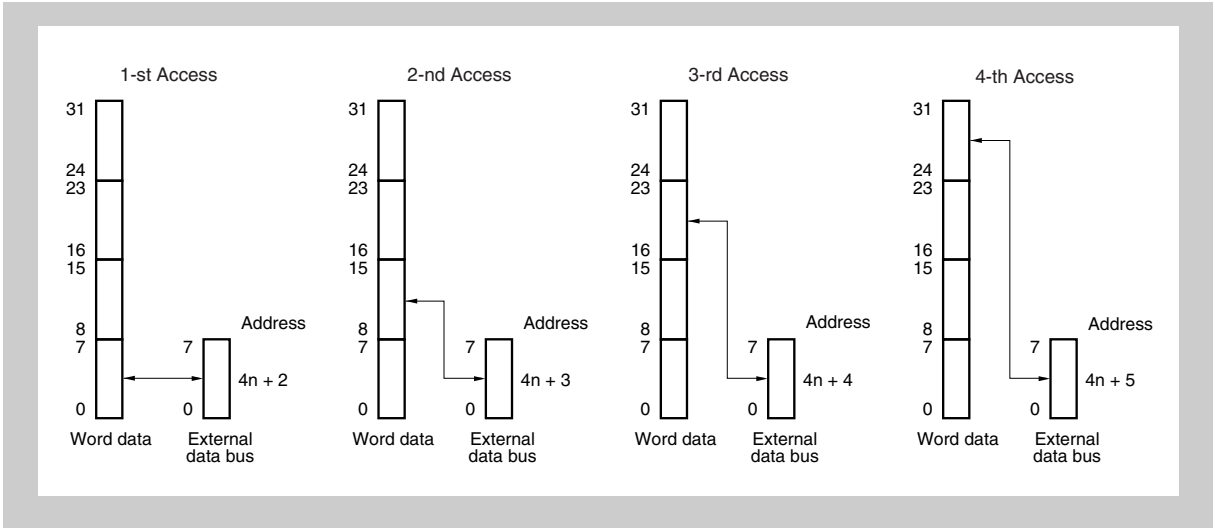


Figure 9-21 Access to address 4n + 2

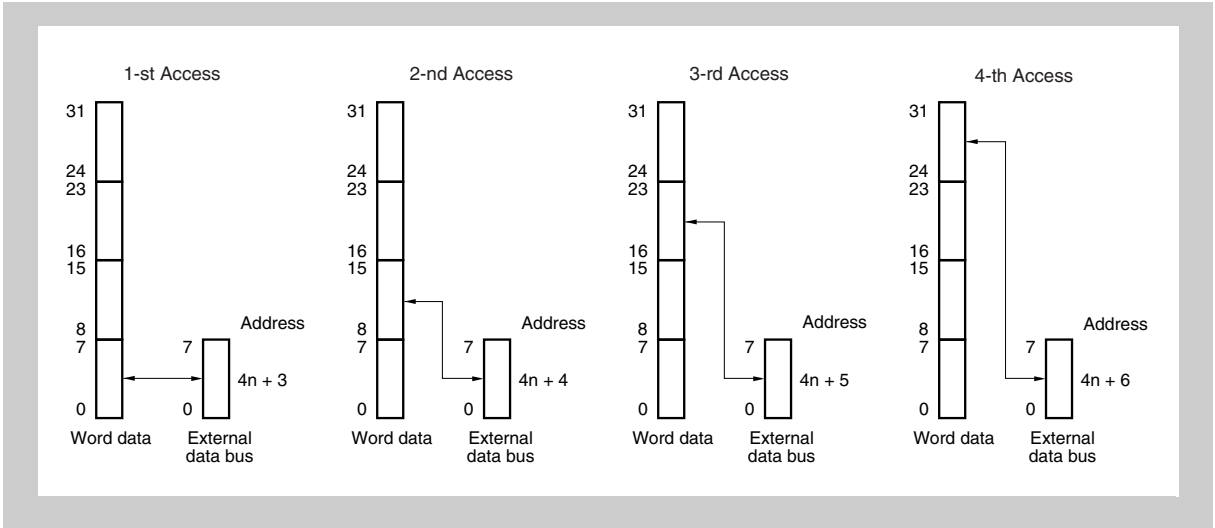


Figure 9-22 Access to address 4n + 3

(b) Big endian

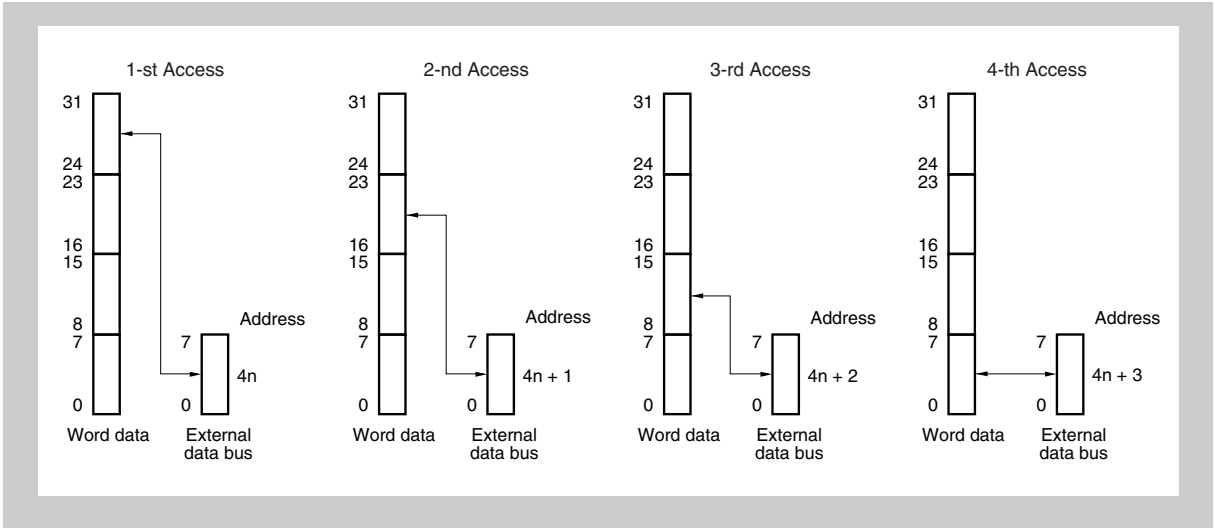


Figure 9-23 Access to address 4n

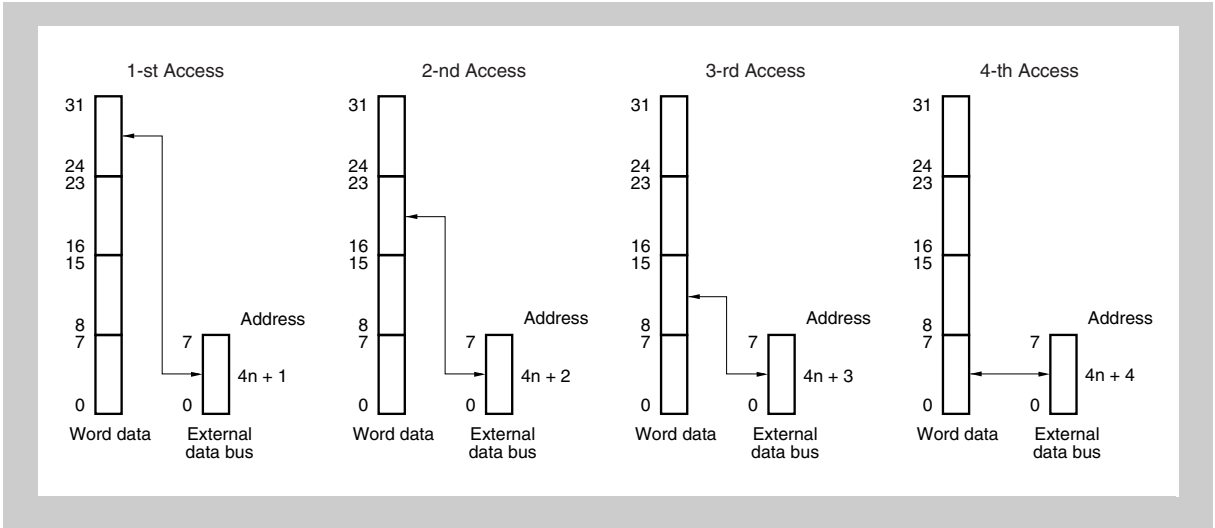


Figure 9-24 Access to address 4n + 1

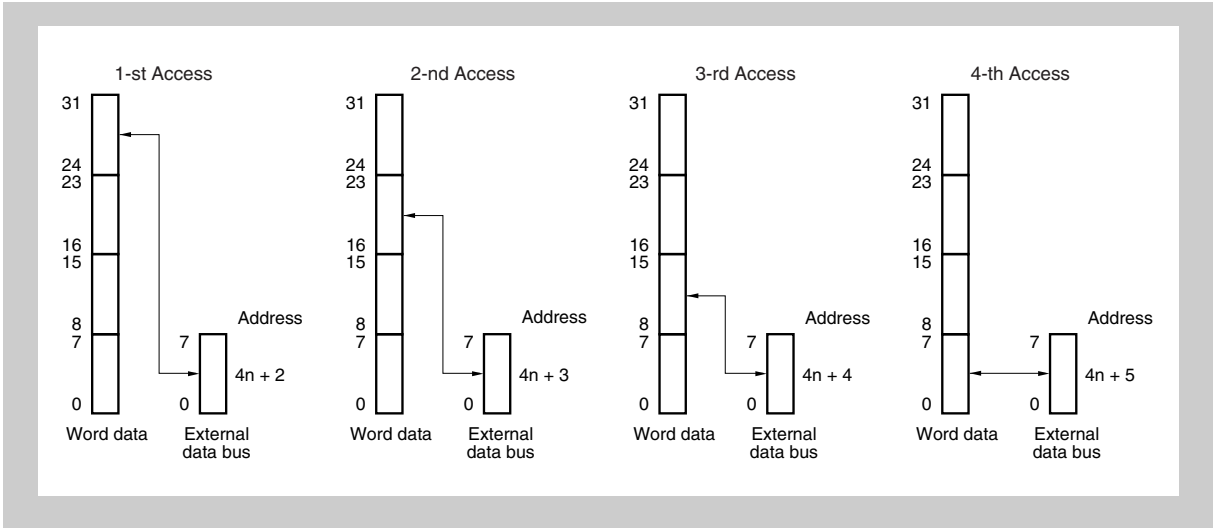


Figure 9-25 Access to address 4n + 2

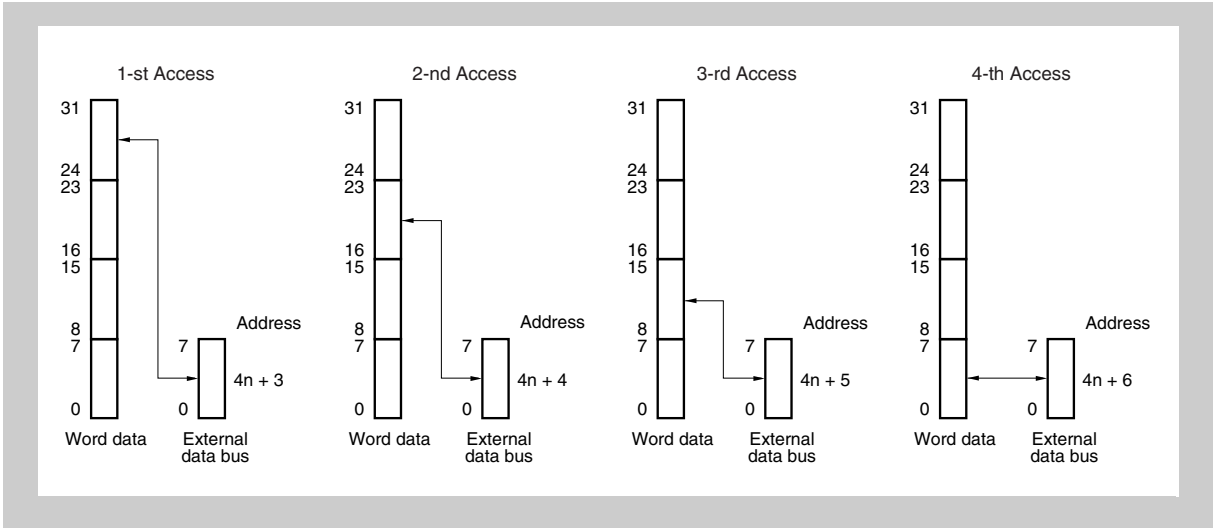


Figure 9-26 Access to address 4n + 3

9.5.2 Access to 16-bit data busses

This section shows how byte, half word and word accesses are performed for a 16 bit data bus.

Access all data in order starting from the lower order side.

(1) Byte access (8 bits)

(a) Little endian

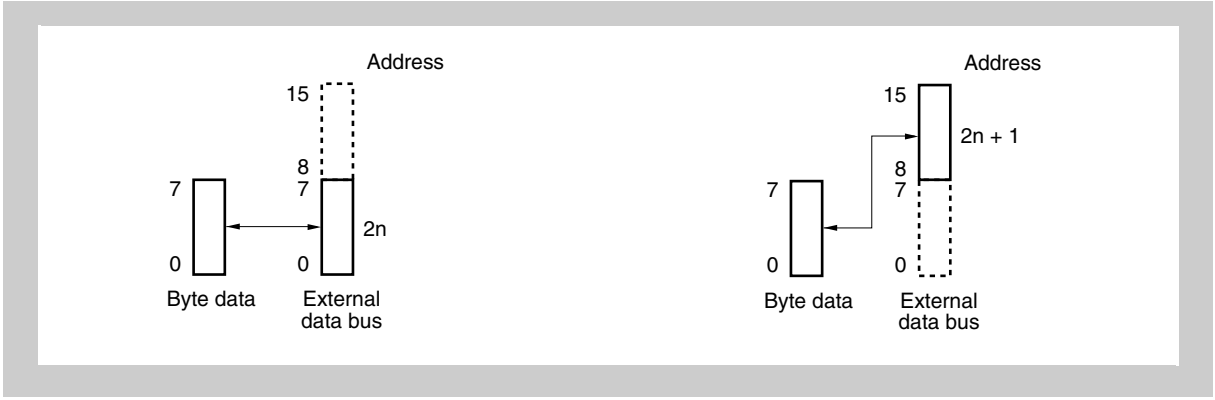


**Figure 9-27 Left: Access to even address (2n)
Right: Access odd address (2n + 1)**

(b) Big endian

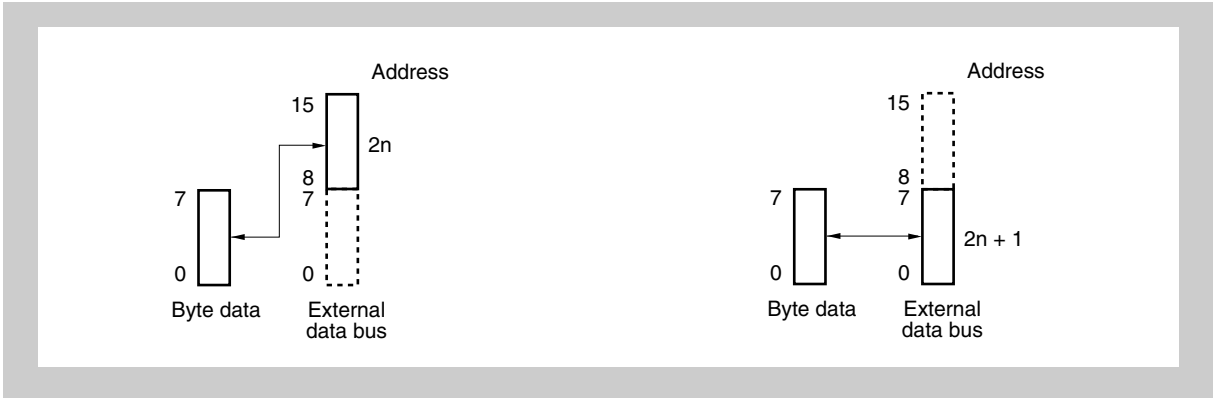


**Figure 9-28 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(2) Halfword Access (16 bits)

(a) Little endian

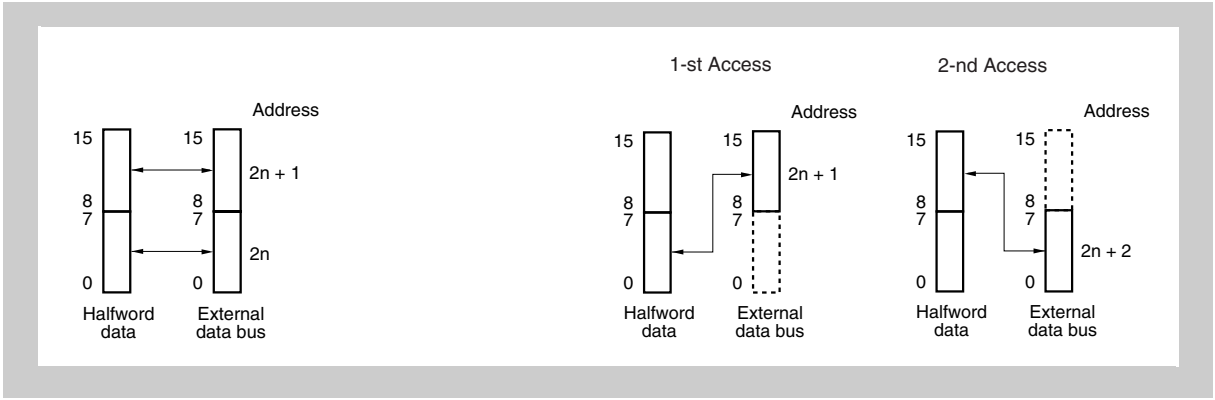


**Figure 9-29 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(b) Big endian

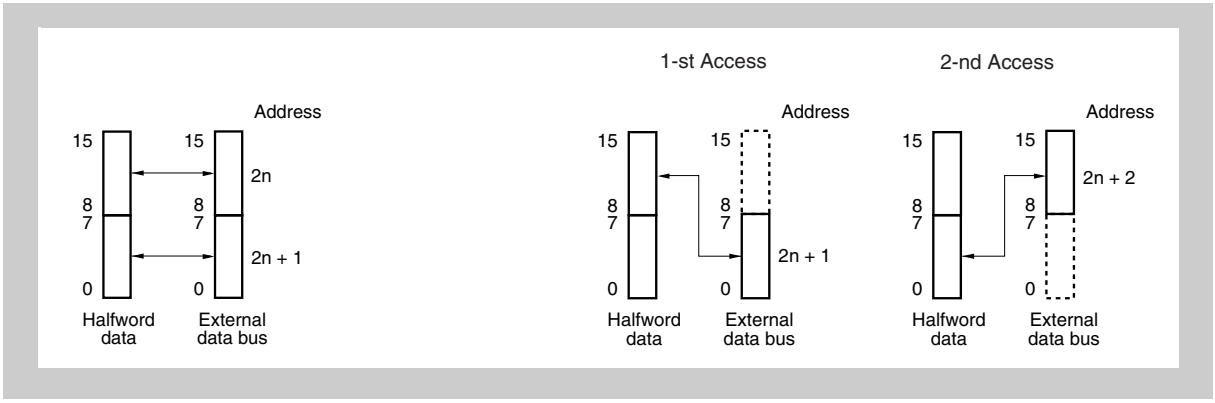


**Figure 9-30 Left: Access to even address (2n)
Right: Access to odd address (2n + 1)**

(3) Word Access (32 bits)

(a) Little endian

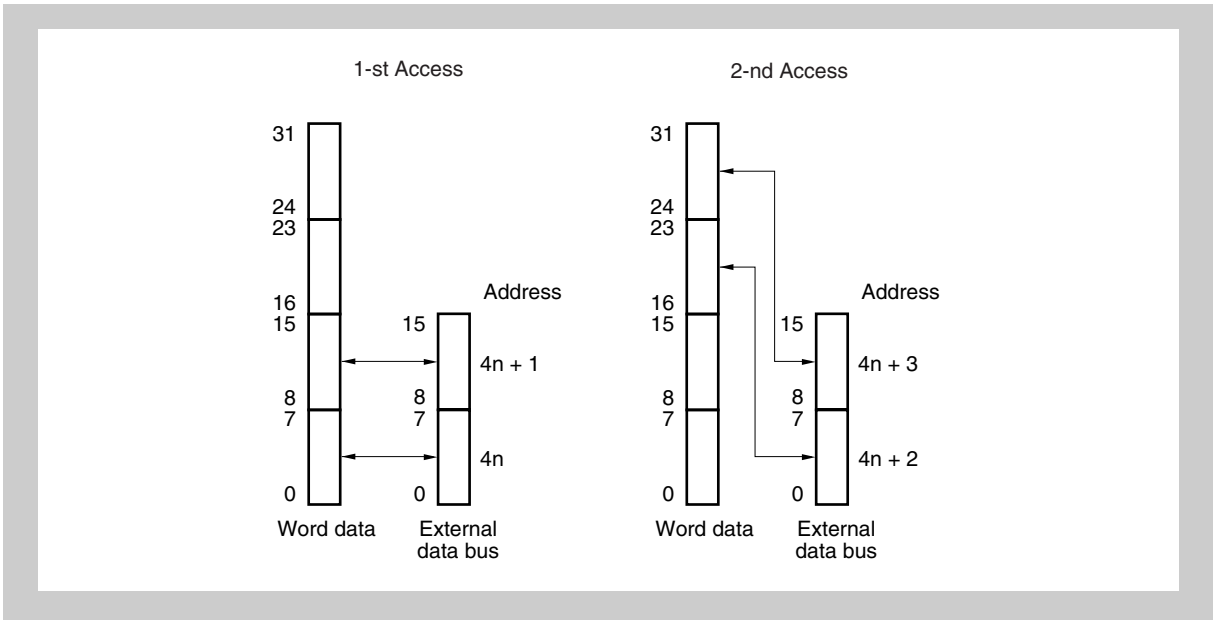


Figure 9-31 Access to address 4n

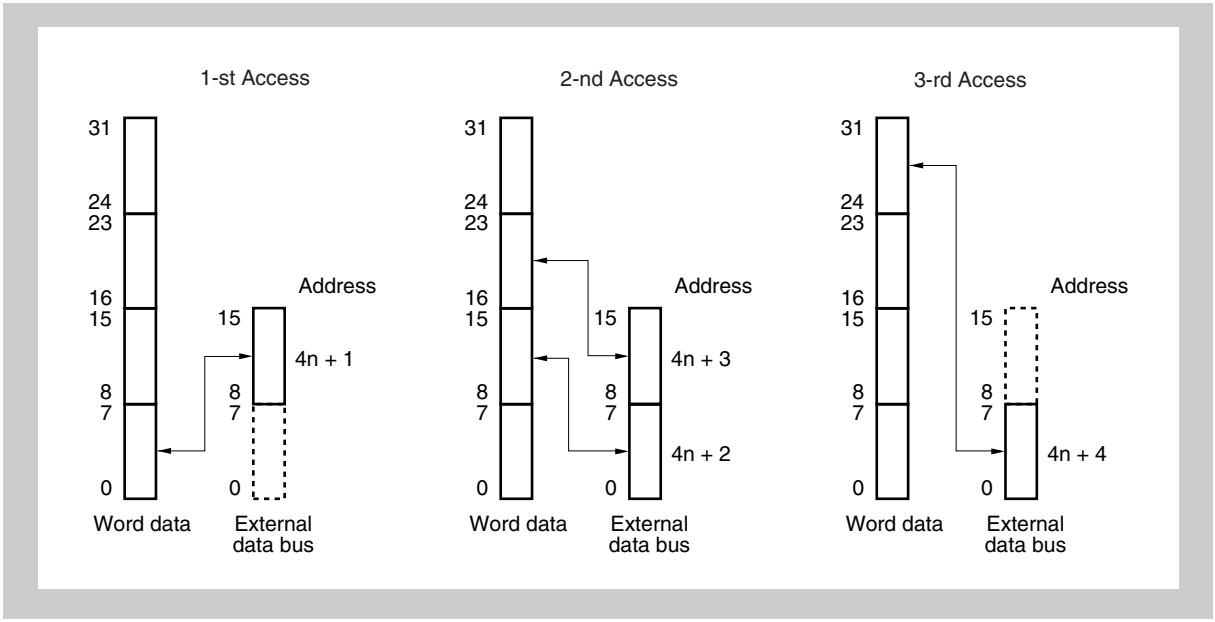


Figure 9-32 Access to address 4n + 1

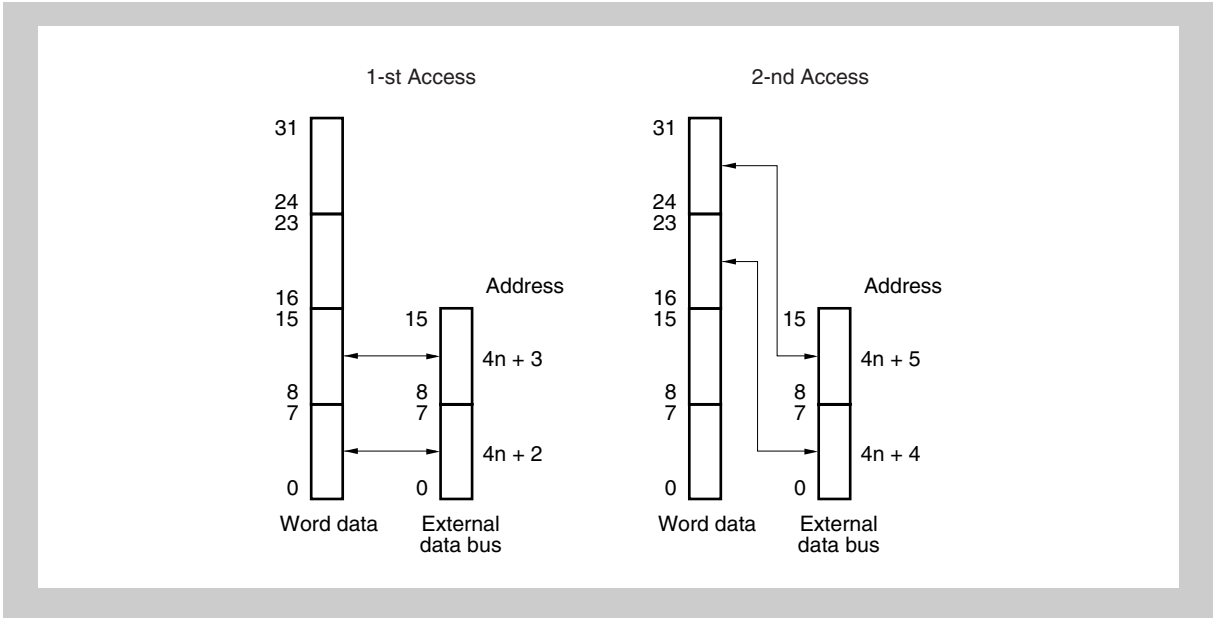


Figure 9-33 Access to address 4n + 2

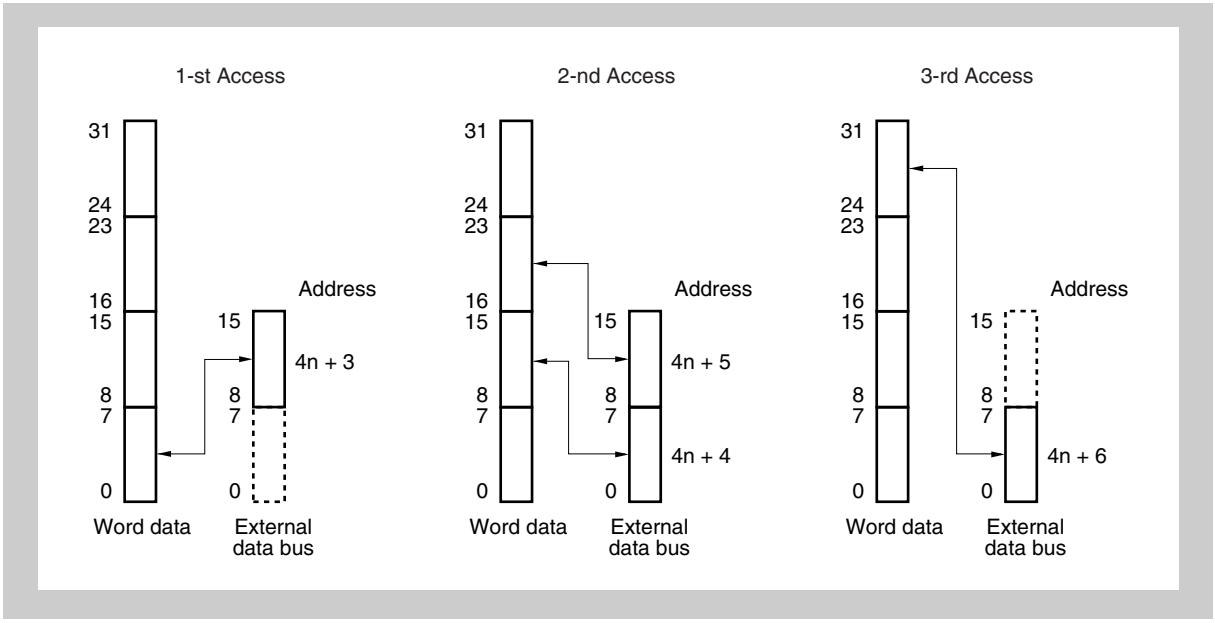


Figure 9-34 Access to address 4n + 3

(b) Big endian

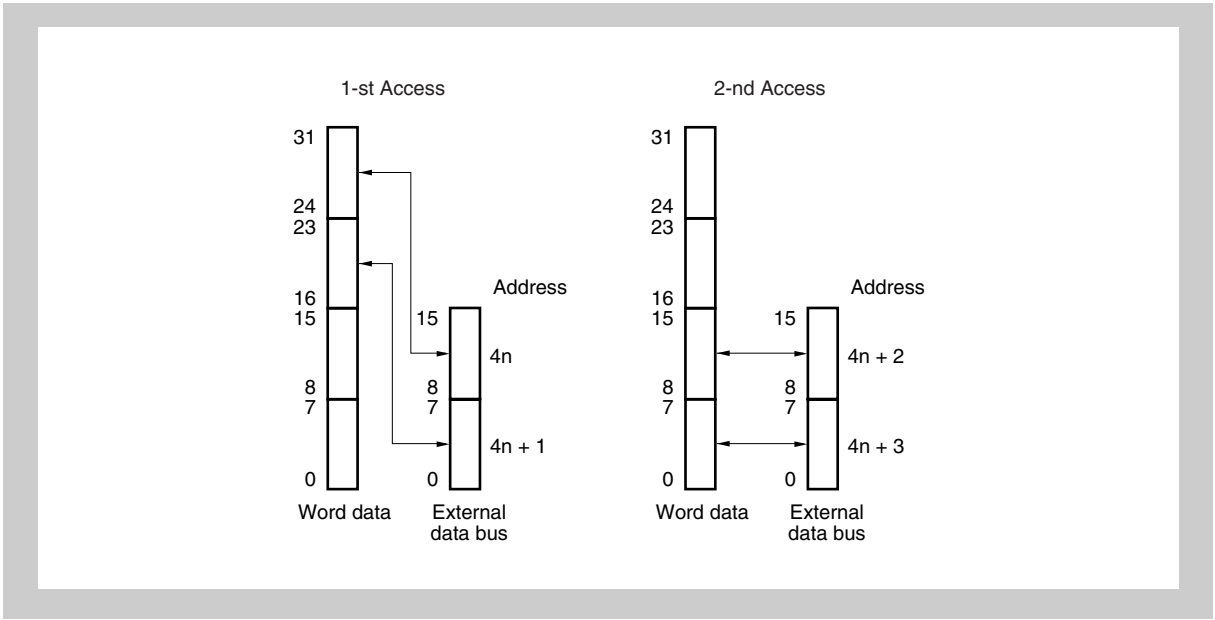


Figure 9-35 Access to address 4n

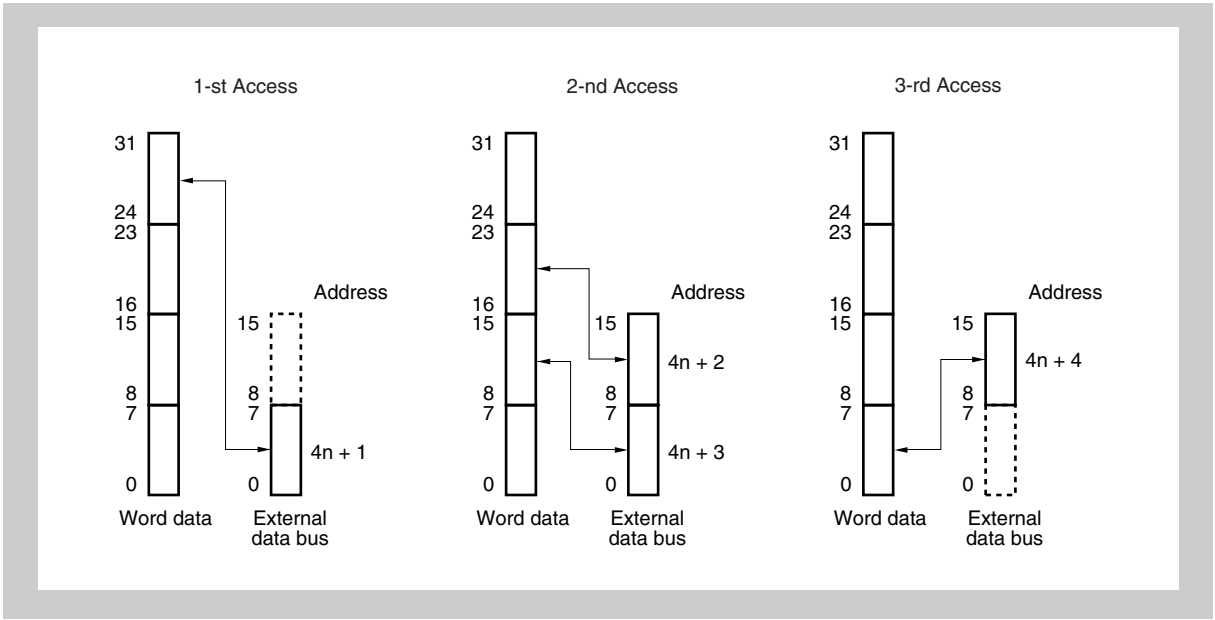


Figure 9-36 Access to address 4n + 1

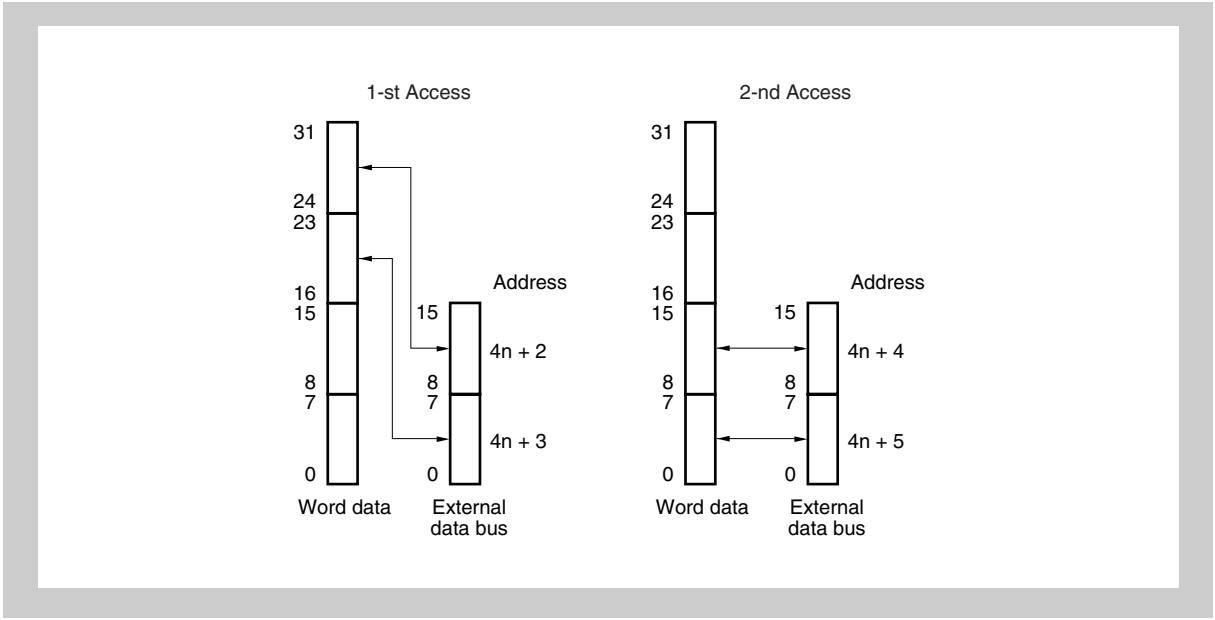


Figure 9-37 Access to address 4n + 2

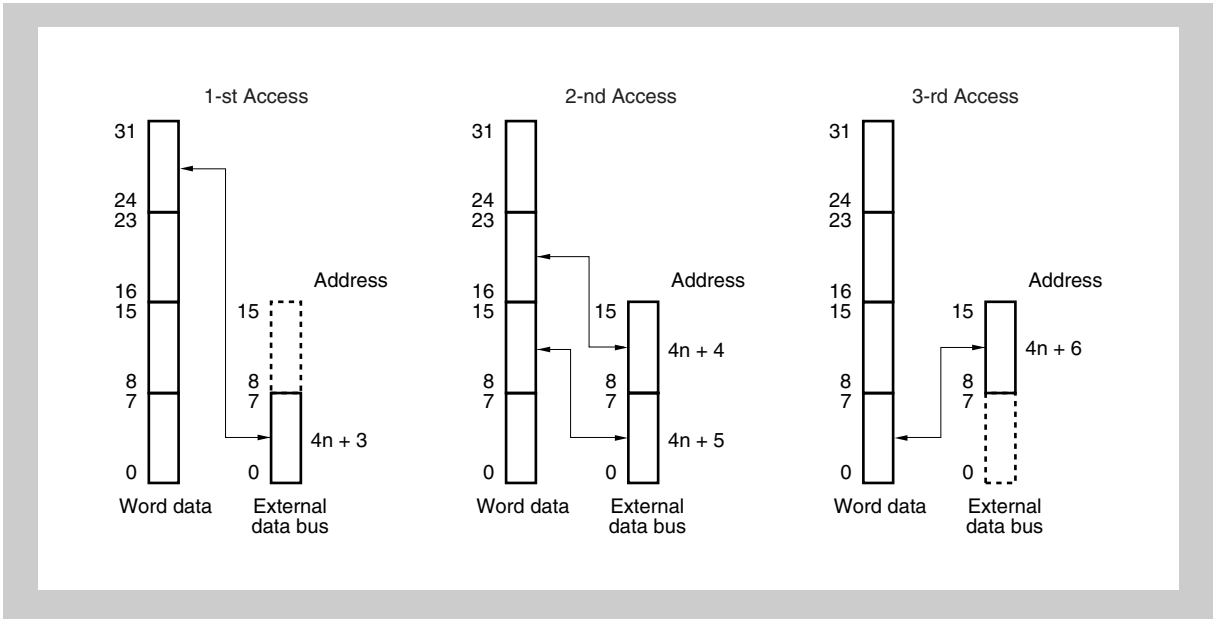


Figure 9-38 Access to address 4n + 3

Chapter 10 DMA Functions (DMA Controller)

The microcontroller includes a Direct Memory Access Controller (DMAC) that executes and controls DMA transfers.

Note Throughout this chapter, the individual channels of the DMA Controller are identified by “n” (n = 0 to 7)

10.1 Overview

The DMAC controls data transfer between internal RAM and I/O, based on DMA requests issued by the on-chip peripheral I/O.

Features summary The DMAC has the following features:

- 8 independently configurable channels
- Programmable access wait and signal timing
- Maximum DMA transfer count: 256 (2^8)

Caution If the RAM error correction function is disabled, DMA operation is prohibited. Thus disable all DMA channels in that case.

10.1.1 Principle of operation

A DMA transfer essentially transfers a predefined number of data units between the internal RAM (iRAM) and peripheral I/O registers. The CPU initiates the transfer and the transfer itself is performed by the DMAC.

DMA channels Each DMA channel has a fixed transfer direction to transfer data between the internal RAM and a peripheral I/O register:

- DMA channels 0, 1, 4, 5: I/O → iRAM
- DMA channels 2, 3, 6, 7: iRAM → I/O

DMA priorities The DMA channel priorities are fixed as follows:
DMA channel 0 > DMA channel 1 > DMA channel 2 > … > DMA channel 7

Transfer data units The size of a data units to be transferred is either 8 bit or 16 bit:

- DMA channels 0 to 3: 16 bit
- DMA channels 4 to 7: 8 bit or 16 bit selectable

Transfer addresses For a DMA transfer, the following addresses have to be defined:

- Source address: address of data to be copied

The source address depends on the DMA channel:

– DMA channels n = 0, 1: implicitly defined fixed ADC registers

– DMA channels n = 4, 5: implicitly defined fixed CSIBn, CSIEn, UARTCn data receive register

– DMA channels n = 2, 3, 6, 7: iRAM start address, defined by MARn registers

- Destination address: address to stored the data to

  The destination address depends on the DMA channel:

  – DMA channels n = 0, 1, 4, 5: iRAM start address, defined by MARn register

  – DMA channels n = 2, 3: Timer S start register address, defined by SARn

  – DMA channels n = 6, 7: implicitly defined fixed CSIBn, CSIEn, UARTCn data transmit registers

Depending on the transfer data unit, source start addresses SARn and destination start addresses MARn are automatically incremented after each data transfer:

- 8-bit data: address increment by 1
- 16-bit data: address increment by 2

Transfer start/stop

The DMA transfer is initiated by defined interrupts of the on-chip peripheral I/Os. These interrupts are called DMA triggers.

To start a DMA transfer, the following conditions have to be fulfilled:

- DMA transfer is enabled (DMAMCL.DEn = 1).
- The number of transfer actions that has to be performed (transfer count) is written to register DTRCn.
- The transfer status is set to “idle” (DMASL.DMASn = 0).
- DMA trigger is generated.

When the defined number of transfer actions is completed, a DMA completion interrupt (INTDMAn) is generated and the DMA transfer is stopped.

Transfer status

The transfer status can be monitored:

- Register DMASL displays the transfer status (idle/in progress/completed).
- Register DTCRn displays the current transfer count (it is decreased with every transfer action).

10.1.2 Forcible termination of DMA transfer

A once started DMA transfer can be forcible terminated by clearing bit DMAMCL.DEn = 0.

If the DMAMCL.DEn bit is cleared during a DMA transfer action, the current transfer action is finished before the DMA transfer is stopped (see *“Example of forcible termination of DMA transfer” on page 300*).

10.1.3 DMA interrupt function

The DMA triggers are interrupts of

- A/D Converters
- serial interfaces
- Timer S

These interrupts are shared with their corresponding DMA transfer completion interrupts (INTDMAn). If a DMA channel is enabled, the peripheral I/O interrupt corresponding to the DMA trigger can no longer generate an interrupt request. Instead, the DMA transfer completion interrupt can generate an interrupt with the corresponding interrupt handler address.

The DMA triggers of Timers S (INTTSnOD and INTTSnCD0) are *not* shared with their DMA transfer completion interrupts (INTDMAn). These DMA completion interrupts have dedicated entries in the interrupt source list (refer to *"Interrupt Controller (INTC)" on page 125*).

Table 10-1 shows the relations between DMA triggers and DMA completion interrupts.

Table 10-1 Relations between DMA triggers and DMA completion interrupts

DMA channel	DMA trigger	DMA completion interrupt		
		Name	Entry in interrupt list[a]	Handler address
0	ADDMARQ0	INTDMA0	INTAD0	0000 0790$_H$
1	ADDMARQ1	INTDMA1	INTAD1	0000 07A0$_H$
2	INTTS0CD0 or INTTS0OD	INTDMA2	INTDMA2	0000 07B0$_H$
3	INTTS1CD0 or INTTS1OD	INTDMA3	INTDMA3	0000 07C0$_H$
4, 5	INTCE0C	INTDMA4, INTDMA5	INTCE0C	0000 0720$_H$
	INTCE1C		INTCE1C	0000 0740$_H$
	INTCB0R		INTCB0R	0000 06C0$_H$
	INTCB1R		INTCB1R	0000 06F0$_H$
	INTUC0R		INTUC0R	0000 00E0$_H$
	INTUC1R		INTUC1R	0000 00F0$_H$
	INTUC2R		INTUC2R	0000 06C0$_H$
6, 7	INTCE0C	INTDMA6, INTDMA7	INTCE0C	0000 0720$_H$
	INTCE1C		INTCE1C	0000 0740$_H$
	INTCB0T		INTCB0T	0000 06B0$_H$
	INTCB1T		INTCIB1T	0000 06E0$_H$
	INTUC0T		INTUC0T	0000 0760$_H$
	INTUC1T		INTUC1T	0000 0780$_H$
	INTUC2T		INTUC2T	0000 06B0$_H$

a) The interrupt/exception list is given in *"Interrupt Controller (INTC)" on page 125*

10.2 Registers

The DMA Controller is controlled by means of the following registers:

Table 10-2 DMA Controller register overview

Register name	Shortcut	Address
DMA wait control registers	DMAWC0	FFFF $FE00_H$
	DMAWC1	FFFF $FE02_H$
DMA transfer SFR start address registers	SAR2	<base> + 24_H
	SAR3	<base> + 26_H
DMA transfer memory start address registers	MARn (n = 0 to 7)	<base> + n x 2_H
DMA trigger factor registers	DTFR4	<base> + 88_H
	DTFR5	<base> + $8A_H$
	DTFR6	<base> + $8C_H$
	DTFR7	<base> + $8E_H$
DMA transfer count registers	DTCRn (n = 0 to 7)	<base> + 40_H + n x 2_H
DMA status register	DMASL	<base> + 62_H
DMA mode control register	DMAMCL	<base> + 60_H
DMA data size control register	DMADSCL	<base> + 64_H

Table 10-3 DMA Controller register base address

Unit	Base address
DMA Controller	FFFF $F300_H$

(1) DMAWCm - DMA wait control registers

The 8-bit DMAWCm registers control the internal bus timing for DMA transfers (m = 0, 1).

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

Address DMAWC0: FFFF $FE00_H$
DMAWC1: FFFF $FE02_H$

Initial Value DMAWC0: 37_H
DMAWC1: 07_H

DMAWC0 The DMAWC0 register must have the same setting as the internal peripheral function wait control register VSWC:

DMAWC0 = VSWC.

Note that the contents of the VSWC register depend on the system clock (see *"Bus and Memory Control (BCU, MEMC)" on page 225* for details).

DMAWC1 The DMAWC1 register must be set as follows:

DMAWC1 = 00_H.

Caution The DMAWCm registers must be set up correctly before the first DMA transfer is started. Do not change these registers afterwards.

(2) SARn - DMA transfer SFR start address registers

The 8-bit SARn register specifies the start address of the TMS register for which the DMA transfer is started on the corresponding DMA channel n (n = 2, 3).

This register is only valid for DMA transfers to timers S.

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

Address SAR2: <base> + 24_H
SAR3: <base> + 26_H

Initial Value Undefined.

7	6	5	4	3	2	1	0
0	0	0	0	SARn3	SARn2	SARn1	SARn0
R	R	R	R	R/W	R/W	R/W	R/W

Writing to SARn.bit[7:4] is ignored.

Table 10-4 SARn register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>SARn[3:0]
(n = 2, 3)</td><td>Sets the start address of the TMS register for which the DMA transfer is started on the corresponding DMA channel n (n = 2, 3).
<table>
<tr><th rowspan="3">SARn3</th><th rowspan="3">SARn2</th><th rowspan="3">SARn1</th><th rowspan="3">SARn0</th><th colspan="4">DMA transfer start address of TMS reload register</th></tr>
<tr><th colspan="2">n = 2</th><th colspan="2">n = 3</th></tr>
<tr><th>Register</th><th>Address</th><th>Register</th><th>Address</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>TS0OPT1A</td><td>FFFF FB40$_H$</td><td>TS1OPT1A</td><td>FFFF FBC0$_H$</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>TS0DTC1A</td><td>FFFF FB42$_H$</td><td>TS1DTC1A</td><td>FFFF FBC2$_H$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>TS0DTC0A</td><td>FFFF FB44$_H$</td><td>TS1DTC0A</td><td>FFFF FBC4$_H$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>TS0CCR5A</td><td>FFFF FB46$_H$</td><td>TS1CCR5A</td><td>0FFFF FBC6$_H$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>TS0CCR4A</td><td>FFFF FB48$_H$</td><td>TS1CCR4A</td><td>FFFF FBC8$_H$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>TS0CCR3B</td><td>FFFF FB4A$_H$</td><td>TS1CCR3B</td><td>FFFF FBCA$_H$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>TS0CCR2B/
TS0PAT1[a]</td><td>FFFF FB4C$_H$</td><td>TS1CCR2B/
TS1PAT1</td><td>FFFF FBCC$_H$[b]</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>TS0CCR1B/
TS0PAT0[c]</td><td>FFFF FB4E$_H$</td><td>TS1CCR1B/
TS1PAT0[d]</td><td>FFFF FBCE$_H$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>TS0CCR0A</td><td>FFFF FB50$_H$</td><td>TS1CCR0A</td><td>FFFF FBD0$_H$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>TS0CCR3A</td><td>FFFF FB52$_H$</td><td>TS1CCR3A</td><td>FFFF FBD2$_H$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>TS0CCR2A</td><td>FFFF FB54$_H$</td><td>TS1CCR2A</td><td>FFFF FBD4$_H$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>TS0CCR1A</td><td>FFFF FB56$_H$</td><td>TS1CCR1A</td><td>FFFF FBD6$_H$</td></tr>
<tr><td colspan="4">other settings than above</td><td colspan="4">prohibited</td></tr>
</table>
</td></tr>
</table>

a) TS0CCR2B in T-PWM mode, TSOPAT1 in all other modes
b) TS1CCR2B in T-PWM mode, TS1PAT1 in all other modes
c) TS0CCR1B in T-PWM mode, TSOPAT0 in all other modes
d) TS1CCR1B in T-PWM mode, TS1PAT0 in all other modes

(3) MARn - DMA transfer memory start address registers

The 16-bit MARn register specifies the subordinated 16 bits of

- the source address (transfer direction iRAM → I/O)
- the destination start address (transfer direction I/O → iRAM)

within the internal RAM area for the DMA channel n.

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

Address <base> + n x 2_H

Initial Value Undefined.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MARn15	MARn14	MARn13	MARn12	MARn11	MARn10	MARn9	MARn8	MARn7	MARn6	MARn5	MARn4	MARn3	MARn2	MARn1	MARn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-5 MARn register contents

Bit position	Bit name	Function
15 to 0	MARn15 to MARn0	Sets the subordinated 16 bits of the DMA transfer start address (A15 to A0).

Caution

1. Due to the internal RAM size, the value written to the MARn register has to be in the range from 0000_H to $EFFF_H$.
2. For DMA channels that are configured for 16-bit transfer data units, the RAM source or destination area must be 16 bit aligned. Thus the memory address must be even, that is MARn.MARn0 = 0.

(4) DTFRn - DMA trigger factor registers

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

The interrupt request set by this register serves as DMA transfer start factor.

This register is only valid for DMA transfers to serial interfaces.

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

Address DTFR4: <base> + 88_H
DTFR5: <base> + $8A_H$
DTFR6: <base> + $8C_H$
DTFR7: <base> + $8E_H$

Initial Value 00_H. This register is cleared by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
0	0	0	0	0	IFCn2	IFCn1	IFCn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution
1. Do not set the same DMA trigger by different DTFRn registers.
2. Do not rewrite the DTFRn register until a started DMA transfer ends (corresponding DTCRn register value is 00_H).
3. Write the DTFRn register before setting the corresponding DTCRn register. Otherwise, if the DMA trigger is present, the DMA transfer might be started by writing the DTFRn register.

Table 10-6 DTFRn register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>IFCn[2:0]
(n = 4 to 7)</td><td>Sets the DMA transfer start trigger of DMA channel n via interrupt requests from on-chip peripheral I/O (n = 4 to 7).
<table>
<tr><th rowspan="2">IFCn2</th><th rowspan="2">IFCn1</th><th rowspan="2">IFCn0</th><th colspan="2">DMA transfer start factor</th></tr>
<tr><th>when n = 4, 5</th><th>when n = 6, 7</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td colspan="2">DMA request from on-chip peripheral I/O disabled</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>INTUC0R</td><td>INTUC0T</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>INTUC1R</td><td>INTUC1T</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>INTCB0R</td><td>INTCB0T</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTCB1R</td><td>INTCB1T</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTCE0C</td><td>INTCE0C</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>INTCE1C</td><td>INTCE1C</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>INTUC2R</td><td>INTUC2T</td></tr>
</table></td></tr>
</table>

(5) DTCRn - DMA transfer count registers

The 8-bit DTCRn register has two functions:

- Before DMA transfer, it sets the transfer count for DMA channel n.
- During DMA transfer, it stores the remaining transfer count.

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

Address <base> + 40_H + n x 2_H

Initial Value Undefined.

7	6	5	4	3	2	1	0
DTCRn7	DTCRn6	DTCRn5	DTCRn4	DTCRn3	DTCRn2	DTCRn1	DTCRn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-7 DTCRn register contents

Bit position	Bit name	Function
7 to 0	DTCRn[7:0]	Sets the transfer count for DMA channel n and stores the remaining transfer count during DMA transfer.

DTCRn7	DTCRn6	…	DTCRn2	DTCRn1	DTCRn0	Remaining DMA transfer counts
0	0	…	0	0	0	256
0	0	…	0	0	1	1
0	0	…	0	1	0	2
0	0	…	0	1	1	3
…	…	…	…	…	…	...
1	1	…	1	0	1	253
1	1	…	1	1	0	254
1	1	…	1	1	1	255

Caution

1. The value set to the DTCRn register is decreased by each DMA transfer of channel n. It does not keep the initial value after the DMA transfer ends. Therefore, after DMA transfer end the DTCRn register value becomes 00_H.
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.
3. Writing to the DTCRn register during DMA transmission (DMAMC.DEn=1) is prohibited.
4. The DMA transfer ends if DTCRn becomes 00_H. Thus even if a new DMA trigger occurs no transfer takes place and no DMA interrupt is generated.

(6) DMASL - DMA status register

The 8-bit DMASL register displays the transfer status of the DMA channels.

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

Address <base> + 62_H

Initial Value 00_H. This register is cleared by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
DMAS7	DMAS6	DMAS5	DMAS4	DMAS3	DMAS2	DMAS1	DMAS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-8 DMASL register contents

Bit position	Bit name	Function
7 to 0	DMASn	Displays the transfer status of DMA channel n: 0: Transfer is idle or on progress 1: Transfer is completed

Note
1. The DMASn bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it.
2. Since the DMASn bit is not cleared by the DMAC, it has to be cleared by software before DMA transfer is started.

(7) DMAMCL - DMA mode control register

The 8-bit DMAMCL register enables/disables the operation of the DMA channels.

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

Address <base> + 60_H

Initial Value 00_H. This register is cleared by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-9 DMAMCL register contents

Bit position	Bit name	Function
7 to 0	DEn	Enables/disables DMA transfer of channel n: 0: Operation disabled 1: Operation enabled

Caution Writing of a DEn bit is prohibited if the corresponding peripheral function, that generates the DMA trigger, is operating.

(8) DMADSCL - DMA data size control register

The 8-bit DMADSCL register controls the transfer data size of DMA channels 4 to 7. The transfer data size of DMA channels 0 to 3 is fixed to 16 bits.

This register is only valid for DMA transfers to serial interfaces.

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

Address <base> + 64_H

Initial Value 00_H. This register is cleared by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
DMADSC7	DMADSC6	DMADSC5	DMADSC4	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-10 DMADSCL register contents

Bit position	Bit name	Function
7 to 4	DMADSCn (n = 4 to 7)	Sets the transfer data size of DMA channel n: 0: 8 bits 1: 16 bits

10.3 Transfer Details

This section presents details and examples of all possible DMA transfers:

- *"DMA transfer from A/D Converter" on page 290*
- *"DMA transfer to Timer S" on page 293*
- *"DMA transfer to/from serial interfaces" on page 296*

Further, an *"Example of forcible termination of DMA transfer" on page 300* is given.

10.3.1 DMA transfer from A/D Converter

Caution For transferring A/D Converter results from the A/D Converter to the internal RAM via DMA the A/D Converter has to be operated in timer triggered scan mode. Refer to *"A/D Converter (ADC)" on page 567* for information how to set up this mode.

The following table summarizes the DMA transfer ADCm → iRAM (m = 0, 1).

Table 10-11 Overview of DMA transfer from A/D Converter

Channel n	Transfer unit	DMA trigger		Transfer			
				Size	Max. count	Source register	Destination start address MARn
0	ADC0	ADDMARQ0	End of conversion interrupt signal of A/D Converter	16 bit	256	ADDMA0 (fixed)	any even address in iRAM
1	ADC1	ADDMARQ1				ADDMA1 (fixed)	

For each DMA trigger, the data will be transferred from the A/D conversion result register for DMA (ADDMAm) to the internal RAM area specified by the destination start address in MARn.

Transferred data The data that is to be transferred is defined by the following:

- A/D Converter scan area
  The number of ADC channels that are to be scanned can be defined in the A/D Converter.
  The DMA transfer is performed for every finished A/D conversion within the A/D Converter scan area. Thus, for every ADC channel, a separate DMA transfer is required.
- DMA transfer count
  The number of DMA transfers has to be defined in the DTCRn register. The total number of DMA transfers can be calculated as
  scan area size × number of DMA triggers

Note The user has to take care that the number of DMA transfer counts complies with the A/D Converter scan area size and the number of A/D Converter start triggers.

For an example see *"Example of DMA transfer of A/D Converter result registers" on page 291*.

(1) Principle of DMA transfer of A/D Converter result registers

CPU
1. Set A/D Converter scan area (ADMm2 register of the A/D converter)
2. Set destination start address in iRAM (MARn register)
3. Set DMA transfer count (1 to 256, DTCRn register)
4. Clear status bit of DMA channel n (DMASL.DMASn = 0)
5. Enable DMA channel n (DMAMCL.DEn = 1)
6. Enable A/D Converter n

ADC
7. Waiting for A/D Converter start trigger
8. A/D sampling and conversion of next channel in ADCR register
9. A/D Converter generates DMA request and triggers step 12
10. If the complete scan area has not been converted yet, go to step 7
11. Go to step 7

DMA
12. Waiting for DMA trigger (ADDMARQm signal)
13. Data transfer from ADDMAm register to destination start address in iRAM
14. Increment MARn register by 2 (destination start address)
15. Decrement DTCRn by 1 (transfer count)
16. If DMA transfer is not terminated (DTCRn > 0), then go to step 12
17. Stop DMA transfer and disable DMA channel n
18. Set DMASL.DMASn = 1 (transfer status)
19. Generate DMA completion interrupt (INTDMAn)

(2) Example of DMA transfer of A/D Converter result registers

DMA transfer for A/D Converter 0 (ADC0):

- A/D Converter scan area is selected from channel 0 to channel 2.
- Destination start address in iRAM is 0FFF $C000_H$
- A/D scan area should be transferred twice, means trigger count is 2.

CPU
1. Set A/D Converter 0 scan area in register ADM02: channel 0 to channel 2
2. Set destination start address to 0FFF $C000_H$: MAR0 = $C000_H$
3. Set DMA transfer count (2 triggers × 3 channels = 6): DTCR0 = 06_H
4. Clear status bit of DMA channel 0 (DMASL.DMAS0 = 0)
5. Enable DMA channel 0: DMAMCL.DE0 = 1
6. Enable A/D Converter 0

ADC0
7. Waiting for ADC0 start trigger
8. A/D sampling and conversion of channel 0
9. A/D Converter generates DMA request signal ADDMARQ0

DMA
10. Data transfer from ADDMA0 register to destination address 0FFF $C000_H$
11. Increment MAR0 register to 0FFF $C002_H$
12. Decrement DTCR0 register to 05_H

ADC0 13. A/D sampling and conversion of channel 1
14. A/D Converter generates DMA request signal ADDMARQ0

DMA 15. Data transfer from ADDMA0 register to destination address 0FFF $C002_H$
16. Increment MAR0 register to 0FFF $C004_H$
17. Decrement DTCR0 register to 04_H

ADC0 18. A/D sampling and conversion of channel 2
19. A/D Converter generates DMA request signal ADDMARQ0

DMA 20. Data transfer from ADDMA0 register to destination address 0FFF $C004_H$
21. Increment MAR0 register to 0FFF $C006_H$
22. Decrement DTCR0 register to 03_H

ADC0 23. Waiting for ADC0 start trigger (ADDMARQ0)
24. A/D sampling and conversion of channel 0
25. A/D Converter generates DMA request signal ADDMARQ0

DMA 26. Data transfer from ADDMA0 register to destination address 0FFF $C006_H$
27. Increment MAR0 register to 0FFF $C008_H$
28. Decrement DTCR0 register to 02_H

ADC0 29. A/D sampling and conversion of channel 1
30. A/D Converter generates DMA request signal ADDMARQ0

DMA 31. Data transfer from ADDMA0 register to destination address 0FFF $C008_H$
32. Increment MAR0 register to 0FFF $C00A_H$
33. Decrement DTCR0 register to 01_H

ADC0 34. A/D sampling and conversion of channel 2
35. A/D Converter generates DMA request signal ADDMARQ0

DMA 36. Data transfer from ADDMA0 register to destination address 0FFF $C00A_H$
37. Increment MAR0 register to 0FFF $C00C_H$
38. Decrement DTCR0 register to 00_H
39. Stop DMA transfer and disable DMA channel 0, since DTCR0 = 00_H
40. Set DMASL.DMAS0 = 1
41. Generate DMA completion interrupt (INTDMA0)

10.3.2 DMA transfer to Timer S

The following table summarizes the DMA transfer iRAM → TMSm (m = 0, 1):

Table 10-12 Overview of DMA transfer to Timer S

Channel n	Transfer unit	DMA trigger		Transfer				
				Size	Max. count	Source address (MARn)	Destination address	
							Start (SARn)	End
2	TMS0	INTTS0OD, INTTS0CD0	TSmCNT top / bottom reversal	16 bit	01_H to 03_H	any even address in iRAM	refer to below table	TSmCCR1A
3	TMS1	INTTS1OD, INTTS1CD0						

For each DMA trigger, the data will be transferred from internal RAM to the capture/compare registers of timer TMSm.

Transferred data The data that is to be transferred is defined by the following destination area:

- The *destination start address*
  It is defined by the SARn.SARn[3:0] register bits. The address offset of the Timer S DMA start register is evaluated from SARn.SARn[3:0] (refer to below table).

- The *destination end address*
  It is fixed to the TSmCCR1A register of Timer TMSm.

- The timer TMS address mapping
  It defines the order in which registers are written to. The table below lists all possible destination registers together with their address offset and related setting of SARn.SRAn[3:0]:

DMA transfer destination address			
TMSm registers	SARn.SARn[3:0]	Address Offset	
TSmOPT1A	00_H	00_H	Selectable as destination start address
TSmDTC1	01_H	02_H	
TSmDTC0	02_H	04_H	
TSmCCR5A	03_H	06_H	
TSmCCR4A	04_H	08_H	
TSmCCR3B	05_H	$0A_H$	
TSmCCR2B/ TSmPAT1	06_H	$0C_H$	
TSmCCR1B/ TSmPAT0	07_H	$0E_H$	
TSmCCR0A	08_H	10_H	
TSmCCR3A	09_H	12_H	
TSmCCR2A	$0A_H$	14_H	
TSmCCR1A	$0B_H$	16_H	Always destination end address

The DMA transfer is performed for every reload of the internal buffer compare registers by the contents of the capture/compare registers TSmCCR0A to TSmCCR5A.

With every DMA transfer, data is written to all registers within the destination area.

For an example see *"Example of DMA transfer of timer S reload" on page 295*.

(1) Principle of DMA transfer of timer S reload

CPU

1. Set Timer S register start address offset (SARn register)
2. Set DMA source address in iRAM (MARn register)
3. Set DMA transfer count (1 to 3, DTCRn register)
4. Clear status bit of DMA channel n (DMASL.DMASn = 0)
5. Enable DMA channel n (DMAMCL.DEn = 1)

DMA

6. Waiting for DMA trigger (INTTSmOD, INTTSmCD0)
7. Data transfer from DMA source address to TMSm destination register
8. Increment MARn register by 2 (source address)
9. Increment SARn register by 2 (destination address)
10. If SARn is less than or equal to TMSm end register address (TSmCCR1A), then repeat steps 7 to 10
11. Decrement DTCRn by 1 (transfer count)
12. Reset DMA destination address pointer with SARn register value.
13. If DMA transfer is not terminated (DTCRn > 0), then go to step 6
14. Stop DMA transfer and disable DMA channel n
15. Set DMASL.DMASn = 1 (transfer status)
16. Generate DMA completion interrupt (INTDMAn)

(2) Example of DMA transfer of timer S reload

DMA transfer for TMS0:

- Start register is TS0CCR0A.
- Source address in the iRAM is 0FFF $C000_H$.
- DMA trigger count is 2.

CPU

1. Set destination start address to TS0CCR0A (SAR2 =08_H)
2. Set source address in iRAM (MAR2 = 0FFF $C000_H$)
3. Set transfer count (DTCR2 = 02_H)
4. Clear DMA status bit of channel 2 (DMASL.DMAS2 = 0)
5. Enable DMA channel 2 (DMAMCL.DE2 = 1)

DMA

6. Waiting for DMA trigger (INTTS0OD or INTTS0CD0)
7. Data transfer from 0FFF $C000_H$ to TS0CC0A
8. Data transfer from 0FFF $C002_H$ to TS0CC3A
9. Data transfer from 0FFF $C004_H$ to TS0CC2A
10. Data transfer from 0FFF $C006_H$ to TS0CC1A
11. Decrement DTCR2 register to 1
12. Waiting for DMA trigger (INTTS0OD or INTTS0CD0)
13. Data transfer from 0FFF $C008_H$ to TS0CC0A
14. Data transfer from 0FFF $C00A_H$ to TS0CC3A
15. Data transfer from 0FFF $C00C_H$ to TS0CC2A
16. Data transfer from 0FFF $C00E_H$ to TS0CC1A
17. Decrementing DTCR2 register to 0
18. Stop DMA transfer and disable DMA channel 2 (DTCR2 register is 00_H)
19. Set DMASL.DMAS0 = 1 (transfer status)
20. Generate DMA completion interrupt (INTDMA2)

10.3.3 DMA transfer to/from serial interfaces

(1) Serial data reception with DMA transfer

The DMA channels 4 and 5 are dedicated to transfer data from serial interfaces CSIE0, CSIE1, CSIB0, CSIB1, UARTC0, UARTC1, UARTC2 to the iRAM.

The following table summarizes the DMA transfer serial interface → iRAM:

Table 10-13 Overview of DMA transfer from serial interfaces

Channel n	Transfer unit	DMA trigger		Transfer			
				Size	Max. count	Source address	Destination start address (MARn)
4	CSIE0	INTCE0C	Receive complete interrupt	8 bit	255	CE0RX0L	Any iRAM address
				16 bit		CE0RX0	Any even iRAM address
	CSIE1	INTCE1C		8 bit		CE1RX0L	Any iRAM address
				16 bit		CE1RX0	Any even iRAM address
	UARTC0	INTUC0R		8 bit		UC0RX	Any iRAM address
				16 bit		Setting prohibited	
	UARTC1	INTUC1R		8 bit		UC1RX	Any iRAM address
				16 bit		Setting prohibited	
	UARTC2	INTUC2R		8 bit		UC2RX	Any iRAM address
				16 bit		Setting prohibited	
	CSIB0	INTCB0R		8 bit		CB0RXL	Any iRAM address
				16 bit		CB0RX	Any even iRAM address
	CSIB1	INTCB1R		8 bit		CB1RXL	Any iRAM address
				16 bit		CB1RX	Any even iRAM address
5	CSIE0	INTCE0C		8 bit		CE0RX0L	Any iRAM address
				16 bit		CE0RX0	Any even iRAM address
	CSIE1	INTCE1C		8 bit		CE1RX0L	Any iRAM address
				16 bit		CE1RX0	Any even iRAM address
	UARTC0	INTUC0R		8 bit		UC0RX	Any iRAM address
				16 bit		Setting prohibited	
	UARTC1	INTUC1R		8 bit		UC1RX	Any iRAM address
				16 bit		Setting prohibited	
	UARTC2	INTUC2R		8 bit		UC2RX	Any iRAM address
				16 bit		Setting prohibited	
	CSIB0	INTCB0R		8 bit		CB0RXL	Any iRAM address
				16 bit		CB0RX	Any even iRAM address
	CSIB1	INTCB1R		8 bit		CB1RXL	Any iRAM address
				16 bit		CB1RX	Any even iRAM address

Source address The source address is any of the listed serial interfaces' data receive registers. It is implicitly defined by the channel number n and the selection made by the DMA trigger factor register DTFRn register.

Destination address The destination address is any iRAM address (for 16-bit transfers an even iRAM address), specified by the DMA memory start address MARn.

DMA trigger Upon each DMA trigger 8-bit or 16-bit data is transferred from the corresponding serial reception register to the internal RAM.

The DMA trigger is the end-of-reception interrupt signal of the selected serial interface. Like the source address, It is defined by the DMA channel number and the DMA trigger factor register DTFRn register.

Transfer size The transfer data size can be set to 8- or 16-bit via the register DMADSCL.

- In case of 8-bit transfer data size, the destination address is incremented by 1 for each DMA trigger.
- In case of 16-bit transfer data size, the destination address MARn must be even, and is incremented by 2 for each DMA trigger.

DMA transfer procedure of serial data reception

CPU

1. Set destination start address in iRAM (MARn register)
2. Select demanded serial interface (DTFRn register)
3. Set up DMA transfer data size (8 or 16 bits) of channel n (DMADSC.DMADSCn)
4. Set DMA transfer count (1 to 255, DTCRn register)
5. Clear status bit of DMA channel n (DMASL.DMASn = 0)
6. Enable DMA channel n (DMAMCL.DEn = 1)

DMA

7. Waiting for occurrence of DMA trigger
8. Data transfer from source in peripheral I/O register to iRAM
9. Decrement DTCRn register by 1
10. If DMA transfer is not terminated (DTCRn > 0), then go to step 7
11. Stop DMA transfer and disable DMA channel n
12. Set DMASL.DMASn = 1 (transfer status)
13. Generate DMA completion interrupt (INTDMAn)

(2) Serial data transmission with DMA transfer

The DMA channels 6 and 7 are dedicated to transfer data from the iRAM to the serial interfaces CSIE0, CSIE1, CSIB0, CSIB1, UARTC0, UARTC1, UARTC2.

The following table summarizes the DMA transfer iRAM → serial interface:

Table 10-14 Overview of DMA transfer from serial interfaces

Channel n	Transfer unit	DMA trigger		Transfer			
				Size	Max. count	Source address (MARn)	Destination start address
6	CSIE0[a]	INTCE0C	Transmit interrupt	8 bit	255	Any iRAM address	CE0TX0L
				16 bit		Any even iRAM address	CE0TX0
	CSIE1[a]	INTCE1C		8 bit		Any iRAM address	CE1TX0L
				16 bit		Any even iRAM address	CE1TX0
	UARTC0	INTUC0T		8 bit		Any iRAM address	UC0TX
				16 bit		Setting prohibited	
	UARTC1	INTUC1T		8 bit		Any iRAM address	UC1TX
				16 bit		Setting prohibited	
	UARTC2	INTUC2T		8 bit		Any iRAM address	UC2TX
				16 bit		Setting prohibited	
	CSIB0	INTCB0T		8 bit		Any iRAM address	CB0TXL
				16 bit		Any even iRAM address	CB0TX
	CSIB1	INTCB1T		8 bit		Any iRAM address	CB1TXL
				16 bit		Any even iRAM address	CB1TX
7	CSIE0[a]	INTCE0C		8 bit		Any iRAM address	CE0TX0L
				16 bit		Any even iRAM address	CE0TX0
	CSIE1[a]	INTCE1C		8 bit		Any iRAM address	CE1TX0L
				16 bit		Any even iRAM address	CE1TX0
	UARTC0	INTUC0T		8 bit		Any iRAM address	UC0TX
				16 bit		Setting prohibited	
	UARTC1	INTUC1T		8 bit		Any iRAM address	UC1TX
				16 bit		Setting prohibited	
	UARTC2	INTUC2T		8 bit		Any iRAM address	UC2TX
				16 bit		Setting prohibited	
	CSIB0	INTCB0T		8 bit		Any iRAM address	CB0TXL
				16 bit		Any even iRAM address	CB0TX
	CSIB1	INTCB1T		8 bit		Any iRAM address	CB1TXL
				16 bit		Any even iRAM address	CB1TX

a) Writing to the chip select data buffer registers CSnCS is not supported by DMA.

Source address The source address is any iRAM address (for 16-bit transfers an even iRAM address), specified by the DMA memory start address MARn.

Destination address The destination address is any of the listed serial interfaces' data transmit registers. It is implicitly defined by the channel number n and the selection made by the DMA trigger factor register DTFRn register.

DMA trigger Upon each DMA trigger 8-bit or 16-bit is transferred from the internal RAM to the corresponding serial transmit register.

The DMA trigger is the transmit-enable interrupt signal of the selected serial interface. Like the destination address, It is defined by the DMA channel number and the DMA trigger factor register DTFRn register.

Transfer size The transfer data size can be set to 8- or 16-bit via the register DMADSCL.

- In case of 8-bit transfer data size, the source address is incremented by 1 for each DMA trigger.
- In case of 16-bit transfer data size, the source address MARn must be even, and is incremented by 2 for each DMA trigger.

DMA Transfer Procedure of Serial Data Transmission

CPU

1. Set source address in iRAM (MARn register)
2. Select demanded serial interface (DTFRn register)
3. Set up DMA transfer data size (8 or 16 bits) of channel n (DMADSC.DMADSCn)
4. Set DMA transfer count (1 to 255, DTCRn register)
5. Clear status bit of DMA channel n (DMASL.DMASn = 0)
6. Enable DMA channel n (DMAMCL.DEn = 1)

DMA

7. Waiting for occurrence of DMA trigger
8. Data transfer from source in iRAM to peripheral I/O register
9. Decrement DTCRn register by 1
10. If DMA transfer is not terminated (DTCRn > 0), then go to step 7
11. Stop DMA transfer and disable DMA channel n
12. Set DMASL.DMASn = 1 (transfer status)
13. Generate DMA completion interrupt (INTDMAn)

10.3.4 Example of forcible termination of DMA transfer

CPU
1. Set DMA transfer count (e.g. DTCRn = 10_H)
2. Start DMA transfer (DMAMCL.DEn = 1)

DMA
3. Waiting for DMA trigger
4. Start first data transfer
5. Decrement DTCRn register by 1 (e.g. DTCRn = 09_H)
6. Waiting for DMA trigger
7. Start next data transfer

CPU
8. Stop DMA transfer forcibly (DMAMCL.DEn = 0)

DMA
9. Decrement DTCRn register by 1 (e.g. DTCRn = 08_H)
10. DMA transfer is stopped completely

Chapter 11 Asynchronous Serial Interface (UARTC)

This microcontroller has three instances of the universal Asynchronous Serial Interface UARTC.

Note Throughout this chapter, the individual instances of UARTCn are identified by “n” (n = 0 to 2), for example UCnCTL0 for the UARTCn control register 0.

11.1 Features

- Transfer rate: 300 bps to 4 Mbps
- Full-duplex communication:
  - Internal UARTC receive data register n (UCnRX)
  - Internal UARTC transmit data register n (UCnTX)
- 2-pin configuration:
  - TXDCn: Transmit data output pin
  - RXDCn: Receive data input pin
- Reception error output function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3
  - Reception complete interrupt (INTUCnR):

    This interrupt occurs upon transfer of receive data from the shift register to receive buffer register n after serial transfer completion, in the reception enabled status.
  - Transmission enable interrupt (INTUCnT):

    This interrupt occurs upon transfer of transmit data from the transmit buffer register to the shift register in the transmission enabled status.
  - Receive error interrupt (INTUCnRE):

    This interrupt occurs upon transfer of erroneous receive data.
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB-/LSB-first transfer selectable
- Transmit/receive data level inversion possible

- 13 to 20 bits selectable for the SBF (Sync Break Field) in the LIN (Local Interconnect Network) communication format
  - Recognition of 11 bits or more possible for SBF reception in LIN communication format
  - SBF reception flag provided
- Extension bit operation possible (uses parity bit as 9th data bit)
- Transfer and reception status flags

11.2 Configuration

The block diagram of the UARTCn is shown below.

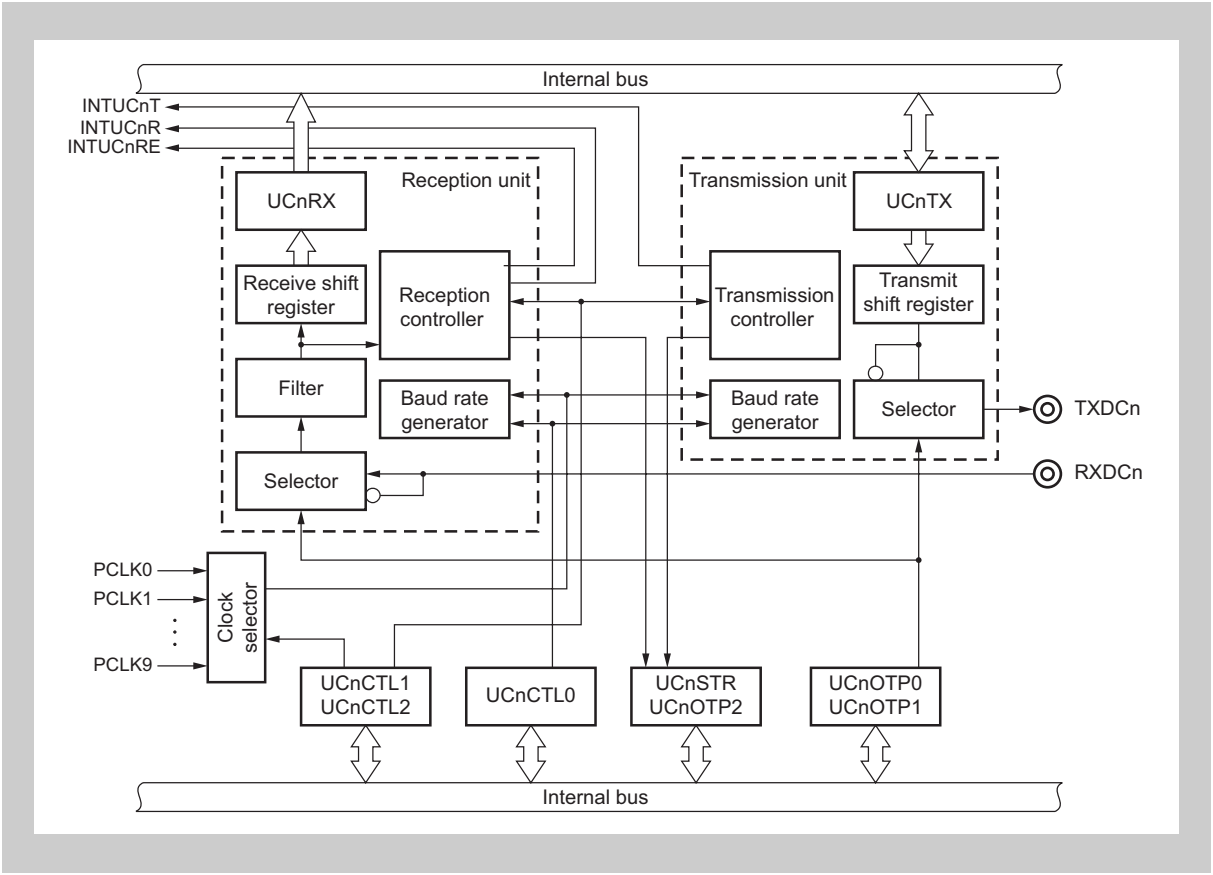


Figure 11-1 Block diagram of Asynchronous Serial Interface UARTCn

Note PCLK0 to PCLK9: peripheral clocks (refer to *"Clock Generator" on page 179*). For the configuration of the baud rate generator, see *Figure 11-11 on page 329*.

UARTCn consists of the following hardware units.

Table 11-1 Configuration of UARTCn

Item	Configuration
Registers	UARTCn control register 0 (UCnCTL0) UARTCn control register 1 (UCnCTL1) UARTCn control register 2 (UCnCTL2) UARTCn option control register 0 (UCnOPT0) UARTCn option control register 1 (UCnOPT1) UARTCn option control register 2 (UCnOPT2) UARTCn status register (UCnSTR) UARTCn receive shift register UARTCn receive data register (UCnRX) UARTCn transmit shift register UARTCn transmit data register (UCnTX)

(1) UARTCn control register 0 (UCnCTL0)

The UCnCTL0 register is an 8-bit register used to specify the UARTCn operation.

(2) UARTCn control register 1 (UCnCTL1)

The UCnCTL1 register is an 8-bit register used to select the input clock for the UARTCn.

(3) UARTCn control register 2 (UCnCTL2)

The UCnCTL2 register is an 8-bit register used to control the baud rate for the UARTCn.

(4) UARTCn option control register 0 (UCnOPT0)

The UCnOPT0 register is an 8-bit register used to control serial transfer for the UARTCn.

(5) UARTCn option control register 1 (UCnOPT1)

The UCnOPT1 register is an 8-bit register used to control the extension bit operation.

(6) UARTCn option control register 2 (UCnOPT2)

The UCnOPT2 register is an 8-bit register indicates the operating status during a reception.

(7) UARTCn status register (UCnSTR)

The UCnSTRn register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error and is reset (to 0) by reading the UCnSTR register.

(8) UARTCn receive shift register

This is a shift register used to convert the serial data input to the RXDCn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UCnRX register.

This register cannot be manipulated directly.

(9) UARTCn receive data register (UCnRX)

The UCnRX register is an 8-bit register that holds receive data. When 7 characters are received, 0 is stored in the highest bit (when data is received LSB first).

In the reception enabled status, receive data is transferred from the UARTCn receive shift register to the UCnRX register in synchronization with the completion of shift-in processing of 1 frame.

Transfer to the UCnRX register also causes the reception complete interrupt request signal (INTUCnR) to be output.

(10) UARTCn transmit shift register

The transmit shift register is a shift register used to convert the parallel data transferred from the UCnTX register into serial data.

When 1 byte of data is transferred from the UCnTX register, the shift register data is output from the TXDCn pin.

This register cannot be manipulated directly.

(11) UARTCn transmit data register (UCnTX)

The UCnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UCnTX register. When data can be written to the UCnTX register (when data of one frame is transferred from the UCnTX register to the UARTCn transmit shift register), the transmission enable interrupt request signal (INTUCnT) is generated.

11.3 UARTC Registers

(1) UCnCTL0 - UARTCn control register 0

The UCnCTL0 register is an 8-bit register that controls the UARTCn serial transfer operation.

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

Reset input sets this register to 10H.

Caution Be sure to set the UCnPWR bit = 1 and the UCnRXE bit = 1 while the RXDCn pin is high level (when UCnRDL bit of UCnOP0 register = 0).
If the UCnPWR bit = 1 and the UCnRXE bit = 1 are set while the RXDCn pin is low level, reception will inadvertently start.

After reset: 10H R/W Address: UC0CTL0 FFFFFA00H,
UC1CTL0 FFFFFA20H,
UC2CTL0 FFFFFA40H

	7	6	5	4	3	2	1	0
UCnCTL0	UCnPWR	UCnTXE	UCnRXE	UCnDIR	UCnPS1	UCnPS0	UCnCL	UCnSL

UCnPWR	UARTCn Operation Control
0	Stops clock operation (UARTCn reset asynchronously)
1	Enables operating clock operation
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.	

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. • When UCnPWR bit = 0, the value written to the UCnTXE bit is ignored.	

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. • When UCnPWR bit = 0, the value written to the UCnRXE bit is ignored.	

UCnDIR	Transfer Direction Selection
0	MSB-first transfer
1	LSB-first transfer
This bit can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.	

UCnPS1	UCnPS0	Parity Selection	
		During Transmission	During Reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	Reception with 0 parity
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check

- These bits can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.
- If “Reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UCnPE bit of the UCnSTA0 register is not set, no error interrupt is output.
- When transmission and reception are performed in the LIN format, set the UCnPS1 and UCnPS0 bits to 00B.

UCnCL	Data Character Length Specification
0	7 bits
1	8 bits

This bit can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.

UCnSL	Stop Bit Length Specification
0	1 bit
1	2 bits

This bit can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.

Note For details of parity, see *“Parity types and operations” on page 326*.

(2) UCnCTL1 - UARTCn control register 1

This register controls the Baud Rate Generator. For details see *“UCnCTL1 - UARTCn control register 1” on page 330*.

(3) UCnCTL2 - UARTCn control register 2

This register controls the Baud Rate Generator. For details see *“UCnCTL2 - UARTCn control register 2” on page 331*.

(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.

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

Reset input sets this register to 14H.

After reset: 14H R/W Address: UC0OPT0 FFFFFA03H,
UC1OPT0 FFFFFA23H,
UC2OPT0 FFFFFA43H

	7	6	5	4	3	2	1	0
UCnOPT0	UCnSRF	UCnSRT	UCnSTT	UCnSLS2	UCnSLS1	UCnSLS0	UCnTDL	UCnRDL

UCnSRF	SBF Reception Flag
0	When UCnPWR of UCnCTL0 register = 0 and UCnRXE of UCnCTL0 register = 0 are set. Also upon normal end of SBF reception.
1	During SBF reception
• SBF (Sync Brake Field) reception is judged during LIN communication. • The UCnSRF bit is held high when a SBF reception error occurs, and then SBF reception is started again.	

UCnSRT	SBF Reception Trigger
0	–
1	SBF reception trigger
• This is the SBF reception trigger bit during LIN communication, and when read, “0” is always read. For SBF reception, set the UCnSRT bit (to 1) to enable reception. • Set the UCnSRT bit after setting the UCnPWR bit of the UCnCTL0 register to 1 and the UCnRXE bit of the UCnCTL0 register to 1.	

UCnSTT	SBF Transmission Trigger
0	–
1	SBF transmission trigger
• This is the SBF transmission trigger bit during LIN communication, and when read, “0” is always read. • Set the UCnSTT bit after setting the UCnPWR bit of the UCnCTL0 register to 1 and the UCnTXE bit of the UCnCTL0 register to 1.	

UCnSLS2	UCnSLS1	UCnSLS0	SBF Length Selection
1	0	1	13-bit output (reset value)
1	1	0	14-bit output
1	1	1	15-bit output
0	1	0	16-bit output
0	0	1	17-bit output
0	0	0	18-bit output
0	1	1	19-bit output
1	0	0	20-bit output
This register can be set when the UCnPWR bit of the UCnCTL0 register is 0 or when the UCnTXE bit of the UCnCTL0 register is 0.			

UCnTDL	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. • This bit can be set when the UCnPWR bit of the UCnCTL0 register is 0 or when the UCnTXE bit of the UCnCTL0 register is 0.	

UCnRDL	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. • This bit can be set when the UCnPWR bit of the UCnCTL0 register is 0 or the UCnRXE bit of the UCnCTL0 register is 0.	

Note Before starting the SBF transmission by UCnOPT0.UAnSTT=1 make sure that no data transfer is ongoing, that means check that UCnSTR.UAnTSF=0.

(5) UCnOPT1 - UARTCn option control register 1

The UCnOPT1 register is an 8-bit register that controls the extension bit operation of the UARTCn.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: UC0OPT1 FFFFFA0AH,
UC1OPT1 FFFFFA2AH,
UC2OPT1 FFFFFA4AH

	7	6	5	4	3	2	1	0
UCnOPT1	0	0	0	0	0	0	0	UCnEBE

UCnEBE	Extension Bit Operation Enable
0	Extension bit operation disabled. Transfer data length set by UCnCL bit of the UCnCTL0 register.
1	Extension bit operation enabled.

- During extension bit operation a 9-th data bit is sent or received instead of the parity bit.
- Extension bit operation is only effective when the parity selection is set to no parity (UCnPS1, UCnPS0 bits = 00B), and the character length is set to 8 bits (UCnCL bit = 1). In all other cases the setting of UCnEBE bit is ignored.

Table 11-2 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[Note]	Stop	
			1	1	Data	Data	Data[Note]	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

Note Insertion of extension bit.

(6) UARTCn option control register 2 (UCnOPT2)

The UCnOPT2 register is an 8-bit register that displays the UARTCn reception status.

The register is read only, and can be read in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R Address: UC0OPT2 FFFFFA0BH,
UC1OPT2 FFFFFA2BH,
UC2OPT2 FFFFFA4BH

	7	6	5	4	3	2	1	0
UCnOPT2	0	0	0	0	0	0	0	UCnRSF

UCnRSF	Receive Status Flag
0	• When UCnCTL0.UCnPWR = 0 or UCnCTL0.UCnRXE = 0 has been set. • 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. 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)	

(7) UCnSTR - UARTCn status register

The UCnSTR register is an 8-bit register that displays the UARTCn transfer status and reception error contents.

This register can be read or written in 8-bit or 1-bit units, but the UCnTSF bit is a read-only bit, while the UCnPE, UCnFE, and UCnOVE bits can both be read and written. However, these bits can only be cleared by writing 0; they cannot be set by writing 1 (even if 1 is written to them, the value is retained).

The initialization conditions are shown below.

Register/Bit	Initialization conditions
UCnSTR register	• Reset • UCnCTL0.UCnPWR = 0
UCnTSF bit	• UCnCTL0.UCnTXE = 0
UCnPE, UCnFE, UCnOVE bits	• 0 write • UCnCTL0.UCnRXE = 0

After reset: 00H R/W Address: UC0STR FFFFFA04H,
UC1STR FFFFFA24H,
UC2STR FFFFFA44H

	7	6	5	4	3	2	1	0
UCnSTR	UCnTSF	0	0	0	0	UCnPE	UCnFE	UCnOVE

UCnTSF	Transfer Status Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnTXE bit of UCnCTL0 register = 0 has been set. • When, following transfer completion, there was no next data transfer from UCnTX
1	Writing to UCnTX register ongoing
The UCnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UCnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while UCnTSF bit = 1.	

UCnPE	Parity Error Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set. • When 0 has been written
1	• When parity of data and parity bit do not match during reception.
• The operation of the UCnPE bit is controlled by the settings of the UCnPS1 and UCnPS0 bits of the UCnCTL0 register. • The UCnPE bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.	

UCnFE	Framing Error Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set • When 0 has been written
1	When no stop bit is detected during reception
• Only the first bit of the receive data stop bits is checked, regardless of the value of the UCnSL bit of the UCnCTL0 register. • The UCnFE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.	

UCnOVE	Overrun Error Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set. • When 0 has been written
1	When receive data has been set to the UCnRXB register and the next receive operation is completed before that receive data has been read
• When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer. • The UCnOVE bit can be both read and written, but it can only be cleared by writing 0 to it. When 1 is written to this bit, the hold status is entered.	

(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 (UCnEBE bit = 0) the data bits are received in the lower byte of the UCnRX register. The lower byte can be read also by 8-bit access of the UCnRXL register.

During LSB-first reception when the data length has been specified as 7 bits and the extension bit operation is disabled, the receive data is transferred to bits 6 to 0 of the UCnRXL register and the MSB always becomes 0. During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UCnRXL register and the LSB always becomes 0.

When an overrun error (UCnOVE bit = 1) occurs, the receive data at this time is not transferred to the UCnRX and UXnRXL register respectively.

The UCnRX register is read-only, in 16-bit units.

The UCnRXL register is read-only, in 8-bit units.

In addition to reset input, the UCnRX register can be set to 1FFH, and the UCnRXL register can be set to FFH respectively, by clearing the UCnPWR bit of the UCnCTL0 register to 0.

After reset: 1FFH R Address: UC0RX FFFFFA06H,
UC1RX FFFFFA26H,
UC2RX FFFFFA46H

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

After reset: FFH R Address: UC0RXL FFFFFA06H,
UC1RXL FFFFFA26H,
UC2RXL FFFFFA46H

	7	6	5	4	3	2	1	0
UCnRXL								

(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 (UCnEBE bit = 0).

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

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

Reset input sets the UCnTX register to 1FFH, and the UCnTXL register to FFH.

After reset: 1FFH R/W Address: UC0TX FFFFFA08H,
UC1TX FFFFFA28H,
UC2TX FFFFFA48H

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

UCnTXL

After reset: FFH R/W Address: UC0TXL FFFFFA08H,
UC1TXL FFFFFA28H,
UC2TXL FFFFFA48H

	7	6	5	4	3	2	1	0
UCnTXL								

11.4 Interrupt Request Signals

The following three interrupt request signals are generated from UARTCn:

- Reception complete interrupt request signal (INTUCnR)
- Receive error interrupt request signal (INTUCnRE)
- Transmission enable interrupt request signal (INTUCnT)

(1) Reception complete interrupt request signal (INTUCnR)

A reception complete interrupt request signal is output when data is shifted into the receive shift register and transferred to the UCnRX register in the reception enabled status.

In case of erroneous reception, the reception error interrupt INTUanRE is generated instead of INTUCnR.

No reception complete interrupt request signal is generated in the reception disabled status.

(2) Receive error interrupt request signal (INTUCnRE)

A receive error interrupt request is generated if an error condition occurred during reception, as reflected by UCnSTR.UCnPE (parity error flag), UCnSTR.UCnFE (framing error flag), UCnSTR.UCnOVE (overrun error flag).

Note that INTUCnR and INTUCnRE do exclude each other: upon correct reception of data only INTUCnR is generated. In case of a reception error INTUCnRE is generated only.

(3) Transmission enable interrupt request signal (INTUCnT)

If transmit data is transferred from the UCnTX register to the UCRTCn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

11.5 Operation

11.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in the figures below, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UCnCTL0 register. UARTCn features additionally the extension bit operation for a ninth transfer data bit, which can be specified in the UCnOPT1 register.

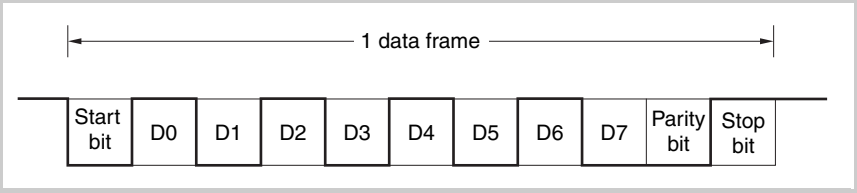
Moreover, control of UART output/inverted output for the TXDCn bit is performed using the UCnOPT0.UCnTDL bit.

- Start bit.....1 bit
- Character bits.....7 bits/8 bits/9 bits
- Parity bit .....Even parity/odd parity/0 parity/no parity
- Stop bit.....1 bit/2 bits

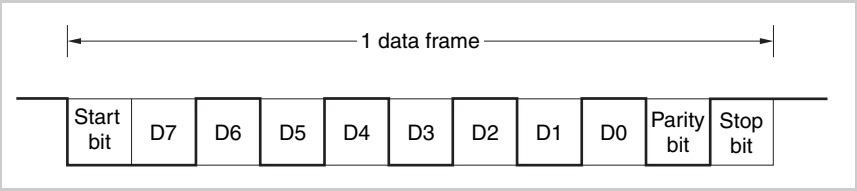
Note Extension bit operation presumes no parity setting.

(1) UARTC transmit/receive data format

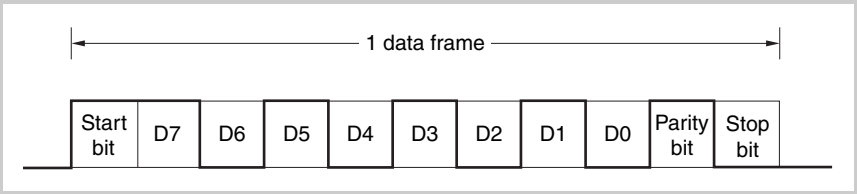
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



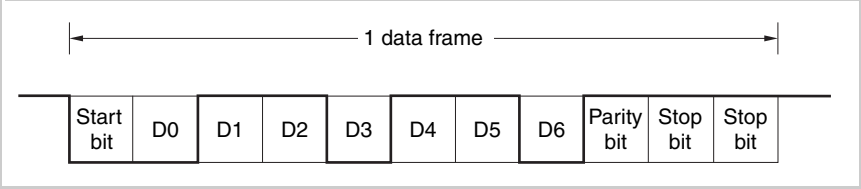
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



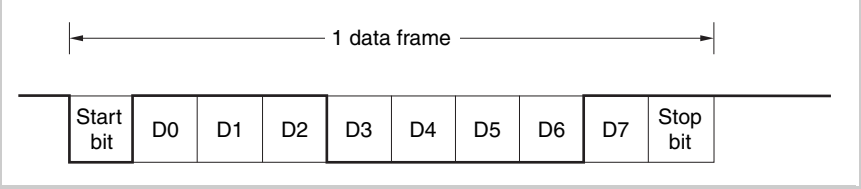
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDCn inversion



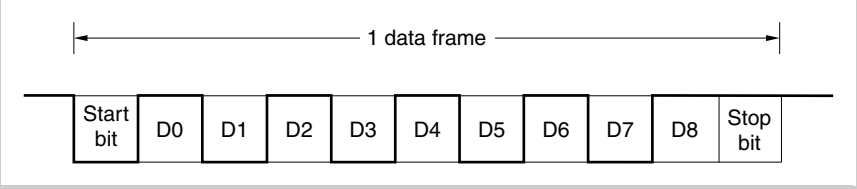
(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



(f) 9-bit data length, LSB first, no parity, 1 stop bit, transfer data: 155H



11.5.2 SBF transmission/reception format

The UARTC has an SBF (Sync Break Field) transmission/reception control function to enable use of the LIN function.

About LIN

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figure 11-2 and *Figure 11-3* outline the transmission and reception manipulations of LIN.

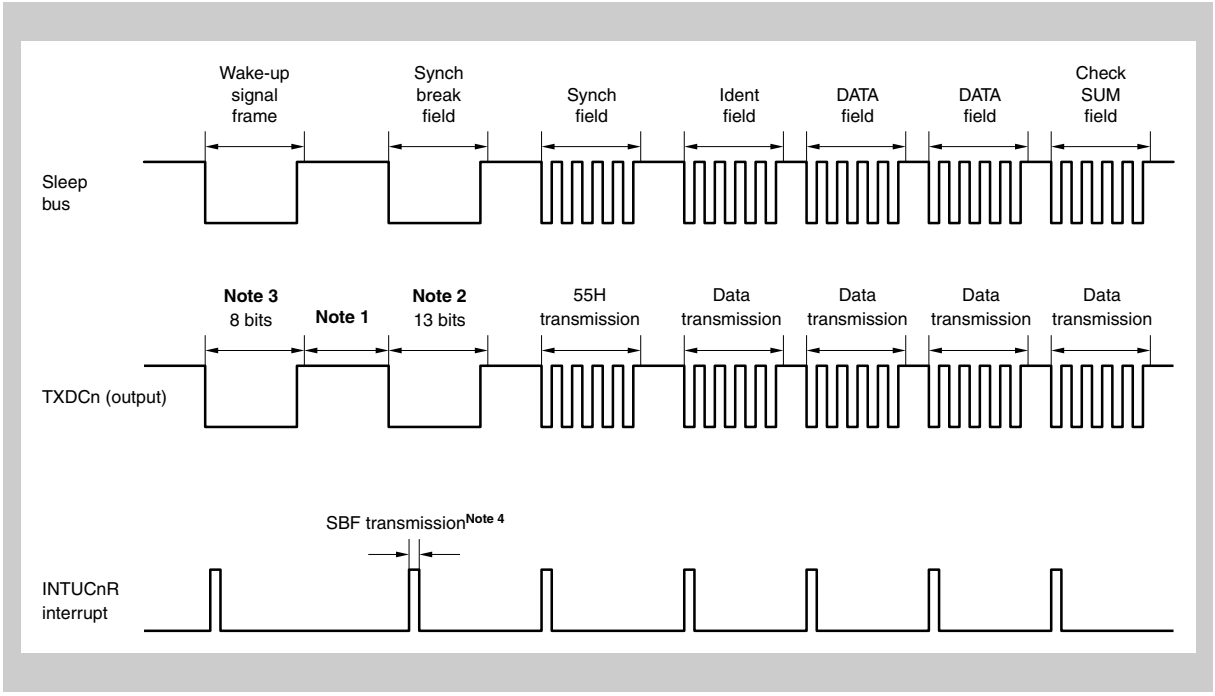


Figure 11-2 LIN transmission manipulation outline

Note

1. The interval between each field is controlled by software.
2. SBF output is performed by hardware. The output width is the bit length set by the UCnOPT0.UCnSLS2 to UCnOPT0.UCnSLS0 bits. If even finer output width adjustments are required, such adjustments can be performed using the UCnCTLn.UCnBRS7 to UCnCTLn.UCnBRS0 bits.
3. 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.

4. A transmission enable interrupt request signal (INTUCnT) is output at the start of each transmission. The INTUCnT signal is also output at the start of each SBF transmission.

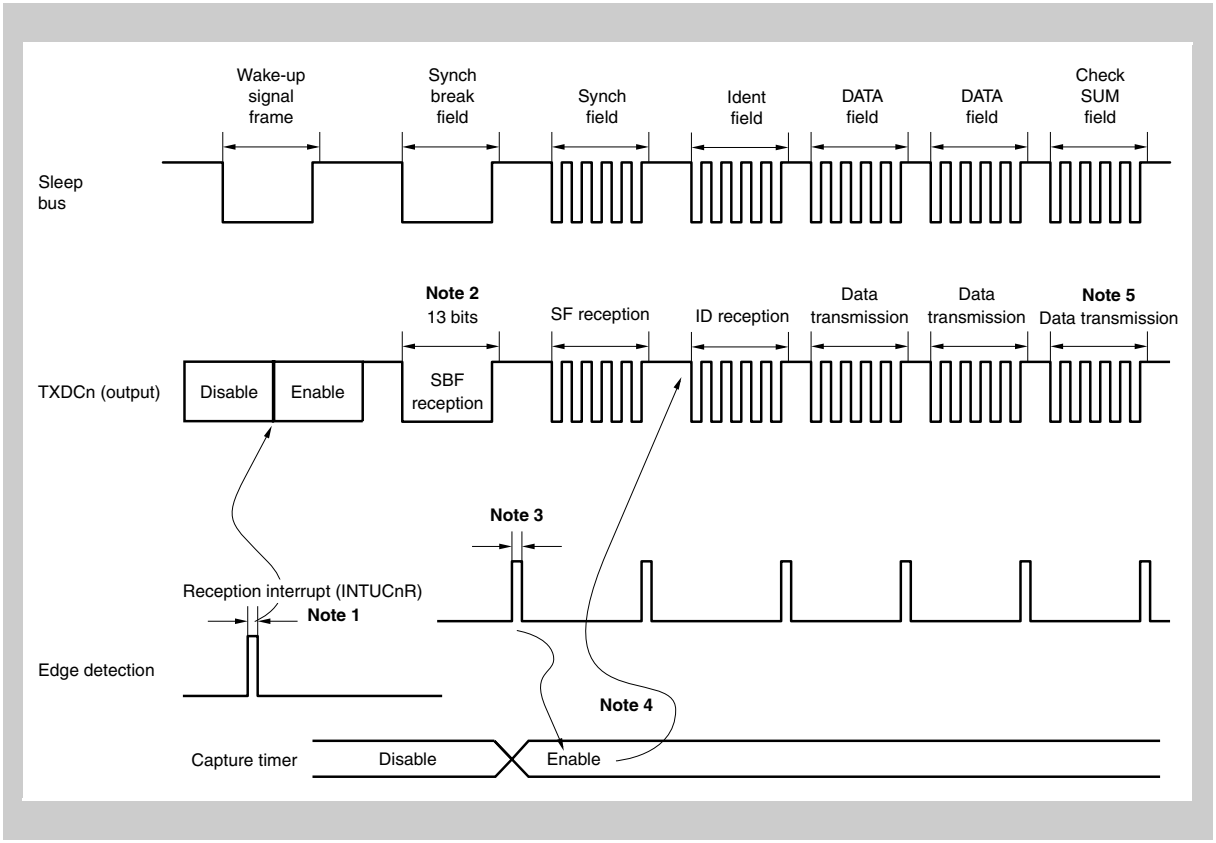


Figure 11-3 LIN reception manipulation outline

Note 1. The wakeup signal is sent by the pin edge detector, UARTCn is enabled, and the SBF reception mode is set.

2. The receive operation is performed until detection of the stop bit. Upon detection of SBF reception of 11 or more bits, normal SBF reception end is judged, and an interrupt signal is output. Upon detection of SBF reception of less than 11 bits, an SBF reception error is judged, no interrupt signal is output, and the mode returns to the SBF reception mode.

3. If SBF reception ends normally, an interrupt request signal is output. The timer is enabled by an SBF reception complete interrupt. Moreover, error detection for the UCnSTR.UCnOVE, UCnSTR.UCnPE, and UCnSTR.UCnFE bits is suppressed and UART communication error detection processing and UARTCn receive shift register and data transfer of the UCnRX register are not performed. The UARTCn receive shift register holds the initial value, FFH.

4. The RXDCn pin is connected to TI (capture input) of the timer, the transfer rate is calculated, and the baud rate error is calculated. The value of the UCnCTL2 register obtained by correcting the baud rate error after dropping UARTC enable is set again, causing the status to become the reception status.

5. Check-sum field distinctions are made by software. UARTCn is initialized following CSF reception, and the processing for setting the SBF reception mode again is performed by software.

11.5.3 SBF transmission

When the UCnCTL0.UCnPWR bit = UCnCTL0.UCnTXE bit = 1, the transmission enabled status is entered, and SBF transmission is started by setting (to 1) the SBF transmission trigger (UCnOPT0.UCnSTT bit).

Thereafter, a low level width of bits 13 to 20 specified by the UCnOPT0.UCnSLS2 to UCnOPT0.UCnSLS0 bits is output. A transmission enable interrupt request signal (INTUCnT) is generated upon SBF transmission start. Following the end of SBF transmission, the UCnSTT bit is automatically cleared. Thereafter, the UART transmission mode is restored.

Transmission is suspended until the data to be transmitted next is written to the UCnTX register, or until the SBF transmission trigger (UCnSTT bit) is set.

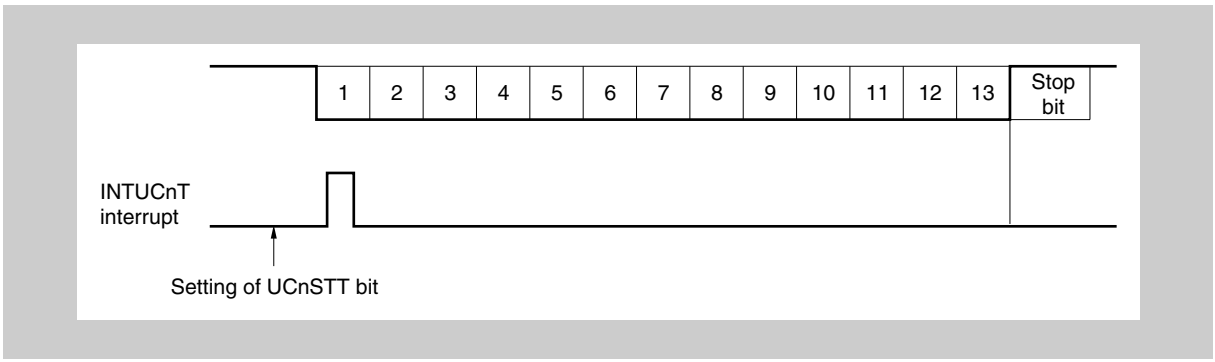


Figure 11-4 SBF transmission

11.5.4 SBF reception

The reception enabled status is achieved by setting the UCnCTL0.UCnPWR bit to 1 and then setting the UCnCTL0.UCnRX bit to 1.

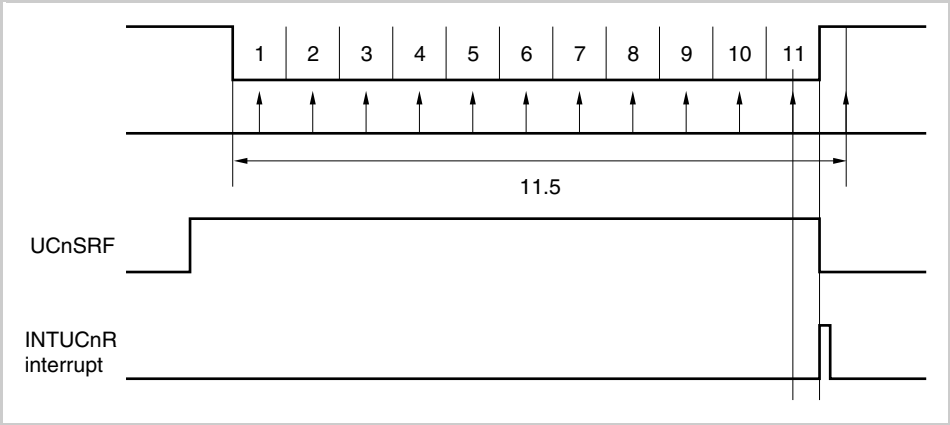
The SBF reception wait status is set by setting the SBF reception trigger (UCnOPT0.UCnSTR bit) to 1.

In the SBF reception wait status, similarly to the UART reception wait status, the RXDCn pin is monitored and start bit detection is performed.

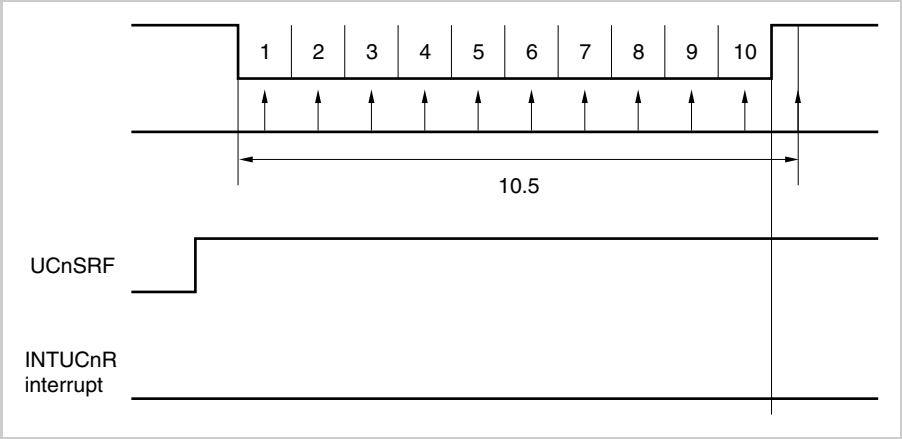
Following detection of the start bit, reception is started and the internal counter counts up according to the set baud rate.

When a stop bit is received, if the SBF width is 11 or more bits, normal processing is judged and a reception complete interrupt request signal (INTUCnR) is output. The UCnOPT0.UCnSRF bit is automatically cleared and SBF reception ends. Error detection for the UCnSTR.UCnOVE, UCnSTR.UCnPE, and UCnSTR.UCnFE bits is suppressed and UART communication error detection processing is not performed. Moreover, data transfer of the UARTCn reception shift register and UCnRX register is not performed and FFH, the initial value, is held. If the SBF width is 10 or fewer bits, reception is terminated as error processing without outputting an interrupt, and the SBF reception mode is returned to. The UCnSRF bit is not cleared at this time.

(a) Normal SBF reception (detection of stop bit in more than 10.5 bits)



(b) SBF reception error (detection of stop bit in 10.5 or fewer bits)



11.5.5 UART transmission

The transmission enabled status is set by setting UCnCTL0.UCnTXE to 1, after UCnCTL0.UCnPWR was set to 1, and transmission is started by writing transmit data to the UCnTX register. The start bit, parity bit, and stop bit are automatically added.

The data in the UCnTX register is transferred to the UARTCn transmit shift register upon the start of the transmit operation.

A transmission enable interrupt (INTUCnT) is generated upon completion of transmission of the data of the UCnTX register to the UARTCn transmit shift register, and thereafter the contents of the UARTCn transmit shift register are output to the TXDCn pin LSB first.

Write of the next transmit data to the UCnTX register is enabled by generating the INTUCnT signal.

Continuous transmission is enabled by writing the data to be transmitted next to the UCnTX register during transfer.

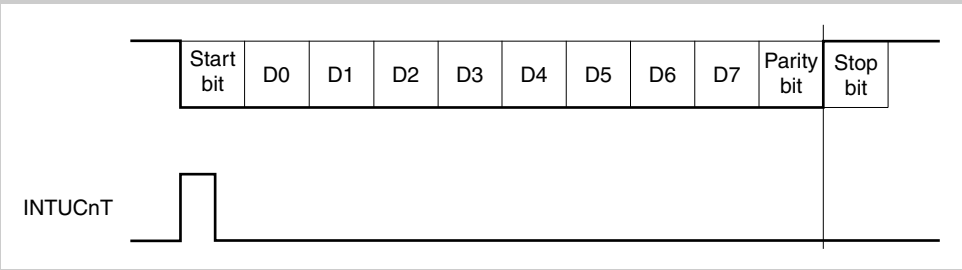


Figure 11-5 UART transmission

Note LSB first

11.5.6 Continuous transmission procedure

UARTCn can write the next transmit data to the UCnTX register when the UARTCn transmit shift register starts the shift operation. The transfer timing of the UARTCn transmit shift register can be judged from the transmission enable interrupt (INTUCnT). Transmission can be performed without interruption even during interrupt processing following the transmission of 1 data frame via the INTUCnT signal, and an efficient communication rate can thus be achieved.

During continuous transmission, overrun (the completion of the next transmission before the first transmission completion processing has been executed) may occur.

An overrun can be detected by incorporating a program that can count the number of transmit data and by referencing transfer status flag (UCnSTR.UCnTSF).

Caution During continuous transmission execution, perform initialization after checking that the UCnSTR.UCnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UCnTSF bit is 1.

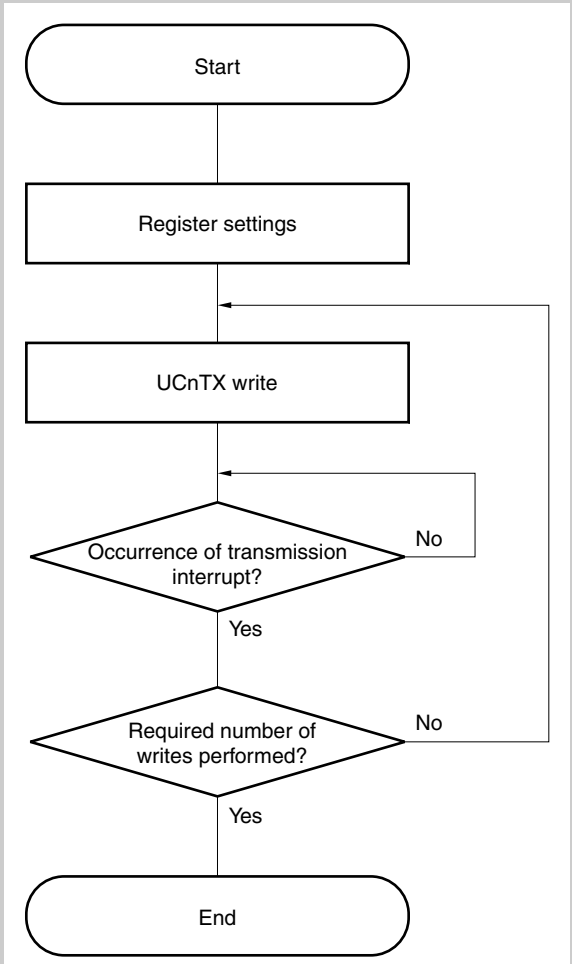


Figure 11-6 Continuous transmission processing flow

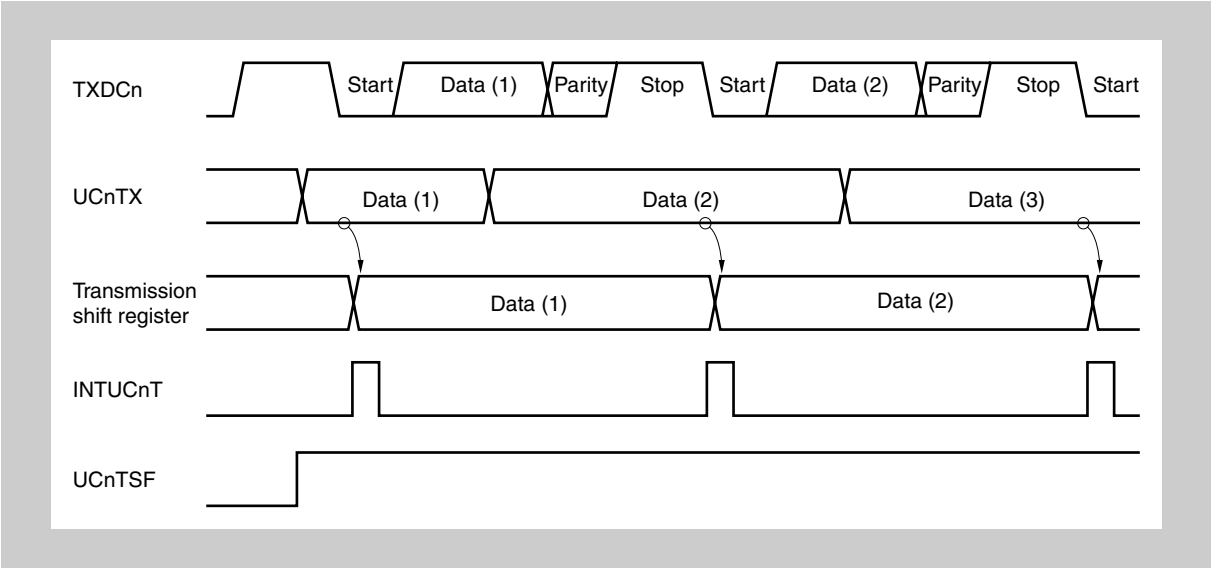


Figure 11-7 Continuous transmission operation timing —transmission start

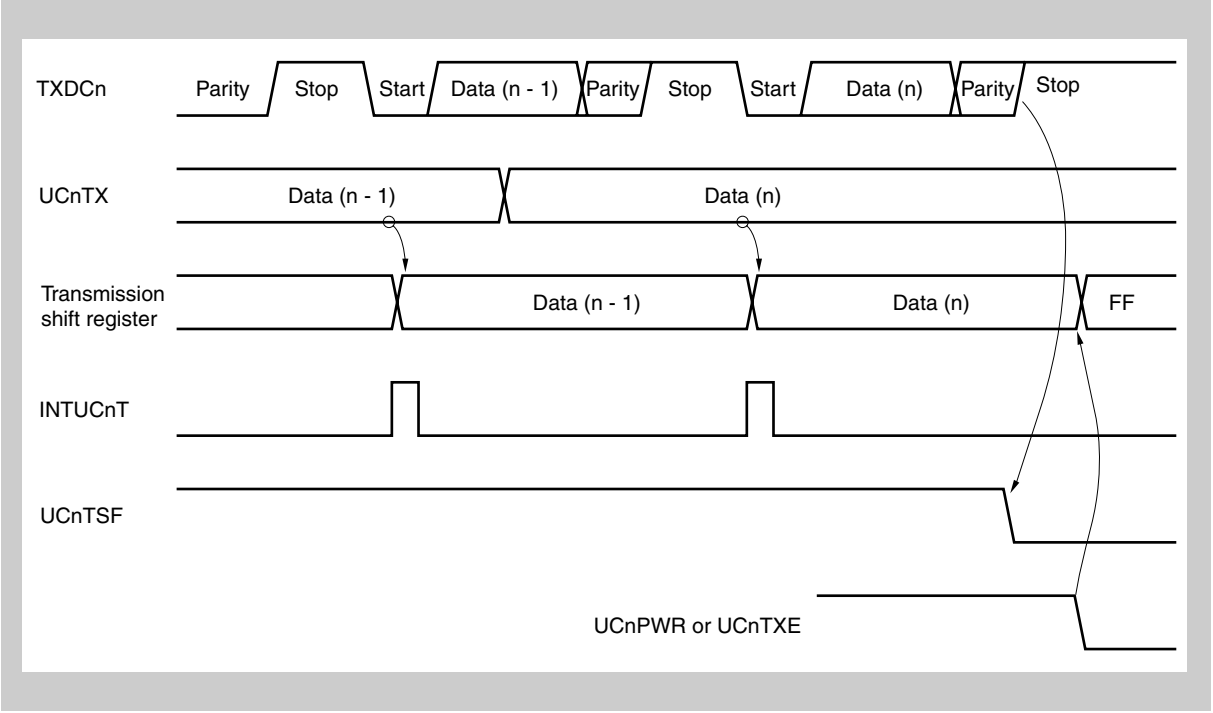


Figure 11-8 Continuous transmission operation timing—transmission end

11.5.7 UART reception

The reception wait status is set by setting UCnCTL0.UCnPWR to 1 and then setting UCnCTL0.UCnRXE to 1. In the reception wait status, the RXDCn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

First, an 8-bit counter starts upon detection of the falling edge of the RXDCn pin. When the 8-bit counter has counted the UCnCTL2 register setting value, the level of the RXDCn pin is monitored again (corresponds to the — mark in Figure 15-19). If the RXDCn pin is low level at this time too, a start bit is recognized. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTCn receive shift register according to the set baud rate. Additionally the flag UCnOPT2.UCnRSF is set (1) to indicate the receive operation status.

When the reception complete interrupt (INTUCnR) is output upon reception of the stop bit, the data of the UARTCn receive shift register is written to the UCnRX register, and the UCnRSF flag is cleared (0) simultaneously. However, if an overrun error occurs (UCnOVE bit = 1), the receive data at this time is not written to the UCnRX register, and a reception error interrupt (INTUCnRE) is output.

Even if a parity error (UCnPE bit = 1) or a framing error (UCnFE bit = 1) occurs during reception, reception continues until the stop bit reception position, but a reception error interrupt (INTUCnRE) is output following reception completion.

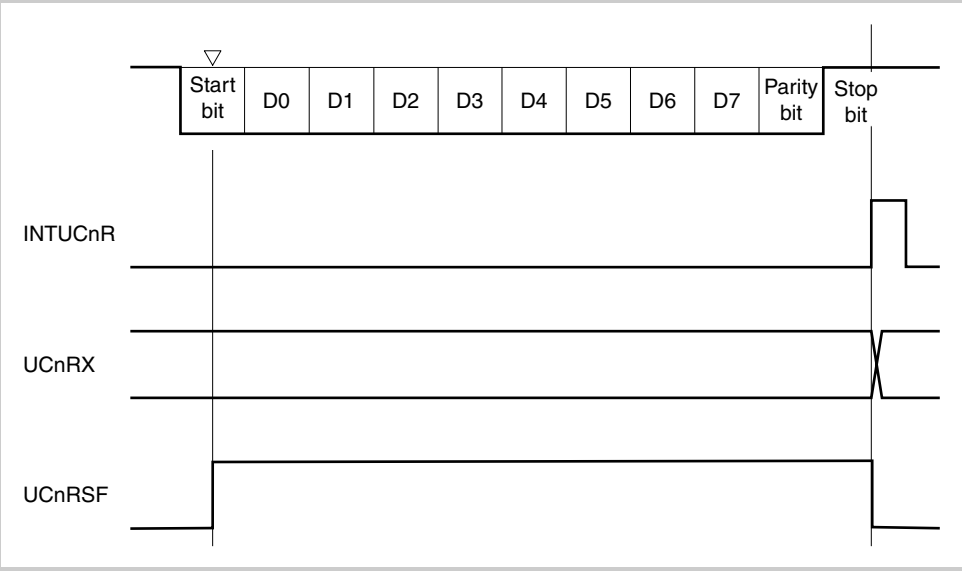


Figure 11-9 UART reception

Caution

1. Be sure to read the UCnRX register even when a reception error occurs. If the UCnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.
2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.
3. When reception is completed, read the UCnRX register after the reception complete interrupt request signal (INTUCnR) has been generated, and clear the UCnPWR or UCnRXE bit to 0. If the UCnPWR or UCnRXE bit is cleared to 0 before the INTUCnR signal is generated, the read value of the UCnRX register cannot be guaranteed.
4. If receive completion processing (INTUCnR signal generation) of UARTCn and the UCnPWR bit = 0 or UCnRXE bit = 0 conflict, the INTUCnR signal may be generated in spite of these being no data stored in the UCnRX register.
   To complete reception without waiting INTUCnR signal generation, be sure to clear (0) the interrupt request flag (UCnRIF) of the UCnRIC register, after setting (1) the interrupt mask flag (UCnRMK) of the interrupt control register (UCnRIC) and then set (1) the UCnPWR bit = 0 or UCnRXE bit = 0.

11.5.8 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the UCnSTR register and a reception error interrupt request signal (INTUCnRE) is output when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the UCnSTR register.

Clear the reception error flag by writing 0 to it after reading it.

Table 11-3 Reception error causes

Error flag	Reception error	Cause
UCnPE	Parity error	Received parity bit does not match the setting
UCnFE	Framing error	Stop bit not detected
UCnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

Caution

1. Be sure to read the UCnRX register even when a reception error occurs. If the UCnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.
2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

Note Note that even in case of a parity or framing error, data is transferred from the receive shift register to the receive data register UCnRX. Consequently the data from UCnRX must be read. Otherwise an overrun error UCnSTR.UCnOVE will occur at reception of the next data.

In case of an overrun error, the receive shift register data is not transferred to UCnRX, thus the previous data is not overwritten.

11.5.9 Parity types and operations

Caution When using the LIN function, fix the UCnPS1 and UCnPS0 bits of the UCnCTL0 register to 00.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

- During transmission
  The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.
  – Odd number of bits whose value is “1” among transmit data: 1
  – Even number of bits whose value is “1” among transmit data: 0
- During reception
  The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(2) Odd parity

- During transmission
  Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.
  – Odd number of bits whose value is “1” among transmit data: 0
  – Even number of bits whose value is “1” among transmit data: 1
- During reception
  The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(3) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

11.5.10 Receive data noise filter

This filter samples the RXDCn pin using the internal system clock (fxx/2).

When the same sampling value is read twice, the match detector output changes and the RXDCn signal is sampled as the input data.

Moreover, since the circuit is as shown in *Figure 11-10*, the processing that goes on within the receive operation is delayed by 2 clocks in relation to the external signal status.

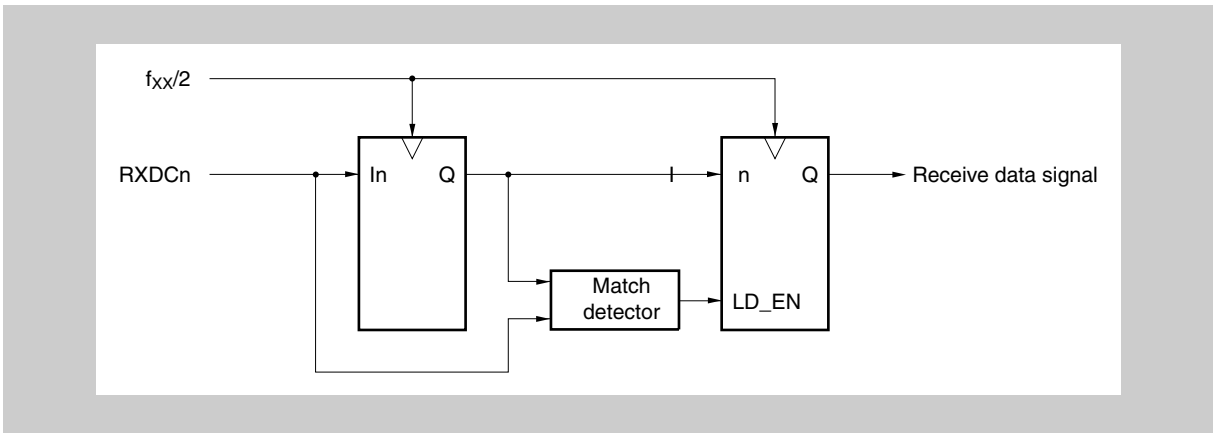


Figure 11-10 Noise filter circuit

11.6 Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTCn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is an 8-bit counter for transmission and another one for reception.

(1) Baud rate generator configuration

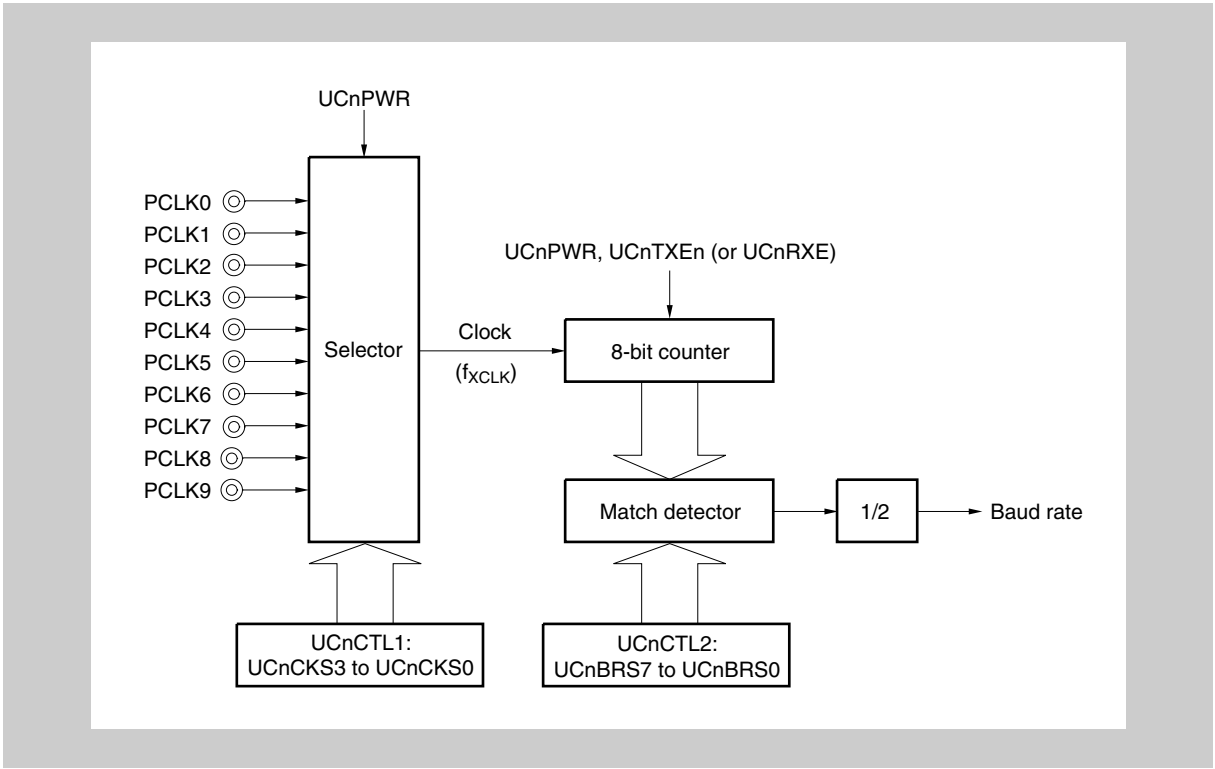


Figure 11-11 Configuration of baud rate generator

Note PCLK0 to PCLK9: peripheral clocks (refer to *"Clock Generator" on page 179*):

PCLK	0	1	2	3	4	5	6	7	8	9
f [MHz]	32	16	8	4	2	1	0.5	0.25	0.125	0.0625

(a) Base clock (Clock)

When the UCnCTL0.UCnPWR bit is 1, the clock selected by the UCnCTL1.UCnCKS[3:0] bits is supplied to the 8-bit counter. This clock is called the base clock. When the UCnPWR bit = 0, the clock is fixed to the low level.

(b) Serial clock generation

A serial clock can be generated by setting the UCnCTL1 register and the UCnCTL2 register.

The base clock is selected by bits UCnCTL1.UCnCKS[3:0] bits.

The frequency division value for the 8-bit counter can be set using the UCnCTL2.UCnBRS[7:0] bits.

(2) UCnCTL1 - UARTCn control register 1

The UCnCTL1 register is an 8-bit register that selects the UARTCn base clock (f_{XCLK}).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: UC0CTL1 FFFFFA01H, UC1CTL1 FFFFFA21H, UC2CTL1 FFFFFA41H

	7	6	5	4	3	2	1	0
UCnCTL1	0	0	0	0	UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0

UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0	Base clock (f_{XCLK}) selection
0	0	0	0	PCLK0 (32 MHz)
0	0	0	1	PCLK1 (16 MHz)
0	0	1	0	PCLK2 (8 MHz)
0	0	1	1	PCLK3 (4 MHz)
0	1	0	0	PCLK4 (2 MHz)
0	1	0	1	PCLK5 (1 MHz)
0	1	1	0	PCLK6 (500 KHz)
0	1	1	1	PCLK7 (250 KHz)
1	0	0	0	PCLK8 (125 KHz)
1	0	0	1	PCLK9 (62.5 KHz)
all others				setting prohibited

Note PCLK0 to PCLK9: peripheral clocks (refer to *"Clock Generator" on page 179*)

Caution Clear the UCnCTL0.UCnPWR bit to 0 before rewriting the UCnCTL1 register.

(3) UCnCTL2 - UARTCn control register 2

The UCnCTL2 register is an 8-bit register that specifies the divisor to control the baud rate (serial transfer speed) clock of UARTCn.

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

Reset input sets this register to FFH.

After reset: FFH R/W Address: UC0CTL2 FFFFFA02H,
UC1CTL2 FFFFFA22H,
UC2CTL2 FFFFFA42H

	7	6	5	4	3	2	1	0
UCnCTL2	UCnBRS7	UCnBRS6	UCnBRS5	UCnBRS4	UCnBRS3	UCnBRS2	UCnBRS1	UCnBRS0

UCn BRS7	UCn BRS6	UCn BRS5	UCn BRS4	UCn BRS3	UCn BRS2	UCn BRS1	UCn BRS0	Default (k)	Serial clock
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	$f_{XCLK}/255$

Note f_{XCLK}: Clock frequency selected by UCnCTL1.UCnCKS[3:0].

Caution Clear the UCnCTL0.UCnPWR bit to 0 or clear the UCnTXE and UCnRXE bits to 00 before rewriting the UCnCTL2 register.

(4) 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)

(5) Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1\right) \times 100\ [\%]$$

Caution

1. The baud rate error during transmission must be within the error tolerance on the receiving side.
2. The baud rate error during reception must satisfy the range indicated in *"Allowable baud rate range during reception" on page 333*.

Example

- System clock frequency f_{xx} = 128 MHz
- Setting value of
  - UCnCTL1.UDnCKS[3:0] = 0001_B: f_{XCLK} = PCLK1 = 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 [%]

(6) Baud rate setting example

Table 11-4 Baud rate generator setting data

Target baud rate [bps]	f_{xx} = 128 MHz				
	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
38400	02H	8 MHz	68H	104	0.16
56000	01H	16 MHz	8FH	143	-0.1
76800	02H	8 MHz	1AH	52	0.16
125000	02H	8 MHz	20H	32	0.00
153600	02H	16 MHz	1AH	52	0.16
250000	02H	8 MHz	10H	16	0.00
312500	00H	32 MHz	33H	51	0.39
1000000	00H	32 MHz	10H	16	0.00
2000000	00H	32 MHz	08H	8	0.00

(7) Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

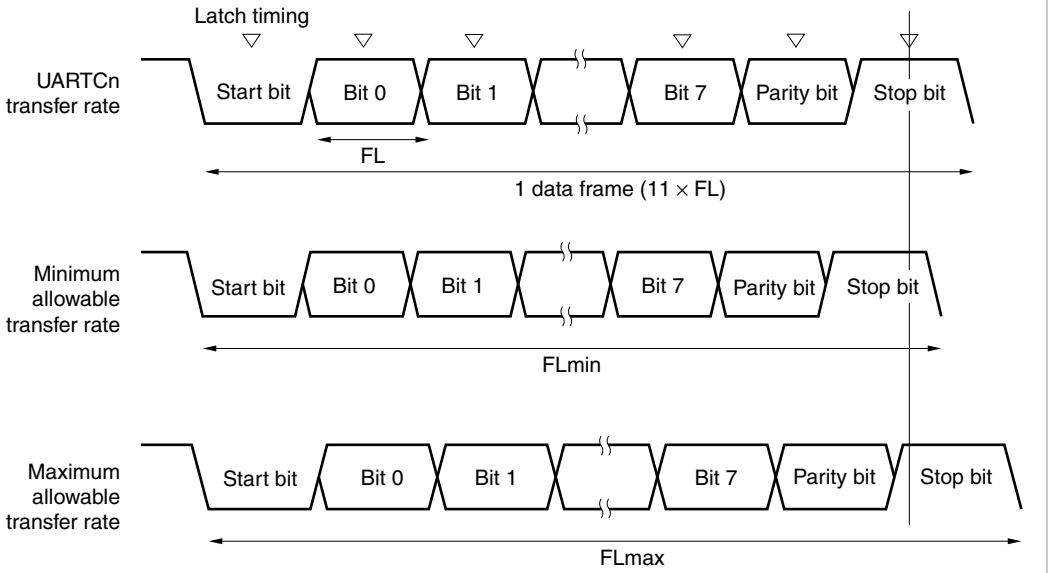


Figure 11-12 Allowable baud rate range during reception

As shown in *Figure 11-12*, the receive data latch timing is determined by the counter set using the UCnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$FL = (Brate)^{-1}$

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.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} \times Brate$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2k} \times FL = \frac{21k-2}{2k} \times FL$$

$$FLmax = \frac{21k-2}{20k} \times FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} \times Brate$$

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 11-5 Maximum/Minimum allowable baud rate error

Division ratio (k)	Maximum allowable baud rate error	Minimum allowable baud rate error
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

Note 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UCnCTL2.UCnBRS[7:0]

(8) Baud rate during continuous transmission

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

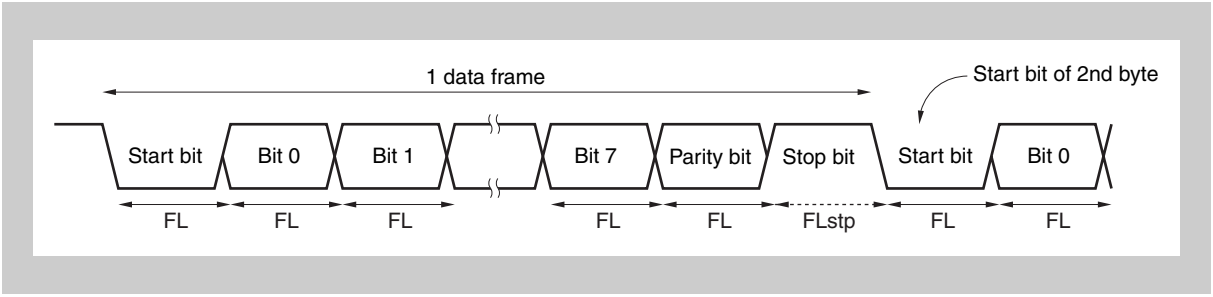


Figure 11-13 Transfer rate during continuous transfer

Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{XCLK}, we obtain the following equation.

$$FLstp = FL + 2/f_{XCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{XCLK})$$

11.7 Cautions

(1) UARTCn operation stop

If both of the following actions in UARTCn happen at the same time the INTUCnR signal may be generated inadvertently and no data is stored in the UCnRX register:

- INTUCnR is generated due to completion of a serial receive operation,
- UCnPWR bit or UCnRXE bit of the UCnCTL0 register is cleared (set to 0).

Workaround To avoid the generation of the INTUCnR signal when UCnPWR bit or UCnRXE bit is cleared (set to 0) do the following:

1. Set (set to 1) the interrupt mask flag (UCnRMK) of the interrupt control register (UCnRIC),
2. Clear (set to 0) the UCnPWR bit or UCnRXE bit of the UCnCTL0 register,
3. Clear (set to 0) the interrupt request flag (UCnRIF) of the UCnRIC register.

(2) UARTCn receive error interrupt

If both of the following actions in UARTCn happen at the same time the INTUCnRE may not be generated and the INTUCnR signal may be generated inadvertently:

- INTUCnRE is generated due to erroneous frame reception.
- UCnPWR bit or UCnRXE bit of control register UCnCTL0 is cleared to 0.

Workaround

1. Set the interrupt mask flag UCnRMK of the interrupt control register UCnRIC to 1 and set the interrupt mask flag UCnREMK of the interrupt control register UCnREIC.
2. Clear the UCnPWR bit or UCnRXE bit of the UCnCTL0 register to 0.
3. Clear the interrupt request flag UCnRIF of the UCnRIC register and clear the interrupt request flag UCnREIF of the UCnREIC register to 0.

Chapter 12 Clocked Serial Interface (CSIB)

The microcontroller has two instances of the Clocked Serial Interface CSIB: CSIB0 to CSIB1.

Note Throughout this chapter, the individual instances of CSIB are identified by “n” (n = 0, 1).

12.1 Features

- Transfer rate: Maximum 8 Mbps
- Master mode and slave mode selectable
- 8-bit to 16-bit transfer, 3-wire serial interface
- 3 interrupt request signals (INTCBnT, INTCBnR, INTCBnRE)
- Serial clock and data phase switchable
- Transfer data length selectable in 1-bit units between 8 and 16 bits
- Transfer data MSB-first/LSB-first switchable
- 3-wire transfer

  SOBn: Serial data output
  SIBn: Serial data input
  SCKBn: Serial clock input/output

  Transmission mode, reception mode, and transmission/reception mode specifiable
- Slave select function supported

  $\overline{\text{SSBn}}$: Serial slave select input
- DMA support
- Dedicated baud rate generator for each interface instance

12.2 Configuration

The following shows the block diagram of CSIBn.

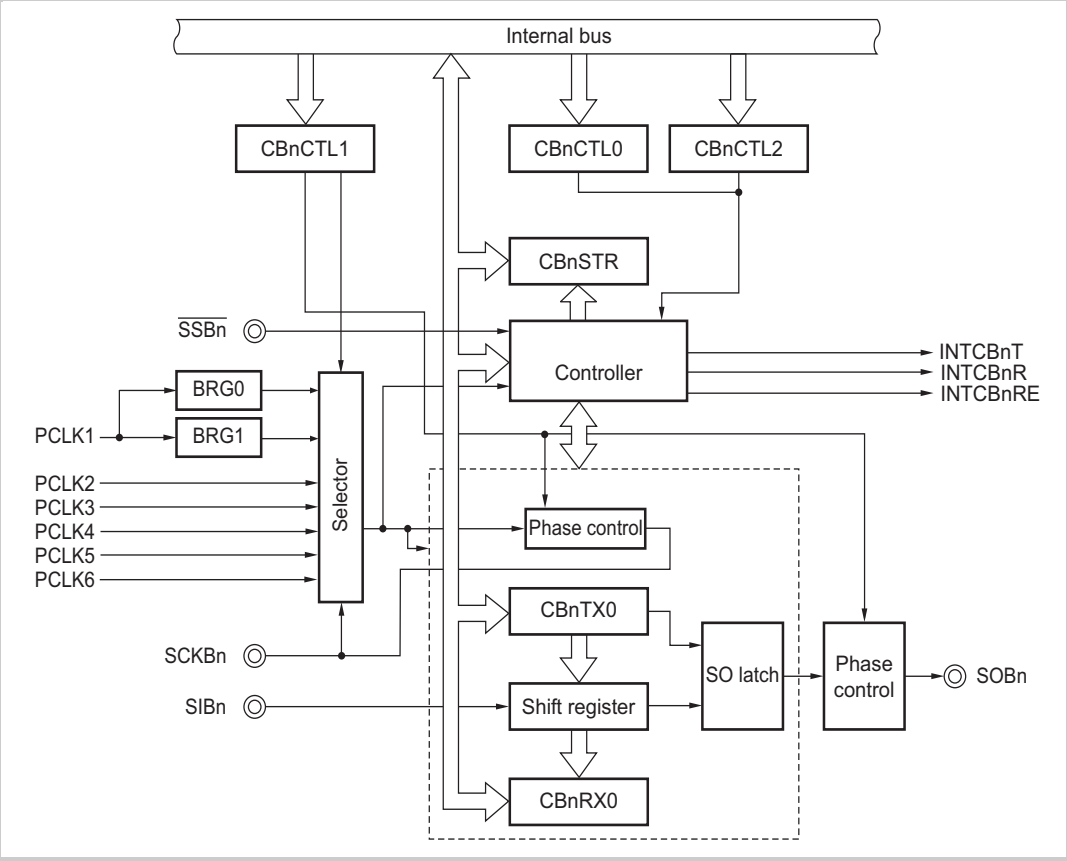


Figure 12-1 Block diagram of CSIBn

Note PCLK1 to PCLK6: peripheral clocks (refer to *"Clock Generator" on page 179*):

PCLK	1	2	3	4	5	6
f [MHz]	16	8	4	2	1	0.5

Additional clocks are generated by dedicated baud rate generators BRG0/BRG1.

CSIBn includes the following hardware.

Table 12-1 Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX0) CSIBn transmit data register (CBnTX0)
Control registers	CSIBn control register 0 (CBnCTL0) CSIBn control register 1 (CBnCTL1) CSIBn control register 2 (CBnCTL2) CSIBn status register (CBnSTR)

(1) CBnRX0 - CSIBn receive data register

The CBnRX0 register is a 16-bit buffer register that holds receive data.

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

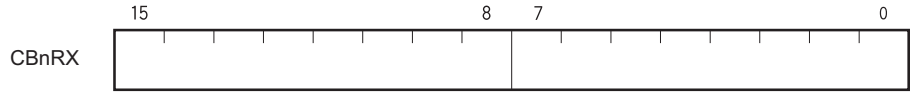
The receive operation is started by reading the CBnRX0 register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRX0L register.

Reset input clears this register to 0000H.

In addition to reset input, the CBnRX0 register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

After reset: 0000H R Address: CB0RX0 FFFFFD04H, CB1RX0 FFFFFD24H



(2) CBnTX0 - CSIB transmit data register

The CBnTX0 register is a 16-bit buffer register used to write the CSIBn transfer data.

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

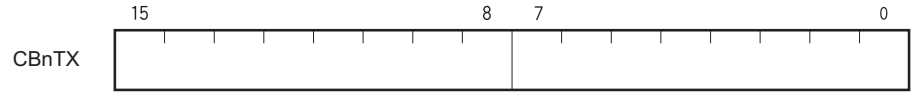
The transmit operation is started by writing data to the CBnTX0 register in the transmission enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnTX0L register.

Reset input clears this register to 0000H.

In addition to reset input, the CBnTX0 register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

After reset 0000H R/W Address: CB0TX0 FFFFFD06H, CB1TX0 FFFFFD26H



Note The communication start conditions are shown below:

Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):
Write to CBnTX0 register

Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):
Write to CBnTX0 register

Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):
Read from CBnRX0 register

12.3 CSIB Control Registers

The following registers are used to control CSIBn.

- CSIBn control register 0 (CBnCTL0)
- CSIBn control register 1 (CBnCTL1)
- CSIBn control register 2 (CBnCTL2)
- CSIBn status register (CBnSTR)

(1) CBnCTL0 - CSIBn control register 0

CBnCTL0 is a register that controls the CSIBn serial transfer operation.

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

Reset input sets this register to 01H.

(1/2)

After reset: 01H R/W Address: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD20H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CBnCTL0	CBnPWR	CBnTXE[Note]	CBnRXE[Note]	CBnDIR[Note]	0	CBnSSE[Note]	CBnTMS[Note]	CBnSCE

CBnPWR	Specification of CSIBn operation disable/enable
0	Disable CSIBn operation and reset the CBnSTR register
1	Enable CSIBn operation
• The CBnPWR bit controls the CSIBn operation and resets the internal circuit.	

CBnTXE[Note]	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOBn output is low level when the CBnTXE bit is 0.	

CBnRXE[Note]	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• 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.	

CBnDIR[Note]	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

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.

(2/2)

CBnSSE[Note 1]	Slave Selection Operation Enable
0	Slave selection function disabled
1	Slave selection function enabled
When the CSIBn serves as slave, it executes transmission/reception in synchronization with the clock only when a low level is input to the $\overline{SSBn}$ pin.	

CBnTMS[Note 1]	Transfer mode specification
0	Single transfer mode
1	Continuous transfer mode

CBnSCE	Specification of start transfer disable/enable
0	Communication start trigger invalid
1	Communication start trigger valid
• In master mode This bit enables or disables the communication start trigger. (a) In single transmission or transmission/reception mode, or continuous transmission or continuous transmission/reception mode a communication operation can be started only when the CBnSCE bit is 1. Set the CBnSCE bit to 1. (b) In single reception mode Clear the CBnSCE bit to 0 before reading the receive data (CBnRX0 register)[Note 2]. (c) In continuous reception mode Clear the CBnSCE bit to 0 one communication clock before reception of the last data is completed[Note 3]. • In slave mode This bit enables or disables the communication start trigger. Set the CBnSCE bit to 1.	

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.

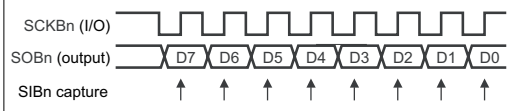
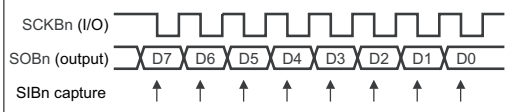
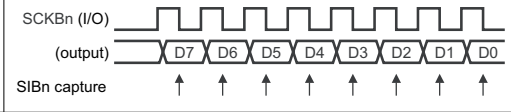
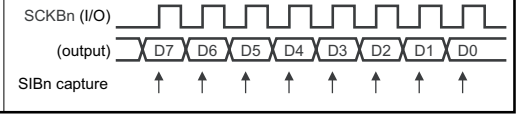
This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.

After reset 00H R/W Address: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD21H,

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	Specification of data transmission/ reception timing in relation to SCKBn
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Communication clock	Mode
0	0	0	BRG0	Master mode
0	0	1	BRG1	Master mode
0	1	0	PCLK2	Master mode
0	1	1	PCLK3	Master mode
1	0	0	PCLK4	Master mode
1	0	1	PCLK5	Master mode
1	1	0	PCLK6	Master mode
1	1	1	External clock (SCKBn)	Slave mode

Note PCLK1 to PCLK6: peripheral clocks (refer to *"Clock Generator" on page 179*):

PCLK	1	2	3	4	5	6
f [MHz]	16	8	4	2	1	0.5

Additional clocks are generated by dedicated baud rate generators BRG0/BRG1.

(3) CBnCTL2 - CSIBn control register 2

CBnCTL2 is an 8-bit register that controls the number of CSIBn serial transfer bits.

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

Reset input clears this register to 00H.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

After reset: 00H R/W Address: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD22H

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Serial register bit length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

Note If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX0 and CBnRX0 registers.

(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 CBnTX0 or CBnRX0 register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

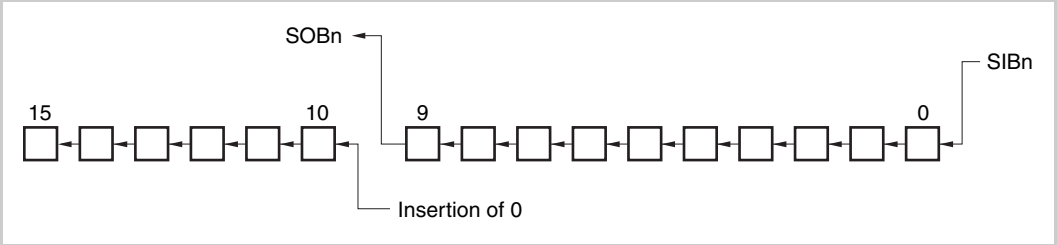


Figure 12-2 (i) Transfer bit length = 10 bits, MSB first

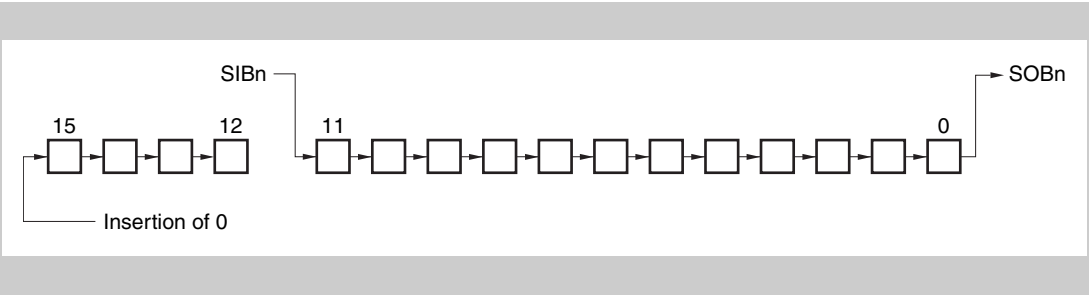


Figure 12-3 (ii) Transfer bit length = 12 bits, LSB first

(4) CBnSTR - CSIBn status register

CBnSTR is an 8-bit register that displays the CSIBn status.

This register can be read or written in 8-bit or 1-bit units, but the CBnTSF flag is read-only.

Reset input clears this register to 00H.

In addition to reset input, the CBnSTR register can be initialized by clearing (0) the CBnCTL0.CBnPWR bit.

After reset 00H R/W Address: CB0STR FFFFFD03H, CB1STR FFFFFD23H

	<7>	6	5	4	3	2	1	<0>
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

CBnTSF	Communication status flag
0	Communication stopped
1	Communicating
• During transmission, this register is set when data is prepared in the CBnTX0 register, and during reception, it is set when a dummy read of the CBnRX0 register is performed. When transfer ends, this flag is cleared to 0 at the last edge of the clock.	

CBnOVE	Overrun error flag
0	No overrun
1	Overrun
• An overrun error occurs when the next reception starts without performing a CPU read of the value of the receive buffer, upon completion of the receive operation. The CBnOVE flag displays the overrun error occurrence status in this case. • The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.	

Note In case of an overrun error, the reception error interrupt INTCBnRE behaves different, depending on the transfer mode:

- Continuous transfer mode

  The reception error interrupt INTCBnRE is generated instead of the reception completion interrupt INTCBnR.
- Single transfer mode

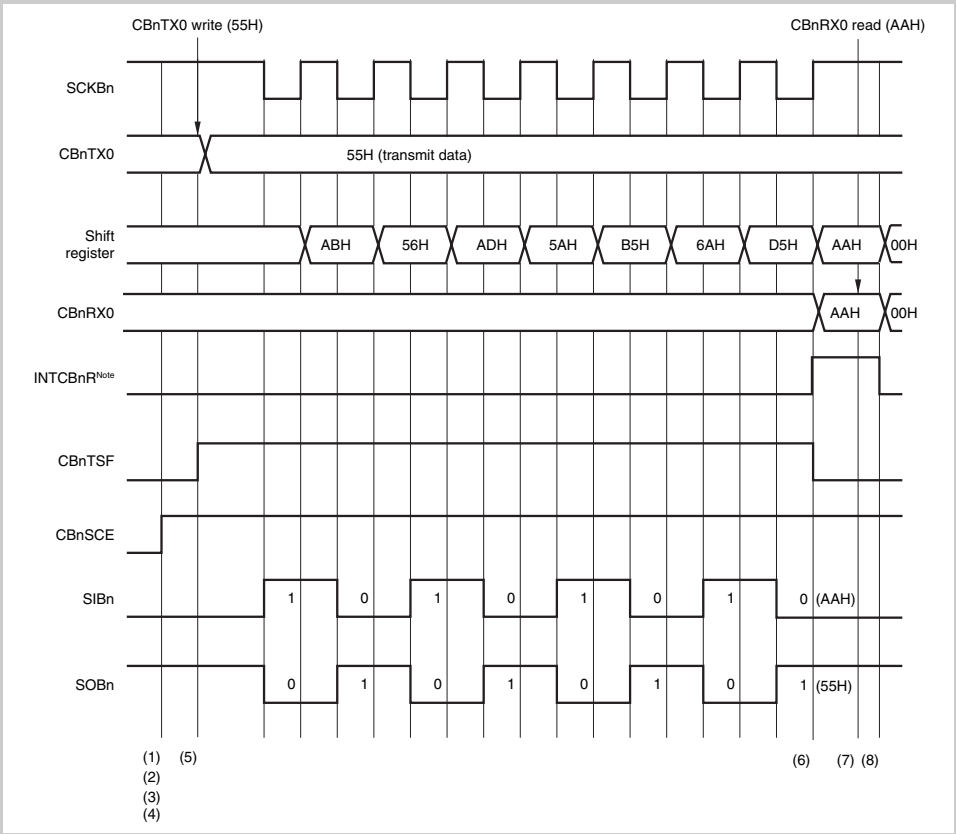
  No interrupt is generated.

In either case the overflow flag CBnSTR.CBnOVE is set to 1 and the previous data in CBnRX0 will be overwritten with the new data.

12.4 Operation

12.4.1 Single transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR 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)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnTXE, CBnRXE, and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Write transfer data to the CBnTX0 register (transmission start).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. Read the CBnRX0 register before clearing the CBnPWR bit to 0.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

In transmission mode or transmission/reception mode, communication is not started by reading the CBnRX0 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:

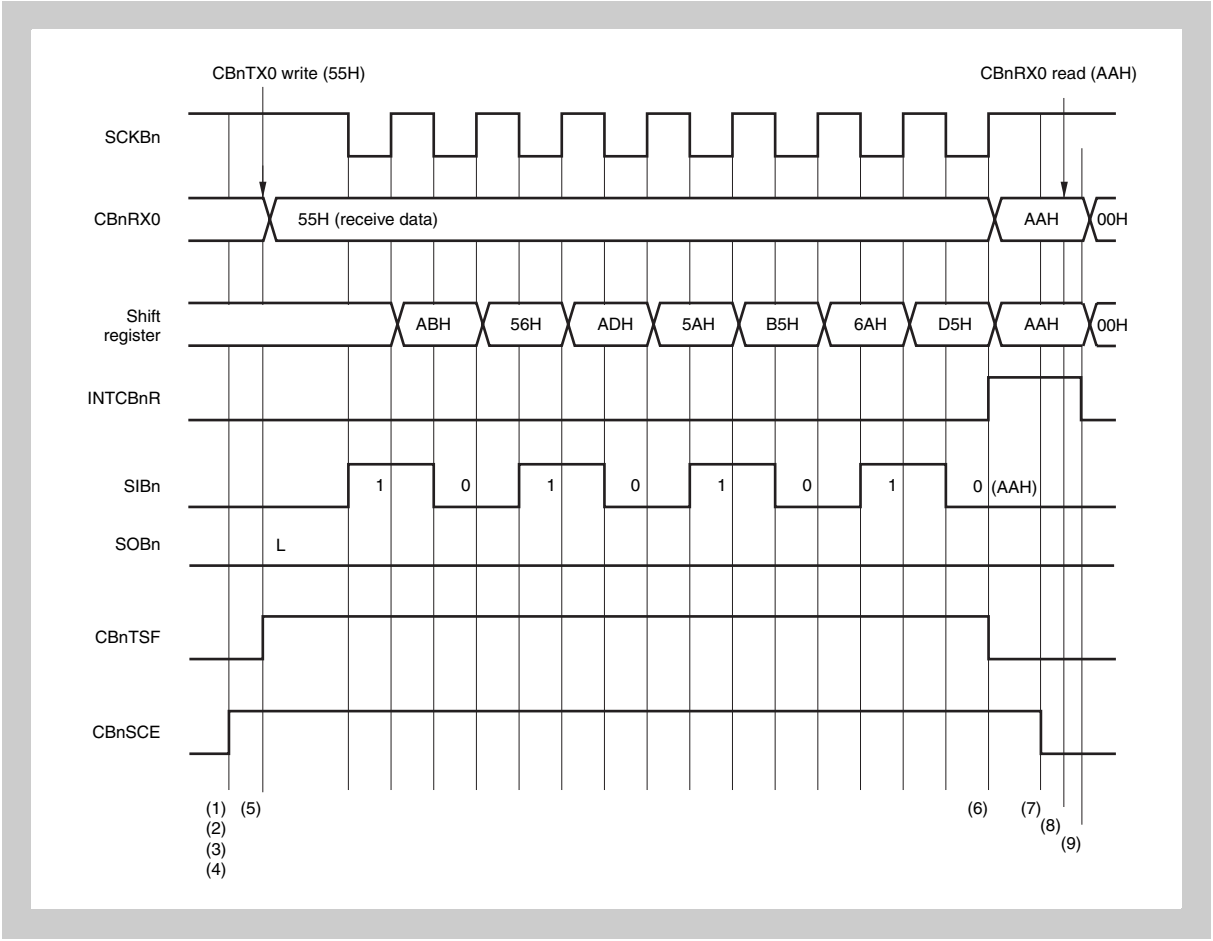
In case the next transmit should be initiated immediately after the occurrence of the reception completion interrupt INTCBnR any write to the CBnTX0 register is ignored as long as the communication status flag is still reflecting an ongoing communication (CBnTSF = 1). Thus the new transmission will not be started.

For 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 CBnTX0 register.

12.4.2 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)



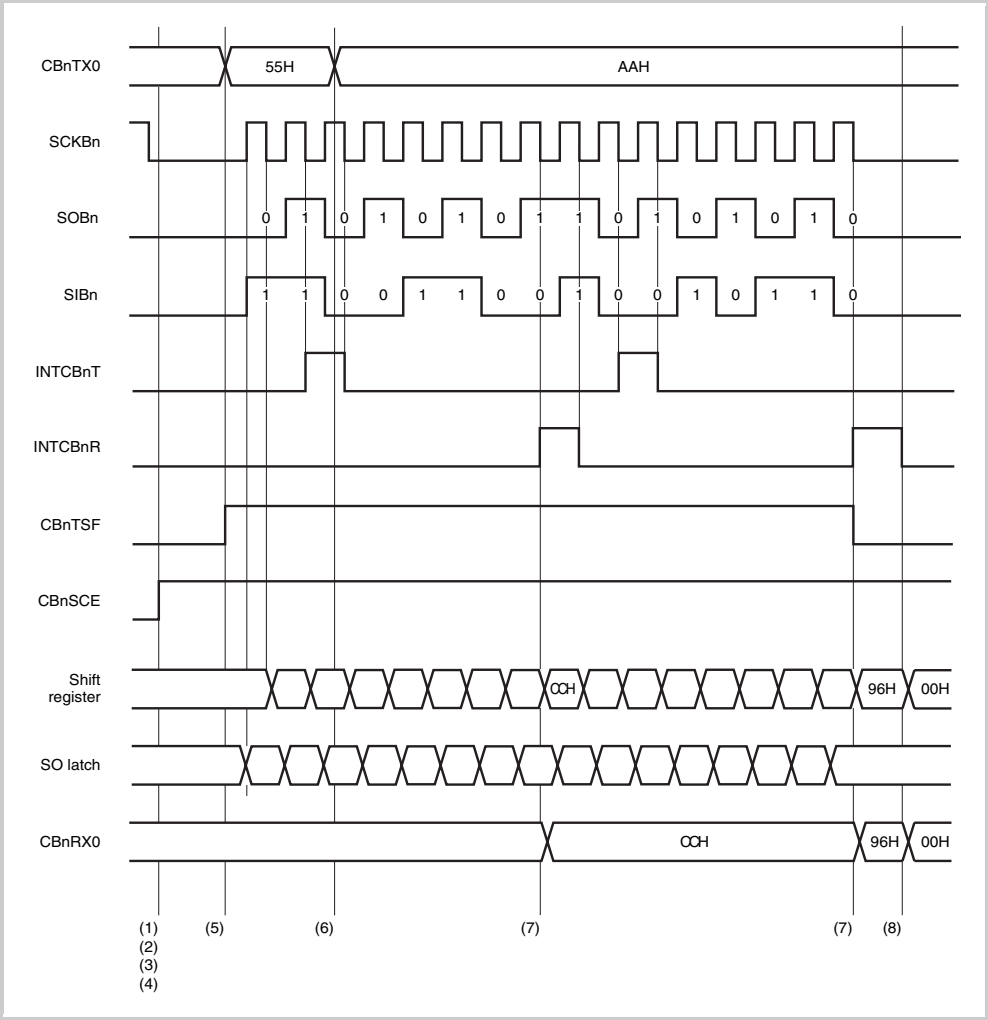
1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1, CBnCTL0.TXE to 0, at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. Set the CBnSCE bit to 0 to set the final receive data status.
8. Read the CBnRX0 register.
9. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the CSIBn operation (end of reception).

To continue transfer, repeat steps (5) and (6) before (7). (At this time, (5) is not a dummy read, but a receive data read combined with the reception trigger.)

Note The processing of steps (3) and (4) can be set simultaneously.

12.4.3 Continuous mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR 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)



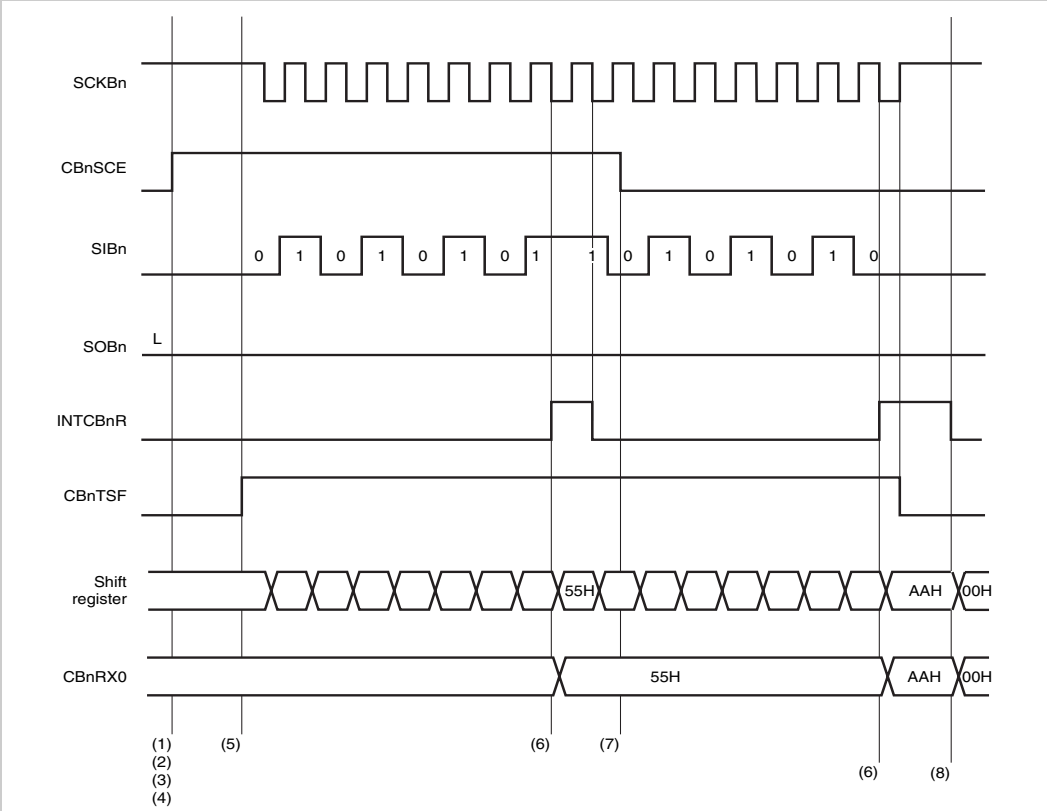
1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnTXE, CBnRXE, and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Write transfer data to the CBnTX0 register (transmission start).
6. The transmission enable interrupt request signal (INTCBnT) is received and transfer data is written to the CBnTX0 register.
7. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

In transmission mode or transmission/reception mode, the communication is not started by reading the CBnRX0 register.

12.4.4 Continuous mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR 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)

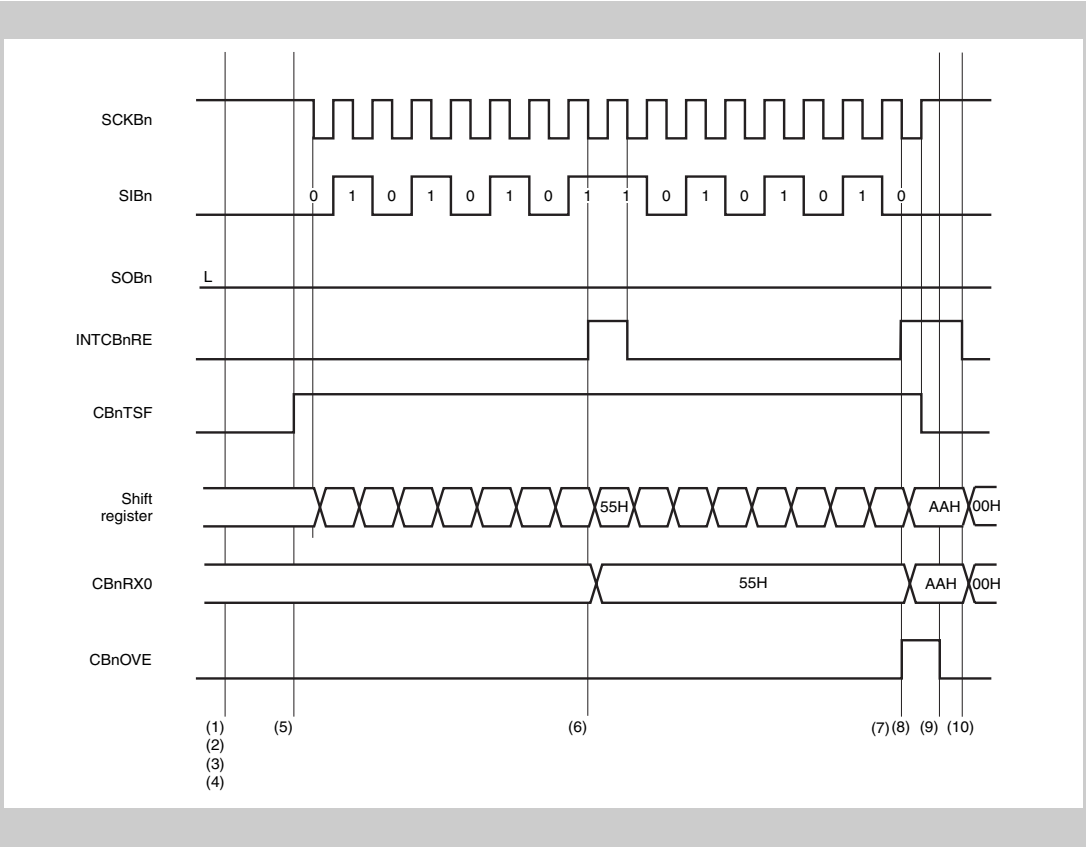


1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
7. Set the CBnCTL0.CBnSCE bit = 0 while the last data being received to set the final receive data status.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

12.4.5 Continuous reception mode (error)

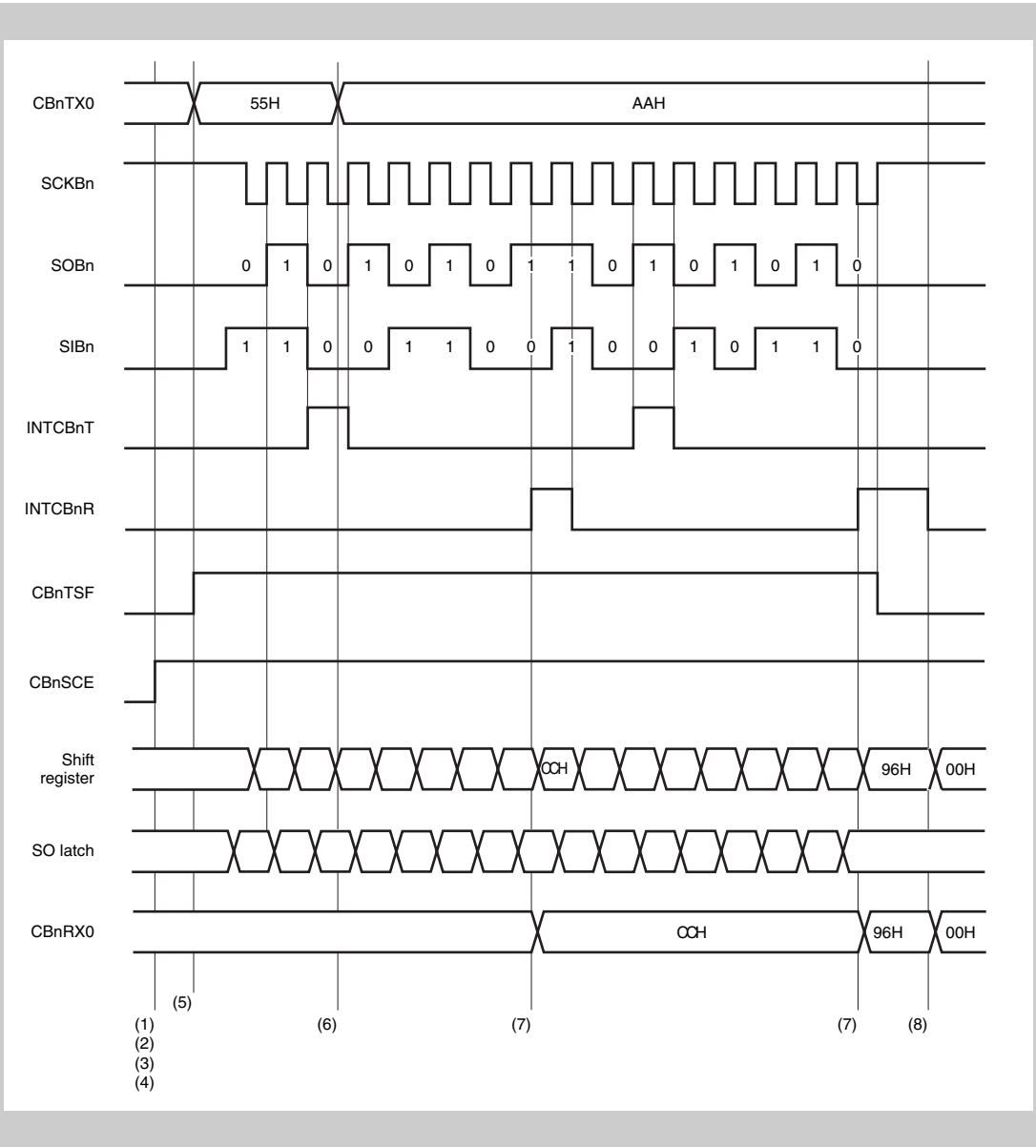
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (see 16.4 (2) CSIBn control register 1 (CBnCTL1)), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit = 1 to enable CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. If the data could not be read before the end of the next transfer, the CBnSTR.CBnOVE flag is set to 1 upon the end of reception and the INTCBnR signal is output.
8. Overrun error processing is performed after checking that the CBnOVE bit = 1 in the INTCBnRE interrupt servicing.
9. Clear CBnOVE bit to 0.
10. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation CSIBn (end of reception).

12.4.6 Continuous mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR 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)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnTXE, CBnRXE and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
4. Set the CBnPWR bit to 1 to enable supply of the CSIBn operation.
5. Write the transfer data to the CBnTX0 register.
6. The transmission enable interrupt request signal (INTCBnT) is received and the transfer data is written to the CBnTX0 register.
7. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register.

8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

Discontinued transmission

In case the CSIB is operating in continuous slave transmission mode (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 12-4 outlines this behaviour.

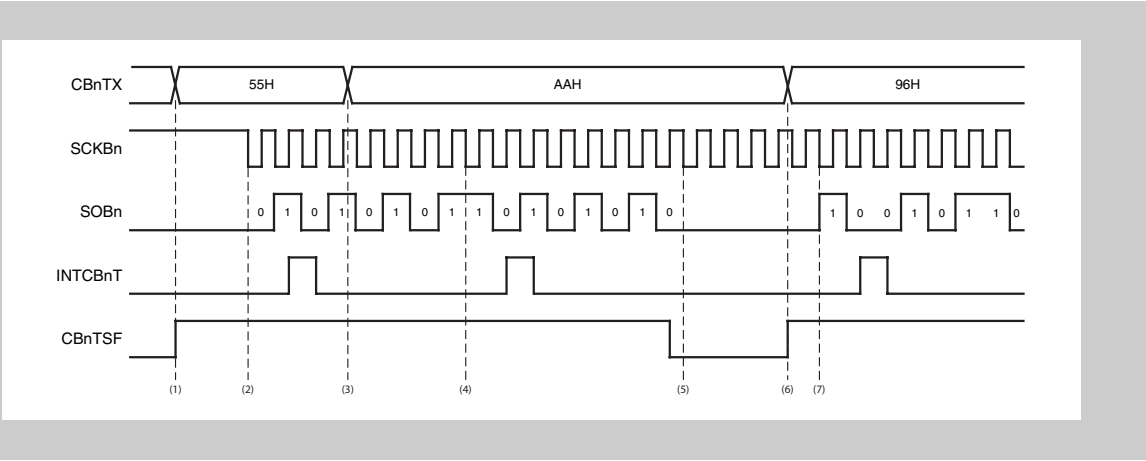


Figure 12-4 Discontinued slave transmission

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.

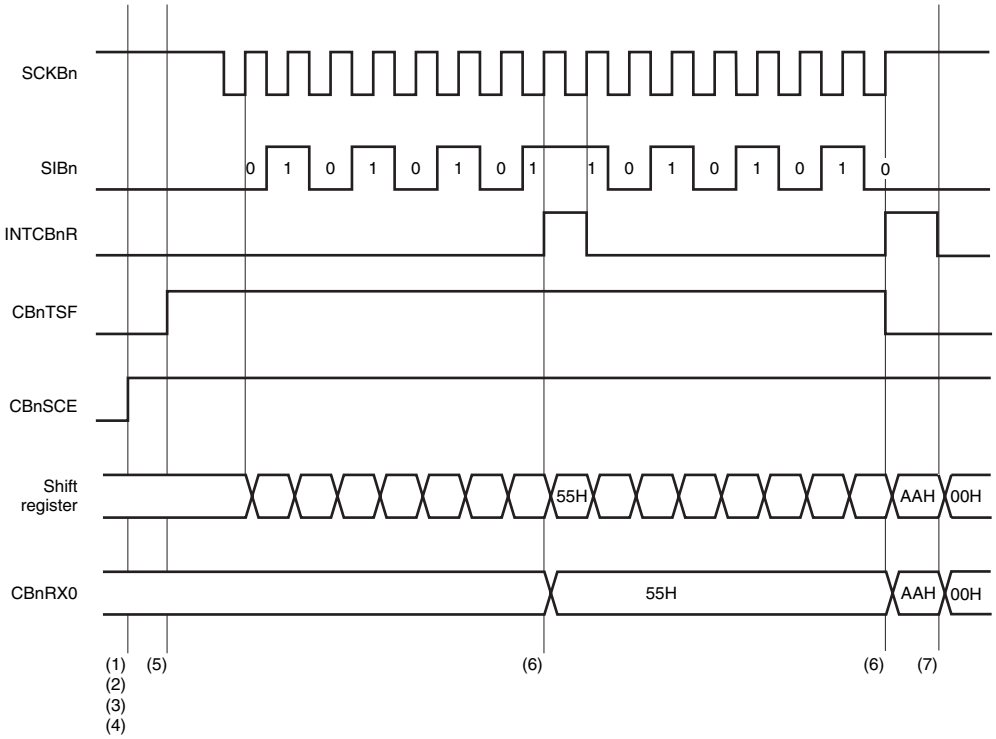
(6) New data (96_H) is written to CBnTX.

(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.

12.4.7 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)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit = 1 to enable CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register.
7. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

12.4.8 Clock timing

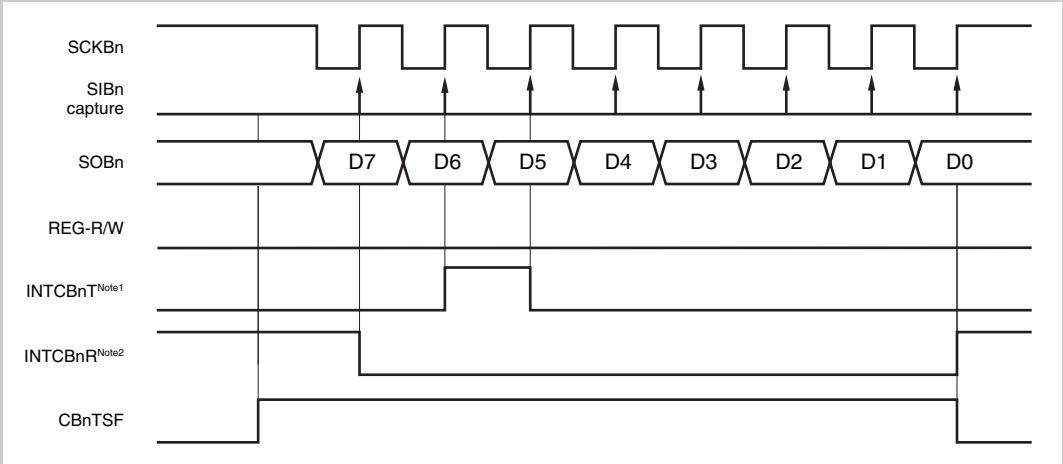


Figure 12-5 (i) Communication type 1 (CBnCKP = 0, CBnDAP = 0)

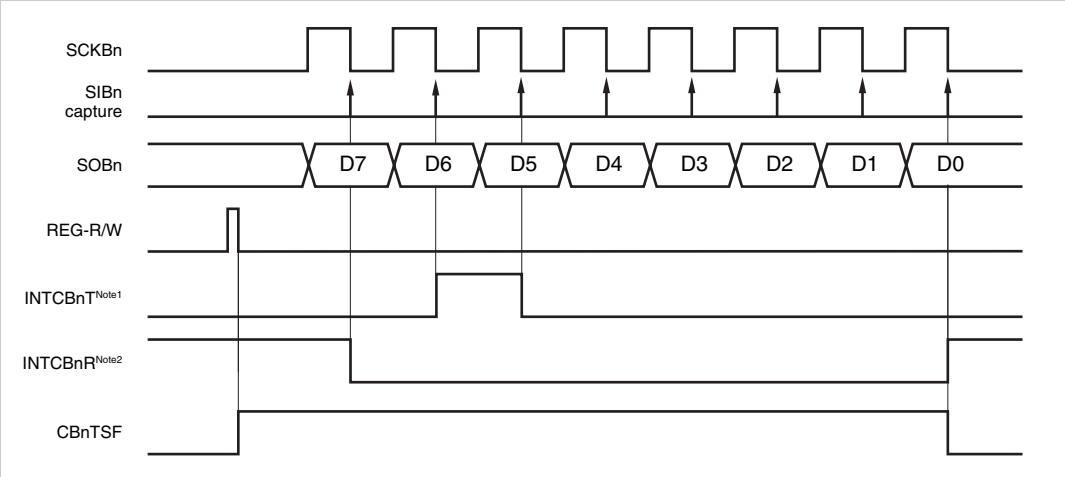


Figure 12-6 (ii) Communication type 3 (CBnCKP = 1, CBnDAP = 0)

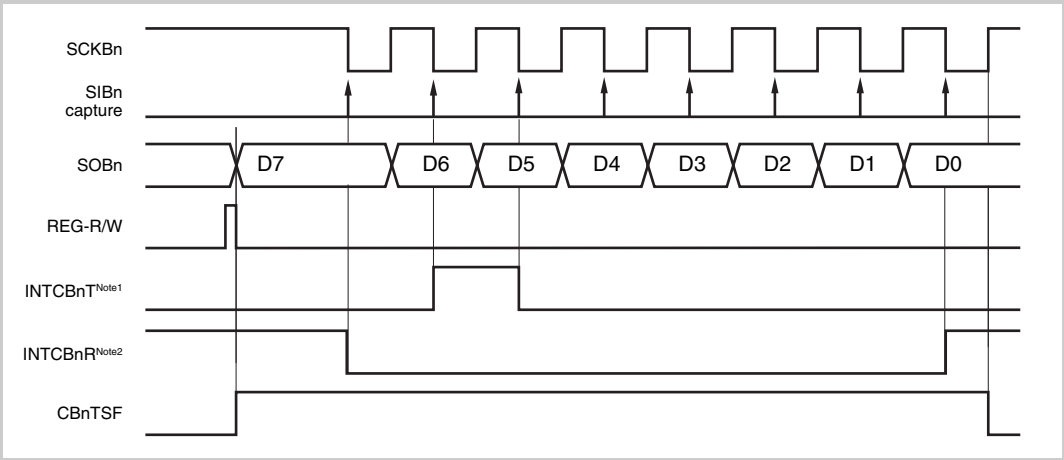


Figure 12-7 (iii) Communication type 2 (CBnCKP = 0, CBnDAP = 1)

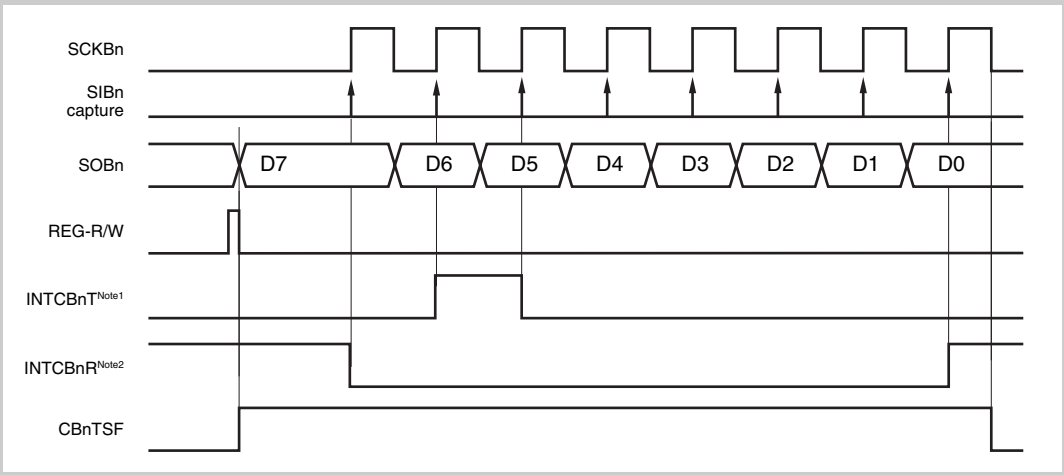


Figure 12-8 (iv) Communication type 4 (CBnCKP = 1, CBnDAP = 1)

Note 1. The INTCBnT interrupt is set when the data written to the transmit buffer is transferred to the data shift register in the continuous transmission or continuous transmission/reception modes. In the single transmission or single transmission/reception modes, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon completion of communication.

2. The INTCBnR interrupt occurs if reception is correctly completed and receive data is ready in the CBnRX0 register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

12.5 Output Pins

(1) SCKBn pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the SCKBn pin output status is as follows.

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	SCKBn pin output
0	Don't care	Don't care	Don't care	Fixed to high level
1	1	1	1	High impedance
	Other than above			Fixed to low level

Note The output level of the SCKBn pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

(2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

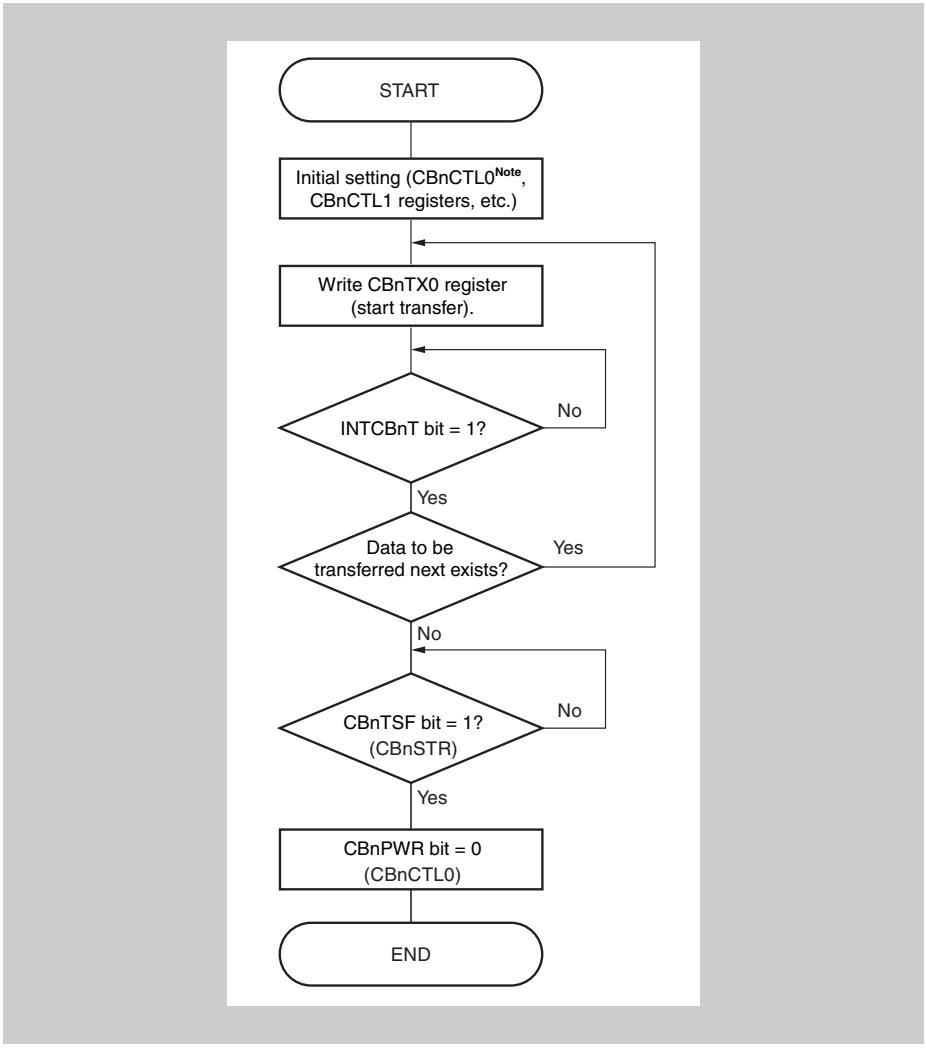
CBnTXE	CBnDAP	CBnDIR	SOBn pin output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTX0 value (MSB)
		1	CBnTX0 value (LSB)

Note 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR bits, and CBnCTL1.CBnDAP bit is rewritten.

2. ×: don't care

12.6 Operation Flow

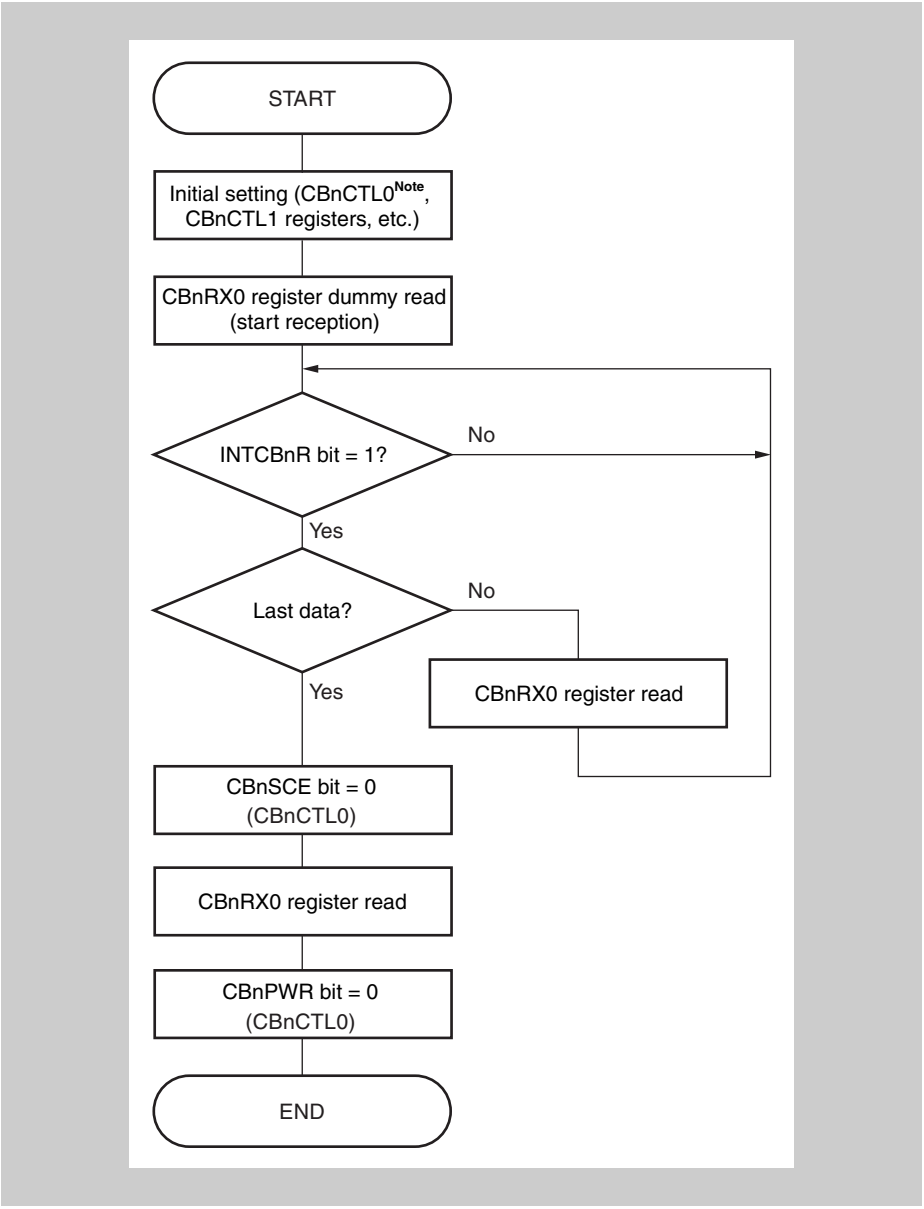
(1) Single transmission



Note Set the CBnSCE bit to 1 in the initial setting.

Caution In the slave mode, data cannot be correctly transmitted if the next transfer clock is input earlier than the CBnTX0 register is written.

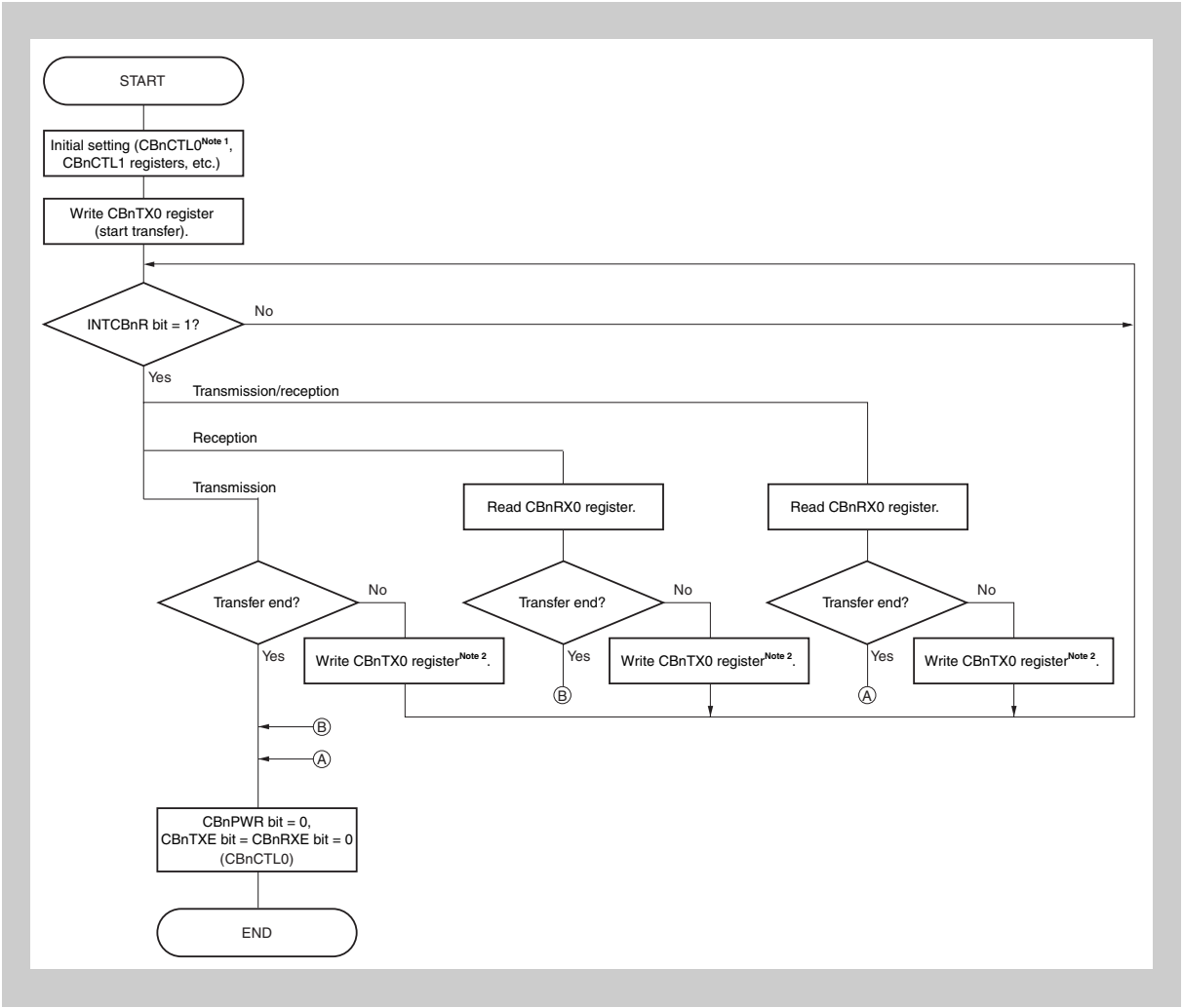
(2) Single reception



Note Set the CBnSCE bit to 1 in the initial setting.

Caution In the single mode, data cannot be correctly received if the next transfer clock is input earlier than the CBnRX0 register is read.

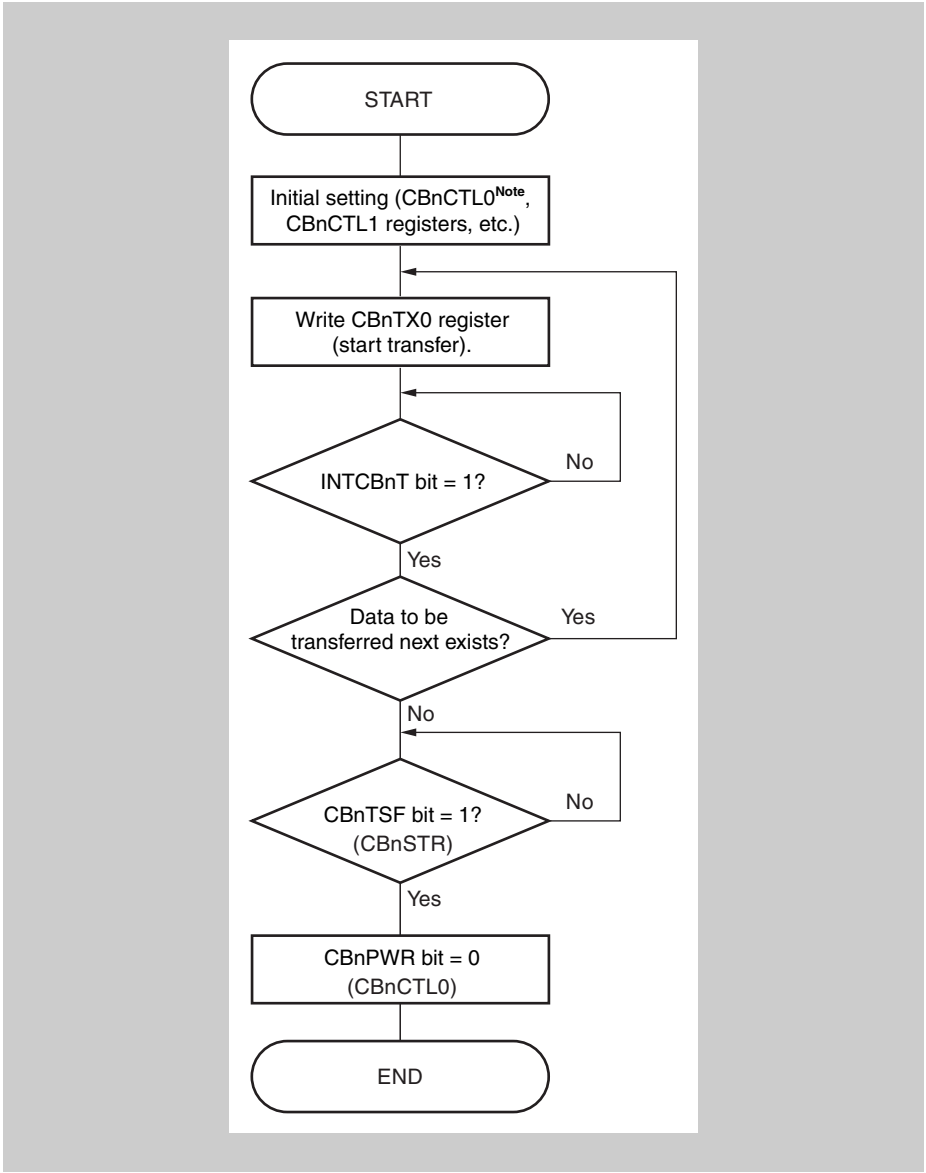
(3) Single transmission/reception



Note
1. Set the CBnSCE bit to 1 in the initial setting.
2. If the next transfer is reception only, dummy data is written to the CBnTX0 register.

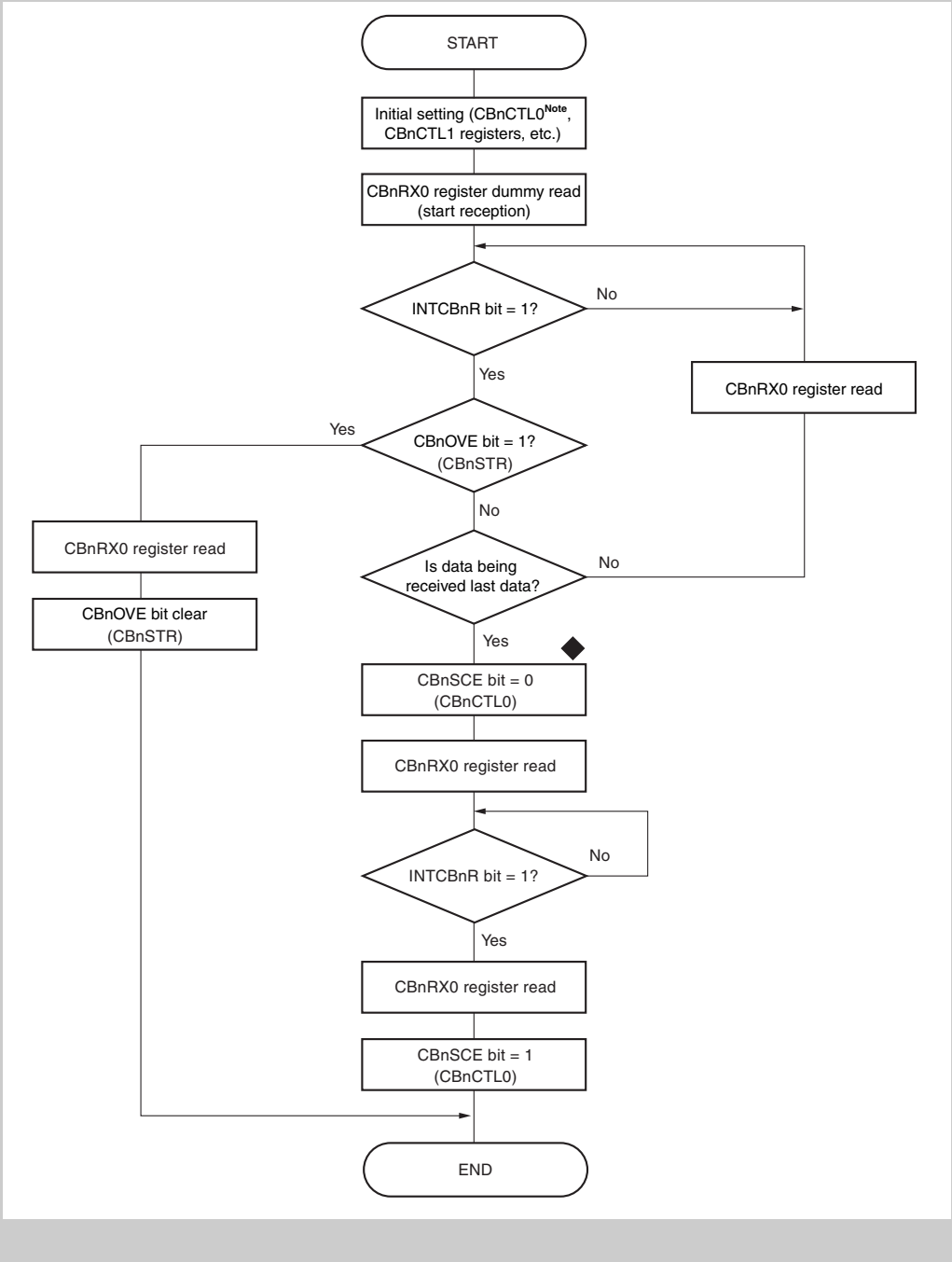
Caution Even in the single mode, the CBnSTR.CBnOVE flag is set to 1. If only transmission is used in the transmission/reception mode, therefore, programming without checking the CBnOVE flag is recommended.

(4) Continuous transmission



Note Set the CBnSCE bit to 1 in the initial setting.

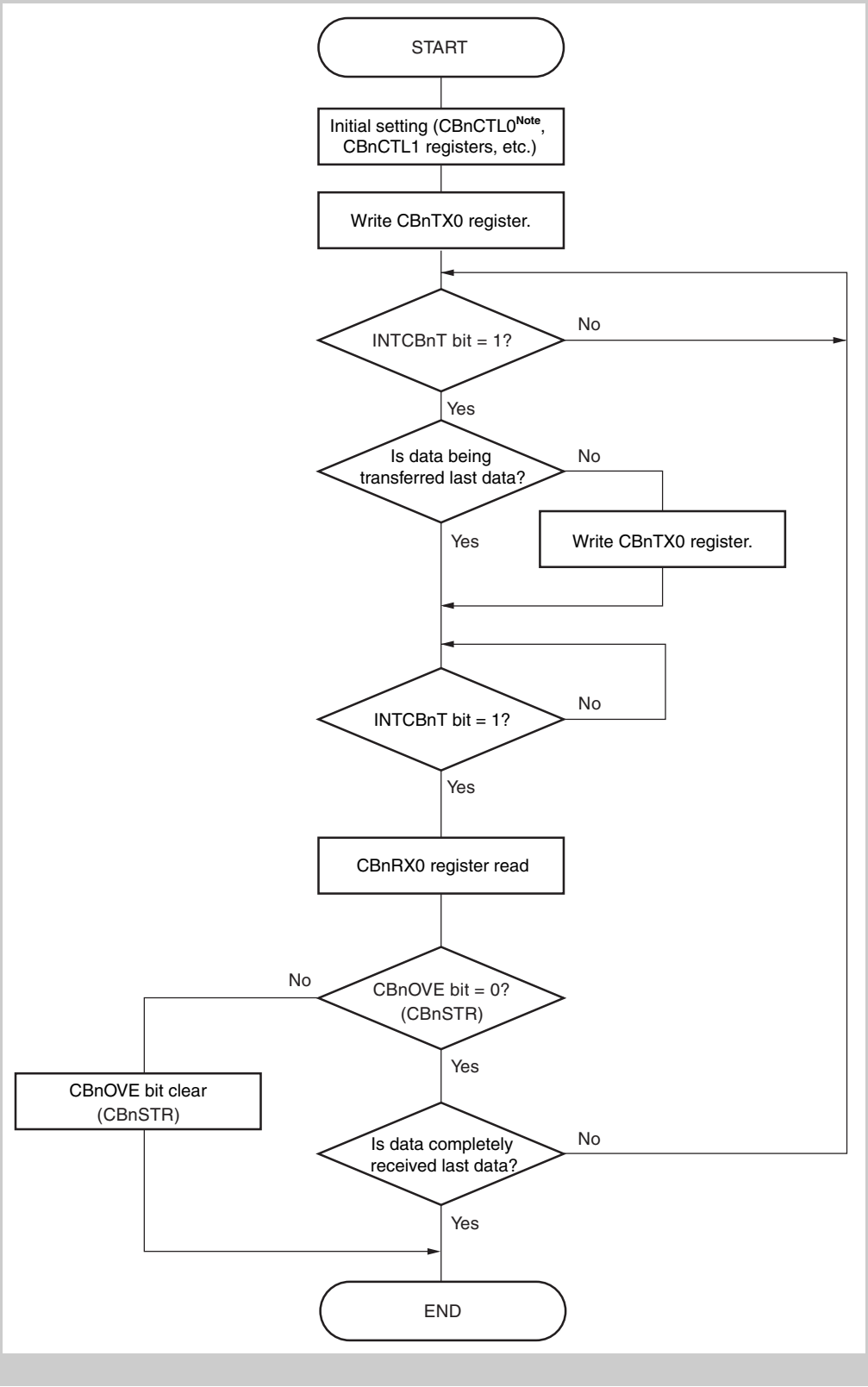
(5) Continuous reception



Note Set the CBnSCE bit to 1 in the initial setting.

Caution In the master mode, the clock is output without limit when dummy data is read from the CBnRX0 register. To stop the clock, execute the flow marked ◆ in the above flowchart.
In the slave mode, malfunction due to noise during communication can be prevented by executing the flow marked ◆ in the above flowchart.
Before resuming communication, set the CBnCTL0.CBnSCE bit to 1, and read dummy data from the CBnRX0 register.

(6) Continuous transmission/reception

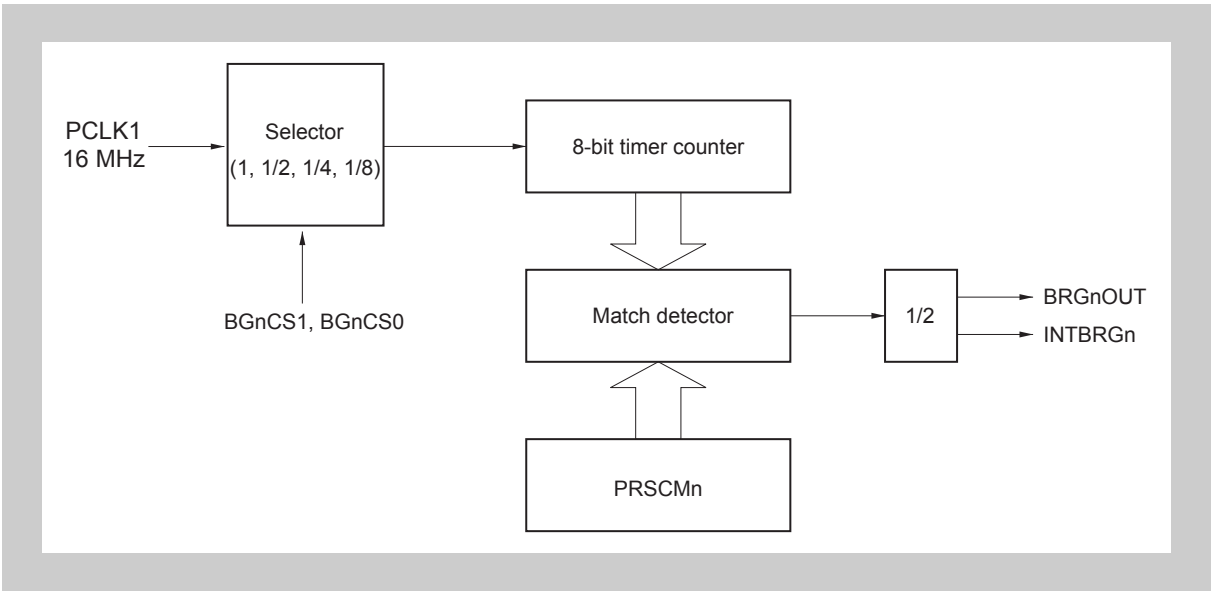


Note Set the CBnSCE bit to 1 in the initial setting.

12.7 Baud Rate Generator

12.7.1 Overview

Each CSIBSn interface is equipped with a dedicated 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.

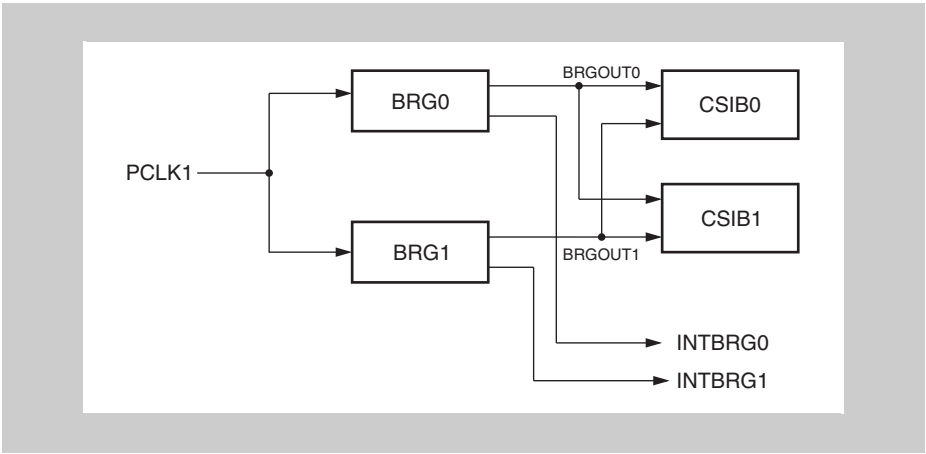


Figure 12-9 Block diagram of CSIBn baud rate generators

Note An unused baud rate generator (BRGm) can be employed as interval timer generating a dedicated interrupt request (INTBRGm).

12.7.2 Baud rate generator registers

(1) Prescaler mode registers (PRSMn)

The PRSMn registers control generation of the baud rate signal for CSIB.

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

Reset input clears these registers to 00H.

After reset: 00H R/W Address: PRSM0 FFFFFDC0H, PRSM1 FFFFFDD0H

	7	6	5	4	3	2	1	0
PRSMn	0	0	0	BGCEn	0	TODISn	BGCSn1	BGCSn0

BGCEn	TODISn	Baud rate generator output		
		BRGOUT	INTBRG	Counter
0	0	Fixed on 0	Fixed on 0	Fixed on 01H
0	1	Fixed on 0	Fixed on 0	Fixed on 01H
1	0	Active	Active	Active
1	1	Fixed on 0	Active	Active

BGCSn1	BGCSn0	Input clock selection (f_{BGCSn})	Setting value (k)
0	0	PCLK1 (16 MHz)	0
0	1	PCLK1 : 2 (8 MHz)	1
1	0	PCLK1 : 4 (4 MHz)	2
1	1	PCLK1 : 8 (2 MHz)	3

Caution
1. Do not rewrite the PRSMn register during operation.
2. Set bits BGCSn0, BGCSn1 and TODISn before setting the BGCEn bit to 1.

(2) Prescaler compare registers (PRSCMn)

The PRSCMn registers are 8-bit compare registers.

These registers can be read or written in 8-bit units.

Reset input clears these registers to 00H.

After reset: 00H R/W Address: PRSCM0 FFFFFDC1H, PRSCM1 FFFFFDD1H

	7	6	5	4	3	2	1	0
PRSCMn	PRSCMn7	PRSCMn6	PRSCMn5	PRSCMn4	PRSCMn3	PRSCMn2	PRSCMn1	PRSCMn0

Caution
1. Do not rewrite the PRSCMn register during operation.
2. Set the PRSCMn register before setting the PRSMn.BGCEn bit to 1.

12.7.3 Baud rate generation

The transmission/reception clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{BRGn} = \frac{PCLK1}{2^k \times N \times 2}$$

f_{BRGn}: BRGn output clock

PCLK1: 16 MHz peripheral clock (refer to *"Clock Generator" on page 179*)

k: PRSMn.BGCSn[1:0] register setting value ($0 \leq k \leq 3$)

N: PRSCMn.PRSCMn[7:0] register value
if PRSCMn = 00H: N = 256

12.8 Cautions

12.8.1 CSIB operation stop

Description If any channel of CSIBn is operated in slave mode and an external clock signal is input via the SCKBn pin while no transmission or reception sequence is in progress the CSIB may stop operating.

Depending on the CSIB operating configuration the CSIB behaves as described below.

- Transmit mode or transmit/receive mode:
  In transmit mode (CBnCTL0.CBnTXE = 1, CBnCTL0.CBnRXE = 0) or transmit/receive mode (CBnCTL0.CBnTXE = 1, CBnCTL0.CBnRXE = 1) a write operation to the CBnTX0 register may trigger the aforementioned behaviour.
  Any write to the related CBnTX0 register will no longer start a transmission sequence. Furthermore the related transmission interrupt request will not be generated.
- Receive mode:
  In receive mode (CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1) a read operation from the CBnRX0 register may trigger the aforementioned behaviour.
  Any read from the related CBnRX0 register will no longer start a receive sequence. Furthermore the related receive interrupt request will not be generated.

The described CSIBn stop condition can be escaped by initiating a system reset or by a sequential clear and set of the CBnCTL0.CBnPWR bit.

Workaround In order to avoid the CSIBn stop condition in slave mode take the following precautions.

- Transmit mode or transmit/receive mode:
  Make sure the external clock via the SCKBn pin is not input while writing to the CBnTX0 register after a transmission sequence is finished.

- Receive mode:
  Make sure the external clock via the SCKBn pin is not input While reading from the CBnRX0 register after a reception sequence is finished.

Chapter 13 Enhanced Queued Clocked Serial Interface (CSIE)

This microcontroller has two instances of the Enhanced Queued Clocked Serial Interface CSIE: CSIE0 and CSIE1.

Note Throughout this chapter, the individual instances of CSIE are identified by “n” (n = 0 to 1), for example CEnCTL2 for the Queued CSIn control register 2.

The individual chip select signals are identified by “m” (m = 0 to 7 for CSIE0 and m = 0 to 3 for CSIE1), for example SCSE06 for the chip select pin 6 of CSIE0.

13.1 Features

- 3-wire serial synchronous transfers
- The following pins are provided to enable a 3-wire serial interface:

Pin Function	CSIE0	CSIE1
Serial data output	SOE0	SOE1
Serial data input	SIE0	SIE1
Serial clock I/O	SCKE0	SCKE1
Chip select	SCSE00 to SCSE07	SCSE10 to SCSE13

- Master mode and slave mode selectable
- Serial clock and data phase selectable
- Transfer date length selectable from 8 to 16 bits in 1-bit units
- Data transfer with MSB- for LSB-first selectable
- Three selectable transfer modes:
  - transmit only mode
  - receive only mode
  - transmit/receive mode
- Transmit and receive FIFO (16 elements)
- Selection between single transfer mode and block transfer mode
- Internal baud rate generator
- Programmable baud rate through BRG output (master) or slave clock
- DMA transfer of received data to memory available
- Maximum SCKE frequency: 8 MHz

Note In all figures of this chapter, except where explicitly mentioned, it is assumed that the chip select signals SCSEn0 to SCSEn7 are set as “active low”, although it is possible to define the active level for each chip select also to “active high”.

13.1.1 Queued CSI block diagram

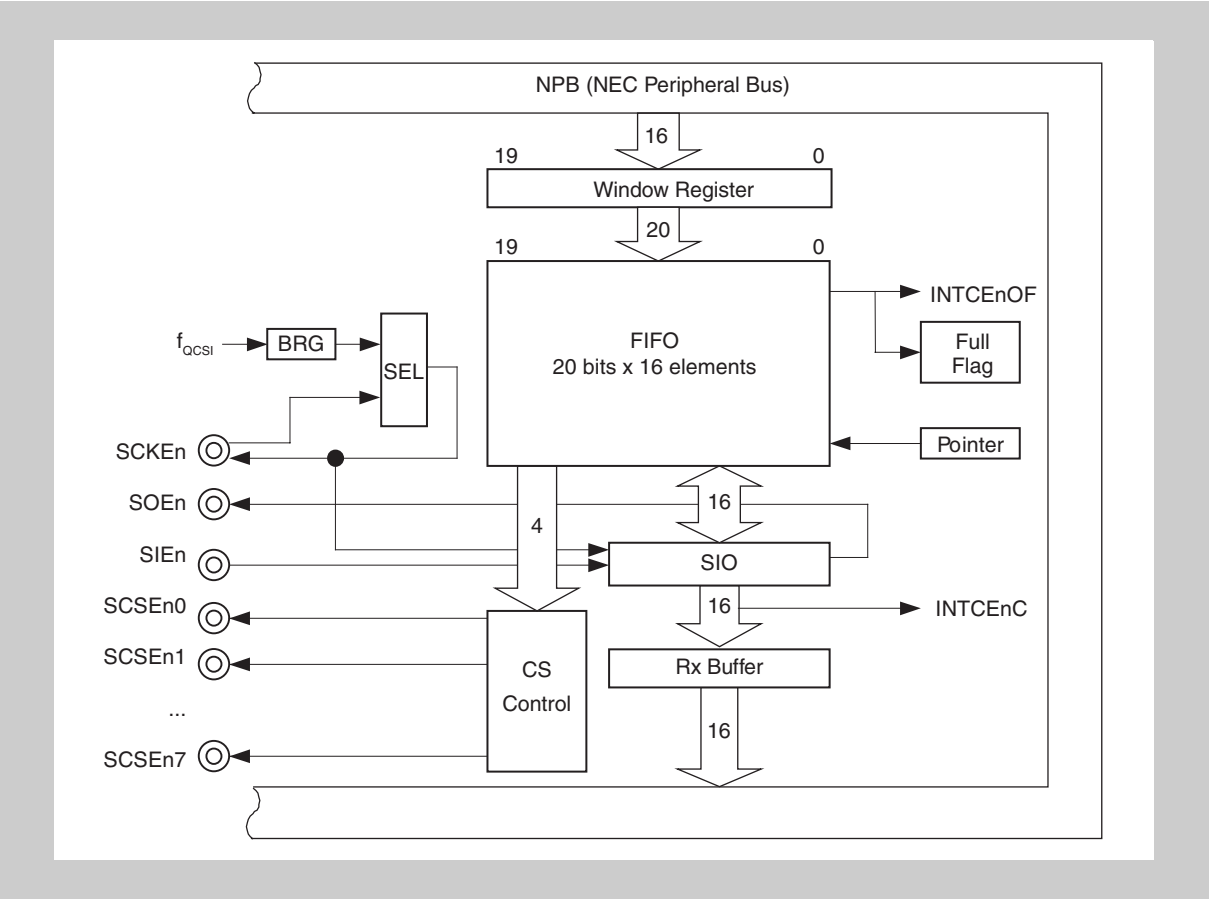


Figure 13-1 Queued CSI Block Diagram

Note 1. CSIE1 offers only four chip select pins: SCSE10 to SCSE13.

2. $f_{QCSI} = f_{PCLK0} = 32$ MHz is the clock supply of the Queued-CSI macro.

13.1.2 Input/Output pins

The table below shows the input/output pins of the CSIE.

Table 13-1 Input/Output Pins of the CSIE

Signal name	I/O	Active level	Disabled level	Function
SCKEn	I/O	–	H	Serial clock signal
SIEn	I	–	–	Input serial data signal
SOEn	O	–	–	Output serial data signal
SCSEn0	O	L[a]	H[a]	Serial peripheral chip select signal 0
SCSEn1	O	L[a]	H[a]	Serial peripheral chip select signal 1
SCSEn2	O	L[a]	H[a]	Serial peripheral chip select signal 2
SCSEn3	O	L[a]	H[a]	Serial peripheral chip select signal 3
SCSEn4[b]	O	L[a]	H[a]	Serial peripheral chip select signal 4
SCSEn5[b]	O	L[a]	H[a]	Serial peripheral chip select signal 5
SCSEn6[b]	O	L[a]	H[a]	Serial peripheral chip select signal 6
SCSEn7[b]	O	L[a]	H[a]	Serial peripheral chip select signal 7

a) The active level is programmable for each chip select.
b) Not available for CSIE1.

13.2 Queued CSI Control Registers

The Enhanced Queued CSIEs are controlled and operated by means of the following registers:

Table 13-2 CSIEn register overview

Register name	Shortcut	Address
Queued CSI control register 0	CEnCTL0	<base>
Queued CSI control register 1	CEnCTL1	<base> + 01_H
Receive data buffer	CEnRX0	<base> + 02_H
Chip Select FIFO buffer	CEnCS	<base> + 04_H
Transmit data FIFO buffer	CEnTX0	<base> + 06_H
CSI Buffer Status Register	CEnSTR	<base> + 08_H
Queued CSI control register 2	CEnCTL2	<base> + 09_H
Queued CSI control register 3	CEnCTL3	<base> + $0C_H$
Queued CSI control register 4	CEnCTL4	<base> + $0D_H$
CSEn0 enhanced timing	CEnOPT0	<base> + 10_H
CSEn1 enhanced timing	CEnOPT1	<base> + 12_H
CSEn2 enhanced timing	CEnOPT2	<base> + 14_H
CSEn3 enhanced timing	CEnOPT3	<base> + 16_H
CSEn4 enhanced timing[a]	CEnOPT4	<base> + 18_H
CSEn5 enhanced timing[a]	CEnOPT5	<base> + $1A_H$
CSEn6 enhanced timing[a]	CEnOPT6	<base> + $1C_H$
CSEn7 enhanced timing[a]	CEnOPT7	<base> + $1E_H$

a) only valid for CSIE0

Table 13-3 CSIEn register base address

CSIE module	Base address
CSIE0	FFFF $FD40_H$
CSIE1	FFFF $FD80_H$

(1) CEnCTL0 - Queued CSI control register 0

The CEnCTL0 registers control the Queued CSI macro's operations. It can be read or written in 1-bit and 8-bit units.

The CEnTMS, CEnDIR, CEnSIT, CEnWE, CEnCSM bits can only be written when CEnTXE = 0 and CEnRXE = 0.

Initial value is 00H by reset.

Address: CE0CTL0=FFFFFD40H, CE1CTL0=FFFFFD80H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CEnCTL0	CEn PWR	CEn TXE	CEn RXE	CEn TMS	CEn DIR	CEn SIT	CEn WE	CEn CSM	R/W	00H

CEnPWR	Queued CSI operation clock control
0	Stop macro operation clock (Reset internal control circuits)
1	Provide macro operation clock
Clearing CEnPWR = “0” resets the internal circuits asynchronously, stops operation and sets the Queued CSI to standby state. Input clock is not provided to internal circuits. Set CEnPWR = “1” to activate the Queued CSI.	

Caution When changing the CEnPWR bit, do not change any other bit at the same time.
While CEnPWR=”0”, the only registers that can be accessed are CEnCTL0, CEnTX0, CEnTX0L, and CEnSTR.
Set the CEnPWR bit before writing any of the other bits of CEnCTL0.

CEnTXE	Transmission enable/disable
0	Transmission disabled
1	Transmission enabled

CEnRXE	Receive enable/disable
0	Receive disabled
1	Receive enabled

CEnTMS	Transfer mode select
0	Single transfer mode
1	Block transfer mode

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

CEnDIR	Serial data direction selection
0	Data is sent/received with MSB first
1	Data is sent/received with LSB first

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

See *"Serial data direction select function" on page 395* for further details on the CEnDIR bit setting.

CEnSIT	Interrupt delay mode select (INTCEnC signal)
0	No delay
1	Half clock delay

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.
This bit is only valid in master mode. In slave mode, no delay is generated.

CEnWE	Transmission wait enable/disable select [Note]
0	Transmission wait disable. Not insert half clock wait at transmission start.
1	Transmission wait enable. Insert 1 clock (SCKE) wait at transmission start.

Note This bit has no effect when CEnCTL4.CEnOPE = 1.

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.
This bit is only valid in master mode. In slave mode, no wait is generated.

CEnCSM	Chip Select mode select [Note]
0	Chip select inactive level output disable. Do not force chip select inactive state after each transfer of a data element.
1	Chip select inactive level output enable. Hold all chip selects inactive for half-length SCKE after each transfer of a data element.

Note This bit has no effect when CEnCTL4.CEnOPE = 1.

Caution Write is permitted only when CEnTXE = 0 and CEnRXE= 0.
This bit is only valid for CEnWE=1.
This bit is only valid in master mode. In slave mode, CS signals are always held at inactive level.

In combination:

CEnWE	CEnCSM	Transmission wait	Chip Select inactive level
0	0	None	Not output
0	1	None	Not output
1	0	One SCKE length clock wait	Not output
1	1	One SCKE length clock wait	Output inactive level of half-length SCKE at first half of transmission wait

See *"Additional timing and delay selections" on page 403* for further details on the timing selections by CEnSIT, CEnWE, CEnCSM bits.

(2) CEnCTL1 - Queued CSI control register 1

The CEnCTL1 register is an 8-bit register that is used to control the serial transfer operations. It can be read or written in 1-bit and 8-bit units.

Initial value is 07H by reset.

Caution This register can be written only while CEnTXE = 0 and CEnRXE = 0.

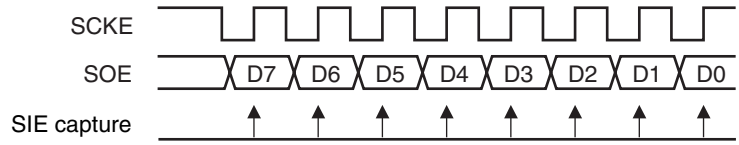
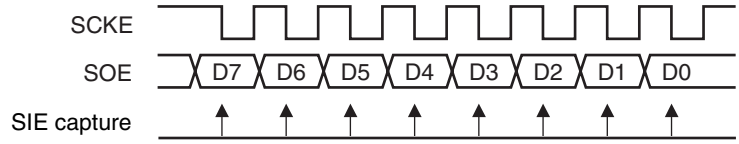
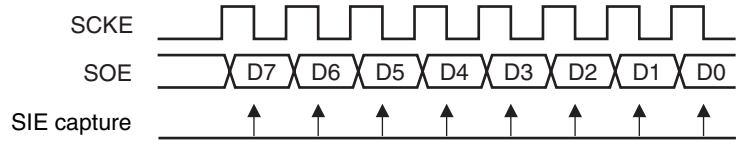
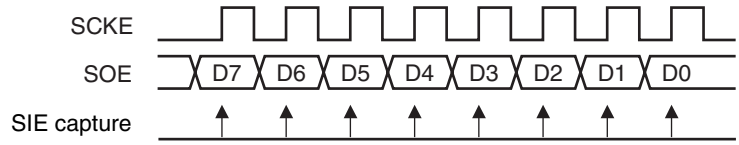
Address: CE0CTL1=FFFFFD41H, CE1CTL1=FFFFFD81H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CEnCTL1	CEn MDL2	CEn MDL1	CEn MDL0	CEn CKP	CEn DAP	CEn CKS2	CEn CKS1	CEn CKS0	R/W	07H

CEnMDL2	CEnMDL1	CEnMDL0	Operation clock	N	Comment
0	0	0	BRG disable	–	BRG stop for power saving
0	0	1	PRSOUT/2	1	See caution below
0	1	0	PRSOUT/4	2	
0	1	1	PRSOUT/6	3	
1	0	0	PRSOUT/8	4	
1	0	1	PRSOUT/10	5	
1	1	0	PRSOUT/12	6	
1	1	1	PRSOUT/14	7	
These bits are used to modulate the selected clock					

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.
CEnMDL[2:0] = [0,0,1] is prohibited when CEnCKS[2:0] = [0,0,0].

See *"Transmission clock select function" on page 398* for further explanation on the transfer clock selection.

CEnCKP	CEnDAP	Clock and Data Phase Selection
0	0	
0	1	
1	0	
1	1	

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

Note CEnCKP: Clock phase selection bit
CEnDAP: Data phase selection bit

CEnCKS2	CEnCKS1	CEnCKS0	Prescaler output (PRSOUT)	Mode	K
0	0	0	f_{QCSI}	Master Mode	0
0	0	1	$f_{QCSI}/2$	Master Mode	1
0	1	0	$f_{QCSI}/4$	Master Mode	2
0	1	1	$f_{QCSI}/8$	Master Mode	3
1	0	0	$f_{QCSI}/16$	Master Mode	4
1	0	1	$f_{QCSI}/32$	Master Mode	5
1	1	0	$f_{QCSI}/64$	Master Mode	6
1	1	1	SCKE (input) **Note**	Slave Mode	–

Note SCKE pin is configured as input mode for serial transfer clock.
f_{QCSI} is the clock supply of the Queued-CSI macro.

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

See *“Slave mode” on page 396* and *“Master mode” on page 397* for further explanation on slave mode and master mode.

The baud rate for the transmission is calculated with the following formula:

$$\text{baudrate} = \frac{f_{QCSI}}{(2 \times N) \times 2^{K}} = \frac{f_{QCSI}}{N \times 2^{(K+1)}} = \frac{f_{QCSI}}{CEnMDL \times 2^{(CEnCKS+1)}}$$

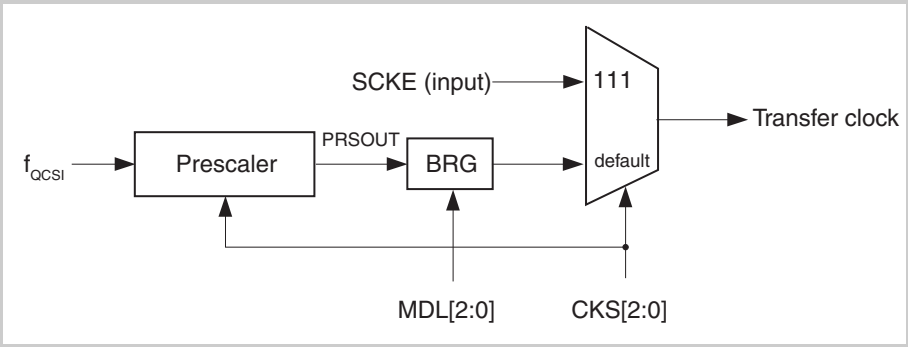


Figure 13-2 Queued CSI Baud Rate Block Diagram

(3) CEnCTL2 - Queued CSI control register 2

The CEnCTL2 register is an 8-bit register that specifies the active voltage level of the chip select pins SCSEn0 to SCSEn3 and the Queued CSI data length. It can be read or written in 1-bit and 8-bit units.

Initial value is 00H by reset.

Caution This register can be written only while CEnTXE = 0 and CEnRXE = 0.

Note To select the active level of the remaining pins, see *"CEnCTL4 - Queued CSI control register 4" on page 380*

Address: CE0CTL2=FFFFFD49H, CE1CTL2=FFFFFD89H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CEnCTL2	CEn CSL3	CEn CSL2	CEn CSL1	CEn CSL0	CEn DL3	CEn DL2	CEn DL1	CEn DL0	R/W	00H

CEnCSLm	Chip Select active level selection
0	Chip Select signal SCSEnm is active low
1	Chip Select signal SCSEnm is active high

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

The chip select settings will be reflected at the outputs immediately after the register is set.

CEnDL3	CEnDL2	CEnDL1	CEnDL0	Transfer data length
0	0	0	0	Data length is 16 bits
0	0	0	1	Setting prohibited
0	0	1	x	
0	1	x	x	
1	0	0	0	Data length is 8 bits
1	0	0	1	Data length is 9 bits
1	0	1	0	Data length is 10 bits
1	0	1	1	Data length is 11 bits
1	1	0	0	Data length is 12 bits
1	1	0	1	Data length is 13 bits
1	1	1	0	Data length is 14 bits
1	1	1	1	Data length is 15 bits

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

See *"Data length select function" on page 396* for further explanation on the data length selection.

(4) CEnCTL3 - Queued CSI Control Register 3

The CEnCTL3 register is an 8-bit register that specifies the number of data elements to be transferred in block transfer mode. It can be read or written in 1-bit and 8-bit units.

Initial value is 00H by reset.

Address: CE0CTL3=FFFFFD4CH, CE1CTL3=FFFFFD8CH

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CEnCTL3	0 Note	0 Note	0 Note	0 Note	CEn SFN3	CEn SFN2	CEn SFN1	CEn SFN0	R/W	00H

Note Unused bits must be written as 0.

CEnSFN3	CEnSFN2	CEnSFN1	CEnSFN0	Transfer data number
0	0	0	0	Transfer 16 data elements
0	0	0	1	Transfer 1 data element
0	0	1	0	Transfer 2 data elements
0	0	1	1	Transfer 3 data elements
0	1	0	0	Transfer 4 data elements
0	1	0	1	Transfer 5 data elements
0	1	1	0	Transfer 6 data elements
0	1	1	1	Transfer 7 data elements
1	0	0	0	Transfer 8 data elements
1	0	0	1	Transfer 9 data elements
1	0	1	0	Transfer 10 data elements
1	0	1	1	Transfer 11 data elements
1	1	0	0	Transfer 12 data elements
1	1	0	1	Transfer 13 data elements
1	1	1	0	Transfer 14 data elements
1	1	1	1	Transfer 15 data elements

(5) CEnCTL4 - Queued CSI control register 4

The CEnCTL4 register is an 8-bit register that specifies the active voltage level of the chip select pins SCSEn4 to SCSEn7, the alternative clock and data phase setting and that controls the enhanced chip select timing. It can be read or written in 1-bit and 8-bit units.

Initial value is 00H by reset.

Caution This register can be written only while CEnTXE = 0 and CEnRXE = 0.

Address: FFFFFD4DH

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CE0CTL4	CE0 CSL7	CE0 CSL6	CE0 CSL5	CE0 CSL4	CE0 CPA	CE0 DPA	CE0 OPE	CE0 MD	R/W	00H

Address: FFFFFD8DH

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CE1CTL4	0	0	0	0	CE1 CPA	CE1 DPA	CE1 OPE	CE1 MD	R/W	00H

CEnCSLm	Chip Select active level selection
0	Chip Select signal SCSEnm is active low
1	Chip Select signal SCSEnm is active high

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

The chip select settings will be reflected at the outputs immediately after the register is set.

CEnCPA	CEnDPA	Alternative Clock and Data phase settings
0	0	Clock and data phase as CKP=0, DAP=0
0	1	Clock and data phase as CKP=0, DAP=1
1	0	Clock and data phase as CKP=1, DAP=0
1	1	Clock and data phase as CKP=1, DAP=1

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

The CPA/DPA bits define an alternative setting for the serial communication within the same Queued-CSI macro. Each chip select can be assigned to either the standard (CKP/DAP) or alternative (CPA/DPA) phase setting when the enhanced timing is active (CEnOPE=1). The CPA/DPA bits have no effect when CEnOPE=0.

Please refer to the description *"CEnCTL1 - Queued CSI control register 1" on page 375* for details on the timings for each clock and data phase.

CEnOPE	Enhanced timing enable/disable
0	Enhanced timing is disabled
1	Enhanced timing is enabled.

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

When the enhanced timing is enabled (CEnOPE=1), the setup time, inter-data time, hold time and idle time for each chip select signal can be defined using the CEnOPTm registers. For CEnOPE=0, the standard timing applies to all chip select signals.

CEnMD	Chip select mode
0	4 chip select mode. Each of the 4 bits of the CEnCS register (CEnCS[3:0]) represents a chip select line.
1	• CSIE0: 8 chip select mode. The combination of the lower 3 bits of the CEnCS line (CEnCS[2:0]) select one out of 8 available chip select lines. • CSIE1: Setting prohibited.

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

4 chip select mode In 4 chip select mode, only 4 chip selects can be used for the communication, but multiple chip selects can be active for the same communication ("broadcast"), as each bit represents a chip select signal.

8 chip select mode In 8 chip select mode, all 8 chip selects are accessible, although only one out of the 8 chip selects can be selected as active for a communication.

(6) CEnRX0 - Receive data buffer registers

The CEnRX0 register is a 16-bit register or separated as upper 8 bits (CEnRX0H) and lower 8 bits (CEnRX0L), that is used to store receive data. It can be read in 8-bit or 16-bit units.

Initial (reset) value is

- in single transfer mode (CEnCTL0.CEnTMS = 0): 0000H
- in block transfer mode (CEnCTL0.CEnTMS = 0): undefined

Address: CE0RX0=FFFFFD42H, CE1RX0=FFFFFD82H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEnRX0																	R/W	refer to above

Caution The receive data buffer register is considered as emptied by the application software whenever the lower 8 bits of the register are read. It is therefore necessary to read CEnRX0H before CEnRX0L when 8-bit access is used.

(7) CEnCS - Chip select data buffer registers

The CEnCS register is a 16-bit register than can be also accessed as 8 bit register (CEnCSL) that stores Chip Select data. It is accessible in 8-bit or 16-bit units.

Initial value is FFFFH by reset.

Following the FIFO write pointer, the value written to CEnCS is stored in the FIFO data buffer as Chip Select bits. The value is stored to these bits when the transmit data is written to its register (CEnTX0 or CEnTX0L).

CEnCS write is prohibited when CEnPWR = 1.

Address: CE0CS=FFFFFD44H, CE1CS=FFFFFD84H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEnCS	-	-	-	-	-	-	-	-	-	-	-	-	CEnCS3	CEnCS2	CEnCS1	CEnCS0	R/W	FFFFH

Caution
1. The Chip Select register is stored in the FIFO buffer when the transmit data is written to its register. It is therefore necessary to write the CEnCS register before the CEnTX0 (CEnTX0L) register.
2. In 8 chip select mode (CEnCTL4.CEnMD = 1) CEnCS3 must be set to 0.

Depending on the chip select mode (defined by the CEnMD flag - see *"CEnCTL4 - Queued CSI control register 4" on page 380* for details) the CEnCS[3:0] bits have a different meaning:

(a) CSIE0

- **4 chip select mode (CE0MD = 0)**

CE0CS bits				Chip select signals			
CE0CS3	CE0CS2	CE0CS1	CE0CS0	SCSE03	SCSE02	SCSE01	SCSE00
0	0	0	0	selected	selected	selected	selected
0	0	0	1	selected	selected	selected	–
0	0	1	0	selected	selected	–	selected
0	0	1	1	selected	selected	–	–
0	1	0	0	selected	–	selected	selected
0	1	0	1	selected	–	selected	–
0	1	1	0	selected	–	–	selected
0	1	1	1	selected	–	–	–
1	0	0	0	–	selected	selected	selected
1	0	0	1	–	selected	selected	–
1	0	1	0	–	selected	–	selected
1	0	1	1	–	selected	–	–
1	1	0	0	–	–	selected	selected
1	1	0	1	–	–	selected	–
1	1	1	0	–	–	–	selected
1	1	1	1	–	–	–	–

Caution "When the enhanced timing mode is enabled by setting CEnCTL4.CEnOPE = 1, setting of CEnCS.CEnCS[3:0] = F_H is prohibited."

- **8 chip select mode (CE0MD = 1)**

CE0CS3	CE0CS2	CE0CS1	CE0CS0	Active chip select signal
0[a]	0	0	0	SCSE00 selected
	0	0	1	SCSE01 selected
	0	1	0	SCSE02 selected
	0	1	1	SCSE03 selected
	1	0	0	SCSE04 selected
	1	0	1	SCSE05 selected
	1	1	0	SCSE06 selected
	1	1	1	SCSE07 selected

a) In 8 chip select mode CE0CS3 must be set to 0.

(b) CSIE1: 4 chip select mode (CE1MD = 0)

CE1CS bits				Chip select signals			
CE1CS3	CE1CS2	CE1CS1	CE1CS0	SCSE13	SCSE12	SCSE11	SCSE10
0	0	0	0	selected	selected	selected	selected
0	0	0	1	selected	selected	selected	–
0	0	1	0	selected	selected	–	selected
0	0	1	1	selected	selected	–	–
0	1	0	0	selected	–	selected	selected
0	1	0	1	selected	–	selected	–
0	1	1	0	selected	–	–	selected
0	1	1	1	selected	–	–	–
1	0	0	0	–	selected	selected	selected
1	0	0	1	–	selected	selected	–
1	0	1	0	–	selected	–	selected
1	0	1	1	–	selected	–	–
1	1	0	0	–	–	selected	selected
1	1	0	1	–	–	selected	–
1	1	1	0	–	–	–	selected
1	1	1	1	–	–	–	–

Note 1. The active level for each chip select is defined in the CEnCTL2 and CEnCTL4 register (CEnCSL[7:0] bits).

2. In 4 chip select mode, multiple chip selects can be active at the same time. When enhanced timing is enabled in 4 chip select mode, the Queued-CSI macro will use the timing of the enabled chip select with the lowest number. But to ensure proper communication, it is mandatory that the timings of all chip selects that can be active at a same time are set to the same values.

Caution "When the enhanced timing mode is enabled by setting CEnCTL4.CEnOPE = 1, setting of CEnCS.CEnCS[3:0] = F_H is prohibited."

(8) CEnTX0 - Transmission data buffer registers

The CEnTX0 register is a 16-bit buffer register, or separated as upper 8 bits (CEnTX0H) and lower 8 bits (CEnTX0L), that stores transmission data. It is accessible in 8-bit or 16-bit units.

Initial value is 0000H by reset.

Incrementing and following the FIFO buffer write pointer, the value written to CEnTX0 is stored to the FIFO buffer as transmission data.

CEnTX0 write is prohibited when CEnPWR = 1.

Address: CE0TX=FFFFFD46H, CE1TX=FFFFFD86H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEnTX0																	R/W	0000H

Caution The transmit data buffer register is considered as written whenever the lower 8 bits of the register are written. It is therefore necessary to write CEnTX0H before CEnTX0L when 8-bit access is used.

(9) CEnSTR - Queued-CSI status registers

The CEnSTR register is an 8-bit register that shows the status of the Queued CSI. It is accessible in 1-bit or 8-bit units.
The bits CEnFLF, CEnEMF and CEnTSF are read only.
Initial value is 20H by reset.

CEnSTR read is prohibited when CEnPWR = 1.

Address: CE0STR=FFFFFD48H, CE1STR=FFFFFD88H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
CEnSTR	CEn PCT	CEn FLF	CEn EMF	CEn TSF	CEn SFP3	CEn SFP2	CEn SFP1	CEn SFP0	R/W	20H

CEnPCT	FIFO buffer pointer clear command
0	No operation
1	Clear all FIFO pointers to "0"

Note Read value is always “0”.

CEnFLF	FIFO buffer full status flag
0	FIFO buffer is not full
1	FIFO buffer is full

Note Read only

CEnEMF	FIFO buffer empty status flag
0	FIFO buffer is not empty
1	FIFO buffer is empty

Note Read only

CEnTSF	Transmission status flag
0	Idle state
1	Transmission in on going or preparing

Note
1. Read only.
2. This bit is cleared to “0” by CEnPWR = 0 or (CEnTXE = 0 and CEnRXE = 0).
3. In single transfer mode, this bit holds “1” from transmission start to FIFO empty.
4. In block transfer mode, this bit holds “1” from transmission start until finish transferring all data to be sent.

CEnSFP3 -0	Transmission data count
0 - 15	In Single transfer mode, CEnSFP[3:0] indicates the number of remaining transfers in the FIFO. This value can be understood as (Write FIFO pointer) - (SIO load pointer)
	In FIFO transfer mode, CEnSFP[3:0] indicates the number of data transfers completed. In case of CEnSFP[3:0] = 0H: • CEnEMF = 0 and CEnSFP[3:0] = 0H: Number of receptions completed = 0 • CEnEMF = 1 and CEnSFP[3:0] = 0H: Number of receptions completed = 16

Note 1. Read only.

2. CEnSFP[3:0] holds its value until RESET or CEnPCT = 1.

Caution CEnFLF, CEnEMF, CEnTSF and CEnSFP[3:0] are continuously updated with the current status of the Queued CSI. This means that a value read might be outdated shortly after the read was executed.
When writing accidentally a 17th data element in FIFO, an overflow interrupt (INTCEnOFF) will occur to indicate the error.

(10) CEnOPTm - Queued CSI enhanced timing registers

The CEnOPTm registers are 16-bit register, that are used to define the timing for the chip select signal CEnm when enhanced timing is active (CEnOPE = 1).

The 16-bit registers (CEnOPTm) can be read in 16-bit units.

Initial value is 0002H by reset.

Caution These register can be written only while CEnTXE = 0 and CEnRXE = 0.

Registers CEnOPT0 to CEnOPT3 are valid for both CSIE0 and CSIE1:

Address: CE0OPT0=FFFFFD50H, CE1OPT0=FFFFFD90H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT0	CEn PHS0	CEn IDL0	CEn ID02	CEn ID01	CEn ID00	CEn HD02	CEn HD01	CEn HD00	CEn DL01	CEn DL00	CEn IN02	CEn IN01	CEn IN00	CEn SP02	CEn SP01	CEn SP00	R/W	0002H

Address: CE0OPT1=FFFFFD52H, CE1OPT1=FFFFFD92H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT1	CEn PHS1	CEn IDL1	CEn ID12	CEn ID11	CEn ID10	CEn HD12	CEn HD11	CEn HD10	CEn DL11	CEn DL10	CEn IN12	CEn IN11	CEn IN10	CEn SP12	CEn SP11	CEn SP10	R/W	0002H

Address: CE0OPT2=FFFFFD54H, CE1OPT2=FFFFFD94H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT2	CEn PHS2	CEn IDL2	CEn ID22	CEn ID21	CEn ID20	CEn HD22	CEn HD21	CEn HD20	CEn DL21	CEn DL20	CEn IN22	CEn IN21	CEn IN20	CEn SP22	CEn SP21	CEn SP20	R/W	0002H

Address: CE0OPT3=FFFFFD56H, CE1OPT3=FFFFFD96H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT3	CEn PHS3	CEn IDL3	CEn ID32	CEn ID31	CEn ID30	CEn HD32	CEn HD31	CEn HD30	CEn DL31	CEn DL30	CEn IN32	CEn IN31	CEn IN30	CEn SP32	CEn SP31	CEn SP30	R/W	0002H

Registers CE0OPT4 to CE0OPT7 are only valid for CSIE0:

Address: CE0OPT4=FFFFFD58H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT4	CEn PHS4	CEn IDL4	CEn ID42	CEn ID41	CEn ID40	CEn HD42	CEn HD41	CEn HD40	CEn DL41	CEn DL40	CEn IN42	CEn IN41	CEn IN40	CEn SP42	CEn SP41	CEn SP40	R/W	0002H

Address: CE0OPT5=FFFFFD5AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT5	CEn PHS5	CEn IDL5	CEn ID52	CEn ID51	CEn ID50	CEn HD52	CEn HD51	CEn HD50	CEn DL51	CEn DL50	CEn IN52	CEn IN51	CEn IN50	CEn SP52	CEn SP51	CEn SP50	R/W	0002H

Address: CE0OPT6=FFFFFD5CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT6	CEn PHS6	CEn IDL6	CEn ID62	CEn ID61	CEn ID60	CEn HD62	CEn HD61	CEn HD60	CEn DL61	CEn DL60	CEn IN62	CEn IN61	CEn IN60	CEn SP62	CEn SP61	CEn SP60	R/W	0002H

Address: CE0OPT7=FFFFFD5EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
CEn OPT7	CEn PHS7	CEn IDL7	CEn ID72	CEn ID71	CEn ID70	CEn HD72	CEn HD71	CEn HD70	CEn DL71	CEn DL70	CEn IN72	CEn IN71	CEn IN70	CEn SP72	CEn SP71	CEn SP70	R/W	0002H

The enhanced timing allows insertion of delays in the chip select timing:

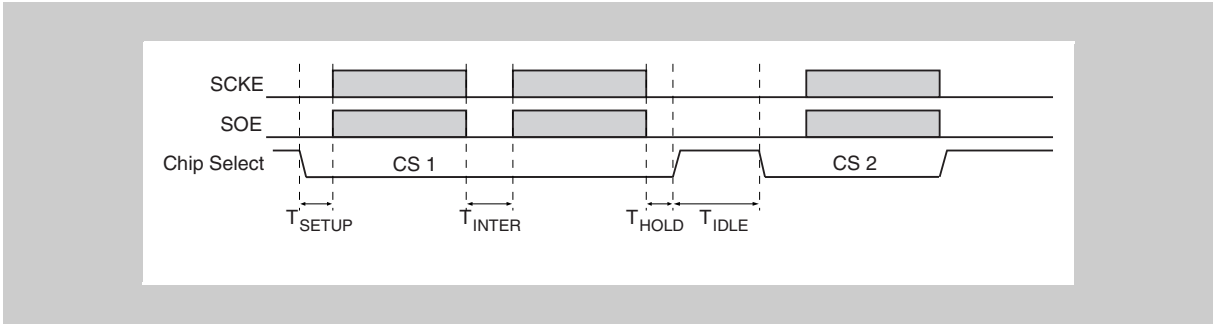


Figure 13-3 General overview on enhanced chip select timing

CEnPHSm	Clock/data phase selection
0	Clock/data phase selection defined by CKP/DAP are used for chip select m
1	Alternative clock/data phase selection defined by CKA/DAP are used for chip select m

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

CEnIDLm	Chip select idle enforcement
0	Chip select m becomes idle whenever a different chip select value is defined the CEnCS[3:0] bits. If the chip select value remains the same or the buffer becomes empty, the chip select stays active.
1	An idle state is always inserted after each transfer.

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

When the multiple data is transferred using the same chip select setting, the chip select line remains active. The CEnIDLm bit forced the chip select line to become idle after each transfer, even if the same chip select setting is used for the next transfer.

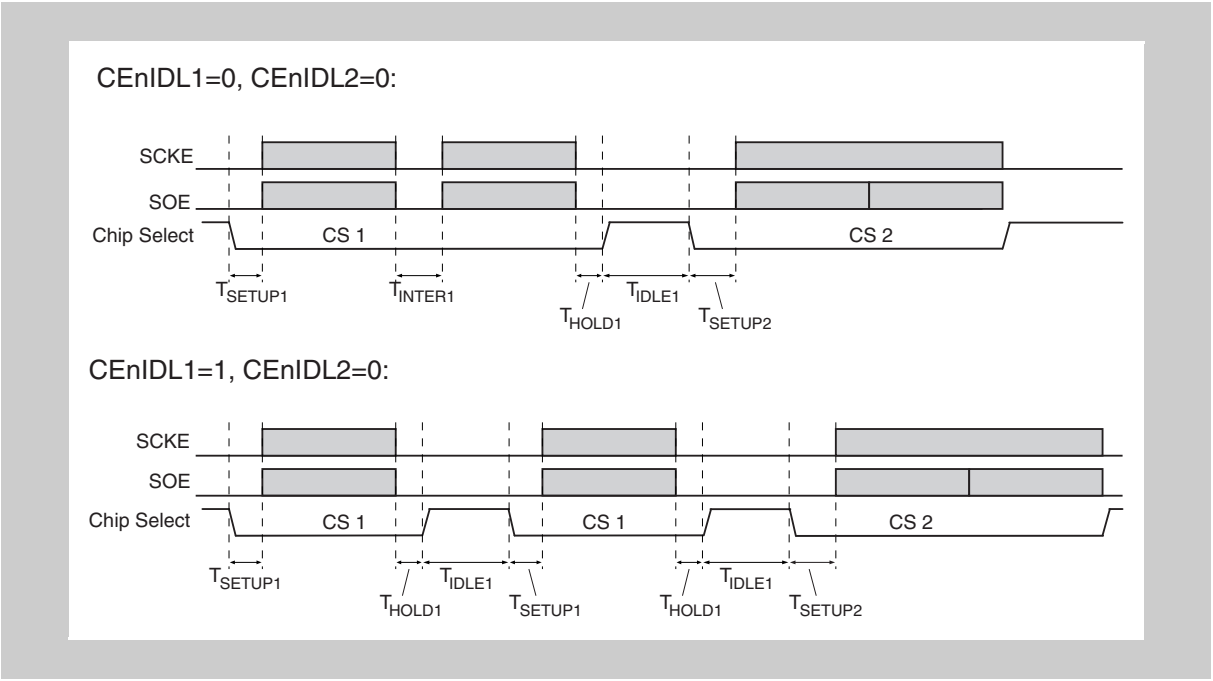


Figure 13-4 Enforced chip select idle communication

CEnIDm2	CEnIDm1	CEnIDm0	Chip select m idle timing [Note]
0	0	0	chip select m is idle for 0 SCKE
0	0	1	chip select m is idle for 0.5 SCKE
0	1	0	chip select m is idle for 1 SCKE
0	1	1	chip select m is idle for 2 SCKE
1	0	0	chip select m is idle for 3 SCKE
1	0	1	chip select m is idle for 4 SCKE
1	1	0	chip select m is idle for 6 SCKE
1	1	1	chip select m is idle for 8 SCKE

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

Note A minimum idle period of around 2-3 macro clock cycles (f_{QCSI}) is applied whenever a CS is set from active to inactive. This time is an addition to the time set by the CEnIDm[2:0] bits.

CEnHDm2	CEnHDm1	CEnHDm0	Chip select m hold timing	
			CEnSIT=0	CEnSIT=1
0	0	0	No hold time for chip select m	hold chip select m for 0.5 SCKE
0	0	1	hold chip select m for 0.5 SCKE	hold chip select m for 1 SCKE
0	1	0	hold chip select m for 1 SCKE	hold chip select m for 1.5 SCKE
0	1	1	hold chip select m for 2 SCKE	hold chip select m for 2.5 SCKE
1	0	0	hold chip select m for 3 SCKE	hold chip select m for 3.5 SCKE
1	0	1	hold chip select m for 4 SCKE	hold chip select m for 4.5 SCKE
1	1	0	hold chip select m for 6 SCKE	hold chip select m for 6.5 SCKE
1	1	1	hold chip select m for 8 SCKE	hold chip select m for 8.5 SCKE

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

Note The hold times given above assume that a new transfer element is waiting in the transfer buffer. When the transfer queue becomes empty, the hold time will be longer than the times given above.

CEnDLm1	CEnDLm0	Chip select m data length selection
0	0	The data length defined by CEnDL[3:0] is used as data length chip select m
0	1	The data length defined by CEnDL[3:0] is used as data length chip select m
1	0	The data length for chip select m is 8 bits (regardless of CEnDL[3:0] setting)
1	1	The data length for chip select m is 16 bits (regardless of CEnDL[3:0] setting)

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

CEnINm2	CEnINm1	CEnINm0	Chip select m inter-data delay timing	
			CEnSIT=0	CEnSIT=1
0	0	0	No inter-data delay	Inter-data delay is 0.5 SCKE
0	0	1	Inter-data delay is 0.5 SCKE	Inter-data delay is 0.5 SCKE
0	1	0	Inter-data delay is 1 SCKE	Inter-data delay is 1 SCKE
0	1	1	Inter-data delay is 2 SCKE	Inter-data delay is 2 SCKE
1	0	0	Inter-data delay is 3 SCKE	Inter-data delay is 3 SCKE
1	0	1	Inter-data delay is 4 SCKE	Inter-data delay is 4 SCKE
1	1	0	Inter-data delay is 6 SCKE	Inter-data delay is 6 SCKE
1	1	1	Inter-data delay is 8 SCKE	Inter-data delay is 8 SCKE

Caution Write is permitted only when CEnTXE = 0 and CEnRXE = 0.

Note 1. Inter-data delay only applies when no idle state is enforced between data (CEnIDLm=0).

2. The inter-data times given above assume that a new transfer element is waiting in the transfer buffer. When the transfer queue becomes empty, the hold time will be longer than the times given above.

CEnSPm2	CEnSPm1	CEnSPm0	Chip select m setup timing
0	0	0	No setup delay
0	0	1	Setup delay is 0.5 SCKE
0	1	0	Setup delay is 1 SCKE
0	1	1	Setup delay is 2 SCKE
1	0	0	Setup delay is 3 SCKE
1	0	1	Setup delay is 4 SCKE
1	1	0	Setup delay is 6 SCKE
1	1	1	Setup delay is 8 SCKE

13.3 Explanation of Queued CSI Functions

13.3.1 Transmit buffer

Chip select data and transmission data can be stored to the transmit FIFO buffer continuously by writing to the CEnCS register and CEnTX0 register. The Writing FIFO pointer is automatically incremented when data is written to CEnTX0. The size of the transmit FIFO buffer is 20 bits × 16 entries.

In slave mode, it is not necessary to set the chip select data.

The transfer start condition (assuming that CEnEMF = 0) is to write to the lower bits of CEnTX0 register. If the transmission data length is 9 bits or more, data should be written by a 16-bit write to CEnTX0 or by two 8-bit writes with first CEnTX0H, then CEnTX0L (in that order). When transmission data length is 8 bits, data should be set by one 8-bit write to CEnTX0L or by one 16-bit write to CEnTX0. For the 16-bit write the upper 8 bits are ignored in the 8-bit transmission.

The CEnFLF bit in the CEnSTR status register is set “1” after 16 writes have been made to the transmit buffer, assuming the Writing FIFO pointer was reset previously.

When a transmission write is attempted while the FIFO is full (CEnFLF=1), the interrupt INTCEnOFF is generated to indicate an overflow. In that case, the transmission and chip select data are discarded and not stored.

When a transfer cycle is finished and the SIO Loading FIFO pointer is incremented, the FIFO buffer of the previous location is considered empty in case of single transfer mode. Refer to *“Description of the block transfer mode” on page 401* for more information on block transfer mode.

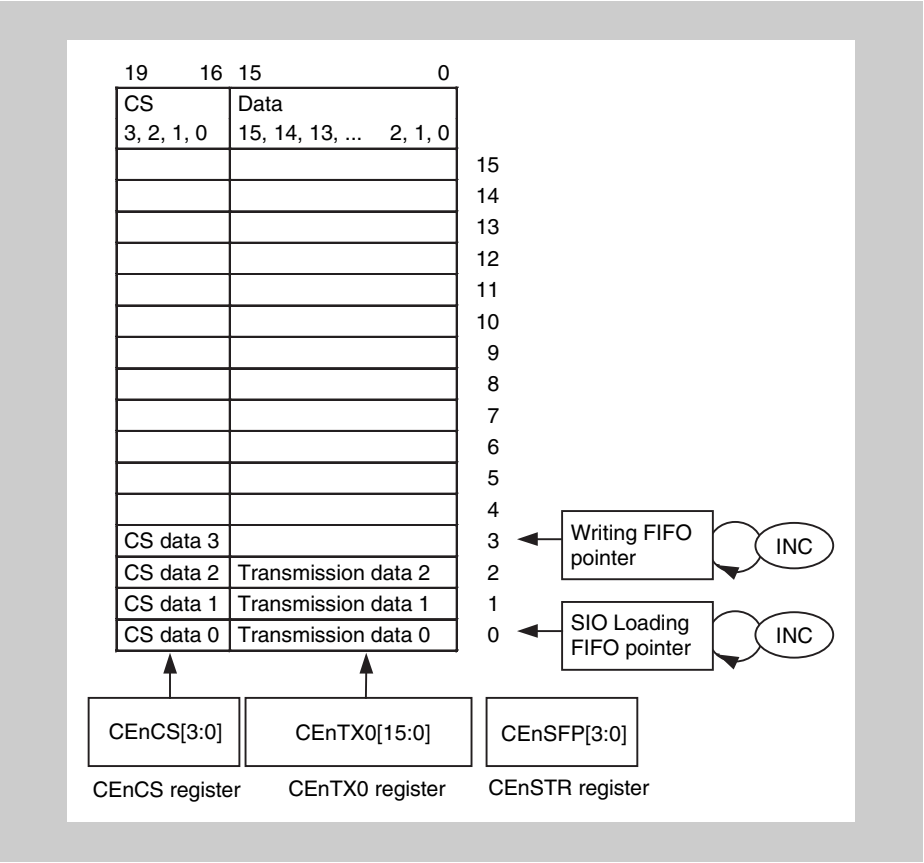


Figure 13-5 Transmit buffer

13.3.2 Serial data direction select function

The serial data direction is selectable using the CEnDIR bit in the CEnCTL0 register. The examples below show the communication for data length of 8 bit (CEnDL[3:0] = [1,0,0,0]):

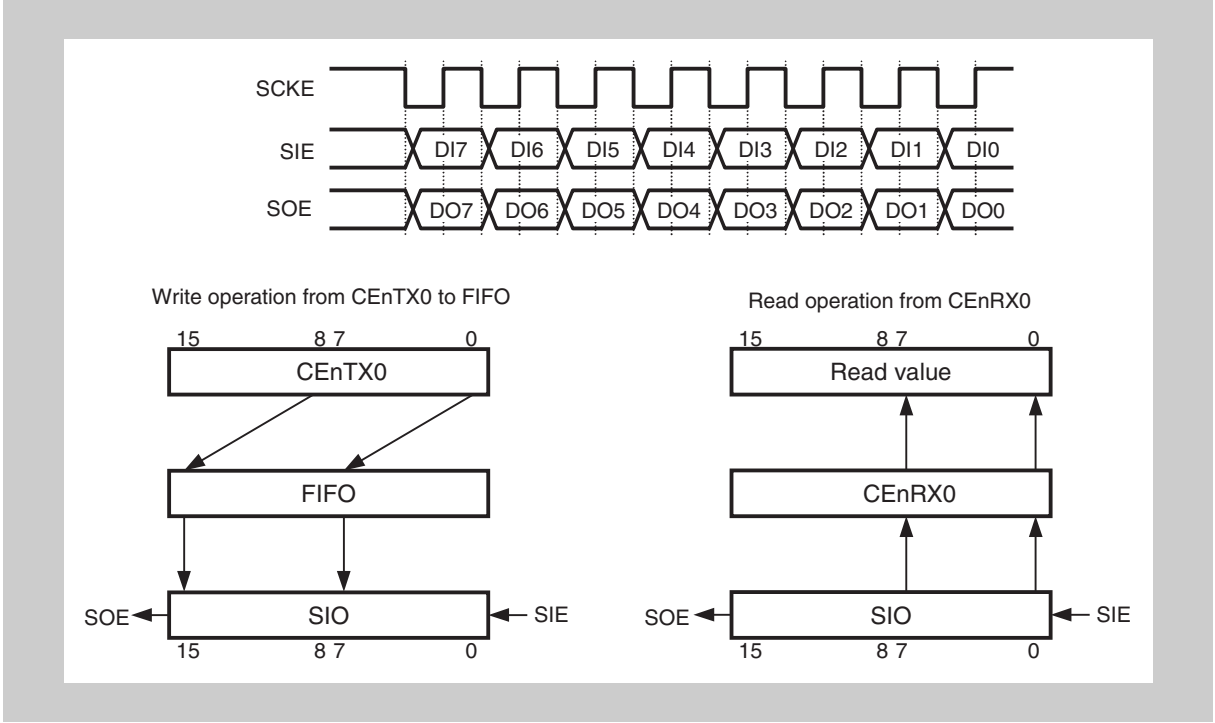


Figure 13-6 Serial data direction select function - msb first (CEnDIR = 0)

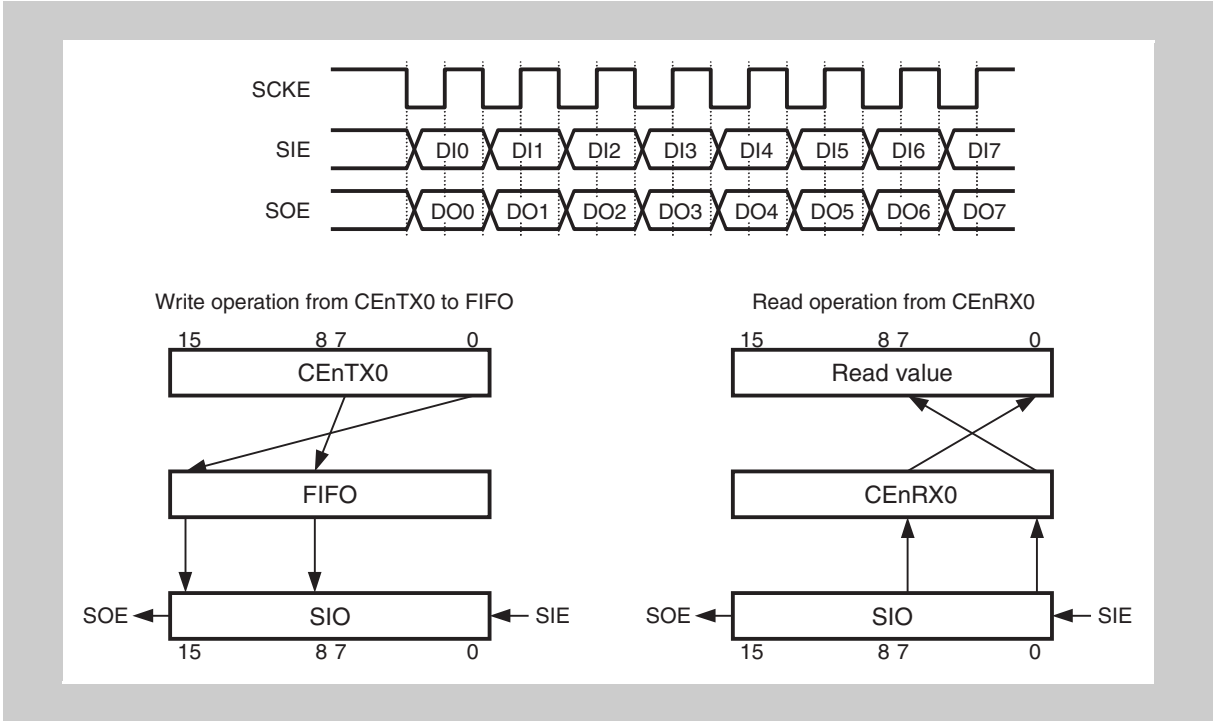


Figure 13-7 Serial data direction select function - LSB first (CEnDIR = 0)

13.3.3 Data length select function

Transmission data length is selectable from 8 bits to 16 bits using the CEnDL[3:0] bits in CEnCTL2 register. The examples below show the communication with MSB first (CEnDIR = 1):

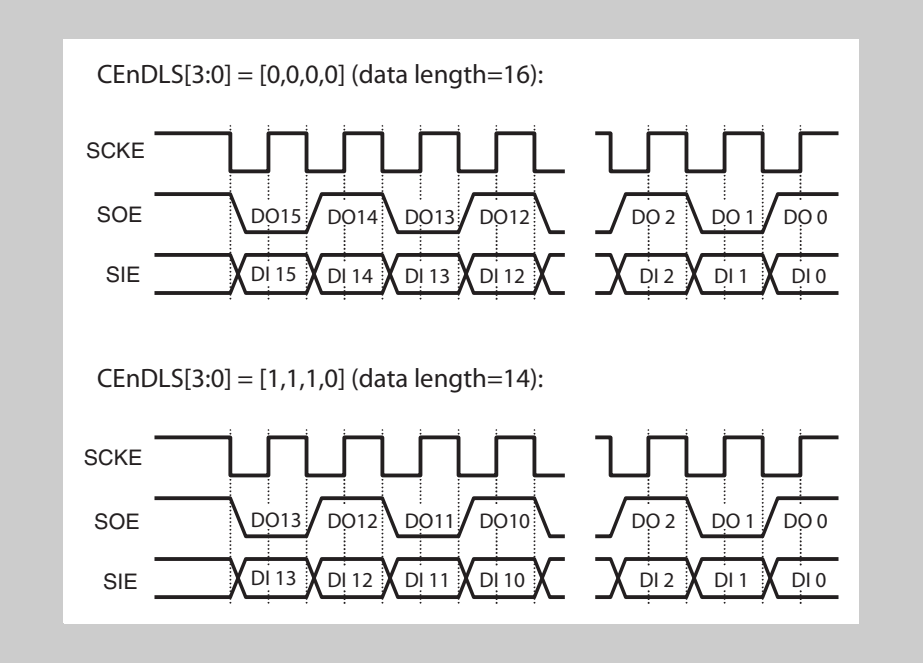


Figure 13-8 Data length select function

13.3.4 Slave mode

When the CEnCKS[2:0] bits in CEnCTL1 are set to [1,1,1], the Queued CSI operates in slave mode. In slave mode, the SCKE serial clock pin becomes input and another device is the CSI communication master. The baud rate generator “BRG” is recommended to be disabled by setting bits CEnMDL[2:0] to [0,0,0] when using slave mode. Also, the chip select pin SCSEn[3:0] outputs are not available in slave mode, as they are only available in master mode.

The example below shows the communication in slave mode for 8 data bits, CEnCKP=0, CEnDAP=0 and MSB first:

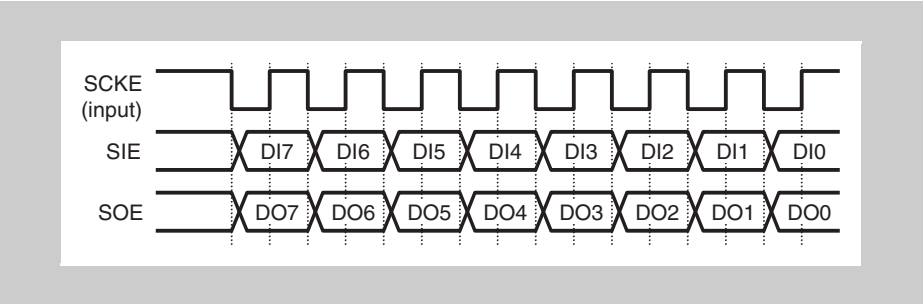


Figure 13-9 Slave mode

13.3.5 Master mode

When the CEnCKS[2:0] bits in CEnCTL1 are not set to [1,1,1], the Queued CSI operates in master mode. In master mode, the SCKE pin is configured as output and the serial communication clock is generated by the Queued CSI module. The SCKE pin's default value is "1" when CEnCKP = 1, and is default "0" when CEnCKP = 0. The SCSEn[3:0] pin outputs are available in master mode.

The example below shows the communication in master mode for 8 data bits, CEnCKP=0, CEnDAP=0 and MSB first:

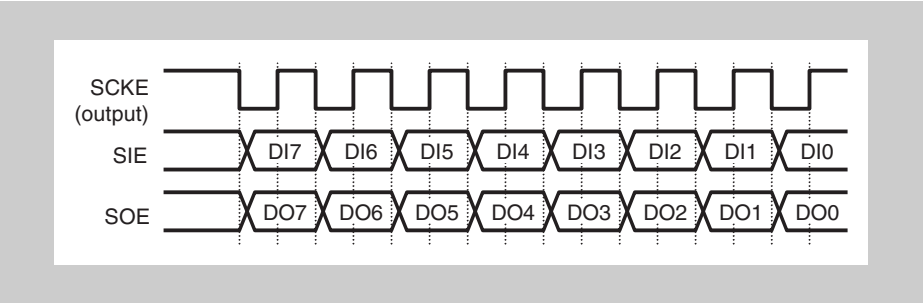


Figure 13-10 Master mode

13.3.6 Transmission clock select function

In Master Mode, the transfer baud rate is selectable using CEnCKS[2:0] bits and CEnMDL[2:0] bits in CEnCTL1 register. The baud rate generator “BRG” counts up at each rising edge of f_{QCSI}.

The example below illustrates the baud rate generation for CEnMDL[2:0] = [0,1,0].

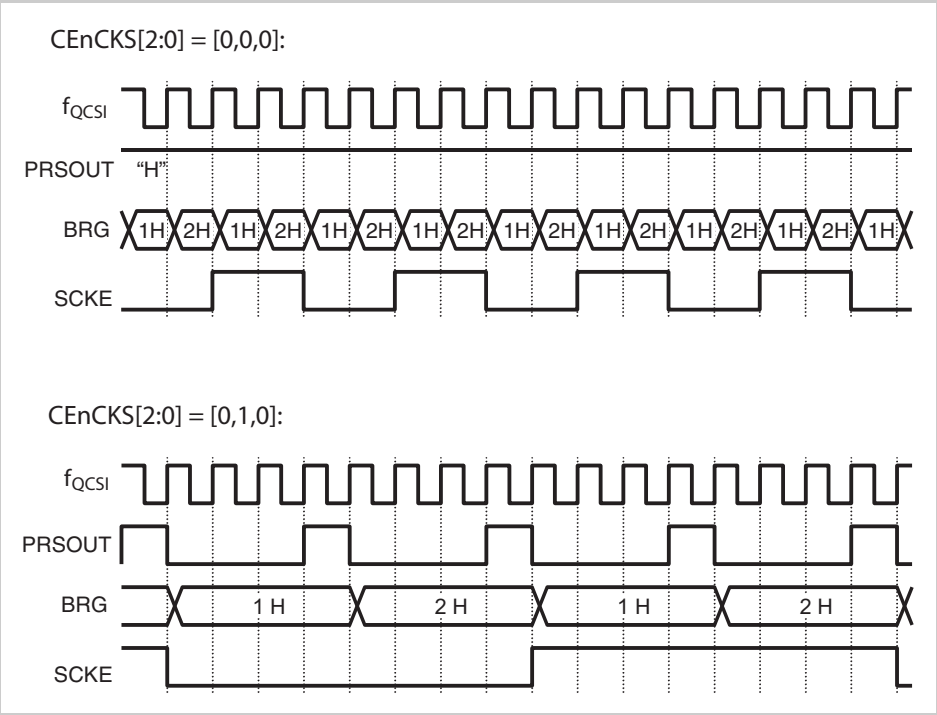


Figure 13-11 Transfer clock select function

13.3.7 Description of the single transfer mode

When the CEnTMS bit in the CEnCTL0 register is set “0”, the Queued CSI operates in single transfer mode.

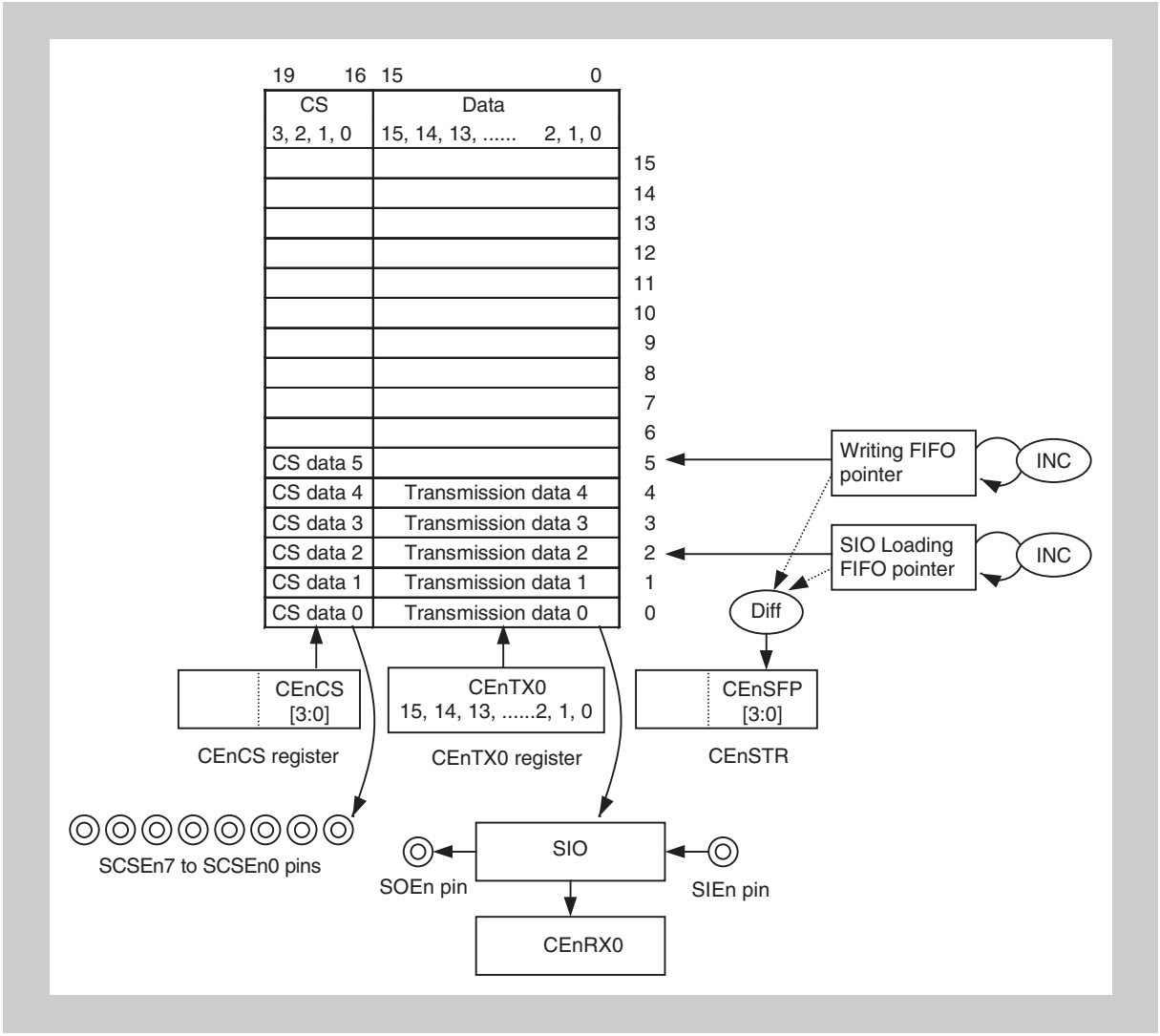


Figure 13-12 Single transfer mode data handling

Transfer start condition in single transfer mode:

[CEnTXE = 1 or CEnRXE = 1] and
[Data exists in FIFO (CEnEMF = 0)]

A transfer starts once the transmission data (pointed to by the SIO Loading FIFO pointer) is transferred from the FIFO buffer to the serial shift register SIO. At that time, the transfer status flag CEnTSF turns to “1”. The SCSEn[3:0] pins output the chip select data from the FIFO buffer.

At the end of the transfer:

1. If CEnRX0 is empty, the received data is stored from SIO to CEnRX0, and transfer end interrupt signal INTCEnC is generated (in transmit only mode, INTCEnC will be generated only when the FIFO buffer becomes empty). Finally, the SIO Loading FIFO pointer is incremented.
2. If CEnRX0 is not empty, the storing of receive data, INTCEnC generation and SIO Loading FIFO pointer incrementing wait for the CEnRX0 to be emptied by a software read operation.

3. In transmit only mode, if transmission data is available in the FIFO buffer, the next transfer will start immediately, regardless of the CEnRX0 buffer condition.

When a transfer finishes and the FIFO buffer is empty (Writing FIFO pointer = SIO Loading FIFO pointer), CEnTSF is cleared “0”.

CEnSFP[3:0] always show the current value of: (Writing FIFO pointer) - (SIO Loading FIFO pointer).

It is recommended to check that CEnFLF = 0 just before data is written to the CEnTX0 register. If CEnFLF = 1 when a write to CEnTX0 is attempted, the overflow interrupt INTCEnOFF is generated and the written data is ignored.

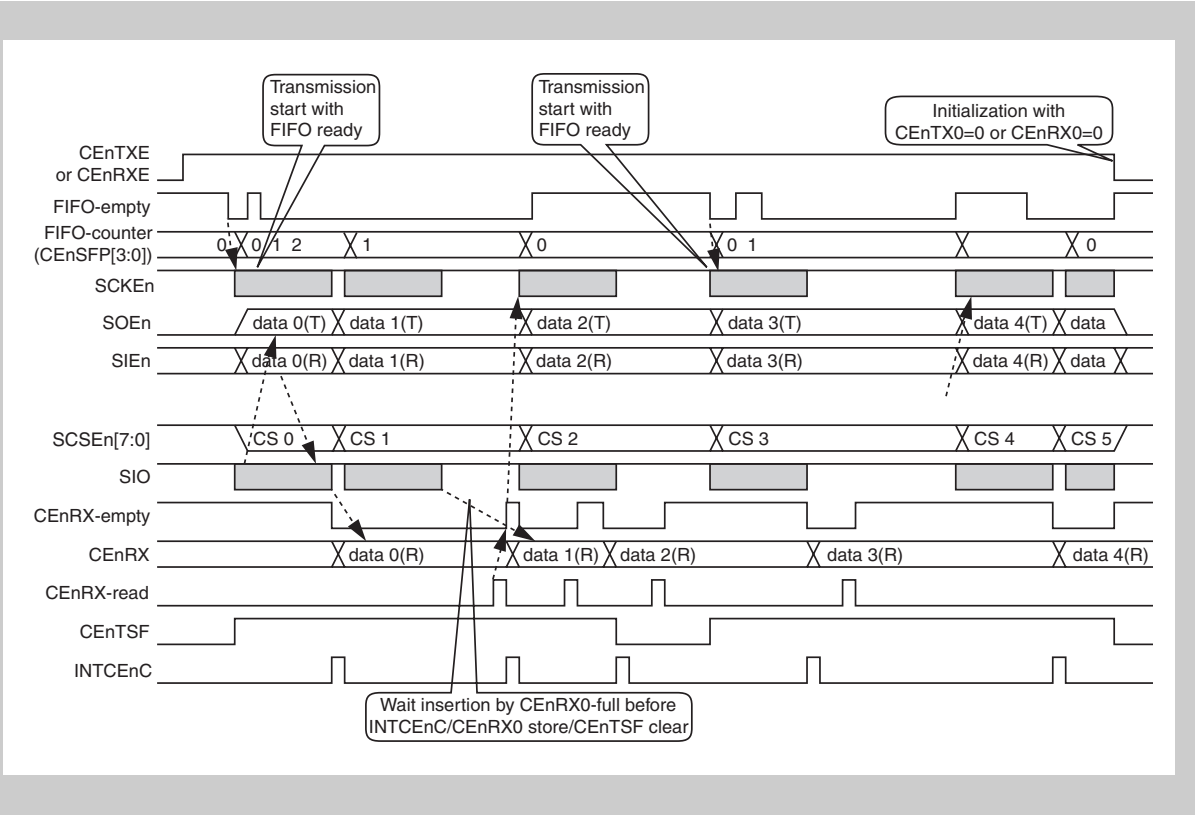


Figure 13-13 Single transfer mode (master, transmit/receive) timing

13.3.8 Description of the block transfer mode

When the CEnTMS bit in the CEnCTL0 register is set “1”, the Queued CSI operates in block transfer mode.

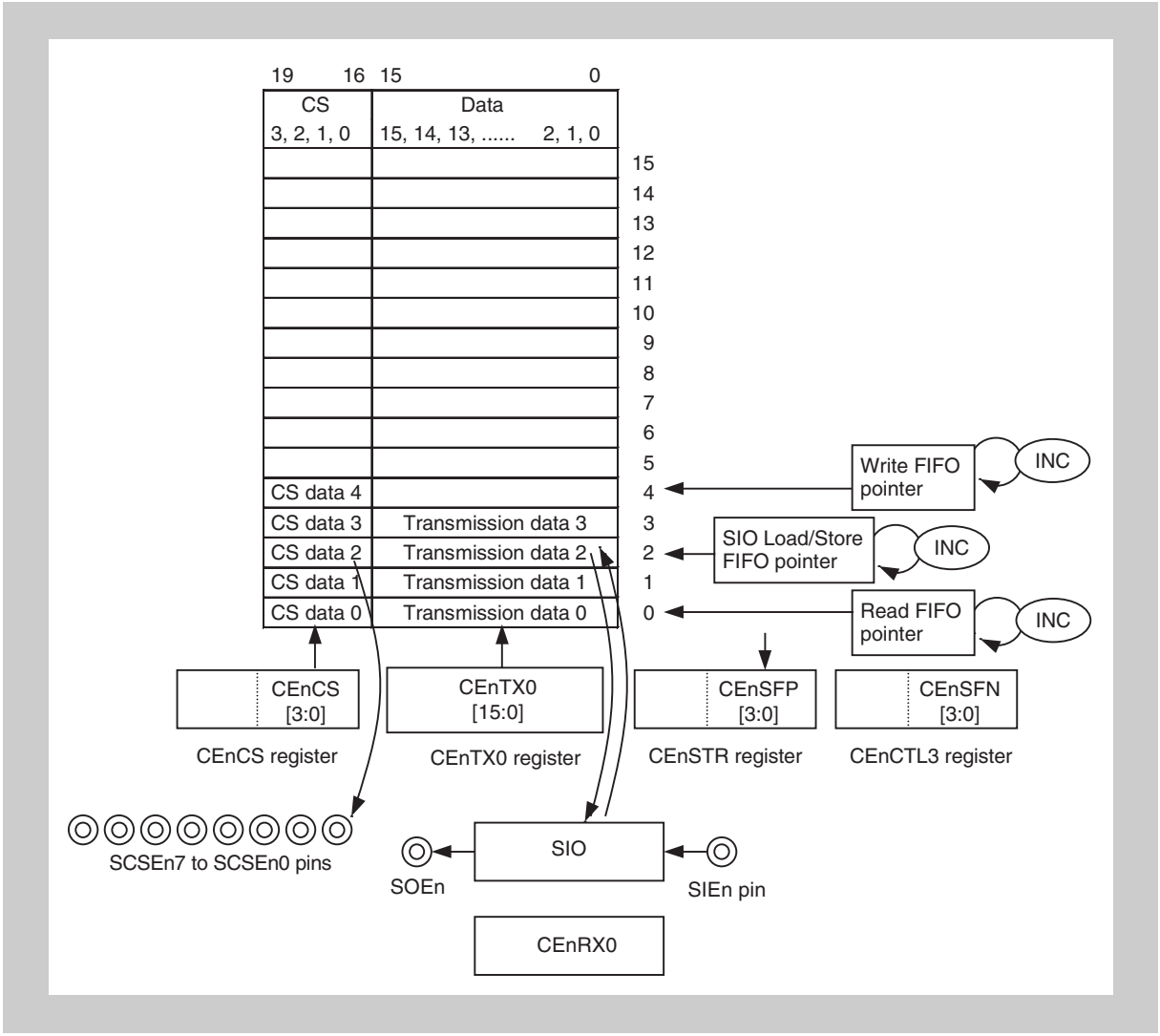


Figure 13-14 Block transfer mode data handling

Transfer start condition in block transfer mode:

[CEnTXE = 1 or CEnRXE = 1] and
[Data exists in FIFO (CEnEMF = 0)]

The transmission data number must be set in CEnSFN[3:0]. Note that writing a value greater than 16 to the CEnCTL3 register is prohibited, as the FIFO buffer design can hold up to 16 elements only.

The transfer starts by copying the first data element - pointed to by SIO Load/Store FIFO pointer - to the SIO shift register. At that time the transmission status flag CEnTSF is set to “1”, and the SCSEn[3:0] pins output the CS value from the FIFO.

When the transfer of the data element is finished, the received data overwrites the location in the FIFO using the SIO Load/Store FIFO pointer, and the SIO Load/Store FIFO pointer is then incremented.

When the transmission/reception counter reaches the value set by CEnSFN[3:0], then CEnTSF is cleared “0” and the transmission/reception end interrupt signal INTCEnC is generated.

After the interrupt occurred, the received data can be read from CEnRX0. The Read FIFO pointer is automatically incremented by the CEnRX0 read operation.

All FIFO pointers must be cleared by setting CEnPCT = 1 before the next transmit/receive cycle can start.

SFP[3:0] represents the [SIO Load/Store FIFO pointer] and shows the number of transmission/receptions completed. In case of SFP[3:0]=0H, the numbers of transmissions/receptions depends on the setting of the SFEMP bit:

SFEMP=0: 0 transmissions/receptions completed

SFEMP=1: 16 transmissions/receptions completed

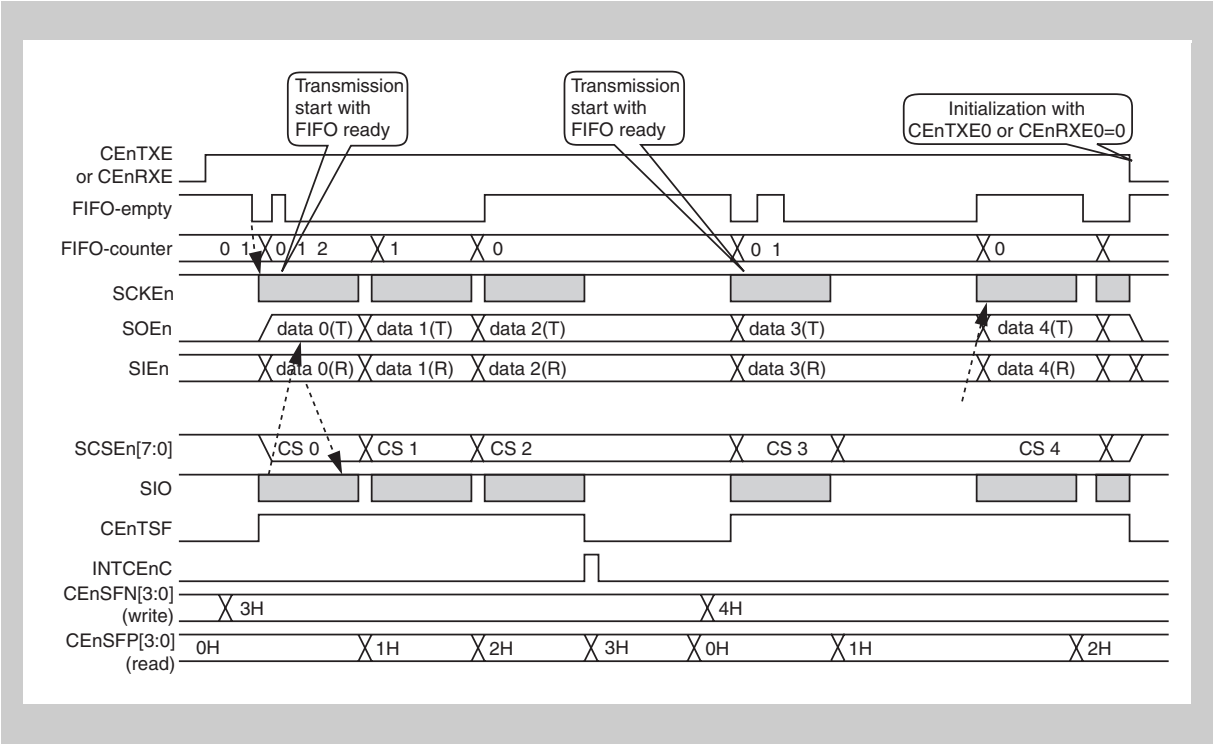


Figure 13-15 Block transfer mode (master. transmit/receive) timing

13.3.9 Description of the operation modes

(1) Transmit only mode

Setting the CEnCTL0 register's CEnTXE =1 and CEnRXE = 0 places the Queued CSI in transmit only mode. A transmission starts when transmit data is written in the CEnTX0 register. The current condition of the CEnRX0 buffer and SIO register no effect. The data in the CEnRX0 and the SIO buffer is undefined after completion of the transmission.

(2) Receive only mode

Setting the CEnCTL0 register's CEnTXE = 0 and CEnRXE = 1 places the Queued CSI in receive only mode. A reception starts when dummy data is written in the CEnTX0 register. It is mandatory, though, that the CEnRX0 and SIO are empty. If a receive operation is terminated while the previous receive data remains unread in CEnRX0, the Queued CSI is placed on wait status until the previous data is completely read and CEnRX0 becomes empty.

(3) Transmit/receive mode

Setting the CEnCTL0 register's CEnTXE = 1 and CEnRXE = 1 places the Queued CSI in transmit/receive mode. A transfer (meaning transmission and reception) starts when transmit data is written in the CEnTX0 register. Note that an empty CEnRX0 or SIO is mandatory. If a receive operation is terminated while the previous receive data remains unread in CEnRX0, the Queued CSI is placed on wait status until the previous data is completely read and CEnRX0 becomes empty.

In block transfer mode with slave mode, only the first dummy data write operation is required. There is no need to write chip-select data, as these bits are ignored.

13.3.10 Additional timing and delay selections

(1) Delay selection of receive termination interrupt signal (INTCEnC)

In master mode, the CEnSIT bit of the CEnCTL0 register can be used to delay the generation of the receive termination interrupt signal (INTCEnC) by a half serial clock cycle (SCKE). The CEnSIT bit takes effect only in the master mode and is ignored in slave mode.

Figure 13-16 below illustrates the CEnSIT function, assuming a setting of CEnSIT=1, CEnWE=0, CEnCKP=0, CEnDAP=0 and CEnDL[3:0] = [1,0,0,0].

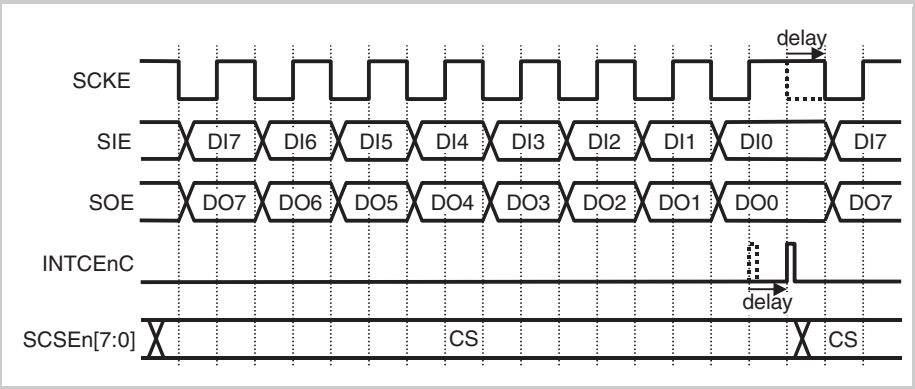


Figure 13-16 Delay selection of receive termination interrupt (INTCEnC)

(2) Selection of transmit wait enable/disable

In master mode, while the enhanced timing is not enabled, the CEnCTL0 register's CEnWE bit setting can be used to delay the start of transmission by one SCKE clock cycle. The CEnWE bit takes effect only in the master mode and is ignored in slave mode.

Figure 13-17 below illustrates the CEnWE function, assuming a setting of CEnWE=1, CEnCSM=0, CEnOPE=0, CEnCKP=0, CEnDAP=0 and CEnDL[3:0] = [1,0,0,0].

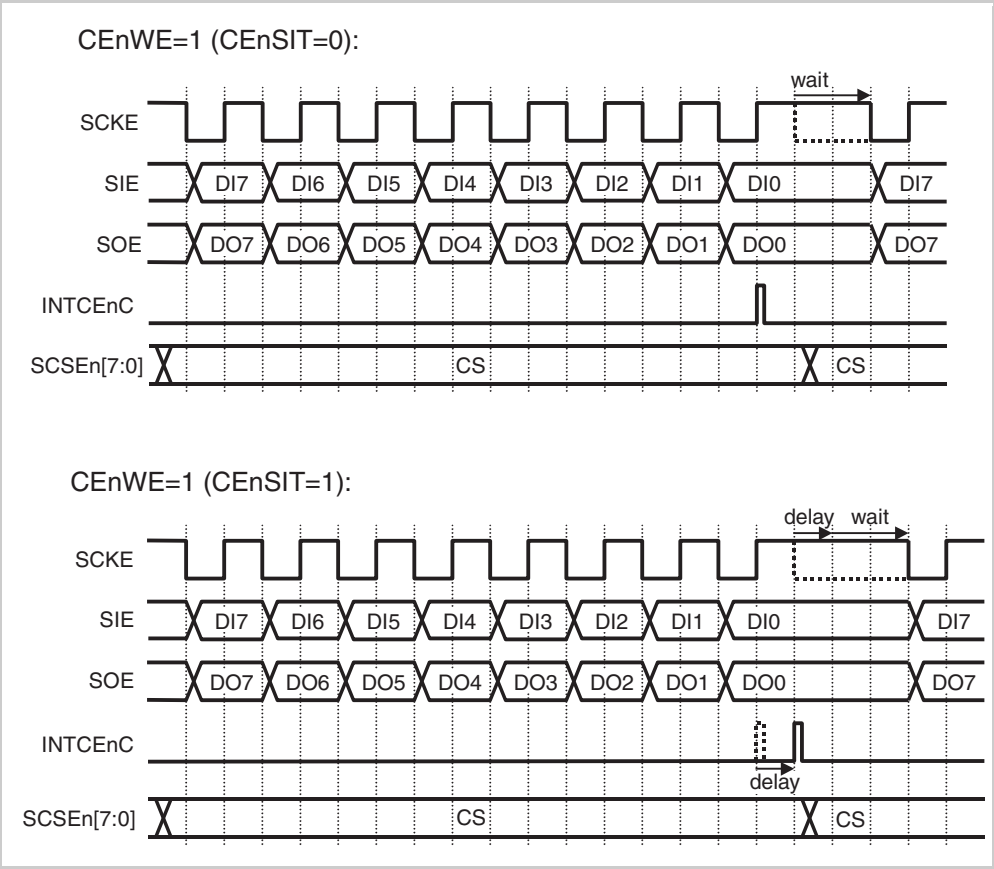


Figure 13-17 Selection of transmit wait enable/disable

(3) Selection of chip-select mode

In master mode with CEnWE = 1 and CEnOPE=0, the CEnCSM bit setting can be used to output an inactive level at the chip select pins SCSEn[3:0] during the delay between data transmissions. The CEnCSM bit takes effect only in the master mode and is ignored in slave mode.

The CEnCSM bit has no effect while CEnWE is set to 0.

Figure 13-18 below illustrates the CEnCSM function, assuming a setting of CEnCKP=0, CEnDAP=0 and CEnDL[3:0] = [1,0,0,0].

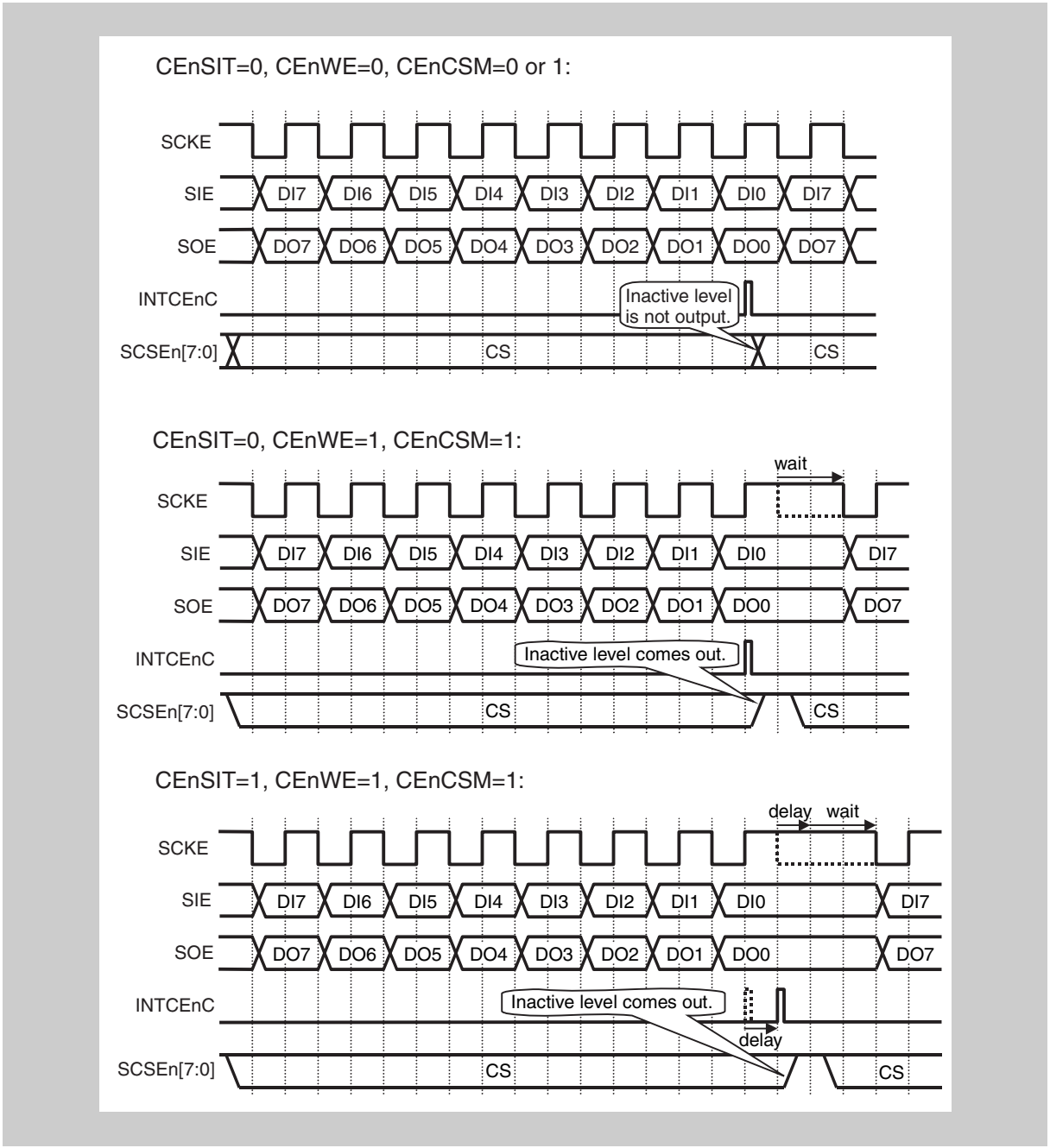


Figure 13-18 Selection of chip-select mode

13.3.11 Default pin levels

(1) SCKE pin's default level

The SCKE pin's default level with the CEnCTL0 register settings CEnPWR=0 or (CEnTXE=0 and CEnRXE = 0) is as follows:

CEnCKP	CEnCKS[2:0]	SCKE default level	
0	1, 1, 1 (slave mode)	1	← Initialization after reset
	Other than 1, 1, 1 (master mode)	1	
1	1, 1, 1 (slave mode)	1	
	Other than 1, 1, 1 (master mode)	0	

Note In slave mode, the SCKE pin is always set to “1”.

(2) SOE pin's default level

The SOE pin's default level with the CEnCTL0 register settings CEnPWR=0 or (CEnTXE=0 and CEnRXE=0) is as follows:

SOE pin's default level	
0	←Initialization after reset

(3) SCSEn0 to SCSEn7 pins' default level

The SCSEn0 to SCSEn7 pins' default level with the CEnCTL0 register settings CEnPWR=0 or (CEnTXE=0 and CEnRXE=0) is as follows:

SCSEn0 to SCSEn7 pins' default level	
inactive	←Initialization after reset

13.3.12 Transmit buffer overflow interrupt signal (INTCEnOF)

When the transmit FIFO buffer contains 16 elements, writing a 17th chip-select data (CEnCS write) or transfer data (CEnTX0 write) results in the generation of the overflow interrupt INTCEnOF. For the 17th item, both chip-select and transfer data values are discarded.

The transmit FIFO buffer contains 16 elements if the FIFO pointer value for the write operation equals the FIFO pointer value for the SIO load operation plus 15. When the transfer is completed and the FIFO buffer pointer for the SIO load operation is incremented, space for one element is available again in transmit buffer.

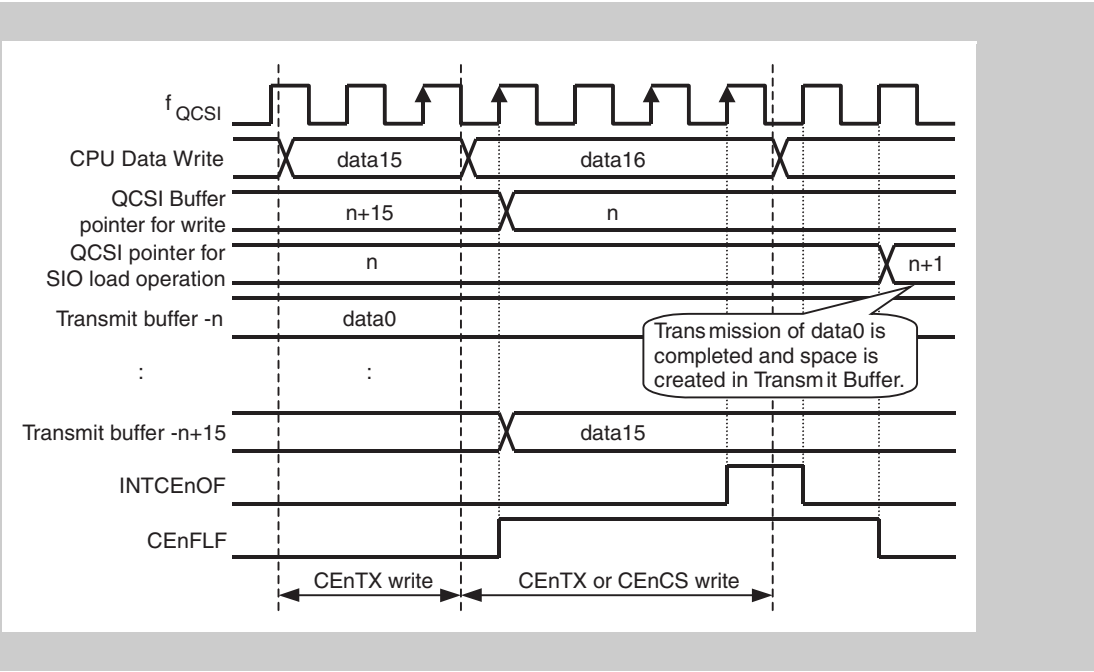


Figure 13-19 Transmit buffer overflow interrupt signal (INTCEnOF)

13.3.13 Enhanced chip select timing

(1) General

The Enhanced Queued-CSI CSIE allows specific timing settings for each chip select line. The figure below shows the general timings of the chip select lines:

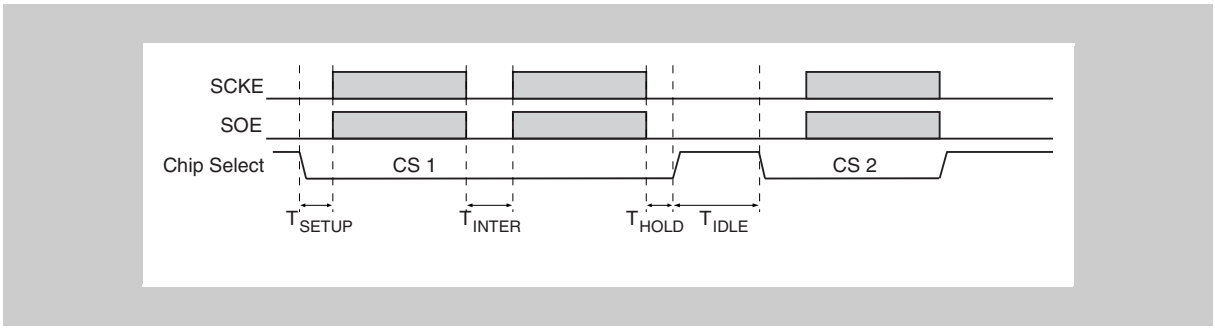


Figure 13-20 General Overview on Enhanced Chip Select Timing

T_{SETUP} is the time between the chip select line becoming active and the first data bit to be transferred. T_{INTER} defines the delay between two transfers while the same chip select is chosen and active. After the last transfer, the chip select line remains active for a period set by T_{HOLD}. Last not least, T_{IDLE} can be used to delay the start of a new transfer with a different chip select setting.

Each delay time is set relative to the serial communication clock SCKE and can range from 0 (no delay) up to 8 serial clocks.

(2) Chip select deactivation

A chip select line is selected by writing the corresponding value to the CEnCS[3:0] bits. The chip select line remains active until a value is written to the CEnCS[3:0] bits that is different from the previous value.

The chip select line remains active even when the transfer queue becomes empty. After a new element is written in the transfer queue, either the inter-data or the hold timing will be applied before the new element is transmitted. In other words, the queue idle time will be seen in addition to the defined chip select timing.

(3) Multiple chip select activation

In 4 chip select mode, multiple chip selects can be active at the same time. When enhanced timing is enabled, the Queued-CSI macro will use the timing of the enabled chip select with the lowest number. But to ensure proper communication, it is mandatory that the timings of all chip selects that can be active at a same time are set to the same values.

(4) Enforced chip select idle setting

The chip select line is hold active until the chip select setting of a transfer element changes. While this is suitable for most peripheral devices connected to the serial interface, some devices require an inactive state of the chip select line after each element transmitted.

The “Force Chip Select Idle” (CEnIDLm) bit is used to force this behaviour. *Figure 13-21* illustrates the difference between a standard communication and a communication with the CEnIDL1 bit set. It is assumed that the delay timings of CS2 are set as T_{INTER2}=0 and T_{HOLD2}=0, and that two elements are transferred to CS1 one before a transfer of two elements to CS2.

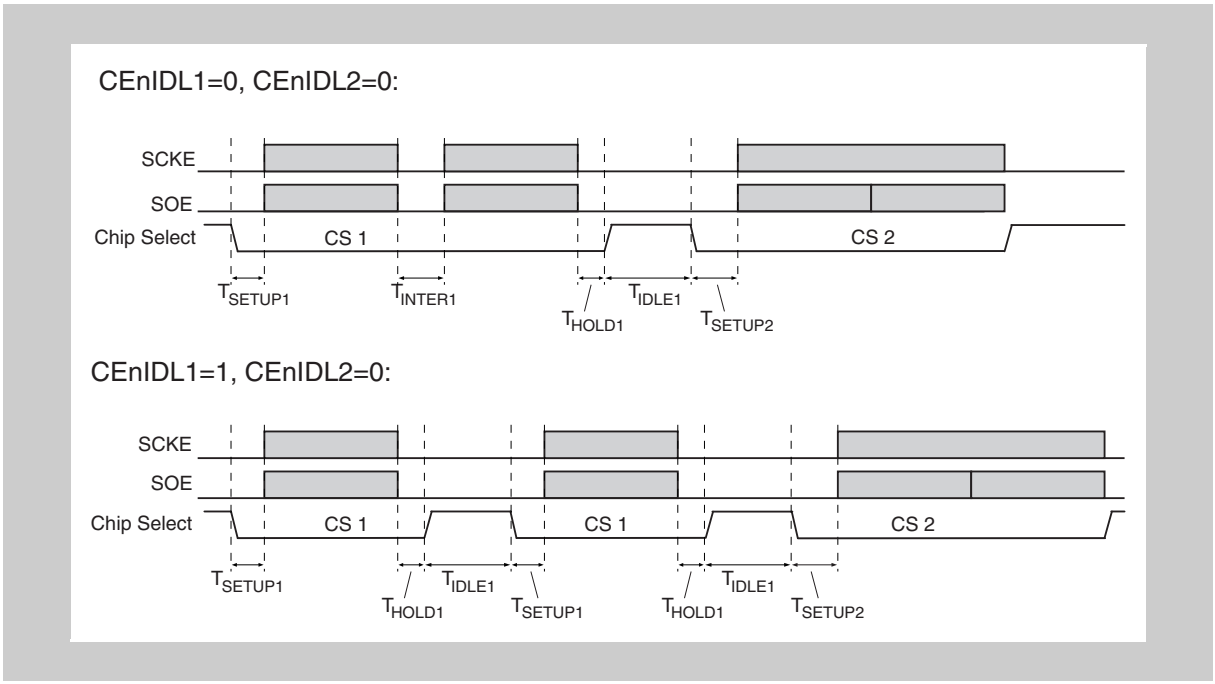


Figure 13-21 Enforced chip select idle example

While the timing for CEnIDL1=0 uses T_{INTER1} as between the two elements for CS1, the setting of CEnILD1=1 forces the T_{HOLD1}, T_{IDLE1} and T_{SETUP1} delays for the chip select line between the elements.

(5) Hold, idle and setup timing relative to $\overline{\text{SCKE}}$

The start of any timing is at the last valid data signal of the SCKE, not at the end of the SCKE cycle. *Figure 13-22* illustrates the timing. It is assumed that CKP=0, DAP=1, T_{HOLD1}=2 SCKE, T_{IDLE1}=1 SCKE, T_{SETUP2}=3 SCKE, and that the data bits per element is 8:

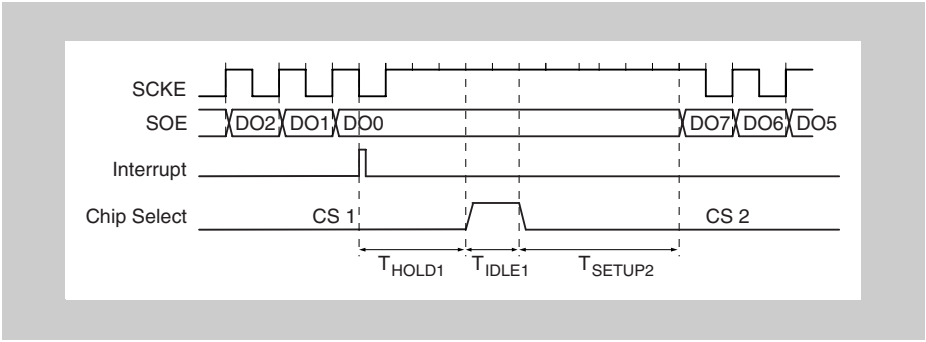
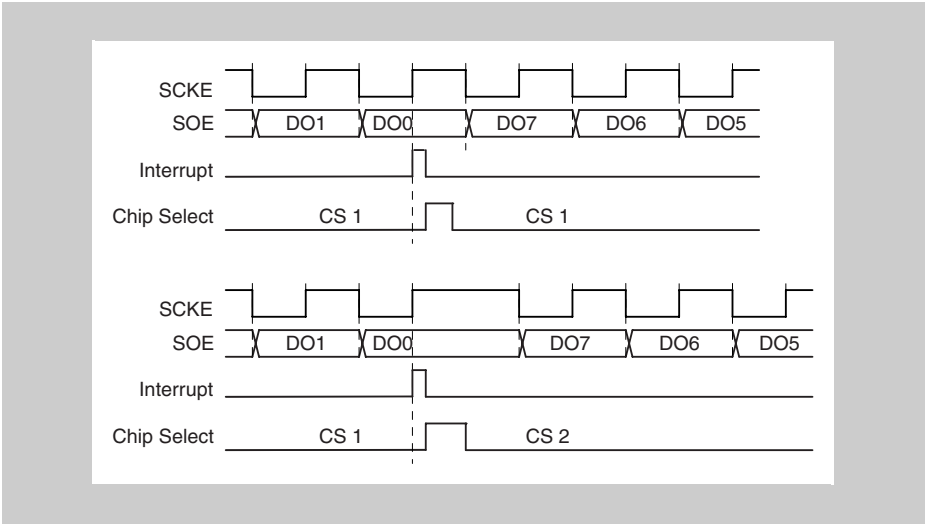


Figure 13-22 Enhanced chip select timing extension relative to SCKE

With the setting of CKP=0 and DAP=1, data is output on the rising edge of SCKE and the data is valid at the falling edge. When the last data bit is sent, the interrupt is generated at the falling edge of the SCKE.

The next figure shows the timing with T_{HOLD1}=T_{IDLE1}=T_{SETUP2}=0, but CEnIDL1=1, CKP/DAP=0/0:



A small chip select idle pulse is generated after the “data valid” edge of SCKE, which is the rising edge in this example due to CKP=DAP=0.

(6) Impact of interrupt delay on enhanced timing

The CEnSIT bit allows to set a delay of half serial communication clock before the transfer complete interrupt is generated. This delay is inserted before any delay defined by the enhanced timing.

Figure 13-23 shows as example a communication with T_{INTER1}=0, T_{HOLD1}=0.5, T_{IDLE1}=1, T_{SETUP2}=0 and T_{INTER2}=1, using single buffer transfer mode and 8-bit data length.

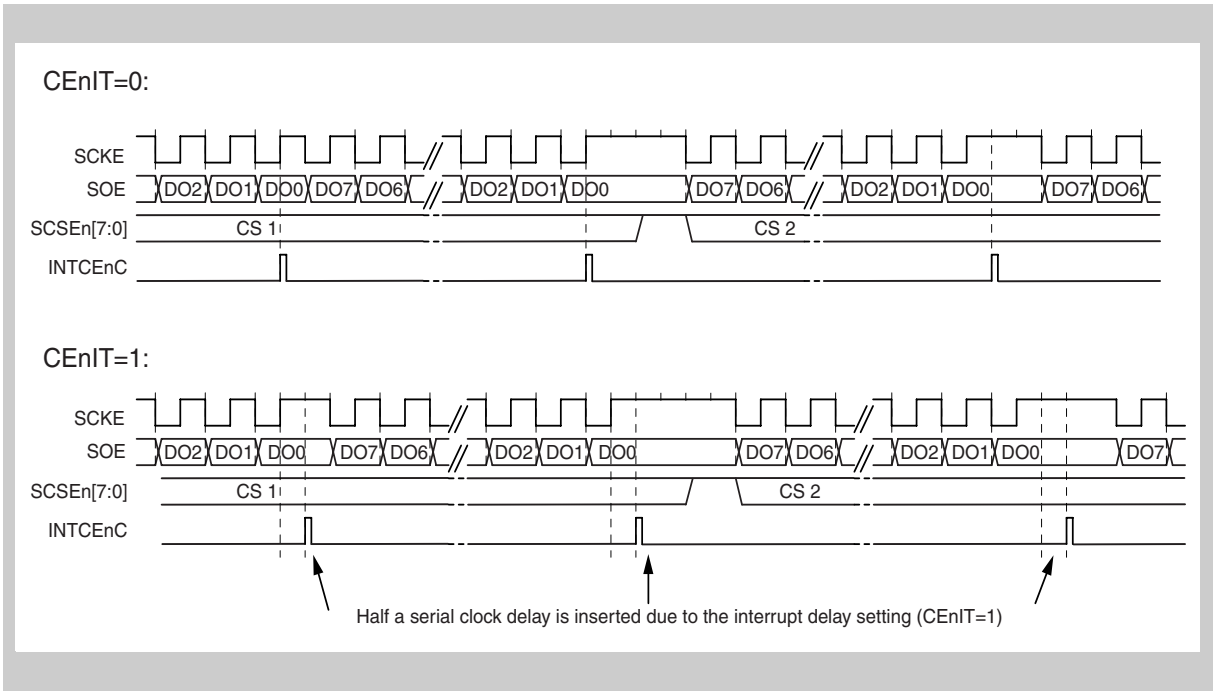


Figure 13-23 Enforced chip select idle timing extension

(7) Data phase details (dap=1) with enhanced timing

The DAP bit defines the phase of the data line relative to the serial clock. While the timing diagram for DAP=0 is straight forward, for DAP=1 there would be two options for the rising edge of the serial clock signal at the last data bit (DO0) when hold time is applied. Please use the figure below as reference.

The rising edge could be either done at the end of the DO0 bit, or it could be done at the end of the hold time (dashed line in the timing diagram). The Queued-CSI module places the rising edge at the end of the data bit.

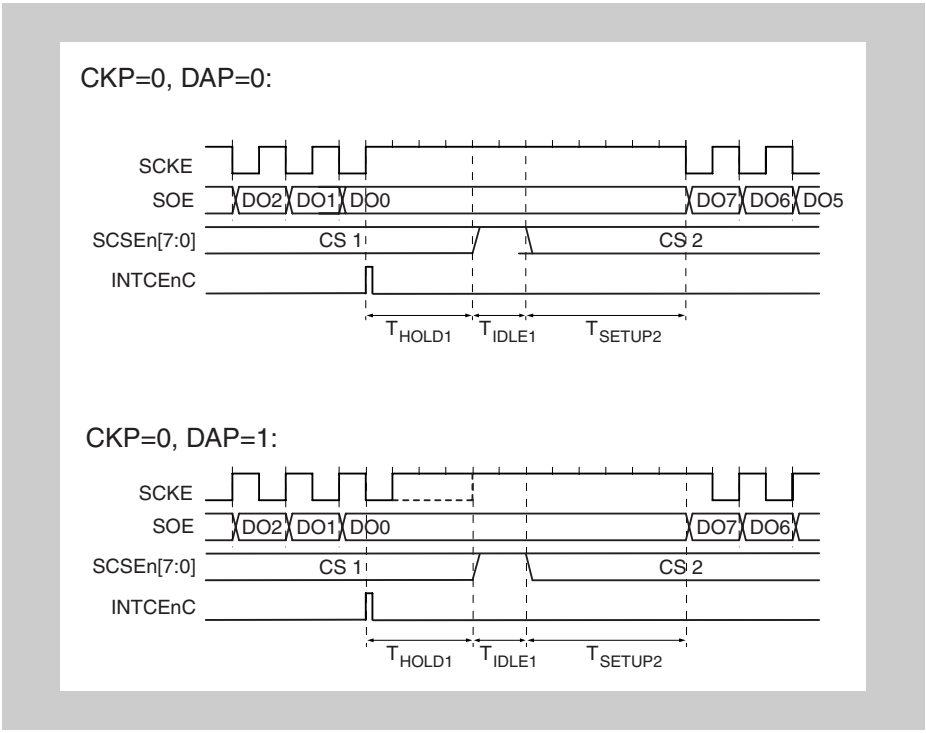


Figure 13-24 Data phase details (DAP=1) with enhanced timing

(8) Change of clock phase and data phase

The enhanced timing also allows the definition of two clock polarity/data phase pairs. The following figures (*Figure 13-25* to *Figure 13-27*) illustrate the transition in polarity down to internal Queued-CSI operating clock level. In a nutshell, there is one macro clock delay between the change in polarity and the activation of the chip select signal.

Except when noted otherwise, it is assumed in the following examples that there is no setup delay defined for CS2. If a delay is defined, the standard rules would apply after the activation of the chip select signal.

Caution *Figure 13-25* shows the synchronisation delay after the new CS signal is activated, while in the other figures the synchronisation delay is ignored to simplify the drawings. The synchronization delay would apply, though!

Figure 13-25 shows the transition from SCSE0 with (CKP=0, DAP=0) to SCSE1 with (CKA=1, DPA=0), using the following settings: T_{HOLD0}=0, T_{IDLE0}=1, T_{SETUP1}=0.5, $f_{SCKE}=f_{QCSI}/16$:

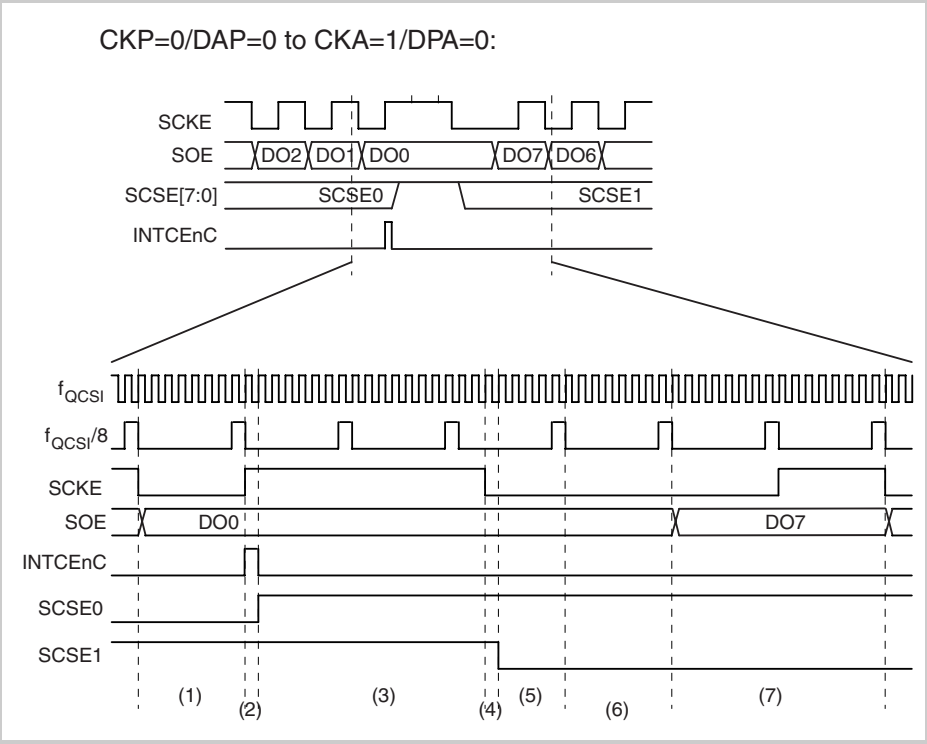


Figure 13-25 CKP=0, DAP=0 to CKA=1, DPA=0 with enhanced timing

1. Is the last first half of the last data bit sent with SCSE0 active
2. At the rising edge of SCKE, the interrupt signal is generated.
3. Represents the idle time (T_{IDLE0}=1). One f_{QCSI} clock after the generation of the interrupt signal, SCSE0 is set to inactive. It is hold inactive for serial clock cycle plus one internal clock f_{QCSI}.
4. After the idle time, the serial clock polarity is changed due to the CKA=1 transition.
5. One internal clock f_{QCSI} after the change of the clock polarity SCSE1 is set

to active. Then the timing is synchronized with the internal baud rate generator that operates continuously at $f_{QCSI}/8$.

6. Is the setup delay of SCSE1 (T_{SETUP1}=0.5).
7. Is the output of the first data bit with SCSEn1 active.

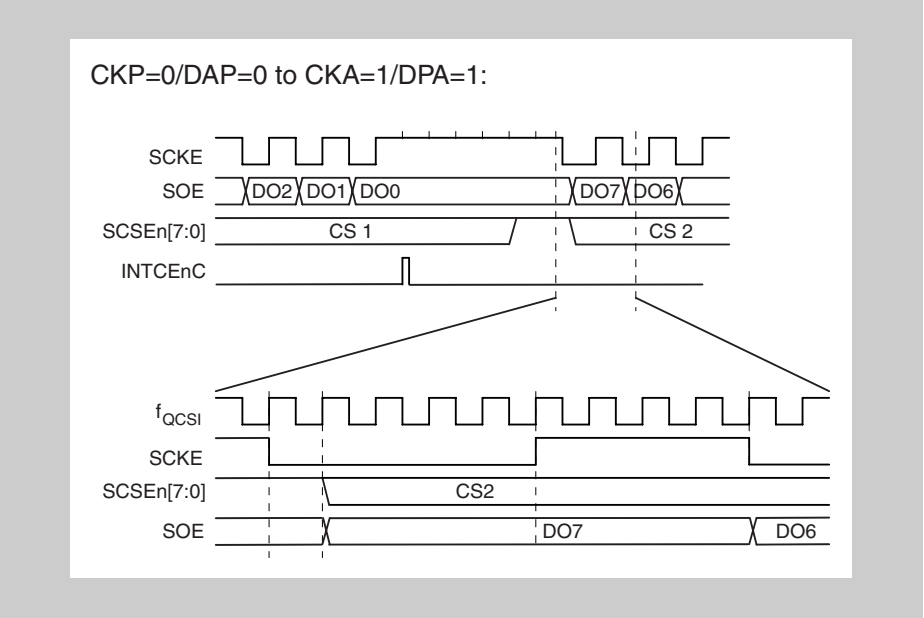


Figure 13-26 CKP=0, DAP=0 to CKA=1, DPA=1 with enhanced timing

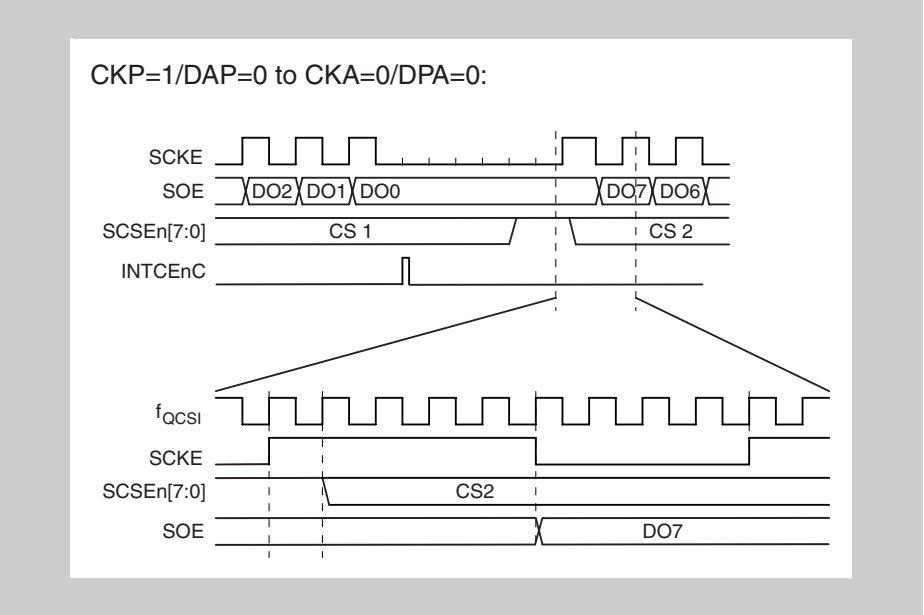


Figure 13-27 CKP=1, DAP=0 to CKA=0, DPA=1 with enhanced timing

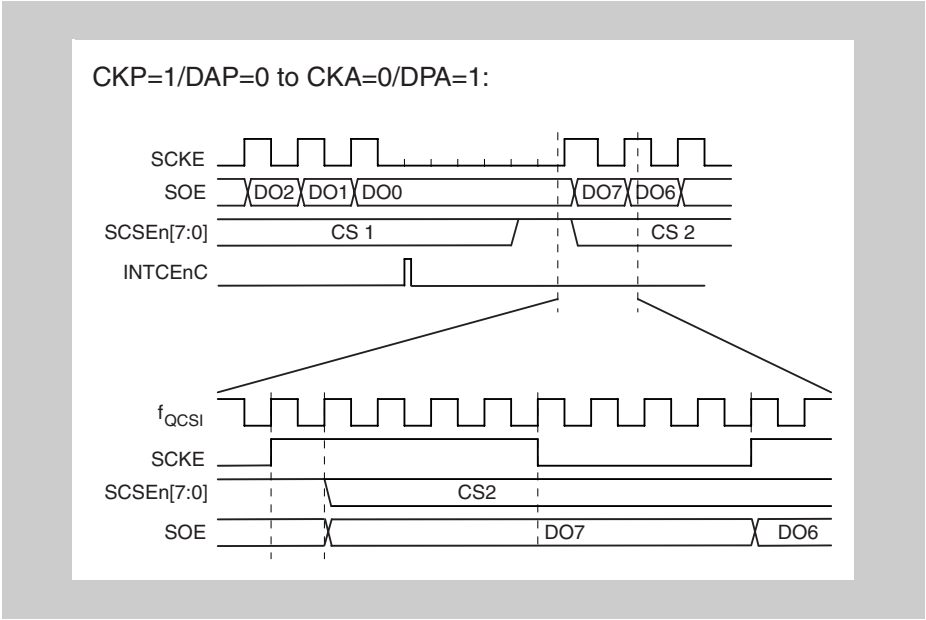


Figure 13-28 CKP=1, DAP=0 to CKA=0, DPA=0 with enhanced timing

13.4 Operating Procedure

13.4.1 Single transfer mode (master mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT=0), transmission wait disabled (CEnWE = 0),

CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

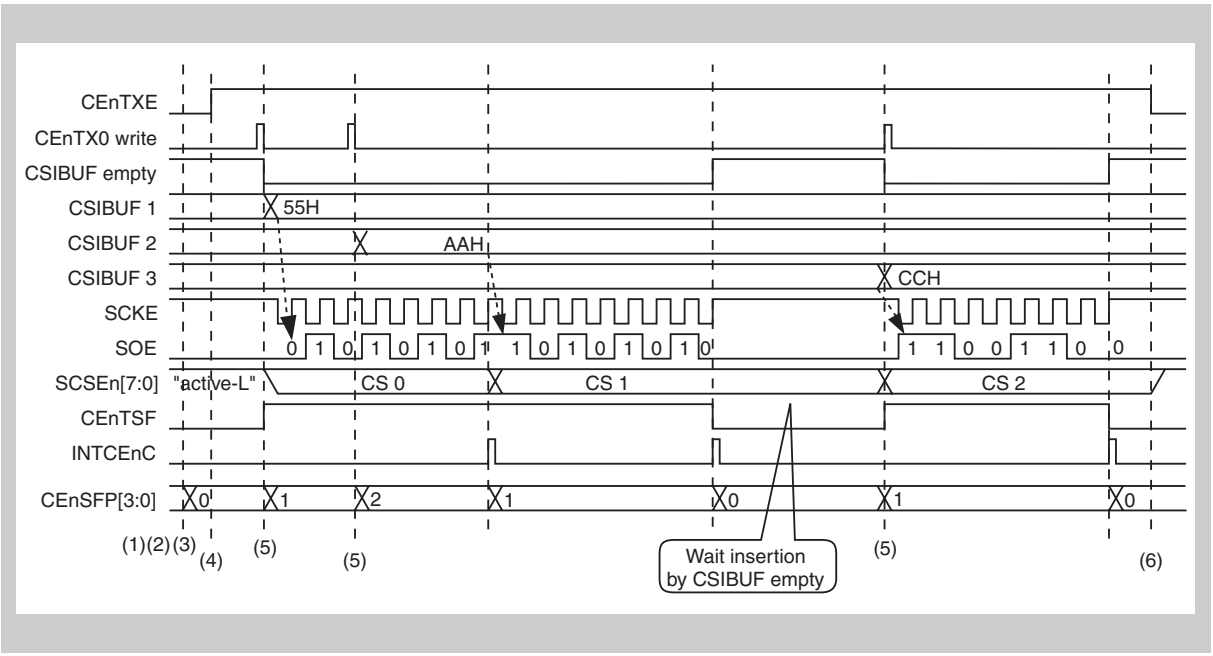


Figure 13-29 Single transfer mode (master, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.

   Repeat step (5) until the last element to be transmitted is written in the CEnCS/CEnTX0 registers.

6. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

13.4.2 Single transfer mode (master mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

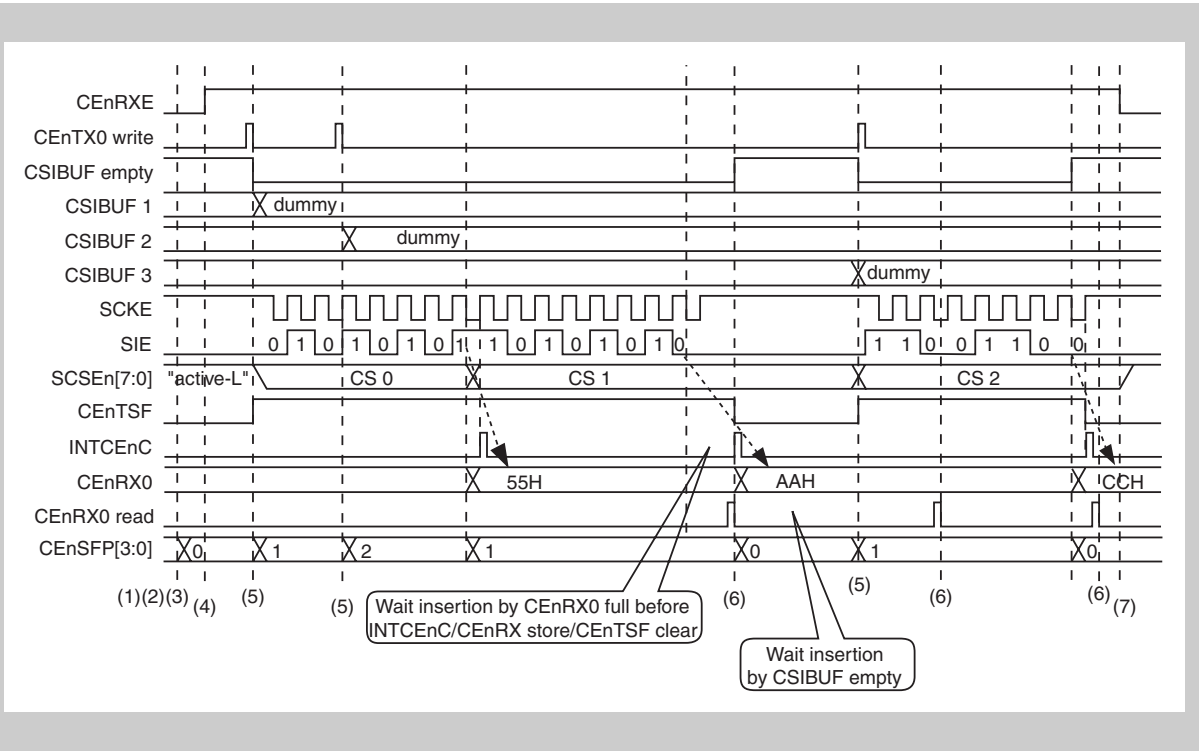


Figure 13-30 Single transfer mode (master, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and dummy transmission data in the CEnCS and CEnTX0 registers in this order (start-of-receive trigger).
6. Check for a reception to be completed by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

   Repeat steps (5) and (6) until the last element is received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

Note The SOE pin is invalid and maintains its signal level, as the output latch is disabled.

13.4.3 Single transfer mode (master mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

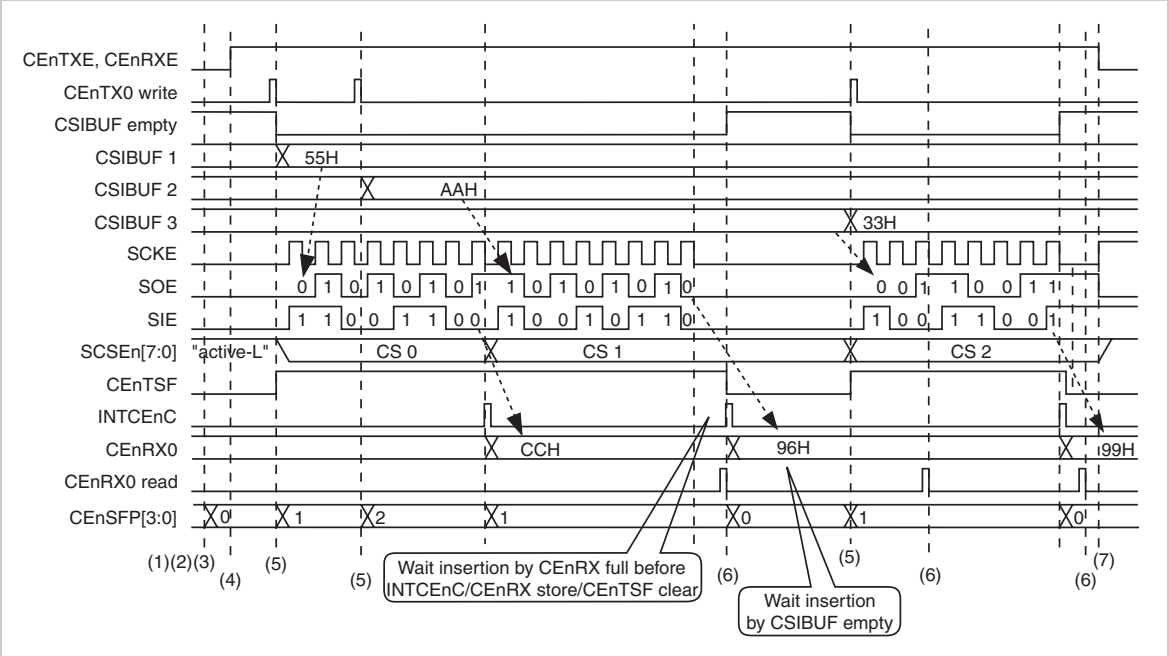


Figure 13-31 Single transfer mode (master, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.
6. Check for a transmission to be finished by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

   Repeat steps (5) and (6) until the last element is send/received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

13.4.4 Single transfer mode (slave mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

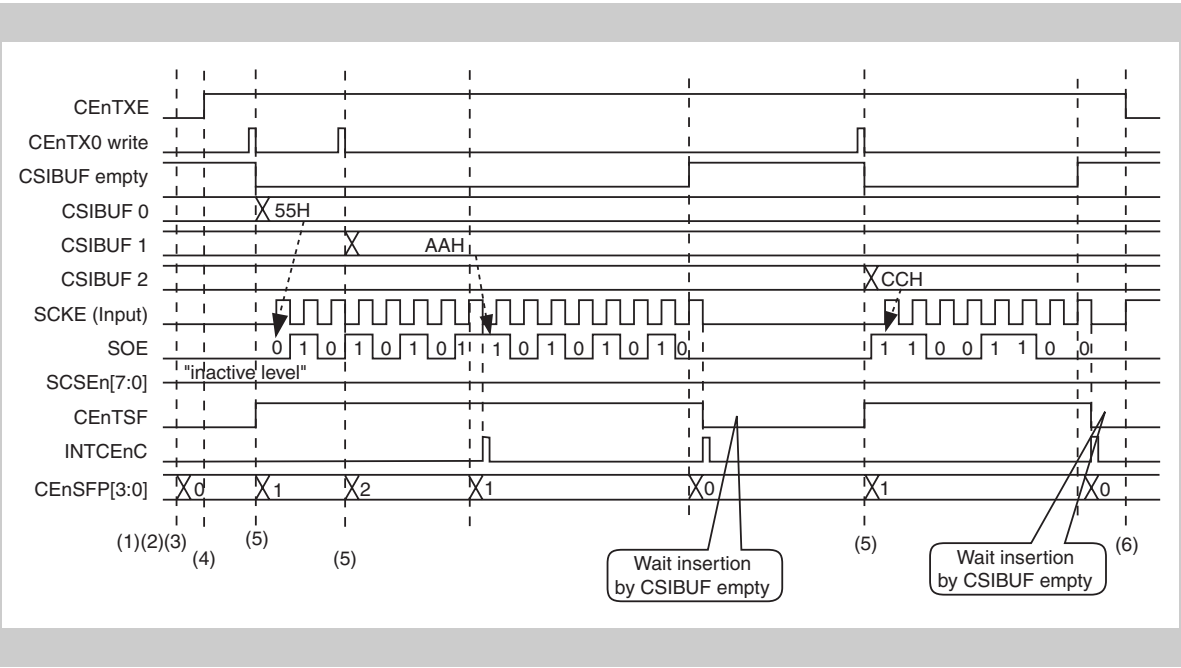


Figure 13-32 Single transfer mode (slave, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)

   Repeat step (5) until the last element to be transmitted is written in the CEnTX0 register.

6. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

To continue transmission, repeat step (5) before executing step (6).

13.4.5 Single transfer mode (slave mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

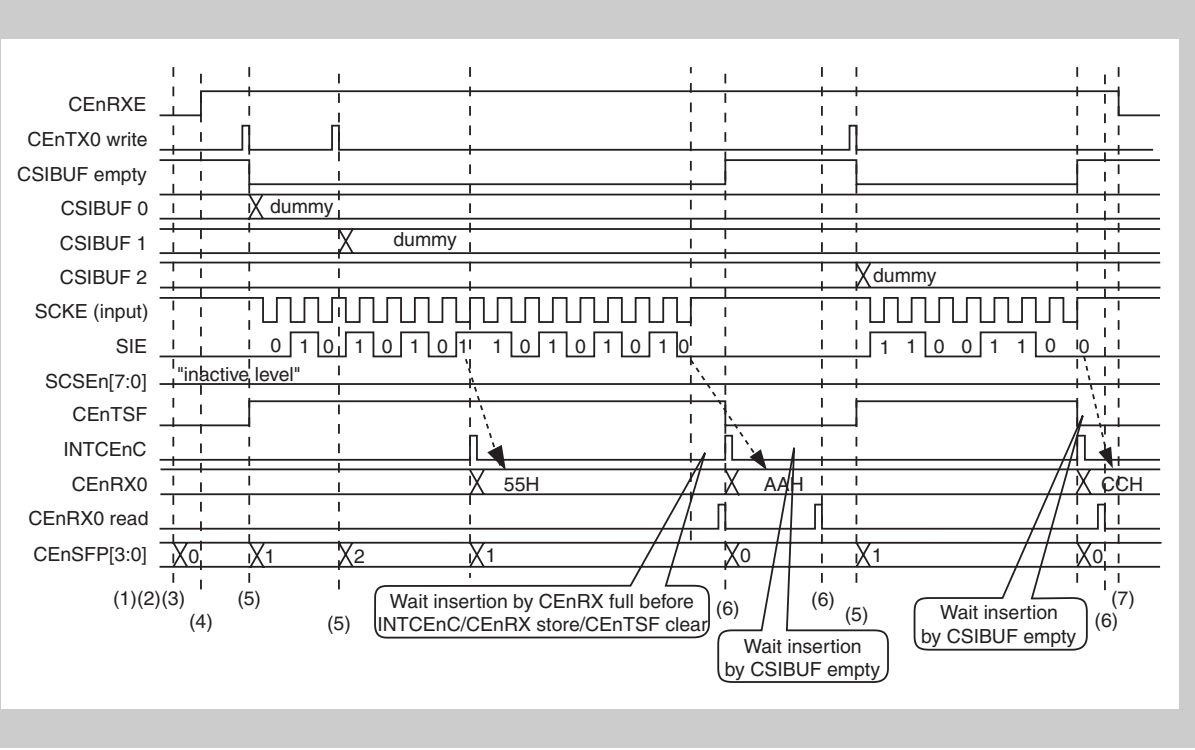


Figure 13-33 Single transfer mode (slave, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write dummy transmission data in the CEnTX0 register (start-of-receive trigger). (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
6. Check for a reception to be completed by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

   Repeat steps (5) and (6) until the last element is received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

Note The SOEn pin is invalid and maintaining its signal level, as the output latch is disabled.

13.4.6 Single transfer mode (slave mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

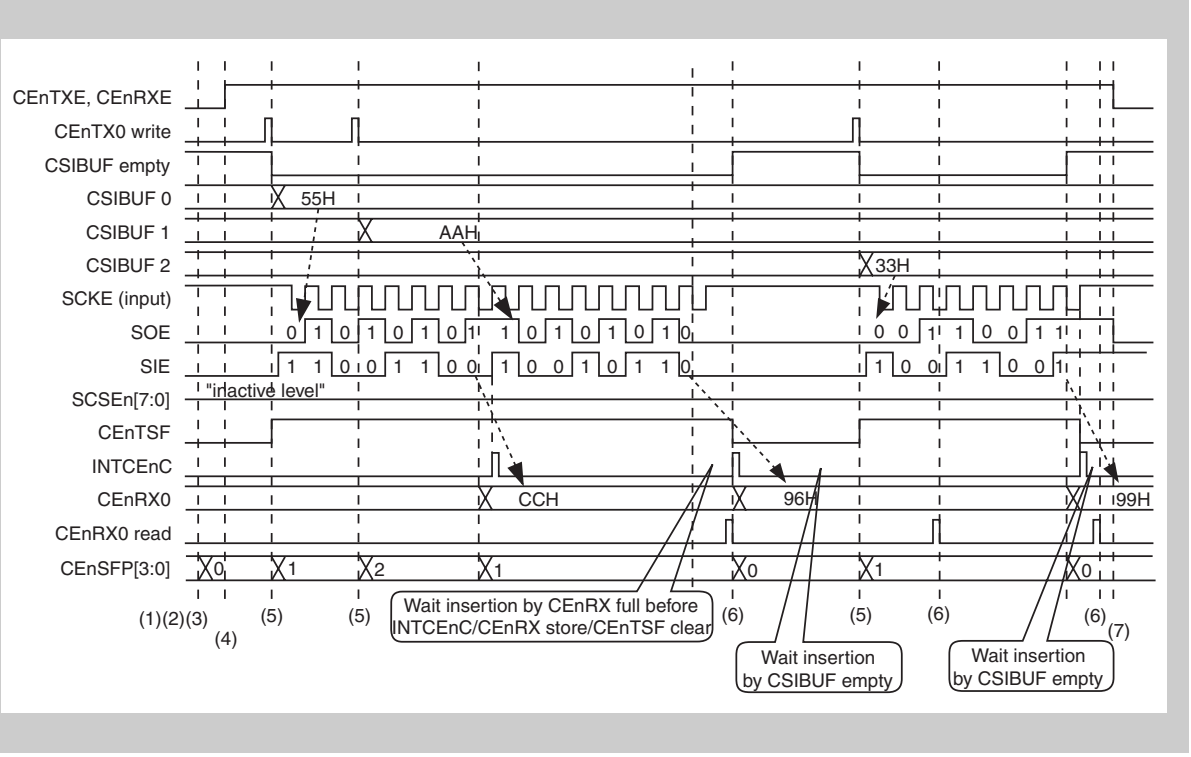


Figure 13-34 Single transfer mode (slave, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transmit mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
6. Check for a reception to be completed by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

   Repeat steps (5) and (6) until the last element is send/received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

13.4.7 Block transfer mode (master mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

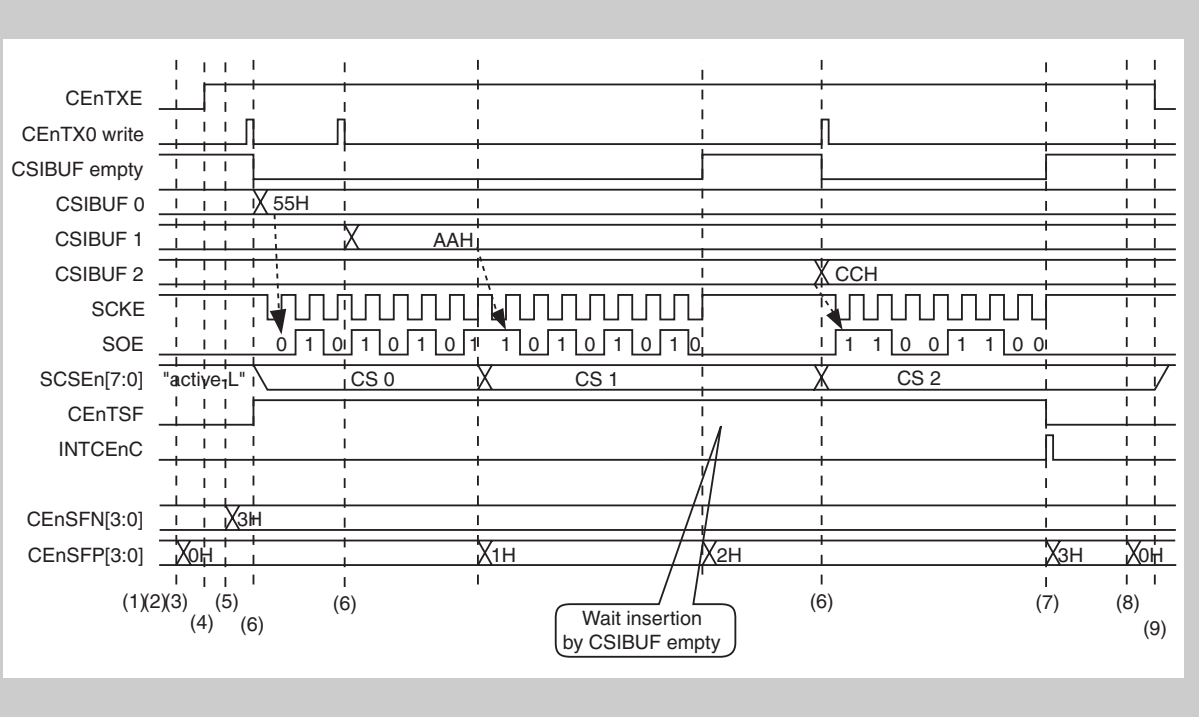


Figure 13-35 Block transfer mode (master, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Set the number of send-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.
7. Wait for the transmissions to be completed by monitoring the INTCEnC interrupt.
8. Write “1” in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

   To continue transmission, repeat steps (5) - (8).

9. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

13.4.8 Block transfer mode (master mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

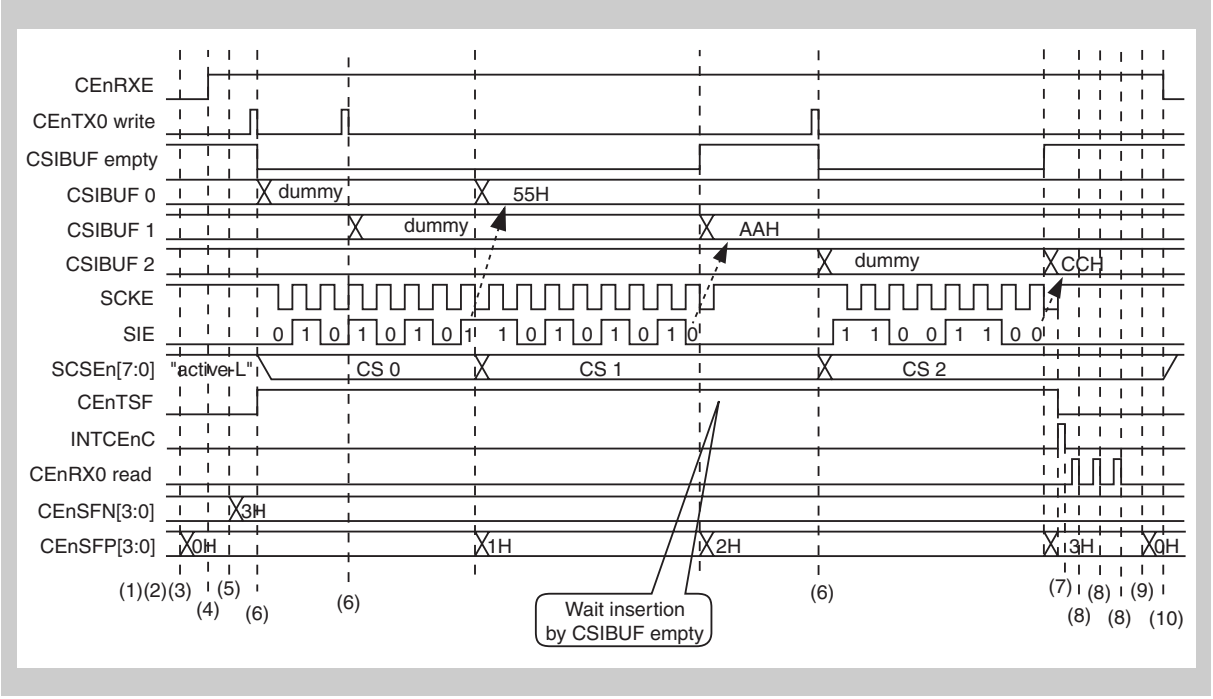


Figure 13-36 Block transfer mode (master, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Set the number of receive-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and dummy transmission data in the CEnCS and CEnTX0 registers in this order (start-of-receive trigger).
7. Wait for the receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write “1” in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

   To continue reception, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

Note The SOEn pin is invalid and maintains its signal level, as the output latch is disabled.

13.4.9 Block transfer mode (master mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

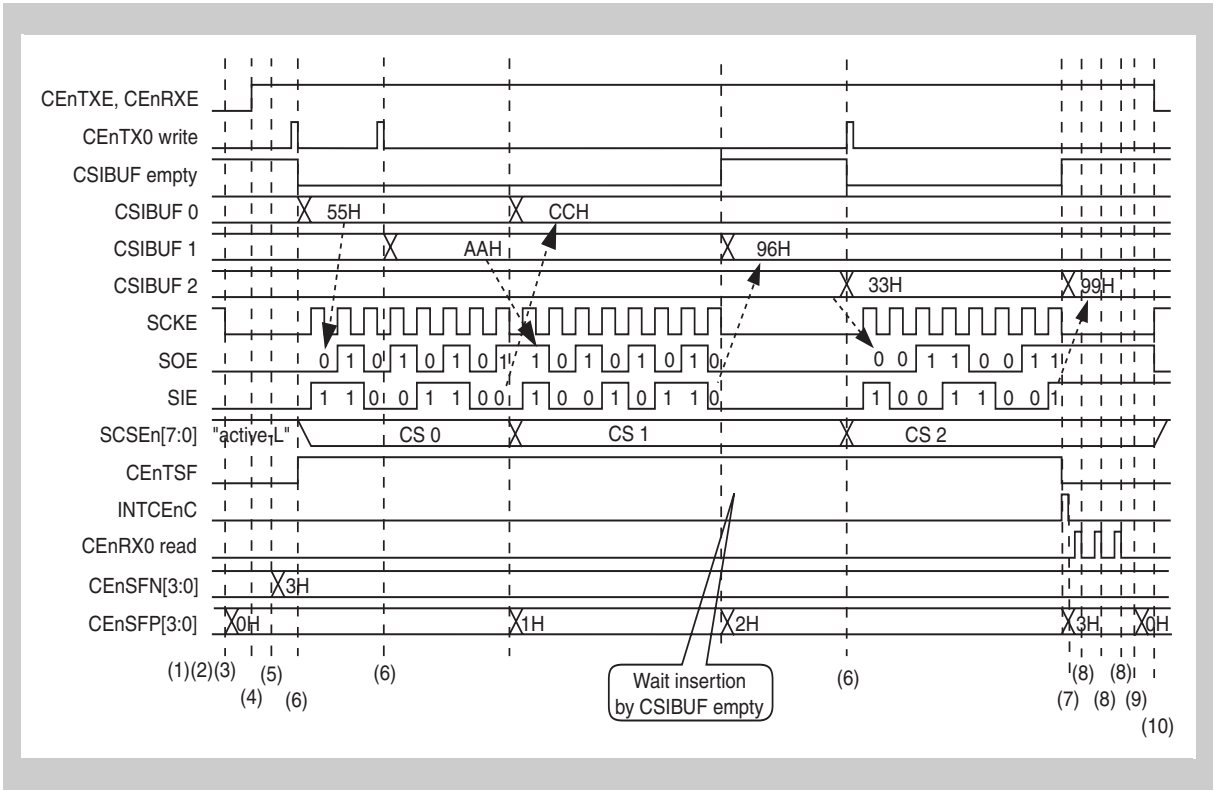


Figure 13-37 Block transfer mode (master, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Set the number of transmit/receive data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.
7. Wait for the transmissions/receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write “1” in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

   To continue transmission/reception, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

13.4.10 Block transfer mode (slave mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

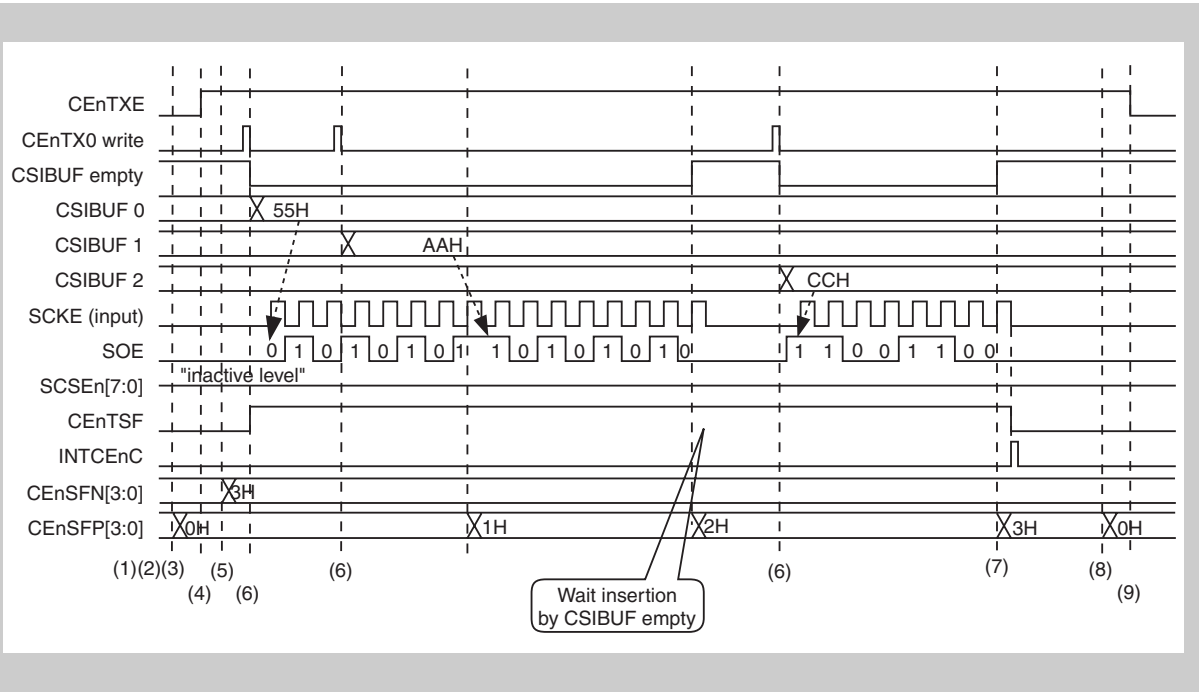


Figure 13-38 Block transfer mode (slave, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Set the number of send-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
7. Wait for the transmissions to be completed by monitoring the INTCEnC interrupt.
8. Write “1” in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

   To continue transmission, repeat steps (5) - (8).

9. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

13.4.11 Block transfer mode (slave mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

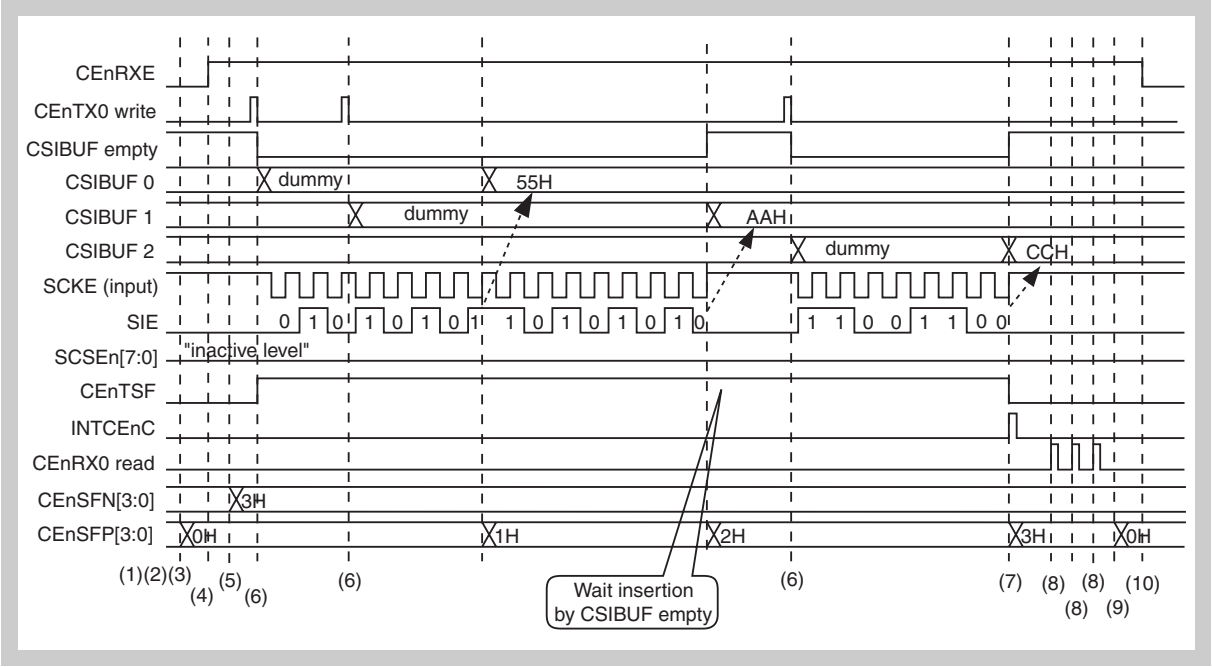


Figure 13-39 Block transfer mode (slave, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Set the number of receive-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write dummy transmission data in the CEnTX0 register (start-of-receive trigger). (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
7. Wait for the receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write “1” in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

   To continue reception, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

Note The SOEn pin is invalid and maintains its signal level, as the output latch is disabled.

13.4.12 Block transfer mode (slave mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

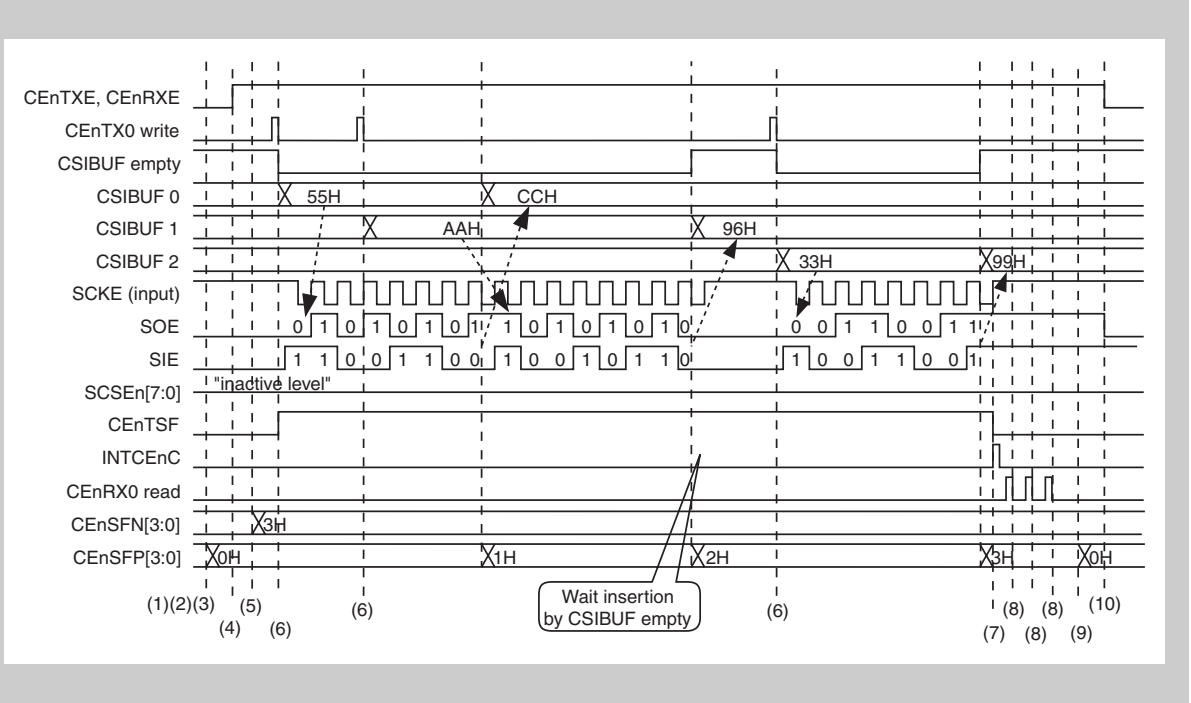


Figure 13-40 Block transfer mode (slave, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write “1” in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Set the number of transmit/receive data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
7. Wait for the transmissions/receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write “1” in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

   To continue transmission, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIEn in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

13.5 Cautions

13.5.1 Inconsistent read data in master mode

Description Data read from the CEnRx0 register may be incorrect due to a wrong data width being applied to the received data. The behaviour occurs if all of the following conditions are met:

- Enhanced timing mode is used (CEnCTL4.CEnOPE = 1)
- Different data width configured for each chip select SCSEnm (different CEnOPTm.CEnDLm[1:0])
- Master mode is enabled (CEnCTL1.CEnCKS[2:0] $\neq 111_B$]
- Receive or Transmit-Receive is enabled (CEnCTL0.CEnRXE = 1)
- Serial data direction is LSB first (CEnCTL0.CEnDIR = 1)

Workaround Configure the CSIEn in such a way that not all of above mentioned conditions apply, e.g. use MSB first or configure the same data length for each chip select SCSEnm.

13.5.2 Transmit/receive enable after CEnTX0 write

Description If transmission (CEnCTL0.CEnTXE = 1) or reception (CEnCTL0.CEnRXE = 1) is enabled after writing data to the transmission data buffer register CEnTX0 the receive/transmission operation may fail.

Workaround Enable transmission/reception before writing to CEnTX0.

Chapter 14 CAN Controller (CAN)

The microcontroller features an on-chip 2-channel CAN (Controller Area Network) controller that complies with the CAN protocol as standardized in ISO 11898.

Note
1. Throughout this chapter, the individual CAN channels are identified by “n” (n = 0, 1), for example CANn, or CnGMCTRL for the CANn global control register.
2. Throughout this chapter, the CAN message buffer registers are identified by “m” (m = 0 to 31), for example C0MDATA4m for CAN0 message data byte 4 of message buffer register m.

14.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (if CAN clock input ≥ 8 MHz, for 32 channels)
- 32 message buffers per channel
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel
- Data bit time, communication baud rate and sample point can be controlled by CAN module bit-rate prescaler register (CnBRP) and bit rate register (CnBTR)
  - As an example the following sample-point configurations can be configured:
  - 66.7%, 70.0%, 75.0%, 80.0%, 81.3%, 85.0%, 87.5%
  - Baud rates in the range of 10 kbps up to 1000 kbps can be configured
- Enhanced features:
  - Each message buffer can be configured to operate as a transmit or a receive message buffer
  - Transmission priority is controlled by the identifier or by mailbox number (selectable)
  - A transmission request can be aborted by clearing the dedicated Transmit-Request flag of the concerned message buffer.
  - Automatic block transmission operation mode (ABT)
  - Time stamp function for CAN channels 0 to n in collaboration with timers capture channels

14.1.1 Overview of functions

Table 14-1 presents an overview of the CAN Controller functions.

Table 14-1 Overview of functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 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)

14.1.2 Configuration

The CAN Controller is composed of the following four blocks.

- NPB interface
  This functional block provides an NPB (NEC 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.

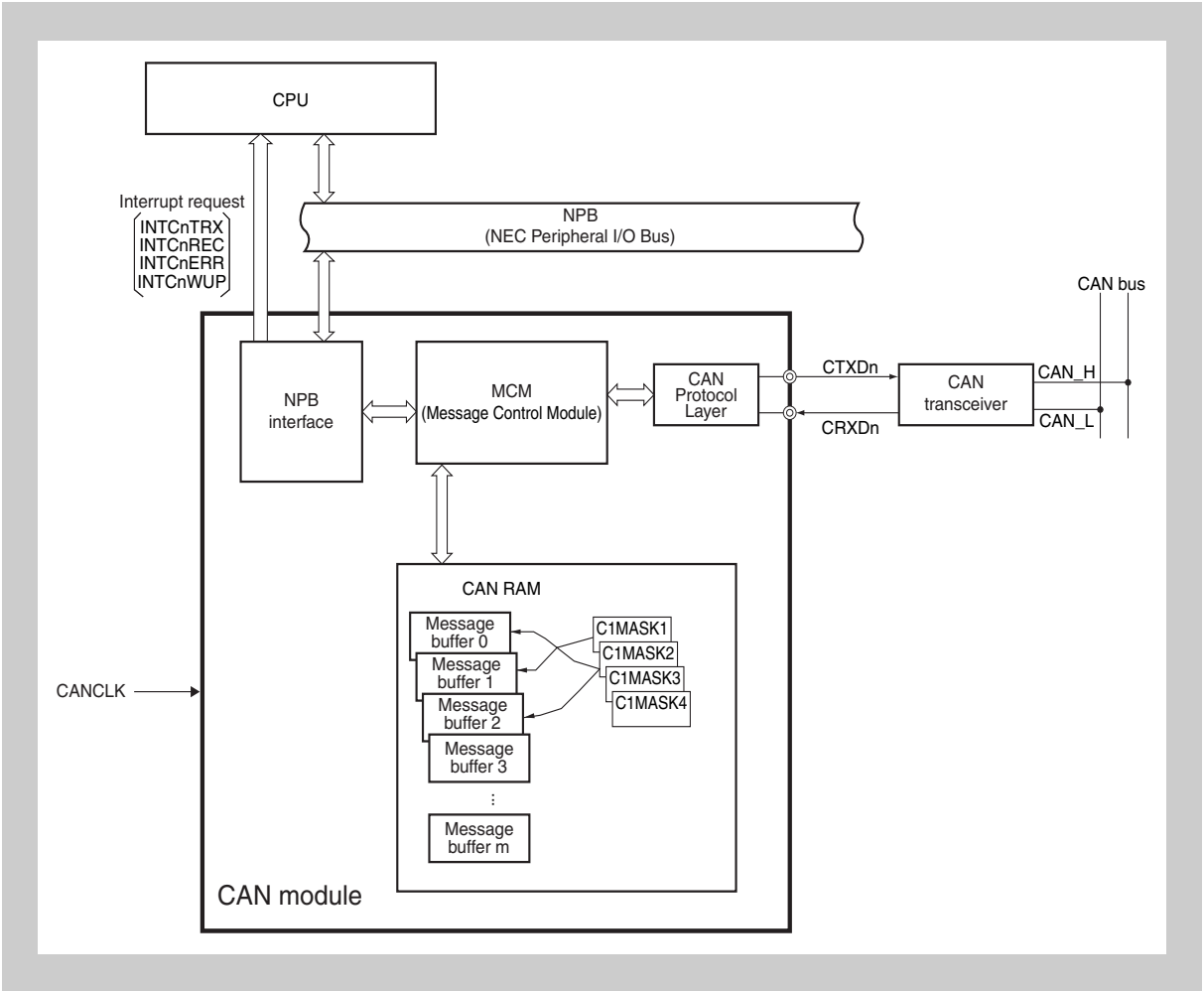


Figure 14-1 Block diagram of CAN module

14.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.

The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

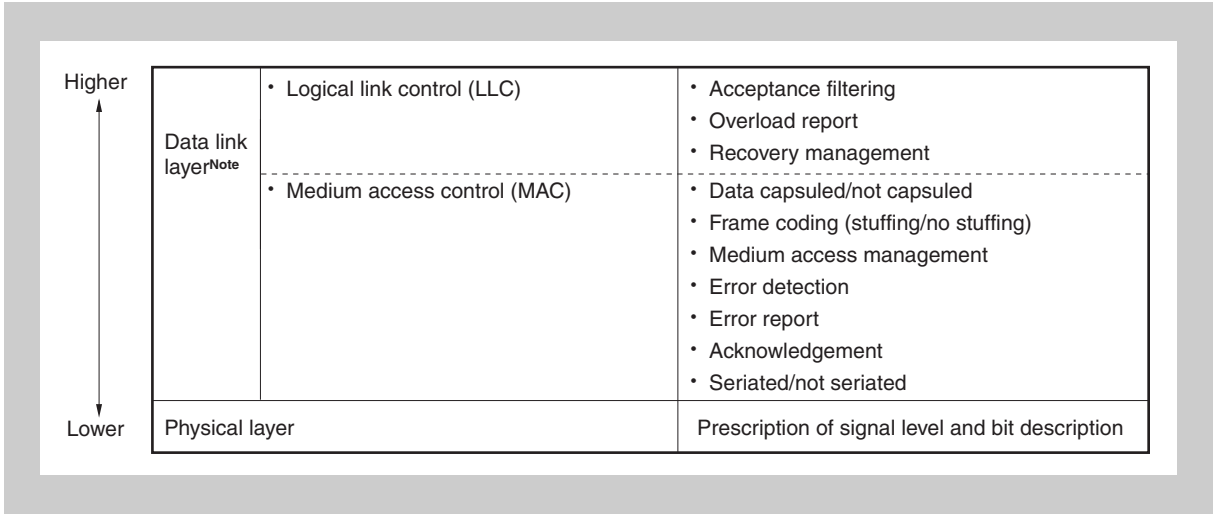


Figure 14-2 Composition of layers

Note CAN Controller specification

14.2.1 Frame format

(1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2,048 messages.

(2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers, which increases the number of messages that can be handled to $2{,}048 \times 2^{18}$ messages.
- An extended format frame is set when “recessive level” (CMOS level of “1”) is set for both the SRR and IDE bits in the arbitration field.

14.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 14-2 Frame types

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

14.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

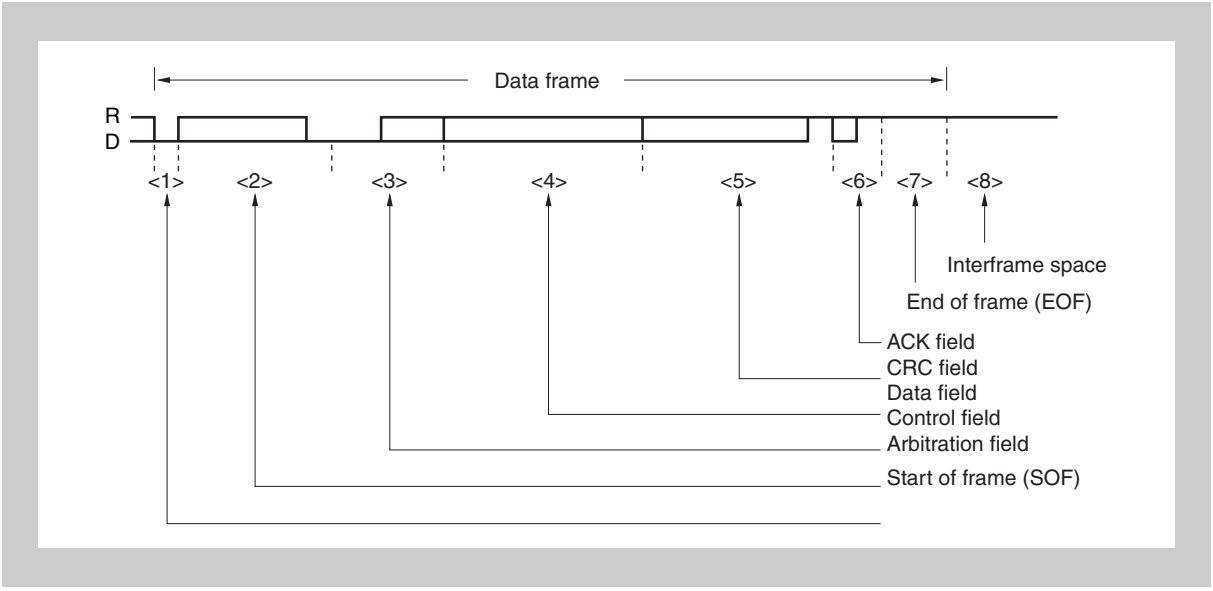


Figure 14-3 Data frame

Note D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

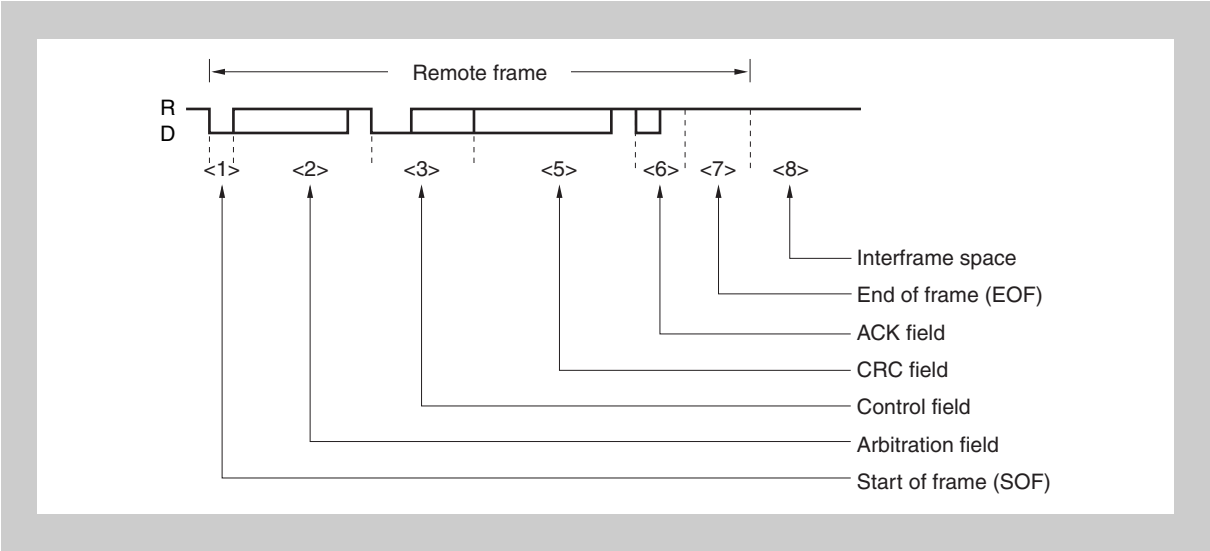


Figure 14-4 Remote frame

Note 1. The data field is not transferred even if the control field's data length code is not "0000_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.

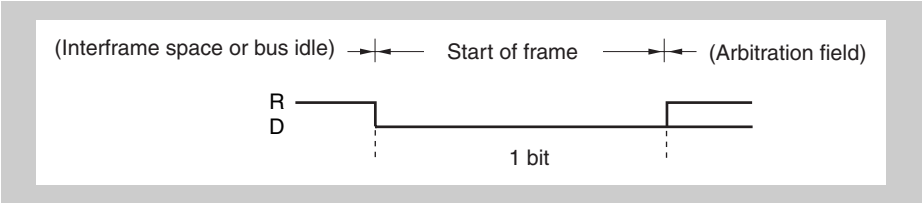


Figure 14-5 Start of frame (SOF)

Note D: Dominant = 0
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

(b) Arbitration field

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

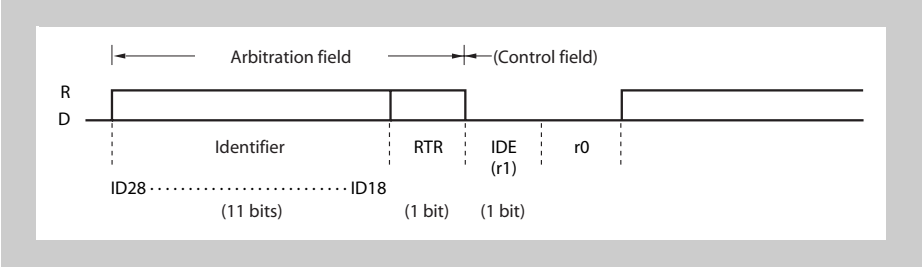


Figure 14-6 Arbitration field (in standard format mode)

Caution 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Note D: Dominant = 0
R: Recessive = 1

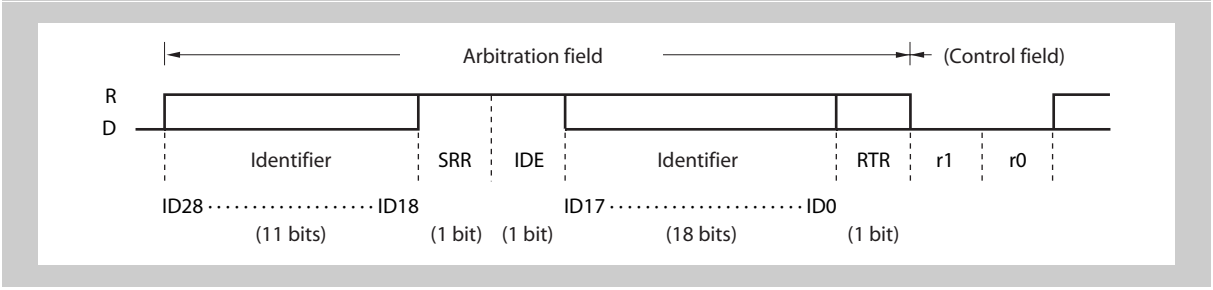


Figure 14-7 Arbitration field (in extended format mode)

Caution 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Note D: Dominant = 0
R: Recessive = 1

Table 14-3 RTR frame settings

Frame type	RTR bit
Data frame	0 (D)
Remote frame	1 (R)

Table 14-4 Frame format setting (IDE bit) and number of identifier (ID) bits

Frame format	SRR bit	IDE bit	Number of bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

(c) Control field

The control field sets “DLC” as the number of data bytes in the data field (DLC = 0 to 8).

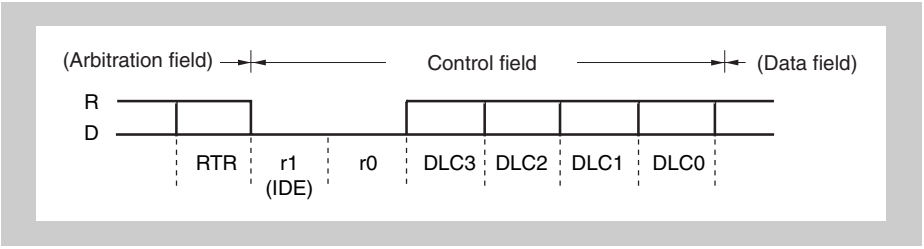


Figure 14-8 Control field

Note D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 14-5 Data length setting

Data length code				Data byte count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution In the remote frame, there is no data field even if the data length code is not 0000_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.

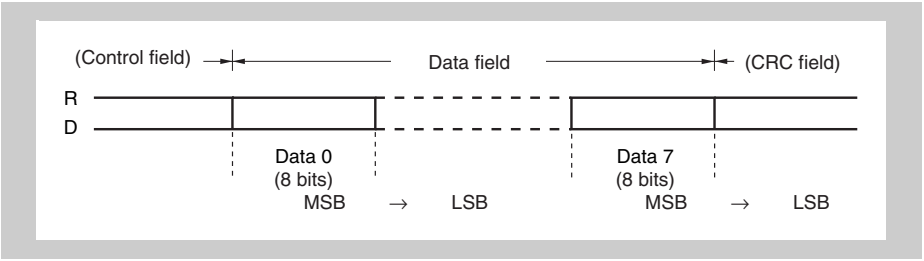


Figure 14-9 Data field

Note D: Dominant = 0
R: Recessive = 1

(e) CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

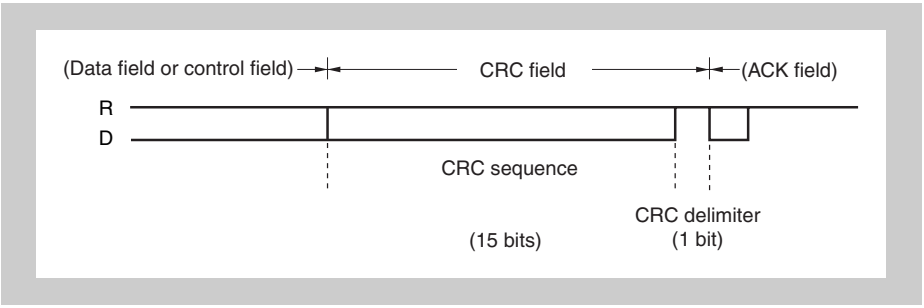


Figure 14-10 CRC field

Note D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.

  $P(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$

- Transmitting node: Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.
- Receiving node: Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

(f) ACK field

The ACK field is used to acknowledge normal reception.

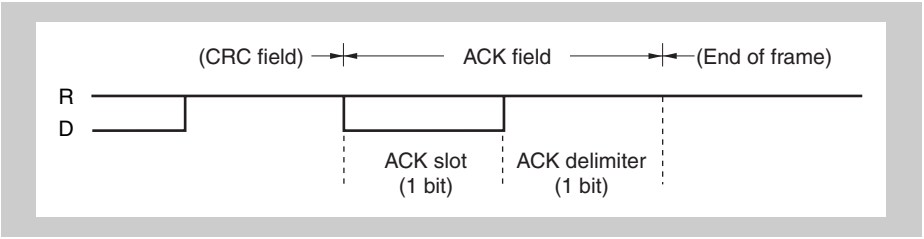


Figure 14-11 ACK field

Note D: Dominant = 0
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

(g) End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

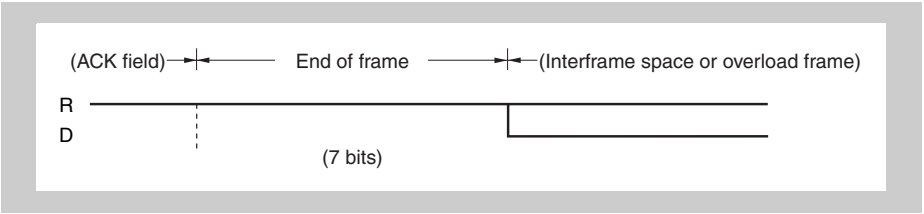


Figure 14-12 End of frame (EOF)

Note D: Dominant = 0
R: Recessive = 1

(h) Interframe space

The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

  – Error active node

  The interframe space consists of a 3-bit intermission field and a bus idle field.

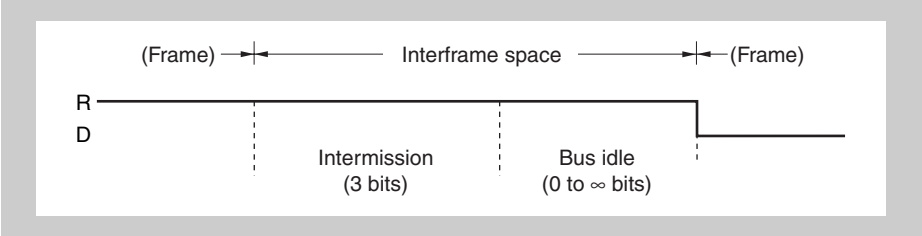


Figure 14-13 Interframe space (error active node)

Note 1. Bus idle: State in which the bus is not used by any node.

2. D: Dominant = 0
   R: Recessive = 1

  – Error passive node

  The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.

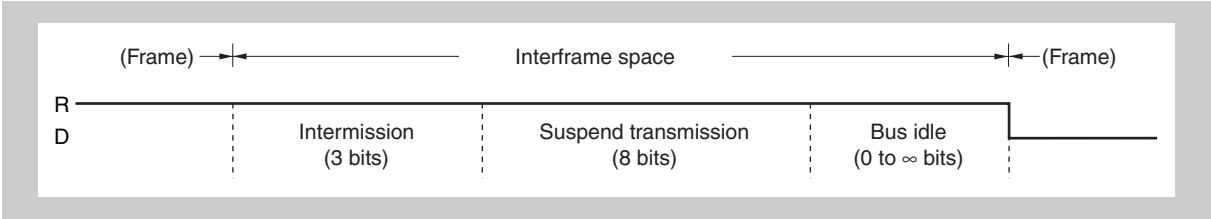


Figure 14-14 Interframe space (error passive node)

Note 1. Bus idle: State in which the bus is not used by any node.
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.

2. D: Dominant = 0
   R: Recessive = 1

  Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

- Operation in error status

Table 14-6 Operation in error status

Error status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

14.2.4 Error frame

An error frame is output by a node that has detected an error.

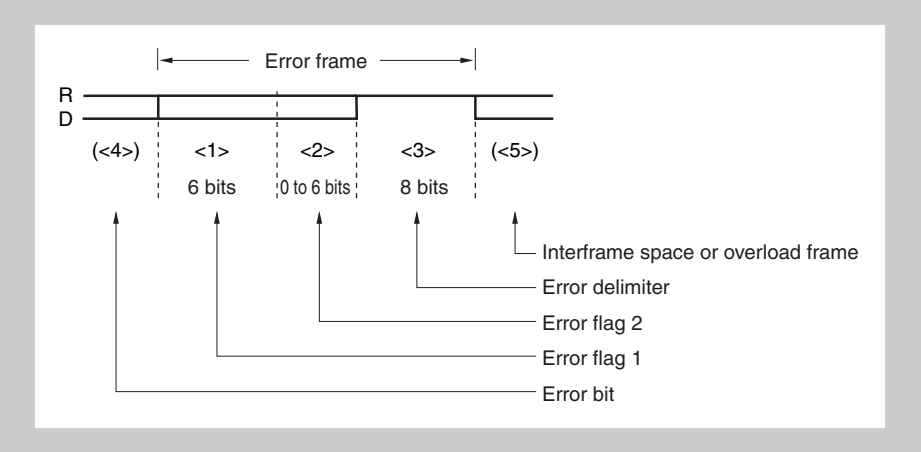


Figure 14-15 Error frame

Note D: Dominant = 0
R: Recessive = 1

Table 14-7 Definition of error frame fields

No.	Name	Bit count	Definition
<1>	Error flag 1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively. If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag 2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

14.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter

Note The CAN is internally fast enough to process all received frames not generating overload frames.

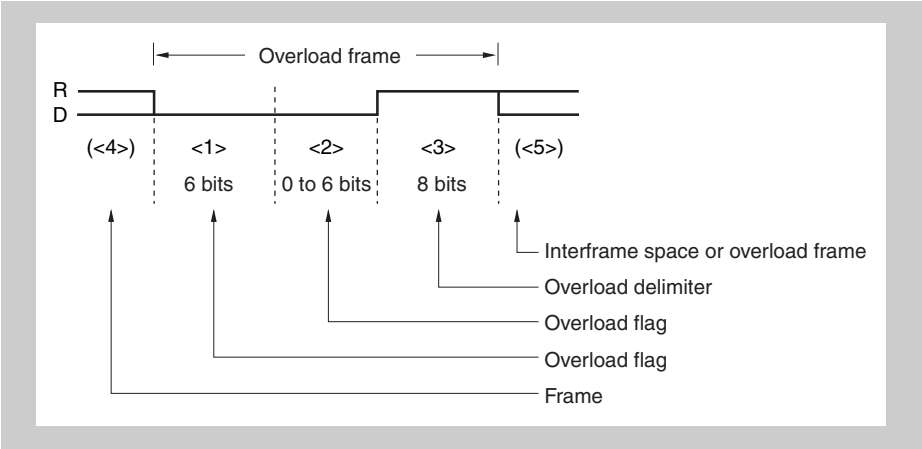


Figure 14-16 Overload frame

Note D: Dominant = 0
R: Recessive = 1

Table 14-8 Definition of overload frame fields

No	Name	Bit count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

14.3 Functions

14.3.1 Determining bus priority

(1) When a node starts transmission:

- During bus idle, the node that output data first transmits the data.

(2) When more than one node starts transmission:

- The node that consecutively outputs the dominant level for the longest from the first bit of the arbitration field has the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).
- The transmitting node compares its output arbitration field and the data level on the bus.

Table 14-9 Determining bus priority

Level match	Continuous transmission
Level mismatch	Stops transmission at the bit where mismatch is detected and starts reception at the following bit

(3) Priority of data frame and remote frame

- When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

Note If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frame takes priority.

14.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1 bit of inverted-level data if the same level continues for 5 bits, in order to prevent a burst error.

Table 14-10 Bit stuffing

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

14.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

14.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

14.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN Controller in waiting mode to achieve low power consumption.

The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

14.3.6 Error control function

(1) Error types

Table 14-11 Error types

Type	Description of error		Detection state	
	Detection method	Detection condition	Transmission/reception	Field/frame
Bit error	Comparison of the output level and level on the bus (except stuff bit)	Mismatch of levels	Transmitting/ receiving node	Bit that is outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check of the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	CRC delimiter ACK field End of frame Error frame Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

(2) Output timing of error frame

Table 14-12 Output timing of error frame

Type	Output timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CEC error	Error frame output is started at the timing of the bit following the ACK delimiter.

(3) Processing in case of error

The transmission node re-transmits the data frame or remote frame after the error frame. (However, it does not re-transmit the frame in the single-shot mode.)

(4) Error state

(a) Types of error states

The following three types of error states are defined by the CAN specification:

- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) as shown in *Table 14-13*.

The present error state is indicated by the CAN module information register (CnINFO).

When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the CnINFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the CnINFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the CnINFO register is set to 1.
- If only one node is active on the bus at startup (i.e., a particular case such as when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

Table 14-13 Types of error states

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 14-14 Error counter

State	Transmission error counter (TEC7 to TEC0 bits)	Reception error counter (REC6 to REC0 bits)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (when REPS = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (when REPS = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1> ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2> A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8 (REPS bit = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (transmitting)	+8 (during reception, when REPS = 0)
When the transmitting node has completed transmission without error (±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 14-17 on page 448*). 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 14-17 on page 448*). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions are satisfied. When the recovery conditions are satisfied (refer to timing <3> in *Figure 14-17 on page 448*), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Completion to be requested operation mode can be confirmed by reading the OPMODE bits of the CnCTRL register.

During the bus-off period and bus-off recovery sequence, the BOFF bit of the CnINFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (REC[6:0]) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC[6:0].

Caution In the bus-off recovery sequence, REC[6:0] counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. To start the bus-off recovery sequence, it is necessary to transit to the initialization mode once. However, when the CAN module is in either CAN sleep mode or CAN stop mode, transition request to the initialization mode is not accepted, thus you have to release the CAN sleep mode first. In this case, as soon as the CAN sleep mode is released, the bus-off recovery sequence starts and no transition to initialization mode is necessary. If the can module detects a dominant edge on the CAN bus while in sleep mode even during bus-off, the sleep mode will be left and the bus-off recovery sequence will start.

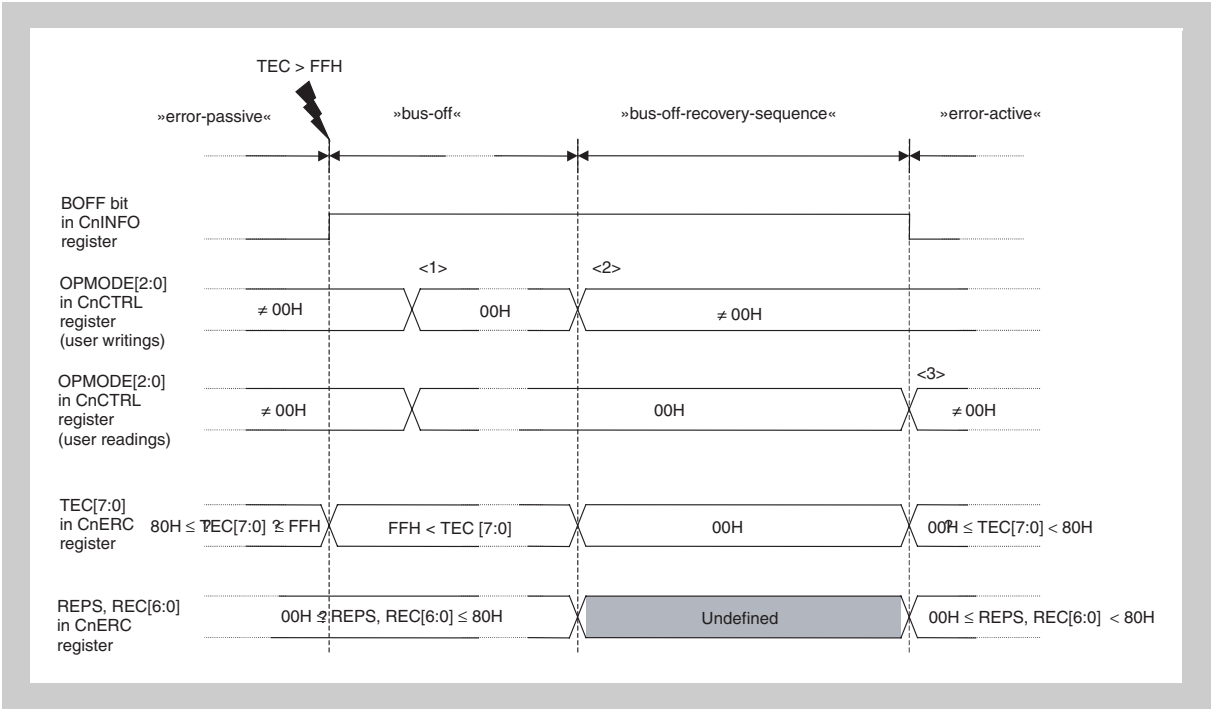


Figure 14-17 Recovery from bus-off state through normal recovery sequence

(b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.

First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, *"Recovery from bus-off state through normal recovery sequence" on page 447*.

Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the CnCTRL register must be set to 1.

As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, refer to the processing in *Figure 14-55 on page 559*.

Caution This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

(6) Initializing CAN module error counter register (CnERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (CnERC) and CAN module information register (CnINFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the CnCTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

Caution
1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the CnERC and CnINFO registers are not initialized.
2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

14.3.7 Baud rate control function

(1) Prescaler

The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer basic system clock (f_{TQ}) derived from the CAN module system clock (f_{CANMOD}), and divided by 1 to 256 (*"CnBRP - CANn module bit rate prescaler register" on page 481*).

(2) Data bit time (8 to 25 time quanta)

One data bit time is defined as shown in *Figure 14-18 on page 450*.

The CAN Controller sets time segment 1, time segment 2, and reSynchronization Jump Width (SJW) of data bit time, as shown in *Figure 14-18*. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

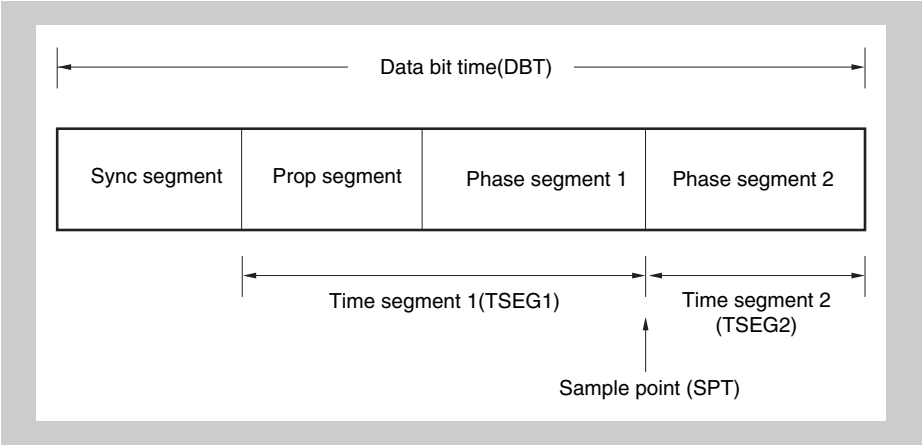


Figure 14-18 Segment setting

Table 14-15 Segment setting

Segment name	Settable range	Notes on setting to conform to CAN specification
Time segment 1 (TSEG1)	2TQ to 15TQ	-
Time segment 2 (TSEG2)	1TQ to 8TQ	IPT of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length less or equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
Resynchronization Jump Width (SJW)	1TQ to 4TQ	The length of time segment 1 minus 1TQ or 4 TQ, whichever is smaller.

Note 1. IPT: Information Processing Time

2. TQ: Time Quanta

Reference: The CAN protocol specification defines the segments constituting the data bit time as shown in *Figure 14-19*.

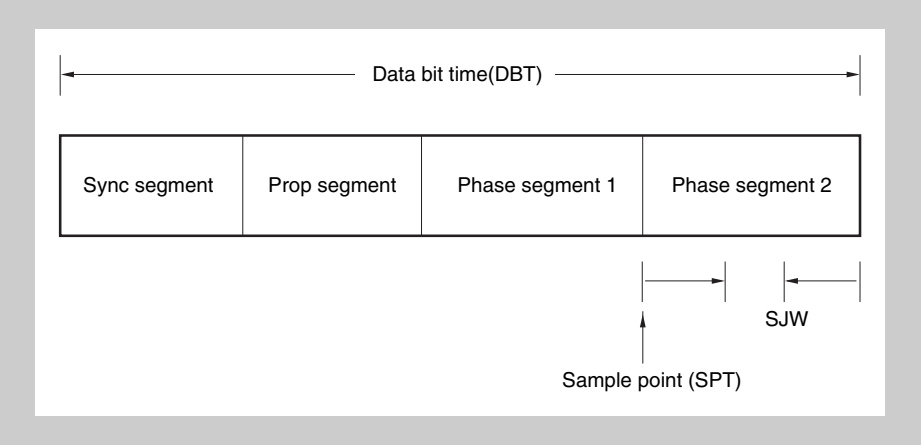


Figure 14-19 Configuration of data bit time defined by CAN specification

Table 14-16 Configuration of data bit time defined by CAN specification

Segment name	Settable range	Notes on setting to conform to CAN specification
Sync segment (Synchronization segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hardware synchronization is established.
Prop segment	Programmable to 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. 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.

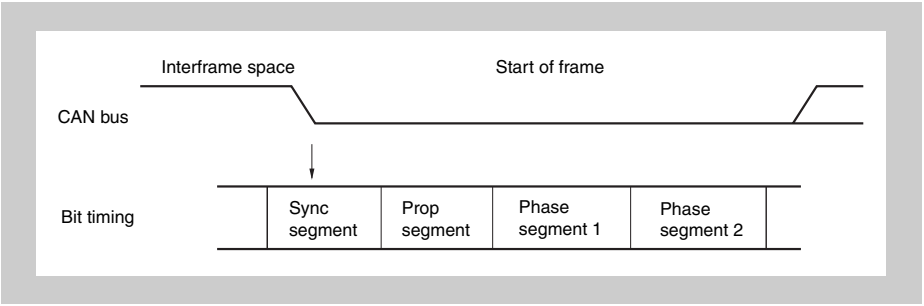


Figure 14-20 Adjusting synchronization of data bit

(b) Resynchronization

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

- The phase error of the edge is given by the relative position of the detected edge and sync segment.

  <Sign of phase error>

  0: If the edge is within the sync segment

  Positive: If the edge is before the sample point (phase error)

  Negative: If the edge is after the sample point (phase error)

  If phase error is positive: Phase segment 1 is lengthened by specified SJW.

  If phase error is negative: Phase segment 2 is shortened by specified SJW.

- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in the baud rate between the transmitting node and receiving node.

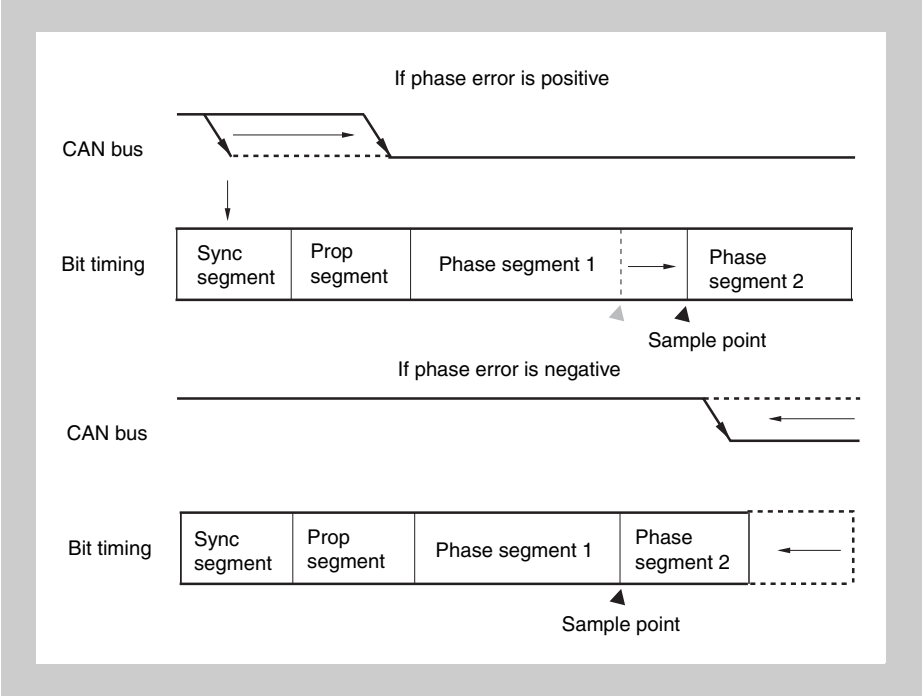


Figure 14-21 Resynchronization

14.4 Connection with Target System

The CAN module has to be connected to the CAN bus using an external transceiver.

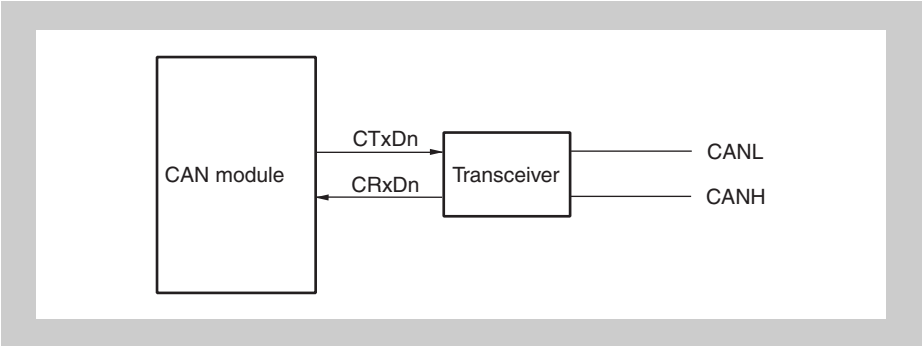


Figure 14-22 Connection to CAN bus

14.5 Internal Registers of CAN Controller

14.5.1 CAN module register and message buffer addresses

In this chapter all register and message buffer addresses are defined as address offsets to different base addresses.

Since all registers are accessed via the programmable peripheral area the bottom address is defined by the BPC register (refer to *"Programmable peripheral I/O area" on page 116* or to *"Programmable peripheral I/O area (PPA)" on page 230*).

The addresses given in the following tables are offsets to the programmable peripheral area base address PBA.

The recommended setting of BPC is $A100_H$. This setting would define the programmable peripheral area base address

PBA = $0840\ 0000_H$

Table 14-17 lists all base addresses used throughout this chapter.

Table 14-17 CAN module base addresses

Base address name	Base address of	Address	Address for BPC =$A100_H$
C0RBaseAddr	CAN0 registers	PBA + 000_H	$0840\ 0000_H$
C0MBaseAddr	CAN0 message buffers	PBA + 100_H	$0840\ 0100_H$
C1RBaseAddr	CAN1 registers	PBA + 600_H	$0840\ 0600_H$
C1MBaseAddr	CAN1 message buffers	PBA + 700_H	$0840\ 0700_H$

In the following <CnRBaseAddr> respectively <CnMBaseAddr> are used for the base address names for CAN channel n.

14.5.2 CAN Controller configuration

Table 14-18 List of CAN Controller registers

Item	Register Name
CANn global registers	CANn global control register (CnGMCTRL)
	CANn global clock selection register (CnGMCS)
	CANn global automatic block transmission control register (CnGMABT)
	CANn global automatic block transmission delay setting register (CnGMABTD)
CANn module registers	CANn module mask 1 register (CnMASK1L, CnMASK1H)
	CANn module mask 2 register (CnMASK2L, CnMASK2H)
	CANn module mask3 register (CnMASK3L, CnMASK3H)
	CANn module mask 4 registers (CnMASK4L, CnMASK4H)
	CANn module control register (CnCTRL)
	CANn module last error information register (CnLEC)
	CANn module information register (CnINFO)
	CANn module error counter register (CnERC)
	CANn module interrupt enable register (CnIE)
	CANn module interrupt status register (CnINTS)
	CANn module bit rate prescaler register (CnBRP)
	CANn module bit rate register (CnBTR)
	CANn module last in-pointer register (CnLIPT)
	CANn module receive history list register (CnRGPT)
	CANn module last out-pointer register (CnLOPT)
	CANn module transmit history list register (CnTGPT)
	CANn module time stamp register (CnTS)
CANn message buffer registers	CANn message data byte 01 register m (CnMDATA01m)
	CANn message data byte 0 register m (CnMDATA0m)
	CANn message data byte 1 register m (CnMDATA1m)
	CANn message data byte 23 register m (CnMDATA23m)
	CANn message data byte 2 register m (CnMDATA2m)
	CANn message data byte 3 register m (CnMDATA3m)
	CANn message data byte 45 register m (CnMDATA45m)
	CANn message data byte 4 register m (CnMDATA4m)
	CANn message data byte 5 register m (CnMDATA5m)
	CANn message data byte 67 register m (CnMDATA67m)
	CANn message data byte 6 register m (CnMDATA6m)
	CANn message data byte 7 register m (CnMDATA7m)
	CANn message data length register m (CnMDLCm)
	CANn message configuration register m (CnMCONFm)
	CANn message ID register m (CnMIDLm, CnMIDHm)
	CANn message control register m (CnMCTRLm)

14.5.3 CAN registers overview

(1) CANn global and module registers

The following table lists the address offsets to the CANn register base address: CnRBaseAddr.

Table 14-19 CANn global and module registers

Address offset	Register name	Symbol	R/W	Access			After reset
				1-bit	8-bit	16-bit	
000_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 14-20 CANn message buffer registers

Address offset	Register name	Symbol	R/W	Access			After reset
				1-bit	8-bit	16-bit	
$mx20_H + 0_H$	CANn message data byte 01 register m	CnMDATA01m	R/W	–	–	√	Undefined
$mx20_H + 0_H$	CANn message data byte 0 register m	CnMDATA0m		–	√	–	Undefined
$mx20_H + 1_H$	CANn message data byte 1 register m	CnMDATA1m		–	√	–	Undefined
$mx20_H + 2_H$	CANn message data byte 23 register m	CnMDATA23m		–	–	√	Undefined
$mx20_H + 2_H$	CANn message data byte 2 register m	CnMDATA2m		–	√	–	Undefined
$mx20_H + 3_H$	CANn message data byte 3 register m	CnMDATA3m		–	√	–	Undefined
$mx20_H + 4_H$	CANn message data byte 45 register m	CnMDATA45m		–	–	√	Undefined
$mx20_H + 4_H$	CANn message data byte 4 register m	CnMDATA4m		–	√	–	Undefined
$mx20_H + 5_H$	CANn message data byte 5 register m	CnMDATA5m		–	√	–	Undefined
$mx20_H + 6_H$	CANn message data byte 67 register m	CnMDATA67m		–	–	√	Undefined
$mx20_H + 6_H$	CANn message data byte 6 register m	CnMDATA6m		–	√	–	Undefined
$mx20_H + 7_H$	CANn message data byte 7 register m	CnMDATA7m		–	√	–	Undefined
$mx20_H + 8_H$	CANn message data length register m	CnMDLCm		–	√	–	0000 $xxxx_B$
$mx20_H + 9_H$	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

14.5.4 Register bit configuration

Table 14-21 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 14-22 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 14-22 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 14-23 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 456*.

14.6 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

- CANn global control register (CnGMCTRL)
- CANn global automatic block transmission control register (CnGMABT)
- CANn module control register (CnCTRL)
- CANn module interrupt enable register (CnIE)
- CANn module interrupt status register (CnINTS)
- CANn module receive history list register (CnRGPT)
- CANn module transmit history list register (CnTGPT)
- CANn module time stamp register (CnTS)
- CANn message control register (CnMCTRLm)

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in *Figure 14-23* below to set or clear the lower 8 bits in these registers.

Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (refer to the bit status after set/clear operation is specified in *Figure 14-26*). *Figure 14-23* shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

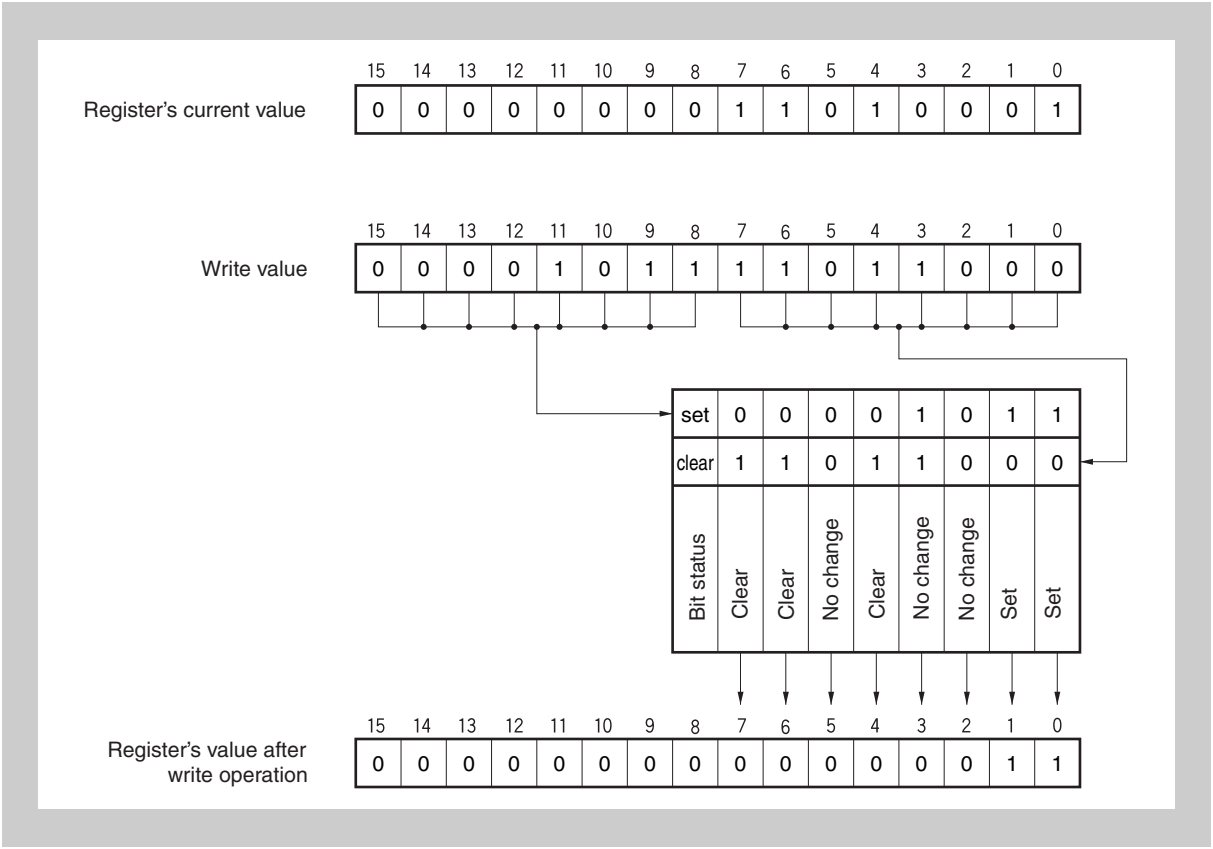


Figure 14-23 Example of bit setting/clearing operations

(1) Bit status after bit setting/clearing operations

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

Set 0 ... 7	Clear 0 ... 7	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

14.7 Control Registers

(1) CnGMCTRL - CANn global control register

The CnGMCTRL register is used to control the operation of the CAN module.

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

Address <CnRBaseAddr> + 000_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 by GOM bit = 0 disabled.
1	Forced shut down by GOM bit = 0 enabled.

Caution 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 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.

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)

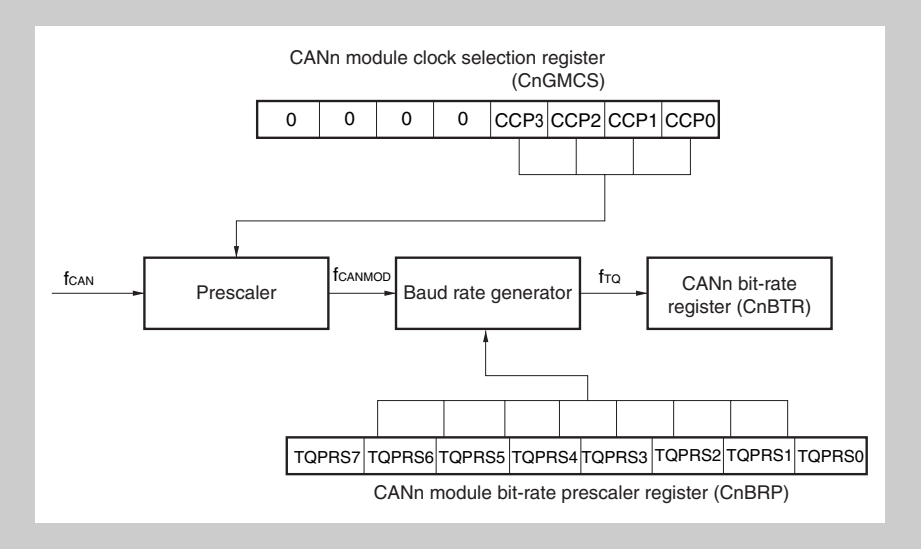


Figure 14-24 CAN module clock

Note f_{CAN}: clock supplied to CAN
f_{CANMOD}: CAN module system clock
f_{TQ}: CAN protocol layer basic system clock

Caution The CnBRP register can be write-accessed only in the initialization mode.

(13) CnBTR - CANn module bit rate register

The CnBTR register is used to control the data bit time of the communication baud rate.

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

Address <CnRBaseAddr> + $05C_H$

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

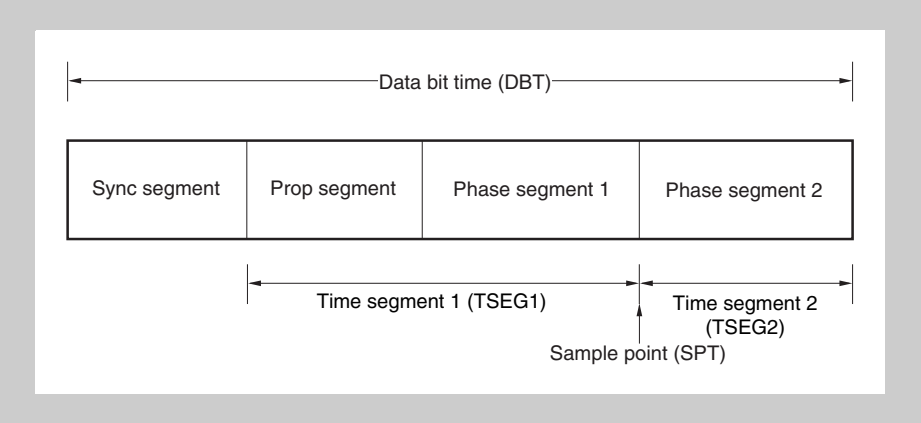


Figure 14-25 Data bit time

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 456*.

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 456.*

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 456.*

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 456.*

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 456.*

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.

14.8 CAN Controller Initialization

14.8.1 Initialization of CAN module

Before CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP[3:0] bits of the CnGMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.

The CAN module is enabled by setting the GOM bit of the CnGMCTRL register.

For the procedure of initializing the CAN module, refer to *"Operation of CAN Controller" on page 538*.

14.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of all CnMCTRLm registers to 0.
- Clear the MA0 bit of all CnMCONFm registers to 0.

14.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module to an operation mode.

(2) To redefine message buffer during reception

Perform redefinition as shown in *Figure 14-38*.

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (see *"Transmission abort process except for in normal operation mode with automatic block transmission (ABT)" on page 517* and *"Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)" on page 517*). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining

the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

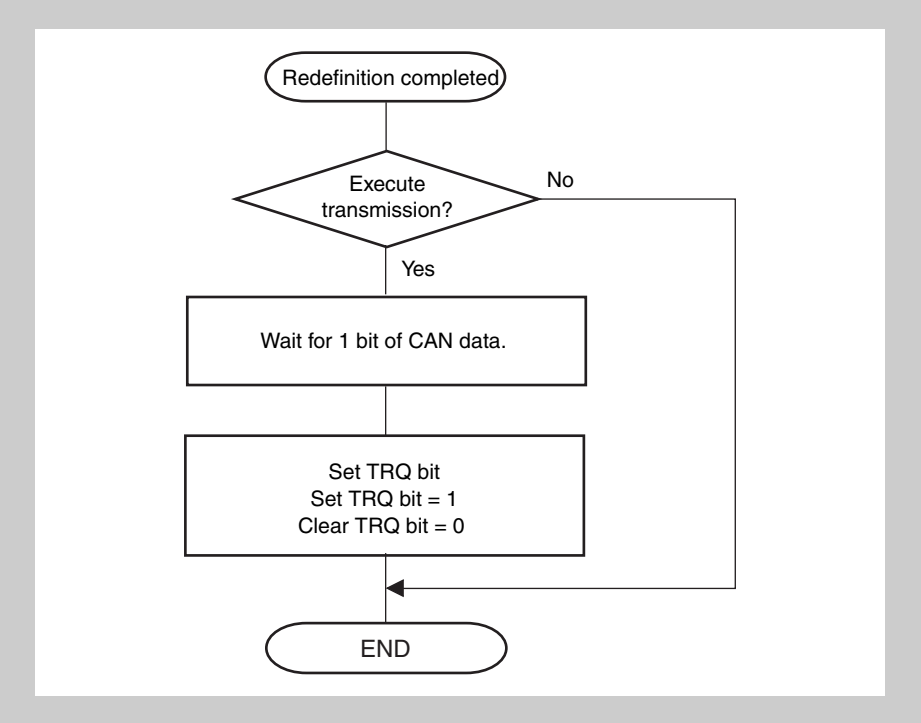


Figure 14-26 Setting transmission request (TRQ) to transmit message buffer after redefinition

Caution

1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in *Figure 14-38 on page 541* is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in *Figure 14-26 on page 500* is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

14.8.4 Transition from initialization mode to operation mode

The CAN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

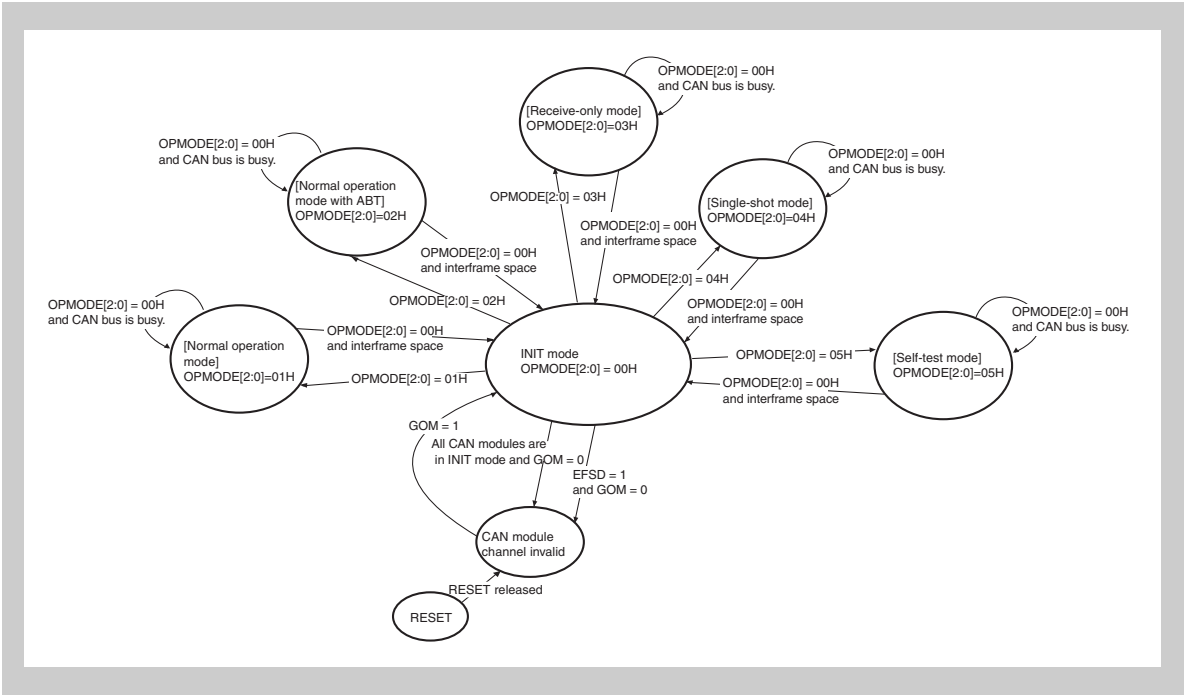


Figure 14-27 Transition to operation modes

The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE[2:0] in the CnCTRL register.

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from an operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the values of the OPMODE[2:0] bits are changed to $000_{B)}$. 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 14-36 on page 539*).

14.8.5 Resetting error counter CnERC of CAN module

If it is necessary to reset the CAN module error counter CnERC and CAN module information register CnINFO when re-initialization or forced recovery from the bus-off status is made, set the CCERC bit of the CnCTRL register to 1 in the initialization mode. When this bit is set to 1, the CnERC and CnINFO registers are cleared to their default values.

14.9 Message Reception

14.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1.)
- Set as a receive message buffer
  (MT[2:0] bits of CnMCONFm register are set to 001_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 14-24 MBRB priorities

Priority	Storing condition if same ID is set	
1 (high)	Unmasked message buffer	DN bit = 0
		DN bit = 1 and OWS bit = 1
2	Message buffer linked to mask 1	DN bit = 0
		DN bit = 1 and OWS bit = 1
3	Message buffer linked to mask 2	DN bit = 0
		DN bit = 1 and OWS bit = 1
4	Message buffer linked to mask 3	DN bit = 0
		DN bit = 1 and OWS bit = 1
5 (low)	Message buffer linked to mask 4	DN bit = 0
		DN bit = 1 and OWS bit = 1

14.9.2 Receive data read

To keep data consistency when reading CAN message buffers, perform the data reading according to *Figure 14-49 on page 552* to *Figure 14-52 on page 556*.

During message reception, the CAN module sets DN of the CnMCTRLm register two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, the MUC bit of the CnMCTRLm register of the message buffer is set. (Refer to *Figure 14-28 on page 504*.)

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.

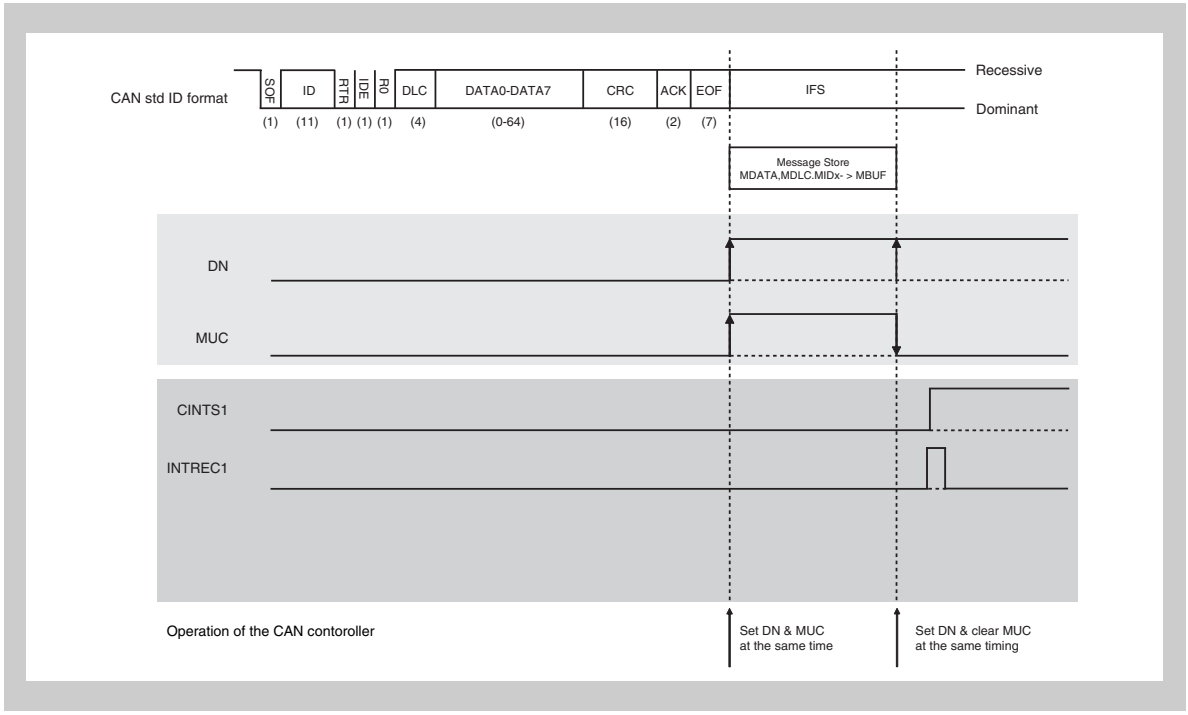


Figure 14-28 DN and MUC bit setting period (for standard ID format)

14.9.3 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding CnLIPT register and the receive history list get pointer (RGPT) with the corresponding CnRGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CnLIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the CnLIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the CnRGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the CnRGPT register, the RGPT pointer is automatically incremented.

If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the CnRGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the CnRGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the newly received message. In this case, after the ROVF bit has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of the DN-bit.

Caution If the history list is in the overflow condition (ROVF is set), reading the history list contents is still possible, until the history list is empty (indicated by RHPM flag set). Nevertheless, the history list remains in the overflow condition, until ROVF is cleared by software. If ROVF is not cleared, the RHPM flag will also not be updated (cleared) upon a message storage of newly received frame. This may lead to the situation, that RHPM indicates an empty history list, although a reception has taken place, while the history list is in the overflow state (ROVF and RHPM are set).

As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.

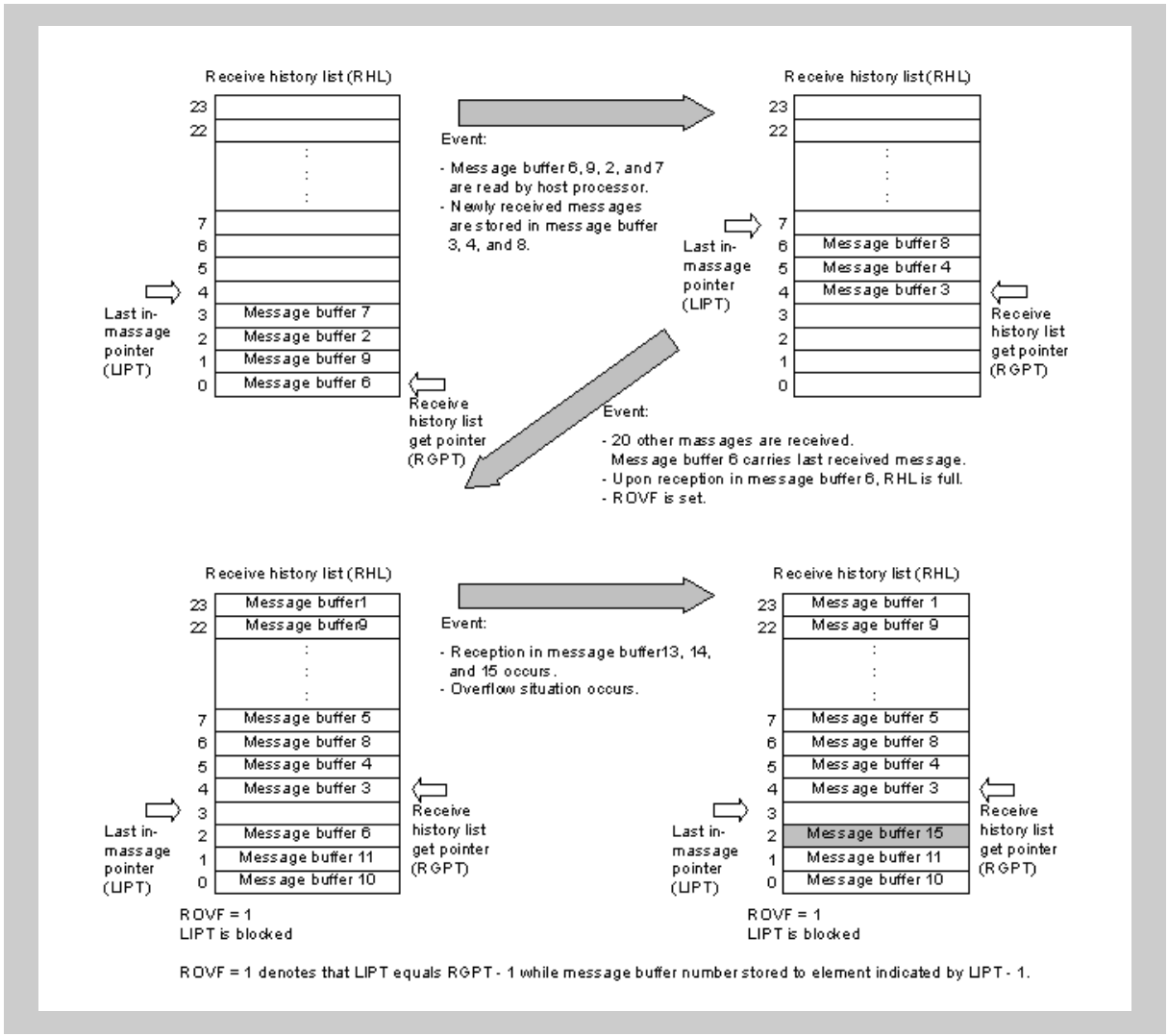


Figure 14-29 Receive history list

14.9.4 Mask function

For any message buffer, which is used for reception, the assignment to one of four global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is in effect, an identifier bit that is defined to be 1 by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as 0 by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are 0 and bits ID24 and ID22 are 1, are to be stored in message buffer 14. The procedure for this example is shown below.

1. Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

2. Identifier to be configured in message buffer 14 (example)
(Using CnMIDL14 and CnMIDH14 registers)

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

Note 1. ID with the ID27 to ID25 bits cleared to 0 and the ID24 and ID22 bits set to 1 is registered (initialized) to message buffer 14.

2. Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT[2:0] of CnMCONF14 register are set to 010_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.

14.9.5 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type. These message buffers can be allocated anywhere in the message buffer memory, they do not even have to follow each other adjacently.

Suppose, for example, the same message buffer type is set to 10 message buffers, message buffers 10 to 19, and the same ID is set to each message buffer. If the first message whose ID matches an ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

When the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and so on. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting the IE bit of the CnMCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k-3) and setting the IE bit of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Caution

1. MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.
2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.
3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.
4. With MBRB, “matching ID” means “matching ID after mask”. Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.
5. The priority between MBRBs is mentioned in the table *Table 14-24*.

14.9.6 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1.)
- Set as a transmit message buffer
  (MT[2:0] bits in CnMCONFm register set to 000_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.

14.10 Message Transmission

14.10.1 Message transmission

A message buffer with its TRQ bit set to 1 participates in the search for the most high-prioritized message when the following conditions are fulfilled. This behavior is valid for all operational modes.

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1.)
- Set as a transmit message buffer
  (MT[2:0] bits of CnMCONFm register set to 000_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).

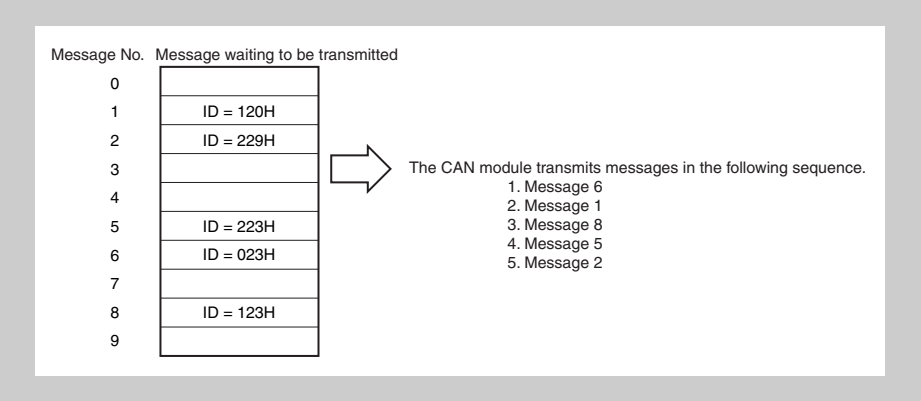


Figure 14-30 Message processing example

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can perform a transmission abort request for the lower priority message. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1 (high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than a message frame with a 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If one or more transmission-pending extended ID message frame has equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5 (low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Note 1. If the automatic block transmission request bit ABTTRG is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.

If the ABT mode was triggered by ABTTRG bit (1), one TRQ bit is set to 1 in the ABT area (buffer 0 through 7). Beyond this TRQ bit, the application can request transmissions (set TRQ bit to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with TRQ bit set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit CINTS0 of the CnINTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTCnTRX is output (if the CIE0 bit of the CnIE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).

2. When changing the contents of a transmit buffer, the RDY flag of this buffer must be cleared before updating the buffer contents. As during internal transfer actions, the RDY flag may be locked temporarily, the status of RDY must be checked by software, after changing it.

14.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been 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).

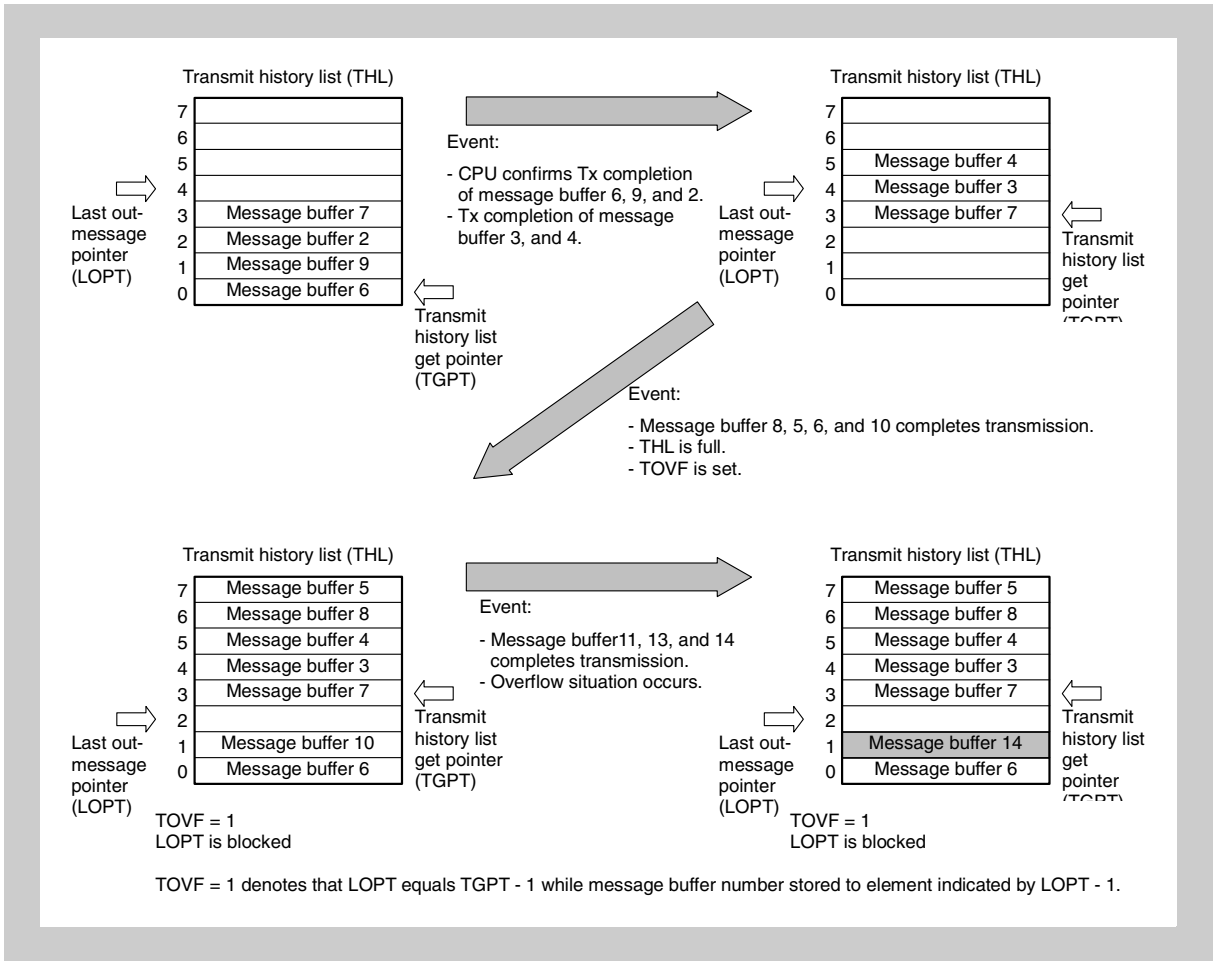


Figure 14-31 Transmit history list

14.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).

By setting the OPMODE[2:0] bits of the CnCTRL register to 010_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.

14.10.4 Transmission abort process

(1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)

The user can clear the TRQ bit of the CnMCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in *Figure 14-45 on page 548*).

(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)

The user can clear the ABTTRG bit of the CnGMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the CnGMABT register = 0, clear the TRQ bit of the CnMCTRLm register to 0. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in *Figure 14-46 on page 549*).

(3) Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear the ABTTRG bit of the CnGMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, refer to the process in *Figure 14-47 on page 550*). If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, refer to the process in *Figure 14-48 on page 551*).

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.

14.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the CnMCONFm register. Setting (1) the RTR bit sets remote frame transmission.

14.11 Power Saving Modes

14.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN Controller to stand-by mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.

In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01_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.

14.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN Controller to stand-by mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.

The CAN stop mode can only be released (entering CAN sleep mode) by writing 01_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.

14.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example for using the power saving modes.

- First, put the CAN module in the CAN sleep mode (PSMODE[1:0] = 01_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).

14.12 Interrupt Function

The CAN module provides 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 14-25 List of CAN module interrupt sources

No.	Interrupt status bit		Interrupt enable bit		Interrupt request signal	Interrupt source description
	Name	Register	Name	Register		
1	CINTS0	CnINTS	CIE0[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).

14.13 Diagnosis Functions and Special Operational Modes

The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of special CAN communication methods.

14.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until “valid reception” is detected, so that the baud rates in the module match (“valid reception” means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the CnCTRL register (1).

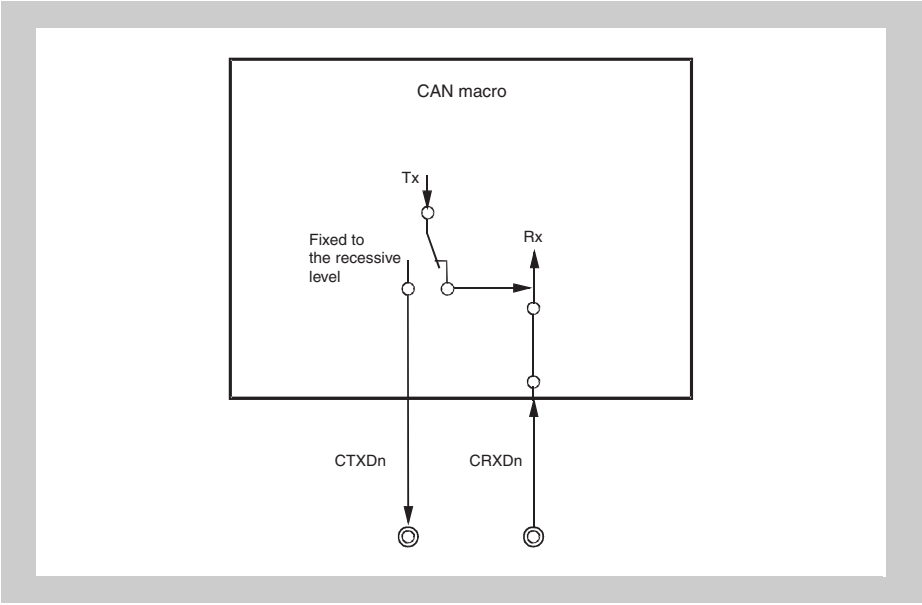


Figure 14-32 CAN module terminal connection in receive-only mode

In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the CAN transmission pin (CTXDn) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter the CnERC.TEC7 to CnERC.TEC0 bits are never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.

Furthermore, in the receive-only mode ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

14.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other behavior of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the CnCTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the following events:

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the CnINTS register respectively, and the type of the error can be identified by reading the LEC[2:0] bits of the CnLEC register.

Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the CnINTS register is set to 1. If the CIE0 bit of the CnIE register is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g., TTCAN level 1).

Caution The AL bit is only valid in single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

14.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTXDn) is fixed to the recessive level.

If the falling edge on the CAN reception pin (CRXDn) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes. To keep the module in the CAN sleep mode, use the CAN reception pin (CRXDn) as a port pin.

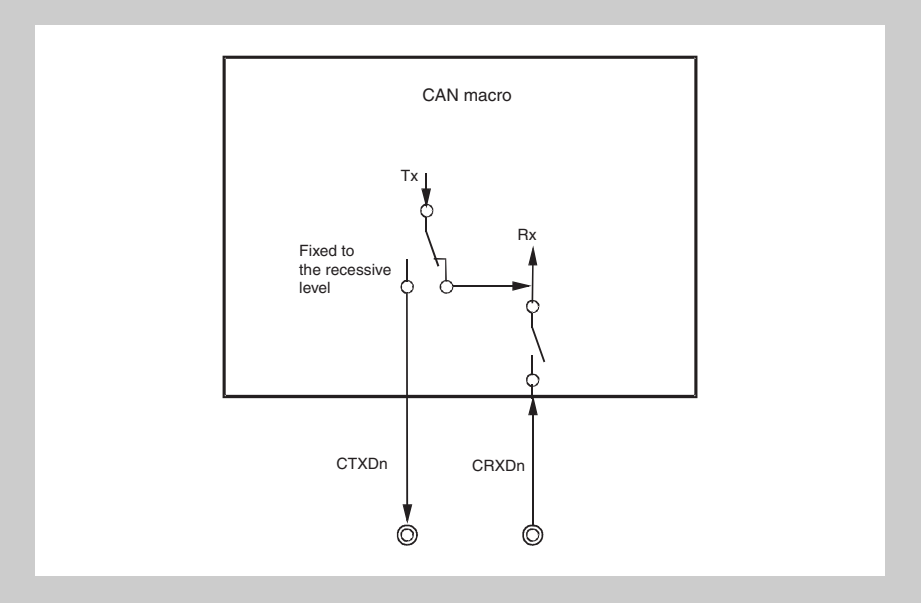


Figure 14-33 CAN module terminal connection in self-test mode

14.13.4 Receive/transmit operation in each operation mode

The following table shows outline of the receive/transmit operation in each operation mode.

Table 14-26 Outline of the receive/transmit in each operation mode

Operation mode	Transmission of data/remote frame	Transmission of ACK	Transmission of error/over load frame	Transmission retry	Automatic block transmission (ABT)	Set of VALID bit	Store data to message buffer
Initialization mode	No	No	No	No	No	No	No
Normal operation mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal operation mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive only mode	No	No	No	No	No	Yes	Yes
Single-shot mode	Yes	Yes	Yes	No[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.

14.14 Time Stamp Function

CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

14.14.1 Time stamp function

The CAN Controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN Controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN Controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. The TSOUT signal can be selected from the following two event sources and is specified by the TSSEL bit of the CnTS register.

- SOF event (start of frame) (TSSEL = 0)
- EOF event (last bit of end of frame) (TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the CnTS register to 1.

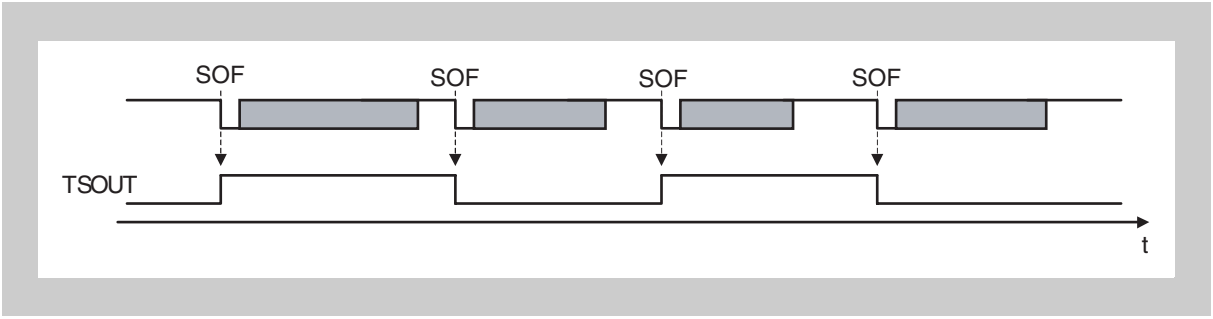


Figure 14-34 Timing diagram of capture signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in *Figure 14-34*, the SOF is used as the trigger event source). To capture a timer value by using the TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

This time stamp function is controlled by the TSLOCK bit of the CnTS register. When TSLOCK is cleared to 0, the TSOUT signal toggles upon occurrence of the selected event. If TSLOCK is set to 1, the TSOUT signal toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by the TSOUT signal, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution The time stamp function using the TSLOCK bit stops toggle of the TSOUT signal by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of the TSOUT signal cannot be stopped by reception of a remote frame. Toggle of the TSOUT signal does not stop when a data frame is received in a message buffer other than message buffer 0.
For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of the TSOUT signal by the TSLOCK bit cannot be used.

14.15 Baud Rate Settings

14.15.1 Baud rate setting conditions

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN Controller, as follows.

- 5TQ ≤ 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 14-27 shows the combinations of bit rates that satisfy the above conditions.

Table 14-27 Settable bit rate combinations (1/3)

Valid bit rate setting					CnBTR register setting value		Sampling point (unit %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8

Table 14-27 Settable bit rate combinations (2/3)

Valid bit rate setting					CnBTR register setting value		Sampling point (unit %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0

Table 14-27 Settable bit rate combinations (3/3)

Valid bit rate setting					CnBTR register setting value		Sampling point (unit %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7[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 14-27* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

14.15.2 Representative examples of baud rate settings

Table 14-28 and *Table 14-29* show representative examples of baud rate settings.

Table 14-28 Representative examples of baud rate settings (f_{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 14-28 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 14-28* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 14-29 Representative examples of baud rate settings (f_{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 14-29 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 14-29* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

14.16 Operation of CAN Controller

The processing procedure for showing in this chapter is recommended processing procedure to operate CAN controller.

Develop the program referring to recommended processing procedure in this chapter.

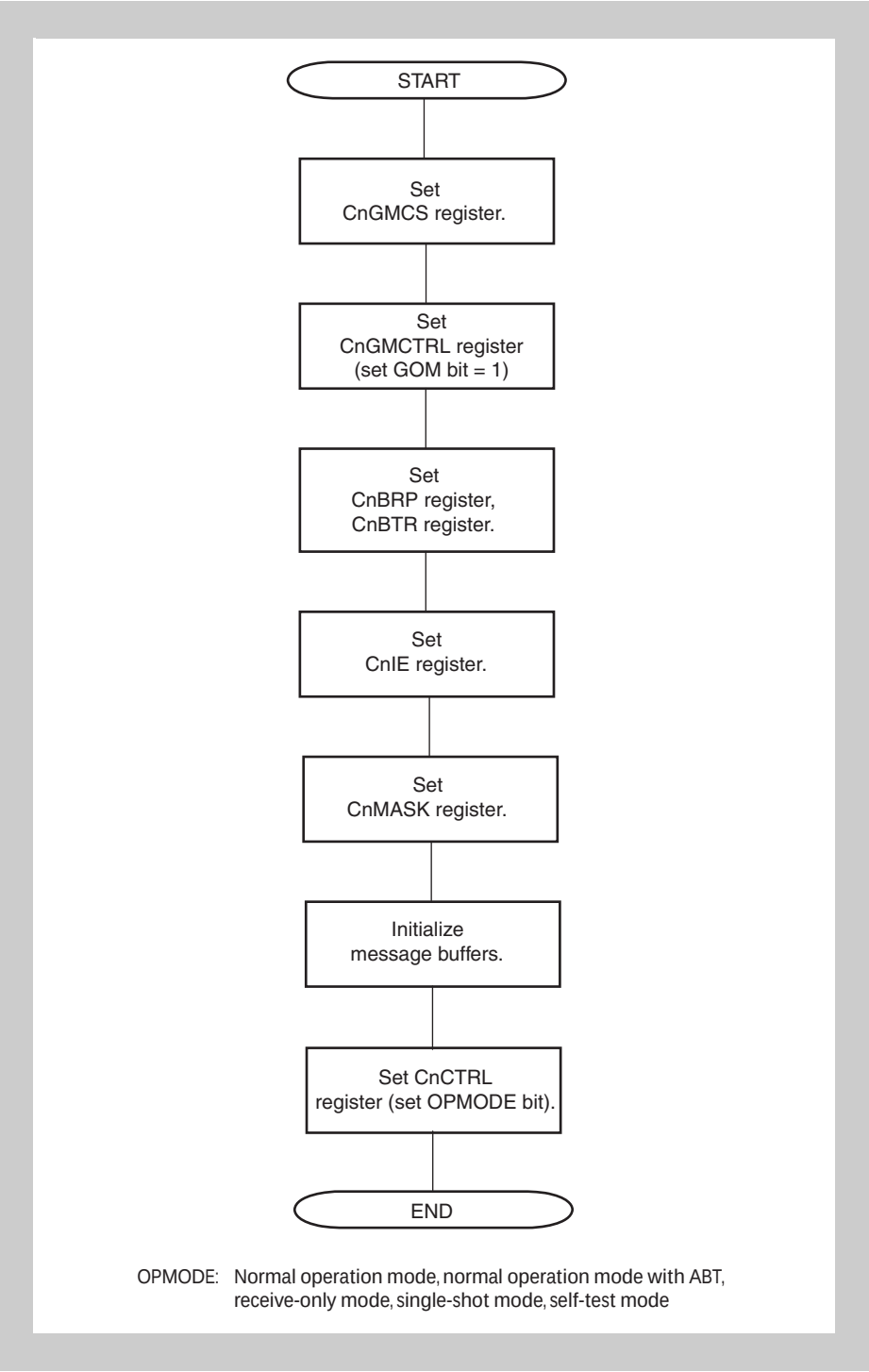


Figure 14-35 Initialization

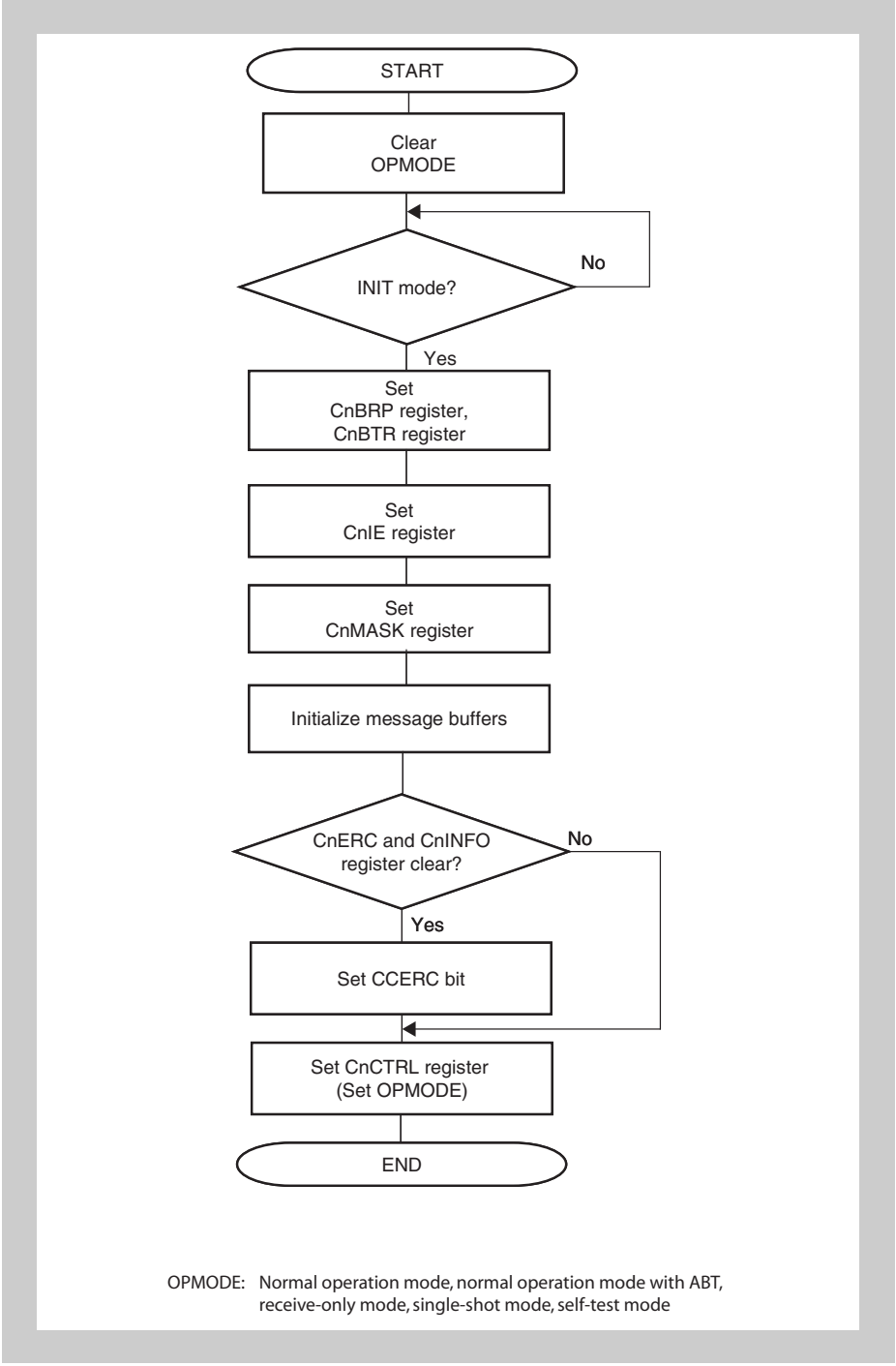


Figure 14-36 Re-initialization

Caution After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the CnCTRL and CnGMCTRL registers (e.g., set a message buffer).

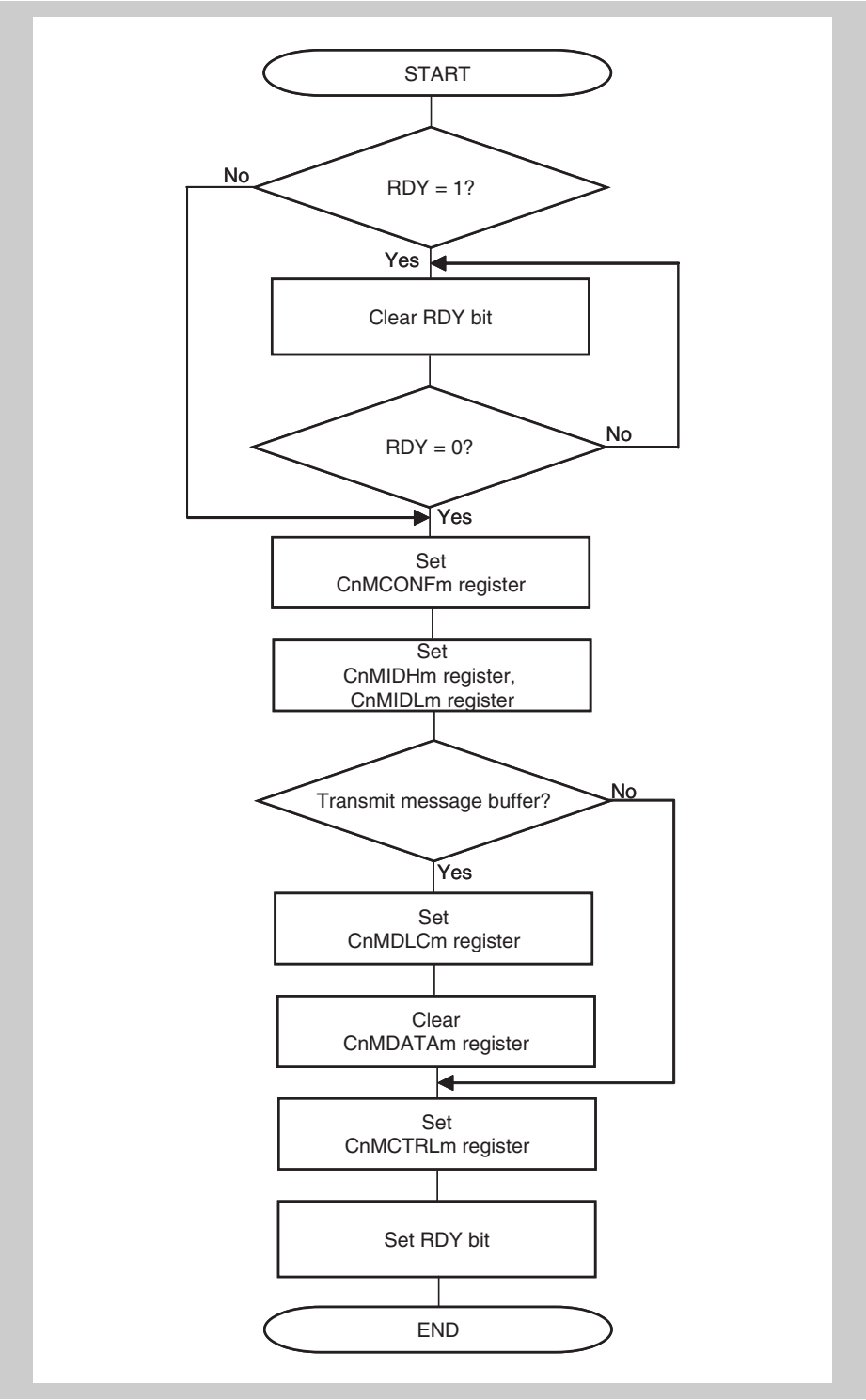
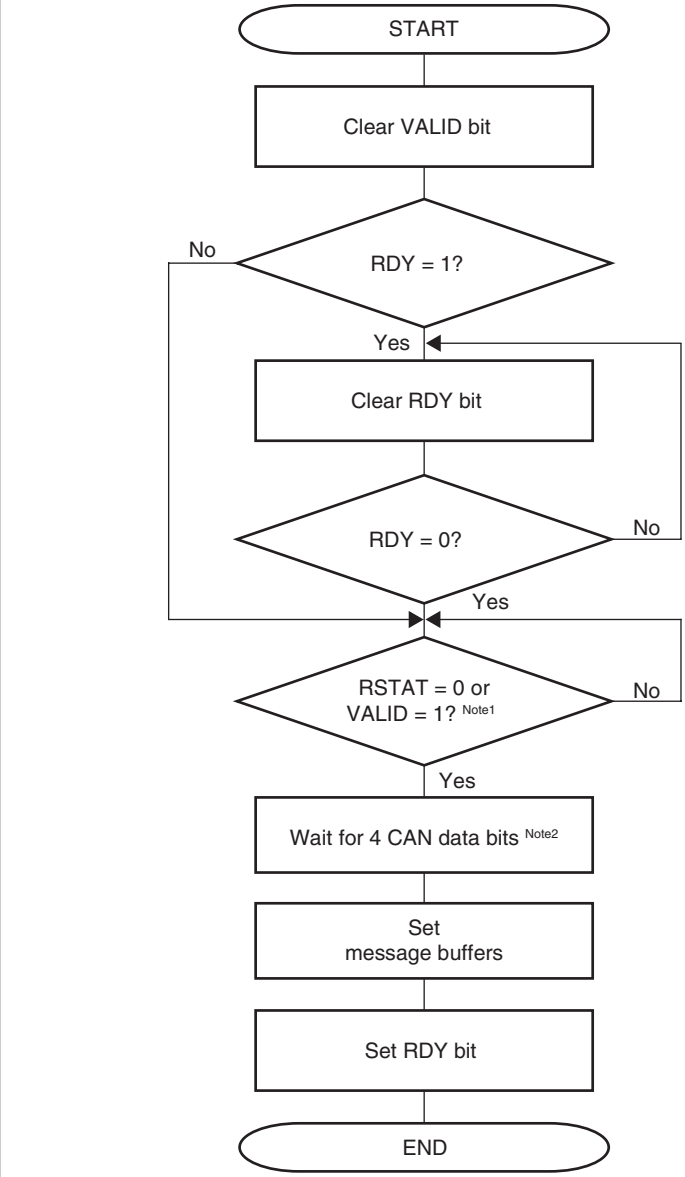


Figure 14-37 Message buffer initialization

Caution

1. Before a message buffer is initialized, the RDY bit must be cleared.
2. Make the following settings for message buffers not used by the application.
   - Clear the RDY, TRQ, and DN bits of the CnMCTRLm register to 0.
   - Clear the MA0 bit of the CnMCONFm register to 0.

Figure 14-38 shows the processing for a receive message buffer (MT[2:0] bits of CnMCONFm register = 001_B to 101_B).



Note1: Confirm that a message is being received because RDY bit must be set after a message is completely received.

Note2: Avoid message buffer redefinition during store operation of message reception by waiting additional 4 CAN data bits.

Figure 14-38 Message buffer redefinition

Figure 14-39 shows the processing for a transmit message buffer during transmission (MT[2:0] bits of CnMCONFm register = 000_B).

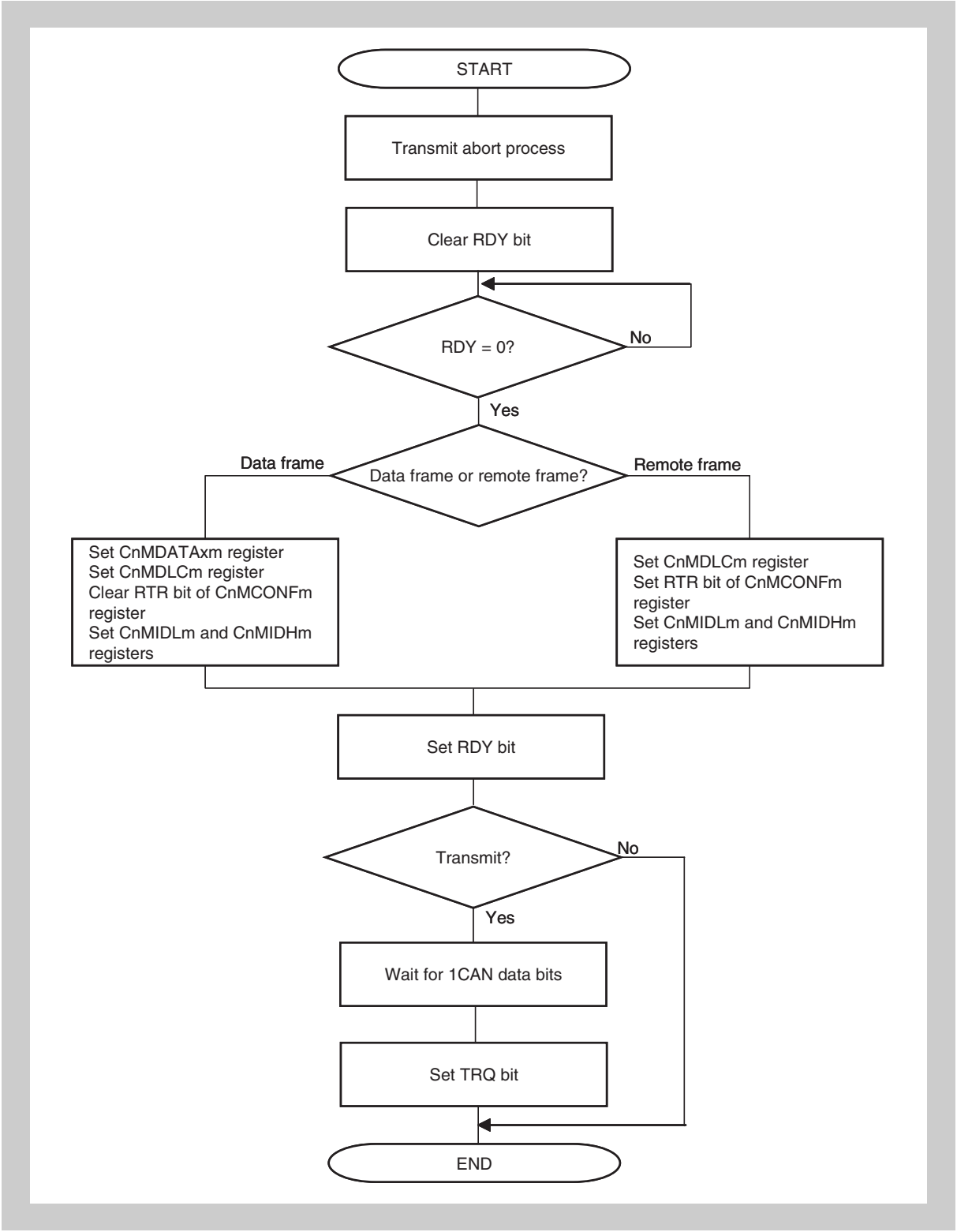


Figure 14-39 Message buffer redefinition during transmission

Figure 14-40 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000_B).

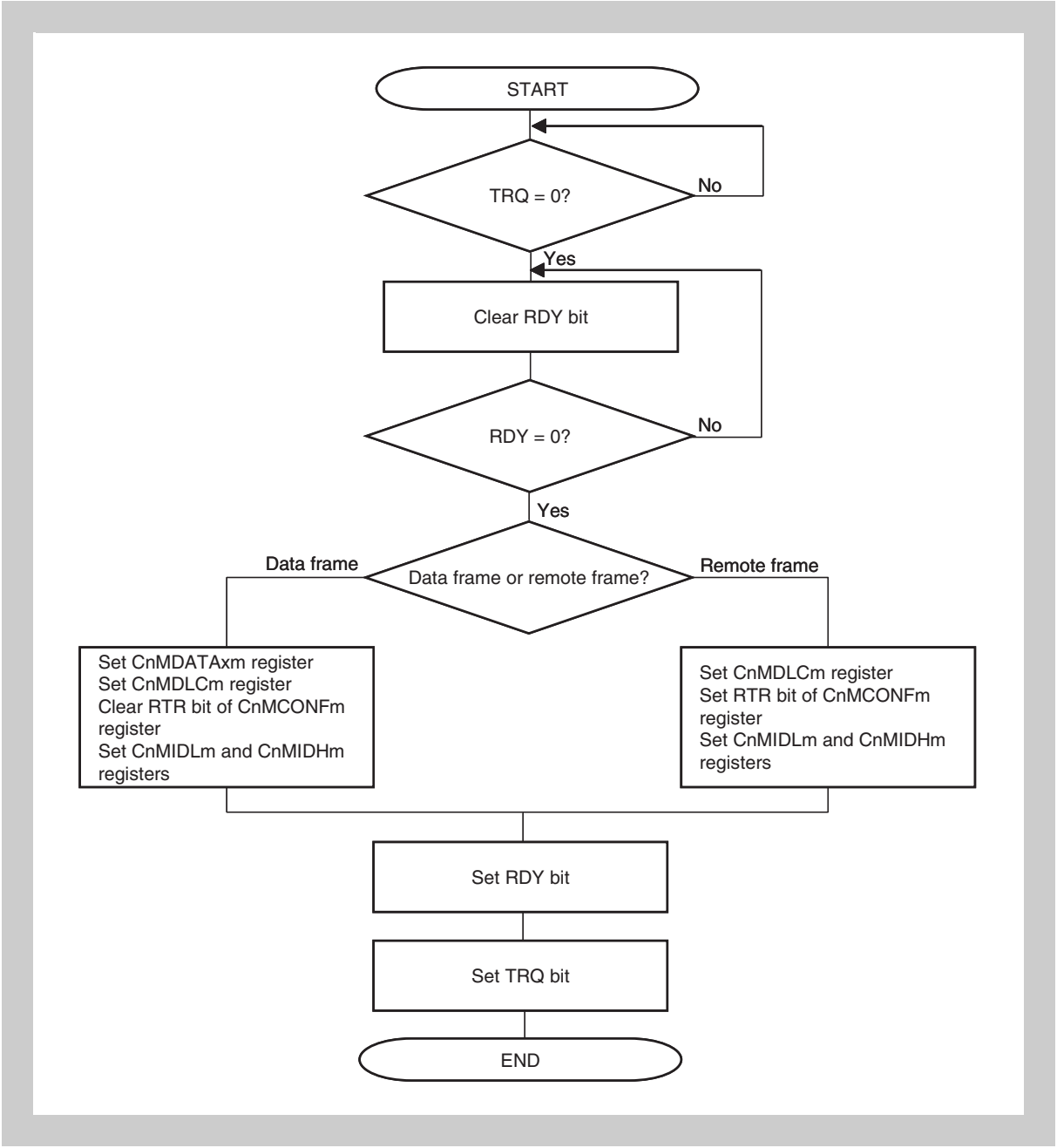


Figure 14-40 Message transmit processing

Caution
1. The TRQ bit should be set after the RDY bit is set.
2. The RDY bit and TRQ bit should not be set at the same time.

Figure 14-41 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000_B)

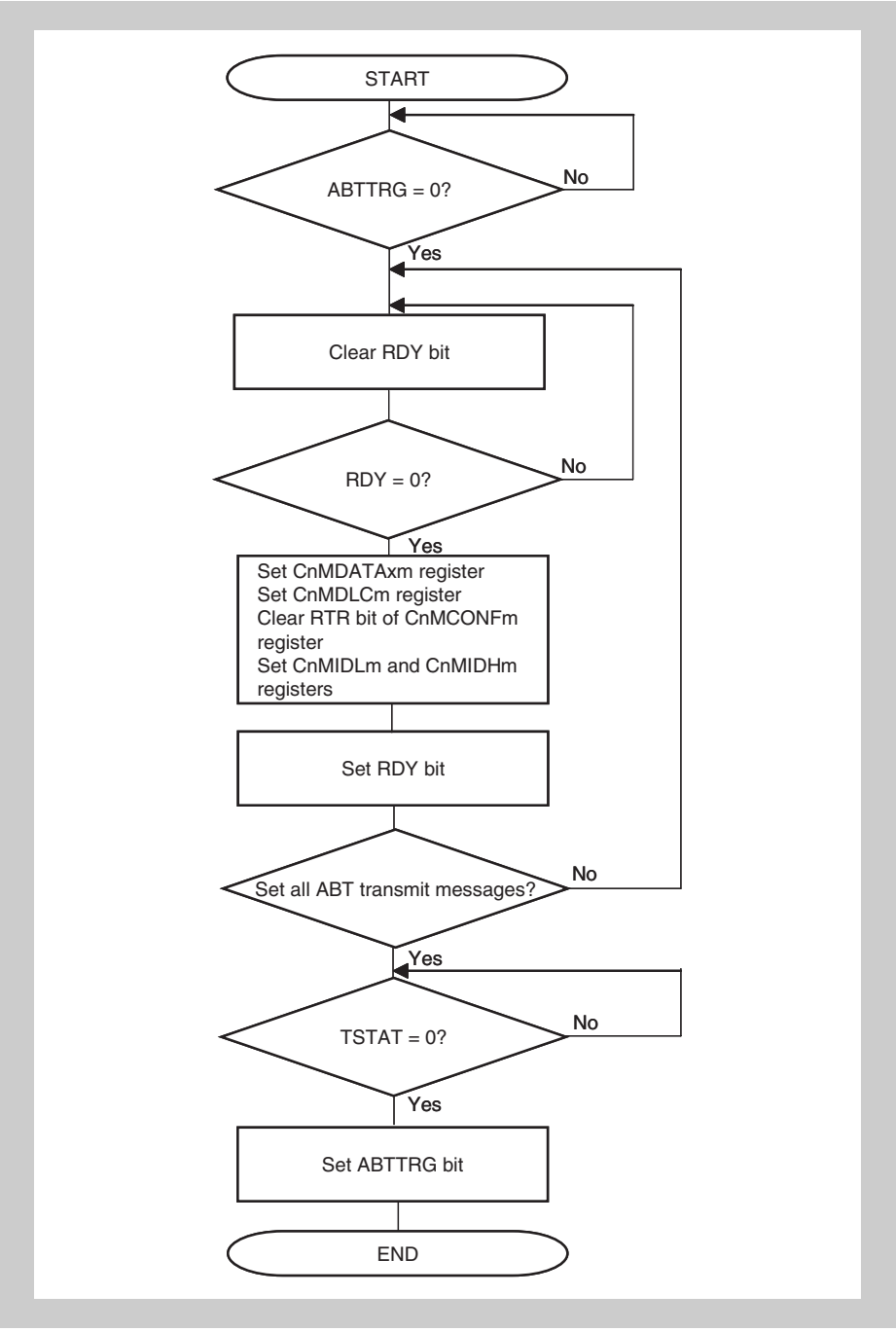


Figure 14-41 ABT message transmit processing

Note This processing (normal operation mode with ABT) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, see *Figure 14-40 on page 543*.

Caution The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. Checking the TSTAT bit and setting the ABTTRG bit to 1 must be processed consecutively.

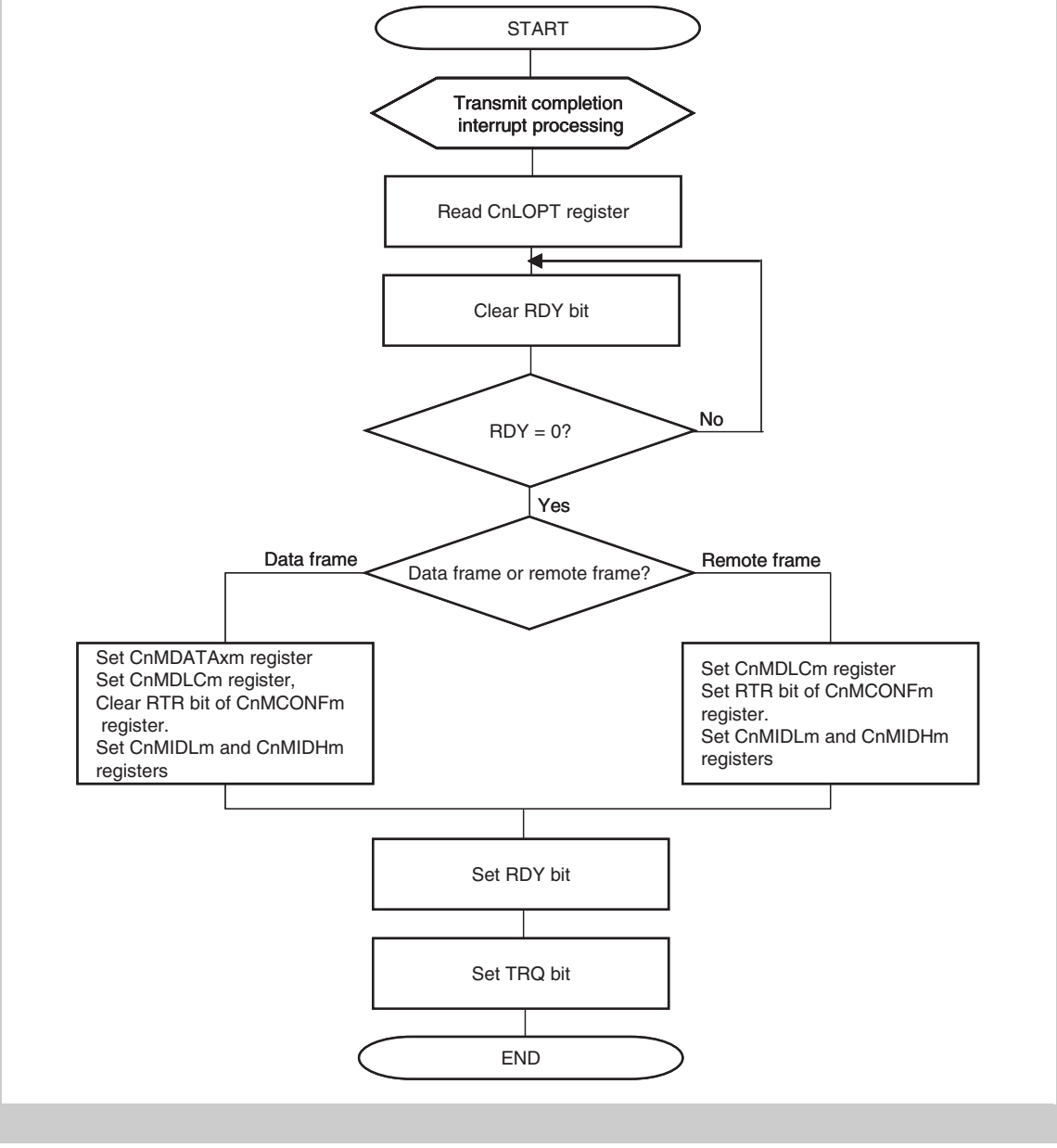


Figure 14-42 Transmission via interrupt (using CnLOPT register)

Caution 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Note Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

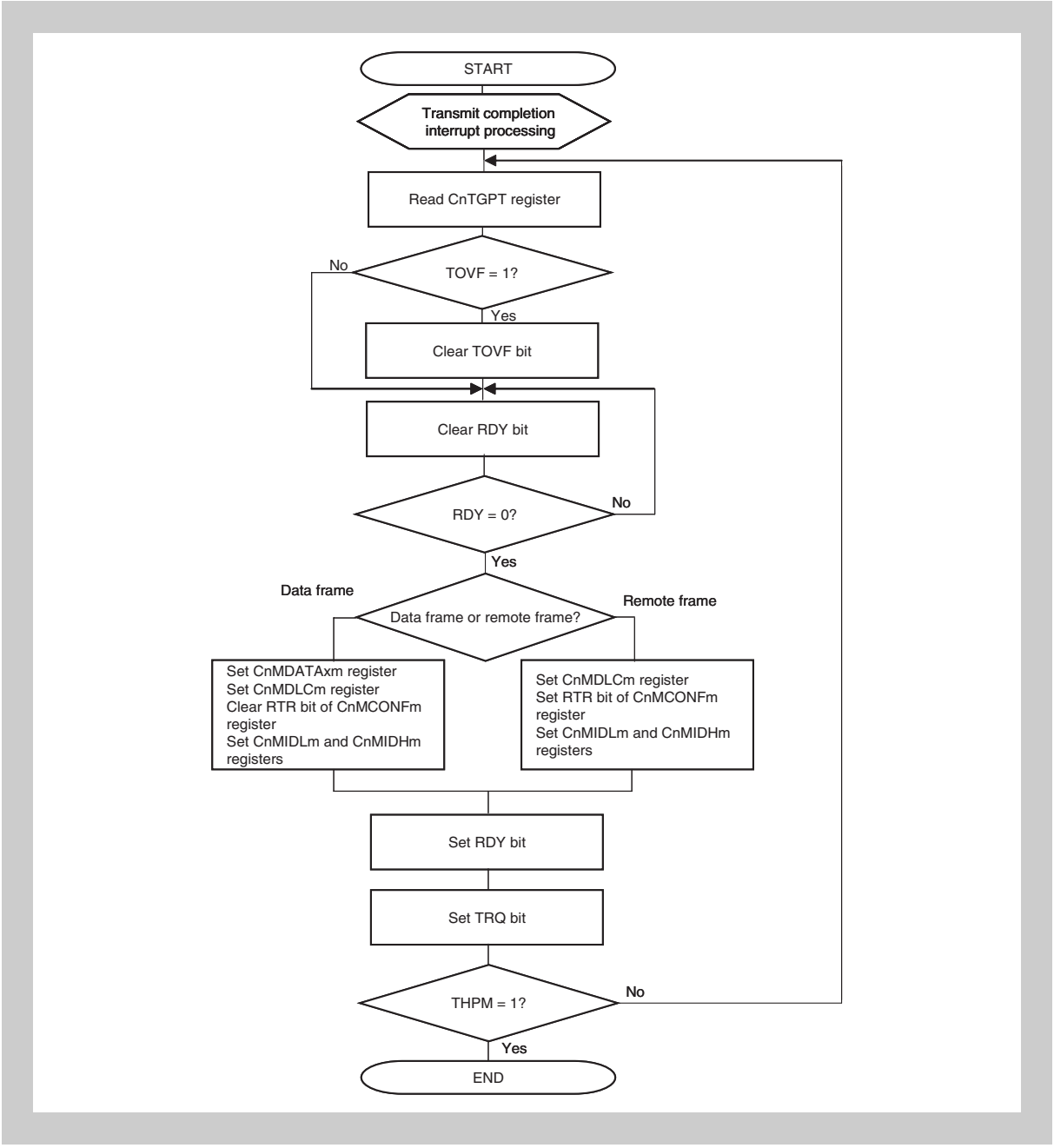


Figure 14-43 Transmission via interrupt (using CnTGPT register)

Caution 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Note 1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of

the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

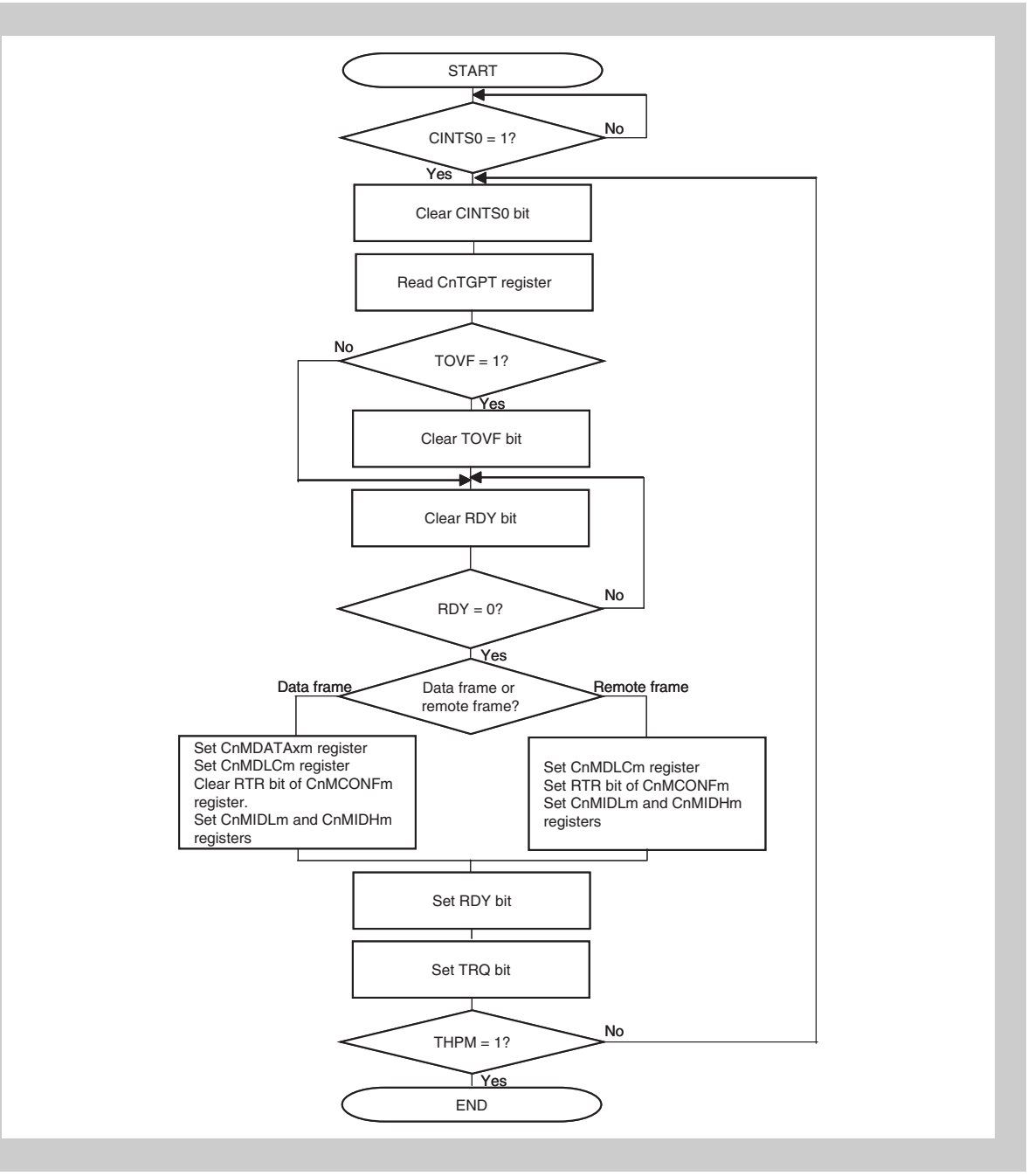


Figure 14-44 Transmission via software polling

Caution

1. The TRQ bit should be set after the RDY bit is set.
2. The RDY bit and TRQ bit should not be set at the same time.

Note 1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

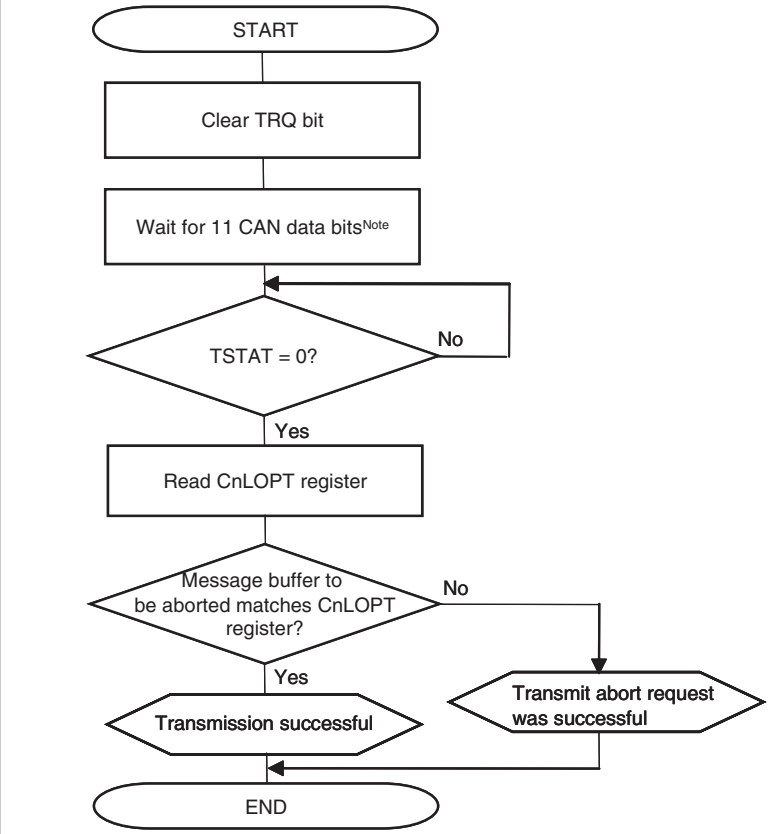


Figure 14-45 Transmission abort processing (except normal operation mode with ABT)

Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

Caution 1. Clear the TRQ bit for aborting transmission request, not the RDY bit.

2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute any new transmission request including in the other message buffers while transmission abort processing is in progress.

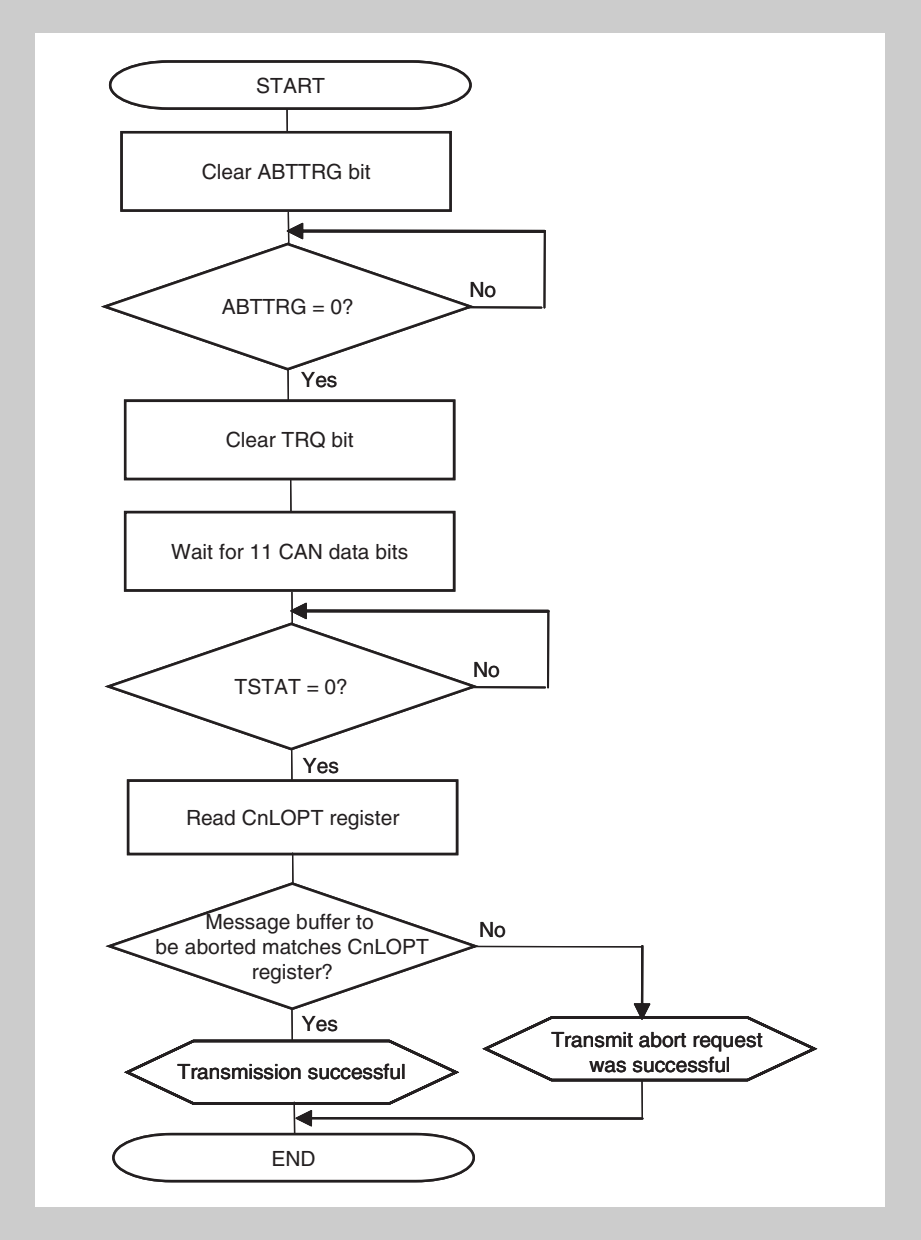


Figure 14-46 Transmission abort processing except for ABT transmission (normal operation mode with ABT)

Caution

1. Clear the TRQ bit for aborting transmission request, not the RDY bit.
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.
3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.
4. Do not execute any new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 14-47 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

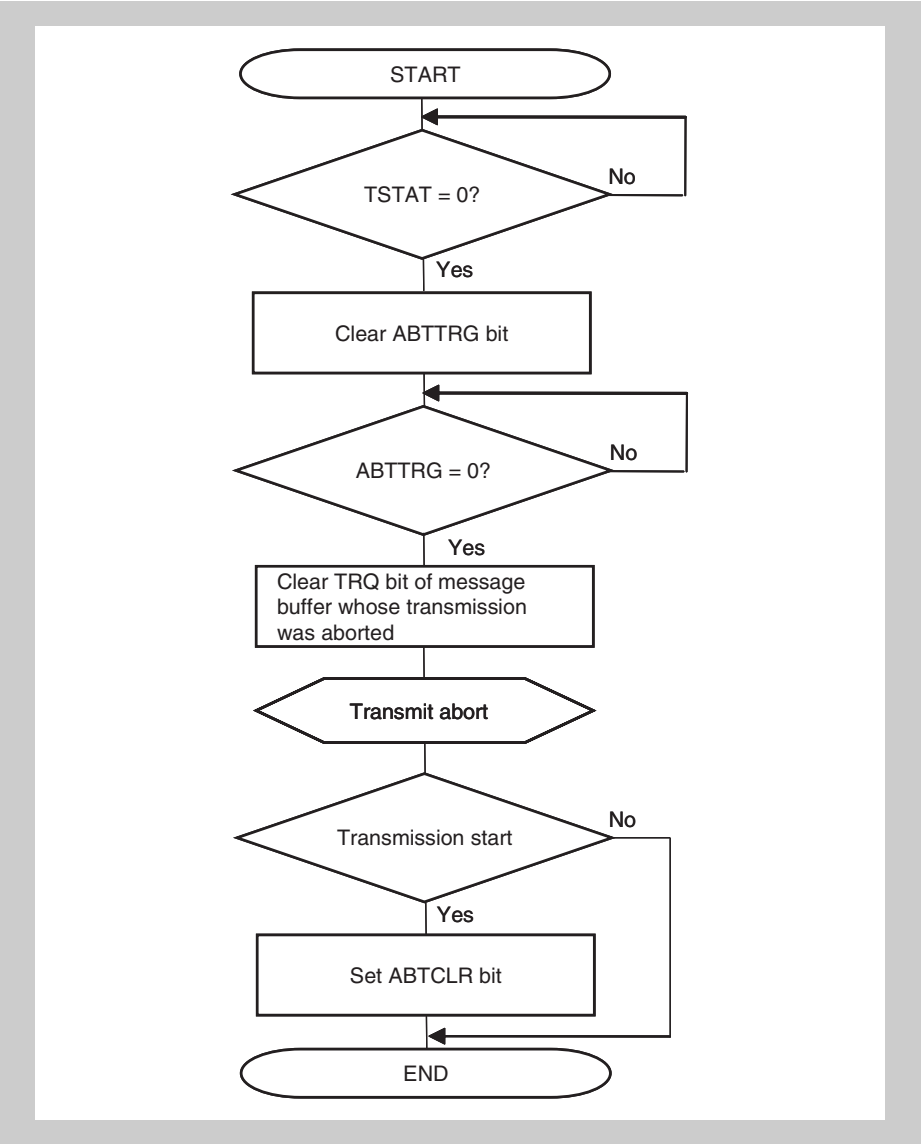


Figure 14-47 Transmission abort processing (normal operation mode with ABT)

Caution

1. Do not set any transmission requests while ABT transmission abort processing is in progress.
2. Make a CAN sleep mode/CAN stop mode transition request after the ABTTRG bit is cleared (after ABT mode is aborted) following the procedure shown in *Figure 14-47* or *Figure 14-48*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 14-45 on page 548*.

Figure 14-48 shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

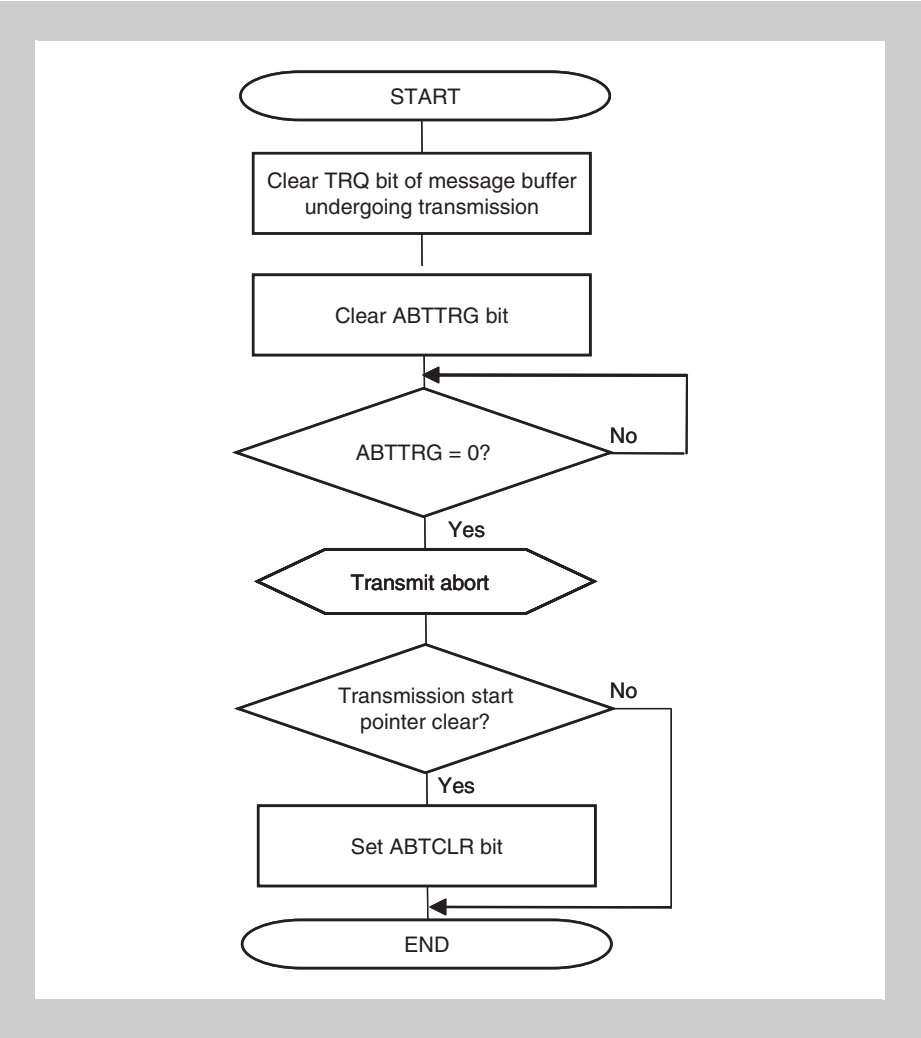


Figure 14-48 ABT transmission request abort processing (normal operation mode with ABT)

Caution

1. Do not set any transmission requests while ABT transmission abort processing is in progress.
2. Make a CAN sleep mode/CAN stop mode request after the ABTTRG bit is cleared (after ABT mode is stopped) following the procedure shown in *Figure 14-47* or *Figure 14-48*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 14-45 on page 548*.

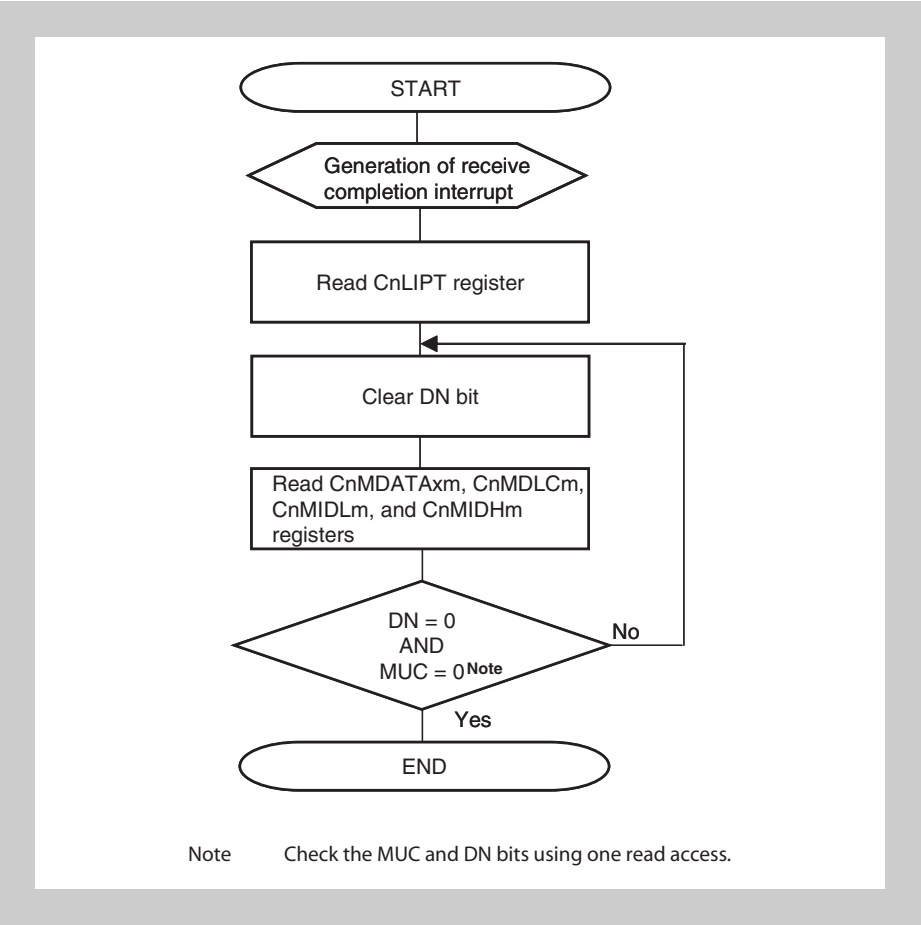


Figure 14-49 Reception via interrupt (using CnLIPT register)

Note Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing RX interrupts.

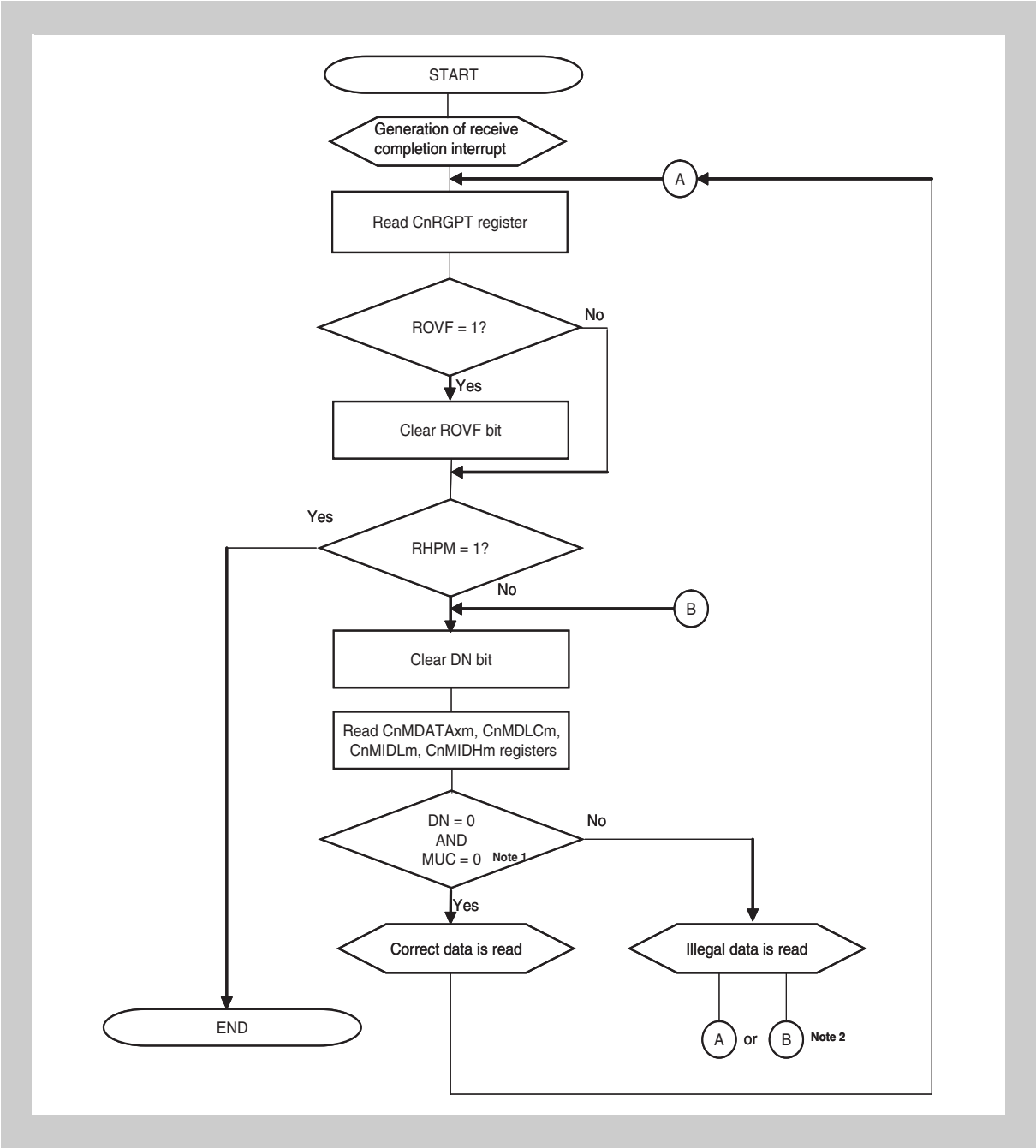


Figure 14-50 Reception via interrupt (using CnRGPT register)

Note 1. Check the MUC and DN bits using one read access.

2. Depending of the processing target of the application, two ways are possible:
– Way A: The message is not processed within this pass, but with the next pass, depending on the timing this can happen latest with the next Receive Interrupt.
Other messages will be processed earlier.
– Way B: The message is processed within this pass, the loop waits on this message.
Other messages will be processed later.

3. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
   It is recommended to cancel any sleep mode requests, before processing RX interrupts.
4. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

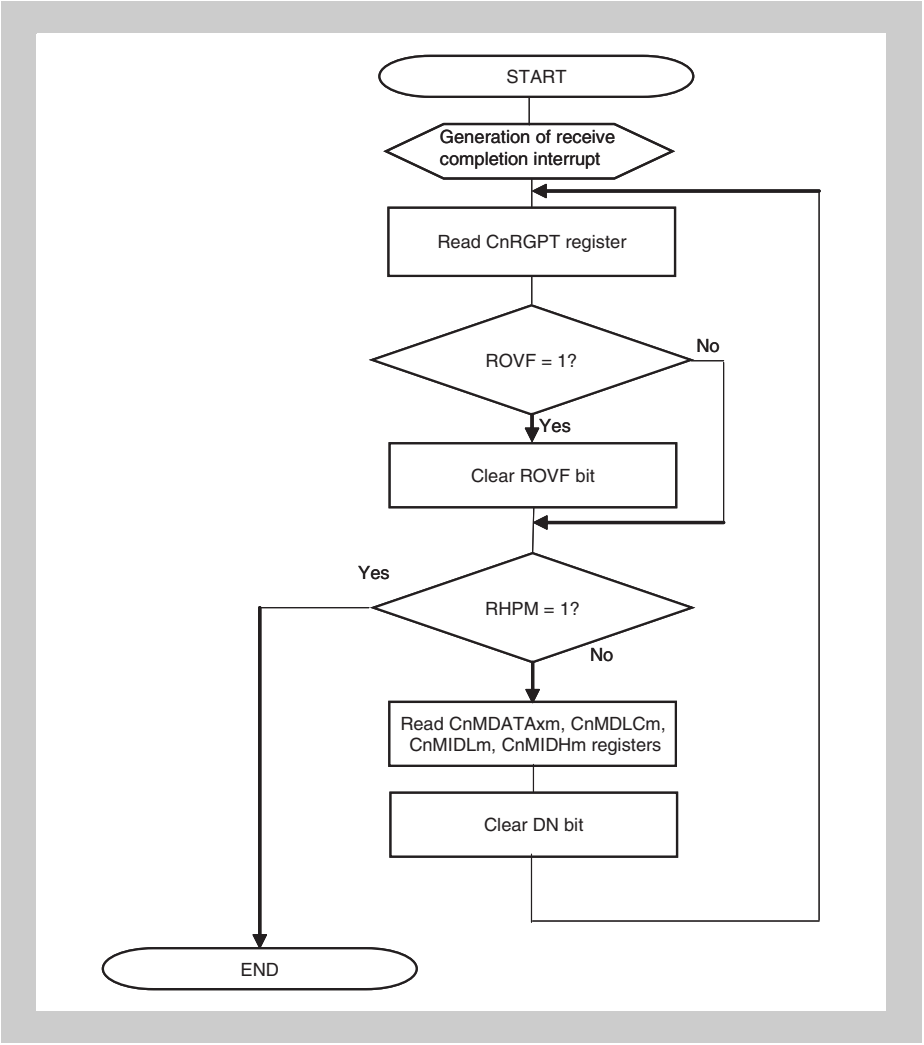


Figure 14-51 Reception via interrupt (using CnRGPT register), alternative way

Note 1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
   It is recommended to cancel any sleep mode requests, before processing RX interrupts.
2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

3. This flow will not provide most recently received data for the application. However, due to less effort on processing, it reduces interrupt load.
4. The overwrite function (CnMCONFm.OWS=1) must not be used with this flow - data inconsistency could occur.
5. It can be used alternatively to *Figure 14-50 on page 553*.

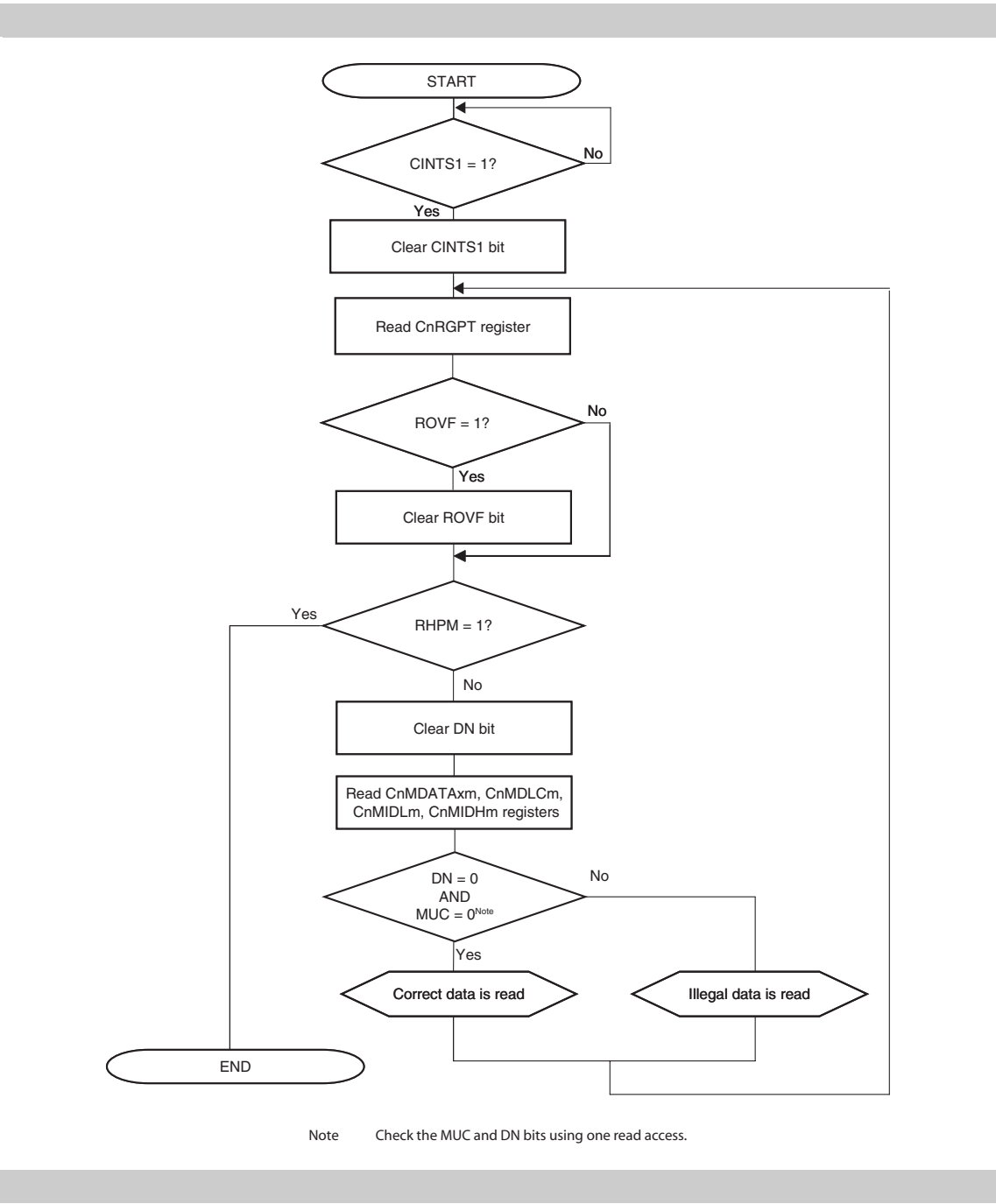


Figure 14-52 Reception via software polling

Note 1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

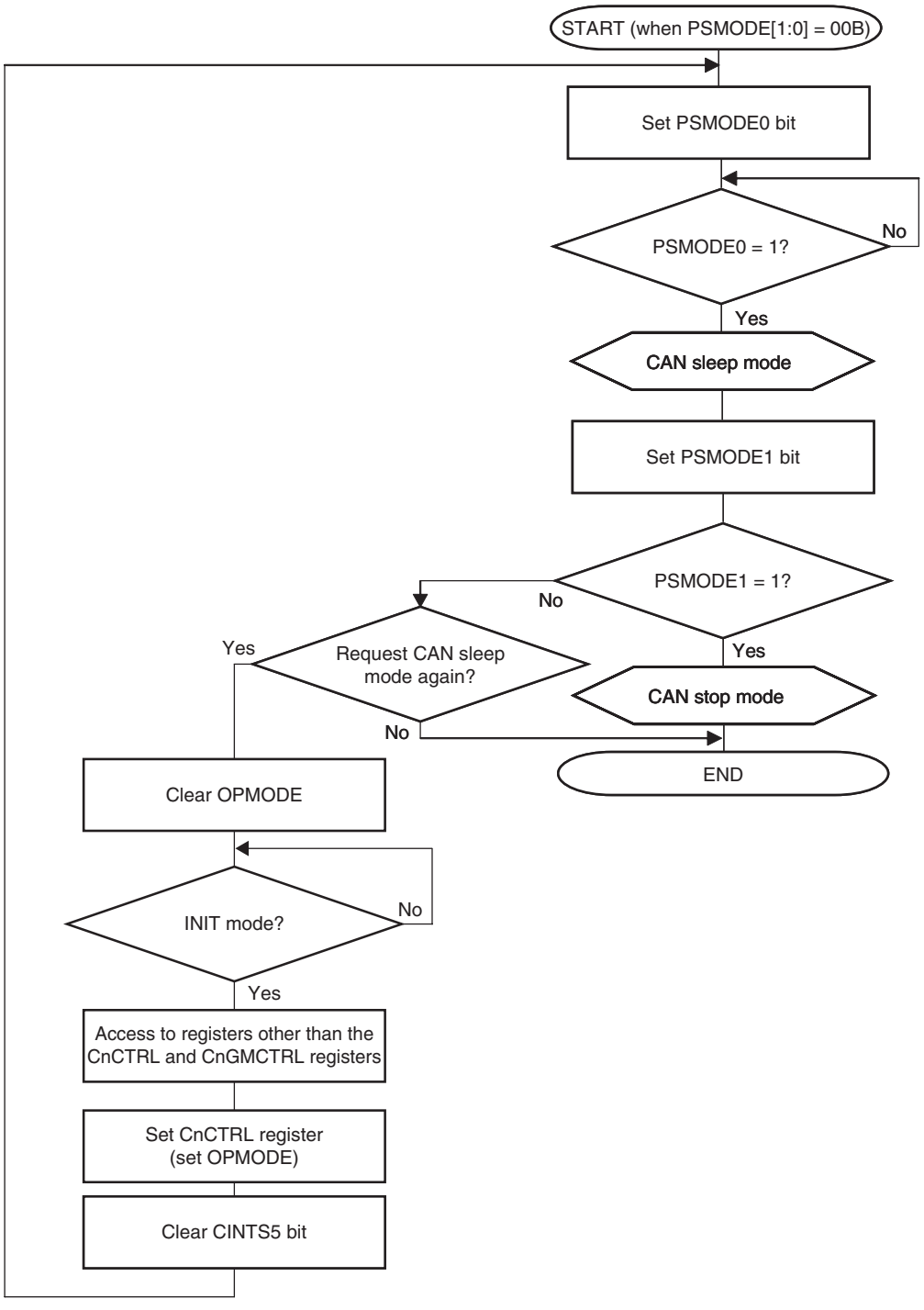


Figure 14-53 Setting CAN sleep mode/stop mode

Caution To abort transmission before making a request for the CAN sleep mode, perform processing according to *Figure 14-45 on page 548* and *Figure 14-47 on page 550*.

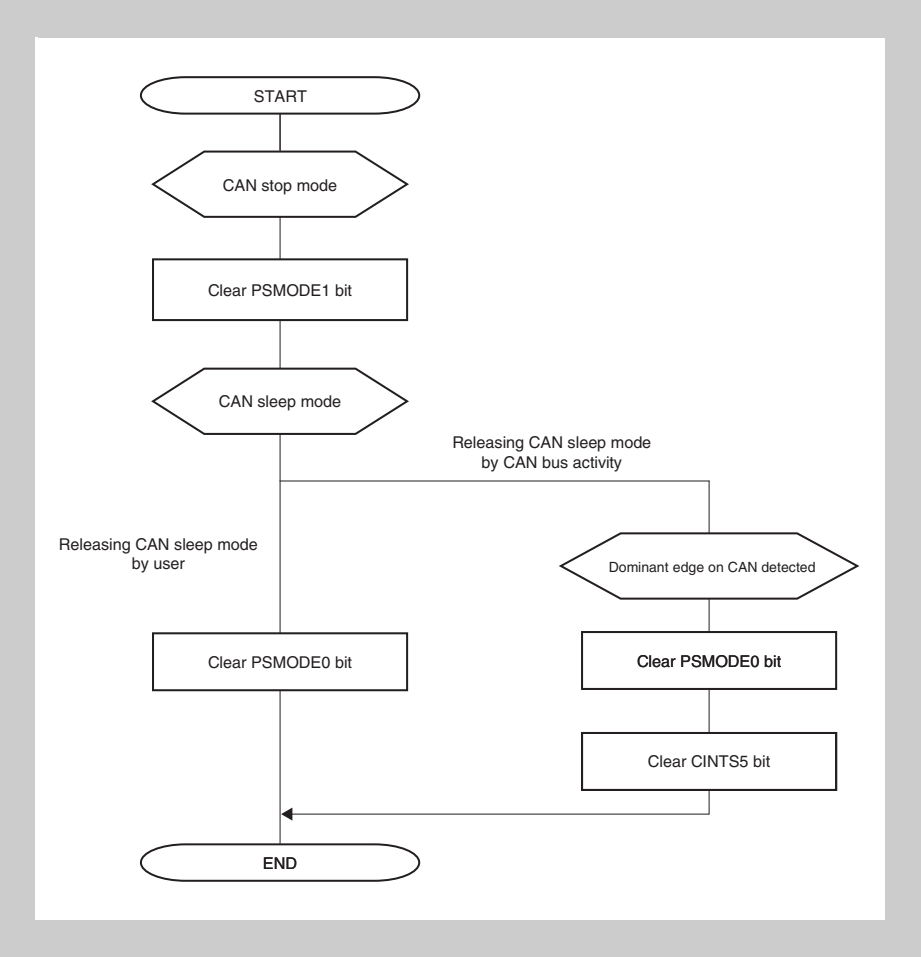


Figure 14-54 Clear CAN sleep/stop mode

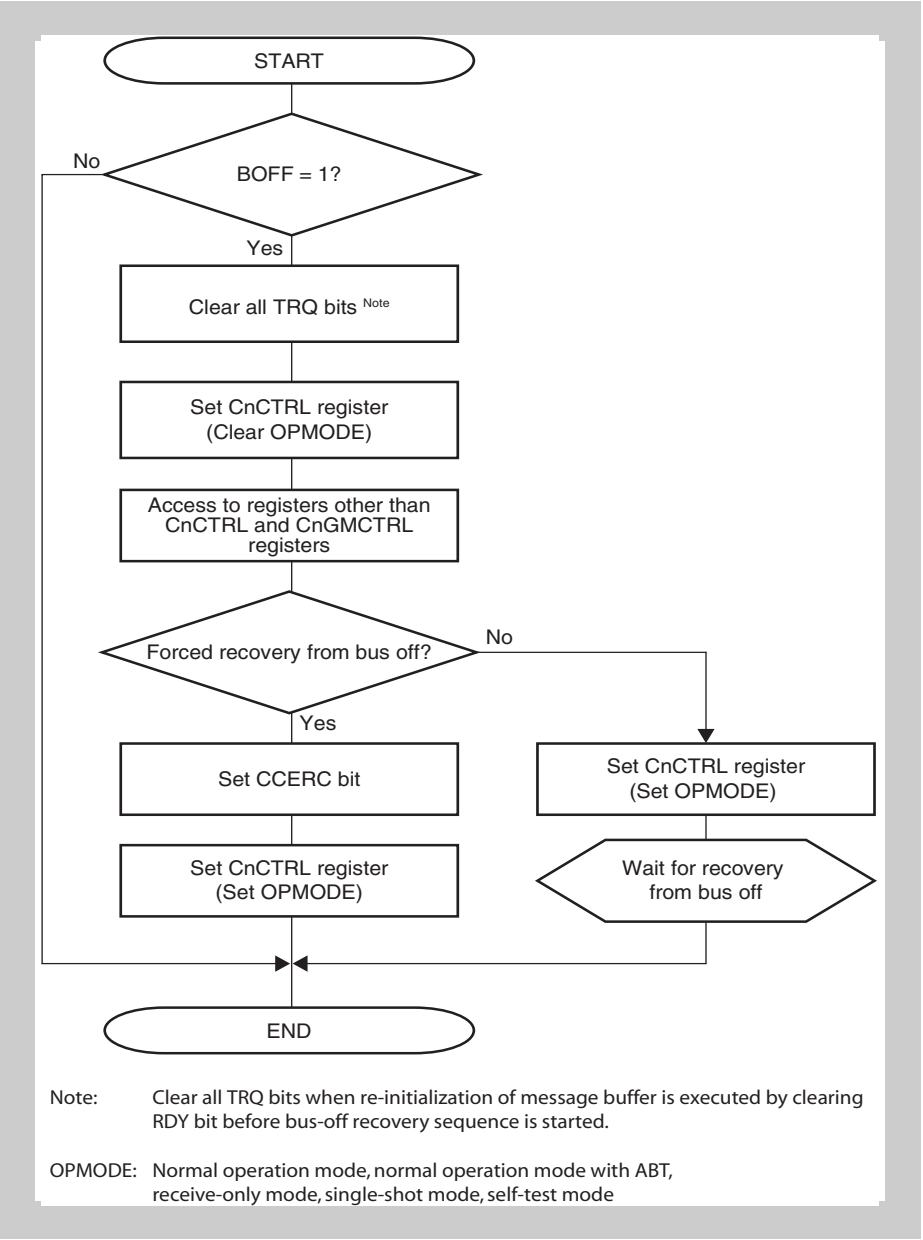


Figure 14-55 Bus-off recovery (except normal operation mode with ABT)

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.
Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

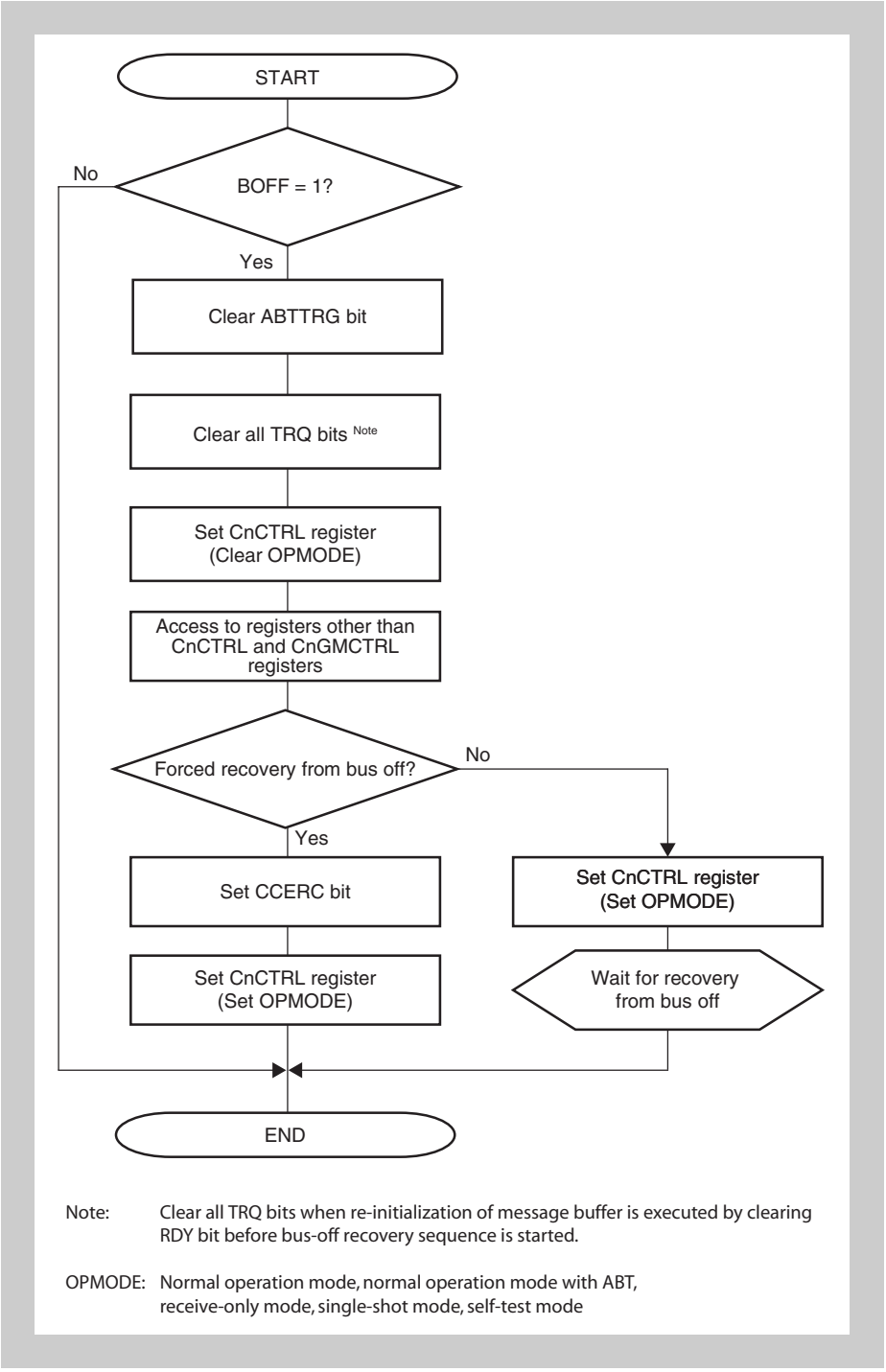


Figure 14-56 Bus-off recovery (Normal Operation Mode with ABT)

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.
Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

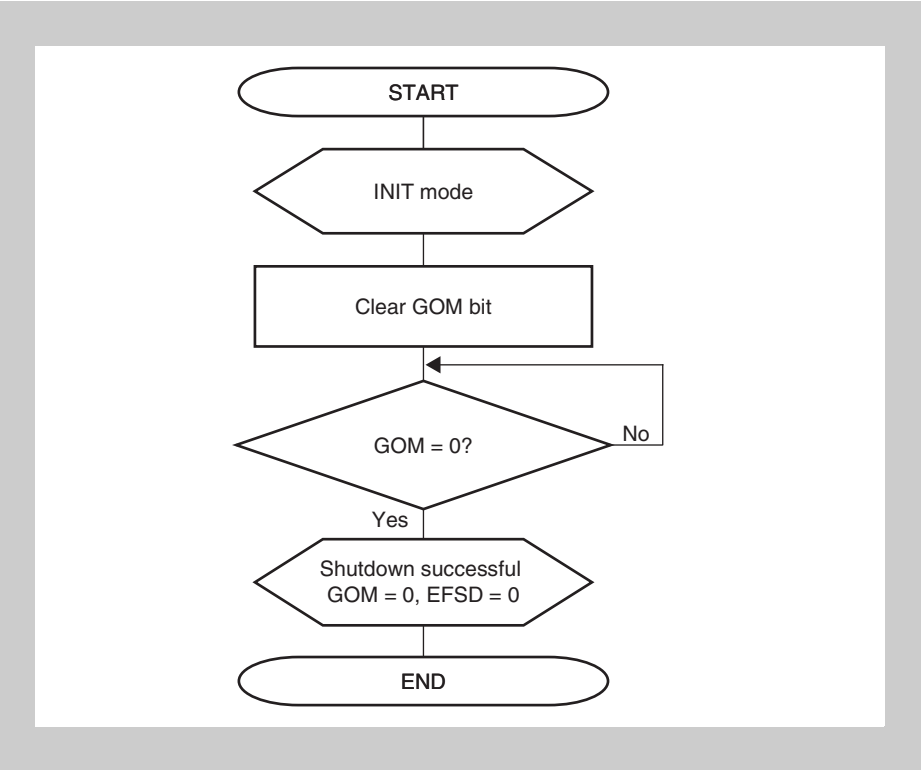


Figure 14-57 Normal shutdown process

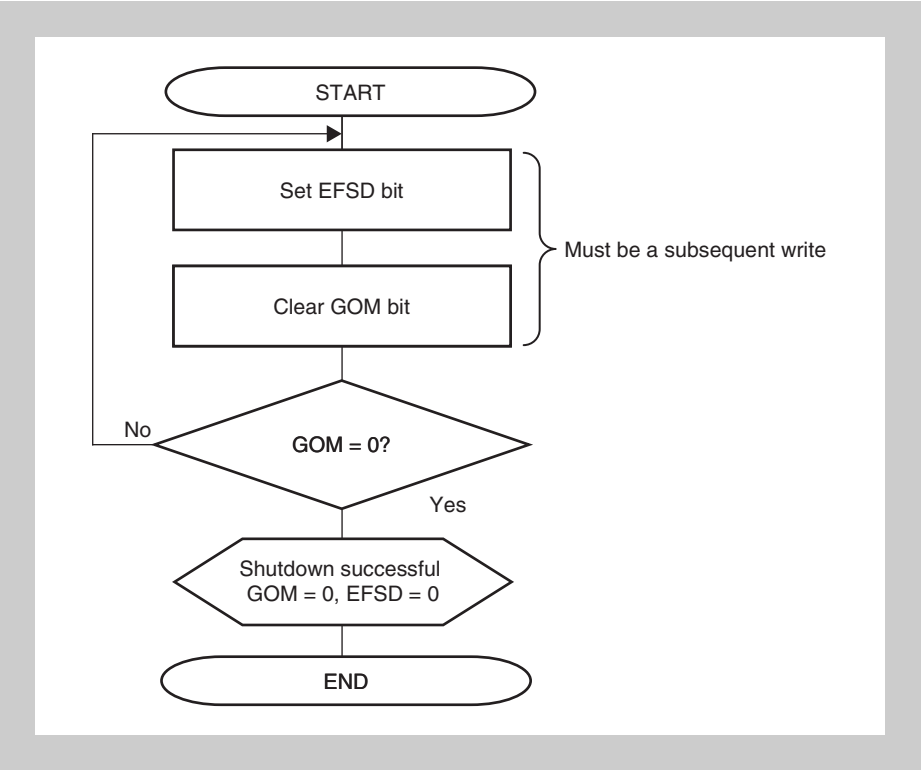


Figure 14-58 Forced shutdown process

Caution Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.

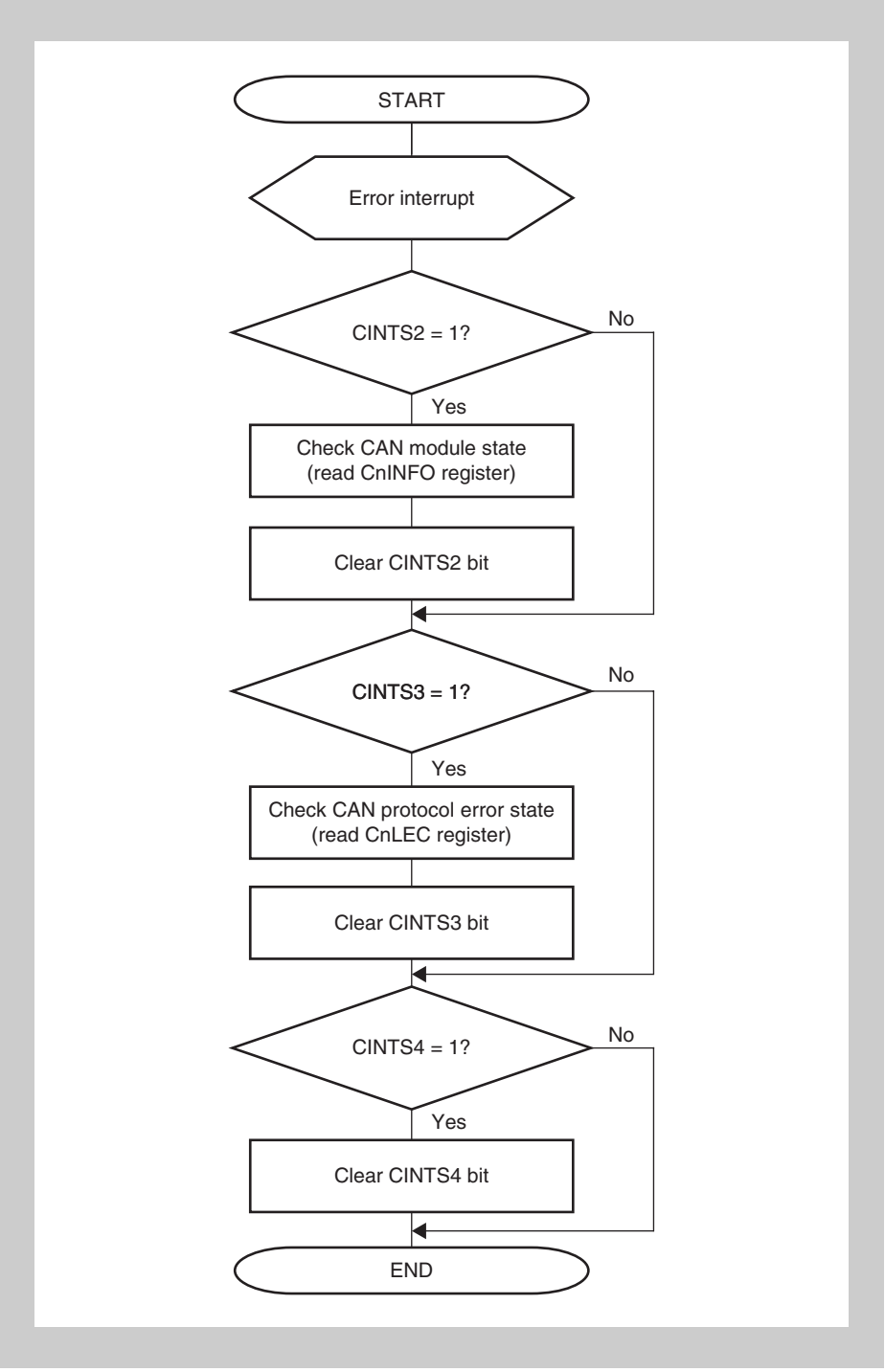


Figure 14-59 Error handling

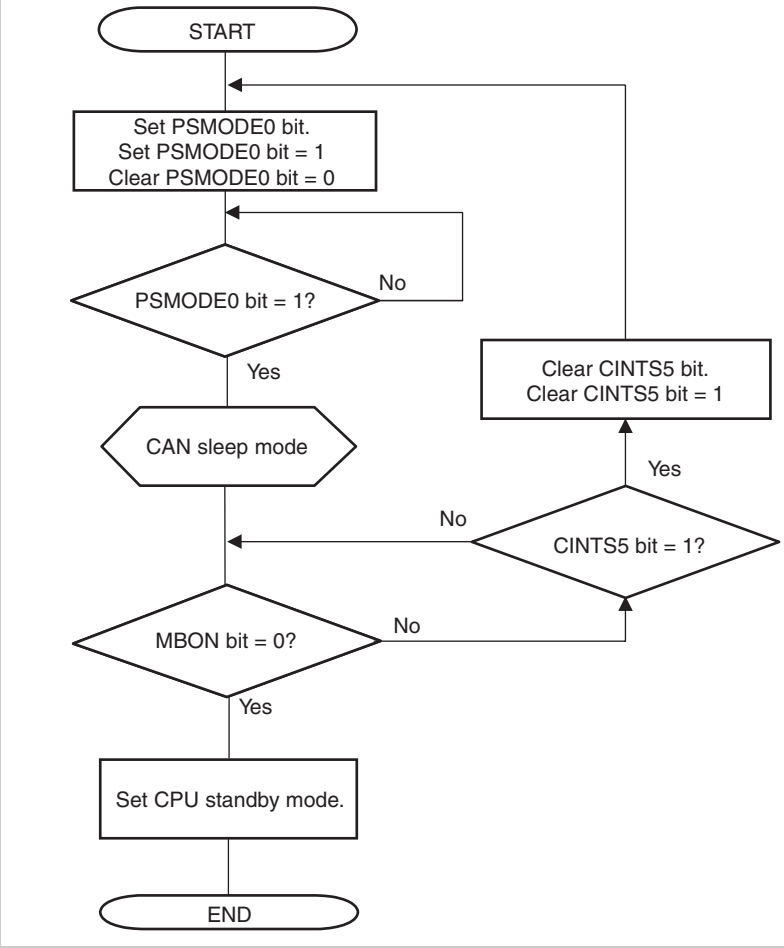


Figure 14-60 Setting CPU stand-by (from CAN sleep mode)

Caution Before the CPU is set in the CPU standby mode, please check if the CAN sleep mode has been reached.
However, after check of the CAN sleep mode, until the CPU is set in the CPU standby mode, the CAN sleep mode may be cancelled by wakeup from CAN bus.

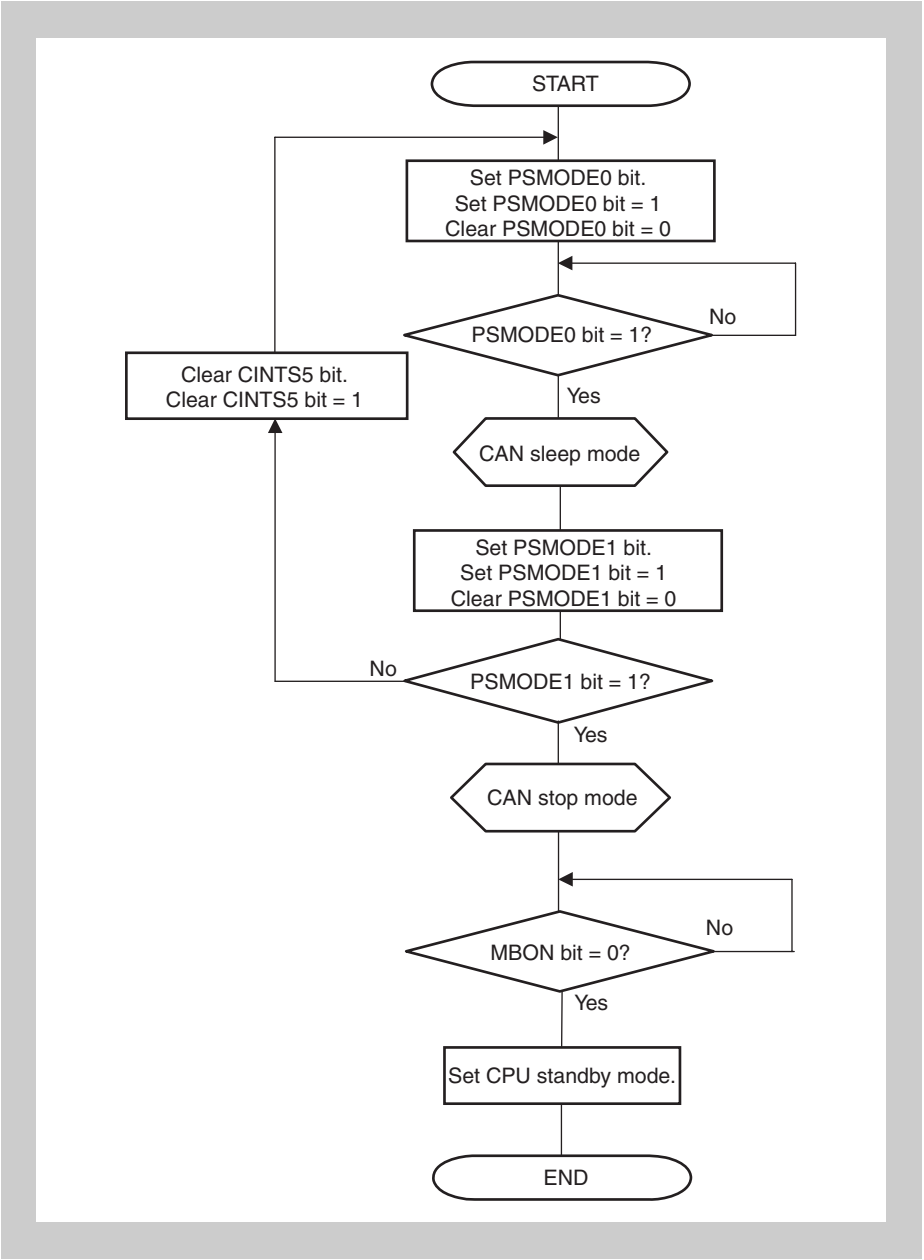


Figure 14-61 Setting CPU stand-by (from CAN stop mode)

Caution The CAN stop mode can only be released by writing 01_B to the PSMODE[1:0] bit of the CnCTRL register and not by a change in the CAN bus state.

Chapter 15 A/D Converter (ADC)

This microcontroller has two instances of the A/D Converter.

Note Throughout this chapter, the individual instances of the A/D Converter are identified by “n” (n = 0, 1). Each A/D Converter provides 10 channels, which are identified by “m” (m = 0 to 9).

15.1 Features

- Analog input: 2 × 10 channels (ANI00 to ANI09, ANI10 to ANI19)
- 10-bit resolution
- On-chip A/D conversion result register (ADCRn0 to ADCRn9):
  10 bits × 10
- A/D conversion trigger mode
  - A/D trigger mode
  - Timer trigger mode
  - External trigger mode
- Successive approximation method
- DMA transfer support of A/D conversion result to internal RAM

15.2 Configuration

The A/D Converter adopts the successive approximation method.

(1) Input circuit

The input circuit selects the analog input (ANIn0 to ANIn9) according to the mode set by the ADMn0, ADMn1, and ADMn2 registers.

(2) C-Array

Holds the charge of the differential voltage between the voltage input from the analog input pins (ANIn0 to ANIn9) and the reference voltage (1/2 AV_{DD}), and redistributes the sampled charges.

(3) C-Dummy

This block holds the reference voltage (1/2 AV_{DD}) and assigns the reference of the comparator input.

(4) Voltage Comparator

The Voltage Comparator compares the C-Array comparison potential with the C-Dummy reference potential.

(5) A/D conversion result register (ADCRnm), A/D conversion result register nH (ADCRnmH)

ADCRnm is a 10-bit register that holds A/D conversion results. Each time A/D conversion is completed, the conversion results are loaded from the successive approximation register (SAR).

$\overline{\text{RESET}}$ input makes this register undefined.

(6) A/D conversion result register for DMA transfer (ADDMAn)

ADDMAn is a 16-bit register that holds the last 10-bit A/D conversion result and an overrun flag for indicating a DMA transfer failure.

(7) ANIn0 to ANIn9 pins

These are 10-channel analog input pins for the A/D Converter n. They input the analog signals to be A/D converted.

Caution Make sure that the voltages input to ANIn0 to ANIn9 do not exceed the rated values. If a voltage higher than AV_{DD} or lower than AV_{SSn} (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

(8) AV_{REFn} pins

This is the pin for inputting the reference voltage of the A/D Converter. It converts signals input to the ANIn0 to ANIn9 pins to digital signals based on the voltage applied between AV_{SSn} and AV_{REFn}.

(9) AV_{SSn} pin

This is the ground pin of the A/D Converter. Always use this pin at the same potential as that of the EV_{SS} pin even when the A/D Converter is not used.

(10) AV_{DD} pin

This is the analog power supply pin of both A/D Converters (ADC0, ADC1).

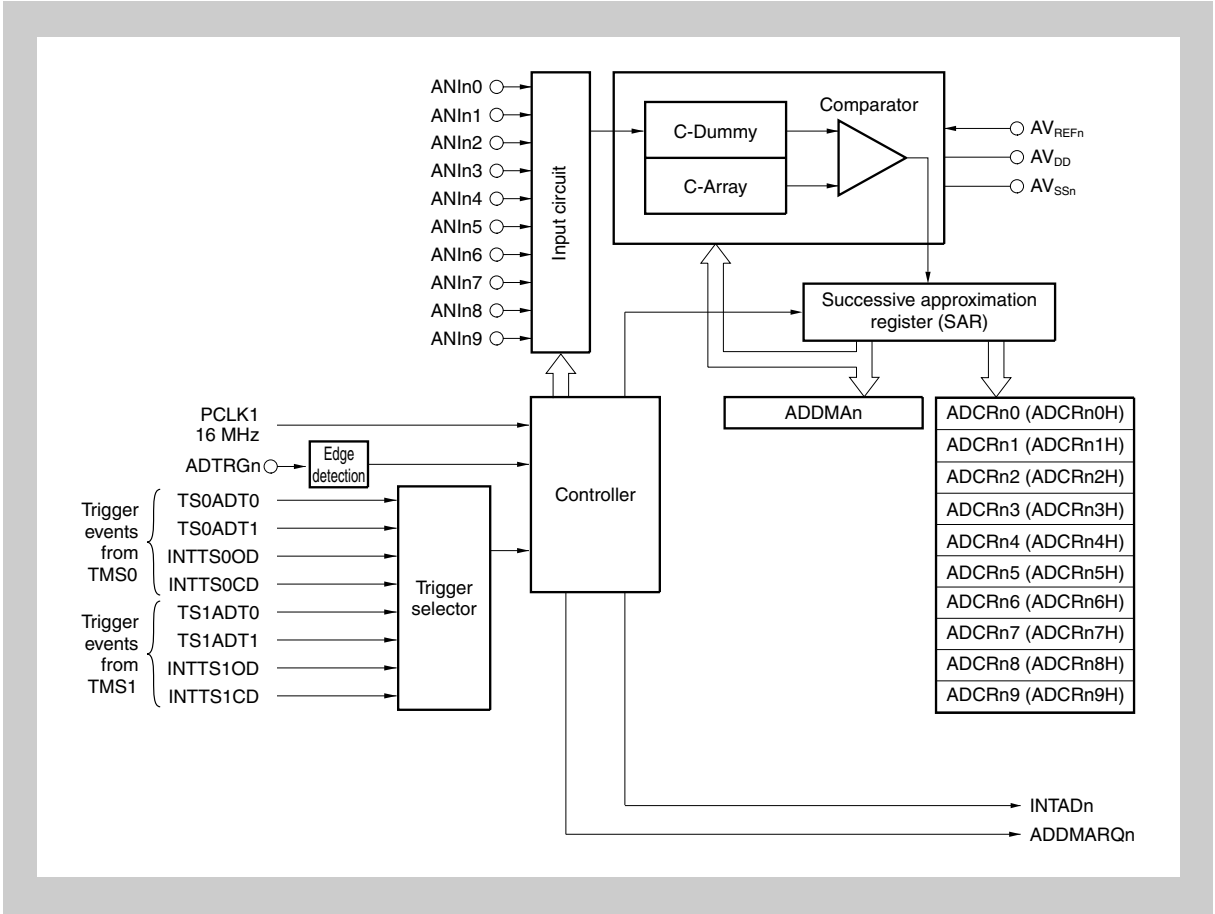


Figure 15-1 Block diagram of A/D Converter (ADCn)

Caution

1. If there is noise at the analog input pins (ANIn0 to ANIn9) or at the reference voltage input pin (AV_{REFn}), that noise may generate an illegal conversion result.
   Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.
   An example of this software processing is shown below.
   - Take the average result of a number of A/D conversions and use that as the A/D conversion result.
   - Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.
2. Do not apply a voltage outside the AV_{SSn} to AV_{REFn} range to the pins that are used as A/D Converter input pins.

15.3 Control Registers

(1) ADMn0 - A/D Converter mode register 0

The ADMn0 register is an 8-bit register that specifies the operation mode, and executes conversion operations.

This register can be read or written in 8-bit or 1-bit units. However, bit 6 can only be read. Writing this bit is ignored.

Reset input sets this register to 00H.

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 A/D trigger mode (ADMn1TRGn[1:0] = 00_B), 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. The A/D conversion can be restarted by writing to ADMn0 while A/D conversion in enabled (ADMn0.ADCEn = 1).
   In that case the conversion operation is initialized and conversion is executed from the beginning after the A/D converter stabilization time has elapsed.
   Note that the BSn and MSn bits must not be changed while A/D conversion in enabled, thus the same values must be re-written to restart the conversion.

After reset: 00H R/W Address: ADM00 FFFFF200H, ADM10 FFFFF240H

	7	6	5	4	3	2	1	0
ADMn0	ADCEn	ADCSn	BSn	MSn	0	0	0	0

ADCEn	A/D Conversion Operation Control of ADCn
0	Disables A/D conversion operation of ADCn
1	Enables A/D conversion operation ADCn

ADCSn	A/D Conversion Status Flag of ADCn
0	A/D conversion of ADCn is stopped
1	A/D conversion of ADCn is operating

BSn	ADCn Buffer Mode Specification
0	1-buffer mode
1	4-buffer mode

MSn	ADCn Operation Mode Specification
0	Scan mode
1	Select mode

(2) ADMn1 - A/D Converter mode register 1

The ADMn1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

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

Reset input sets this register to 00H.

Caution

1. Changing the setting of ADMn1is prohibited while A/D conversion is enabled (ADCEn bit of the ADMn0 register = 1).
2. The A/D conversion can be restarted by writing to ADMn1 while A/D conversion in enabled (ADMn0.ADCEn = 1).
   In that case the conversion operation is initialized and conversion is executed from the beginning after the A/D converter stabilization time has elapsed.
   Note that the ADnM1 value must not be changed while A/D conversion in enabled, thus the same value must be re-written to restart the conversion.

After reset: 00H R/W Address: ADM01 FFFFF201H, ADM11 FFFFF241H

	7	6	5	4	3	2	1	0
ADMn1	EGAn1	EGAn0	TRGn1	TRGn0	0	0	FRn1	FRn0

EGAn1	EGAn0	Valid Edge Specification of External Trigger Input (ADTRGn)
0	0	No edge detected (does not operate as external trigger)
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges, falling and rising edge detected

TRGn1	TRGn0	ADCn Trigger Mode Specification
0	0	A/D trigger mode (conversion starts with ADMn0.ADCEn = 1)
0	1	Timer trigger mode
1	0	External trigger mode
1	1	Setting prohibited

Table 15-1 A/D Converter mode register 1 (ADMn1)

FRn1	FRn0	Number of conversion clocks	Conversion Operation Time	A/D Stabilization Time[Note]
0	0	32	2.0 μs	1.0 μs
0	1	64	4.0 μs	2.0 μs
1	0	96	6.0 μs	2.5 μs
1	1	128	8.0 μs	2.5 μs
Others than above		Setting prohibited		

Note After 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.

(3) ADMn2 - A/D Converter mode register 2

The ADMn2 register is an 8-bit register that specifies the analog input pin of the A/D Converter n.

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

Reset input sets this register to 00H.

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 (ADCEn bit of the ADMn0 register = 1).
3. The A/D conversion can be restarted by writing to ADMn1 while A/D conversion in enabled (ADMn0.ADCEn = 1).
   In that case the conversion operation is initialized and conversion is executed from the beginning after the A/D converter stabilization time has elapsed.
   Note that the ADnM2 value must not be changed while A/D conversion in enabled, thus the same value must be re-written to restart the conversion.

After reset: 00H R/W Address: ADM02 FFFFF202H, ADM12 FFFFF242H

	7	6	5	4	3	2	1	0
ADMn2	0	0	0	0	ANISn3	ANISn2	ANISn1	ANIn0

Table 15-2 A/D Converter mode register 2 (ADMn2)

ANIn3	ANIn2	ANIn2	ANIn0	Specification of Analog Input Pins for A/D Conversion	
				Select Mode	Scan Mode
0	0	0	0	ANIn0	ANIn0
0	0	0	1	ANIn1	ANIn0, ANIn1
0	0	1	0	ANIn2	ANIn0 to ANIn2
0	0	1	1	ANIn3	ANIn0 to ANIn3
0	1	0	0	ANIn4	ANIn0 to ANIn4
0	1	0	1	ANIn5	ANIn0 to ANIn5
0	1	1	0	ANIn6	ANIn0 to ANIn6
0	1	1	1	ANIn7	ANIn0 to ANIn7
1	0	0	0	ANIn8	ANIn0 to ANIn8
1	0	0	1	ANIn9	ANIn0 to ANIn9
Others than above				Setting prohibited	

(4) ADTRSELn - A/D Converter trigger source select register

The ADTRSELn register is an 8-bit register that specifies the timer trigger signal in the timer trigger mode (TRGn1, TRGn0 bits of ADMn1 register = 01_B).

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

Reset input sets this register to 00H.

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 (ADCEn bit = 1).

After reset: 00H R/W Address: ADTRSEL0 FFFFF270H, ADTRSEL1 FFFFF272H

	7	6	5	4	3	2	1	0
ADTRSELn	0	0	0	0	TSELn3	TSELn2	TSELn1	TSELn0

Table 15-3 A/D Converter trigger source select register (ADTRSELn)

TSELn 3	TSELn 2	TSELn 1	TSELn 0	Trigger Source Selection in Timer Trigger Mode
0	0	0	1	TS0ADT0 signal (from TMS0)
0	0	1	0	TS0ADT1 signal (from TMS0)
0	0	1	1	TS1ADT0 signal (from TMS1)
0	1	0	0	TS1ADT1 signal (from TMS1)
0	1	0	1	INTTS0OD interrupt (from TMS0)
0	1	1	0	INTTS0CD interrupt (from TMS0)
0	1	1	1	INTTS1OD interrupt (from TMS1)
1	0	0	0	INTTS1CD interrupt (from TMS1)
Others than above				Setting prohibited

(5) ADCRn0 to ADCRn9, ADCRn0H to ADCRn9H - A/D conversion result registers

The ADCRnm register is a 10-bit register holding the A/D conversion results.

These registers are read-only in 16-bit or 8-bit units. When 16-bit access is performed, the ADCRnm register is specified, and when 8 bit access is performed, the ADCRnmH register holding the higher 8 bits of the conversion result is specified.

When reading the 10-bit data of the A/D conversion results from the ADCRnm register, only the higher 10 bits are valid and the lower 6 bits are always read as 0.

Reset input causes an undefined register content.

Address: ADCR00 FFFFF210H, ADCR10 FFFFF250H,
ADCR01 FFFFF212H, ADCR11 FFFFF252H,
ADCR02 FFFFF214H, ADCR12 FFFFF254H,
ADCR03 FFFFF216H, ADCR13 FFFFF256H,
ADCR04 FFFFF218H, ADCR14 FFFFF258H,
ADCR05 FFFFF21AH, ADCR15 FFFFF25AH,
ADCR06 FFFFF21CH, ADCR16 FFFFF25CH,
ADCR07 FFFFF21EH, ADCR17 FFFFF25EH,
ADCR08 FFFFF220H, ADCR18 FFFFF260H,
ADCR09 FFFFF222H, ADCR19 FFFFF262H,

After reset: Undefined R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRnm	ADnm9	ADnm8	ADnm7	ADnm6	ADnm5	ADnm4	ADnm3	ADnm2	ADnm1	ADnm0	0	0	0	0	0	0

ADCRnmH

Address: ADCR00H FFFFF211H, ADCR10H FFFFF251H,
ADCR01H FFFFF213H, ADCR11H FFFFF253H,
ADCR02H FFFFF215H, ADCR12H FFFFF255H,
ADCR03H FFFFF217H, ADCR13H FFFFF257H,
ADCR04H FFFFF219H, ADCR14H FFFFF259H,
ADCR05H FFFFF21BH, ADCR15H FFFFF25BH,
ADCR06H FFFFF21DH, ADCR16H FFFFF25DH,
ADCR07H FFFFF21FH, ADCR17H FFFFF25FH,
ADCR08H FFFFF221H, ADCR18H FFFFF261H,
ADCR09H FFFFF223H, ADCR19H FFFFF263H,

After reset: Undefined R

	7	6	5	4	3	2	1	0
ADCRnmH	ADnm9	ADnm8	ADnm7	ADnm6	ADnm5	ADnm4	ADnm3	ADnm2

The correspondence between each analog input pin and the ADCRnm register is shown in Table 15-4 below.

Table 15-4 Assignment of A/D conversion result registers to analog input pins

Analog Input Pin	Assignment of A/D Conversion Result Registers	
	Select 1 Buffer Mode/ Scan Mode	Select 4 Buffer Mode
ANIn0	ADCRn0, ADCRn0H	ADCRn0 to ADCRn3, ADCRn0H to ADCRn3H
ANIn1	ADCRn1, ADCRn1H	
ANIn2	ADCRn2, ADCRn2H	
ANIn3	ADCRn3, ADCRn3H	
ANIn4	ADCRn4, ADCRn4H	ADCRn4 to ADCRn7, ADCRn4H to ADCRn7H
ANIn5	ADCRn5, ADCRn5H	
ANIn6	ADCRn6, ADCRn6H	
ANIn7	ADCRn7, ADCRn7H	
ANIn8	ADCRn8, ADCRn8H	ADCRn0 to ADCRn3, ADCRn0H to ADCRn3H
ANIn9	ADCRn9, ADCRn9H	

The relationship between the analog voltage input to the analog input pins (ANIn0 to ANIn9) and the A/D conversion result (of the A/D conversion result register (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 the upper 10 bit of the A/D conversion result register, i.e. ADCRnm[15:6]

Figure 15-2 on page 578 shows the relationship between the analog input voltage and the A/D conversion results.

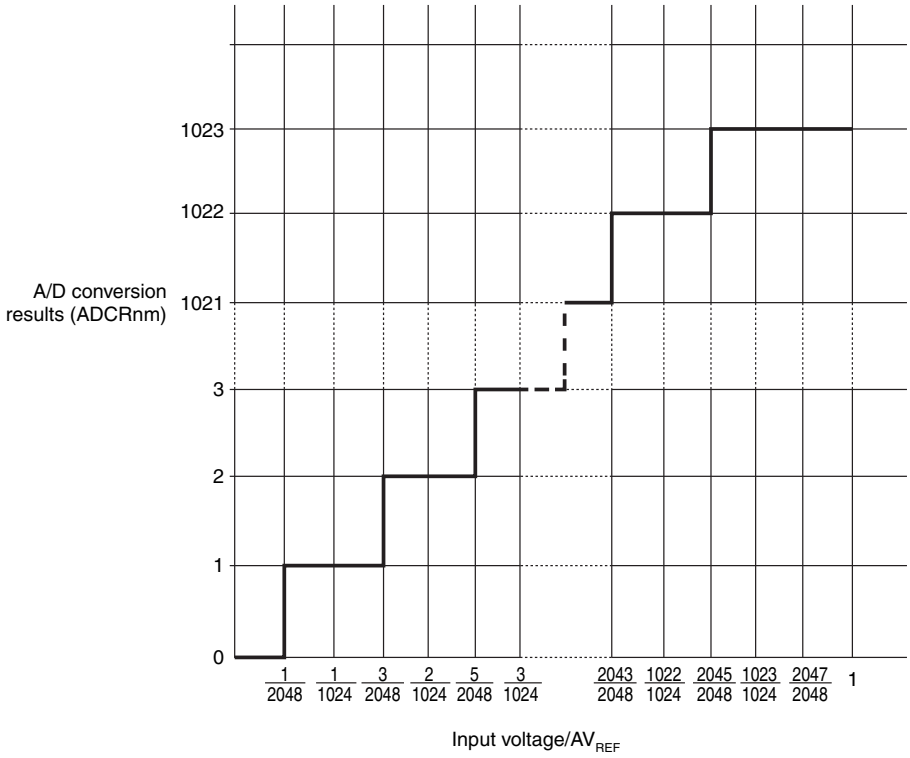


Figure 15-2 Relationship between analog input voltage and A/D conversion results

(6) ADDMAn - A/D conversion result register for DMA

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.

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

Reset input causes an undefined register content.

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.

After reset: Undefined R Address: ADDMA0 FFFFF22EH, ADDMA1 FFFFF26EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMAn	ADDMAn9	ADDMAn8	ADDMAn7	ADDMAn6	ADDMAn5	ADDMAn4	ADDMAn3	ADDMAn2	ADDMAn1	ADDMAn0	0	0	0	0	0	ODFn

ADDMAn9 to ADDMAn0	A/D Conversion Result for DMA Transfer
000H to 3FFH	Latest A/D conversion result value

ODFn	Overrun Detection Flag
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.

- The ODFn flag is used for indicating a DMA transfer failure of the A/D conversion results.
- The ODFn flag is cleared (0), when the A/D conversion is stopped (ADCEn bit of the ADMn0 register is cleared to 0).

15.4 Operation

15.4.1 Basic operation

A/D conversion is executed by the following procedure.

1. The selection of the analog input and specification of the operation mode, trigger mode, etc. should be specified using the ADMn0, ADMn1 or ADMn2 registers[Note 1].
   When the ADCEn bit of the ADMn0 register is set to 1, A/D conversion starts in the A/D trigger mode. In the timer trigger mode and external trigger mode, the trigger standby state[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.

Note 1. If the setting of the ADMn0, ADMn1 or ADMn2 registers is changed during A/D conversion, the operation immediately before is stopped, and the result of the conversion is not stored in the ADCRnm register. The A/D conversion operation is then initialized, and conversion is executed from the beginning again.

2. During the timer trigger mode and external trigger mode, if the ADCEn bit of the ADMn0 register is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal (ADCSn bit in the ADMn0 register = 1), and the trigger standby state (ADCSn bit = 0) is returned when the A/D conversion operation ends.

15.4.2 Operation mode and trigger mode

Various conversion operations can be specified for the A/D Converter by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADMn0 and ADMn1registers.

The following table shows the relationship between the operation mode and trigger mode.

Table 15-5 Relationship between operation mode and trigger mode

Trigger Mode	Operation Mode		Register Set Value	
			ADMn0	ADMn1
A/D trigger	Select	1 buffer	xx010000B	xx0000xxB
		4 buffers	xx110000B	
	Scan		xx000000B	
Timer trigger	Select	1 buffer	xx010000B	xx0100xxB
		4 buffers	xx110000B	
	Scan		xx000000B	
External trigger	Select	1 buffer	xx010000B	xx1000xxB
		4 buffers	xx110000B	
	Scan		xx000000B	

(1) Trigger mode

There are three types of trigger modes that serve as the start timing of A/D conversion processing: A/D trigger mode, timer trigger mode, and external trigger mode. These trigger modes are set by the TRGn1 and TRGn0 bits of the ADMn1 register.

- **A/D trigger mode**

  This mode starts the conversion timing of the analog input set to the ANIn0 to ANIn9 pins, and by setting the ADCEn bit of the ADMn0 register to 1, starts A/D conversion. Unless the ADCEn bit is cleared to 0 after conversion, the next conversion operation is repeated. If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

- **Timer trigger mode**

  This mode specifies the conversion timing of the analog input set for the ANIn0 to ANIn9 pins using signals from the inverter timer S (TMS0, TMS1).

  The ADTRSELn register specifies the analog input conversion timing by selecting one of the A/D Converter trigger signals connected to the 16-bit inverter timer S (TMS0, TMS1).

  If the ADCEn bit of the ADMn0 register is set to 1, the A/D Converter waits for an event input, and starts conversion when the event occurs (ADCSn bit of the ADMn0 register = 1). When conversion has finished, the converter waits for an event input again (ADCSn bit = 0). If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

- **External trigger mode**

  This mode specifies the conversion timing of the analog input to the ANIn0 to ANIn9 pins using the ADTRGn pin.

  The EGAn1 and EGAn0 bits of the ADMn1 register are used to specify the valid edge to be input to the ADTRGn pin.

  When the ADCEn bit of the ADMn0 register 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 (ADCSn bit of the ADMn0 register = 1). When the converter has finished its conversion operation, it waits for an external trigger again (ADCSn bit = 0).

  If the valid edge is detected at the ADTRGn pin during conversion, conversion is executed from the beginning again.

  If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

(2) Operation mode

There are two operation modes that set the ANIn0 to ANIn9 pins: select mode and scan mode. The select mode has sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the BSn and MSn bits of the ADMn0 register.

- **Select mode**

  In this mode, one analog input specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm). For this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results.

– **1-buffer mode**
In this mode, one analog input specified by the ADM2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm). The ANInm and ADCRnm register correspond one to one, and an A/D conversion end interrupt (INTADn) is generated each time one A/D conversion ends. After conversion has finished, the next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

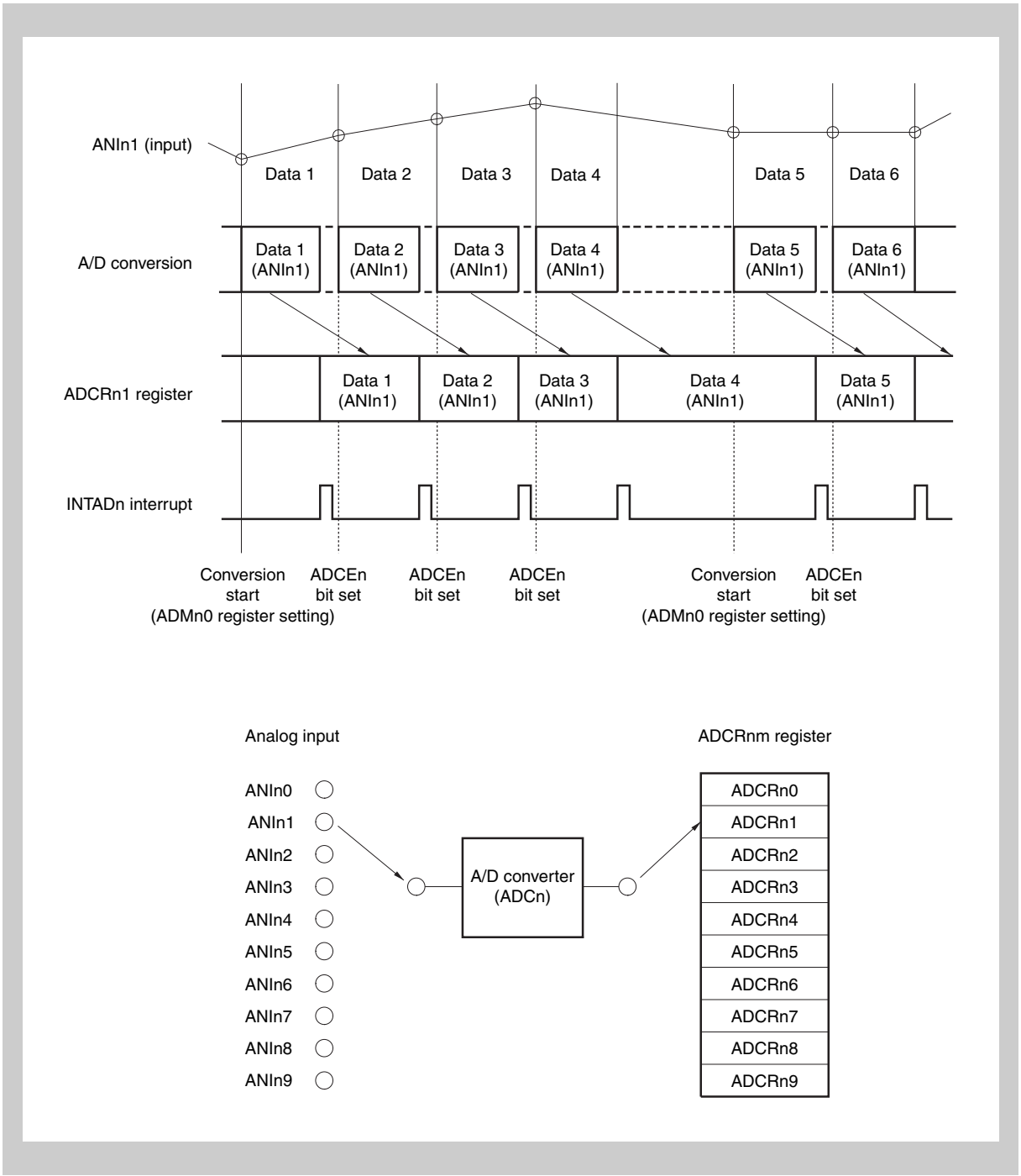


Figure 15-3 Select mode operation timing: 1-buffer mode (conversion of ANIn1 in A/D trigger mode)

– **4-buffer mode**

In this mode, one analog input is A/D converted and the results are stored in the 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 = 0 to 3 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.

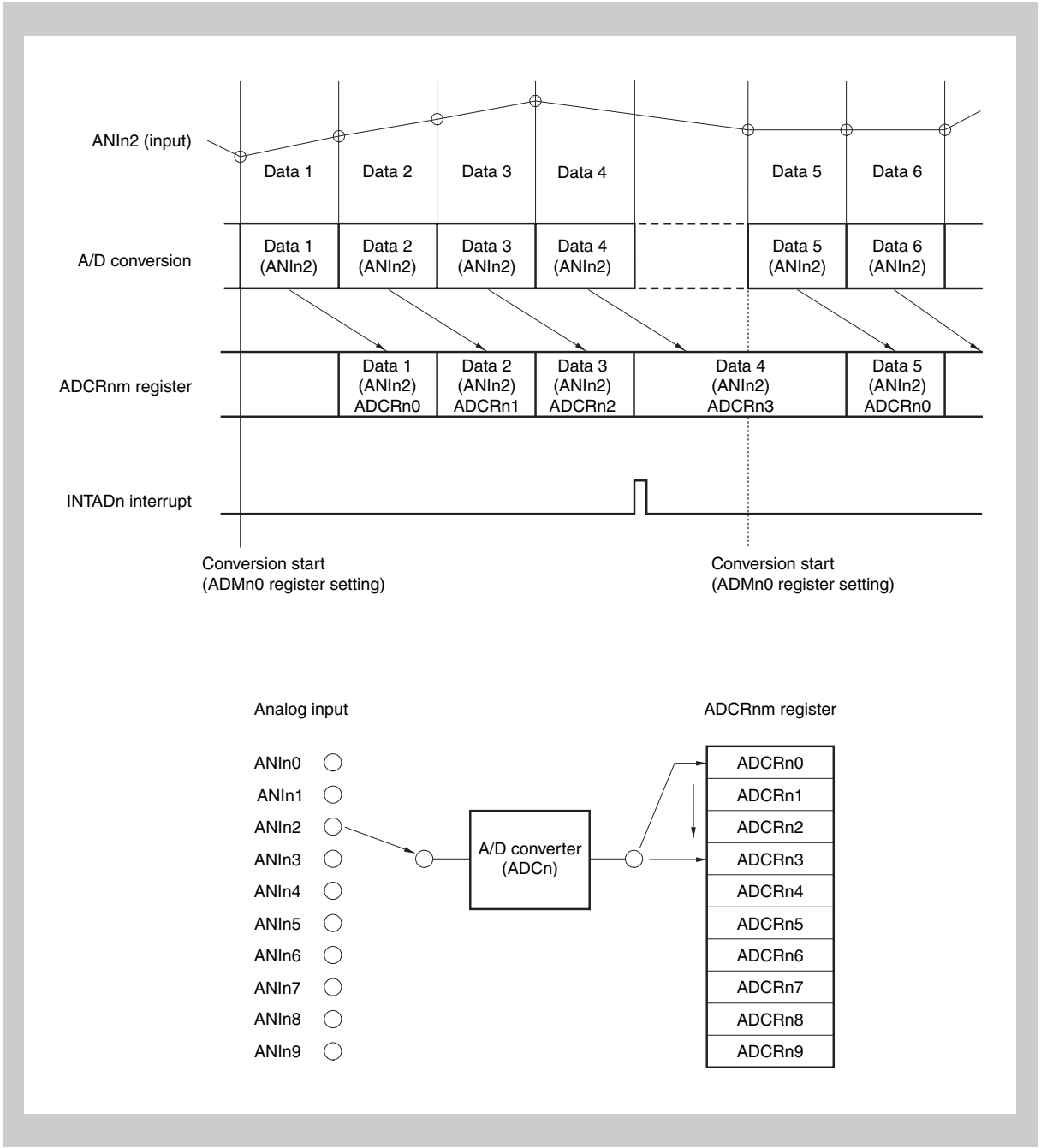


Figure 15-4 Select mode operation timing: 4-buffer mode (conversion of ANIn2 in A/D trigger mode)

- **Scan mode**

  In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input. When the conversion of the specified analog input ends, the A/D conversion end interrupt (INTADn) is generated. After conversion has finished, the next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

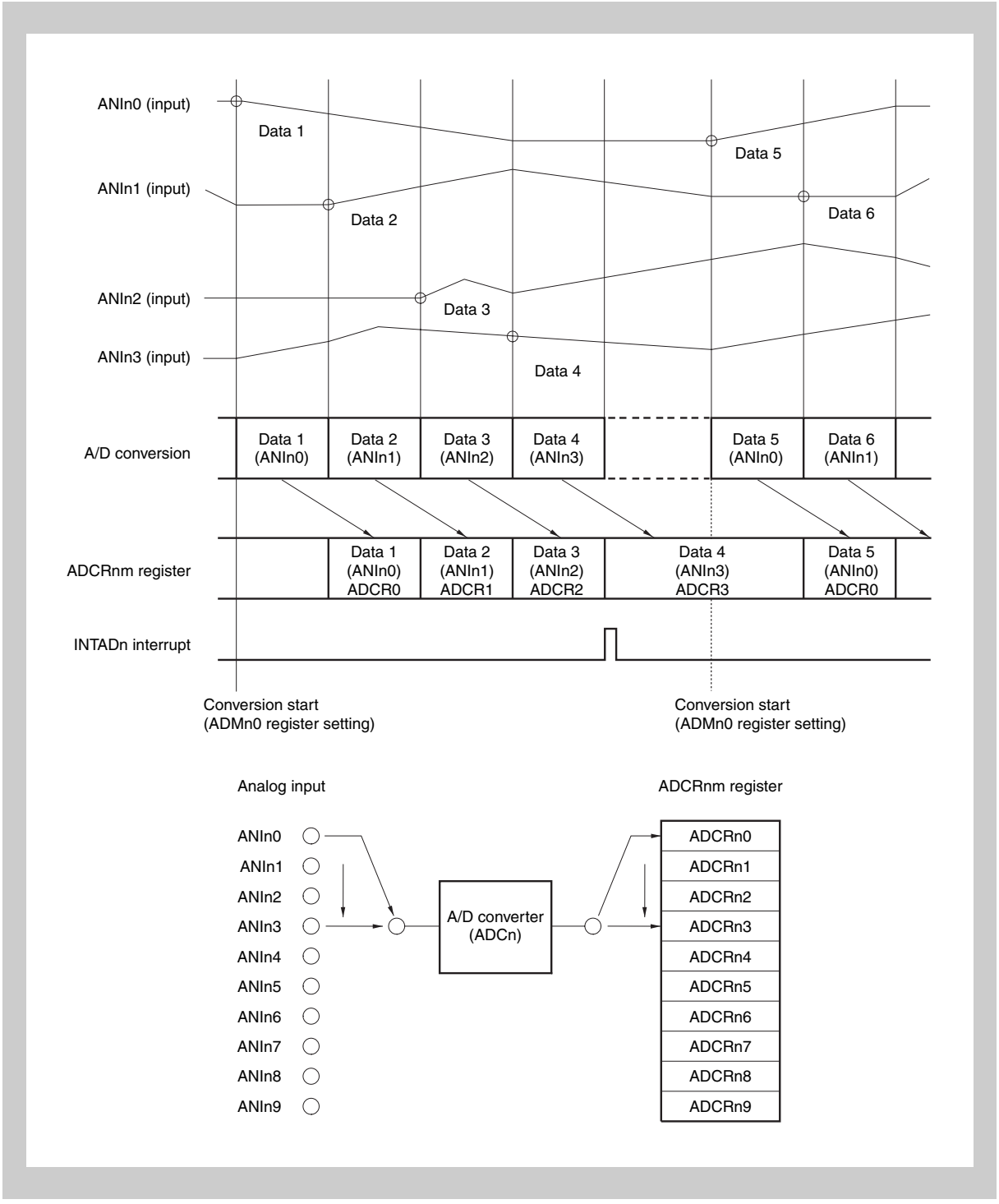


Figure 15-5 Scan mode operation timing: 4-channel scan (conversion of ANI0 to ANI3 in A/D trigger mode)

15.5 Operation in A/D Trigger Mode

When the ADCEn bit of the ADMn0 register is set to 1, A/D conversion is started.

15.5.1 Select mode operation

In this mode, the analog input specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are supported according to the storing method of the A/D conversion results.

(1) 1-buffer mode (A/D trigger select: 1 buffer)

In this mode, one analog input is A/D converted once. The conversion results are stored in one ADCRn register. The analog input and ADCRn register correspond one to one.

Each time an A/D conversion is executed, an A/D conversion end interrupt (INTAD) is generated and A/D conversion ends. The next conversion operation is repeated, unless the ADCE bit of the ADM0 register is cleared to 0.

Table 15-6 Correspondence between analog input pins and ADCRnm register (A/D trigger select: 1 buffer)

Analog Input	A/D Conversion Result Register
ANInm	ADCRnm

This mode is most appropriate for applications in which the results of each first-time A/D conversion are read.

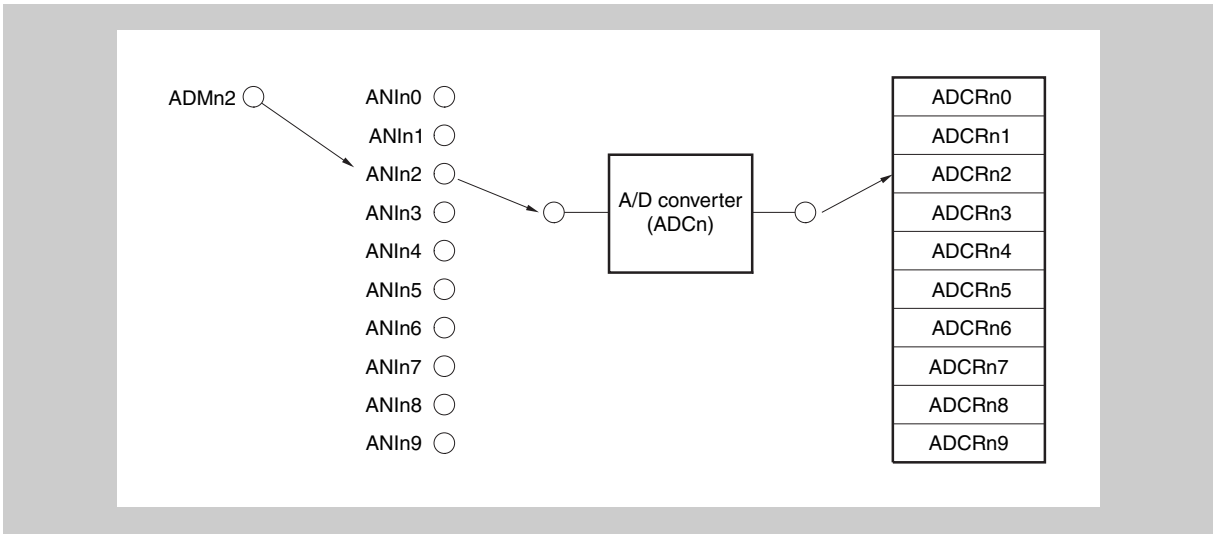


Figure 15-6 Example of 1-buffer mode operation (A/D trigger select: 1 buffer)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. ANIn2 is A/D converted
3. The conversion result is stored in ADCRn2 register
4. The INTAD interrupt is generated

(2) 4-buffer mode (A/D trigger select: 4 buffers)

In this mode, one analog input is A/D converted four times and the results are stored in the 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 ADCEn bit of the ADMn0 register is cleared to 0.

Table 15-7 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	ADCRn0 (1st time)
	ADCRn1 (2nd time)
	ADCRn2 (3rd time)
	ADCRn3 (4th time)

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

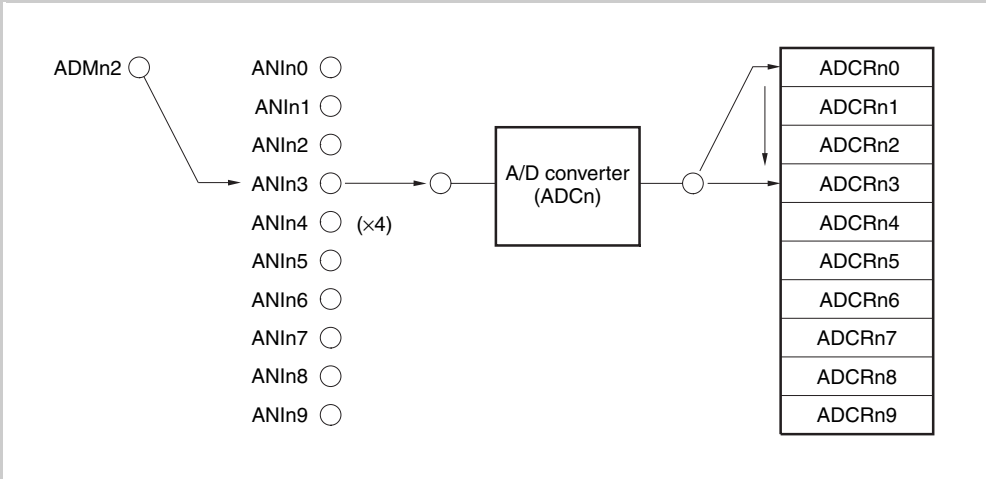


Figure 15-7 Example of 4-buffer mode operation (A/D trigger select: 4 buffers)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. ANIn3 is A/D converted
3. The conversion result is stored in ADCRn0 register
4. ANIn3 is A/D converted
5. The conversion result is stored in ADCRn1 register
6. ANIn3 is A/D converted
7. The conversion result is stored in ADCRn2 register
8. ANIn3 is A/D converted

9. The conversion result is stored in ADCRn3 register
10. The INTAD interrupt is generated

15.5.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input.

When conversion of all the specified analog input ends, the A/D conversion end interrupt (INTADn) is generated, and A/D conversion is stopped. The next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

Table 15-8 Correspondence between analog input pins and ADCRnm register (A/D trigger scan)

Analog Input	A/D Conversion Result Register
ANIn0 . . . ANInm[Note]	ADCRn0 . . . ADCRnm

Note 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.

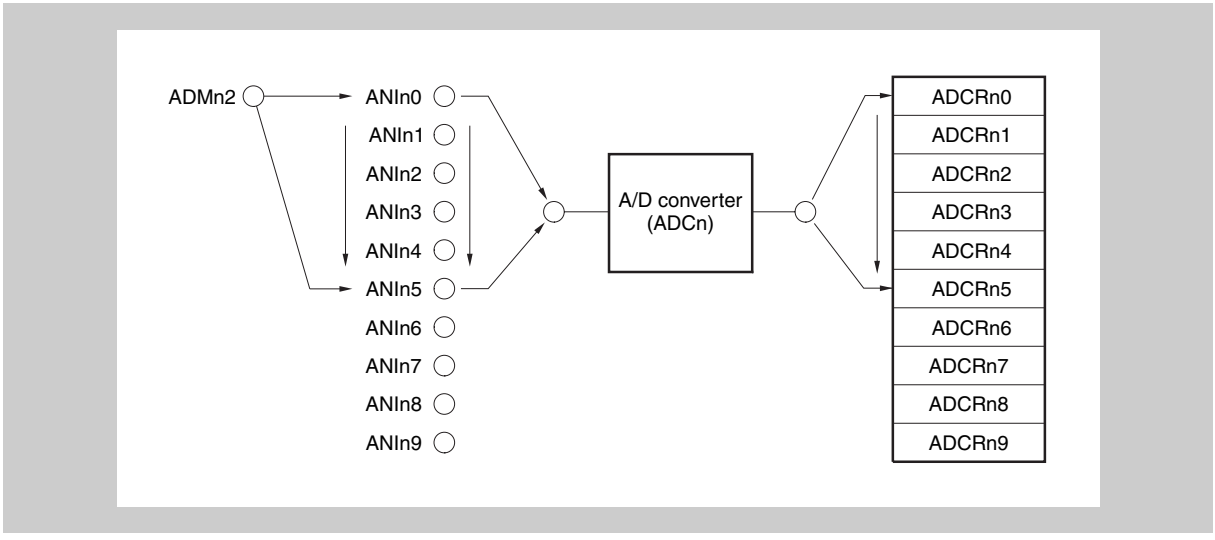


Figure 15-8 Example of scan mode operation (A/D trigger scan)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. ANIn0 is A/D converted
3. The conversion result is stored in ADCRn0
4. ANIn1 is A/D converted
5. The conversion result is stored in ADCRn1
6. ANIn2 is A/D converted

7. The conversion result is stored in ADCRn2
8. ANIn3 is A/D converted
9. The conversion result is stored in ADCRn3
10. ANIn4 is A/D converted
11. The conversion result is stored in ADCRn4
12. ANIn5 is A/D converted
13. The conversion result is stored in ADCRn5
14. The INTAD interrupt is generated

15.6 Operation in Timer Trigger Mode

In this mode, the conversion timing of the analog input signal set by the ANIn0 to ANIn9 pins is defined by a timer event signal.

The analog input conversion timing is generated when an A/D Converter trigger signal from the timers is generated.

When the ADCEn bit of the ADMn0 register is set to 1, the A/D Converter waits for the timer signal and starts conversion when the timer event occurs (ADCSn bit of the ADMn0 register = 1). When conversion is finished, the converter waits for a timer event signal again (ADCSn bit = 0).

If the timer event signal occurs during conversion, the conversion operation is executed from the beginning again.

If data is written to the ADMn0 to ADMn2 registers during conversion, the conversion operation is stopped and executed from the beginning again.

15.6.1 Select mode operation

In this mode, an analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are provided according to the storing method of the A/D conversion results.

(1) 1-buffer mode operation (timer trigger select: 1 buffer)

In this mode, one analog input is A/D converted once and the conversion results are stored in one ADCRnm register.

One analog input is A/D converted once using the trigger of the timer event signals and the results are stored in one ADCRnm register. An A/D conversion end interrupt (INTADn) is generated for each A/D conversion.

Unless the ADCEn bit of the ADMn0 register is cleared to 0, A/D conversion is repeated each time a timer event signal is generated.

Table 15-9 Correspondence between analog input pins and ADCRnm register (1-buffer mode (timer trigger select: 1 buffer))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

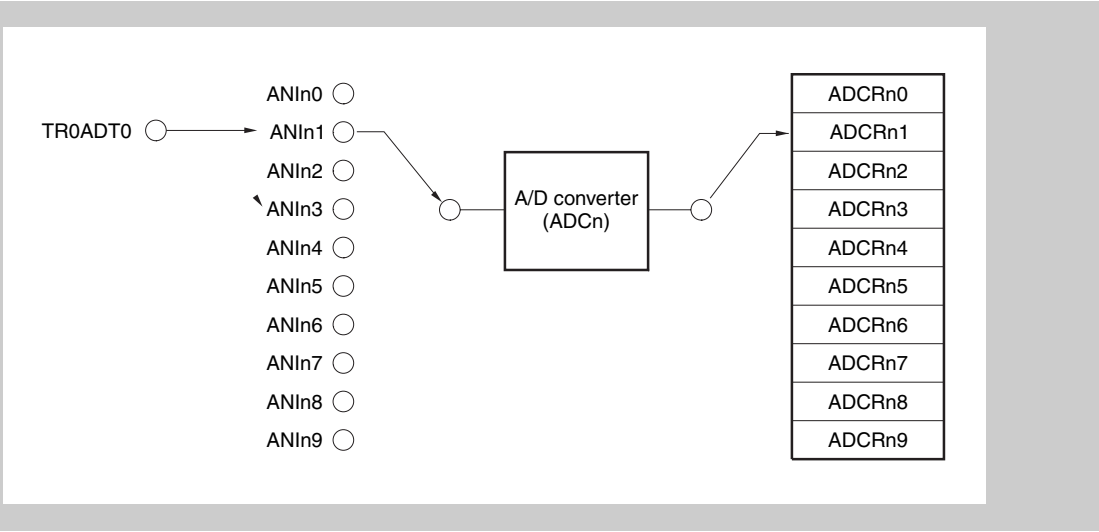


Figure 15-9 Example of 1-buffer mode operation (timer trigger select: 1 buffer) (ANIn1)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. The TS0ADT0 signal is generated
3. ANIn1 is A/D converted
4. The conversion result is stored in ADCRn1
5. The INTADn interrupt is generated

(2) 4-buffer mode operation (timer trigger select: 4 buffers)

In this mode, A/D conversion of one analog input is executed four times, and the results are stored in the ADCRnm register.

One analog input is A/D converted four times using the timer event signals as a trigger, and the results are stored in four ADCRnm registers. The A/D conversion end interrupt (INTADn) is generated when the four A/D conversions end.

After conversion has finished, the next conversion is repeated when a timer event signal is generated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Table 15-10 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	ANI0 to ANI3	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)
	ANI4 to ANI7	ADCRn4 (1st time)
		ADCRn5 (2nd time)
		ADCRn6 (3rd time)
		ADCRn7 (4th time)
	ANIn8, ANIn9	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)

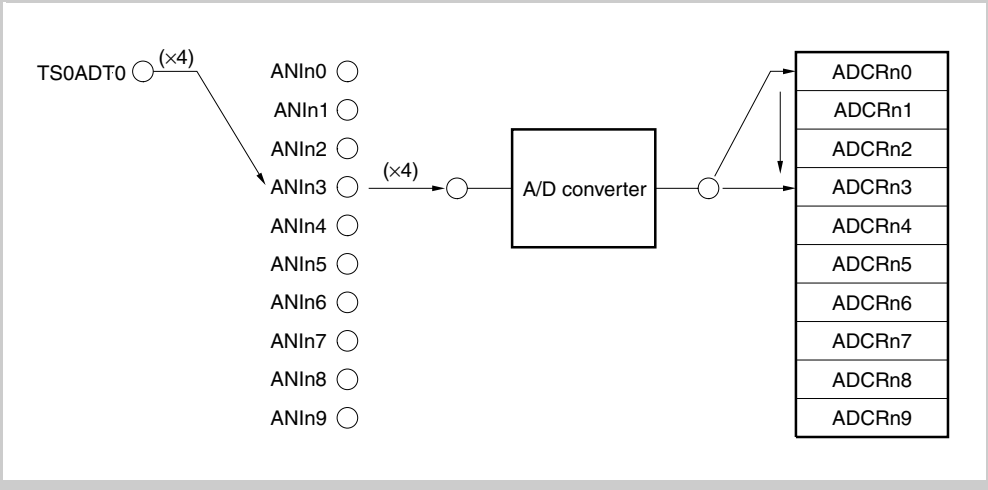


Figure 15-10 Example of 4-buffer mode operation (timer trigger select: 4 buffers) (ANIn3)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. TheTS0ADT0 signal is generated
3. ANIn3 is A/D converted
4. The conversion result is stored in ADCR0
5. ANIn3 is A/D converted
6. The conversion result is stored in ADCR1
7. ANIn3 is A/D converted
8. The conversion result is stored in ADCR2
9. ANIn3 is A/D converted
10. The conversion result is stored in ADCR3
11. The INTADn interrupt is generated

15.6.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin and are A/D converted the specified number of times using the timer event signal as a trigger.

The result of conversion is stored in the ADCRnm register corresponding to the analog input. When all the specified analog input signals have been converted, an A/D conversion end interrupt (INTADn) occurs.

After conversion has finished, the A/D Converter waits for a trigger unless the 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 15-11 Correspondence between analog input pins and ADCRnm register (scan mode (timer trigger scan))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

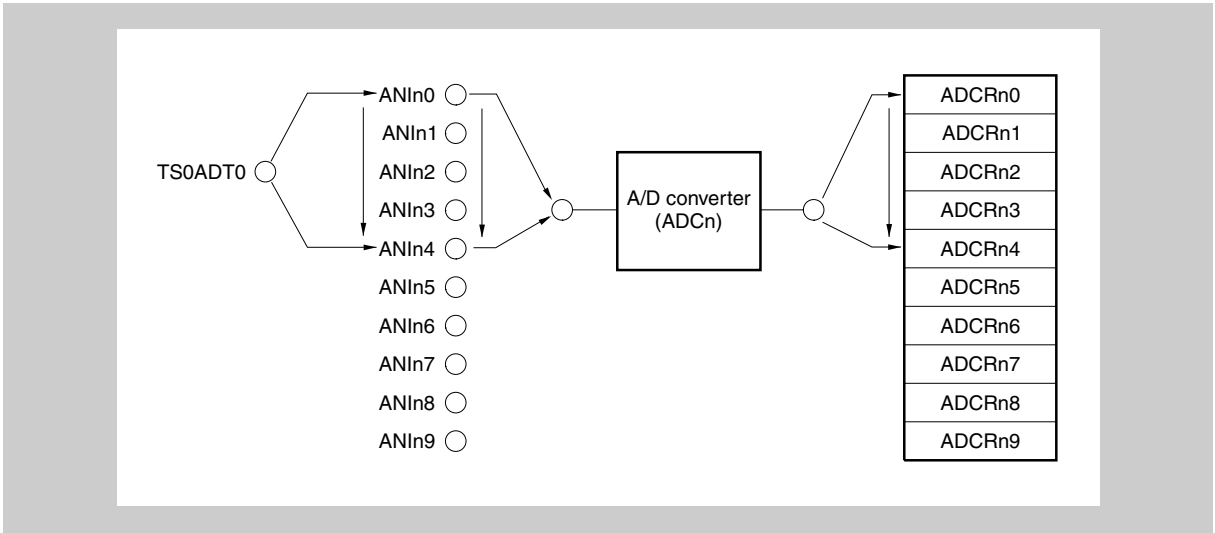


Figure 15-11 Example of scan mode operation (timer trigger scan) (ANIn0 to ANIn4)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. The TS0ADT0 signal is generated
3. ANIn0 is A/D converted
4. The conversion result is stored in ADCRn0
5. ANIn1 is A/D converted
6. The conversion result is stored in ADCRn1
7. ANIn2 is A/D converted
8. The conversion result is stored in ADCRn2
9. ANIn3 is A/D converted
10. The conversion result is stored in ADCRn3
11. ANIn4 is A/D converted
12. The conversion result is stored in ADCRn4
13. The INTADn interrupt is generated

15.7 Operation in External Trigger Mode

In this mode, the conversion timing of the analog signals input to the ANIn0 to ANIn9 pins is specified by the ADTRGn pin.

Detection of the valid edge at the ADTRGn input pin is specified by using the EGAn1 and EGAn0 bits of the ADMn1 register.

When the ADCEn bit of the ADMn0 register 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 (ADCSn bit of the ADMn0 register = 1). When the converter has ended conversion, it waits for the external trigger again (ADCSn bit = 0).

If the valid edge is detected at the ADTRGn pin during conversion, conversion is executed from the beginning again.

If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and executed from the beginning again.

15.7.1 Select mode operation

In this mode, one analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, there are two select modes: 1-buffer mode and 4-buffer mode, according to the storing method of the conversion results.

(1) 1-buffer mode (external trigger select: 1 buffer)

In this mode, one analog input is A/D converted using the ADTRGn signal as a trigger. The conversion results are stored in one ADCRnm register. The analog input and the A/D conversion results register correspond one to one. The A/D conversion end interrupt (INTADn) is generated for each A/D conversion, and A/D conversion is stopped.

Table 15-12 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 ADCEn bit of the ADMn0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRGn pin.

This mode is most appropriate for applications in which the results are read after each A/D conversion.

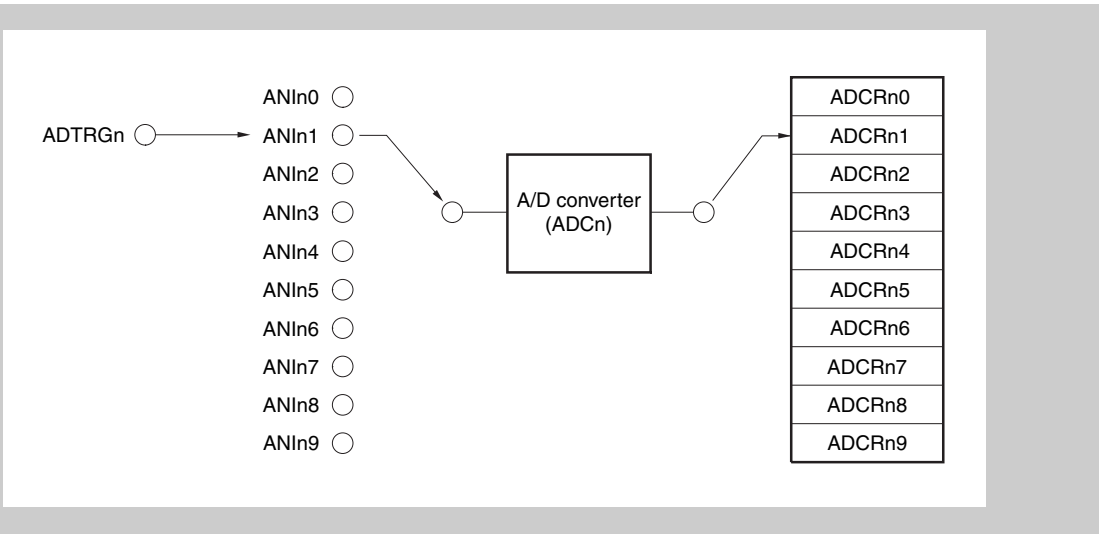


Figure 15-12 Example of 1-buffer mode operation (external trigger select: 1 buffer) (ANIn1)

1. The ADCEn bit of ADMn0 is set to 1 (enable)
2. The external trigger is generated
3. ANIn1 is A/D converted
4. The conversion result is stored in ADCRn1
5. The INTADn interrupt is generated

(2) 4-buffer mode (external trigger select: 4 buffers)

In this mode, one analog input is A/D converted four times using the ADTRGn signal as a trigger and the results are stored in the ADCRnm register. The A/D conversion end interrupt (INTADn) is generated and A/D conversion is stopped after the 4th A/D conversion.

Table 15-13 Correspondence between analog input pins and ADCRnm register (external trigger select: 4 buffers))

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANI0 to ANI3	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)
	ANI4 to ANI7	ADCRn4 (1st time)
		ADCRn5 (2nd time)
		ADCRn6 (3rd time)
		ADCRn7 (4th time)
	ANIn8, ANIn9	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)

While the ADCEn bit of the ADMn0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRGn pin.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

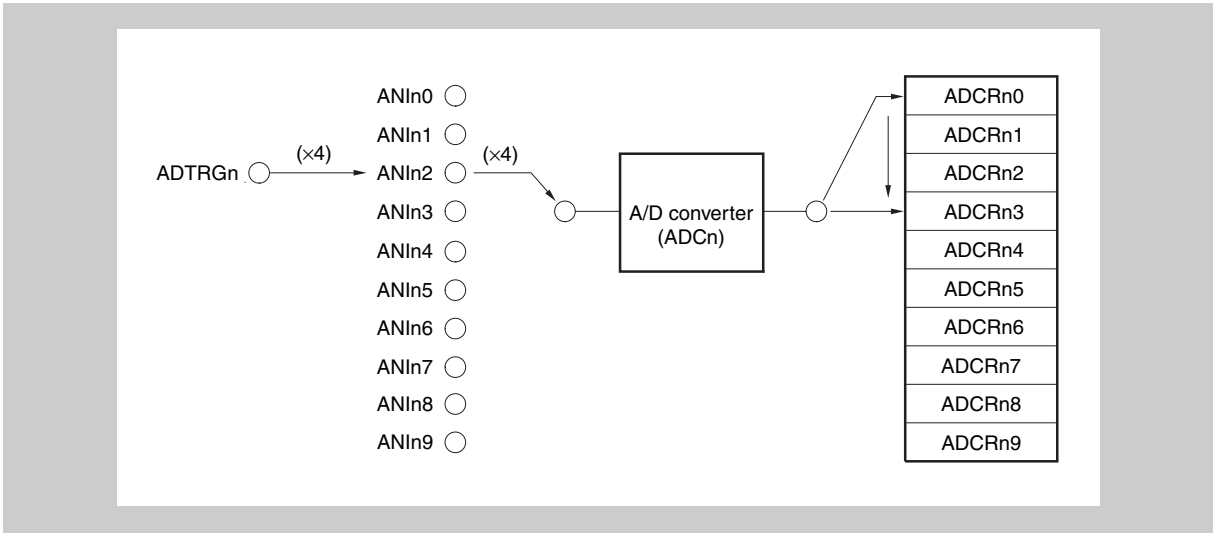


Figure 15-13 Example of 4-buffer mode operation (external trigger select: 4 buffers) (ANIn2)

1. The ADCEn bit of ADMn0 register is set to 1 (enable)
2. The external trigger is generated
3. ANIn2 is A/D converted
4. The conversion result is stored in ADCRn0
5. ANIn2 is A/D converted

6. The conversion result is stored in ADCRn1
7. ANIn2 is A/D converted
8. The conversion result is stored in ADCRn2
9. ANIn2 is A/D converted
10. The conversion result is stored in ADCRn3
11. The INTAD interrupt is generated

15.7.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin using the ADTRGn signal as a trigger, and A/D converted. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input ANInm.

When conversion of all the specified analog inputs has ended, the A/D conversion end interrupt (INTADn) is generated.n Unless the ADCE bit of the ADMn0 register 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 15-14 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 ADCEn bit of the ADMn0 register is 1, A/D conversion is started again.

This is most appropriate for applications in which multiple analog inputs are constantly monitored.

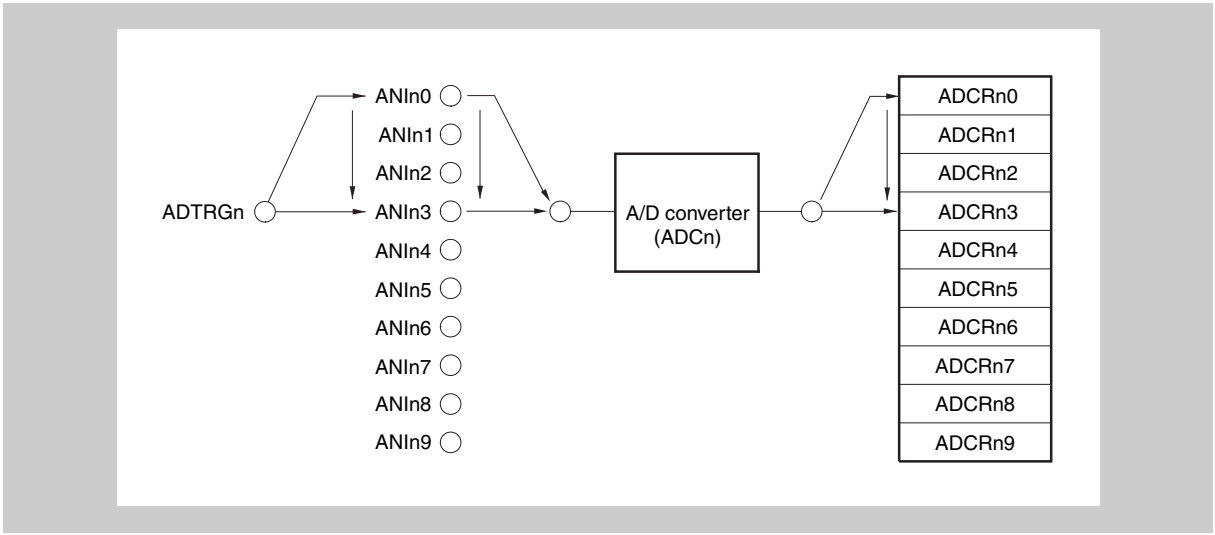


Figure 15-14 Example of scan mode operation (external trigger scan) (ANIn0 to ANIn3)

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

15.8 Precautions

(1) Stopping conversion operation

When the ADCEn bit of the ADMn0 register 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 FRnm 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, the ADMn0, ADMn1, and ADMn2 registers as well as the ADCRnm register hold the value.

(4) Input range of ANIn0 to ANIn9

Use the input voltage at ANIn0 to ANIn9 within the specified range. If a voltage outside the range of AV_{REF} is input to any of these pins (even within the absolute maximum rating range), the converted value of the channel is undefined. In addition, the converted value of the other channels may also be affected.

(5) Conflicts

- **Conflict between writing A/D conversion result registers (ADCRnm, ADCRnmH) at end of conversion and reading ADCRnm and ADCRnmH registers by instruction**

  Reading the ADCRnm and ADCRnmH registers takes precedence. After these registers have been read, the new conversion result is written to the ADCRnm and ADCRnmH registers.

- **Conflict between writing ADCRnm and ADCRnmH at end of conversion and input of external trigger signal**

  The external trigger signal is not accepted during A/D conversion. Therefore, it is not accepted while ADCRnm and ADCRnmH are being written.

- **Conflict between writing ADCRnm and ADCRnmH at end of conversion and writing ADMn1 or ADMn2 register**

If ADMn1 or ADMn2 register is written immediately after ADCRnm and ADCRnmH have been written on completion of A/D conversion, the conversion result is written to the ADCRnm and ADCRnmH registers, but the A/D conversion end interrupt (INTADn) may not occur depending on the timing.

(6) Unexpected AD conversion start in timer trigger mode/external trigger mode

Description If the ADC is operated in timer trigger mode (ADMn1.TRGn[1:0] = 01_B) or in external trigger mode (ADMn1.TRGn[1:0] = 10_B) the ADC may start conversion autonomously on changing the AD conversion enable bit (ADMn0.ADCEn) from 0 to 1 before occurence of the first configured trigger event (timer trigger or external trigger).

Workaround Whenever changing the ADC enable bit (ADMn0.ADCEn) from 0 to 1 write two times in sequence 1 to the ADC enable bit.

Make sure that no maskable or non-maskable interrupt may occur between the 1st and 2nd write instruction. Use the processor status word to temporarily disable non-maskable interrupts (PSW.NP = 1) or non-maskable interrupts (PSW.ID = 1) if interrupt occurence cannot be excluded.

- Save PSW (optional)
- Disable INT and/or NMI (optional)
- 1st write 1 to ADMn0.ADCEn (byte or bit access)
- 2nd write 1 to ADMn0.ADCEn (byte or bit access)
- Restore PSW (optional)

Example (C - GHS Compiler):

```
unsigned int tmp;

tmp=__GETSR();          // Save PSW
__SETSR(0x000000a0|tmp);// Disable NMI and INT in PSW
ADM00 = 0x80;           // 1st write to ADMn0
ADM00 = 0x80;           // 2nd write to ADMn0
__SETSR(tmp);           // Restore PSW
```

Example (ASM):

```
stsr PSW, r14           -- Save PSW
mov 0x000000a0, r13
or r14, r13
ldsr r13, PSW           -- Disable NMI and INT in PSW
set1 7, ADM00[r0]       -- 1st write to ADMn0
set1 7, ADM00[r0]       -- 2nd write to ADMn0
ldsr r14, PSW           -- Restore PSW
```

Note This workaround is effective only in case code is executed from internal flash memory. While debugging do not set a breakpoint on the 2nd write instruction to ADMn0.

Chapter 16 16-Bit Timer/Event Counter AA (TAA)

The microcontroller has ten instances of this 16-bit timer/event counter AA, TAA0 to TAA9. The timer is upward compatible to Timer P used in various other devices of the V850E and the V850ES family, including V850E/PHO3. It offers new additional features for enhanced output control and for 32-bit capture.

Throughout this chapter, the individual instances of Timer AA are identified by “n” (n = 0 to 9), for example, TAAnCTL0 for the TAAn control register 0.

16.1 Features

Timer AA (TAA) is a 16-bit timer/event counter provided with general-purpose functions.

TAA can perform the following operations:

- 16-bit-accuracy PWM output timer
- Interval timer
- External event counter function
- Timer synchronised operation function
- One-shot pulse output
- Pulse interval and frequency measurement counter
- Free running function
- External trigger pulse output function

16.2 Function Outline

- Capture trigger input signal × 2
- External trigger input signal × 1
- Clock select × 8
- External event count input × 1
- Readable counter × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer output (TOAAn0, TOAAn1) × 2
- 32-bit capture by cascading two timer AA (TAA0+TAAx, TAA4+TAAy, TAA8+TAA9).

Note x = 1 to 3 and y = 5 to 7

16.3 Configuration

TAA includes the following hardware.

Table 16-1 Timer TAA registers and external connections

Item	Configuration
Timer register	16-bit counter
Registers	• TAAn timer capture/compare registers 0, 1 (TAAnCCR0, TAAnCCR1) • TAAn timer read buffer register (TAAnCNT) • CCR0 buffer register, CCR1 buffer register
Input selection registers	• TAAn input selection registers (TAAIC0 to TAAIC2)
Timer output	TOAAx0, TOAAn1
Timer input	TIAAn0, TIAAx1, TTRGAAx, TEVTAAn
Control registers	• TAAn control registers 0, 1 (TAAnCTL0, TAAnCTL1) • TAAn I/O control registers 0 to 3 (TAAnIOC0, TAAnIOC2, TAAnIOC4) • TAAn option registers 0, 1 (TAAnOPT0, TAAnOPT1) • TAAn Specific overflow value control register (TAAnSVC)

Note n = 0 to 9

Timer AA (TAA) pins are alternate function of port pins. For how to set the alternate function, refer to the description of the registers in *"Pin Functions" on page 1*.

The block diagram of the timer TAA is shown below. *Figure 16-2* and *Figure 16-3* show the block diagrams of the input circuits of the different timers TAAn.

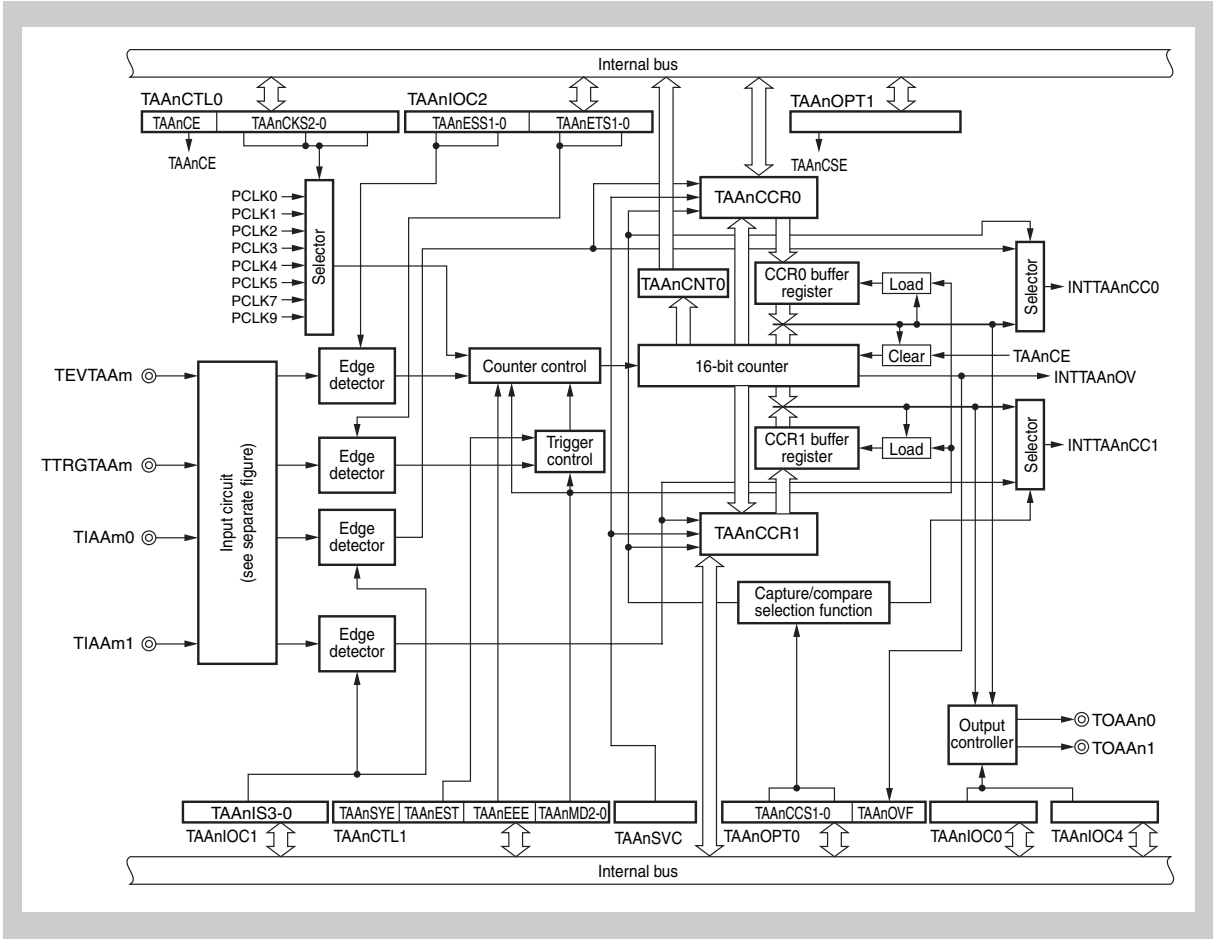


Figure 16-1 Block Diagram of Timer AA

Note 1. n = 0 to 9

2. PCLKn: peripheral clocks (refer to *"Clock Generator" on page 179*):

PCLK	0	1	2	3	4	5	7	9
f [MHz]	32	16	8	4	2	1	0.25	0.0625

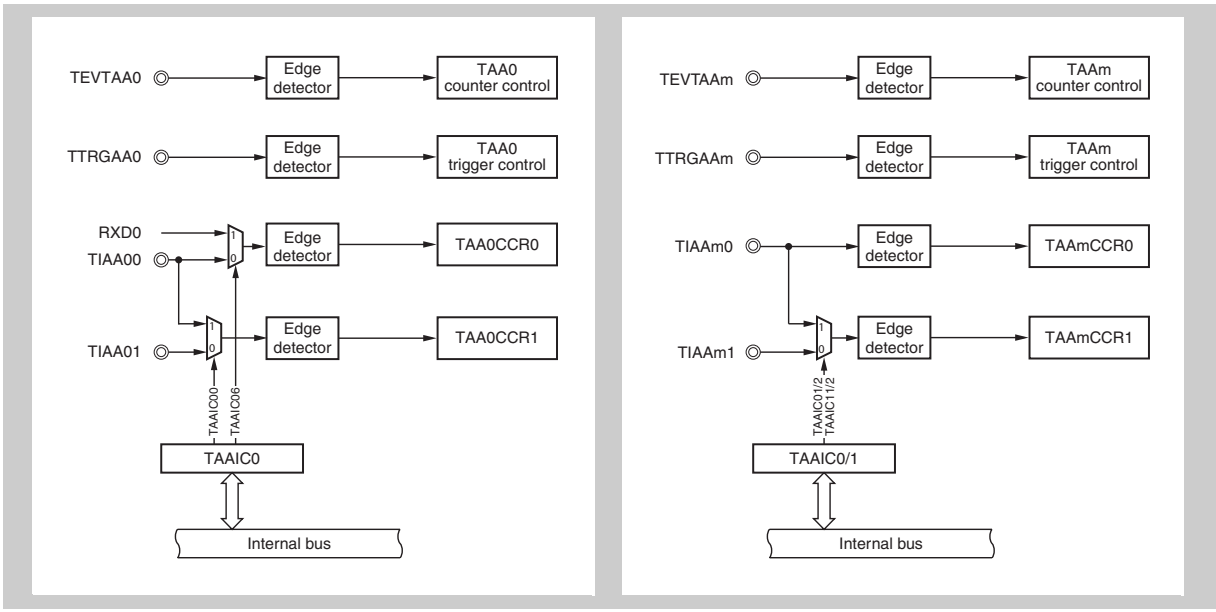


Figure 16-2 Input circuit of TAA0 (left) and TAAm with m = 1, 2, 5, 6 (right)

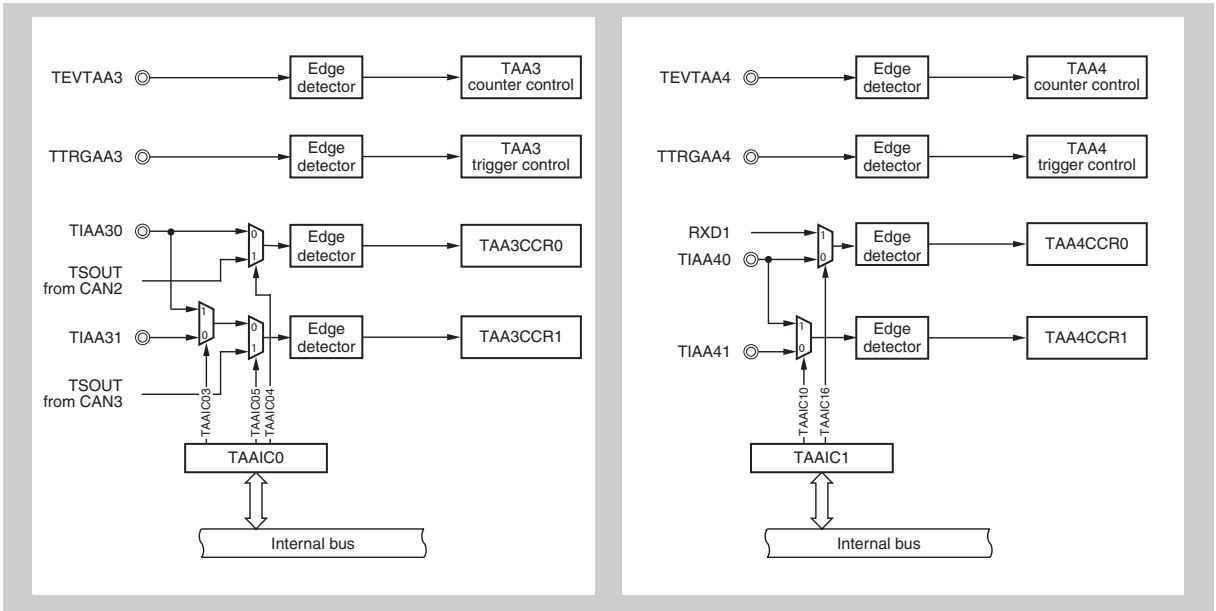


Figure 16-3 Input circuit of TAA3 (left) and TAA4 (right)

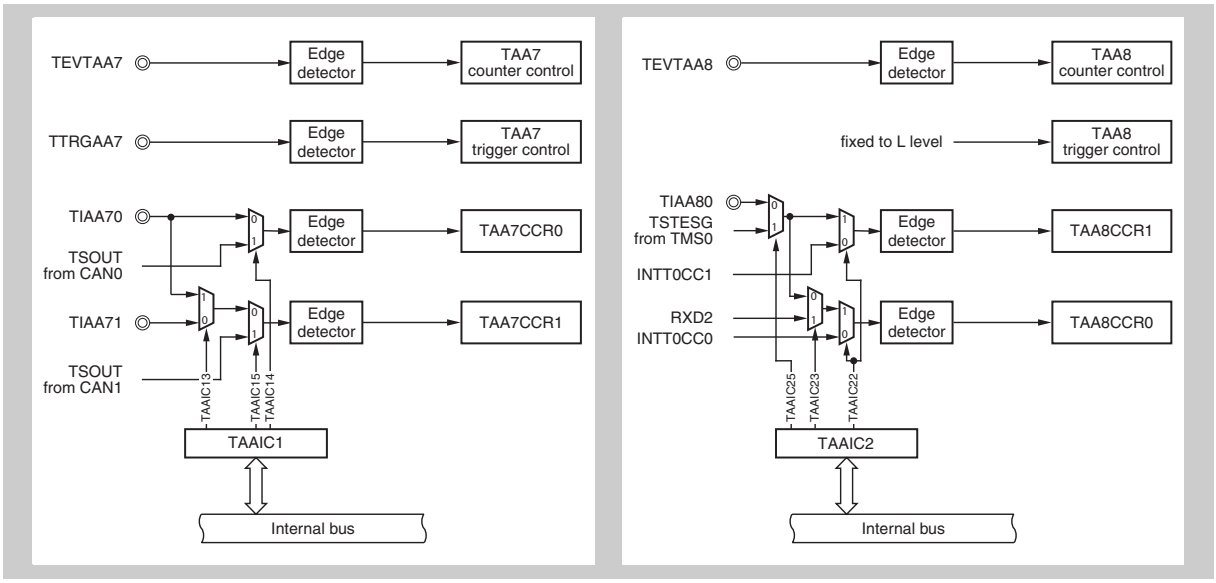


Figure 16-4 Input circuit of TAA7 (left) and TAA8 (right)

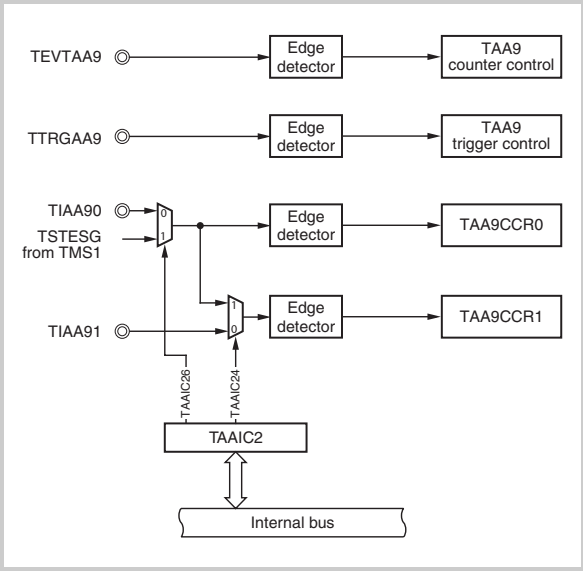


Figure 16-5 Input circuit of TAA9

(1) TAAnCCR0 - TAA capture/compare register 0

The TAAnCCR0 register is a 16-bit register that operates either as capture register or as a compare register.

In free-running mode, this register can be used as a capture register or as a compare register specified by bit TAAnOPT0.TAAnCCS0.

In the pulse width measurement mode, this register can be used only as a capture register (the compare function cannot be used.)

In all modes other than free-running mode and pulse width measurement mode, this register is used as a compare register.

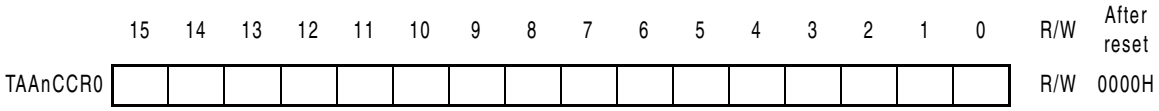
After a $\overline{\text{RESET}}$, TAAnCCR0 register default status is compare register.

$\overline{\text{RESET}}$ input clears this register to 0000H.

TAAnCCR0 is also cleared to 0000H if the internal operation clock is disabled by TAAnCTL0.TAAnCE = 0.

Caution When external event counter mode is used, do not set TAAnCCR0 register to 0000H.

Address: TAA0CCR0 FFFFF606H, TAA1CCR0 FFFFF616H,
TAA2CCR0 FFFFF626H, TAA3CCR0 FFFFF636H,
TAA4CCR0 FFFFF646H, TAA5CCR0 FFFFF656H,
TAA6CCR0 FFFFF666H, TAA7CCR0 FFFFF676H,
TAA8CCR0 FFFFF686H, TAA9CCR0 FFFFF6B6H



- When used as a compare register, TAAnCCR0 can be rewritten when TAAnCE = 1, as shown below:

TAA operation mode	TAAnCCR0 register writing method
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Any time write
Pulse width measurement mode	Not applicable (used as capture register)

- When used as capture register, the count value is stored in TAAnCCR0 upon capture trigger (TIAAn0) input edge detection.

Note The value of TAAnCCR0 register is read when TAAnCE bit of TAAn control register 0 (TAAnCTL0) equals 1.

(2) TAAnCCR1 - TAA capture/compare register 1

The TAAnCCR1 register is a 16-bit register that operates either both as a capture register or as a compare register.

In free-running mode, this register can be used as a capture register or as a compare register specified by bit TAAnOPT0.TAAnCCS1.

In the pulse width measurement mode, this register can be used only as a capture register (the compare function cannot be used.)

In all modes other than free-running mode and pulse width measurement mode, this register is used as a compare register.

After $\overline{\text{RESET}}$, TAAnCCR1 register default status is compare register.

$\overline{\text{RESET}}$ input clears this register to 0000H.

TAAnCCR0 is also cleared to 0000H if the internal operation clock is disabled by TAAnCTL0.TAAnCE = 0.

Caution When external event counter mode is used, do not set TAAnCCR1 register to 0000H.

Address: TAA0CCR1 FFFFF608H, TAA1CCR1 FFFFF618H,
TAA2CCR1 FFFFF628H, TAA3CCR1 FFFFF638H,
TAA4CCR1 FFFFF648H, TAA5CCR1 FFFFF658H,
TAA6CCR1 FFFFF668H, TAA7CCR1 FFFFF678H,
TAA8CCR1 FFFFF688H, TAA9CCR1 FFFFF6B8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
TAAnCCR1																	R/W	0000H

- When used as a compare register TAAnCCR1 can be rewritten when TAAnCE = 1, as below mentioned.

TAA operation mode	TAAnCCR1 register writing method
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Any time write
Pulse width measurement mode	Not applicable (used as capture register)

- When used as a capture register

  Count value is stored in TAAnCCR1 upon capture trigger (TIAAn1) input edge detection.

Note The value of TAAnCCR1 register is read when TAAnCE bit of TAAn control register 0 (TAAnCTL0) equals 1.

(3) TAAnCNT - TAA counter read buffer register

TAAnCNT register is a read buffer register that can read 16-bit counter values.

This register is read-only, using a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 0000H.

When TAAnCE bit of TAAnCTL0 equals 0, 0000H is read from this register.

The value of the register is read when TAAnCE bit = 1.

Address: TAA0CNT FFFFF60AH, TAA1CNT FFFFF61AH,
TAA2CNT FFFFF62AH, TAA3CNT FFFFF63AH,
TAA4CNT FFFFF64AH, TAA5CNT FFFFF65AH,
TAA6CNT FFFFF66AH, TAA7CNT FFFFF67AH,
TAA8CNT FFFFF68AH, TAA9CNT FFFFF6BAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W	After reset
TAAnCNT																	R	0000H

16.4 Input Selection Registers

(1) TAAIC0 - Timer input control register 0

The TAAIC0 register is an 8-bit register that controls the external input pin source of the capture register 1 of TAA0 to TAA3.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TAAIC0	0	TAAIC06	TAAIC05	TAAIC04	TAAIC03	TAAIC02	TAAIC01	TAAIC00	FFFFF6F0H	R/W	00H

TAAIC06	TAA0CCR0 Register Capture Source Input Selection
0	Capture source input is pin P10/TIAA00/TEVTAA1
1	Capture source input is RXD0

TAAIC05	TAAIC03	TAA3CCR1 Register Capture Source Input Selection
0	0	Capture source input is pin P17/TIAA31/TEVTAA2
0	1	Capture source input is pin P16/TIAA30/TTRGAA2
1	0	Capture source input is CAN3 time trigger (TSOUT3)
1	1	

TAAIC04	TAA3CCR0 Register Capture Source Input Selection
0	Capture source input is pin P16/TIAA30/TTRGAA2
1	Capture source input is CAN2 time trigger (TSOUT2)

TAAIC02	TAA2CCR1 Register Capture Source Input Selection
0	Capture source input is pin P15/TIAA21/TTRGAA3
1	Capture source input is pin P14/TIAA20/TEVTAA3

TAAIC01	TAA1CCR1 Register Capture Source Input Selection
0	Capture source input is pin P13/TIAA11/TEVTAA0
1	Capture source input is pin P12/TIAA10/TTRGAA0

TAAIC00	TAA0CCR1 Register Capture Source Input Selection
0	Capture source input is pin P11/TIAA01/TTRGAA1
1	Capture source input is pin P10/TIAA00/TEVTAA1

Caution Do not change the timer input when the timer is in operation.

(2) TAAIC1 - Timer input control register 1

The TAAIC1 register is an 8-bit register that controls the external input pin source of the capture register 1 of TAA4 to TAA7, as well as the internal time trigger source from the CAN Controllers to both capture registers 0 and 1 of TAA7.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TAAIC1	0	TAAIC16	TAAIC15	TAAIC14	TAAIC13	TAAIC12	TAAIC11	TAAIC10	FFFFF6F1H	R/W	00H

TAAIC16	TAA4CCR0 Register Capture Source Input Selection
0	Capture source input is pin P20/TIAA40/TEVTAA5
1	Capture source input is RXD1

TAAIC15	TAAIC13	TAA7CCR1 Register Capture Source Input Selection
0	0	Capture source input is pin P27/TIAA71/ TEVTAA6
0	1	Capture source input is pin P26/TIAA70/ TTRGAA6
1	0	Capture source input is CAN1 time trigger (TSOUT1)
1	1	

TAAIC14	TAA7CCR0 Register Capture Source Input Selection
0	Capture source input is pin P26/TIAA70/ TTRGAA6
1	Capture source input is CAN0 time trigger (TSOUT0)

TAAIC12	TAA6CCR1 Register Capture Source Input Selection
0	Capture source input is pin P25/TIAA61/TTRGAA7
1	Capture source input is pin P24/TIAA60/TEVTAA7

TAAIC11	TAA5CCR1 Register Capture Source Input Selection
0	Capture source input is pin P23/TIAA51/TEVTAA4
1	Capture source input is pin P22/TIAA50/TTRGAA4

TAAIC10	TAA4CCR1 Register Capture Source Input Selection
0	Capture source input is pin P21/TIAA41/TTRGAA5
1	Capture source input is pin P20/TIAA40/TEVTAA5

Caution Do not change the timer input when the timer is in operation.

(3) TAAIC2 - Timer input control register 2

The TAAIC2 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMT0 and TMT1, as well as the internal source of both capture registers 0 and 1 of TAA8 and TAA9.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	R/W	After reset
TAAIC2	0	TAAIC26	TAAIC25	TAAIC24	TAAIC23	TAAIC22	TAAIC21	TAAIC20	FFFFF6F2H	R/W	00H

TAAIC26	TAAIC24	Capture Source Input Selection of	
		TAA9CCR0	TAA9CCR1
0	0	Pin P101/TIAA90/TTRGAA9	Pin P102/TIAA91/TEVTAA9
0	1		Pin P101/TIAA90/TTRGAA9
1	0	TSTESG signal from TMS1	Pin P102/TIAA91/TEVTAA9
1	1		TSTESG signal from TMS1

TAAIC22	TAAIC25	TAAIC23	Capture Source Input Selection of	
			TAA8CCR0	TAA8CCR1
0	X	X	INTT0CC0 signal from TMT0	INTT0CC1 signal from TMT0
1	0	0	Pin P100/TIAA80/TOAA81TEVTAA81	
	0	1	RXD2	Pin P100/TIAA80/TOAA81TEVTAA81
	1	0	TSTESG signal from TMS0	
	1	1	RXD2	TSTESG signal from TMS0

TAAIC21	TT1CCR1 Register Capture Source Input Selection
0	Capture source input is pin P74/TIT11/TEVTT0
1	Capture source input is pin P73/TIT10/TTRGT0

TAAIC20	TT1CCR1 Register Capture Source Input Selection
0	Capture source input is pin P71/TIT01/TTRGT1
1	Capture source input is pin P70/TIT00/TEVTT1

Caution Do not change the timer input when the timer is in operation.

16.5 Control Registers

(1) TAAnCTL0 - TAA control register 0

TAAn control register 0 is an 8-bit register that controls the operation of timer AA.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

The same value can always be written to the TAAnCTL0 register by software.

Address: TAA0CTL0 FFFFF600H, TAA1CTL0 FFFFF610H,
TAA2CTL0 FFFFF620H, TAA3CTL0 FFFFF630H,
TAA4CTL0 FFFFF640H, TAA5CTL0 FFFFF650H,
TAA6CTL0 FFFFF660H, TAA6CTL0 FFFFF670H,
TAA8CTL0 FFFFF680H, TAA7CTL0 FFFFF6B0H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
TAAnCTL0	TAAn CE	0	0	0	0	TAAn CKS2	TAAn CKS1	TAAn CKS0	R/W	00H

TAAnCE	Timer AAn operation control
0	Disable internal operating clock operation (TAAn is asynchronously reseted)
1	Enable internal operating clock operation
Internal operating clock control and TAAn asynchronous reset are performed with the TAAnCE bit. When TAAnCE bit is cleared to 0, the internal operating clock of TAAn stops (fixed to low level) and TAAn is reset asynchronously. When the TAAnCE bit is set to 1, the internal operating clock is enabled within 2 input clocks, and TAAn counts up.	

TAAnCKS2	TAAnCKS1	TAAnCKS0	Internal count clock selection
0	0	0	PCLK0
0	0	1	PCLK1
0	1	0	PCLK2
0	1	1	PCLK3
1	0	0	PCLK4
1	0	1	PCLK5
1	1	0	PCLK7
1	1	1	PCLK9

Caution Set bits TAAnCKS2 to TAAnCKS0 only when TAAnCE = 0.
When TAAnCE bit setting is changed from 0 to 1, TAAnCKS2 to TAAnCKS0 bits can be set simultaneously.

(2) TAAnCTL1 - TAA timer control register 1

TAAn control register 1 is an 8-bit register that controls the operation of timer AA.

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

16-bit access is not possible.

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0CTL1 FFFFF601H, TAA1CTL1 FFFFF611H,
TAA2CTL1 FFFFF621H, TAA3CTL1 FFFFF631H,
TAA4CTL1 FFFFF641H, TAA5CTL1 FFFFF651H,
TAA6CTL1 FFFFF661H, TAA7CTL1 FFFFF671H,
TAA8CTL1 FFFFF681H, TAA9CTL1 FFFFF6B1H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
TAAnCTL1	TAAn SYE	TAAn EST	TAAn EEE	0	0	TAAn MD2	TAAn MD1	TAAn MD0	R/W	00H

TAAnSYE	Tuned operation mode enable control
0	Independent operation mode (asynchronous operation mode)
1	Tuned operation mode (specification of slave operation) In this mode, timer AA can operate in synchronization with a master timer. **Master timer** / **Slave timer**: TAA0 / TAA1, TAA2, TAA3; TAA4 / TAA5, TAA6, TAA7 For the tuned operation mode, refer to *"Timer Synchronization Operation Function" on page 674*. **Caution:** Be sure to clear the TAA0SYE and TAA4SYE bits to 0 (master timer) and set the TAAmSYE bits (m = 1 to 3, 5 to 7) to 1 when respectively used as slave timers.

Caution In the synchronous operation mode, the master timer can be used only in the PWM mode, external trigger pulse output mode, pulse output mode, and free-running mode.
The slave timers can be used in the free-running mode only.
Setting the external event count mode, one-shot pulse mode, and pulse width measurement mode is prohibited.

TAAnEST	Software trigger control
0	No operation
1	In one-shot pulse mode: One-shot pulse software trigger
	In external trigger pulse output mode: Pulse output software trigger
The TAAnEST bit functions as a software trigger in the one-shot pulse mode or external trigger pulse output mode (this bit is invalid in any other mode). By setting TAAnEST to 1 when TAAnCE = 1, a software trigger is issued. Therefore, be sure to set TAAnEST to 1 after setting TAAnCE = 1. The TTRGAAx pin is used for an external trigger. The read value of the TAAnEST bit is always 0.	

TAAnEEE	Count clock selection
0	Use the internal clock (clock selected with TAAnCKS2 to TAAnCKS0 bits of TAAnCTL0 register)
1	Use external clock (TEVTAAn input edge)
The valid edge is specified with TAAnEES1 and TAAnEES0 bits when TAAnEEE bit = 1 (external clock TEVTAAn).	

TAAnMD2	TAAnMD1	TAAnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event counter mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse mode
1	0	0	PWM mode
1	0	1	Free-running mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Caution Set bits TAAnEEE and TAAnMD2 to TAAnMD0 when TAAnCE = 0. (The same value can be written when TAAnCE = 1.) The operation is not guaranteed when rewriting is performed when TAAnCE = 1. If rewriting was mistakenly performed, clear TAAnCE to 0 and then set the bits again.

Note n = 0 to 9; x = 0 to 7, 9

(3) TAAnIOC0 - TAA dedicated I/O control register 0

The TAAnIOC0 register is an 8-bit register that controls the timer output.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0IOC0 FFFFF602H, TAA1CTL0 FFFFF612H,
TAA2IOC0 FFFFF622H, TAA3IOC0 FFFFF632H,
TAA4IOC0 FFFFF642H, TAA5IOC0 FFFFF652H,
TAA6IOC0 FFFFF662H, TAA7IOC0 FFFFF672H,
TAA8IOC0 FFFFF682H, TAA9IOC0 FFFFF6B2H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
TAAnIOC0	0	0	0	0	TAAn OL1	TAAn OE1	TAAn OL0	TAAn OE0	R/W	00H

TAAnOLm	TOAAnm output level setting
0	Normal output
1	Inverted output
This bit can be used to invert the timer output	

TAAnOEm	TOAAnm output setting
0	Timer output is disabled (Low level and high level are output from TOAAnm pin when TAAnOL = 0 and TAAnOL = 1, respectively.)
1	Timer output is enabled (TOAAnm pin outputs pulses.)

Caution

1. Rewrite bits TAAnOLm and TAAnOEm when TAAnCE = 0 (the same value can be written when TAAnCE = 1.). If rewriting was mistakenly performed, clear TAAnCE to 0 and then set the bits again.
2. To enable the timer output, be sure to set the corresponding alternate-function pins TAAnIS3 to TAAnIS0 of the TAAnIOC1 register to “No edge detection” and invalidate the capture operation. Then set the corresponding alternate-function port to output mode.

Note m = 0, 1

(4) TAAnIOC1 - TAA dedicated I/O control register 1

The TAAnIOC1 register is an 8-bit register that controls the valid edge for the external input signals (TIAAn0 and TIAAn1).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0IOC1 FFFFF603H, TAA1IOC1 FFFFF613H,
TAA2IOC1 FFFFF623H, TAA3IOC1 FFFFF633H,
TAA4IOC1 FFFFF643H, TAA5IOC1 FFFFF653H,
TAA6IOC1 FFFFF663H, TAA7IOC1 FFFFF673H,
TAA8IOC1 FFFFF683H, TAA9IOC1 FFFFF6B3H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
TAAnIOC1	0	0	0	0	TAAn IS3	TAAn IS2	TAAn IS1	TAAn IS0	R/W	00H

TAAnIS3	TAAnIS2	Capture input (TIAAn1) valid edge setting
0	0	No edge detection (capture operation is invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TAAnIS1	TAAnIS0	Capture input (TIAAn0) valid edge detection
0	0	No edge detection (capture operation is invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Caution

1. Rewrite bits TAAnIS3 to TAAnIS0 when TAAnCE = 0 (the same value can be written when TAAnCE = 1.). If rewriting was mistakenly performed, clear TAAnCE to 0 and then set the bits again.
2. Bits TAAnIS3 to TAAnIS0 are valid only in the free-running capture mode and pulse width measurement mode. In all the other modes, capture operation is not performed.
3. If used as the capture input, be sure to set the corresponding alternate-function pins TAAnOE1 and TAAnOE0 of the TAAnIOC0 register to “Timer output is disabled” and set the capture input valid edge. Then set the corresponding alternate-function port to input mode.

(5) TAAnIOC2 - TAA I/O control register 2

The TAAnIOC2 register is an 8-bit register that controls the valid edge for external event count input signals (TEVTAAn) and external trigger input signal (TTRGAAx).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0IOC2 FFFFF604H, TAA1IOC2 FFFFF614H,
TAA2IOC2 FFFFF624H, TAA3IOC2 FFFFF634H,
TAA4IOC2 FFFFF644H, TAA5IOC2 FFFFF654H,
TAA6IOC2 FFFFF664H, TAA7IOC2 FFFFF674H,
TAA8IOC2 FFFFF684H, TAA9IOC2 FFFFF6B4H

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
TAAn IOC2	0	0	0	0	TAAn EES1	TAAn EES0	TAAn ETS1	TAAn ETS0	R/W	00H

TAAnEES1	TAAnEES0	External event count input valid edge setting (TEVTAAn)
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TAAnETS1	TAAnETS0	External trigger input valid edge detection (TTRGAAn)
0	0	No edge detection
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Caution

1. Rewrite TAAnEES1, TAAnEES0, TAAnETS1, and TAAnETS0 bits when TAAnCE = 0 (the same value can be written when TAAnCE = 1). If rewriting was mistakenly performed, clear TAAnCE to 0 and then set the bits again.
2. TAAnEES1 and TAAnEES0 bits are valid only when TAAnEEE = 1 or when the external event count mode has been set (TAAnCTL1.TAAnMD[2:0] = 001_B).
3. TAAnETS1 and TAAnETS0 bits are only valid when the external trigger pulse output mode or one-shot pulse mode is set (TAAnMD[2:0] = 010_B or 011_B).

Note n = 0 to 9; x = 0 to 7, 9

(6) TAAnIOC4 - TAA I/O control register 4

The TAAnIOC4 register is an 8-bit register that controls the output function of Timer AA.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0IOC4=FFFFF60CH, TAA1IOC4=FFFFF61CH,
TAA2IOC4=FFFFF62CH, TAA3IOC4=FFFFF63CH,
TAA4IOC4=FFFFF64CH, TAA5IOC4=FFFFF65CH,
TAA6IOC4=FFFFF66CH, TAA7IOC4=FFFFF67CH,
TAA8IOC4=FFFFF68CH, TAA9IOC4=FFFFF6BCH

Symbol	7	6	5	4	3	2	1	0	R/W	After reset
TAAnIOC4	0	0	0	0	TAAn OS1	TAAn OR1	TAAn OS0	TAAn OR0	R/W	00H

TAAnOS1	TAAnOR1	Toggle Control of TOAAn1
0	0	Standard operation.
0	1	Force output level to inactive at next toggle event
1	0	Force output level to active at next toggle event
1	1	Freeze current output level.

Note
1. After forcing the output level to either active or inactive, the TOAAn1 maintains this level (=no toggling afterwards) until the TAAnOS1 and TAAnOR1 are cleared to standard operation.
2. The forcing of an output level is executed at the time of the next upcoming toggle event, while the freeze becomes effective immediately.

TAAnOS0	TAAnOR0	Toggle Control of TOAAn0
0	0	Standard operation.
0	1	Force output level to inactive at next toggle event
1	0	Force output level to active at next toggle event
1	1	Freeze current output level.

Note
1. After forcing the output level to either active or inactive, the TOAAn1 maintains this level (=no toggling afterwards) until the TAAnOS1 and TAAnOR1 are cleared to standard operation.
2. The forcing of an output level is executed at the time of the next upcoming toggle event, while the freeze becomes effective immediately.

Caution TAAnIOC4 can only be used when interval mode or free-running compare mode is selected. For all other modes make to set the register to 00H.

(7) TAAnOPT0 - TAA option register 0

The TAAnOPT0 register is an 8-bit register used to set the capture/compare operation and detect overflow.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0OPT0 FFFFF605H, TAA1OPT0 FFFFF615H,
TAA2OPT0 FFFFF625H, TAA3OPT0 FFFFF635H,
TAA4OPT0 FFFFF645H, TAA5OPT0 FFFFF655H,
TAA6OPT0 FFFFF665H, TAA7OPT0 FFFFF675H,
TAA8OPT0 FFFFF685H, TAA9OPT0 FFFFF6B5H

Symbol	7	6	5	4	3	2	1	<0>	R/W	After reset
TAAnOPT0	0	0	TAAn CCS1	TAAn CCS0	0	0	0	TAAn OVF	R/W	00H

TAAnCCS1	TAAnCCR1 register capture/compare selection
0	Compare register selection
1	Capture register selection
The TAAnCCS1 bit setting is valid only in the free-running mode.	

TAAnCCS0	TAAnCCR0 register capture/compare selection
0	Compare register selection
1	Capture register selection
The TAAnCCS0 bit setting is valid only in the free-running mode.	

TAAnOVF	Timer AA overflow detection
Set (1)	Overflow occurrence
Reset (0)	TAAnOVF bit write or TAAnCE = 0

- The TAAnOVF bit is set when the 16-bit counter value overflows from FFFFH to 0000H in the free-running mode and the pulse width measurement mode.
- An interrupt request signal (INTTAAnOV) is generated as soon as TAAnOVF bit is set (1).
  The INTTAAnOV signal is not generated in any mode other than free-running mode and the pulse width measurement mode.
- The TAAnOVF bit is not cleared even when the TAAnOVF bit and the TAAnOPT0 register are read when TAAnOVF = 1.
- The TAAnOVF bit can be both read and written, but 1 cannot be written to the TAAnOVF bit from the CPU. Writing 1 has no influence on timer AA operation.

Caution Rewrite TAAnCCS1 and TAAnCCS0 bits when TAAnCE = 0 (the same value can be written when TAAnCE = 1.). If rewriting was mistakenly performed, clear TAAnCE to 0 and then set the bits again.

(8) TAAnOPT1 - TAA option register 1

The TAAnOPT1 register is an 8-bit register used to set the 32-bit capture mode by cascading two Timer AA.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA1OPT1=FFFFF61DH, TAA2OPT1=FFFFF62DH,
TAA3OPT1=FFFFF63DH, TAA5OPT1=FFFFF65DH,
TAA6OPT1=FFFFF66DH, TAA7OPT1=FFFFF67DH,
TAA9OPT1=FFFFF6BDH

Symbol	7	6	5	4	3	2	1	<0>	R/W	After reset
TAAnOPT1	TAAn CSE	0	0	0	0	0	0	0	R/W	00H

TAAnCSE	TAAnCCR1 register capture/compare selection
0	16-bit non-cascaded mode
1	Set 32-bit cascaded capture mode. Timer AAn becomes the upper 16-bit and slave. The master timer is TAAm with m=n-1.

Note
1. When setting TAAnCSE, the timer becomes the upper 16-bit of a 32-bit timer.
2. Cascading is only available for capture with free-running counter.
3. The following pairs of timers can be cascaded:
   - TAA0 and TAA1
     (TAA0 will become master and will hold the lower 16-bit value)
   - TAA2 and TAA3
     (TAA2 will become master and will hold the lower 16-bit value)

The table below shows the effects of the TAAnCSE flag on the timer operation:

	TAAnCSE=0	TAAnCSE=1
Operating clock	macro clock from clock tree	macro clock of TAAm
Count Enable	TAAnCE bit of TAAnCTL0	TAAmCE bit of TAAm
Count Clock	selected by TAAnCKS[2:0]	Counter overflow from TAAm
Capture Signal 0	TIAAn0 input with edge filter as selected by TAAnIS[1:0]	TIAAm0 with edge filter selected for TAAm
Capture Signal 1	TIAAn1 input with edge filter as selected by TAAnIS[3:2]	TIAAm1 with edge filter selected for TAAm
Capture Interrupt	INTTAAnCC0 or INTTAAnCC1	INTTAAmCC0 or INTTAAmCC1

n=1 or 3; m= (n-1).

For details on the 32-bit capture mode, please refer to *"32-bit Capture in free-running cascade mode" on page 664*.

(9) TAAnSVC - TAA specific overflow value control register

The TAAnSVC controls the setting of defined overflow values OVFV to the TAAn capture/compare registers in case of overflows.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Address: TAA0SVC FFFFF60EH, TAA1SVC FFFFF61EH,
TAA2SVC FFFFF62EH, TAA3SVC FFFFF63EH,
TAA4SVC FFFFF64EH, TAA5SVC FFFFF65EH,
TAA6SVC FFFFF66EH, TAA7SVC FFFFF67EH,
TAA8SVC FFFFF68EH, TAA9SVC FFFFF6BEH

Symbol	7	6	5	4	3	2	1	<0>	R/W	After reset
TAAnSVC	TAAn ENC1	TAAn CSE1	TAAn CSD1	TAAn CSR1	TAAn ENC0	TAAn CSE0	TAAn CSD0	TAAn CSR0	R/W	00H

TAAnENCm	TAAnCCRm register overflow value enable
0	Disable overflow value set
1	Enable overflow value set TAAnCCRm is overwritten upon INTAAnOV generation.

TAAnCSEm	TIAnm compare value selection
0	If TIAnm = L (0) during overflow: overflow value OVFV = 0xFFFF If TIAnm = H (1) during overflow: overflow value OVFV = 0x0000
1	If TIAnm = H (1) during overflow: overflow value OVFV = 0xFFFF If TIAnm = L (0) during overflow: overflow value OVFV = 0x0000
TAAnCSEm determines also the selection of the valid capture input trigger edge of the input signals TIAAnm in combination with TAAnIOC1.TAAnIS[3:2] and TAAnIOC1.TAAnIS[1:0] respectively. Upon an overflow event a comparsion is made between the level of TIAnm input signal and the TAAnCSEMm bit. The result of the comparison determines the overflow value: • if TIAnm = TAAnCSEm: overflow value OVFV = 0xFFFF (match) • if TIAnm = not TAAnCSEm: overflow value OVFV = 0x0000 (no match) The overflow value finally set to the TAAnCCRm register depends also on TAAnCSDm and TAAnCSRm.	

TAAnCSDm	TAAnCCRm overwrite selection
0	Overflow value is set to TAAnCCRm in case of an overflow event, independent of TIAnm level and TAAnCSEn.
1	Overflow value, set to TAAnCCRm in case of an overflow event, depends on level of TIAnm and TAAnCSEn: • if TIAnm = TAAnCSEm: overflow value OVFV set to TAAnCCRm • if TIAnm = not(TAAnCSEm): TAAnCCRm remains unchanged
By use TAAnCSDm overwriting of TAAnCCRm by an overflow value OVFV can be prohibited, if TIAnm = not(TAAnCSEm).	

TAAnCSRm	Overflow value OVFV inversion control
0	Overflow value OVFV is not inverted.
1	Overflow value OVFV is inverted.
By use of TAAnCSRm the overflow value OVFV written to TAAnCCRm can be inverted.	

Caution

1. Write to the TAAnSVC register is only permitted, when TAAnCTL0.TAAnCE = 0.
2. The capture inputs TIAn0 and TIAn1 of TAAn must be connected externally, so that both inputs get the same signal.

Overwriting of the TAAnCCRm register coincide with the generation of the overflow interrupt INTTAAnCCm and setting of the overflow flag TAAnOPT0.TAAnOVF.

The following table summarizes the values set to the TAAnCCRm register under the various control settings of TAAnSVC, in case the overflow setting function is enabled for capture/compare channel m by TAAnSVC.TAAnENCm = 1.

Valid edge specification

The valid edges of the capture trigger inputs TIAAnm are determined by TAAnSVC.TAAnCSE0 and TAAnIOC1.TAAnIS[3:2], respectively TAAnIOC1.TAAnIS[1:0]. These bits must be set according to *Table 16-2*.

Table 16-2 Overflow value setting conditions (1/2)

TAAnCSE0 / TAAnCSE1	TAAnCSD0 / TAAnCSD1	TAAnCSR0 / TAAnCSR1	TIAn0 / TIAn1	TAAnIS1 / TAAnIS3	TAAnIS0 / TAAnIS2	TAAnCCR0 / TAAnCCR1	Remark
0	0	0	L	X	1	0xFFFF	TIAnm = TAAnCSEm
			H	X	1	0x0000	$\overline{\text{TIAnm}}$ = TAAnCSEm
		1	L	X	1	0x0000	reverse of above
			H	X	1	0xFFFF	
	1	0	L	X	1	0xFFFF	TIAnm = TAAnCSEm
			H	X	1	no change	$\overline{\text{TIAnm}}$ = TAAnCSEm
		1	L	X	1	no change	reverse of above
			H	X	1	0xFFFF	

Table 16-2 Overflow value setting conditions (2/2)

TAAnCSE0 / TAAnCSE1	TAAnCSD0 / TAAnCSD1	TAAnCSR0 / TAAnCSR1	TIAn0 / TIAn1	TAAnIS1 / TAAnIS3	TAAnIS0 / TAAnIS2	TAAnCCR0 / TAAnCCR1	Remark
1	0	0	L	1	X	0x0000	$\overline{\text{TIAnm}}$ = $\overline{\text{TAAnCSEm}}$
			H	1	X	0xFFFF	TIAnm = TAAnCSEm
		1	L	1	X	0xFFFF	reverse of above
			H	1	X	0x0000	
	1	0	L	1	X	no change	$\overline{\text{TIAnm}}$ = $\overline{\text{TAAnCSEm}}$
			H	1	X	0xFFFF	TIAnm = TAAnCSEm
		1	L	1	X	0xFFFF	reverse of above
			H	1	X	no change	

The following flowchart illustrates how the final content of the TAAnCCRm is evaluated.

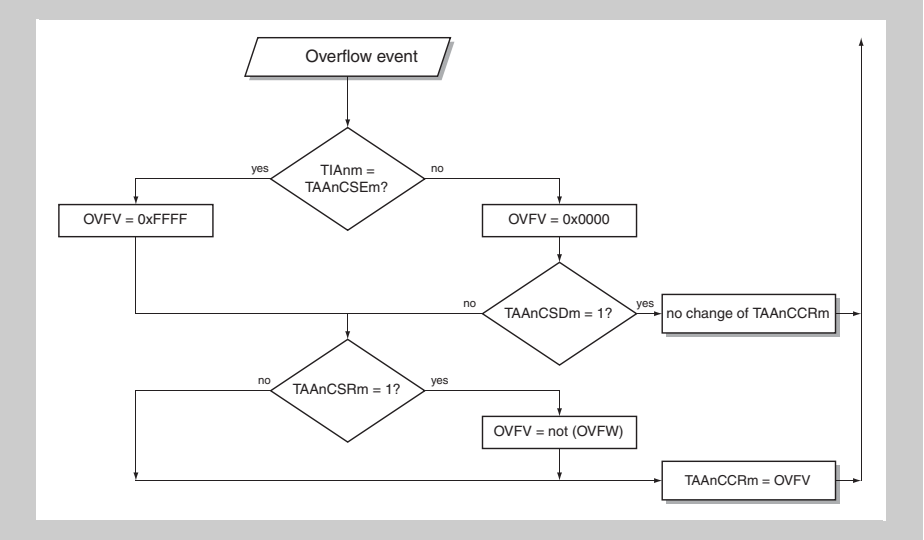


Figure 16-6 Evaluation of overflow value OVFV

For further information concerning the overflow value function refer to *"Specific TAAnCCRm overflow value write" on page 669*.

16.6 Operation

Timer AA can perform the following operations when not in cascade mode:

Operation	TAAnEST Software trigger bit	TIAAn0 External trigger input	TAAnEEE Count clock selection	Capture/Compare Selection	Compare Write
Interval timer mode	Invalid	Invalid	Internal/TIAAn0 pin	Compare only	Any time write
External event counter mode	Invalid	Invalid	External only	Compare only	Any time write
External trigger pulse output mode[Note]	Valid	Valid	Internal only	Compare only	Reload
One-shot pulse output mode[Note]	Valid	Valid	Internal only	Compare only	Any time write
PWM mode	Invalid	Invalid	Internal/TIAAn0 pin	Compare only	Reload
Free-running mode	Invalid	Invalid	Internal/TIAAn0 pin	Capture/compare selectable	Any time write
Pulse width measurement mode[Note]	Invalid	Invalid	Internal only	Capture only	Not applicable

Note To use the external trigger pulse output mode, one-shot pulse mode, or pulse width measurement mode, select a count clock by clearing the TAAnEEE bit of the TAAnCTL1 register to 0.

16.6.1 Anytime write and reload

TAAnCCR0 and TAAnCCR1 register rewrite is possible for timer AA during timer operation (TAAnCE = 1), but the write method (any time write, reload) differs depending on the mode.

(1) Anytime write

When data is written to the TAAnCCRm register during timer operation, it is transferred at any time to CCRm buffer register and used as the 16-bit counter comparison value.

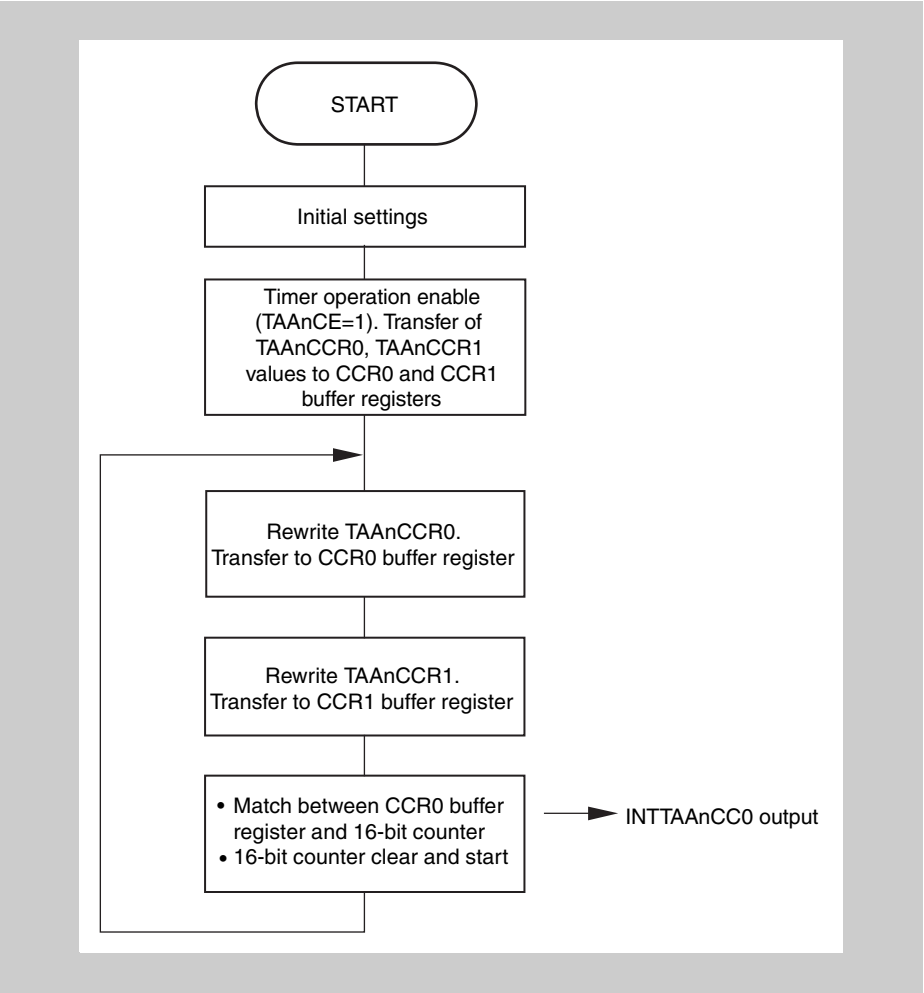


Figure 16-7 Flowchart of basic operation for anytime write

Note 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0 to 9; m = 0, 1

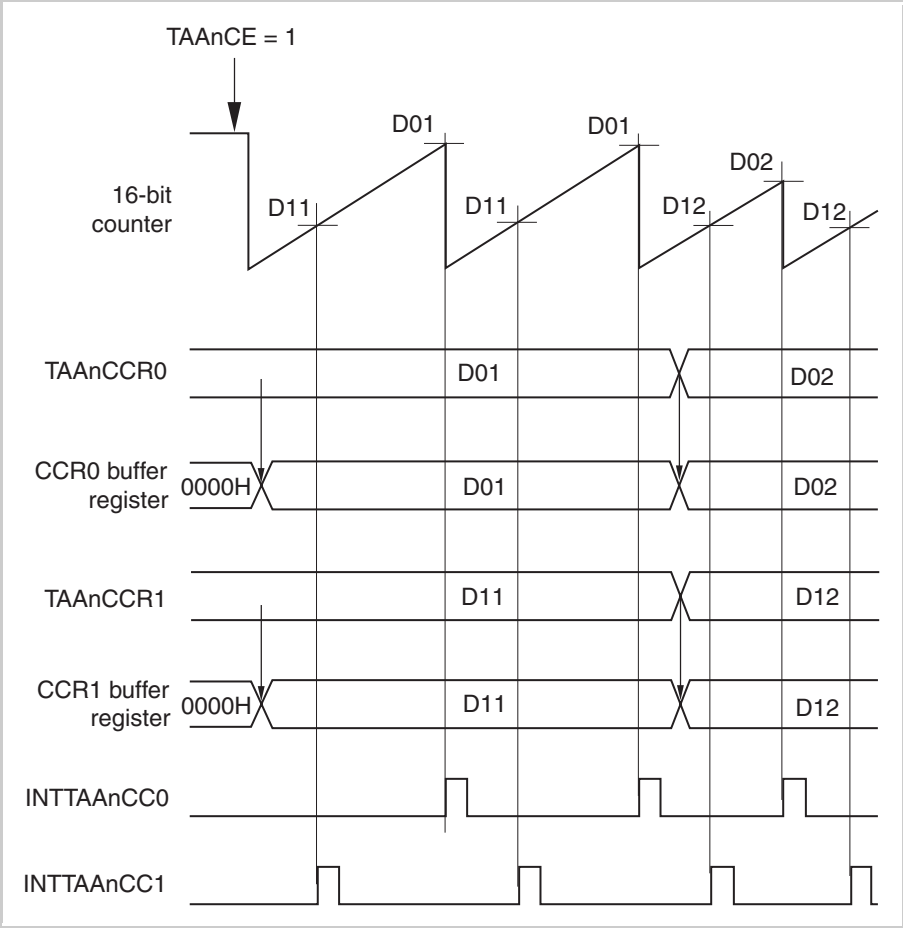


Figure 16-8 Timing diagram for anytime write

D01, D02: Setting values of TAAnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TAAnCCR1 register (0000H to FFFFH)

The above timing chart illustrates an example of the operation in the interval timer mode.

(2) Reload

When data is written to the TAAnCCR0 and TAAnCCR1 registers during timer operation, it is compared with the value of the 16-bit counter via the CCRm buffer register. The values of the TAAnCCR0 and TAAnCCR1 registers can be rewritten when TAAnCE = 1.

So that the set values of the TAAnCCR0 and TAAnCCR1 registers are compared with the value of the 16-bit counter (the set values are reloaded to the CCRm buffer register), the value of the TAAnCCR0 register must be rewritten and then a value must be written to the TAAnCCR1 register before the value of the 16-bit counter matches the value of TAAnCCR0. When the value of the TAAnCCR0 register matches the value of the 16-bit counter, the values of the TAAnCCR0 and TAAnCCR1 registers are reloaded.

Whether the next reload timing is made valid or not is controlled by writing to the TAAnCCR1 register. Therefore, write the same value to the TAAnCCR1 register when it is necessary to rewrite the value of only the TAAnCCR0 register.

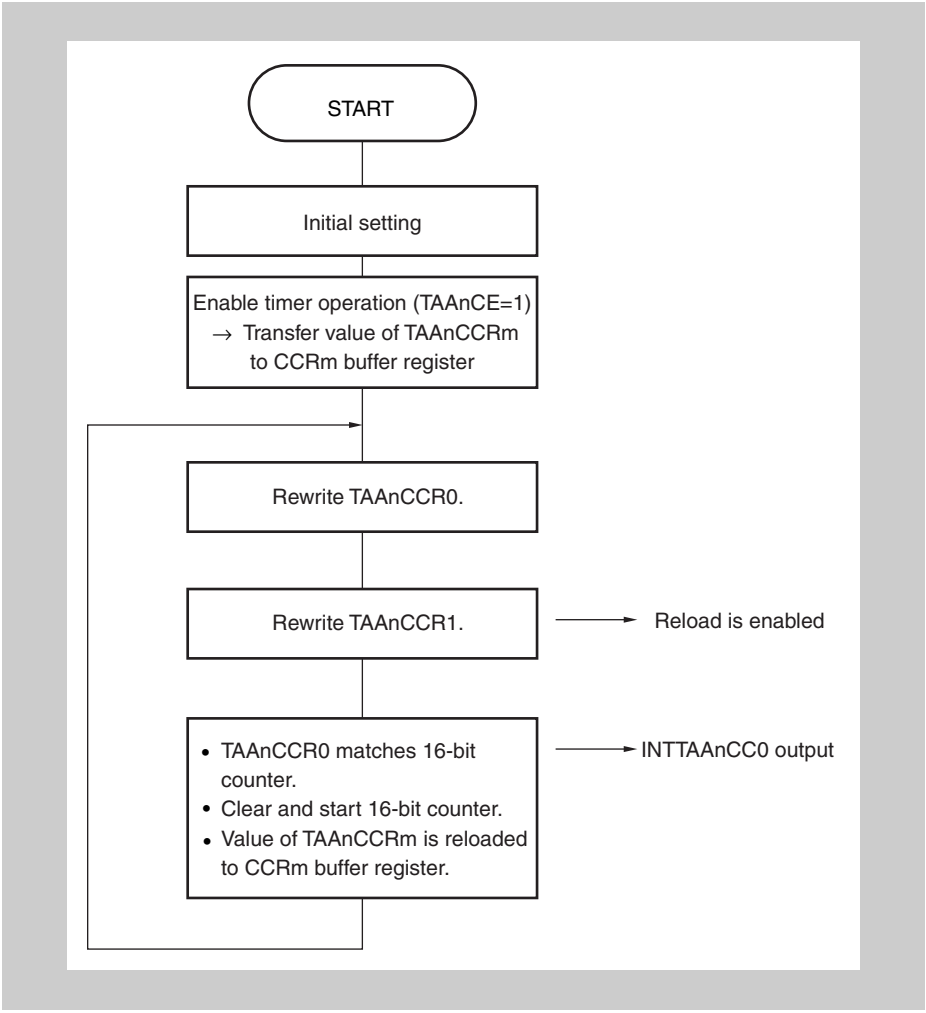


Figure 16-9 Flowchart of basic operation for reload

Caution Writing to the TAAnCCR1 register includes an operation to enable reload. Therefore, rewrite the TAAnCCR1 register after rewriting the TAAnCCR0 register.

Note 1. Above flowchart illustrates an example of the PWM mode operation.

2. n = 0 to 9; m = 0, 1

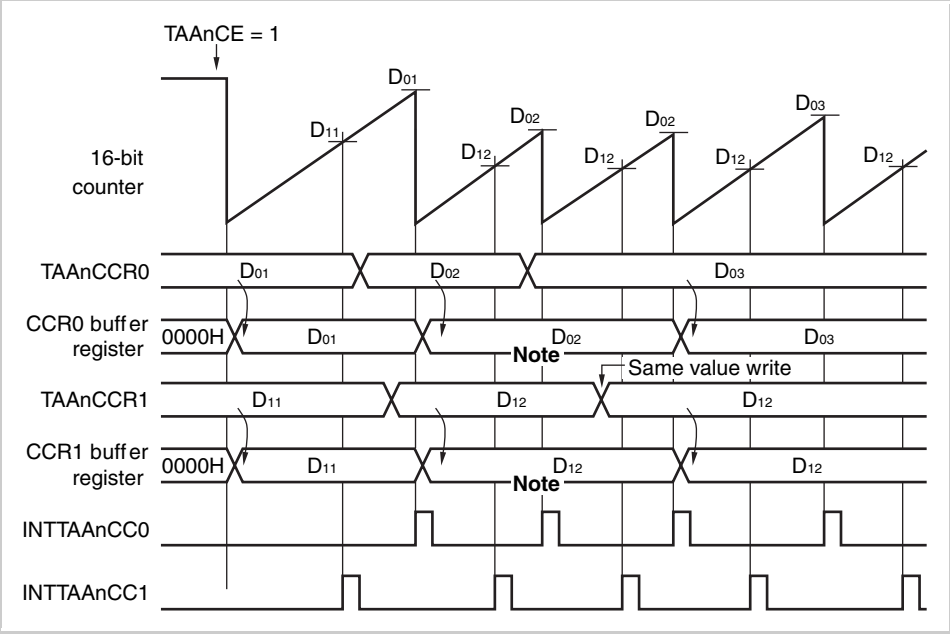


Figure 16-10 Timing chart for reload

Note Reload is not performed because TAAnCCR1 register is not written.

D01, D02, D03: Setting value of TAAnCCR0 register (0000H to FFFFH)
D11, D12: Setting value of TAAnCCR1 register (0000H to FFFFH)

Above flowchart illustrates PWM mode operation.

16.6.2 Interval timer mode (TAAnMD2 to TAAnMD0 = 000_B)

In the interval timer mode, an interrupt request signal (INTTAAnCC0) is generated upon a match between the setting value of the TAAnCCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared. The TAAnCCR0 register can be rewritten when TAAnCE = 1, and when a value is set to TAAnCCR0 with a write instruction from the CPU, it is transferred to the CCR0 buffer register through any time write mode, and is compared with the 16-bit counter value.

In the interval timer mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

16-bit counter clearing using the TAAnCCR1 register is not performed. However, the setting value of the TAAnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTAAnCC1) is output if these values match. In addition, TOAA1n pin output is also possible by setting the TAAnOE1 bit to 1.

When the TAAnCCR1 register is not used, it is recommended to set FFFFH as the setting value for the TAAnCCR1 register.

When performing timer output with the TOAAn1 pin, set the same values to the TAAnCCR0 register and the TAAnCCR1 register since the 16-bit timer counter cannot be cleared with the TAAnCCR1 register.

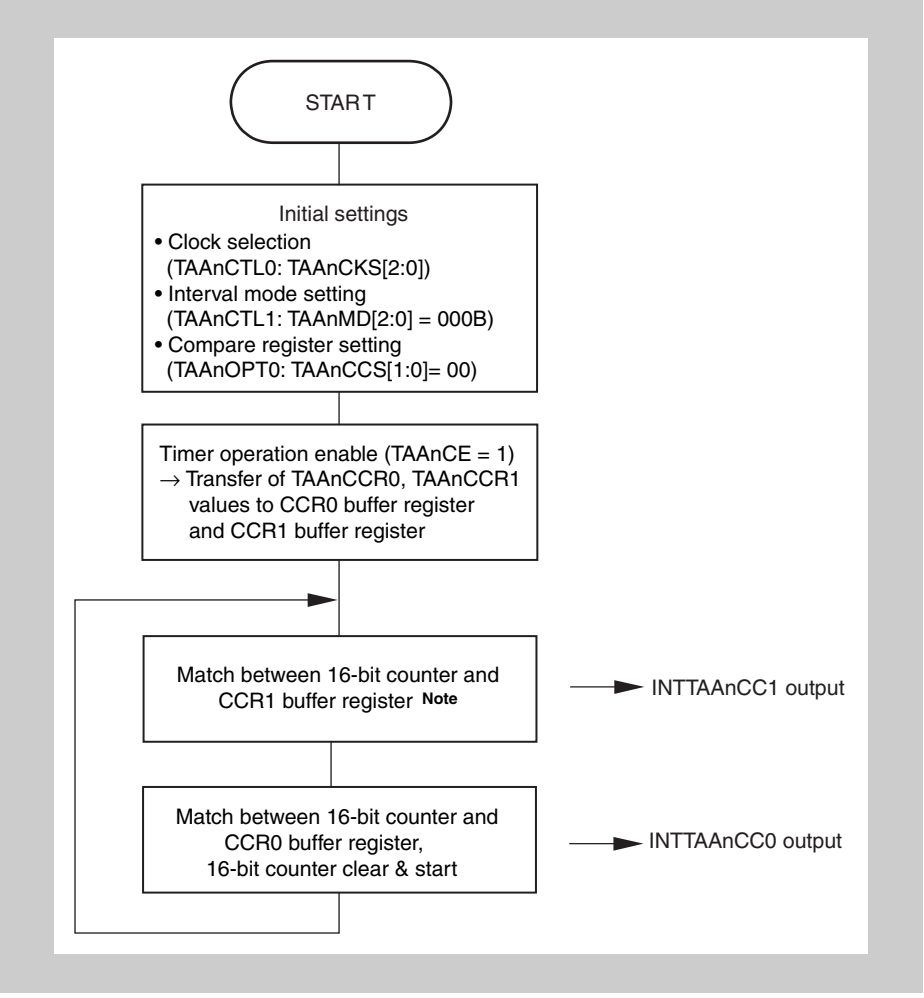


Figure 16-11 Flowchart of basic operation in interval timer mode

Note The 16-bit counter is not cleared when its value matches the value of TAAnCCR1.

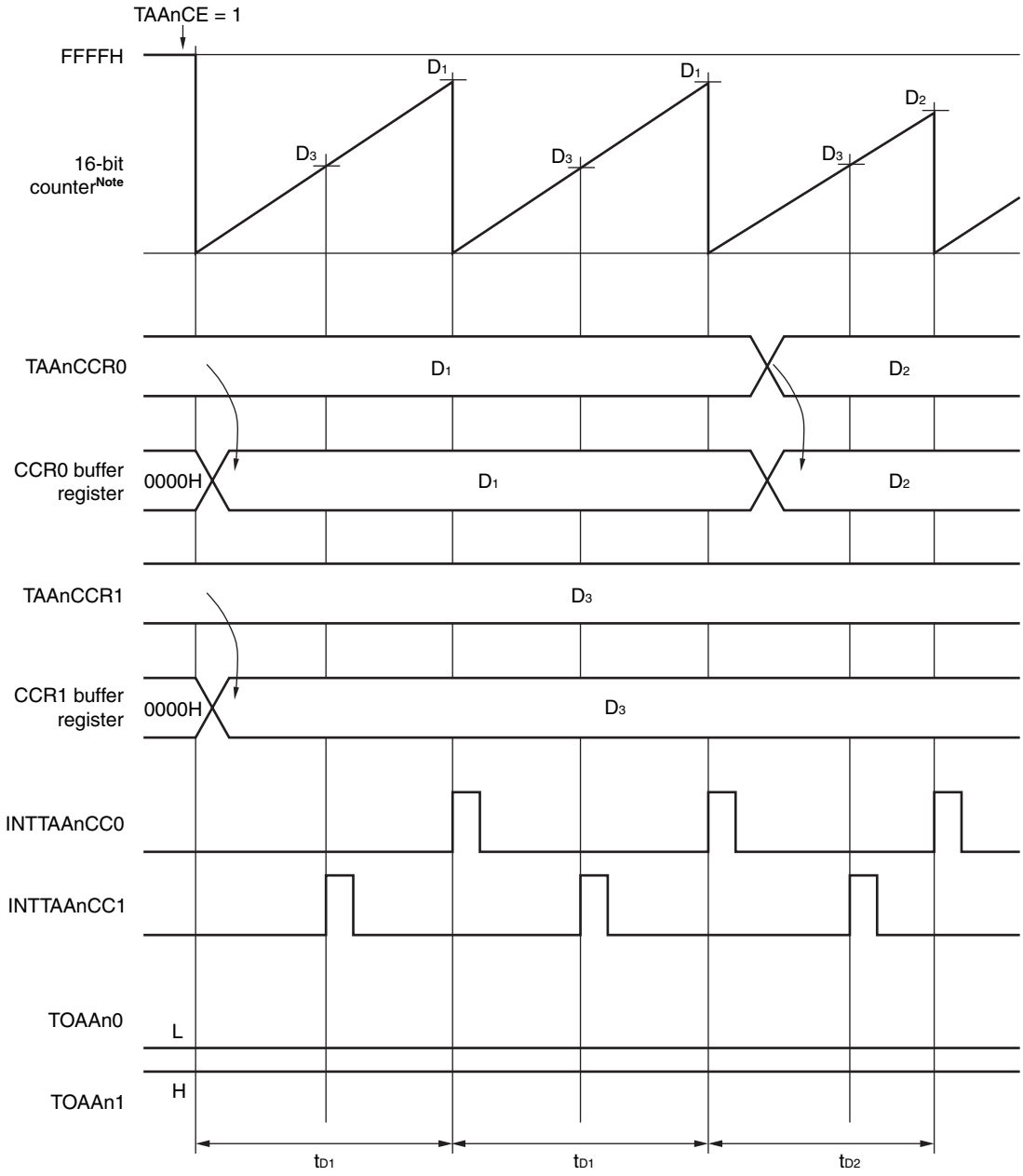


Figure 16-12 Basic operation timing in interval timer mode When D1 > D2 > D3; only TAAnCCR0 register value is written, and TOAAn0 and TOAAn1 are not output (TAAnOE0 = 0, TAAnOE1 = 0, TAAnOL0 = 0, TAAnOL1 = 1)

Note The 16-bit counter is not cleared when its value matches the value of TAAnCCR1.

D1, D2: Setting values of TAAnCCR0 register (0000H to FFFFH)
D3: Setting value of TAAnCCR1 register (0000H to FFFFH)

Interval time (t_{Dn})= (Dn + 1) × (count clock cycle)

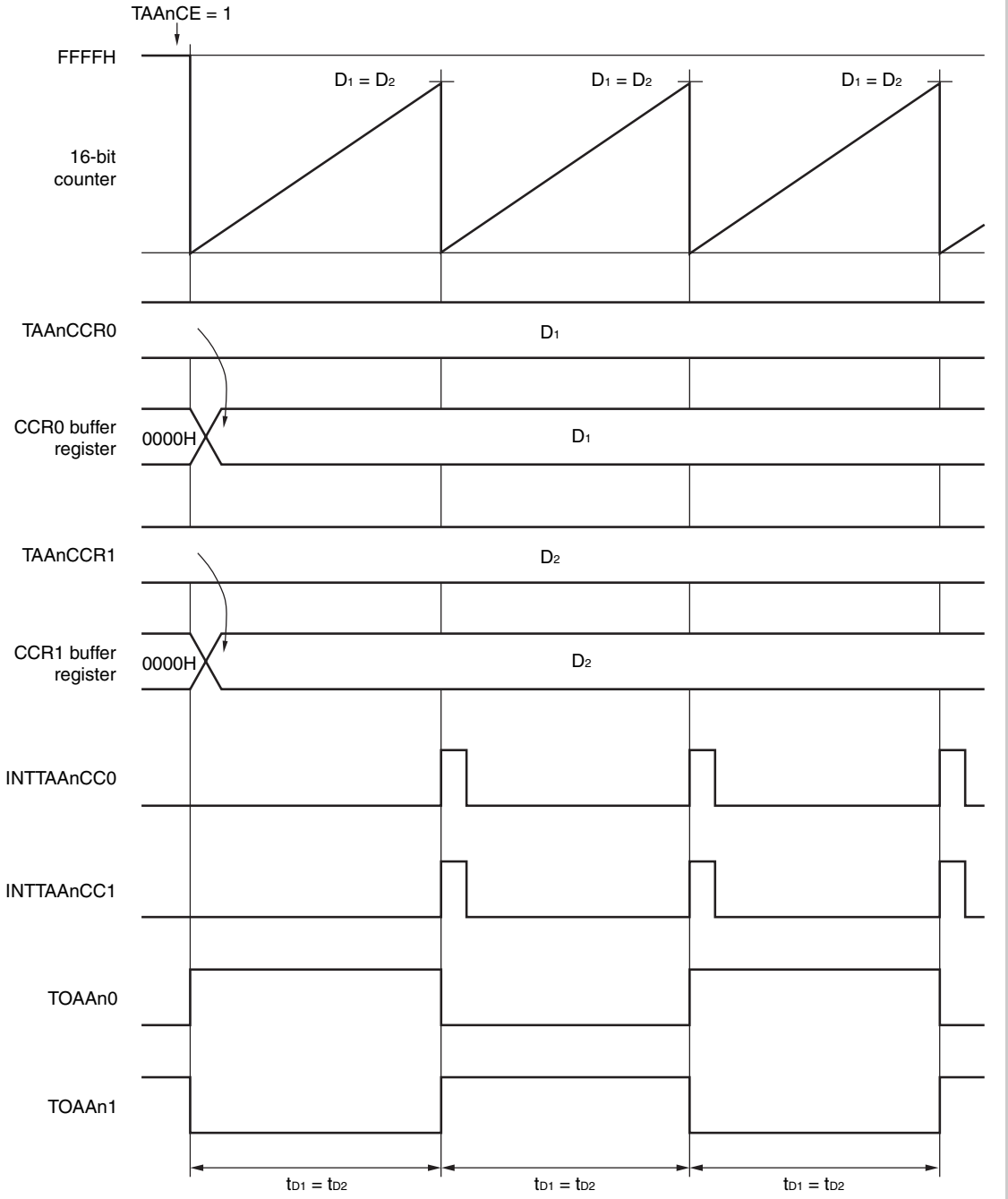


Figure 16-13 Basic operation timing in interval timer mode When D1 = D2; TAAnCCR0 and TAAnCCR1 are not rewritten, and TOAAn0 and TOAAn1 are output (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 1)

D1: Setting value of TAAnCCR0 register (0000H to FFFFH)
D2: Setting value of TAAnCCR1 register (0000H to FFFFH)

Interval time (t_{Dn}) = (Dn + 1) × (count clock cycle)

16.6.3 External event counter mode (TAAnMD2 to TAAnMD0 = 001_B)

In the external event count mode, the external event count input (TEVTAAn pin input) is used as a count-up signal. Regardless of the setting of the TAAnEEE bit of the TAAnCTL0 register, 16-bit timer/event counter AA counts up the external event count input (TEVTAAn pin input) when it is set in the external event count mode. In the external event count mode, an interrupt request (INTTAAnCC0) is generated when the set value of the TAAnCCR0 register matches the value of the 16-bit counter, and the value of the 16-bit counter is cleared.

When a value is set to the TAAnCCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through any time write, and is compared with the 16-bit counter value.

In the external event counter mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

The 16-bit counter can not be cleared using TAAnCCR1 register. However, the setting value of the TAAnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTAAnCC1) is output if these values match.

Moreover, TOAAnm pin output is also possible by setting the TAAnOEm bit to 1.

When performing timer output with the TOAAn1 pin, set the same values to TAAnCCR0 register and TAAnCCR1 register since the 16-bit counter cannot be cleared with CCR1 buffer register.

The TAAnCCR0 register can be rewritten when TAAnCE = 1. When TAAnCCR1 register is not used, it is recommended to set TAAnCCR1 register to FFFFH.

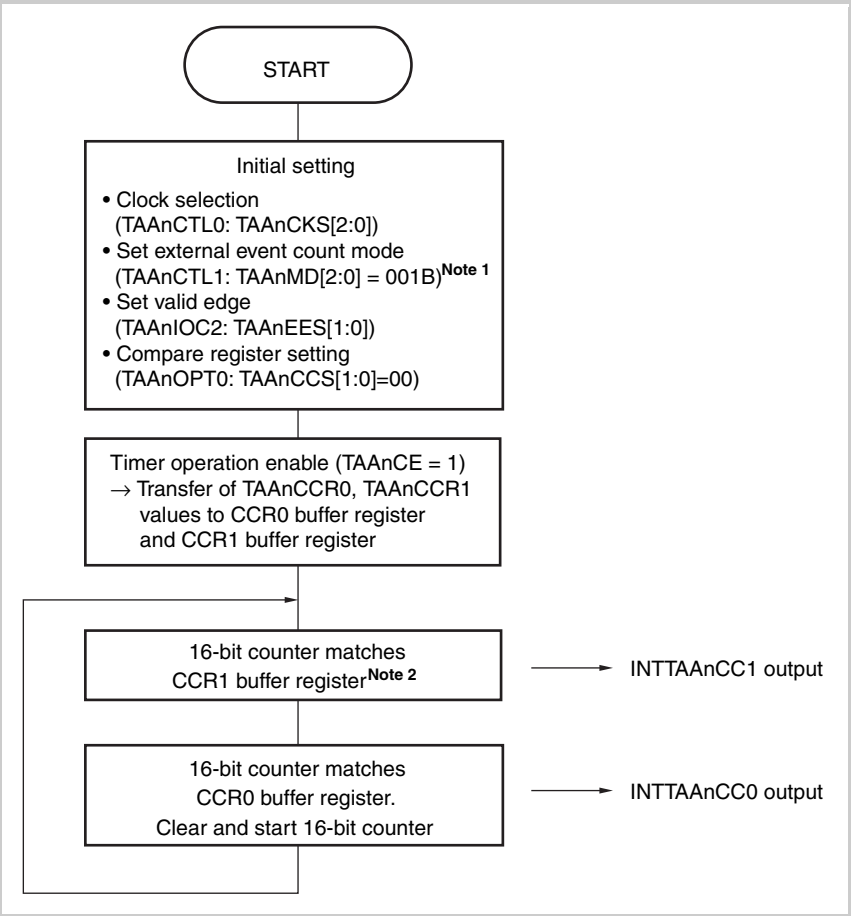


Figure 16-14 Flowchart of basic operation in external event counter mode

Note 1. Selection of the TAAnEEE bit has no influence.

2. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

3. n = 0 to 9; m = 0, 1

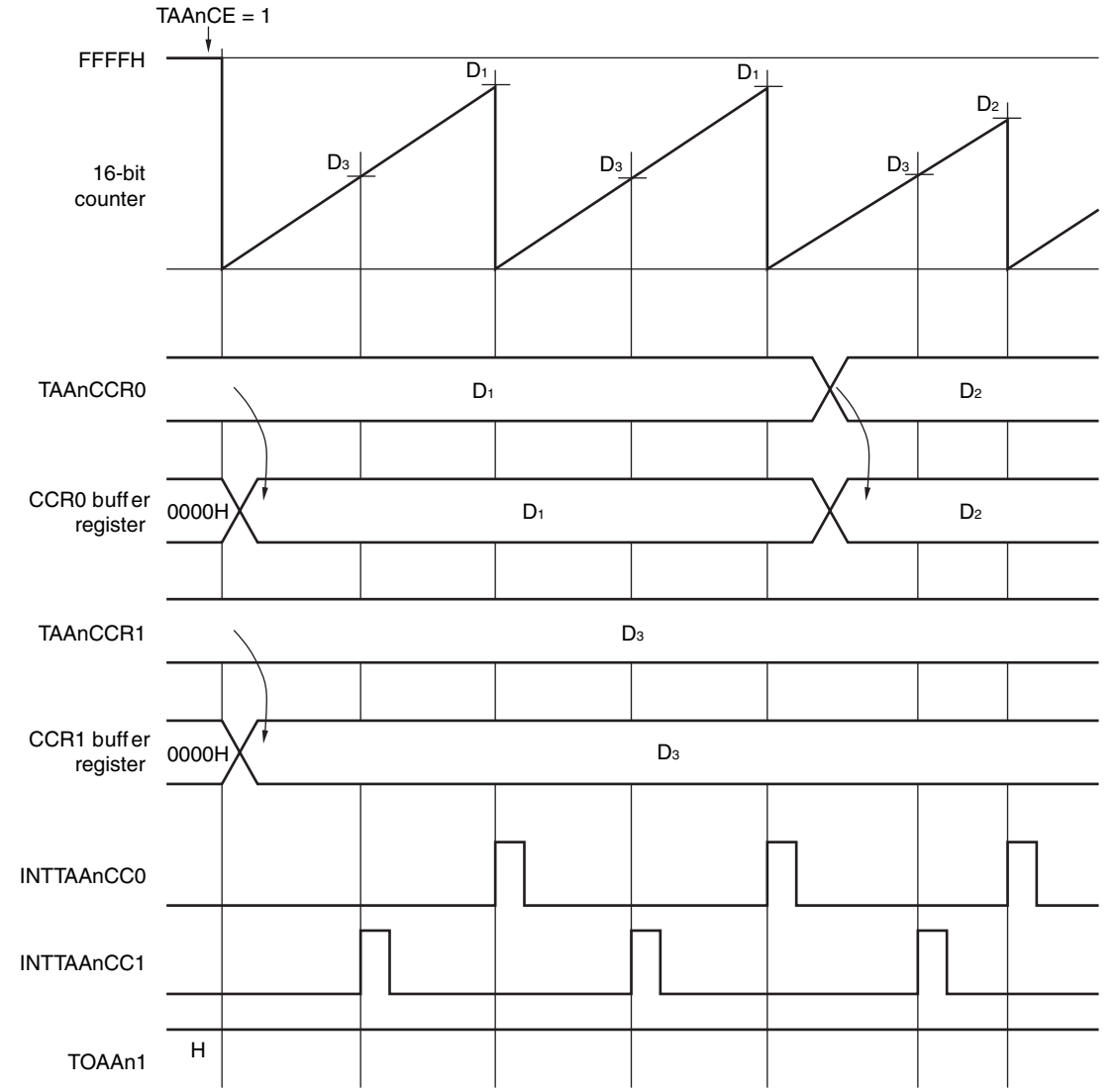


Figure 16-15 Basic operation timing in external event counter mode When D1 > D2 > D3; rewrite TAAnCCR0 only; TOAAn1 is not output (TAAnOE0 = 0, TAAnOE1 = 0, TAAnOL0 = 0, TAAnOL1 = 1)

D1, D2: Setting values of TAAnCCR0 register (0000H to FFFFH)
D3: Setting value of TAAnCCR1 register (0000H to FFFFH)

Number of event counts = (Dn + 1)

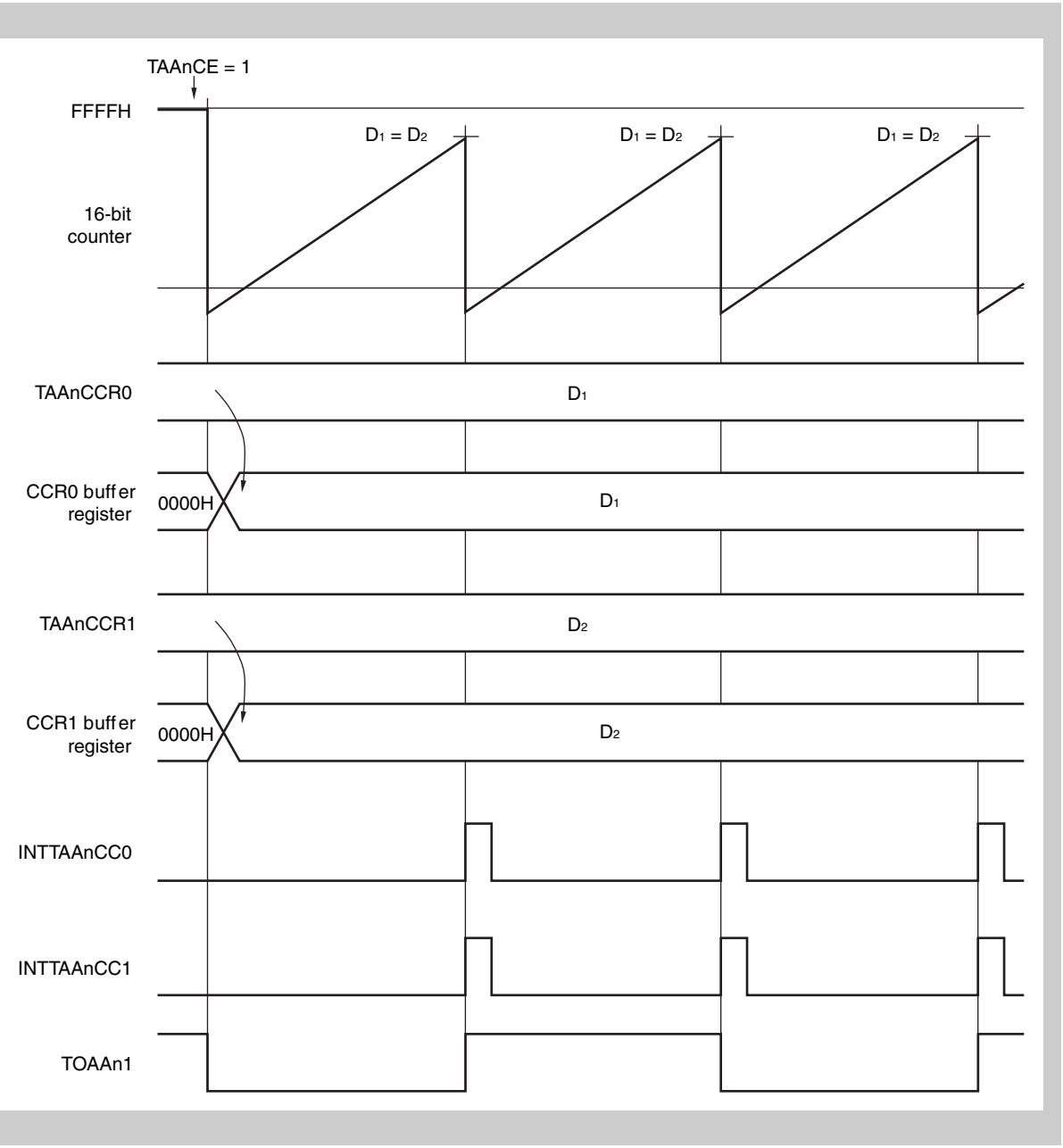


Figure 16-16 Basic operation timing in external event counter mode When D1 = D2; TAAnCCR0 and TAAnCCR1 are not rewritten, TOAAn1 is output (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 1)

D1: Setting value of TAAnCCR0 register (0000H to FFFFH)
D2: Setting value of TAAnCCR1 register (0000H to FFFFH)

Number of event count = (Dn + 1)

16.6.4 External trigger pulse mode (TAAnMD2 to TAAnMD0 = 010_B)

When TAAnCE = 1 in the external trigger pulse mode, the 16-bit counter stops at FFFFH and waits for input of an external trigger (TTRGAAn pin input). When the counter detects the edge of the external trigger (TTRGAAn pin input), it starts counting up. The duty factor of the signal output from the TOAAn1 pin is set by a reload register (TAAnCCR1) and the period is set by a compare register (TAAnCCR0).

Rewriting the TAAnCCR0 and TAAnCCR1 registers is enabled when TAAnCE = 1.

To ensure that the selected values of the TAAnCCR0 and TAAnCCR1 registers after rewriting are compared with the value of the 16-bit counter (reloaded to the CCRm buffer register), the TAAnCCR0 register and then the TAAnCCR1 register must be written before the value of the 16-bit counter matches the value of the TAAnCCR0 register.

When the value of the TAAnCCR0 register later matches the value of the 16-bit counter, the values of the TAAnCCR0 and TAAnCCR1 registers are reloaded to the CCRm buffer register.

Whether the next reload timing is made valid or not is controlled by writing to the TAAnCCR1 register. Therefore, write the same value to the TAAnCCR1 register when it is necessary to rewrite the value of only the TAAnCCR0 register.

Reload is invalid when only the TAAnCCR0 register is rewritten. To stop timer AA, clear TAAnCE to 0. If the edge of the external trigger (TTRGAAn pin input) is detected more than once in the external trigger pulse mode, the 16-bit counter is cleared at the point of edge detection, and resumes counting up. To realize the same function as the external trigger pulse mode by using a software trigger instead of the external trigger input (TTRGAAn pin input) (software trigger pulse mode), a software trigger is generated by setting the TAAnEST bit of the TAAnCTL1 register to 1.

When using a software trigger, a square wave that has one cycle of the PWM waveform as half of its own cycle can also be outputed from the TOAAn0 pin.

The waveform of the external trigger pulse is output from TOAAn1. A toggle output is produced from the TOAAn0 pin when the value of the TAAnCCR0 register matches the value of the 16-bit counter.

In the external trigger pulse mode, the capture function of the TAAnCCR0 and TAAnCCR1 registers cannot be used because these registers can be used only as compare registers.

Caution In the external trigger pulse mode, select the internal clock (TAAnEEE bit of TAAnCTL1 register = 0) for the count clock.

Note 1. For the reload operation when TAAnCCR0 and TAAnCCR1 are rewritten during timer operation, refer to *"PWM mode (TAAnMD2 to TAAnMD0 = 100_B)" on page 646*.

2. n = 0 to 9; m = 0, 1

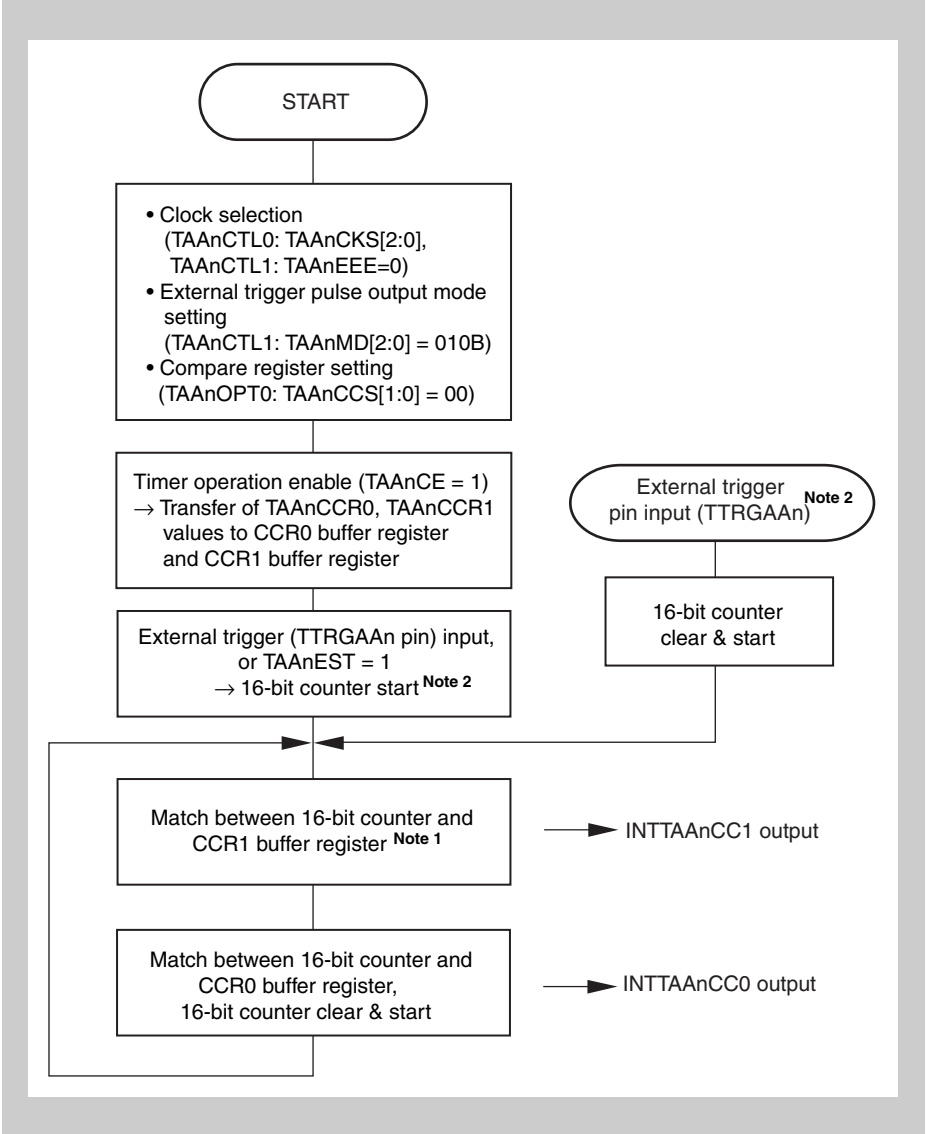


Figure 16-17 Flowchart of basic operation in external trigger pulse output mode

Note 1. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

2. External trigger input pin is not available for TAA8.

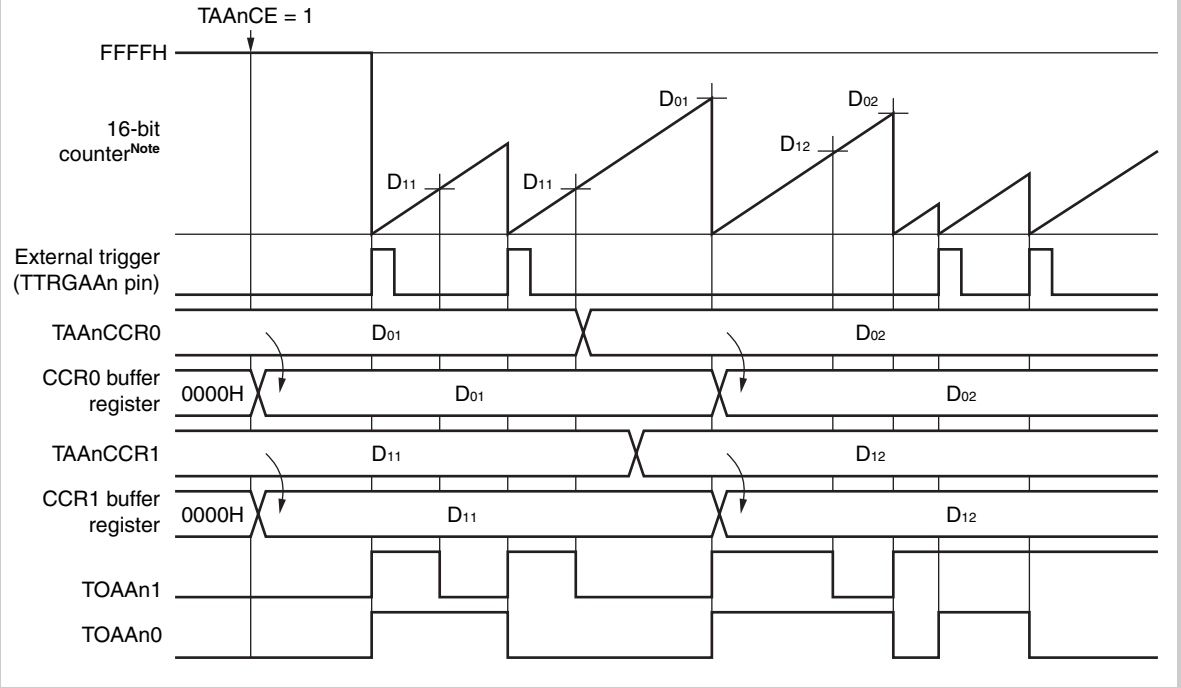


Figure 16-18 Basic operation timing in external trigger pulse output mode (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

Note The 16-bit counter is not cleared when it matches the CCR1 buffer register.

D01, D02: Setting value of TAAnCCR0 register (0000H to FFFFH)
D11, D12: Setting value of TAAnCCR1 register (0000H to FFFFH)

Duty of TOAAn1 output =
(Set value of TAAnCCR1 register) / (Set value of TAA0CCR0 register)
Cycle of TOAAn1 output =
(Set value of TAAnCCR0 register) Þ (Count clock cycle)

16.6.5 One-shot pulse mode (TAAnMD2 to TAAnMD0 = 011_B)

When TAAnCE is set to 1 in the one-shot pulse mode, the 16-bit counter waits for the setting of the TAAnEST bit (to 1) or a trigger that is input when the edge of the TTRGAAn pin is detected, while holding FFFFH. When the trigger is input, the 16-bit counter starts counting up.

When the value of the 16-bit counter matches the value of the CCR1 buffer register that has been transferred from the TAAnCCR1 register, TOAAn1 goes high. When the value of the 16-bit counter

matches the value of the CCR0 buffer register that has been transferred from the TAAnCCR0 register, TOAAn1 goes low, and the 16-bit counter is cleared to 0000H and stops. Input of a second or subsequent trigger is ignored while the 16-bit counter is operating. Be sure to input a second trigger while the 16-bit counter is stopped at 0000H. In the one shot pulse mode, rewriting the TAAnCCR0 and TAAnCCR1 registers is enabled when TAAnCE = 1. The set values of the TAAnCCR0 and TAAnCCR1 registers become valid after a write instruction from the CPU is executed. They are then transferred to the CCR0 and CCR1 buffer registers, and compared with the value of the 16-bit counter. The waveform of the one-shot pulse is output from the TOAAn1 pin. The TOAAn0 pin produces a toggle output when the value of the 16-bit counter matches the value of the TAAnCCR0 register. In the one-shot pulse mode, the TAAnCCR0 and TAAnCCR1 registers function only as compare registers. They cannot be used as capture registers.

Caution

1. In the one-shot pulse mode, select the internal clock (TAAnEEE bit of TAAnCTL1 register = 0) as the count clock.
2. In the one-shot pulse mode, it is prohibited to set the TAAnCCR1 register to 0000H.

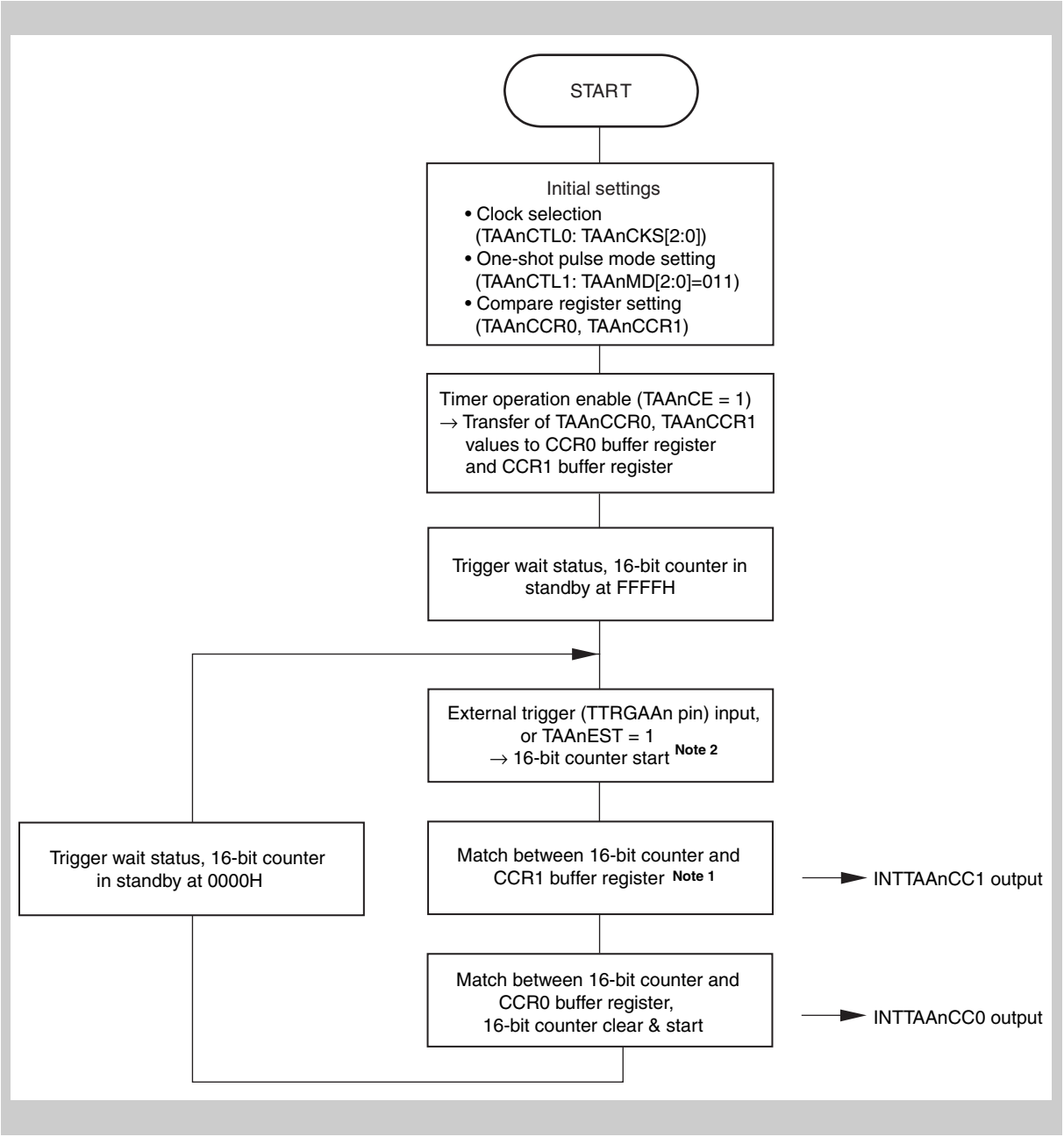


Figure 16-19 Flowchart of basic operation in one-shot pulse mode

Note 1. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

2. External trigger input pin is not available for TAA8.

Caution The 16-bit counter is not cleared when a trigger input is performed during the count-up operation of the 16-bit counter.

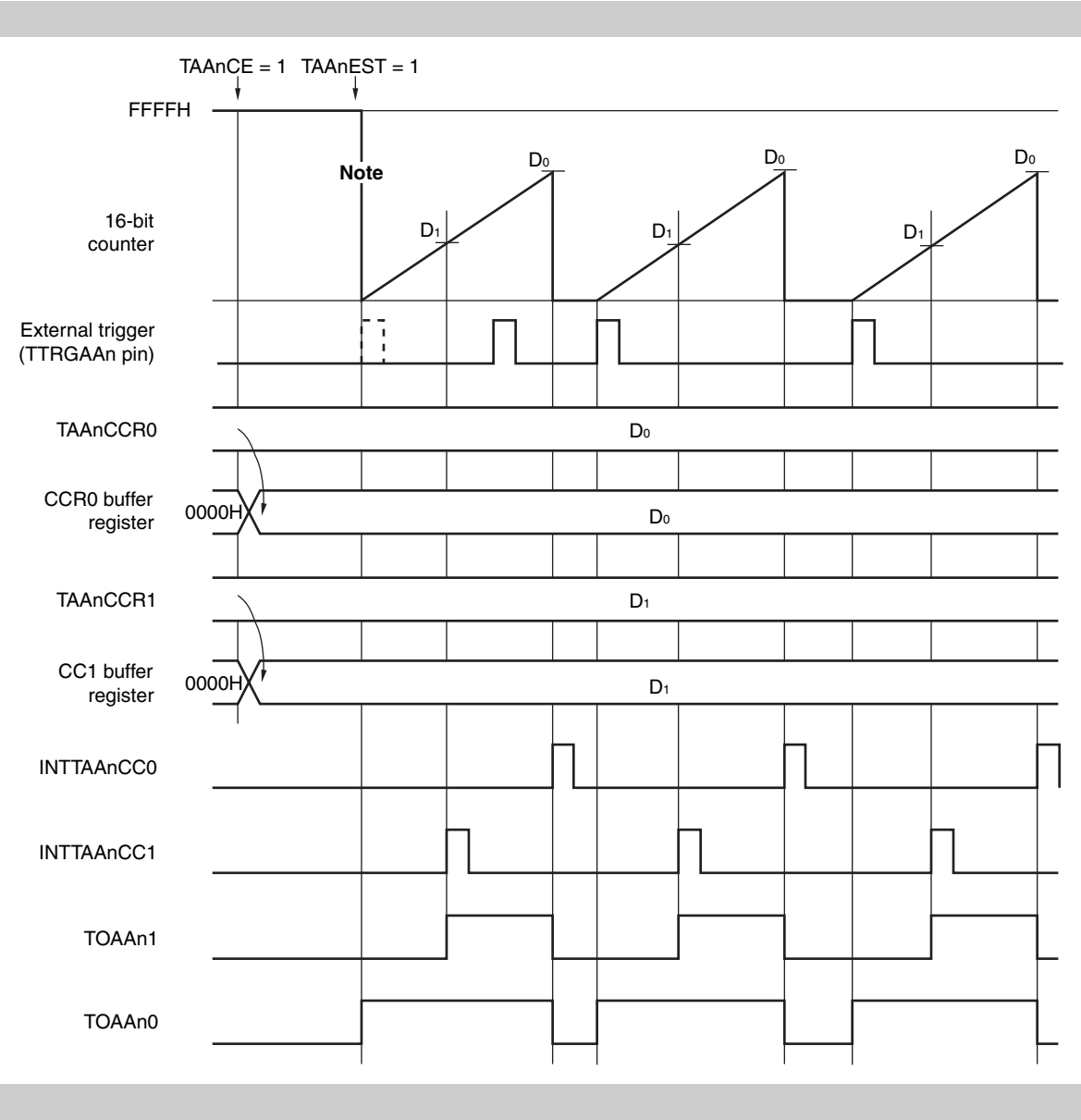


Figure 16-20 Timing of basic operation in one-shot pulse mode (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

Note The 16-bit counter starts counting up when either TAAnEST = 1 is set or the external trigger (TTRGAAn) is input.

D0: Setting value of TAAnCCR0 register (0000H to FFFFH)
D1: Setting value of TAAnCCR1 register (0000H to FFFFH)

16.6.6 PWM mode (TAAnMD2 to TAAnMD0 = 100_B)

In the PWM mode, TAAn capture/compare register 1 (TAAnCCR1) is used to set the duty factor and TAAn capture/compare register 0 (TAAnCCR0) is used to set the cycle. By using these two registers and operating the timer, variable-duty PWM is output.

Rewriting the TAAnCCR0 and TAAnCCR1 registers is enabled when TAAnCE = 1.

So that the set values of the TAAnCCR0 and TAAnCCR1 registers are compared with the value of the 16-bit counter (reloaded to the CCR0 and CCR1 buffer registers), the TAAnCCR0 register must be rewritten and then a value must be written to the TAAnCCR1 register before the value of the 16-bit counter matches the value of the TAAnCCR0 register.

The values of the TAAnCCR0 and TAAnCCR1 registers are reloaded when the value of the TAAnCCR0 register later matches the value of the 16-bit counter. Whether the next reload timing is made valid or not is controlled by writing to the TAAnCCR1 register. Therefore, write the same value to the TAAnCCR1 register even when only the value of the TAAnCCR0 register needs to be rewritten. Reload is invalid when only the value of the TAAnCCR0 register is rewritten.

To stop timer AA, clear TAAnCE to 0.

The waveform of PWM is output from the TOAAn1 pin. The TOAAn0 pin produces a toggle output when the 16-bit counter matches the TAAnCCR0 register.

In the PWM mode, the TAAnCCR0 and TAAnCCR1 registers are used only as compare registers. They cannot be used as capture registers.

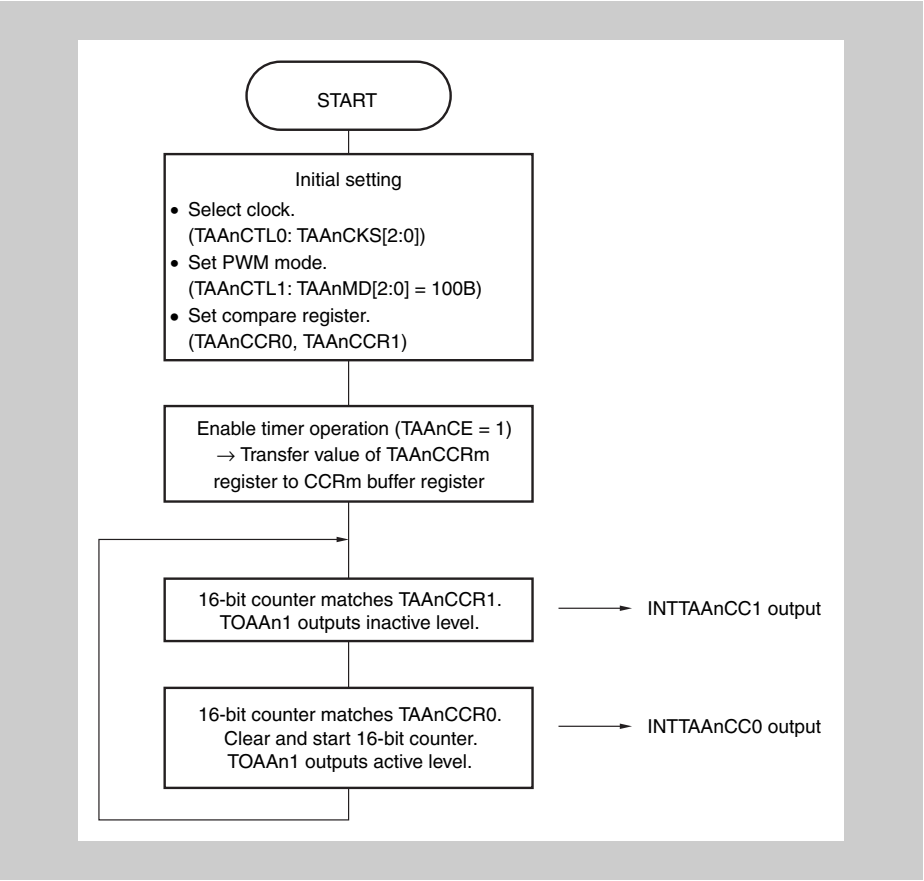


Figure 16-21 Flowchart of basic operation in PWM mode When values of TAAnCCR0, TAAnCCR1 registers are not rewritten during timer operation

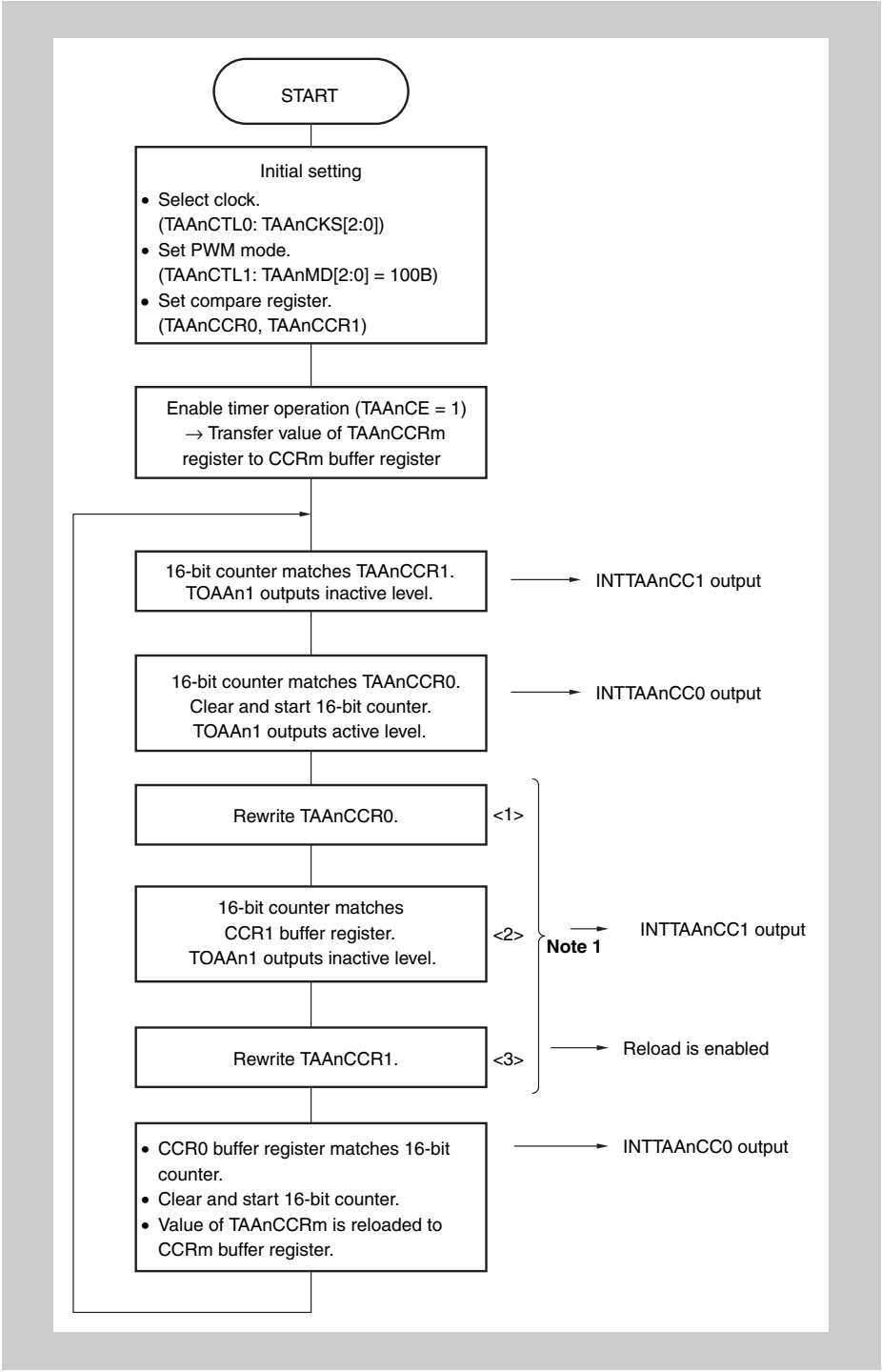


Figure 16-22 Flowchart of basic operation in PWM mode When values of TAAnCCR0, TAAnCCR1 registers are rewritten during timer operation

Note 1. The timing of <2> in the above flowchart may differ depending on the rewrite timing of steps <1> and <3> and the value of TAAnCCR1, but make sure that step <3> comes after step <1>.

2. n = 0 to 9; m = 0, 1

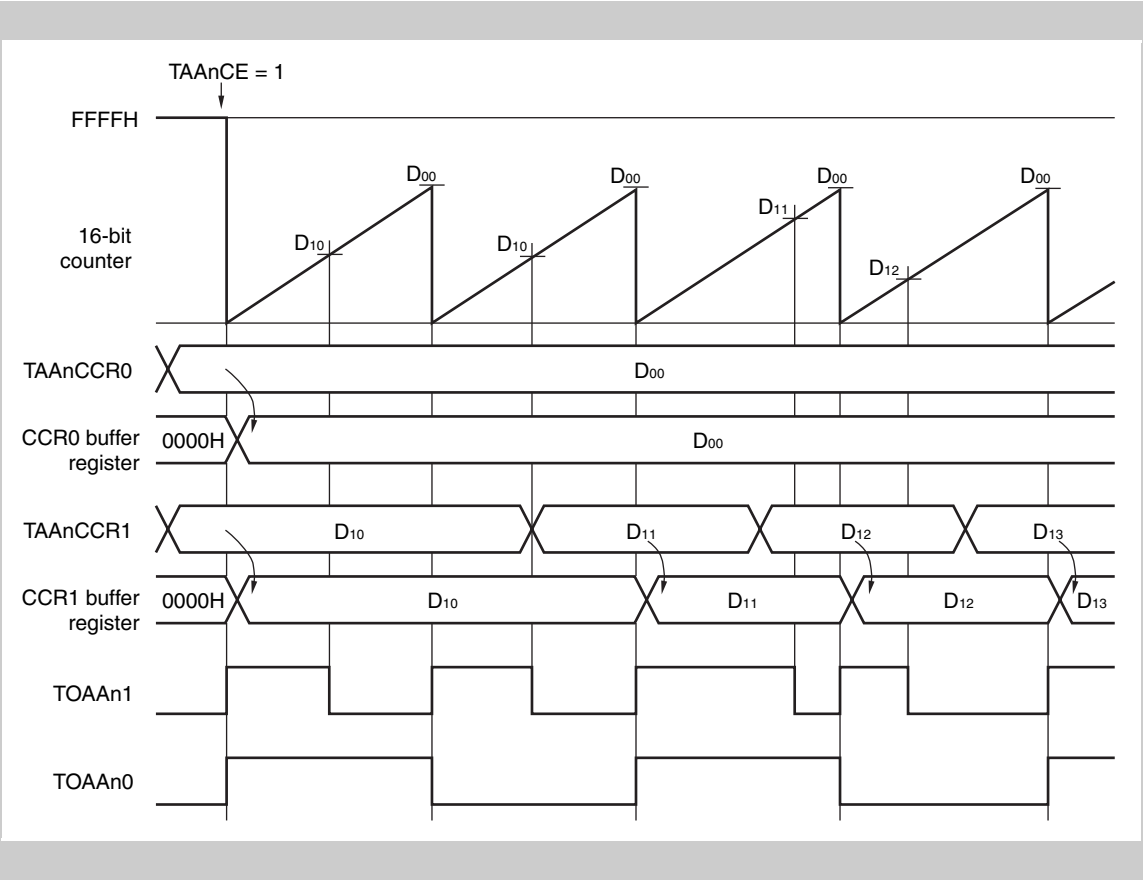


Figure 16-23 Basic Operation Timing in PWM mode When rewriting TAAnCCR1 value (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

D00: Set value of TAAnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Set value of TAAnCCR1 register (0000H to FFFFH)

Duty of TOAAn1 output =
(Set value of TAAnCCR1 register) / (Set value of TAA0CCR0 register)
Cycle of TOAAn1 output =
(Set value of TAAnCCR0 register) x (Count clock cycle)
Toggle width of TOAAn0 output =
(Set value of TAAnCCR0 register + 1) x (Count clock period)

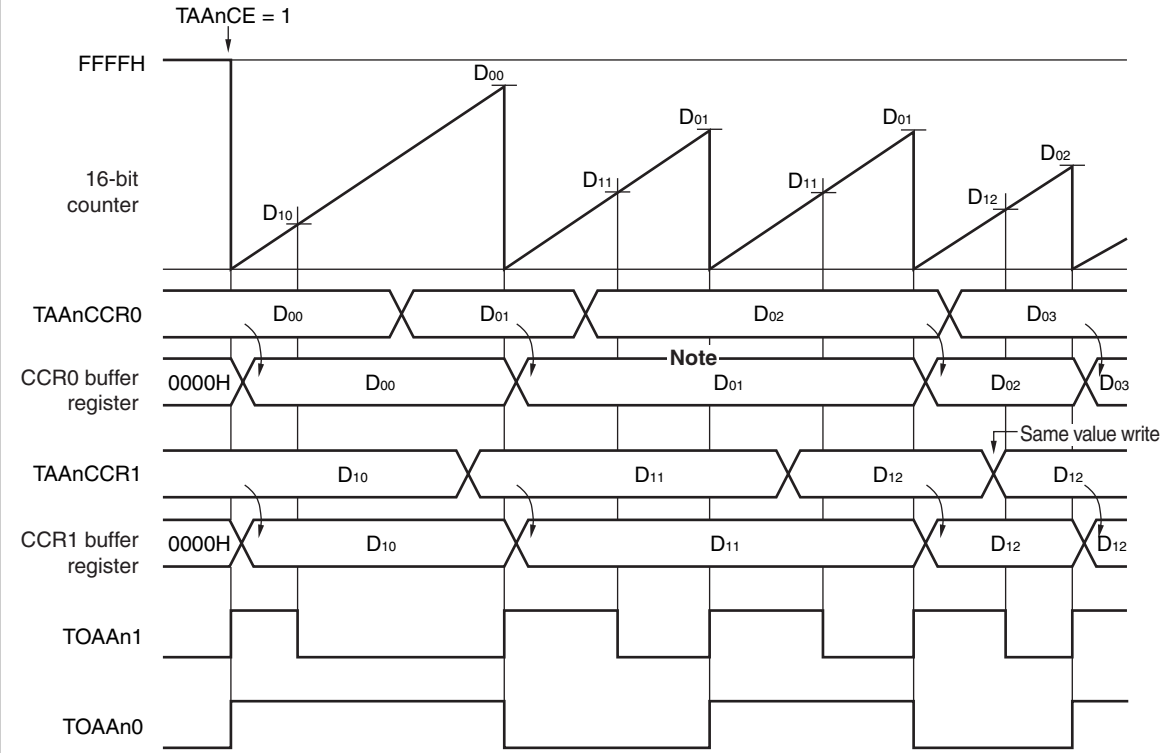


Figure 16-24 Basic operation timing in PWM mode When TAAnCCR0, TAAnCCR1 values are rewritten (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

Note Reload is not performed because the TAAnCCR1 register was not rewritten.

D00, D01, D02, D03: Setting values of TAAnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Setting values of TAAnCCR1 register (0000H to FFFFH)

Duty of TOAAn1 output =
(Set value of TAAnCCR1 register) / (Set value of TAA0CCR0 register)
Cycle of TOAAn1 output =
(Set value of TAAnCCR0 register) x (Count clock cycle)
Toggle width of TOAAn0 output =
(Set value of TAAnCCR0 register + 1) x (Count clock cycle)

16.6.7 Free-running mode (TAAnMD2 to TAAnMD0 = 101_B)

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 TAAnCCS1 and TAAnCCS0 bits.

The settings of the TAAnCCS1 and TAAnCCS0 bits of the TAAnOPT0 register are valid only in the free-running mode.

TAAnCCS1	Operation
0	Use TAAnCCR1 register as compare register
1	Use TAAnCCR1 register as capture register

TAAnCCS0	Operation
0	Use TAAnCCR0 register as compare register
1	Use TAAnCCR0 register as capture register

- Using TAAnCCR1 register as compare register

  An interrupt is output upon a match between the 16-bit counter and the CCR1 buffer register in the free-running mode (interval function).

  Rewrite during compare timer operation is enabled and performed with any time write mode. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)

  When timer output (TOAAn1) has been enabled, TOAAn1 performs toggle output upon a match between the 16-bit counter and the CCR1 buffer register.
- Using TAAnCCR1 register as capture register

  The value of the 16-bit counter is saved to the TAAnCCR1 register upon TIAAn1 pin edge detection.
- Using TAAnCCR0 register as compare register

  An interrupt is output upon a match between the 16-bit counter and the CCR0 buffer register in the free-running mode (interval function).

  Rewrite during compare timer operation is enabled and performed with any time write mode. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)

  When timer output (TOAAn0) has been enabled, TOAAn0 performs toggle output upon a match between the 16-bit counter and the CCR0 buffer register.
- Using TAAnCCR0 register as capture register

  The value of the 16-bit counter is saved to the TAAnCCR0 register upon TIAAn0 pin edge detection.

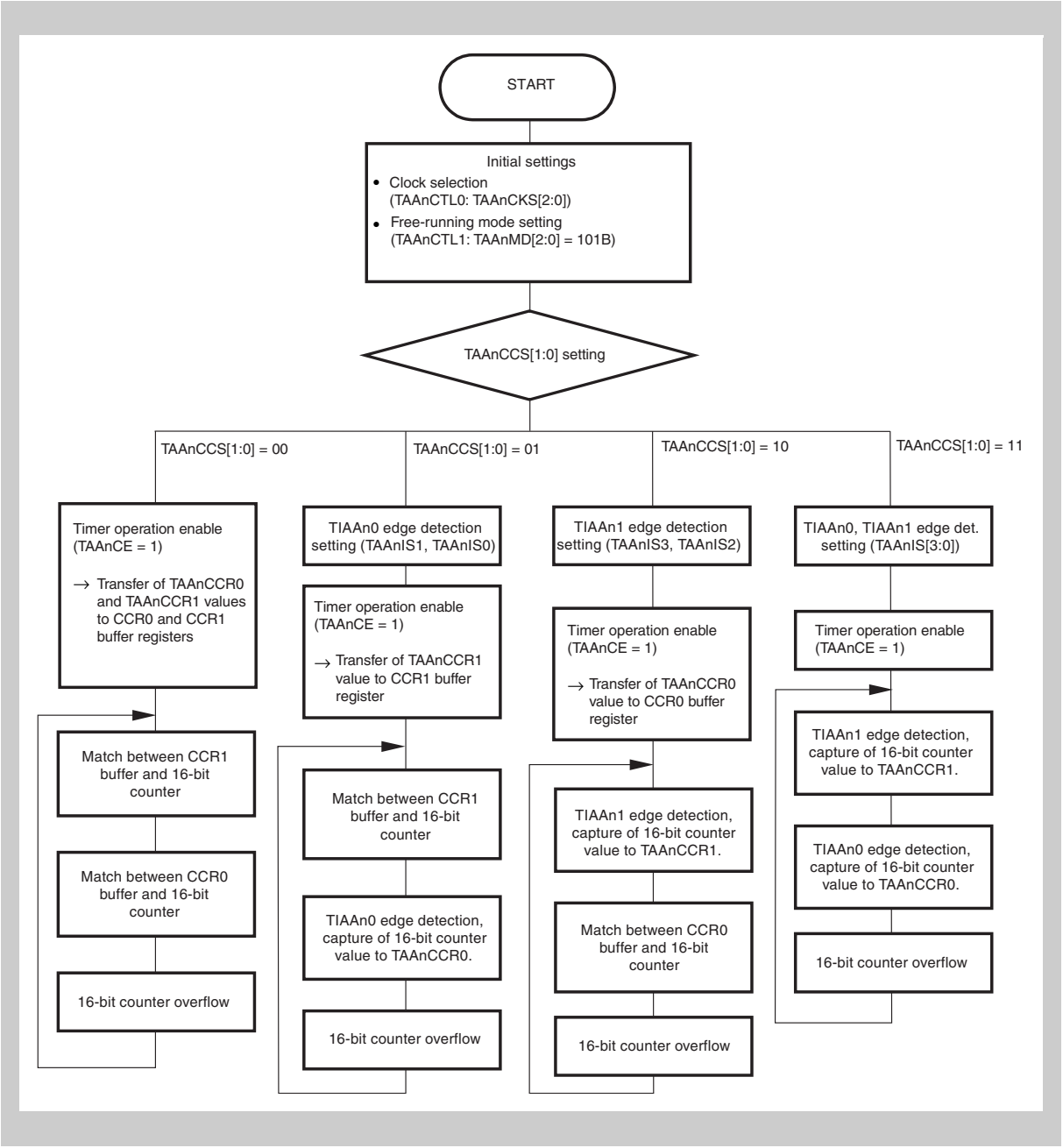


Figure 16-25 Flowchart of basic operation in free-running mode

(1) When TAAnCCS1 = 0, and TAAnCCS0 = 0 settings (interval function description, compare function)

When TAAnCE = 1 is set, the 16-bit counter counts from 0000H to FFFFH and the free-running count-up operation continues until TAAnCE = 0 is set.

In this mode, when a value is written to the TAAnCCR0 and TAAnCCR1 registers, they are transferred to the CCR0 buffer register and the CCR1 buffer register (any time write mode). In this mode, no one-shot pulse is output even when an one-shot pulse trigger is input. Moreover, when TAAnOEm = 1 is set, TOAAnm performs toggle output upon a match between the 16-bit counter and the CCRm buffer register.

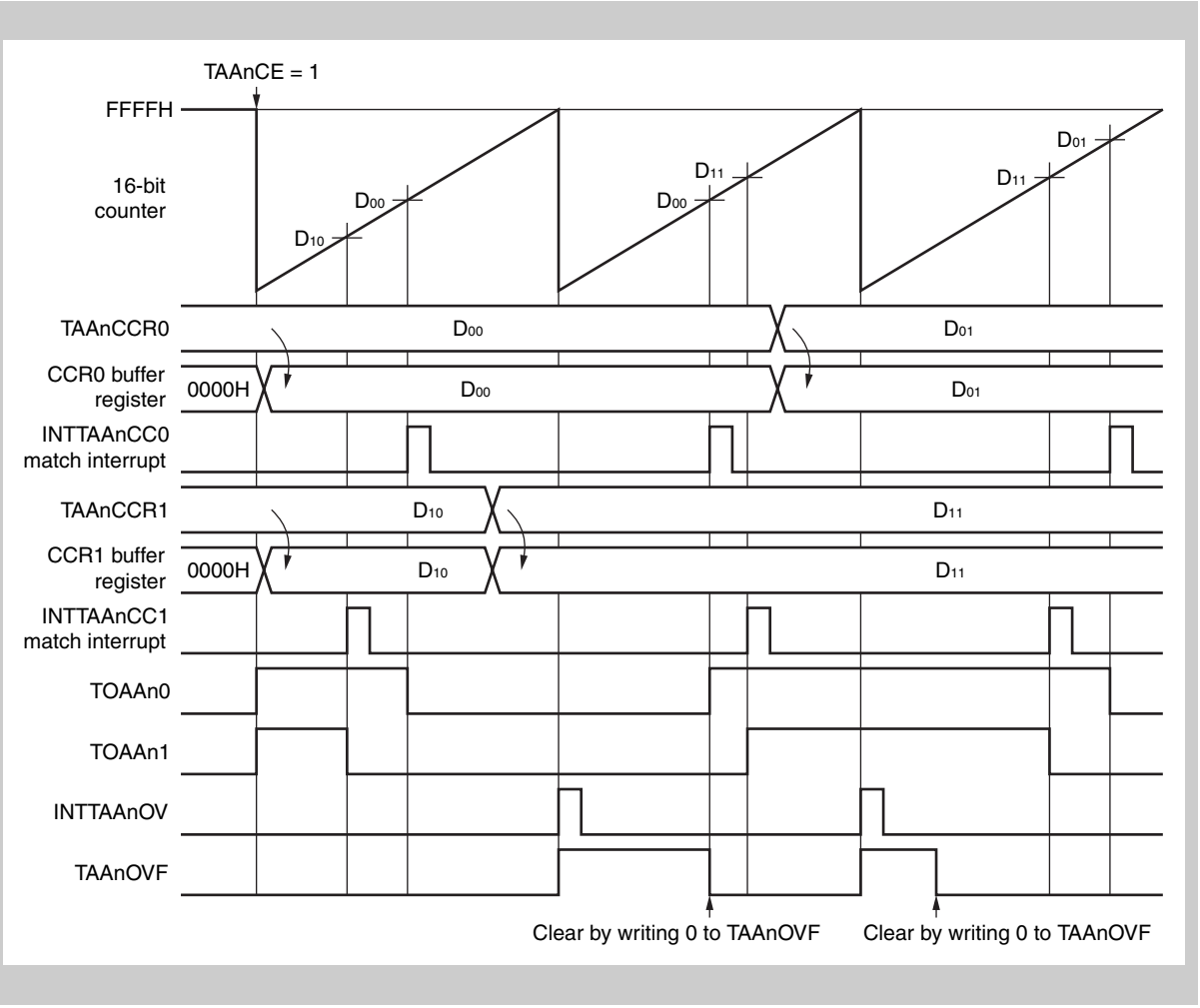


Figure 16-26 Basic Operation Timing In Free-running Mode (TAAnCCS1 = 0, TAAnCCS0 = 0) (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

D00, D01: Setting values of TAAnCCR0 register (0000H to FFFFH)
D10, D11: Setting values of TAAnCCR1 register (0000H to FFFFH)

TOAAnm output goes high when counting is started.

Note n = 0 to 9; m = 0, 1

(2) When TAAnCCS1 = 1 and TAAnCCS0 = 1 settings (capture function description)

When TAAnCE = 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TAAnCE = 0 is set. During this time, values are captured by capture trigger operation and are written to the TAAnCCR0 and TAAnCCR1 registers.

Regarding capture close to the overflow (FFFFH), judgment is made using the overflow flag (TAAnOVF). However, if overflow occurs twice (two or more free-running cycles), the capture trigger interval cannot be judged with the TAAnOVF flag. In this case, the system should be revised.

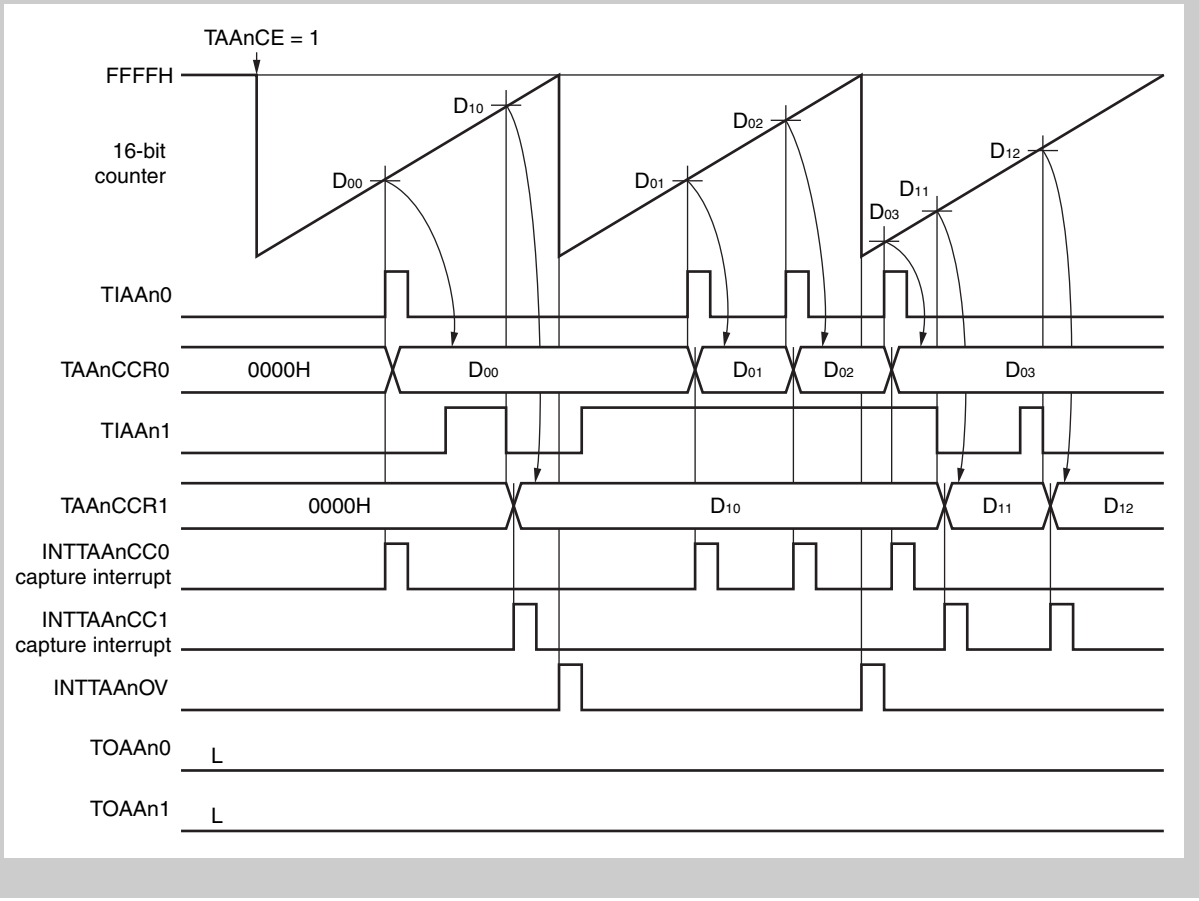


Figure 16-27 Basic operation timing in free-running mode (TAAnCCS1 = 1, TAAnCCS0 = 1) (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

D00, D01, D02, D03:
Values captured to TAAnCCR0 register (0000H to FFFFH)
D10, D11, D12:
Values captured to TAAnCCR1 register (0000H to FFFFH)

TIAAn0: Set to rising edge detection (TAAnIS1, TAAnIS0 = 01)
TIAAn1: Set to falling edge detection (TAAnIS3, TAAnIS2 = 10)

(3) When TAAnCCS1 = 1 and TAAnCCS0 = 0

When TAAnCE = 1 is set, the counter counts from 0000H to FFFFH and free-running count-up operation continues until TAAnCE = 0 is set. The TAAnCCR0 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 TAAnCCR0 register as an interval function. Even if TAAnOE1 = 1 to realize the output function, TAAnCCR1 register cannot control TOAAn1 because it is used as capture register.

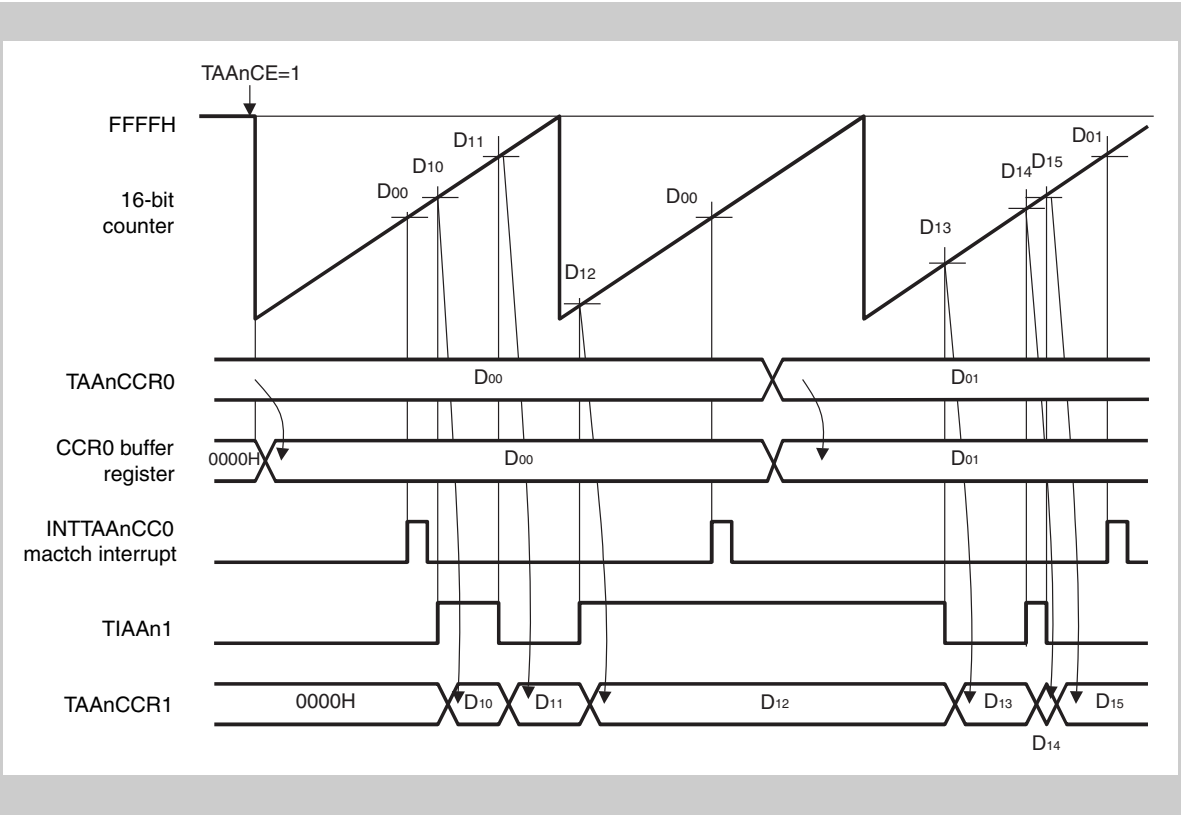


Figure 16-28 Basic operation timing in free-running mode (TAAnCCS1 = 1, TAAnCCS0 = 0) (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

D00, D01:
Setting compare values of TAAnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13, D14, D15:
Values captured to TAAnCCR1 register (0000H to FFFFH)

TIAAn1:
Set to detection of both rising and falling edges (TAAnIS3, TAAnIS2 = 11)

(4) When TAAnCCS1 = 0 and TAAnCS0 = 1

When TAAnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TAAnCE = 0 is set. The TAAnCCR1 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 TAAnCCR1 register as an interval function. When TAAnOE1 = 1 is set, TOAAn1 performs toggle output upon mach between the value of the 16-bit counter and the setting value of the TAAnCCR1 register.

Even if TAAnOE0 = 1 to realize the output function, TAAnCCR0 register cannot control TOAAn0 because it is used as capture register.

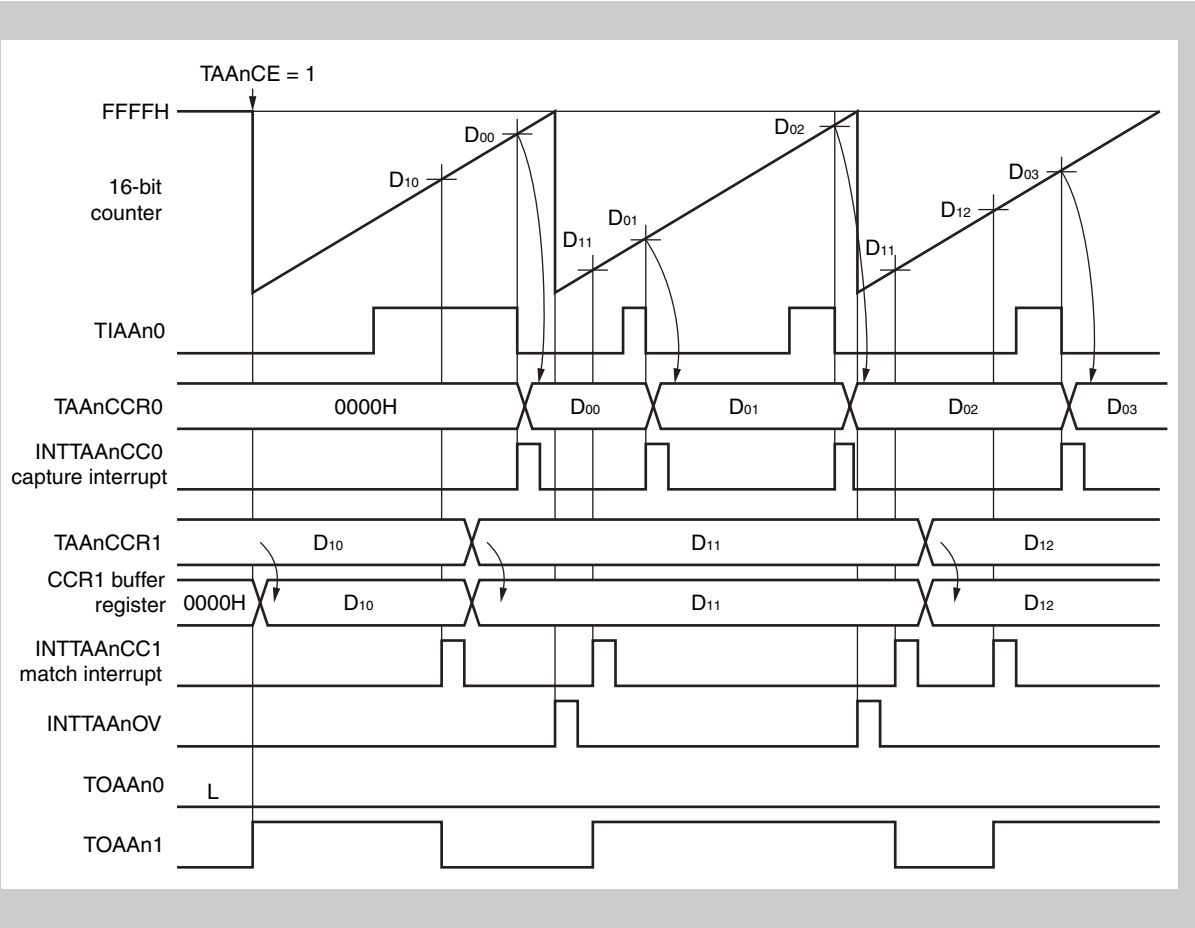


Figure 16-29 Basic operation timing in free-running mode (TAAnCCS1 = 0, TAAnCCS0 = 1) (TAAnOE0 = 1, TAAnOE1 = 1, TAAnOL0 = 0, TAAnOL1 = 0)

D00, D01, D02, D03:
Values captured to TAAnCCR0 register (0000H to FFFFH)
D10, D11, D12:
Setting compare value of TAAnCCR1 register (0000H to FFFFH)

TIAAn0: Set to falling edge detection (TAAnIS1, TAAnIS0 = 10)

(5) Overflow flag

When the counter overflows from FFFFH to 0000H in the free-running mode, the overflow flag (TAAnOVF) is set to 1 and an overflow interrupt (INTTAAnOV) is output.

The overflow flag is cleared by the CPU when writing 0 to it.

16.6.8 Pulse width measurement mode (TAAnMD2 to TAAnMD0 = 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 (TAAnCCR0), or capture register 1 (TAAnCCR1) respectively, and the 16-bit counter is cleared upon edge detection of the TIAAn0 pin, or TIAAn1 respectively. The external input pulse width can be measured as a result.

However, when measuring a large pulse width that exceeds 16-bit counter overflow, perform judgment with the overflow flag. Since measurement of pulses for which overflow occurs twice or more is not possible, adjust the operating frequency of the 16-bit counter.

Depending on the selected capture input sources and specified edge detection three different measurement methods can be applied.

1. Pulse period measurement
2. Alternating pulse width and pulse space measurement:
   This requires a fast interrupt handling, in order to measure pulse width and pulse space correctly.
3. Simultaneous pulse width and pulse space measurement:
   Both capture inputs are required to measure pulse width and pulse space simultaneously.

The measurements methods are explained in the following sub-chapters.

Caution In the pulse width measurement mode, select the internal clock (TAAnEEE of TAAnCTL1 register = 0).

(1) Pulse period measurement

The pulse period of a signal can be measured in the pulse width measurement mode, when the edge detection of one of the inputs TIAAn0 and TIAAn1 is set either to “rising edge” or “falling edge”. The detection of the other input should be set to “no edge detection”.

By detection of the specified edge the resulting value is captured in the corresponding capture register (TAAnCCR0 or TAAnCCR1), and the timer is cleared and restarts counting.

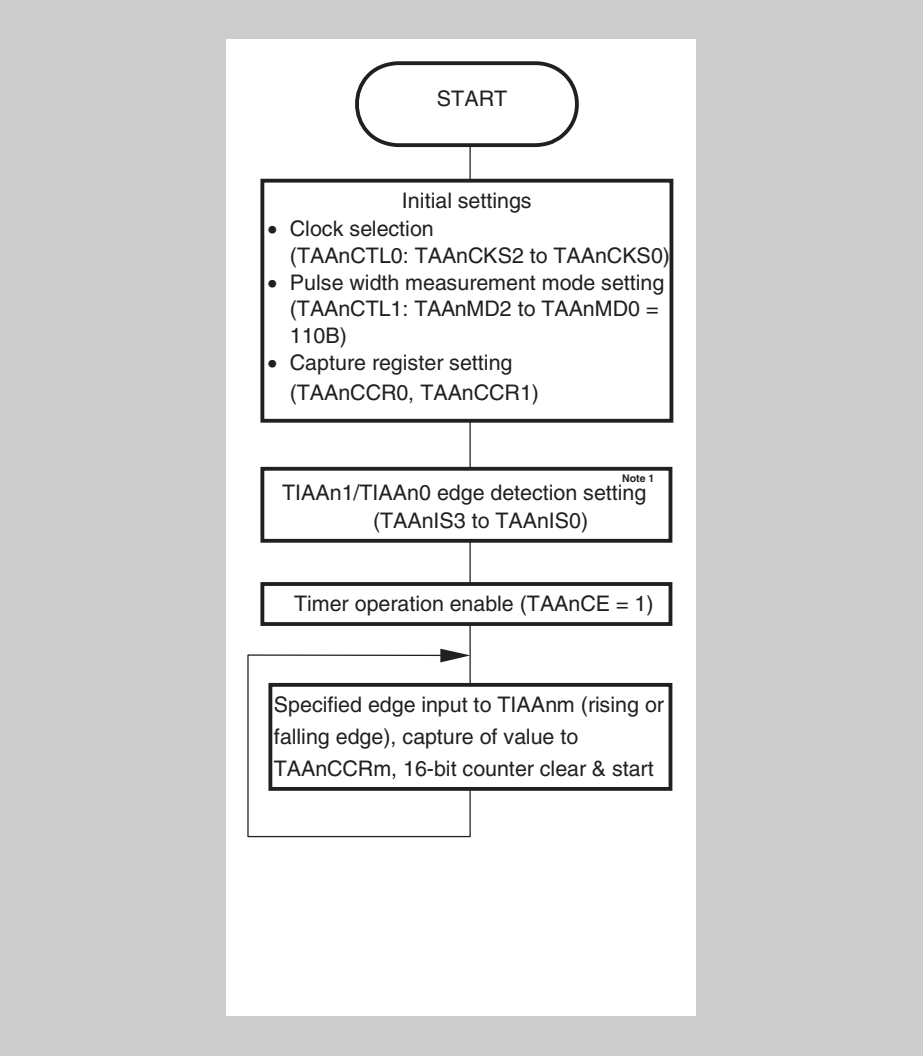


Figure 16-30 Flowchart of pulse period measurement

Note 1. External pulse input is possible for both TIAAn0 and TIAAn1, but only one should be selected for the pulse period measurement.
Specify either “rising edge” or “falling edge” for edge detection. Specify the edge of the external input pulse that is not used as “no edge detection”.

2. n = 0 to 9; m = 0, 1

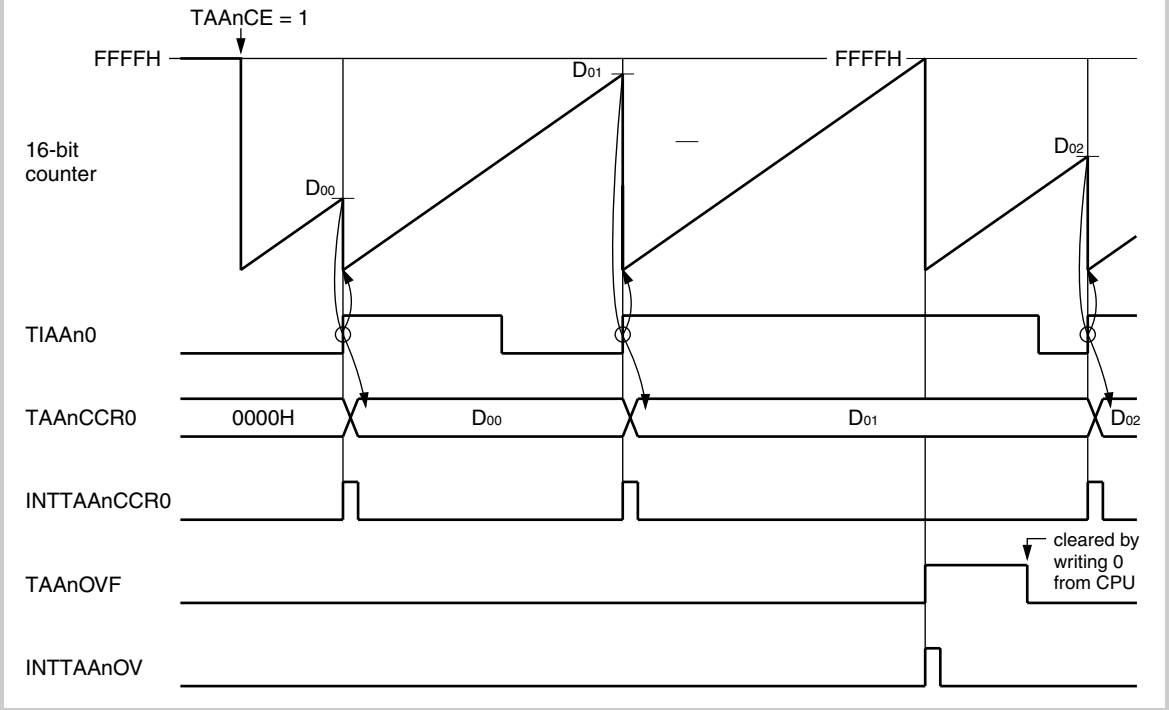


Figure 16-31 Basic operation timing of pulse period measurement

D_{00}, D_{01}, D_{02}: Values captured to TAAnCCR0 register (0000H to FFFFH)

TIAAn0: Set to detection of rising edge (TAAnIS1, TAAnIS0 = 01B)
TIAAn1: Set to no edge detection (TAAnIS3, TAAnIS2 = 00B)

(2) Alternating pulse width and pulse space measurement

The pulse period of a signal can be measured in the pulse width measurement mode alternating in one capture register, when the edge detection of one of the inputs TIAAn0 and TIAAn1 is set to “both rising and falling edges”. The detection of the other input should be set to “no edge detection”.

By detection of a falling or rising edge the resulting value is captured in the corresponding capture register (TAAnCCR0 or TAAnCCR1), and the timer is cleared and restarts counting.

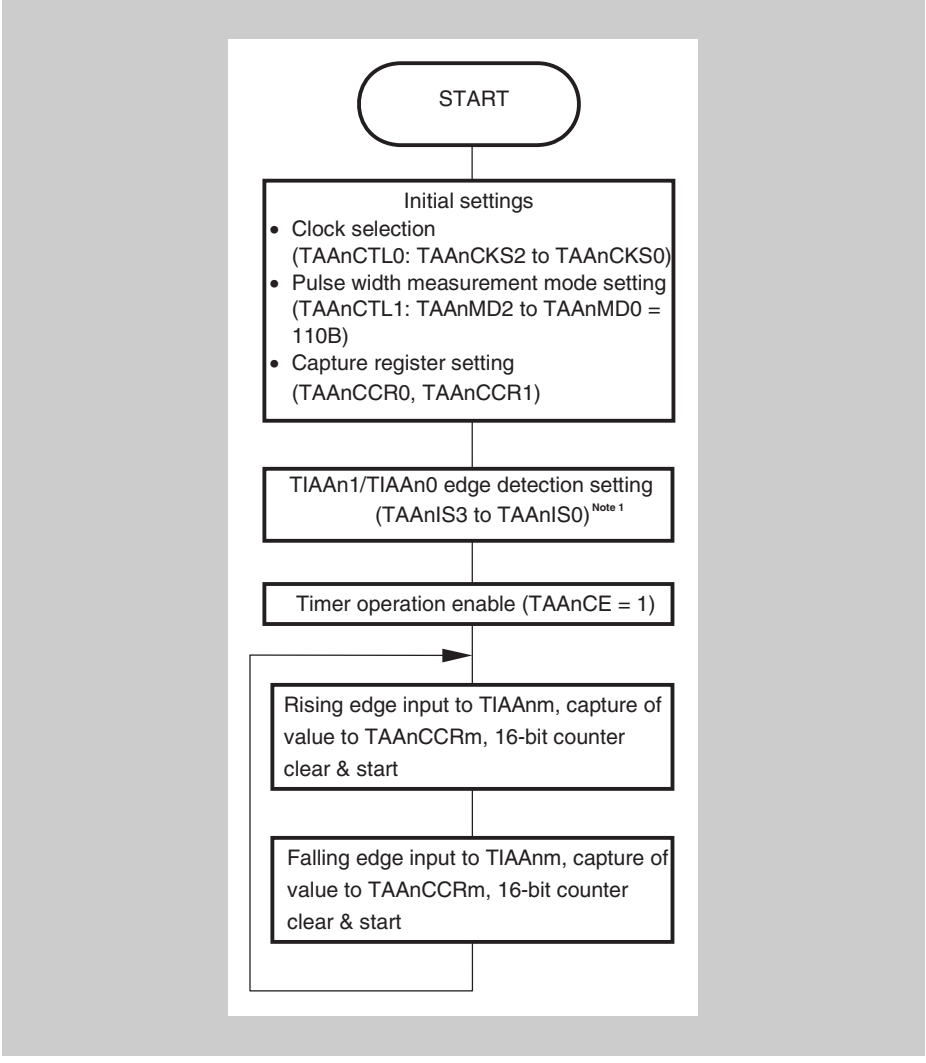


Figure 16-32 Flowchart of alternating pulse width and pulse space measurement

Note 1. External pulse input is possible for both TIAAn0 and TIAAn1, but only one should be selected for the alternating pulse width and pulse space measurement.
Specify “both rising and the falling edges” for edge detection. Specify the edge of the external input pulse that is not used as “no edge detection”.

2. n = 0 to 9; m = 0, 1

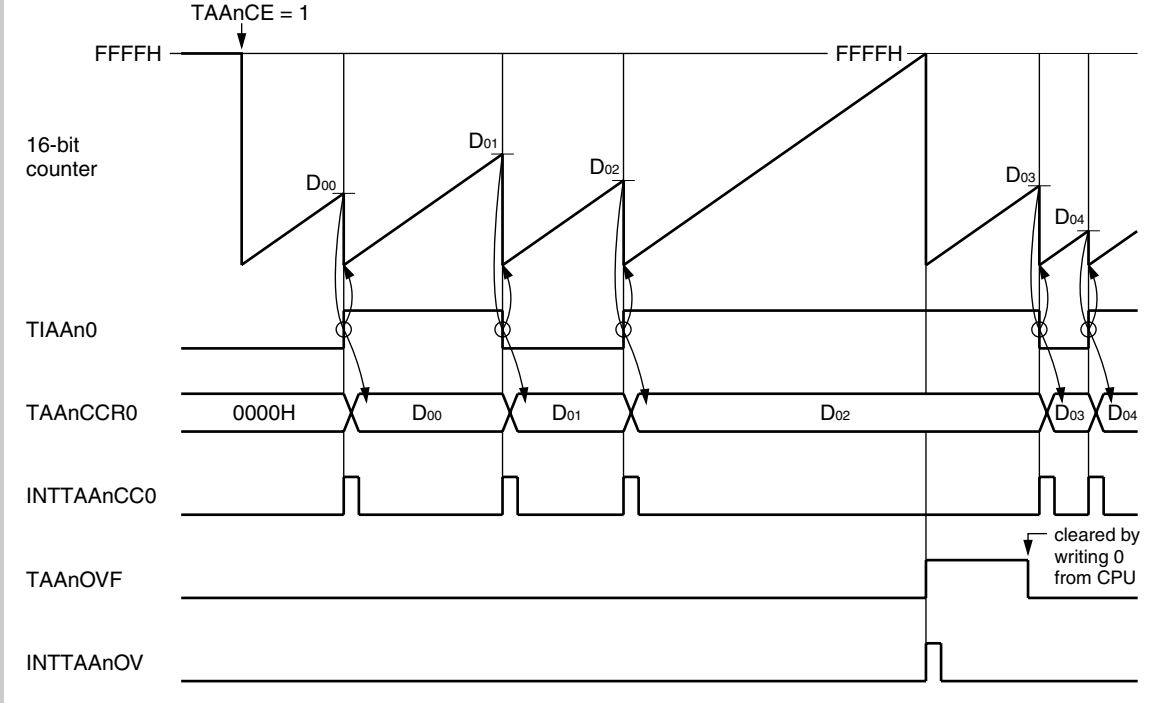


Figure 16-33 Basic operation timing of alternating pulse width and pulse space measurement

D_{00}, D_{01}, D_{02}, D_{03}, D_{04}:
Values captured to TAAnCCR0 register (0000H to FFFFH)

TIAAn0:
Set to detection of both rising and falling edges (TAAnIS1, TAAnIS0 = 11B)
TIAAn1:
Set to no edge detection (TAAnIS3, TAAnIS2 = 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 TIAAn0 and TIAAn1, where both inputs detect opposite edges. Alternatively the signal can be input to TIAAn0 only, when the capture source input selection for capture register 1 is used (refer to *"TAAIC0 - Timer input control register 0" on page 613* and *"TAAIC1 - Timer input control register 1" on page 614* and *"TAAIC2 - Timer input control register 2" on page 615*).

By detection of the specified edge the resulting values of pulse width or pulse space are captured in the corresponding capture registers (TAAnCCR0, TAAnCCR1), and the timer is cleared and restarts counting.

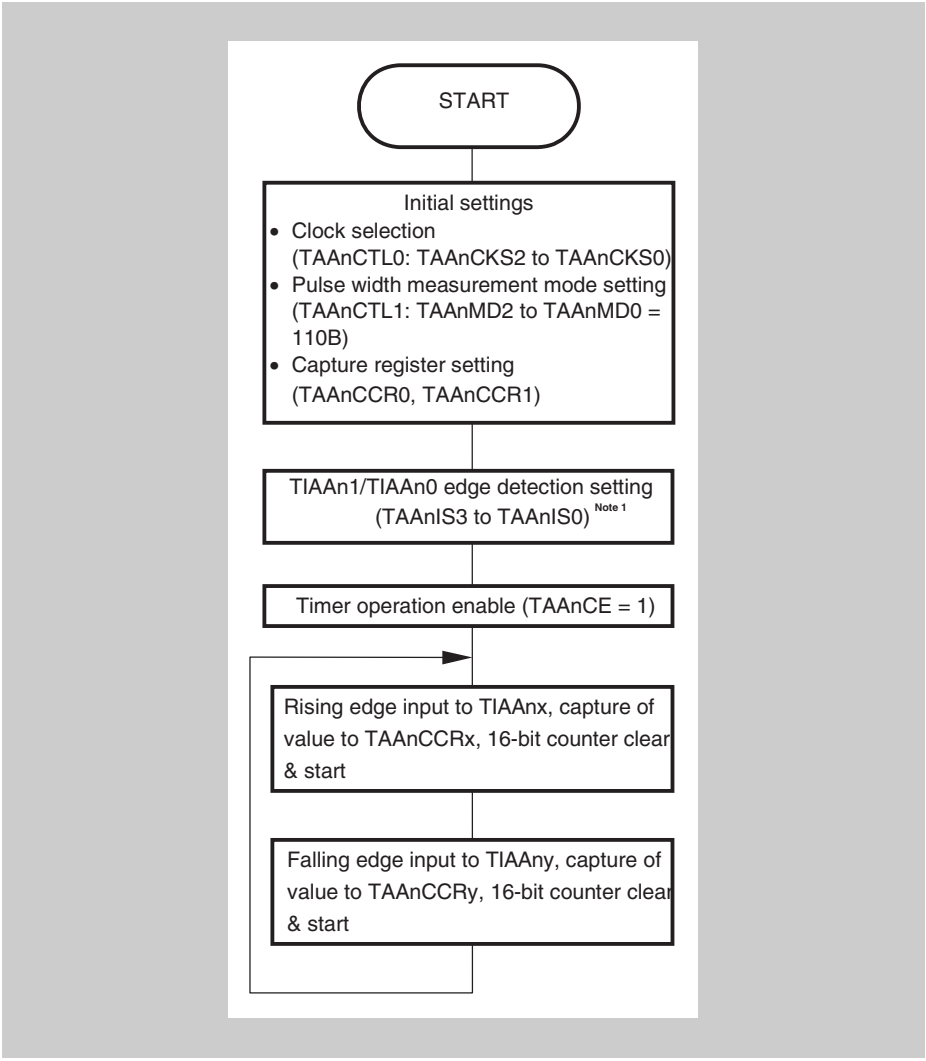


Figure 16-34 Flowchart of simultaneous pulse width and pulse space measurement

Note 1. External pulse input must be input to both TIAAn0 and TIAAn1, or to TIAAn0 only, if the internal connection between both inputs is selected. Specify "rising edge" for edge detection of first input, and "falling edge" for the second input, or vice versa.

2. n = 0 to 9; x = 0, 1
y = 0 when x = 1; y = 1 when x = 0

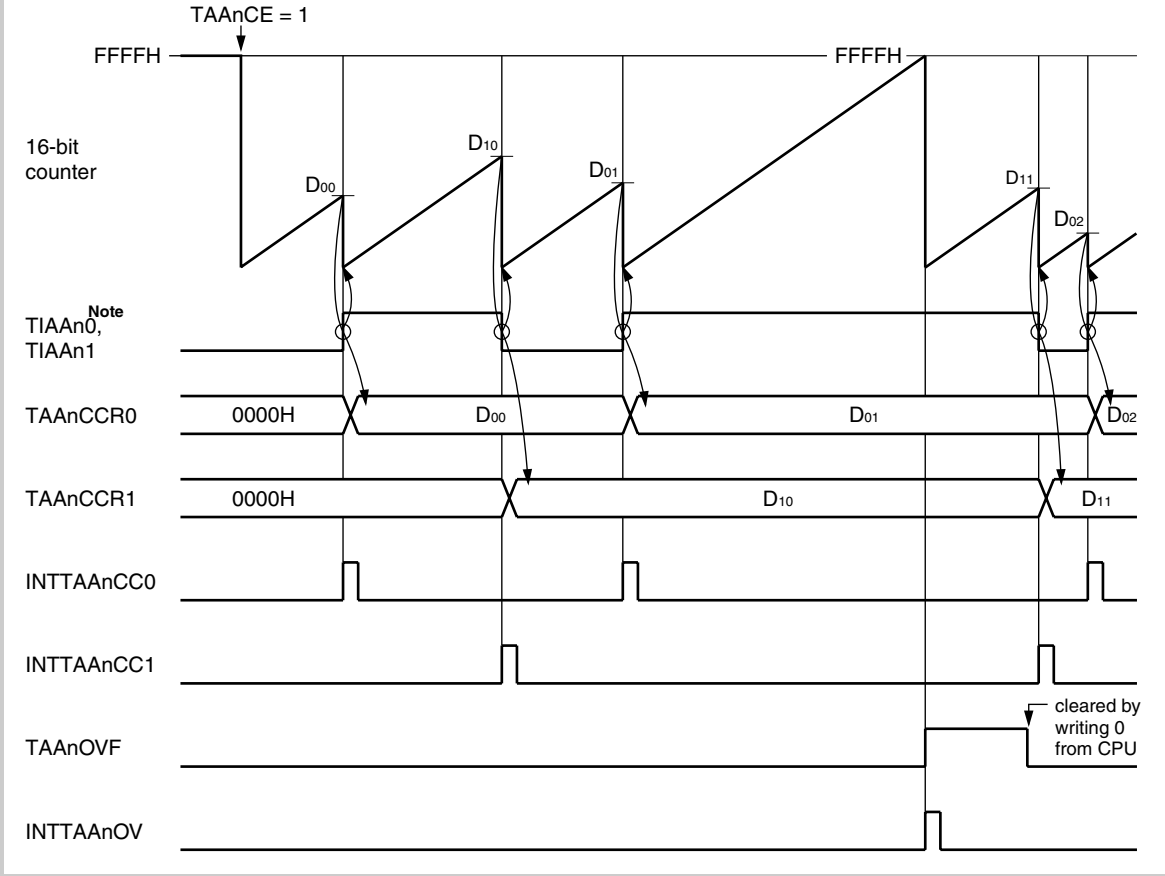


Figure 16-35 Basic operation timing of simultaneous pulse width and pulse space measurement

Note The signal to measure has to be assigned to both inputs, TIAAn0 and TIAAn1. This can be done either by external pin connection, or internally when selecting TIAAn1 input on TIAAn0 pin. In case of internal connection the signal has to be input on TIAAn0 pin.

D_{00}, D_{01}, D_{02}: Values captured to TAAnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Values captured to TAAnCCR1 register (0000H to FFFFH)

TIAAn0: Set detection to rising edge (TAAnIS1, TAAnIS0 = 01B)
TIAAn1: Set detection to falling edge (TAAnIS3, TAAnIS2 = 10B)

16.6.9 32-bit Capture in free-running cascade mode

Two Timer AA (TAA0 in combination with one of TAA1 to TAA3, or TAA4 in combination with one of TAA5 to TAA7, or TAA8 and TAA9) can be cascaded to operate as a 32-bit capture timer. In cascade mode, the timer with the lower number (TAA0, TAA4 or TAA8) is used to control the operation (master timer). Both cascaded timers have to be initialized as free-running timers.

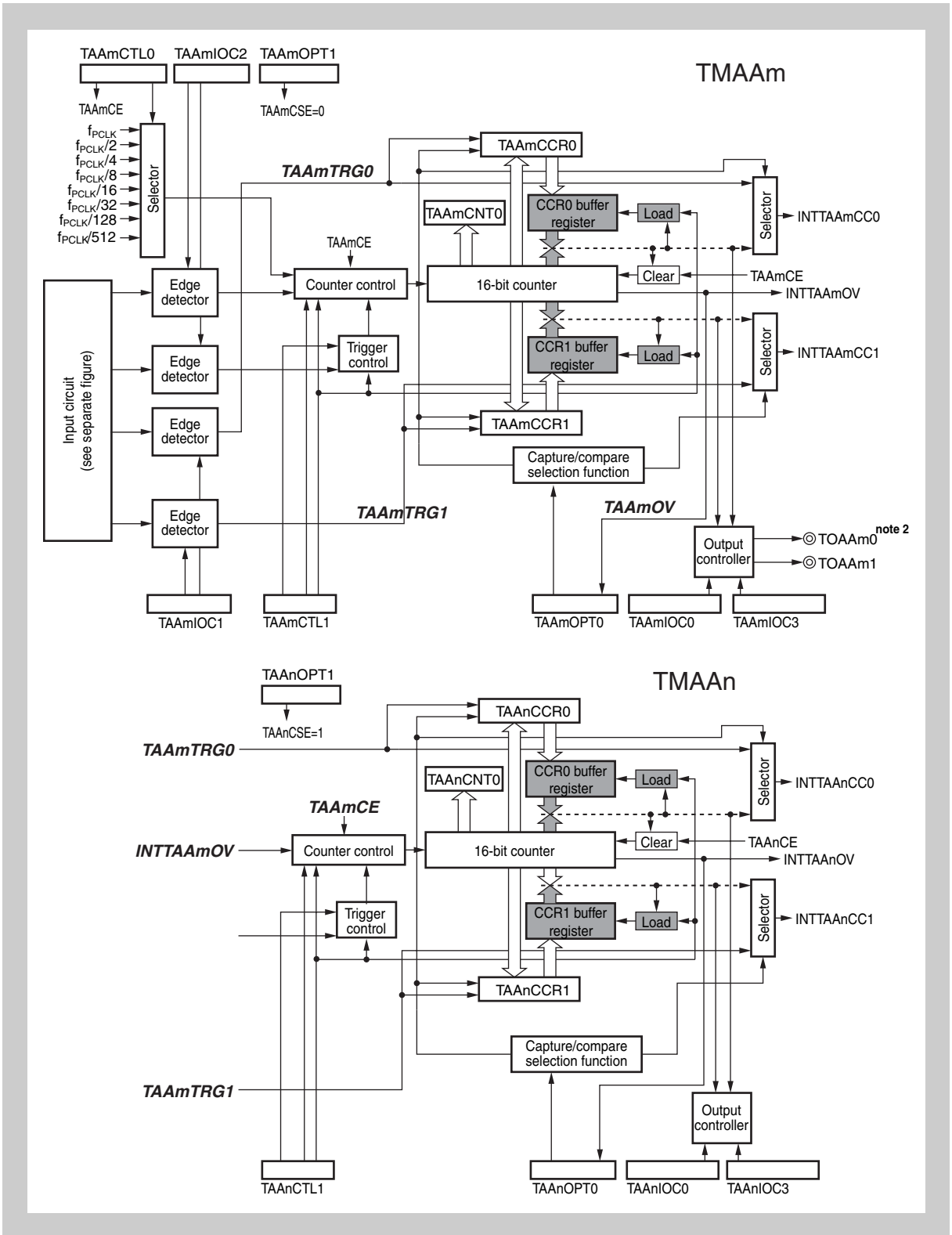


Figure 16-36 Block diagram of TAAm and TAAn in 32-bit capture mode

Note 1. m = 0, 4, or 8
n = 1 to 3, when m = 0
n = 5 to 7, when m = 4
n = 9, when m = 8

2. TOAAm0 is not available for TAA8 (m = 8).

3. Explanation of signals can be found in *Figure 16-1 on page 607*.

4. Block diagrams of the input circuits can be found in *Figure 16-2 on page 608* and *Figure 16-3 on page 608*,

Figure 16-36 shows the block diagram of TAAm and TAAn in cascade mode. Signals that are irrelevant in cascade mode are not shown, the connections to the internal bus are also hidden for better readability of the image, as *Figure 16-1 on page 607* can be used for a in-depth look of each timer.

Note Cascading of two TAA is only allowed for free-running mode with both capture/compare registers set to capture mode. Proper operation of TAAm and TAAn is not guaranteed for any other setting.

Figure 16-37 shows the recommended flow for setting up TAAm and TAAn in cascade mode. As TAAm is used for general control, TAAn is set up first and set in cascaded operation by setting the TAAnCSE bit to 1. Then TAAm is initialized by selecting the proper clock setting and capture trigger input. Only TIAAm0 and TIAAm1 can be used as external capture trigger.

Note When cascading TAAm and TAAn, set TAAnCSE=1 and TAAmCSE=0.

Operation starts when the count enable flag of TAAm (TAAmCE) is set to 1. The counter of TAAm is used for the lower 16-bit of the 32-bit count value, while the upper 16-bit are handled by TAAn.

Whenever the counter of TAAm overflows, the counter is cleared to 0, interrupt INTTAAmOV is generated and the counter of TAAn is incremented by 1. When the counter of TAAn overflows, the counter is also cleared to 0 and interrupt INTTAAnOV is generated.

When a capture trigger 0/1 is detected by TAAm, a capture of the lower 16-bit counter value to TAAmCCR0/1 and of the upper 16-bit counter value to TAAnCCR0/1 at the same time. The interrupts of the TAAm will indicate the capture (INTTAAmCC0/1).

Figure 16-38 on page 667 shows an example of a 32-bit capture timing.

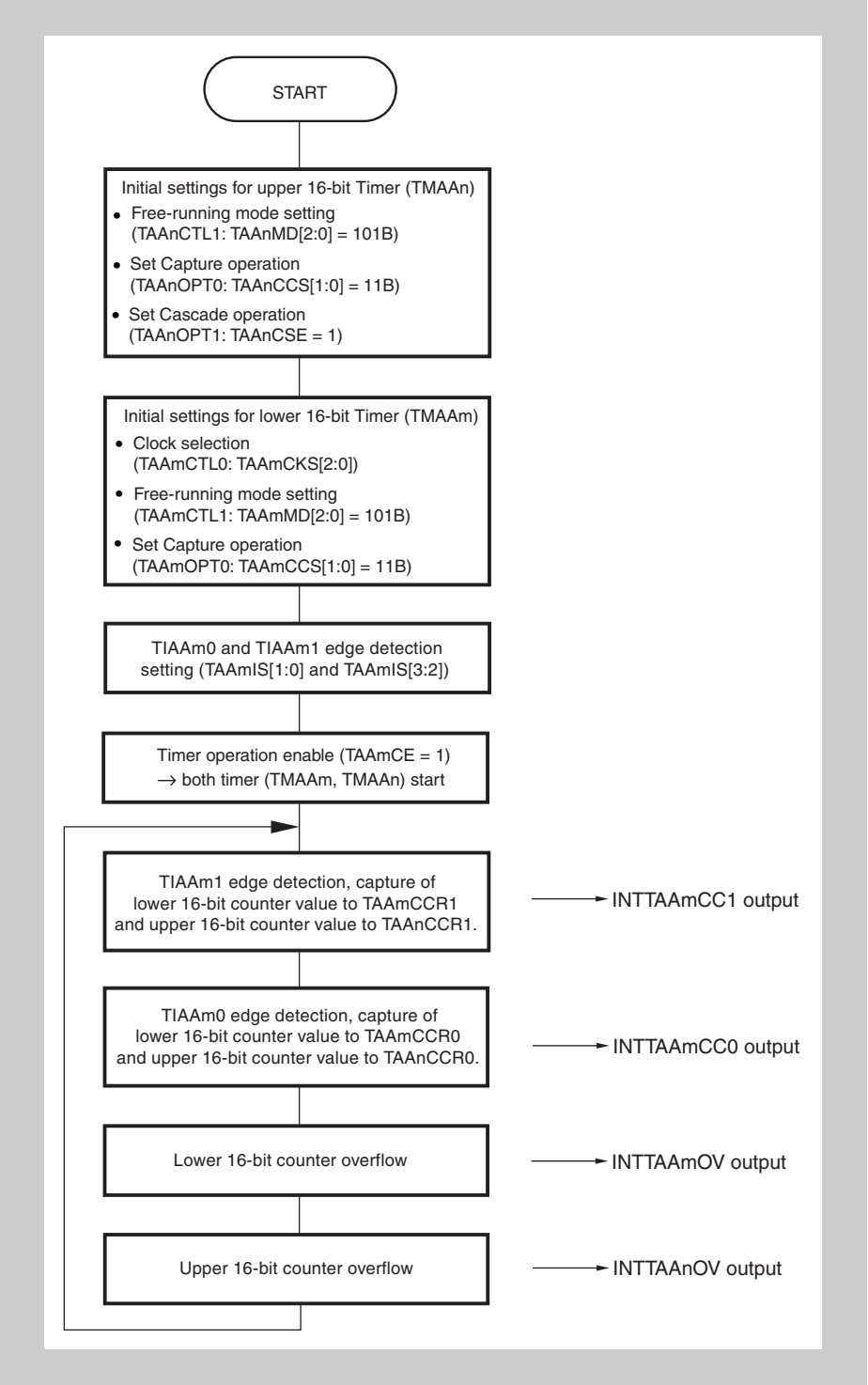


Figure 16-37 Basic flow of 32-bit capture mode

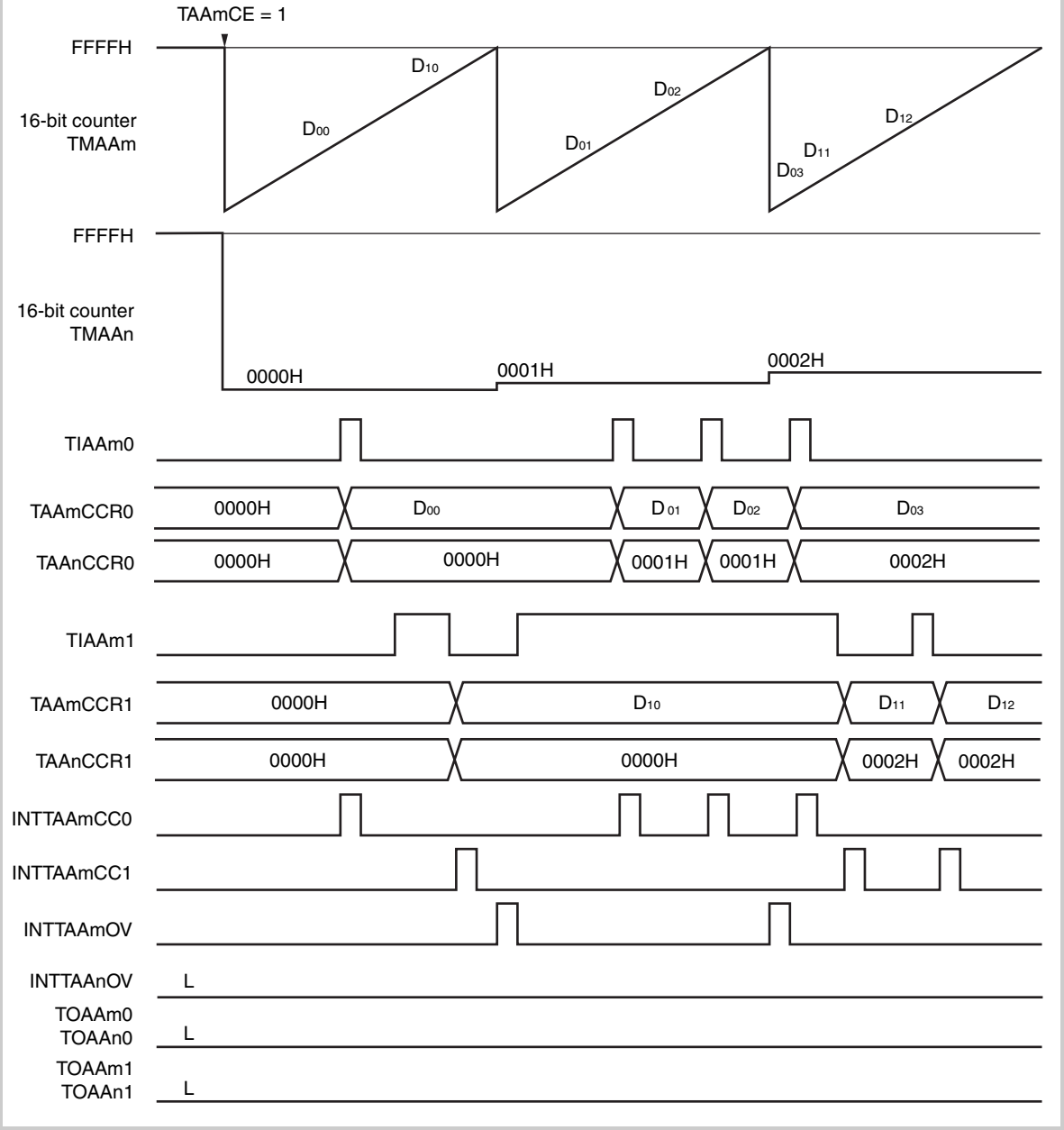


Figure 16-38 Basic timing of 32-bit capture mode

Note m = 0, 4, or 8
n = 1 to 3, when m = 0
n = 5 to 7, when m = 4
n = 9, when m = 8

As the 32-bit resolution is achieved by cascading two individual TAA, a direct read of the 32-bit capture value is not possible. To ensure that the data is not corrupted during read operation, the following procedure for reading needs to be followed:

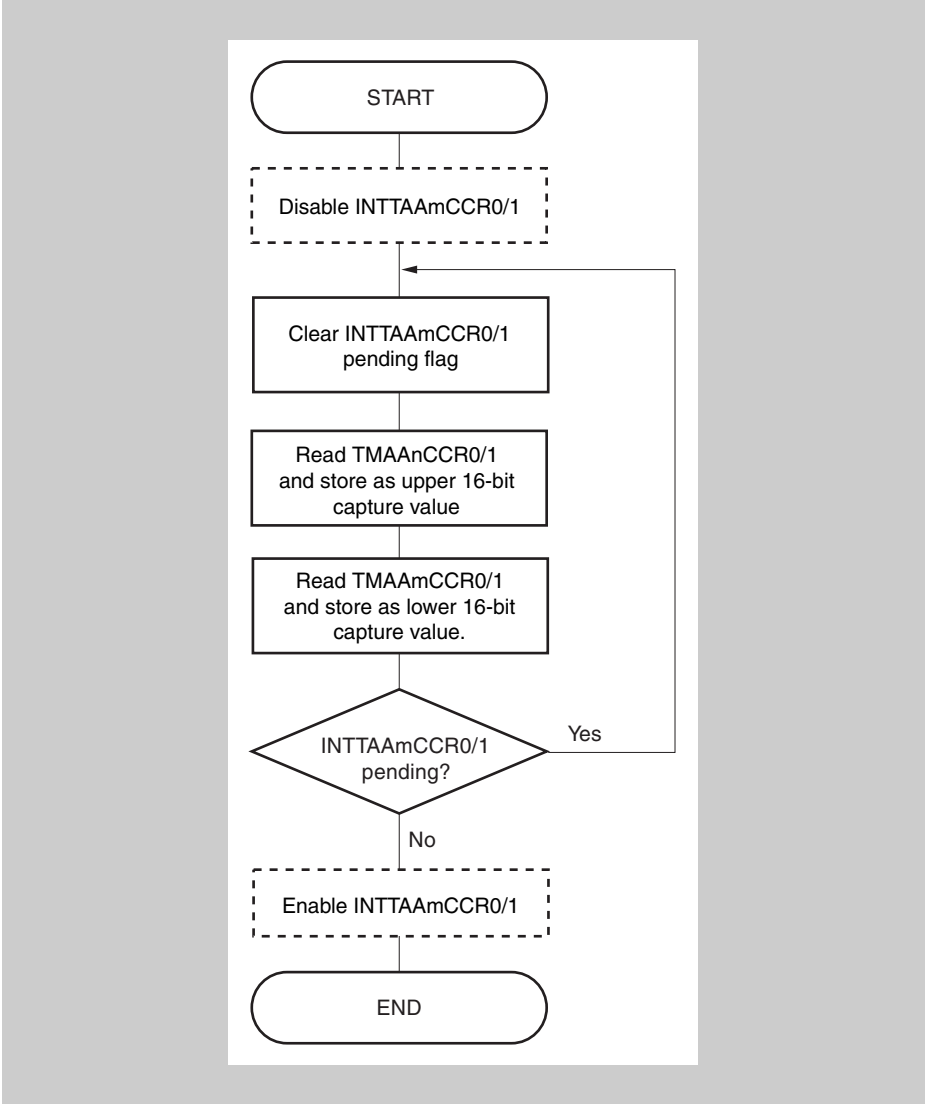


Figure 16-39 Flow of 32-bit read (capture or counter value)

Disabling the capture interrupt (INTTAAmCCR0/1) is not required if the read sequence is done in the interrupt service routine, as nesting of the same interrupt is not possible. However, if the read operation is done in a normal routine while the interrupt signal is also assigned to a interrupt service routine, disabling the interrupt is mandatory, otherwise corrupted data might be read.

The same flow can be used for reading the timer counter value. In this case the relevant interrupt which pending flags needs to be cleared and checked is INTTAAmOV. Please note that you can either read the upper 16-bit counter (TAAnCNT) and then the lower 16-bit counter (TAAmCNT) or vice versa. While both methods work, the read values can be slightly different, as the count operation of the lower 16-bit counter continues while the upper 16-bit timer is read:

- When reading the upper 16-bit first, the lower 16-bit might be incremented during that read.
- When reading the lower 16-bit first, the value might be already “old” after reading the upper 16-bit.

The software programmer needs to decide which method is considered better for the application.

16.6.10 Specific TAAnCCRm overflow value write

This section describes some examples of the specific overflow valu write function.

(1) General operation

Conditions

- TAAnIOC1.TAAnIS[1:0] = 01_B: TIAAn0 rising edge detection
- TAAnIOC1.TAAnIS[3:2] = 10_B: TIAAn1 falling edge detection
- TAAnSVC.TAAnCSD0 = 0: TAAnCCR0 always overwritten upon overflow
- TAAnSVC.TAAnCSD1 = 0: TAAnCCR1 always overwritten upon overflow
- TAAnSVC.TAAnCSR0 = 0: OVFV for TAAnCCR0 not reversed
- TAAnSVC.TAAnCSR1 = 0: OVFV for TAAnCCR1 not reversed
- TAAnSVC.TAAnENCm = 1: overflow value write function enabled
- TAAnSVC.TAAnCSE0 = 0: OVFV = 0x0000, if TIAnm = L at overflow
- TAAnSVC.TAAnCSE1 = 1: OVFV = 0xFFFF, if TIAnm = H at overflow
- TAAnCTL0.TAAnCE = 1: enable TAAn operation

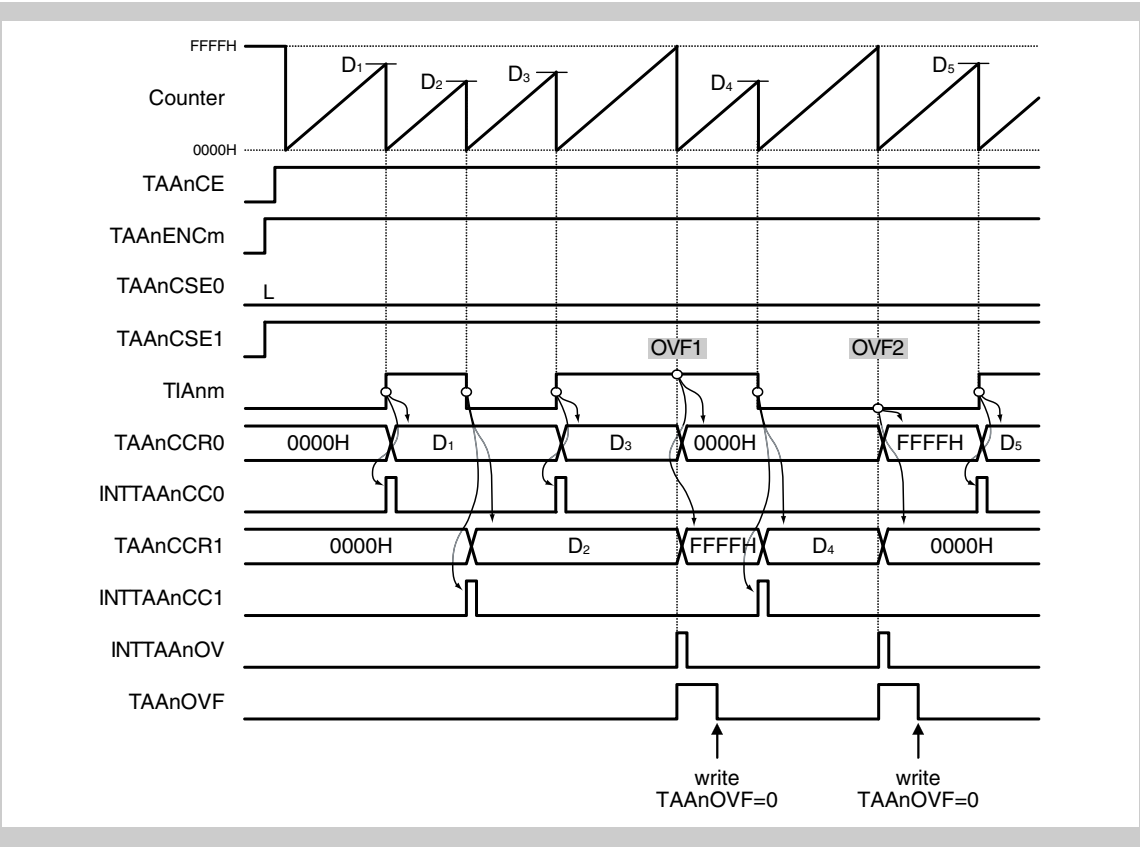


Figure 16-40 Specific overflow value write: general operation

OVF1 TAAnCCR0 = 0x0000, because TAAnCSE0 = L and TIAnm = H

OVF2 TAAnCCR1 = 0xFFFF, because TAAnCSE1 = L and TIAnm = H

(2) Concurrent capture edge and overflow

When an overflow and a valid edge at the TIAnm input is detected concurrently, the capture operation has priority and the the overflow value is not written to TAAnCCRm. Thus no overflow interrupt INTTAAnOV is generated.

Conditions

- TAAnIOC1.TAAnIS[1:0] = 01_B: TIAAn0 rising edge detection
- TAAnIOC1.TAAnIS[3:2] = 10_B: TIAAn1 falling edge detection
- TAAnSVC.TAAnCSD0 = 0: TAAnCCR0 always overwritten upon overflow
- TAAnSVC.TAAnCSD1 = 0: TAAnCCR1 always overwritten upon overflow
- TAAnSVC.TAAnCSR0 = 0: OVFV for TAAnCCR0 not reversed
- TAAnSVC.TAAnCSR1 = 0: OVFV for TAAnCCR1 not reversed
- TAAnSVC.TAAnENCm = 1: overflow value write function enabled
- TAAnSVC.TAAnCSE0 = 0: OVFV = 0x0000, if TIAnm = L at overflow
- TAAnSVC.TAAnCSE1 = 1: OVFV = 0xFFFF, if TIAnm = H at overflow
- TAAnCTL0.TAAnCE = 1: enable TAAn operation

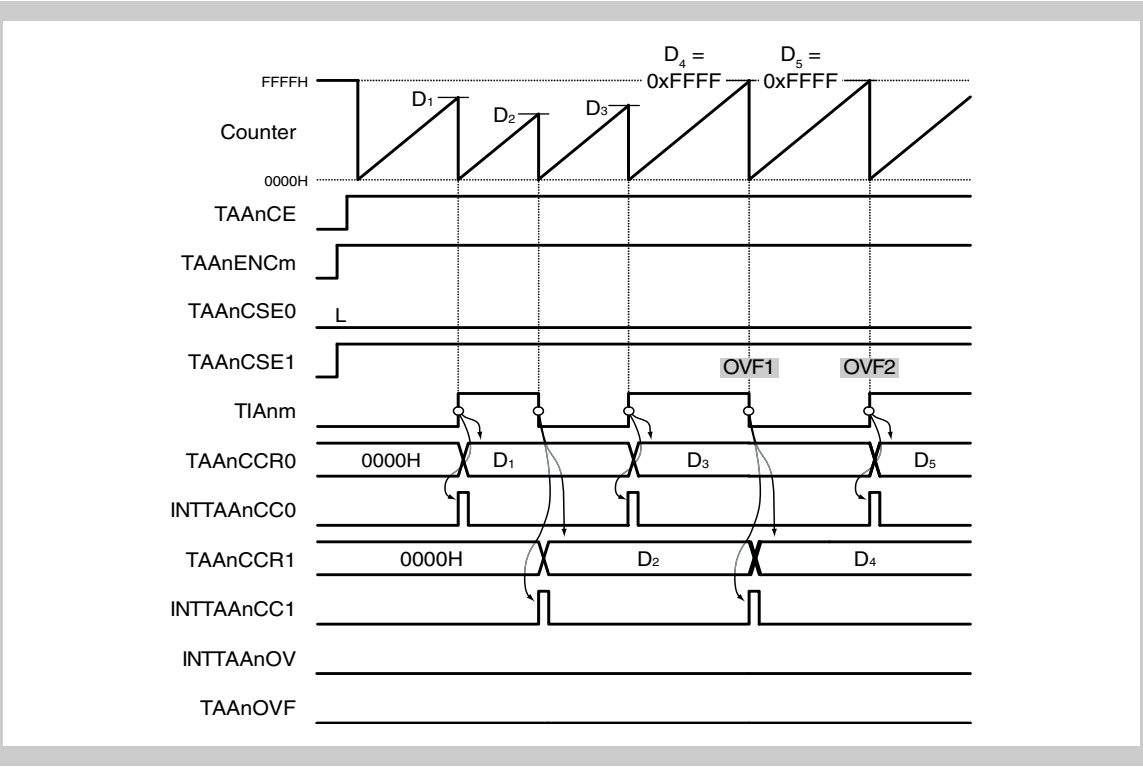


Figure 16-41 Specific overflow value write: concurrent capture edge and overflow

OVF1 Though the counter overflows, D4 = 0xFFFF is captured in TAAnCCR1 because of a valid falling edge at TIAn1. Thus no overflow interrupt INTTAAnOV is generated.

OVF2 Though the counter overflows, D5 = 0xFFFF is captured in TAAnCCR0 because of a valid rising edge at TIAn0.

(3) Capture edge imemdiately after overflow

In case an overflow event is immediately followed by a valid capture edge, i.e. with the next count clock, the specified overflow value remains in the concerned TAAnCCRm register for a single count clock, as the valid capture edge captures stores the counter value to the TAAnCCRm.

Conditions

- TAAnIOC1.TAAnIS[1:0] = 01_B: TIAAn0 rising edge detection
- TAAnIOC1.TAAnIS[3:2] = 10_B: TIAAn1 falling edge detection
- TAAnSVC.TAAnCSD0 = 0: TAAnCCR0 always overwritten upon overflow
- TAAnSVC.TAAnCSD1 = 0: TAAnCCR1 always overwritten upon overflow
- TAAnSVC.TAAnCSR0 = 0: OVFV for TAAnCCR0 not reversed
- TAAnSVC.TAAnCSR1 = 0: OVFV for TAAnCCR1 not reversed
- TAAnSVC.TAAnENCm = 1: overflow value write enabled
- TAAnSVC.TAAnCSE0 = 0: rising edge
- TAAnSVC.TAAnCSE0 = 1: falling edge
- TAAnCTL0.TAAnCE = 1: enable TAAn operation

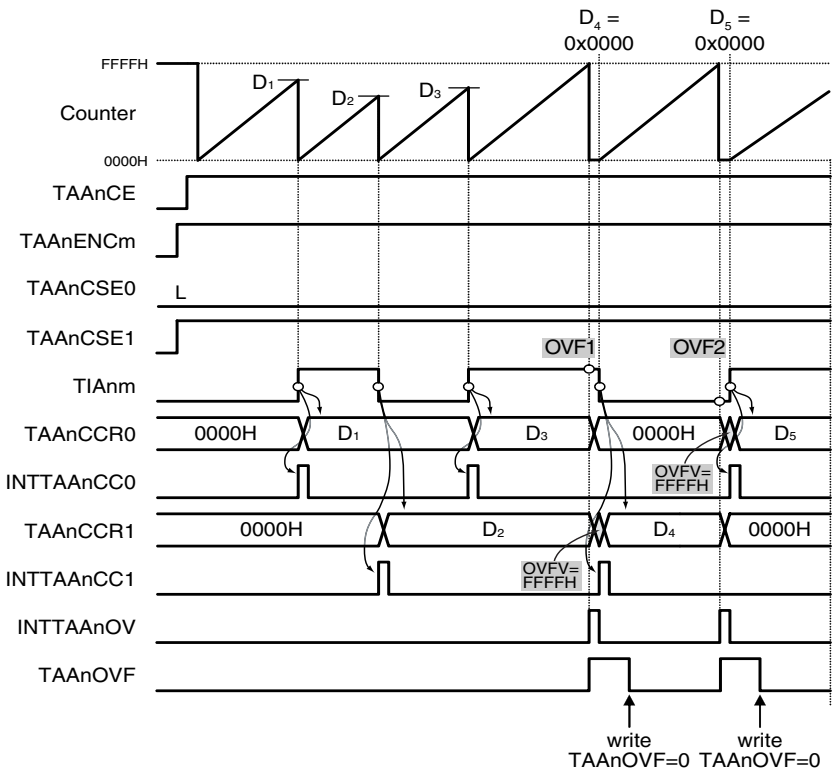


Figure 16-42 Specific overflow value write: capture edge immediately after overflow

OVF1 TAAnCCR0 is set to overflow value OVFV = 0x0000 (TIAnm = H and TAAnCSE0 = 0) and TAAnCCR1 is set to overflow value OVFV = 0xFFFF (TIAnm = H and TAAnCSE1 = 1).
The immediately following falling edge of TIAm, valid for capture channel 1, captures D_4 = 0x0000 in TAACCR1, while TAAnCCR0 remains with the overflow value 0x0000.

OVF2 TAAnCCR0 is set to overflow value OVFV = 0xFFFF (TIAnm = L and TAAnCSE0 = 0) and TAAnCCR1 is set to overflow value OVFV = 0x0000 (TIAnm = L and TAAnCSE1 = 1).
The immediately following rising edge of TIAm, valid for capture channel 0,

captures D_5 = 0x0000 in TAACCR0, while TAAnCCR1 remains with the overflow value 0x0000.

(4) TAAnCCRm overwrite prohibition

Conditions

- TAAnIOC1.TAAnIS[1:0] = 01_B: TIAAn0 rising edge detection
- TAAnIOC1.TAAnIS[3:2] = 10_B: TIAAn1 falling edge detection
- TAAnSVC.TAAnCSD0 = 0: TAAnCCR0 always overwritten upon overflow
- TAAnSVC.TAAnCSD1 = 1: TAAnCCR1 overwritten upon overflow and if TIAnm = not(TAAnCSE1)
- TAAnSVC.TAAnCSR0 = 0: OVFV for TAAnCCR0 not reversed
- TAAnSVC.TAAnCSR1 = 0: OVFV for TAAnCCR1 not reversed
- TAAnSVC.TAAnENCm = 1: overflow value write enabled
- TAAnSVC.TAAnCSE0 = 0: rising edge
- TAAnSVC.TAAnCSE0 = 1: falling edge
- TAAnCTL0.TAAnCE = 1: enable TAAn operation

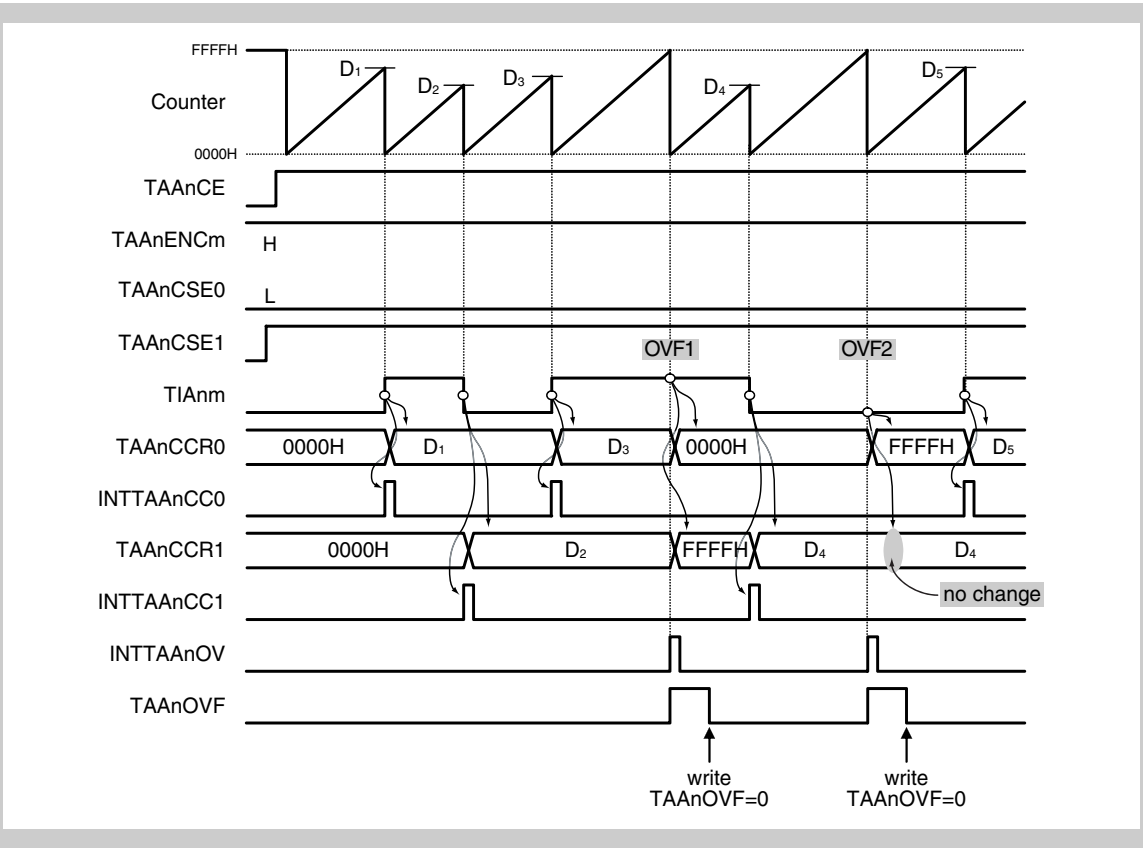


Figure 16-43 Specific overflow value write: TAAnCCRm overwrite prohibition

OVF1 TAAnCCR0 is set to overflow value OVFV = 0x0000 (TIAnm = H and TAAnCSE0 = 0) and TAAnCCR1 is set to overflow value OVFV = 0xFFFF (TIAnm = H and TAAnCSE1 = 1).

OVF2 TAAnCCR0 is set to overflow value OVFV = 0xFFFF (TIAnm = L and TAAnCSE0 = 0).
TAAnCCR1 is not overwritten since TAAnCSD1 = 1, TAInm = L while TAAnCSE1 = 1.

(5) Overflow value write without overflow event

The setting of TAAnSVC.TAAnCSEm does not impact normal capture behaviour, if no overlow occurs.

Conditions

- TAAnIOC1.TAAnIS[1:0] = 01_B: TIAAn0 rising edge detection
- TAAnIOC1.TAAnIS[3:2] = 10_B: TIAAn1 falling edge detection
- TAAnSVC.TAAnENCm = 1: overflow value write enabled

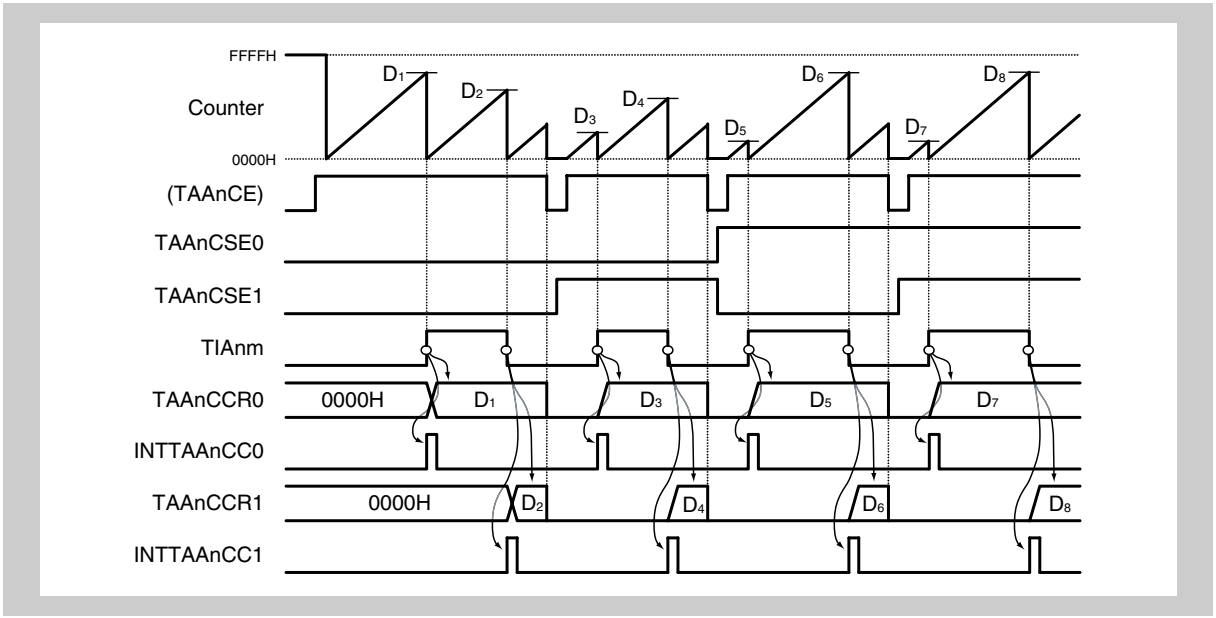


Figure 16-44 Specific overflow value write:

- Changing the value of TAAnSVC.TAAnCSELm has no influence to the normal capture operation, if no overflow event occurs.

16.7 Timer Synchronization Operation Function

Timers AA have a timer synchronized operation function (tuned operation mode). Master timer and incorporated slave timers of the corresponding timer group (listed in *Table 16-3*) start and clock synchronously. When the master timer is cleared, the slave timers are cleared synchronously, too.

Table 16-3 Tuned Operation Mode of Timer

Master Timer	Slave Timer		
TAA0	TAA1	TAA2	TAA3
TAA4	TAA5	TAA6	TAA7

Caution

1. The tuned operation mode is enabled or disabled by the TAAnSYE bit of the TAAnCTL1 register.
2. Set the tuned operation mode using the following procedure.
   - Set the TAAnSYE bit of the TAAnCTL1 register of the slave timer to enable the tuned operation. Set the TAAnMD2 to TAAnMD0 bits of the TAAnCTL1 register of the slave timer to the free-running mode (101B).
   - Set the timer mode of the master timer by using the TAAnMD2 to TAAnMD0 bits of the TAAnCTL1 register to PWM mode (110B), external trigger pulse output mode (010B), or free-running mode (101B). At this time, do not set the TAAnSYE bit of the TAAnCTL1 register of the master timer.
   - Set the compare register value of the master and slave timers.
   - Set the TAAnCE bit of the TAAnCTL0 register of the slave timer to enable operation on the internal operating clock.
   - Set the TAAnCE bit of the TAAnCTL0 register of the master timer to enable operation on the internal operating clock.

Table 16-4 and *Table 16-5* show the timer modes that can be used in the tuned operation mode

(√: settable, ×: not settable).

Table 16-4 Timer modes usable in tuned operation mode

Master Timer	Free-Running Mode	PWM Mode	External Trigger Pulse Output Mode
TAA0	√	√	√
TAA4	√	√	√

Table 16-5 Timer output functions

Tuned Timer Group	Timer	Pin	Free-Running Mode		PWM Mode	
			Tuning OFF	Tuning ON	Tuning OFF	Tuning ON
0	TAA0 (master)	TOAA00	PPG	←	Toggle	←
		TOAA01	PPG	←	PWM	←
	TAA1 (slave)	TOAA10	PPG	←	Toggle	PWM
		TOAA11	PPG	←	PWM	←
	TAA2 (slave)	TOAA20	PPG	←	Toggle	PWM
		TOAA21	PPG	←	PWM	←
	TAA3 (slave)	TOAA30	PPG	←	Toggle	PWM
		TOAA31	PPG	←	PWM	←
1	TAA4 (master)	TOAA40	PPG	←	Toggle	←
		TOAA41	PPG	←	PWM	←
	TAA5 (slave)	TOAA50	PPG	←	Toggle	PWM
		TOAA51	PPG	←	PWM	←
	TAA6 (slave)	TOAA60	PPG	←	Toggle	PWM
		TOAA61	PPG	←	PWM	←
	TAA7 (slave)	TOAA70	PPG	←	Toggle	PWM
		TOAA71	PPG	←	PWM	←

Note The timing of transmitting data from the compare register of the master timer to the compare register of the slave timer is as follows.

PPG: CPU write timing (anytime write mode)

Toggle, PWM: Timing at which timer counter and compare register match TOAAn0 (n = 0 to 9) (reload mode)

Chapter 17 16-Bit Timer/Event Counter S (TMS)

This microcontroller has two instances of the Timer S (TMS), TMS0 to TMS1.

Note Throughout this chapter, the individual instances of Timer S are identified by “n” (n = 0 to 1), for example TSnCTL1 for the Timer Sn control register 1.

17.1 Features

Timer S is a 16-bit timer/counter with Motor Control Functions

- Count clock resolution: Minimum 31.25 ns (using 32 MHz count clock)
- Operation modes supporting general-purpose timer and each motor control method
- Reload buffer-equipped compare register
- 10-bit dead time counter
  - Dead time counter with reload buffer
  - Dead time value independent of positive phase to negative phase or negative phase to positive phase can be set.
- A/D conversion trigger signal generation
  - A/D conversion trigger generated by two compare registers, TSnCCR4 and TSnCCR5
- A/D conversion trigger
  - Individual thinning out function for TSnADTRG0/TSnADTRG1 signal: Thinning out ratio settable to 1/1, 1/2, 1/4, 1/8
  - Dedicated output pins (TOSn7), set with TSnADTRG0 signal and reset with TSnADTRG1 signal
- Interrupt thinning out function
  - Thinning out ratios: 1/1 to 1/32
- Enforced output stop function: ESO
  - Output high impedance of TOSn0 to TOSn7 pins can be executed during ESOn input.
- Compare value setting
  - Reload (batch rewrite function)/anytime rewrite function can be selected**Note**.
- Reload mode
  - Reload is enabled by writing to the TSnCCR1 register (the reload request flag TSnRSF is set) to enable simultaneous transfer of multiple registers
  - Enables transfer at the following reload timings: Peak, trough, and peak and trough**Note**
  - Provides reload request flag TSnRSF
  - Allocates register address enabling DMA transfer
  - Reload thinning out function

- High-accuracy T-PWM mode
  - Enables 0-100% duty PWM output including dead time reduction
  - Improves output resolution without software load because whether or not to use additional pulse(s) in PWM output of count-up operation can be controlled with compare register LSB.
  - Extended operation modes (center method, symmetric interchange method) available for special PWM output behavior, where the duty cycle is split into two phases.
- 180° excitation control
  - Semi-automatic driving function (Enables trigger input from 2-phase encoder and 3-phase encoder)
  - Enables switching from high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode during operation.
- 120° excitation control/Special 120° excitation control
  - Semi-automatic driving function (Enables trigger generation by the offset coordinated with 2-phase encoder, 3-phase encoder, and timer T)
- 3-phase encoder function (Enables signal input such as hall sensor)
- Count clock can be selected from the following 8: f_{PCLK}, $f_{PCLK}/2$, $f_{PCLK}/4$, $f_{PCLK}/8$, $f_{PCLK}/16$, $f_{PCLK}/32$, $f_{PCLK}/128$ and $f_{PCLK}/512$
- The active level of output pins TOSn0 to TOSn7 can be set individually for each pin.
- Fail safe function (Warning/error interrupt occurrence enabled)
  - Simultaneous active output detection function in positive/negative phase
  - Abnormal input detection function for 3-phase encoder

Note High-accuracy T-PWM mode

17.2 Configuration

Timer Sn consists of the following hardware.

Table 17-1 Timer Sn configuration

Item	Configuration
Counter	16-bit counter × 1 16-bit sub-counter × 1 10-bit dead time counter × 3
Register	Timer Sn counter read register (TSnCNT) Timer Sn sub-counter read register (TSnSBC) Timer Sn dead time setting registers 0, 0A, 1, 1A (TSnDTC0, TSnDTC0A, TSnDTC1, TSnDTC1A[a]) Timer Sn compare registers 0 to 5 (TSnCCR0 to TSnCCR5, TSnCCR0A to TSnCCR5A[a]) Timer Sn compare registers 0A to 5A (TSnCCR0 to TSnCCR5, TSnCCR0A to TSnCCR5A[a]) Timer Sn compare registers 1B to 3B (TSnCCR1B to TSnCCR3B) Timer Sn pattern registers 0, 1 (TSnPAT0, TSnPAT1)
Timer input pin	6 pins (TTRGSn, TEVTSn, TAPTSn0 to TAPTSn2, ESOn)[b]
Timer output pin	8 pins (TOSn0 to TOSn7)[b]
Timer input signal	TSnSTCI0, TSnSTCI1, TSnARD
Timer output signal	TSnADTRG0, TSnADTRG1, TSnAEDO, TSnESG, TSnTSF
Control register	Timer Sn control registers 0, 1 (TSnCTL0, TSnCTL1) Timer Sn I/O control registers 0, 2, 4 (TSnIOC0, TSnIOC2, TSnIOC4) Timer Sn option registers 0 to 7 (TSnOPT0 to TSnOPT7) Timer Sn option register 1A (TSnOPT1, TSnOPT1A[a]) Timer Sn extension control register (TSnXCTL)
Interrupt	Compare match interrupt (INTTSnCC0 to INTTSnCC5) Peak interrupt (INTTSnCD0) Trough interrupt (INTTSnOD) Overflow interrupt (INTTSnOV) Error interrupt (INTTSnER) Warning interrupt (INTTSnWN)

a) Throughout this manual the register names with an “A” suffix are functional identical to there counterparts without the “A” suffix. Thus all references to register names without suffix are also valid to the registers with suffix.

b) Alternate-function pin

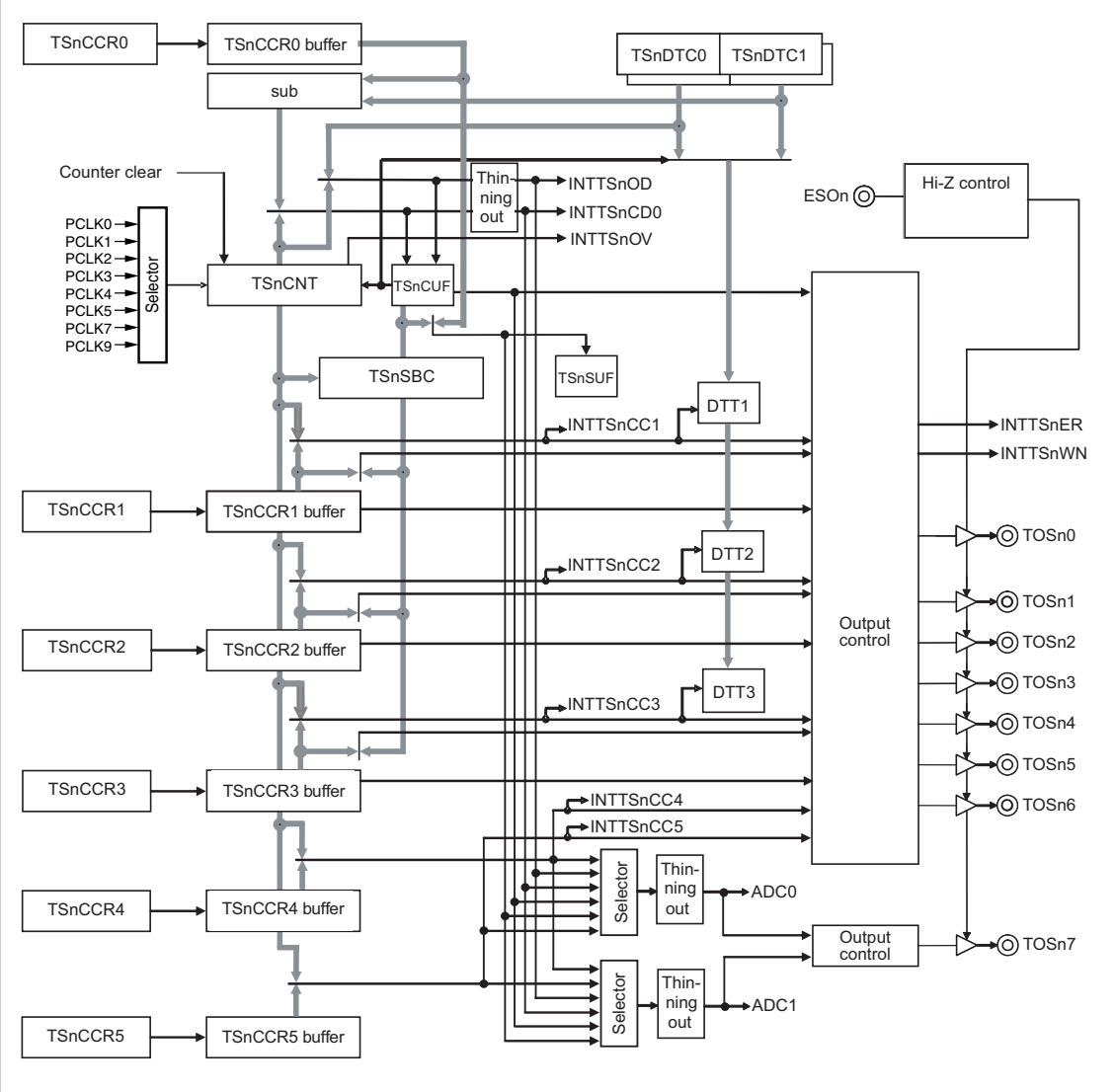


Figure 17-1 Timer Sn block diagram

TSnCCR0 to TSnCCR5:	Compare registers 0 to 5
TSnCCR0 buffer to TSnCCR5 buffer:	Compare buffer registers 0 to 5
TSnDTC0, TSnDTC1:	Timer Sn dead time setting registers 0, 1
TSnSBC:	Timer Sn sub-counter read register
TSnCUF:	Bit 1 of timer Sn option register 0
TSnSUF:	Bit 1 of timer Sn option register 6
INTTSnOD:	Trough interrupt
INTTSnCD0:	Peak interrupt
INTTSnCC1 to INTTSnCC5:	Compare match interrupts 1 to 5
ESOn:	Timer Sn output stop signal input pin
TOSn0 to TOSn7:	Timer output pins 0 to 7
ADC0, ADC1:	A/D Converters 0, 1
DTT1 to DTT3:	Dead time counters
Thinning out:	Thinning out circuit

Note PCLKn: peripheral clocks (refer to *“Clock Generator” on page 179*):

PCLK	0	1	2	3	4	5	7	9
f [MHz]	32	16	8	4	2	1	0.25	0.0625

(1) Timer Sn compare register 0, 0A (TSnCCR0, TSnCCR0A)

The TSnCCR0 and TSnCCR0A registers are 16-bit compare registers. The TSnCCR0A register is a mirror of the TSnCCR0 register for optimized DMA support.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input sets these registers to 0000H.

Caution Write to bit 0 of the TSnCCR0 register, or TSnCCR0A respectively, is ignored in high-accuracy T-PWM mode.

Bit 0 reads 0.

Address: TS0CCR0: FFFFF598H, TS1CCR0: FFFFF5D8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR0																	0000H

Address: TS0CCR0A: FFFFFB50H, TS1CCR0A: FFFFFBD0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR0A																	0000H

The write access method for the TSnCCR0 register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR0 register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Caution The TSnCCR0 register, or TSnCCR0A respectively, setting must be as follows to set the carrier wave cycle in the high-accuracy T-PWM mode.

TSnCCR0 or TSnCCR0A register value =
Count clock number of carrier wave cycle
+ TSnDTC0 orTSnDTC0A register value
+ TSnDTC1 or TSnDTC1A register value

Refer to *"16-bit counter operation in high-accuracy T-PWM mode" on page 873* for the carrier wave and dead time setting.

Note
1. Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.
2. Throughout this manual the register name TSnCCR0 means also the register TSnCCR0A.

(2) Timer Sn compare register 1, 1A (TSnCCR1, TSnCCR1A)

The TSnCCR1 and TSnCCR1A registers are 16-bit compare registers. The TSnCCR1A register is a mirror of the TSnCCR1 register for optimized DMA support.
When a compare register is rewritten in the reload mode, the reload request flag (TSnRSF) of TSnOPT6 is set to 1 by write access to the TSnCCR1 register or TSnCCR1A register, and then all of the TSnCCR0 to TSnCCR5, TSnCCR0A to TSnCCR5A, TSnCCR1B to TSnCCR3B, TSnDTC0, TSnDTC1, TSnOPT1, TSnPAT0, and TSnPAT1 registers are rewritten at the next reload timing.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input sets these registers to 0000H.

Note Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR1 register or TSnCCR1A register in the high-accuracy T-PWM mode. Refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 877* for the additional pulse control function.

Address: TS0CCR1: FFFFF59EH, TS1CCR1: FFFFF5DEH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR1																	0000H

Address: TS0CCR1A: FFFFFB56H, TS1CCR1A: FFFFFBD6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR1A																	0000H

The write access method for the TSnCCR1/TSnCCR1A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR1/TSnCC1A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note 1. Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

2. Throughout this manual the register name TSnCCR1 means also the register TSnCCR1A.

(3) Timer Sn compare register 1B (TSnCCR1B)

The TSnCCR1B register ia a 16-bit compare register, which is used by the extension operation mode of the high-accuracy T-PWM mode.
This register can be read/written with a 16-bit manipulation instruction.
Reset input sets this register to 0000H.

Note Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR1B register in the high-accuracy T-PWM mode. Refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 877* for the additional pulse control function.

Address: TS0CCR1B: FFFFFB4EH, TS1CCR1B: FFFFFBCEH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR1B																	0000H

The write access method for the TSnCCR1B register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR1B register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

(4) Timer Sn compare register 2, 2A (TSnCCR2, TSnCCR2A)

The TSnCCR2 and TSnCCR2A registers are 16-bit compare registers. The TSnCCR2A register is a mirror of the TSnCCR2 register for optimized DMA support.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input set these registers to 0000H.

Note The additional pulse control function operates if 1 is set to bit 0 of the TSnCCR2 register or TSnCCR2A register in the high-accuracy T-PWM mode. Refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 877* for the additional pulse control function.

Address: TS0CCR2: FFFFF59CH, TS1CCR2: FFFFF5DCH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR2																	0000H

Address: TS0CCR2A: FFFFFB54H, TS1CCR2A: FFFFFBD4H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR2A																	0000H

The write access method for the TSnCCR2/TSnCCR2A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR2/TSnCCR2A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note 1. Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

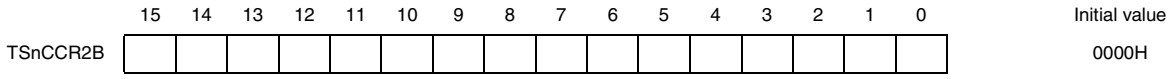
2. Throughout this manual the register name TSnCCR2 means also the register TSnCCR2A.

(5) Timer Sn compare register 2B (TSnCCR2B)

The TSnCCR2B register ia a 16-bit compare register, which is used by the extension operation mode of the high-accuracy T-PWM mode.
This register can be read/written with a 16-bit manipulation instruction.
Reset input sets this register to 0000H.

Note Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR2B register in the high-accuracy T-PWM mode. Refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 877* for the additional pulse control function.

Address: TS0CCR2B: FFFFFB4CH, TS1CCR2B: FFFFFBCCH



The write access method for the TSnCCR2B register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR2B register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

(6) Timer Sn compare register 3, 3A (TSnCCR3, TSnCCR3A)

The TSnCCR3 and TSnCCR3A registers are 16-bit compare registers. The TSnCCR3A register is a mirror of the TSnCCR3 register for optimized DMA support.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input set these registers to 0000H.

Note Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR3 register or TSnCCR3A register in the high-accuracy T-PWM mode. Refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 877* for the additional pulse control function.

Address: TS0CCR3: FFFFF59AH, TS1CCR3: FFFFF5DAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR3																	0000H

Address: TS0CCR3A: FFFFFB52H, TS1CCR3A: FFFFFBD2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR3A																	0000H

The write access method for the TSnCCR3/TSnCCR3A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR3B register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note 1. Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

2. Throughout this manual the register name TSnCCR3 means also the register TSnCCR3A.

(7) Timer Sn compare register 3B (TSnCCR3B)

The TSnCCR3B register ia a 16-bit compare register, which is used by the extension operation mode of the high-accuracy T-PWM mode.
This register can be read/written with a 16-bit manipulation instruction.
Reset input sets this register to 0000H.

Note Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR3B register in the high-accuracy T-PWM mode. Refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 877* for the additional pulse control function.

Address: TS0CCR3B: FFFFFB4AH, TS1CCR3B: FFFFFBCAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR3B																	0000H

The write access method for the TSnCCR3B register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR3B register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

(8) Timer Sn compare register 4, 4A (TSnCCR4, TSnCCR4A)

The TSnCCR4 and TSn CCR4A registers are 16-bit compare registers. The TSnCCR4A register is a mirror of the TSnCCR4 register for optimized DMA support.
The match interrupt signal of the 16-bit counter and the TSnCCR4 or TSnCCR4A register can be selected as the A/D conversion trigger in the high-accuracy T-PWM mode, PWM mode with dead time, special 120° excitation mode, and special pattern output mode.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input set these registers to 0000H.

Note Bit 0 of the TSnCCR4 and TSnCCR4A registers is ignored in the high-accuracy T-PWM mode.

Address: TS0CCR4: FFFFF592H, TS1CCR4: FFFFF5D2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR4																	0000H

Address: TS0CCR4A: FFFFFB48H, TS1CCR4A: FFFFFBC8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR4A																	0000H

The write access method for the TSnCCR4/TSnCCR4A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR4/TSnCCR4A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note 1. Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

2. Throughout this manual the register name TSnCCR4 means also the register TSnCCR4A.

(9) Timer Sn compare register 5, 5A (TSnCCR5, TSnCCR5A)

The TSnCCR5 and TSnCCR5A registers are 16-bit compare register. The TSnCCR5A register is a mirror of the TSnCCR5 register for optimized DMA support.
The match interrupt signal of the 16-bit counter and the TSnCCR5 register, or TSnCCR5A register respectively, can be selected as the A/D conversion trigger in the high-accuracy T-PWM mode, PWM mode with dead time, special 120° excitation mode, and special pattern output mode.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input set these registers to 0000H.

Note Bit 0 of the TSnCCR5 or TsnCCR5A register is ignored in the high-accuracy T-PWM mode.

Address: TS0CCR5: FFFFF590H, TS1CCR5: FFFFF5D0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR5																	0000H

Address: TS0CCR5A: FFFFFB46H, TS1CCR5A: FFFFFBC6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCCR5A																	0000H

The write access method for the TSnCCR5/TSnCCR5A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR5/TSnCCR5A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

Note 1. Refer to *"Compare register rewriting" on page 761* for details on the compare register rewriting operation.

2. Throughout this manual the register name TSnCCR5 means also the register TSnCCR5A.

(10) Timer Sn counter read register (TSnCNT)

The TSnCNT register is a timer read register that can read 16-bit counter values.
This is a read-only register read with a 16-bit manipulation instruction.
Reset input or TSnCE of the TSnCTL0 register = 0 sets this register to 0000H.
The TSnCNT register is set to FFFFH (or FFFEH in the high-accuracy T-PWM mode) in the interval from when from TSnCE of the TSnCTL0 register = 1 to the count-up operation.

Address: TS0CNT: FFFFF5A6H, TS1CNT: FFFFF5E6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnCNT																Note	0000H

Note Bit 0 reads 0 in the high-accuracy T-PWM mode.

(11) Timer Sn sub-counter read register (TSnSBC)

The TSnSBC register can read the 16-bit counter value.
This is a read-only register read with a 16-bit manipulation instruction.
Reset input or clearing TSnCE of the TSnCTL0 register to 0 sets this register to 0000H.

Note Operable only in the high-accuracy T-PWM mode or PWM mode with dead-time.

Address: TS0SBC: FFFFF5A8H, TS1SBC: FFFFF5E8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnSBC																Note	0000H

Note Bit 0 reads 0 in the high-accuracy T-PWM mode.

(12) Timer Sn dead time setting register 0, 0A (TSnDTC0, TSnDTC0A)

The TSnDTC0 and TSnDTC0A registers are 10-bit register which specify the dead time value. The TSnDTC0A register is a mirror of the TSnDTC0 register for optimized DMA support.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input sets these register to 0000H.

Caution

1. The dead time counter is enabled only in the PWM mode (TSnDSE of the TSnOPT0 register is set to 1), high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode (TSnDSE of the TSnOPT0 register is set to 1). In the other modes, be sure to set the TSnDTC0 register, or TSnDTC0A register respectively, to 0000H.
2. Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* regardless of any operation mode of timer Sn when TSnDTC0, or TSnDTC0A register respectively, is rewritten to another value during operation of timer Sn (when TSnCE of the TSnCTL0 register is set to 1).
3. Dead time is not inserted when the TSnDTC0 register, or TSnDTC0A register respectively, is set to 0000H.
4. Bits 15 to 10 and 0 are fixed to 0.

Address: TS0DTC0: FFFFF58EH, TS1DTC0: FFFFF5CEH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnDTC0	0	0	0	0	0	0										0	0000H

Address: TS0DTC0A: FFFFFB44H, TS1DTC0A: FFFFFBC4H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnDTC0A	0	0	0	0	0	0										0	0000H

Note Throughout this manual the register name TSnDTC0 means also the register TSnDTC0A.

(13) Timer Sn dead time setting register 1, 1A (TSnDTC1, TSnDTC1A)

The TSnDTC1 and TSnDTC1A registers are 10-bit registers which specify the dead time value. The TSnDTC1A register is a mirror of the TSnDTC1 register for optimized DMA support.
These registers can be read/written with a 16-bit manipulation instruction.
Reset input sets these registers to 0000H.

Caution

1. The dead time counter is enabled only in the PWM mode (TSnDSE of the TSnOPT0 register is set to 1), high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode (TSnDSE of the TSnOPT0 register is set to 1). In the other modes, be sure to set the TSnDTC1, or TSnDTC1A register respectively, register to 0000H.
2. Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* regardless of any operation mode of timer Sn when TSnDTC1, or TSnDTC1A register respectively, is rewritten to another value during operation of timer Sn (when TSnCE of TSnCTL0 register is set to 1).
3. Dead time is not inserted when the TSnDTC1 register, or TSnDTC1A register respectively, is set to 0000H.
4. Bits 15 to 10 and 0 are fixed to 0.

Address: TS0DTC1: FFFFF58CH, TS1DTC1: FFFFF5CCH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnDTC1	0	0	0	0	0	0										0	0000H

Address: TS0DTC1A: FFFFFB42H, TS1DTC1A: FFFFFBC2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
TSnDTC1A	0	0	0	0	0	0										0	0000H

Note Throughout this manual the register name TSnDTC1 means also the register TSnDTC1A.

17.3 Control Register

(1) Timer Sn control register 0 (TSnCTL0)

The TSnCTL0 register is an 8-bit register that controls the timer Sn operation. This register can be read/written with an 8-bit or 1-bit manipulation instruction. Reset input sets this register to 00H.

Caution Only the TSnCE bit of the TSnCTL0 register can be changed during operation of timer Sn (when the value of TSnCE of the TSnCTL0 register is 1). Do not rewrite any other bit than the TSnCE bit.

Address: TS0CTL0: FFFFF580H, TS1CTL0: FFFFF5C0H

	7	6	5	4	3	2	1	0	Initial value
TSnCTL0	TSnCE	0	0	0	0	TSnCKS2	TSnCKS1	TSnCKS0	00H

Bit position	Bit name	Function
7	TSnCE	Specifies the timer Sn operation. 0: Count disabled (Reset timer Sn to asynchronous) 1: Count clock operation enabled When setting the TSnCE bit to 0, the internal operation clock of timer Sn stops and resets timer Sn to asynchronous. When the TSnCE bit is set to 1, the internal operation of timer Sn is enabled and starts count-up operation. See *Table 17-2 on page 695*. **Note:** The functions initialized by TSnCE = 0 are as follows. • Internal register and internal latch circuit except for register writable from CPU • TSnOVF flag, flags of the TSnOPT6 register • 16-bit counter, 16-bit sub-counter, dead time counter, the TSnCNT register and TSnSBC register • TSnCCR0 to TSnCCR5 buffer registers, TSnPAT0 buffer register, TSnPAT1 buffer register, TSnDTC0 buffer register and TSnDTC1 buffer register Timer output (inactive level output)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>TSnCKS2
TSnCKS1
TSnCKS0</td><td>Selects the count clock of timer Sn.
<table>
<tr><th>TSnCKS2</th><th>TSnCKS1</th><th>TSnCKS0</th><th>Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>PCLK0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>PCLK1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>PCLK2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>PCLK3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>PCLK4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>PCLK5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>PCLK7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>PCLK9</td></tr>
</table>
Note: 1. Set the TSnCKS2 to TSnCKS0 bits while TSnCE = 0. The TSnCKS2 to TSnCKS0 bits can be changed simultaneously when the setting of TSnCE is changed from 0 to 1.
2. The TSnCKS2 to TSnCKS0 bits are valid only when TSnEEE of the TSnCTL1 register is set to 0.
3. f_{PCLKn}: Peripheral I/O system clock
f_{TMSn}: Timer Sn base clock ($f_{TMSn} = f_{PCLK0}$)</td></tr>
</table>

Table 17-2 Timer Sn Count Clock and Time from TSnCE = 1 Setting to Count-up Operation

Count clock	TSnCKS2	TSnCKS1	TSnCKS0	Min.	Max.
PCLK0	0	0	0	4 base clocks	5 base clocks
PCLK1	0	0	1		
PCLK2	0	1	0		
PCLK3	0	1	1	4 base clocks	5 base clocks + 1 count clock
PCLK4	1	0	0		
PCLK5	1	0	1		
PCLK7	1	1	0		
PCLK9	1	1	1		

Note Base clock: f_{TMSn}

(2) Timer Sn control register 1 (TSnCTL1)

The TSnCTL1 register is an 8-bit register that controls the timer Sn operation. This register can be read/written with an 8-bit or 1-bit manipulation instruction. Reset input sets this register to 00H.

Caution Only TSnEST bit can be changed during the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1). Do not rewrite any other bit than the TSnEST bit. However, write access of the same value is possible.

Address: TS0CTL1: FFFFF581H, TS1CTL1: FFFFF5C1H

	7	6	5	4	3	2	1	0	Initial value
TSnCTL1	TSnSYE	TSnEST	TSnEEE	0	TSnMD3	TSnMD2	TSnMD1	TSnMD0	00H

Caution The default value 0 of bit 4 must not changed.

Bit position	Bit name	Function
7	TSnSYE	Selects the operation mode of timer Sn. 0: Timer Sn stand-alone operation mode 1: Synchronous operation mode (slave) **Note: 1.** Refer to *"16-bit Timer/Event Counter T (TMT)" on page 1019* for the master timer setting. **2.** Refer to *"Timer Synchronization" on page 1017* for the settings and operations in the synchronous operation mode.
6	TSnEST	Sets to enable/disable software trigger control. 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 function operates as a software trigger by setting TSnEST to 1 during the timer Sn operation (when the value of TSnCE of TSnCTL0 register is 1). The function operates as a software trigger by setting TSnCE = 1 when the value of TSnEST is 1 while TSnCE = 0. **Note:** The TSnEST bit is always read as "0".
5	TSnEEE	Specifies count clock. 0: Use the clock selected with the TSnCKS2 to TSnCKS0 bits. 1: Use the external clock (TEVTSn pin input edge). The valid edge during TSnEEE = 1 (external clock: TEVTSn pin) is specified with the TSnEES1 and TSnEES0 bits of the TSnIOC2 register. The TSnEEE bit is ignored in the external event count mode.

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3-0</td><td>TSnMD3,
TSnMD2,
TSnMD1,
TSnMD0</td><td>Selects the timer mode.
<table>
<tr><th>Timer Mode</th><th>TSnMD3</th><th>TSnMD2</th><th>TSnMD1</th><th>TSnMD0</th></tr>
<tr><td>Interval timer mode</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>External event count mode</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>External trigger pulse output mode</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>One-shot pulse mode</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>PWM mode</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Free-running mode</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Triangular-wave PWM mode</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>High-accuracy T-PWM mode</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>PWM mode with dead time</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>120° excitation mode</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Special 120° excitation mode</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>Special pattern output mode</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Setting prohibited</td><td colspan="4">Other than above</td></tr>
</table></td></tr>
</table>

(3) Timer Sn I/O control register 0 (TSnIOC0)

The TSnIOC0 register is a 16-bit register that controls the timer output (TOSn0 to TOSn7 pins).
TSnIOC0H enables access to the higher 8 bits of the TSnIOC0 register and TSnIOC0L enables access to the lower 8 bits of the TSnIOC0 register.
Reset input sets this register to 0000H.

Caution Set the TSnIOC0 register when TSnCE is set to 0, in order to prevent the case that dead time cannot be secured, or short pulse is generated on output pins. Write access of the same value is possible for the TSnIOC0 register during the timer Sn operation (when TSnCE of the TSnCTL0 register is set to 1).

Address: TS0IOC0H: FFFFF583H, TS1IOC0H: FFFFF5C3H
TS0IOC0L: FFFFF582H, TS1IOC0L: FFFFF5C2H

	15	14	13	12	11	10	9	8	Initial value
TSnIOC0	TSnOL7	TSnOE7	TSnOL6	TSnOE6	TSnOL5	TSnOE5	TSnOL4	TSnOE4	0000H
	7	6	5	4	3	2	1	0	
	TSnOL3	TSnOE3	TSnOL2	TSnOE2	TSnOL1	TSnOE1	TSnOL0	TSnOE0	

	7	6	5	4	3	2	1	0	Initial value
TSnIOC0H	TSnOL7	TSnOE7	TSnOL6	TSnOE6	TSnOL5	TSnOE5	TSnOL4	TSnOE4	00H

	7	6	5	4	3	2	1	0	Initial value
TSnIOC0L	TSnOL3	TSnOE3	TSnOL2	TSnOE2	TSnOL1	TSnOE1	TSnOL0	TSnOE0	00H

Bit position	Bit name	Function
15	TSnOL7	Sets the TOSn7 pin output level. 0: Active level is high level. 1: Active level is low level.
14	TSnOE7	Enables/disables the TOSn7 pin output. 0: Timer output disabled (Output is inactive level of TSnOL7.) 1: Timer output enabled
13	TSnOL6	Sets the TOSn6 pin output level. 0: Active level is high level. 1: Active level is low level.
12	TSnOE6	Enables/disables the TOSn6 pin output. 0: Timer output disabled (Output is inactive level of TSnOL6 bit). 1: Timer output enabled
11	TSnOL5	Sets the TOSn5 pin output level. 0: Active level is high level. 1: Active level is low level.
10	TSnOE5	Enables/disables the TOSn5 pin output. 0: Timer output disabled (Output is inactive level of TSnOL5 bit). 1: Timer output enabled
9	TSnOL4	Sets the TOSn4 pin output level. 0: Active level is high level. 1: Active level is low level.

Bit position	Bit name	Function
8	TSnOE4	Enables/disables the TOSn4 pin output. 0: Timer output disabled (Output is inactive level of TSnOL4 bit). 1: Timer output enabled
7	TSnOL3	Sets the TOSn3 pin output level. 0: Active level is high level. 1: Active level is low level.
6	TSnOE3	Enables/disables the TOSn3 pin output. 0: Timer output disabled (Output is inactive level of TSnOL3 bit). 1: Timer output enabled
5	TSnOL2	Sets the TOSn2 pin output level. 0: Active level is high level. 1: Active level is low level.
4	TSnOE2	Enables/disables the TOSn2 pin output. 0: Timer output disabled (Output is inactive level of the TSnOL2 bit) 1: Timer output enabled
3	TSnOL1	Sets the TOSn1 output level. 0: Active level is high level. 1: Active level is low level.
2	TSnOE1	Enables/disables the TOSn1 pin output. 0: Timer output disabled (Output is inactive level of the TSnOL1 bit). 1: Timer output enabled
1	TSnOL0	Sets the TOSn0 output level. 0: Active level is high level. 1: Active level is low level.
0	TSnOE0	Enables/disables the TOSn0 pin output. 0: Timer output disabled (Output is inactive level of the TSnOL0 bit). 1: Timer output enabled

(4) Timer Sn I/O control register 2 (TSnIOC2)

The TSnIOC2 register is an 8-bit register that controls the valid edge for the external event count input (TEVTSn pin) and external trigger input (TTRGSn pin).
This register can be read/written with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 00H.

Caution Set the TSnIOC2 register while operation of timer Sn is stopped (when the value of TSnCE of the TSnCTL0 register is 0). Write access of the same value is possible for the TSnIOC2 register during the timer Sn operation (TSnCE of the TSnCTL0 register is set to 1).

Address: TS0IOC2: FFFFF584H, TS1IOC2: FFFFF5C4H

	7	6	5	4	3	2	1	0	Initial value
TSnIOC2	0	0	0	0	TSnEES1	TSnEES0	TSnETS1	TSnETS0	00H

Bit position	Bit name	Function
3, 2	TSnEES1, TSnEES0	Sets the valid edge of external event count input (TEVTSn pin). TSnEES1 / TSnEES0 / Valid edge 0 / 0 / No edge detection (Trigger is masked.) 0 / 1 / Detects rising edge. 1 / 0 / Detects falling edge. 1 / 1 / Detects both edges. **Note:** The TSnEES1 and TSnEES0 bits are valid only when TSnEEE of the TSnCTL1 register is set to 1, or when the external event count mode (TSnMD3 to TSnMD0 of the TSnCTL1 register is set to [0, 0, 0, 1]).
1, 0	TSnETS1, TSnETS0	Sets the valid edge of the external trigger input (TTRGSn pin). TSnETS1 / TSnETS0 / Valid edge 0 / 0 / No edge detection (Trigger is masked.) 0 / 1 / Detects rising edge. 1 / 0 / Detects falling edge. 1 / 1 / Detects both edges. **Note:** The TSnETS1 and TSnETS0 bits are valid only when the external trigger pulse output mode or one-shot pulse mode (TSnMD3 to TSnMD0 of the TSnCTL1 register is set to [0, 0, 1, 0] or [0, 0, 1, 1]).

(5) Timer Sn I/O control register 4 (TSnIOC4)

The TSnIOC4 register is an 8-bit register that controls timer output (TOSn1 to TOSn6 pins).
This register can be read/written with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 00H.

Caution Set the TSnIOC4 register while operation of timer Sn is stopped (TSnCE of the TSnCTL0 register is set to 0).

Address: TS0IOC4: FFFFF585H, TS1IOC4: FFFFF5C5H

	7	6	5	4	3	2	1	0	Initial value
TSnIOC4	0	TSnTBA2	TSnTBA1	TSnTBA0	0	TSnWOC	0	TSnEOC	00H

Bit position	Bit name	Function
6	TSnTBA2	Selects whether to detect the simultaneous active states of the TOSn5 and TOSn6 pins. 0: Don't detect the simultaneous active states of the TOSn5 and TOSn6 pins. 1: Detect the simultaneous active states of the TOSn5 and TOSn6 pins. **Note:** If simultaneous active states are detected while TSnEOC = 1 or TSnTBA2 = 1, the positive/negative phase simultaneous active state detection flag (TSnTBF) is set to 1 and an error interrupt (INTTSnER) occurs.
5	TSnTBA1	Selects whether to detect the simultaneous active states of the TOSn3 and TOSn4 pins. 0: Don't detect the simultaneous active states of the TOSn3 and TOSn4 pins. 1: Detect the simultaneous active states of the TOSn3 and TOSn4 pins. **Note:** If simultaneous active states are detected while TSnEOC = 1 or TSnTBA1 = 1, the positive/negative phase simultaneous active detection flag (TSnTBF) is set to 1 and an error interrupt (INTTSnER) occurs.

Bit position	Bit name	Function
4	TSnTBA0	Selects whether to detect the simultaneous active states of the TOSn1 and TOSn2 pins. 0: Don't detect the simultaneous active states of the TOSn1 and TOSn2 pins. 1: Detect the simultaneous active states of the TOSn1 and TOSn2 pins. **Note:** If simultaneous active states are detected while TSnEOC = 1 or TSnTBA0 = 1, the positive/negative phase simultaneous active detection flag (TSnTBF) is set to 1 and an error interrupt (INTTSnER) occurs.
2	TSnWOC	Selects the warning detection during motor control. 0: Disable the warning interrupt (INTTSnWN) 1: Enable the warning interrupt (INTTSnWN) **Note:** Refer to *"Warning interrupt function " on page 826* for details on warning interrupt control.
0	TSnEOC	Selects the error status detection during motor control. 0: Disable the error interrupt (INTTSnER). 1: Enable the error interrupt (INTTSnER). **Note:** Refer to *"Error interrupt function" on page 823* for details on error interrupt control.

(6) Timer Sn option register 0 (TSnOPT0)

The TSnOPT0 register is an 8-bit register that detects overflow.
This register can be read/written with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 00H.

Caution Do not rewrite the TSnDSE bit of the TSnOPT0 register during the timer Sn operation (when TSnCE of the TSnCTL0 register is set to 1). However, write access of the same value is possible.

Address: TS0OPT0: FFFFF586H, TS1OPT0: FFFFF5C6H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT0	0	0	0	0	TSnDSE	TSnCMS	TSnCUF	TSnOVF	00H

Bit position	Bit name	Function
3	TSnDSE	Enables/disables dead time control in the PWM mode and in the special pattern output mode. 0: Disables dead time control 1: Enables dead time control **Note:** The TSnDSE bit is valid only in PWM mode and in special pattern output mode, but is invalid in the other modes.
2	TSnCMS	Specifies the transfer timing of the compare register in the high-accuracy T-PWM mode. 0: Reload mode (batch rewrite function) When writing to the TSnCCR1 register, reload is enabled and then the value is updated (reloaded) simultaneously at the next reload timing. Reload is not enabled even if written to a register other than TSnCCR1. 1: Anytime rewrite mode Writing to each compare register under update operation by each independent compare register enables write access value updating anytime. 4 clocks are needed from register write until the value is transferred. In the case of the TSnCCR1 register, 5 clocks are needed. **Caution:** Set TSnCMS to 0 in any mode other than the high-accuracy T-PWM mode because the transfer timing is fixed.

Bit position	Bit name	Function
1	TSnCUF	This is a flag indicating the 16-bit counter count status. 0: 16-bit counter is in count up status. 1: 16-bit counter is in count down status. **Note:** The TSnCUF bit is valid only in the high-accuracy T-PWM mode and in the triangular-wave PWM mode, but is invalid (TSnCUF = 0) in any other mode.
0	TSnOVF	This is a flag indicating the timer Sn overflow. It is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H. This flag is cleared by writing 0 or setting TSnCE to 0. An overflow interrupt (INTTSnOV) occurs at the same time that the TSnOVF bit is set to 1. **Caution:** 1. Overflow occurs only in the free-running mode or high-accuracy T-PWM mode. In the high-accuracy T-PWM mode, the TSnOVF bit may be set to 1 if the setting value condition of the TSnDTC0 and TSnDTC1 registers is incorrect. 2. **The TSnOVF bit is not cleared even if the TSnOVF bit and the TSnOPT0 register are read while TSnOVF = 1.** 3. **Reading or writing is possible for the TSnOVF bit, but writing 1 to the TSnOVF bit is ignored.**

(7) Timer Sn option registers 1, 1H, 1L (TSnOPT1, TSnOPT1H, TSnOPT1L) Timer Sn option registers 1A, 1AH, 1AL (TSnOPT1A, TSnOPT1AH, TSnOPT1AL)

The TSnOPT1 and TSnOPT1A registers are 16-bit registers which enable/disable peak/trough interrupt and set the interrupt thinning out ratio. The TSnOPT1A register is a mirror of the TSnOPT1 register for optimized DMA support.
These registers can be read/written with a 16-bit, 8-bit, or 1-bit manipulation instruction.
TSnOPT1H, and TSnOPT1AH respectively, is an 8-bit register, which can access the higher 8 bits of the TSnOPT1 register, and TSnOPT1A register respectively. TSnOPT1L, and TSnOPT1AL respectively, is an 8-bit register, which can access the lower 8 bits of the TSnOPT1 register, and TSnOPT1AL respectively.
Reset input sets these registers to 0000H, and 00H respectively.

Caution The register rewriting methods for the TSnOPT1, TSnOPT1H, TSnOPT1L, TSnOPT1A, TSnOPT1AH, and TSnOPT1AL registers during count operation in the high-accuracy T-PWM mode are as follows based on the setting of the TSnCMS bit of the TSnOPT0 register.

- TSnCMS = 0: Reload rewriting
- TSnCMS = 1: Anytime rewriting

The reload rewriting method is used in the external trigger pulse output mode, PWM mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.

Address: TS0OPT1: FFFFF58AH, TS1OPT1: FFFFF5CAH
TS0OPT1H: FFFFF58BH, TS1OPT1H: FFFFF5CBH
TS0OPT1L: FFFFF58AH, TS1OPT1L: FFFFF5CAH

	15	14	13	12	11	10	9	8	Initial value
TSnOPT1	0	0	0	0	0	0	TSnRTE	TSnRBE	0000H
	7	6	5	4	3	2	1	0	
	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0	

	7	6	5	4	3	2	1	0	Initial value
TSnOPT1H	0	0	0	0	0	0	TSnRTE	TSnRBE	00H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT1L	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0	00H

Address: TS0OPT1A: FFFFFB40H, TS1OPT1A: FFFFFBC0H
TS0OPT1AH: FFFFFB41H, TS1OPT1AH: FFFFFBC1H
TS0OPT1AL: FFFFFB40H, TS1OPT1AL: FFFFFBC0H

	15	14	13	12	11	10	9	8	Initial value
TSnOPT1A	0	0	0	0	0	0	TSnRTE	TSnRBE	0000H
	7	6	5	4	3	2	1	0	
	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0	

	7	6	5	4	3	2	1	0	Initial value
TSnOPT1AH	0	0	0	0	0	0	TSnRTE	TSnRBE	00H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT1AL	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0	00H

Bit position	Bit name	Function
9	TSnRTE	Enables/disables reload at the peak reload timing. 0: Disables the reload operation at the peak timing of 16-bit counter. 1: Enables the reload operation at the peak timing of 16-bit counter. **Note:** The TSnRTE bit is valid only in the external trigger pulse output mode, PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.
8	TSnRBE	Enables/disables reload at the trough reload timing. 0: Disables reload operation at the trough timing for 16-bit counter. 1: Enables reload operation at the trough timing for 16-bit counter. **Note:** The TSnRBE bit is valid only in the high-accuracy T-PWM mode and triangular-wave PWM mode.
7	TSnICE	Enables/disables occurrence of the peak interrupt (INTTSnCD0). 0: Disables the peak interrupt (INTTSnCD0) at the peak timing for 16-bit counter. Not subject to interrupt thinning out. 1: Enables the peak interrupt (INTTSnCD0) at the peak timing for 16-bit counter. Subject to interrupt thinning out. **Note:** The TSnICE bit is valid only in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.
6	TSnIOE	Enables/disables occurrence of the trough interrupt (INTTSnOD). 0: Disables the trough interrupt. Not subject to interrupt thinning out. 1: Enables the trough interrupt. Subject to interrupt thinning out. **Note:** The TSnIOE bit is valid only in high-accuracy T-PWM mode and triangular-wave PWM output mode.

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>TSnRDE</td><td>Enables/disables thinning out at the reload timing.
0: No reload thinning out
Reload timing is generated based on the setting of the TSnRTE and TSnRBE bits.
1: Reload thinning out provided
Reload timing is generated in the same interval as interrupt thinning out.

Note: The TSnRDE bit is valid only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.
The TSnRDE bit must be set to 1 in any mode other than the above.</td></tr>
<tr><td>4 to 0</td><td>TSnID4,
TSnID3,
TSnID2,
TSnID1,
TSnID0</td><td>Specifies the thinning out ratio of interrupts (INTTSnCD0, INTTSnOD).
<table>
<tr><th>TSnID4</th><th>TSnID3</th><th>TSnID2</th><th>TSnID1</th><th>TSnID0</th><th>Thinning out Ratio</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>No thinning out</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1/3</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1/4</td></tr>
<tr><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td></tr>
<tr><td>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: During the timer Sn operation (when TSnCE of the TSnCTL0 register is set to 1), the interrupt thinning out counter is cleared if write access is executed to the TSnOPT1 and TSnOPT1L registers, TSnOPT1A and TSnOPT1AL registers respectively (including when the same value is written to the TSnID4 to TSnID0 bits).
The interrupt thinning out counter is not cleared by write access of the TSnOPT1H register, and TSnOPT1AH register respectively.

Note: The TSnID4 to TSnID0 bits are valid only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.</td></tr>
</table>

Table 17-3 Reload, peak interrupt and trough interrupt

Mode	TSnCMS	TSnRDE	TSnICE	TSnRTE	TSnIOE	TSnRBE	Reload	INTTSnCD0	INTTSnOD
PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	1	x	x	x	x	x	Setting prohibited		
	0	0	x	0	x	x	No		
	0	0	x	1	x	x	Peak	Occurrence **Note 1**	
	0	1	0	0	x	x	No		
	0	1	0	1	x	x	No		
	0	1	1	0	x	x	No		
	0	1	1	1	x	x	Peak**Note 3**	Occurrence	-
Triangular-wave PWM mode	1	x	x	x	x	x	Setting prohibited		
	0	0	x	x	x	0	No		
	0	0	x	x	x	1	Trough	-	Occurrence **Note 2**
	0	1	x	x	0	0	No		
	0	1	x	x	0	1	No		
	0	1	x	x	1	0	No		
	0	1	x	x	1	1	Trough**Note 3**	-	Occurrence
High-accuracy T-PWM mode	1	x	x	x	x	x	Anytime	Occurrence **Note 1**	Occurrence **Note 2**
	0	0	x	0	x	0	No		
	0	0	x	0	x	1	Trough	Occurrence **Note 1**	Occurrence **Note 2**
	0	0	x	1	x	0	Peak	Occurrence **Note 1**	Occurrence **Note 2**
	0	0	x	1	x	1	Peak/trough	Occurrence **Note 1**	Occurrence **Note 2**
	0	1	0	0	0	0	No		
	0	1	0	0	0	1	No		
	0	1	0	0	1	0	No		
	0	1	0	0	1	1	Trough**Note 3**	-	Occurrence
	0	1	0	1	0	0	No		
	0	1	0	1	0	1	No		
	0	1	0	1	1	0	No		
	0	1	0	1	1	1	Trough**Note 3**	-	Occurrence
	0	1	1	0	0	0	No		
	0	1	1	0	0	1	No		
	0	1	1	0	1	0	No		
	0	1	1	0	1	1	Trough**Note 3**	Occurrence	Occurrence
	0	1	1	1	0	0	Peak**Note 3**	Occurrence	-
	0	1	1	1	0	1	Peak**Note 3**	Occurrence	-
	0	1	1	1	1	0	Peak**Note 3**	Occurrence	Occurrence
	0	1	1	1	1	1	Peak**Note 3**/Trough**Note 3**	Occurrence	Occurrence

Mode	TSnCMS	TSnRDE	TSnICE	TSnRTE	TSnIOE	TSnRBE	Reload	INTTSnCD0	INTTSnOD
External trigger pulse output mode	1	x	x	x	x	x	Setting prohibited		
	0	0	x	0	x	x	No		
	0	0	x	1	x	x	Peak	-	-
	0	1	0	0	x	x	Setting prohibited		
	0	1	0	1	x	x	Setting prohibited		
	0	1	1	0	x	x	Setting prohibited		
	0	1	1	1	x	x	Setting prohibited		

Note
1. Only when TSnICE = 1
2. Only when TSnIOE = 1
3. At the timing at which thinning out of the peak interrupt or the trough interrupt does not occur.
4. X: 0 or 1
5. Throughout this manual the register name TSnOPT1(H/L) means also the register TSnOPT1A(H/L).

(8) Timer Sn option registers 2, 2H, 2L (TSnOPT2, TSnOPT2H, TSnOPT2L)

These TSnOPT2 registers are 16-bit registers that control A/D conversion trigger output (TSnADTRG0 signal).
These registers can be read/written with a 16-bit, 8-bit, or 1-bit manipulation instruction.
TSnOPT2H can access the higher 8 bits of the TSnOPT2 register. TSnOPT2L can access the lower 8 bits of the TSnOPT2 register.
Reset input sets these registers to 0000H.

Caution The write timing is delayed due to wait insertion during register access under the following conditions if serial write to the TSnOPT2 register is executed during the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1)

Write address	Next write address	Write delay
TSnOPT2L	TSnOPT2, TSnOPT2L	Delay
	TSnOPT2H	No delay
TSnOPT2H	TSnOPT2, TSnOPT2H	Delay
	TSnOPT2L	No delay
TSnOPT2	TSnOPT2, TSnOPT2H, TSnOPT2L	Delay

Address: TS0OPT2H: FFFFF5A1H, TS1OPT2H: FFFFF5E1H
TS0OPT2L: FFFFF5A0H, TS1OPT2L: FFFFF5E0H

	15	14	13	12	11	10	9	8	Initial value
TSnOPT2	0	0	0	0	0	0	TSnACC1	TSnACC0	0000H
	7	6	5	4	3	2	1	0	
	TSnAT07	TSnAT06	TSnAT05	TSnAT04	TSnAT03	TSnAT02	TSnAT01	TSnAT00	

	7	6	5	4	3	2	1	0	Initial value
TSnOPT2H	0	0	0	0	0	0	TSnACC1	TSnACC0	00H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT2L	TSnAT07	TSnAT06	TSnAT05	TSnAT04	TSnAT03	TSnAT02	TSnAT01	TSnAT00	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>9, 8</td><td>TSnACC1,
TSnACC0</td><td>Specifies the thinning out ratio of the A/D conversion trigger (TSnADTRG0 signal).
<table>
<tr><th>TSnACC1</th><th>TSnACC0</th><th>Thinning out Ratio</th></tr>
<tr><td>0</td><td>0</td><td>No thinning out</td></tr>
<tr><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>1</td><td>0</td><td>1/4</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td></tr>
</table>
Caution: During the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1), the TSnADTRG0 signal thinning out counter is cleared if write access to the TSnOPT2 and TSnOPT2H registers is executed (including when the same value is written to the TSnACC1 and TSnACC0 bits).

Note: During the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1), the TSnADTRG0 signal thinning out counter is not cleared by write access to the TSnOPT2L register.</td></tr>
<tr><td>7</td><td>TSnAT07</td><td>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) at the (peak) timing when the 16-bit sub-counter switches from count-up operation to count-down operation.
0: Don’t use 16-bit sub-counter peak timing as A/D conversion trigger.
1: Use 16-bit sub-counter peak timing as A/D conversion trigger.

Caution: 1. The TSnAT07 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT07 bit to 0 in any other mode.
2. Do not set the TSnAT07 bit to 1 when TSnDTC0 ≠ 0000H, TSnDTC1 = 0000H. The A/D conversion trigger is not generated at the peak timing of the 16-bit sub-counter.</td></tr>
<tr><td>6</td><td>TSnAT06</td><td>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) at the (trough) timing when the 16-bit sub-counter switches from count-down operation to count-up operation.
0: Don’t use 16-bit sub-counter trough timing as A/D conversion trigger.
1: Use 16-bit sub-counter trough timing as A/D conversion trigger.

Caution: 1. The TSnAT06 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT06 bit to 0 in any other mode.
2. Do not set the TSnAT06 bit to 1 when TSnDTC0 = 0000H, TSnDTC1 ≠ 0000H. The A/D conversion trigger is not generated at the trough timing of the 16-bit sub-counter.</td></tr>
</table>

Bit position	Bit name	Function
5	TSnAT05	Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC5 interrupt occurs during count-down operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. **Caution:** The TSnAT05 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT05 to 0 in any other mode.
4	TSnAT04	Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC5 interrupt occurs during count-up operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. **Caution:** The TSnAT04 bit can be set to 1 only in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT04 to 0 in any other mode.
3	TSnAT03	Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC4 interrupt occurs during count-down operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. **Caution:** The TSnAT03 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT03 to 0 in any other mode.

Bit position	Bit name	Function
2	TSnAT02	Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC4 interrupt occurs during count-up operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. **Caution:** The TSnAT02 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead-time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT02 to 0 in any other mode.
1	TSnAT01	Enables the A/D conversion trigger (TSnADTRG0 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (peak interrupt). 0: Don't use the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger. 1: Use the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger. **Caution:** The TSnAT01 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT01 to 0 in any other mode.
0	TSnAT00	Enables the A/D conversion trigger (TSnADTRG0 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (trough interrupt). 0: Don't use the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger. 1: Use the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger. **Caution:** The TSnAT00 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT00 to 0 in any other mode.

(9) Timer Sn option registers 3, 3H, 3L (TSnOPT3, TSnOPT3H, TSnOPT3L)

The TSnOPT3 registers are 16-bit registers that control A/D conversion trigger output (TSnADTRG1 signal).
These registers can be read or written with a 16-bit, 8-bit, or 1-bit manipulation instruction.
TSnOPT3H enables access to the higher 8 bits of the TSnOPT3 register and TSnOPT3L enables access to the lower 8 bits of the TSnOPT3 register.
Reset input sets these registers to 0000H.

Caution The write timing is delayed due to wait insertion during register access under the following conditions if serial write to the TSnOPT3 register is executed during the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1).

Write address	Next write address	Write delay
TSnOPT3L	TSnOPT3, TSnOPT3L	Delay
	TSnOPT3H	No delay
TSnOPT3H	TSnOPT3, TSnOPT3H	Delay
	TSnOPT3L	No delay
TSnOPT3	TSnOPT3, TSnOPT3H, TSnOPT3L	Delay

Address: TS0OPT3H: FFFFF5A3H, TS1OPT3H: FFFFF5E3H
TS0OPT3L: FFFFF5A2H, TS1OPT3L: FFFFF5E2H

	15	14	13	12	11	10	9	8	Initial value
TSnOPT3	0	0	0	0	0	0	TSnACC3	TSnACC3	0000H
	7	6	5	4	3	2	1	0	
	TSnAT17	TSnAT16	TSnAT15	TSnAT14	TSnAT13	TSnAT12	TSnAT11	TSnAT10	

	7	6	5	4	3	2	1	0	Initial value
TSnOPT3H	0	0	0	0	0	0	TSnACC3	TSnACC3	00H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT3L	TSnAT17	TSnAT16	TSnAT15	TSnAT14	TSnAT13	TSnAT12	TSnAT11	TSnAT10	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>9, 8</td><td>TSnACC3,
TSnACC2</td><td>Specifies the thinning out ratio of the A/D conversion trigger (TSnADTRG1 signal).
<table>
<tr><th>TSnACC3</th><th>TSnACC3</th><th>Thinning out Ratio</th></tr>
<tr><td>0</td><td>0</td><td>No thinning out</td></tr>
<tr><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>1</td><td>0</td><td>1/4</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td></tr>
</table>
Caution: During the timer Sn operation (when TSnCE of the TSnCTL0 register is set to 1), the TSnADTRG1 signal thinning out counter is cleared if write access to the TSnOPT3 and TSnOPT3H registers is executed (including when the same value is written to for the TSnACC3 and TSnACC2 bits).

Note: During the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1), the TSnADTRG1 signal thinning out counter is not cleared by write access to the TSnOPT3L register.</td></tr>
<tr><td>7</td><td>TSnAT17</td><td>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) at the (peak) timing when the 16-bit sub-counter switches from count-up operation to count-down operation.
0: Don’t use 16-bit sub-counter peak timing as A/D conversion trigger.
1: Use 16-bit sub-counter peak timing as A/D conversion trigger.

Caution: 1. The TSnAT17 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT17 bit to 0 in any other mode.
2. Do not set the TSnAT17 bit to 1 when TSnDTC0 ≠ 0000H, TSnDTC1 = 0000H. The A/D conversion trigger is not generated at the peak timing of the 16-bit sub-counter.</td></tr>
<tr><td>6</td><td>TSnAT16</td><td>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) at the (trough) timing when the 16-bit sub-counter switches from count-down operation to count-up operation.
0: Don’t use 16-bit sub-counter trough timing as A/D conversion trigger.
1: Use 16-bit sub-counter trough timing as A/D conversion trigger.

Caution: 1. The TSnAT16 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT16 bit to 0 in any other mode.
2. Do not set the TSnAT16 bit to 1 when TSnDTC0 = 0000H, TSnDTC1 ≠ 0000H. The A/D conversion trigger is not generated at the trough timing of the 16-bit sub-counter.</td></tr>
</table>

Bit position	Bit name	Function
5	TSnAT15	Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC5 interrupt occurs during count-down operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. **Caution:** The TSnAT15 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT15 to 0 in any other mode.
4	TSnAT14	Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC5 interrupt occurred during count-up operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. **Caution:** The TSnAT14 bit can be set to 1 only in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT14 to 0 in any other mode.
3	TSnAT13	Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC4 interrupt occurs during count-down operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger. **Caution:** The TSnAT13 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT13 to 0 in any other mode.

Bit position	Bit name	Function
2	TSnAT12	Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC4 interrupt occurs during count-up operation of the 16-bit counter. 0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. 1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger. **Caution:** The TSnAT12 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead-time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT12 to 0 in any other mode.
1	TSnAT11	Enables the A/D conversion trigger (TSnADTRG1 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (peak interrupt). 0: Don't use occurrence of the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger. 1: Use occurrence of the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger. **Caution:** The TSnAT11 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT11 to 0 in any other mode.
0	TSnAT10	Enables the A/D conversion trigger (TSnADTRG1 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (trough interrupt). 0: Don't use occurrence of the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger. 1: Use occurrence of the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger. **Caution:** The TSnAT10 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT10 to 0 in any other mode.

(10) Timer Sn option register 4 (TSnOPT4)

The TSnOPT4 register is an 8-bit register that controls timer output (TOSn1 to TOSn6) by software and pattern output order.
This register can be read/written with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 00H.

Address: TS0OPT4: FFFFF588H, TS1OPT4: FFFFF5C8H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT4	TSnSOC	0	0	0	TSnPSC	TSnIPC2	TSnIPC1	TSnIPC0	00H

Bit position	Bit name	Function
7	TSnSOC	Selects the timer output (TOSn1 to TOSn6) control by software 0: Software control is disabled. 1: Software control is enabled. When the TSnSOC bit is set to 1, timer output is switched to the output pattern of the software control function set by the TSnIPC2 to TSnIPC0 bits. Dead time is ensured by using the dead time 16-bit counter. Refer to *"Timer output initial pattern setting in each mode and function" on page 731* for details. **Caution:** The setting of TSnSOC = 1 and TSnADC of the TSnOPT5 register = 1 is prohibited. **Note:** The setting of the TSnSOC bit is valid in the high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, and special excitation mode.
3	TSnPSC	Selects the pattern output order during semi-automatic driving. 0: Switch the timer output (TOSn1 to TOSn6) into the normal rotation order. 1: Switch the timer output (TOSn1 to TOSn6) into the reverse rotation order. The TSnSPC bit specifies the timer output pattern order with output pattern set with the TSnIPC2 to TSnIPC0 bits as the initial pattern in the 120° excitation mode, special 120° excitation mode, and 180° excitation control function. The TSnPSC bit is valid when TSnSOC of the TSnOPT5 register is set to 1, or when TSnSOC is set to 0, TSnPOT of the TSnOPT5 register is set to 1, and TSnPSS is set to 1. **Caution:** 1. The TSnPSC bit must be changed when TSnCE is set to 0 or TSnPOT is set to 0. The timer output pattern may differ from the expected one if rewritten when TSnPOT is set to 1. 2. **TSnCE = 1 setting must be executed after matching the PSC bit order with the input signal variation logic when the signal input to the TAPTSn0 to TAPTSn2 pins varies while the timer Sn operation is stopped (when the TSnCE of the TSnCTL0 register is set to 0).** **Note:** Refer to *"Overview of 120° excitation mode" on page 928* for the normal/reverse rotation output order.

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>TSnIPC2,
TSnIPC1,
TSnIPC0</td><td>Set the initial pattern of the timer output in the 120° excitation mode, special 120° excitation mode and 180° excitation control function.
Refer to “Timer output initial pattern setting in each mode and function” on page 731 for details.
<table>
<tr><th>TSnIPC2</th><th>TSnIPC1</th><th>TSnIPC0</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Inactive output if there is no change from the initial value (000B).
Output level remains the level before the change if there is a change from any value to the initial value (000B).</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>The output level remains the level before the change.</td></tr>
<tr><td colspan="3">Other than above</td><td>Timer output based on the pattern.</td></tr>
</table></td></tr>
</table>

(a) Function of the TSnIPC2 to TSnIPC0 bits and output of TOSn1 to TOSn6 pins by setting the TSnSOC and TSnADC bits

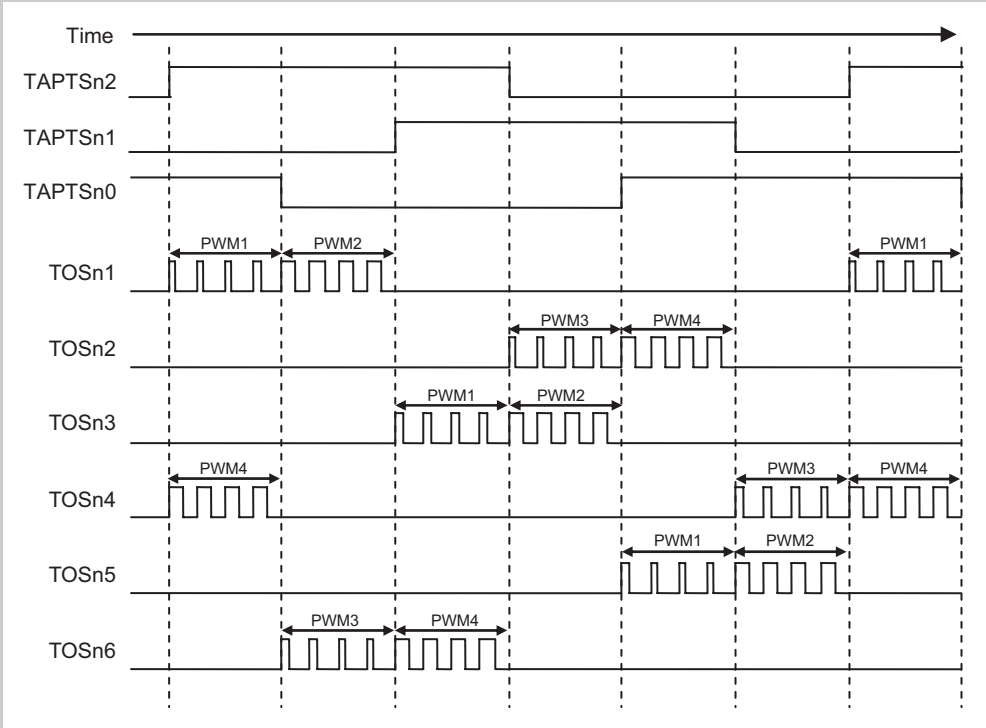
Operation mode	TSnSOC bit of TSnOPT4 register	TSnADC[Note] bit of TSnOPT5 register	Function of TSnIPC2 to TSnIPC0 bit of TSnOPT4 register	Output of TOSn1 to TOSn6 pins
High-accuracy T-PWM mode	1	0	Pattern setting	Software 180° excitation
	0	1	Initial pattern setting	Semi-automatic driving 180° excitation
	0	0	Not used	High-accuracy T-PWM
	1	1	Setting prohibited	
PWM mode with dead time	1	0	Pattern setting	Software 180° excitation
	0	1	Setting prohibited	
	0	0	Not used	PWM with dead time
	1	1	Setting prohibited	
120° excitation mode	1	0	Pattern setting	Software 180° excitation
	0	1	Initial pattern setting	Semi-automatic driving 180° excitation
	0	0	Pattern setting	120° excitation
	1	1	Setting prohibited	
Special 120° excitation mode	1	0	Pattern setting for software output controlling	Software 180° excitation
	0	1	Initial pattern setting	Semi-automatic driving 180° excitation
	0	0	Initial pattern setting	Special 120° excitation mode
	1	1	Setting prohibited	

Caution TSnPOT of the TSnOPT5 register must be set to 1 if setting the TSnIPC2 to TSnIPC0 bits when using the 180° excitation control function.

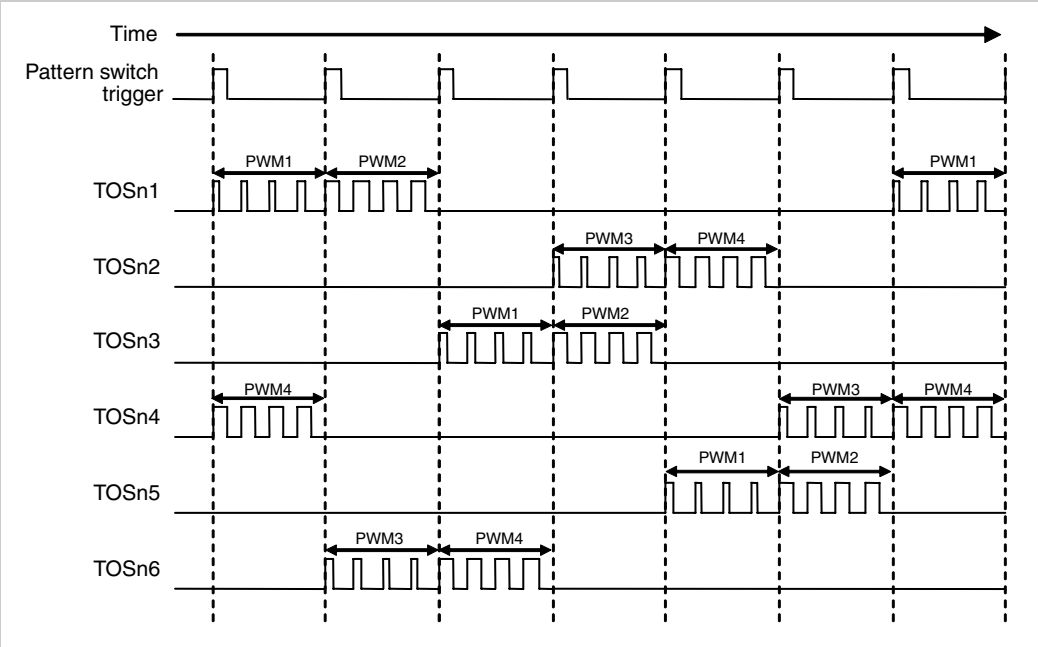
(b) Timer output order in each mode and function

- In 120° excitation mode
  The TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register are set when TSnPOT of the TSnOPT5 register is set to 1 and while the timer Sn operation is stopped (when TSnCE of the TSnCTL0 register is set to 0), and then timer output starts with the set pattern when the TSnCE bit of the TSnCTL0 register is set to 1.

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

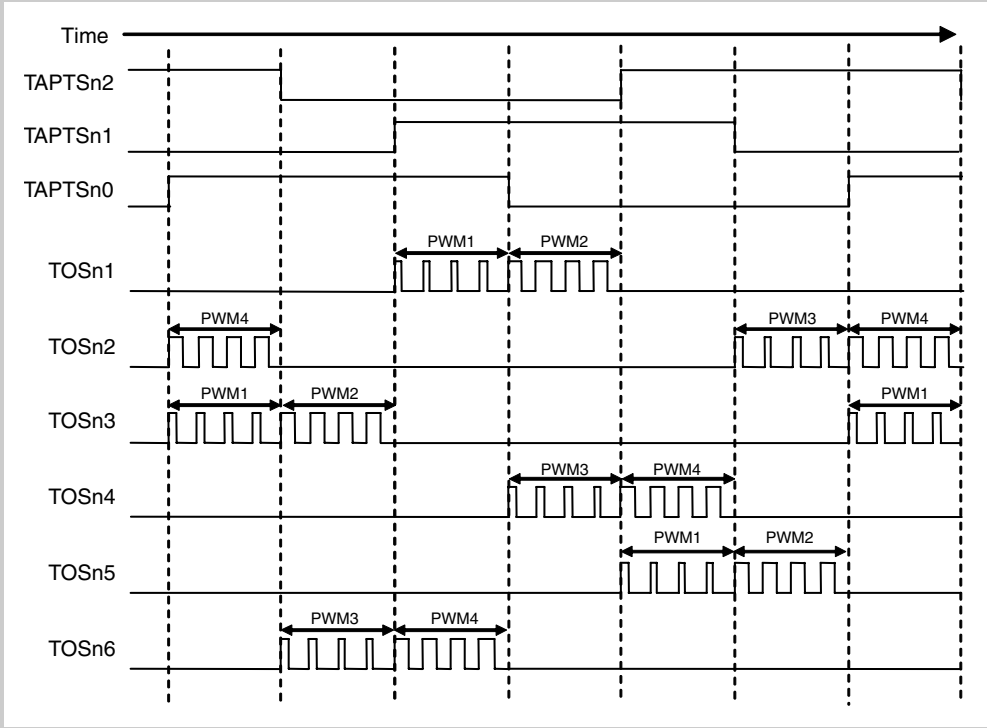


Normal rotation:TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

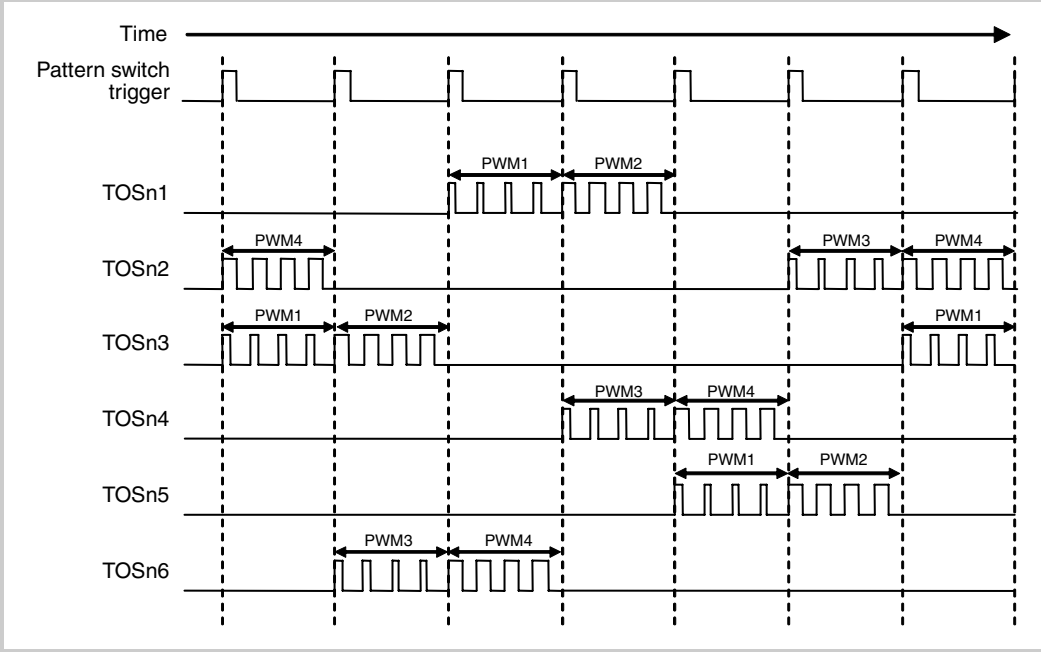


Note PWM1 to PWM4 indicate PWM output using the 16-bit counter and the TSnCCR1 to TSnCCR4 registers.

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1



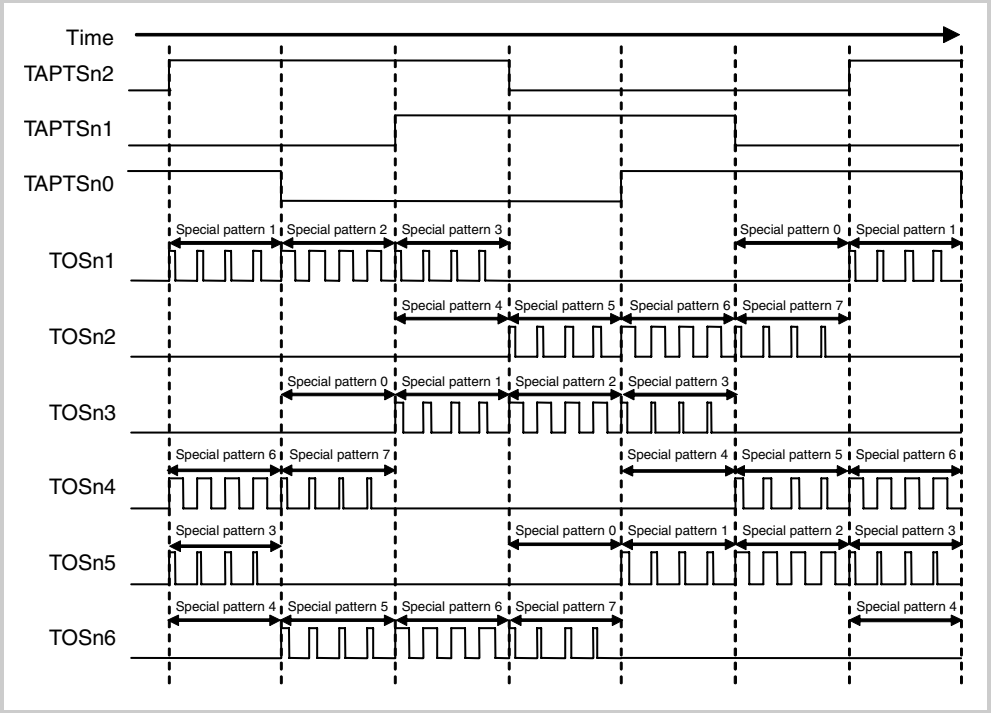
Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1



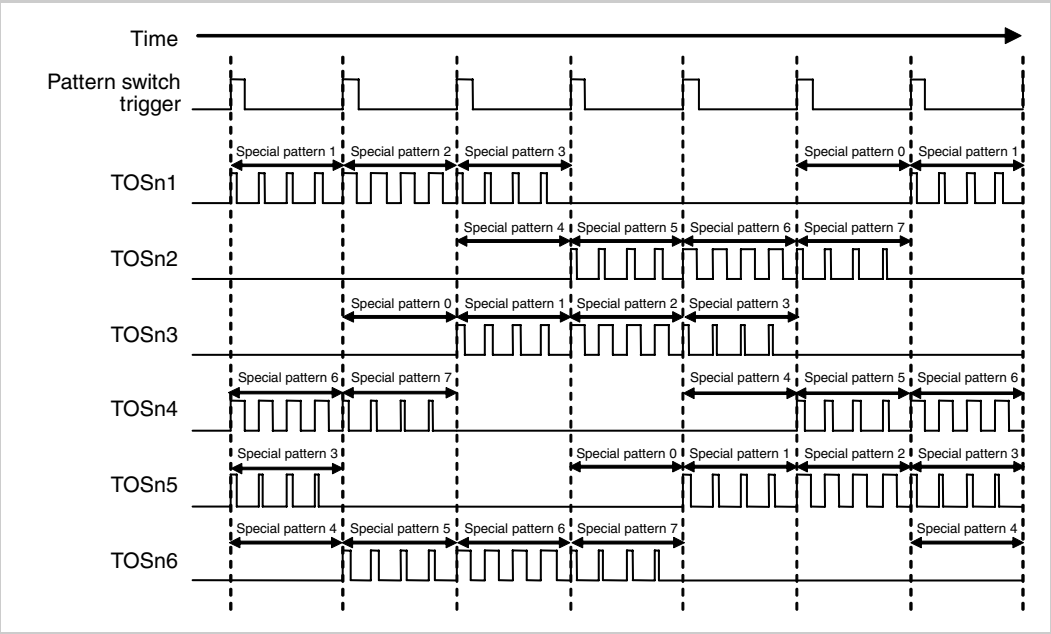
Note PWM1 to PWM4 indicate PWM output using the 16-bit counter and the TSnCCR1 to TSnCCR4 registers.

- In special 120° excitation mode

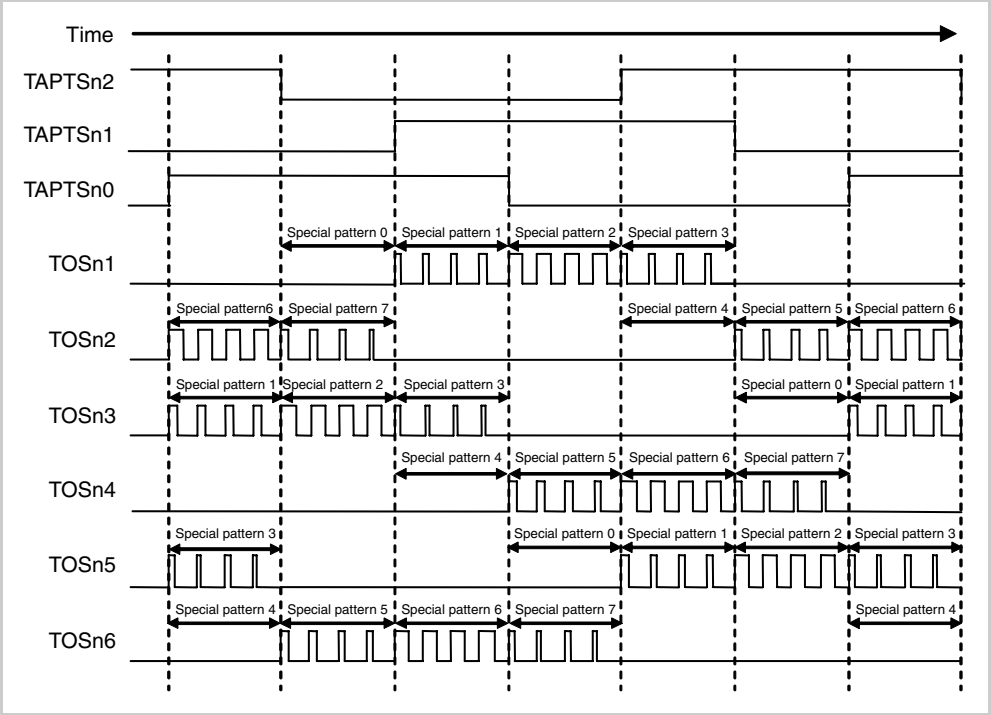
Normal rotation:TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0



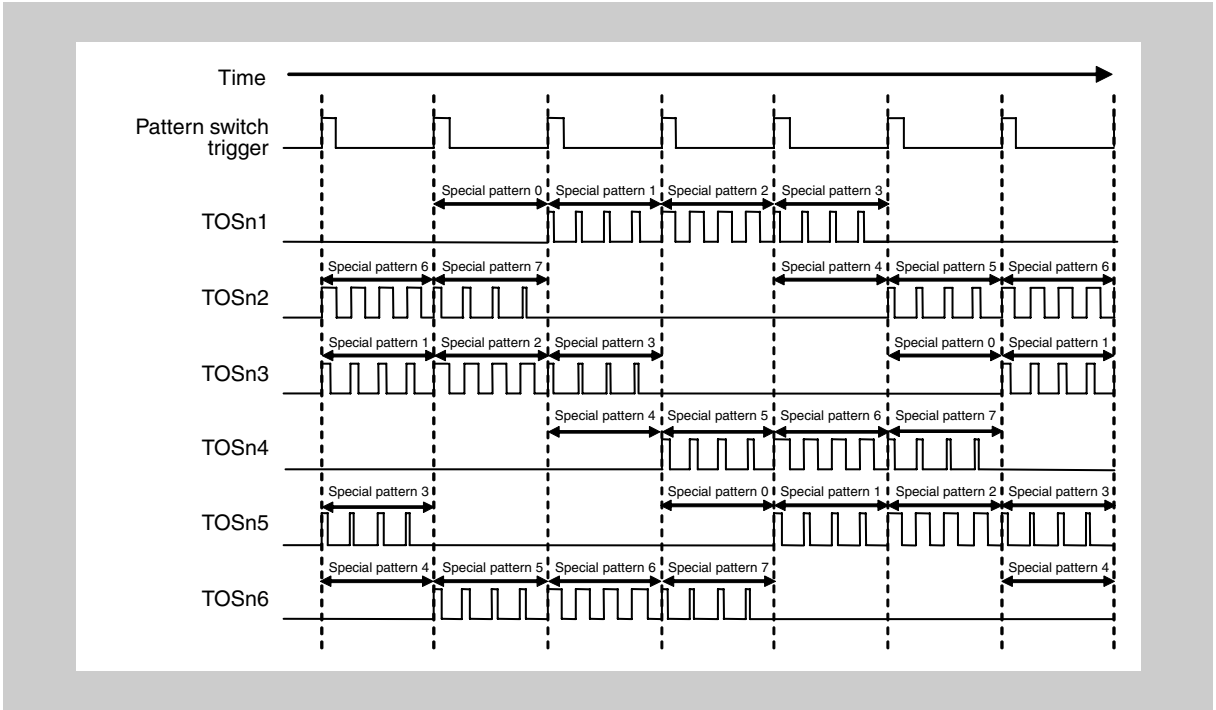
Normal rotation:TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0



Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1



Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1



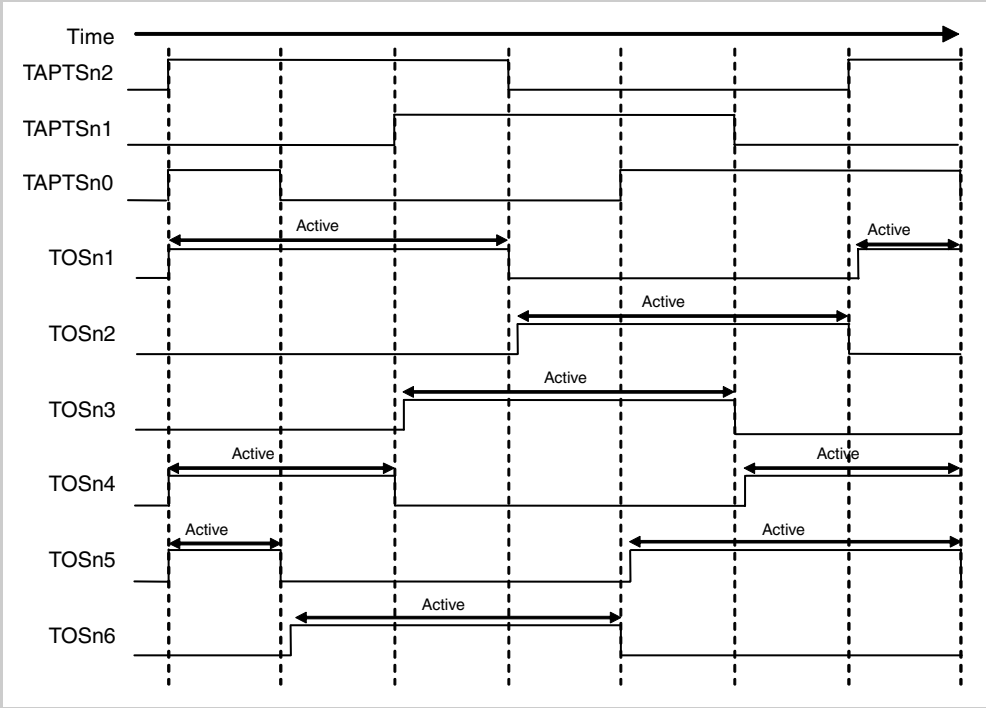
Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

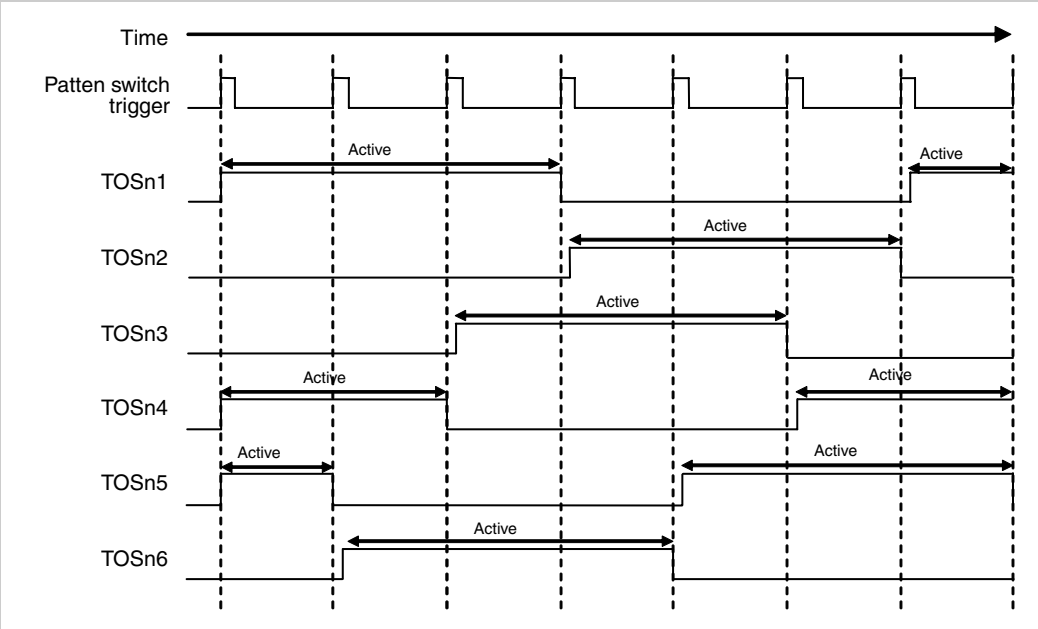
Note Special patterns 0 to 3 are output using the TSnPAT0 register.
Special pattern 0: Bits 3 to 0
Special pattern 1: Bits 7 to 4
Special pattern 2: Bits 11 to 8
Special pattern 3: Bits 15 to 12
Special patterns 4 to 7 are output using the TSnPAT1 register.
Special pattern 4: Bits 3 to 0
Special pattern 5: Bits 7 to 4
Special pattern 6: Bits 11 to 8
Special pattern 7: Bits 15 to 12

- When using 180° excitation control function
  The TSnIPC2 to TSnIPC0 bits start timer output with the setting pattern when TSnADC is set to 1.

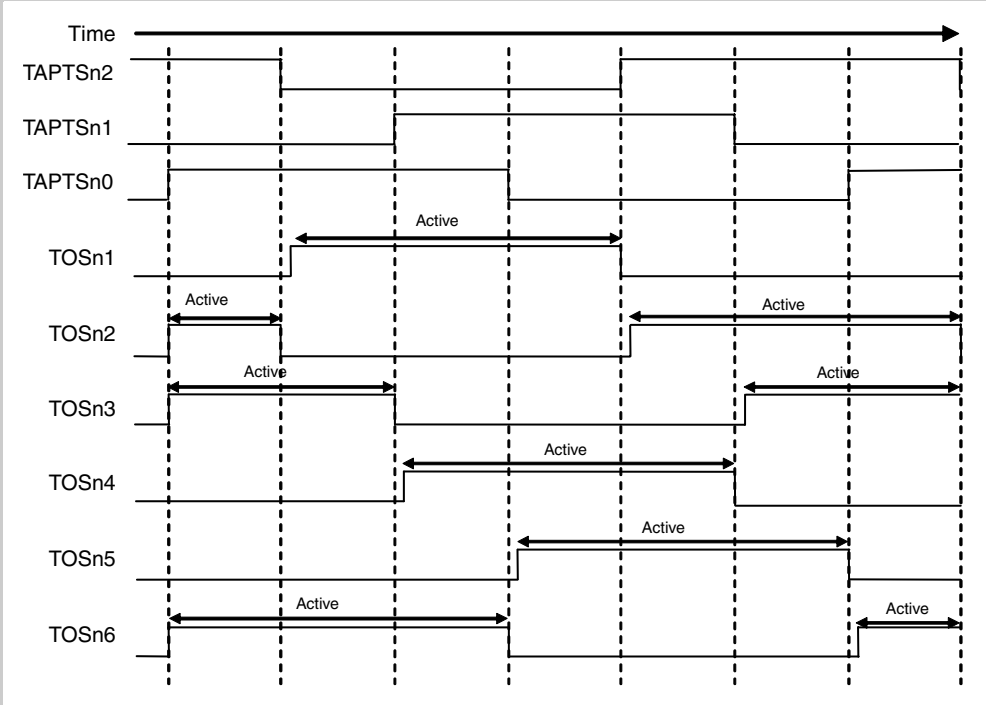
Normal rotation:TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 0,
TSnOPT7 register: TSnIDC = 0



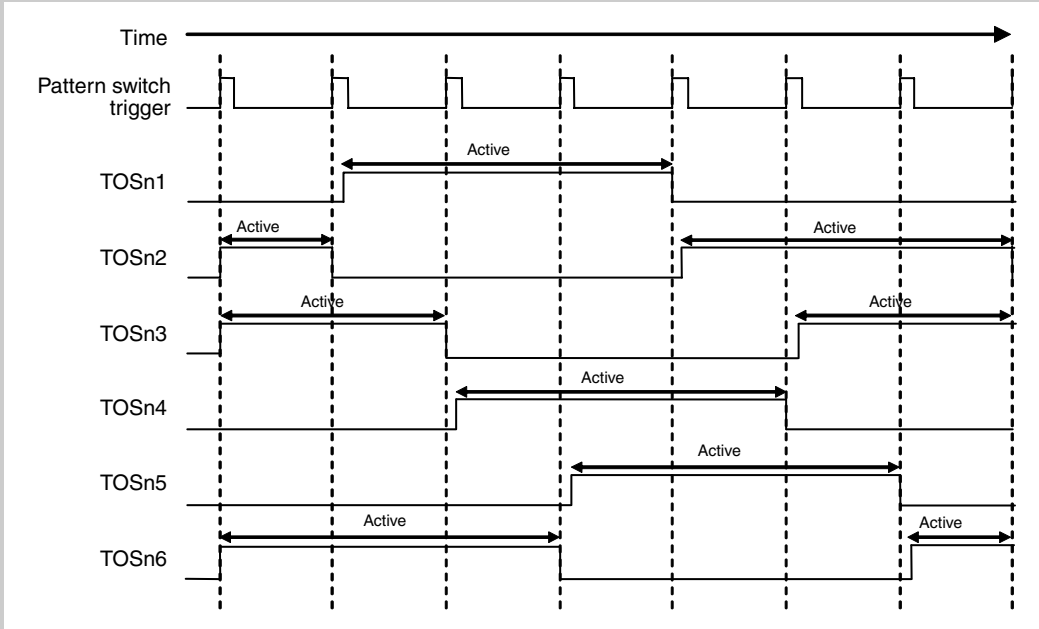
Normal rotation: TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 0



Reverse rotation: TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 0,
TSnOPT7 register: TSnIDC = 1



Reverse rotation: TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 1



Each phase (TOSn1 to TOSn6) outputs the active level for the period of electric angle 180° and then outputs the inactive level for the period of electric angle 180°.

(c) Timer output initial pattern setting in each mode and function

- In 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 1	Note 2
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 1	Note 2
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM1	PWM2	INACT	Note 1	Note 2
TOSn2	PWM3	PWM4	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM2	INACT	INACT	INACT	INACT	PWM1	Note 1	Note 2
TOSn4	INACT	INACT	PWM3	PWM4	INACT	INACT	Note 1	Note 2
TOSn5	INACT	PWM1	PWM2	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM3	PWM4	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.

2. Output remains the level before the change.

3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers

4. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM2	PWM1	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM4	PWM3	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM2	PWM1	INACT	INACT	Note 1	Note 2
TOSn4	PWM3	INACT	INACT	INACT	INACT	PWM4	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM2	PWM1	Note 1	Note 2
TOSn6	INACT	PWM4	PWM3	INACT	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM2	PWM1	INACT	Note 1	Note 2
TOSn2	PWM4	PWM3	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM1	INACT	INACT	INACT	INACT	PWM2	Note 1	Note 2
TOSn4	INACT	INACT	PWM4	PWM3	INACT	PWM3	Note 1	Note 2
TOSn5	INACT	PWM2	PWM1	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM4	PWM3	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.

2. Output remains the level before the change.

3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers

4. INACT: Inactive level output

- In special 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	Note 1	Note 2
TOSn2	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Note 1	Note 2
TOSn3	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Note 1	Note 2
TOSn4	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	Note 1	Note 2
TOSn5	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Note 1	Note 2

Note 1. INACT if no change from the initial value. If changed from any value to [0, 0, 0], output remains the level before the change.

2. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if
   TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4 and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

Note 1. Special patterns 0 to 7: Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers

2. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

←——————————————— Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	Note 1	Note 2
TOSn3	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	Note 1	Note 2
TOSn4	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Note 1	Note 2
TOSn5	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Note 1	Note 2
TOSn6	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	Note 1	Note 2
TOSn2	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Note 1	Note 2
TOSn3	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Note 1	Note 2
TOSn4	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	Note 1	Note 2
TOSn5	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
2. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting leads the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

Note
1. Special patterns 0 to 7:Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
2. INACT: Inactive level output

- When using 180° excitation function

TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 0

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 1

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to 000B, output remains the level before the change.

2. Output remains the level before the change.

3. ACT: Active level output

4. INACT: Inactive level output

(11) Timer Sn option register 5 (TSnOPT5)

The TSnOPT5 register is an 8-bit register that controls 180° excitation control function and timer output (TOSn1 to TOSn6) pattern.

This register can be read/written with an 8-bit or 1-bit manipulation instruction.

Reset input sets this register to 00H.

Address: TS0OPT5: FFFFF589H, TS1OPT5: FFFFF5C9H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT5	TSnADC	TSnPOT	TSnPSS	TSnPTS	TSnTSF	TSnOPF2	TSnOPF1	TSnOPF0	00H

Bit position	Bit name	Function
7	TSnADC	Selects the 180° excitation control function of semi-automatic driving. 0: Semi-automatic driving 180° excitation control function disabled 1: Semi-automatic driving 180° excitation control function enabled When the TSnADC bit is set to 1, the timer output pattern is switched to the pattern preset by the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. Upon input of the next pattern output trigger, the next pattern is output. **Caution:** 1. Do not set the TSnSOC bit of the TSnOPT4 register to 1. 2. **Set the output pattern to TSnIPC2 to TSnIPC0 in advance before setting TSnADC to 1.** **Note:** The TSnADC bit is valid in the high-accuracy T-PWM mode, 120° excitation mode, and special 120° excitation mode.
6	TSnPOT	Selects the pattern output trigger. 0: Output pattern switch by using external pattern input pins (TAPTSn0 to TAPTSn2 pins) (pattern switch method). 1: Output pattern switch by rising edge of the TSnSTCI0 and TSnSTCI1signals (trigger switch method). The TSnPOT bit setting is valid in the 120° excitation mode, special 120° excitation mode and during 180° excitation control in each mode.
5	TSnPSS	Selects the pattern output switch factor. 0: Switch the pattern output order using input signal to the TSnARD signal. 1: Switch the pattern output order using the TSnPSC bit of the TSnOPT4 register. **Caution:** The TSnPSS bit is valid only when the TSnPOT bit is set to 1.

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>4</td><td>TSnPTS</td><td>Selects enabling/disabling toggle output of the TSnAEDO and TSnESG signals.
0: Disable toggle output to the TSnAEDO and TSnESGNote signals at the edge detection timing of TAPTSn0 to TAPTSn2 pins.
1: Enable toggle output to the TSnAEDO and TSnESGNote signals at the edge detection timing of the TAPTSn0 to TAPTSn2 pins.

The level of the TSnAEDO and TSnESG signals is retained if the TSnPTS bit is changed from 1 to 0 during the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1).

Note: The TSnESG signal is provided only in the 120° excitation mode and special 120° excitation mode.</td></tr>
<tr><td>3</td><td>TSnTSF</td><td>This is a flag that indicates the order of pattern change detected with the TAPTSn0 to TAPTSn2 pins.
0: TAPTSn0 to TAPTSn2 pins are input in the normal rotation order.
1: TAPTSn0 to TAPTSn2 pins are input in the reverse rotation order.
<table><tr><td>Normal rotation</td><td>⟶</td></tr><tr><td>Reverse rotation</td><td>⟵</td></tr><tr><td>TAPTSn2 to TAPTSn0</td><td>[1, 0, 1] [1, 0, 0] [1, 1, 0] [0, 1, 0] [0, 1, 1] [0, 0, 1]</td></tr></table>
Note: The normal rotation or reverse rotation can be detected after the first change of the TAPTSn0 to TAPTSn2 pins after TSnCE = 1 is set.
For details, refer to “Input pattern change order detection” on page 747.</td></tr>
<tr><td>2 to 0</td><td>TSnOPF2,
TSnOPF1,
TSnOPF0</td><td>These are flags that indicate the pattern output of the timer output (TOSn1 to TOSn6).
Refer to “Timer output pattern in each mode and function” on page 739 for details.
The TSnOPF0 to TSnOPF2 bits are displayed in the 120° excitation mode, special 120° excitation mode, and when using 180° excitation control function.</td></tr>
</table>

(a) Timer output pattern in each mode and function

- Software control function

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the output level before the change.

2. Output level remains the level before the change.

3. ACT: Active level output

4. INACT: Inactive level output

- In 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 1	Note 2
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 1	Note 2
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC= 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM1	PWM2	INACT	Note 1	Note 2
TOSn2	PWM3	PWM4	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM2	INACT	INACT	INACT	INACT	PWM1	Note 1	Note 2
TOSn4	INACT	INACT	PWM3	PWM4	INACT	INACT	Note 1	Note 2
TOSn5	INACT	PWM1	PWM2	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM3	PWM4	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the output level before the change.

2. Output remains the level before the change.

3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers

4. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

←———————————— Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM2	PWM1	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM4	PWM3	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM 2	PWM 1	INACT	INACT	Note 1	Note 2
TOSn4	PWM3	INACT	INACT	INACT	INACT	PWM4	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM2	PWM1	Note 1	Note 2
TOSn6	INACT	PWM4	PWM3	INACT	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

←———————————— Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM 2	PWM 1	INACT	Note 1	Note 2
TOSn2	PWM4	PWM3	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM1	INACT	INACT	INACT	INACT	PWM2	Note 1	Note 2
TOSn4	INACT	INACT	PWM4	PWM3	INACT	INACT	Note 1	Note 2
TOSn5	INACT	PWM 2	PWM1	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM4	PWM3	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.

2. Output remains the level before the change.

3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers

4. INACT: Inactive level output

- In special 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	Note 1	Note 2
TOSn2	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Note 1	Note 2
TOSn3	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Note 1	Note 2
TOSn4	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	Note 1	Note 2
TOSn5	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
2. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

Note
1. Special patterns 0 to 7: Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
2. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	Note 1	Note 2
TOSn3	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	Note 1	Note 2
TOSn4	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Note 1	Note 2
TOSn5	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Note 1	Note 2
TOSn6	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or
TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

←---------- Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	Note 1	Note 2
TOSn2	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Note 1	Note 2
TOSn3	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Note 1	Note 2
TOSn4	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	Note 1	Note 2
TOSn5	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
2. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

Note
1. Special patterns 0 to 7:Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
2. INACT: Inactive level output

- When using 180° excitation function

TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.

2. Output remains the level before the change.

3. ACT: Active level output

4. INACT: Inactive level output

(b) Input pattern change order detection

Changes in the pattern (normal rotation/reverse rotation) input to the TAPTSn2 to TAPTSn0 pins by the TSnTSF flag are detected from the first change of the TAPTSn2 to TAPTSn0 pins from the time when TSnCE = 1 is set.

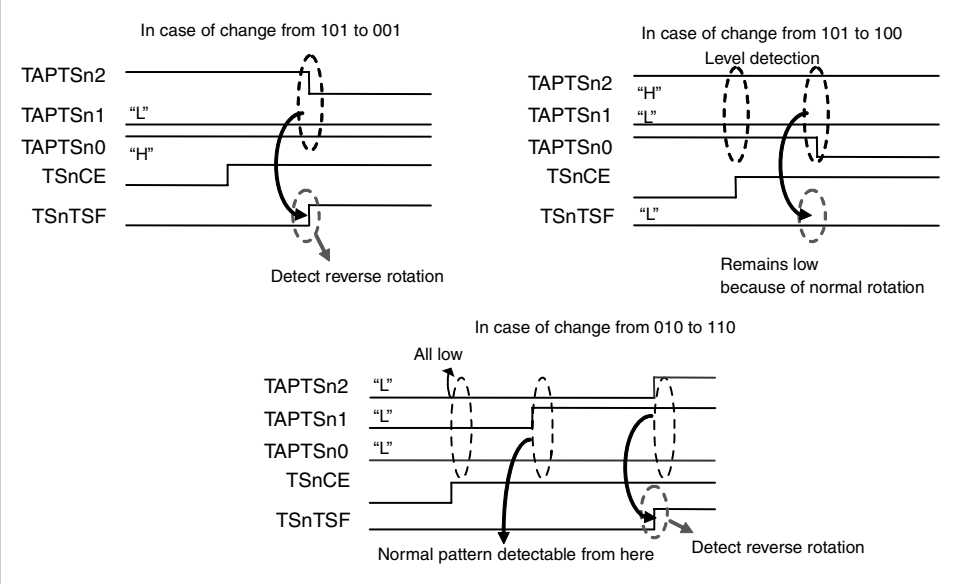


Figure 17-2 Detection example of change in pattern (normal rotation/reverse rotation) Input to TAPTSn2 to TAPTSn0 Pins

(12) Timer Sn option register (TSnOPT6)

The TSnOPT6 register is a 16-bit register that controls each flag of timer Sn. This register can be read/written with a 16-bit, 8-bit, or 1-bit manipulation instruction.
Reset input or clearing TSnCE to 0 sets this register to 0000H.

Note Refer to *"Flags" on page 787* for each flag function.

Address: TS0OPT6H: FFFFF5A5H, TS1OPT6H: FFFFF5E5H
TS0OPT6L: FFFFF5A4H, TS1OPT6L: FFFFF5E4H

	15	14	13	12	11	10	9	8	Initial value
TSnOPT6	0	0	0	0	0	0	0	TSnPPF	0000H
	7	6	5	4	3	2	1	0	
	TSnPTF	TSnTDF	TSnNDF	TSnPRF	TSnPEF	TSnTBF	TSnSUF	TSnRSF	

	7	6	5	4	3	2	1	0	Initial value
TSnOPT6H	0	0	0	0	0	0	0	TSnPPF	00H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT6L	TSnPTF	TSnTDF	TSnNDF	TSnPRF	TSnPEF	TSnTBF	TSnSUF	TSnRSF	00H

Bit position	Bit name	Function
8	TSnPPF	This is a flag that compares input pattern (TAPTSn2 to TAPTSn0 pins) with output pattern (TOSn1 to TOSn6 pins) while the TSnPPC bit of the TSnOPT7 register is set to 1, and detects the differences. 0: Did not detect any phase difference between the TAPTSn2 to TAPTSn0 pins and TOSn1 to TOSn6 pins. 1: Detected phase difference between the TAPTSn2 to TAPTSn0 pins and TOSn1 to TOSn6 pins. The TSnPPF flag is set to 1 when the input/output pattern difference is detected, and the warning interrupt (INTTSnWn) occurs. This flag can be cleared by writing 0. **Note: 1.** The TSnPPF flag is valid when the TSnSOC bit of the TSnOPT4 register is set to 0, the TSnPOT bit of the TSnOPT5 register is set to 1 or the TSnPPC bit of the TSnOPT7 register is set to 1. **2.** The TSnPPF flag is valid in the 120° excitation mode and special 120° excitation mode.
7	TSnPTF	This is a flag that detects abnormal toggle of the TAPTSn2 to TAPTSn0 pins when the value of the TSnPTC1 bit of the TSnOPT7 register is 1. 0: Did not detect any abnormal toggle of the TAPTSn2 to TAPTSn0 pins. 1: Detected abnormal toggle of the TAPTSn2 to TAPTSn0 pins. The TSnPTF flag is set to 1 when the TAPTSn2 to TAPTSn0 pins (TSnAEDO signal toggle) are changed three times or more during the trigger of the TSnSTCI0 signal, or when the TAPTSn2 to TAPTSn0 pins (TSnAEDO signal toggles) are changed three times or more during the trigger of the TSnSTCI1 signal. Then a warning interrupt (INTTSnWN) occurs. The flag can be cleared by writing 0. **Note: 1.** The TSnPTF flag is valid when the TSnSOC bit of the TSnOPT4 register is set to 0, the TSnPOT bit of the TSnOPT5 register is set to 1, or TSnPTC of the TSnOPT7 register is set to 1. **2.** The TSnPTF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.
6	TSnTDF	This is a flag that detects the simultaneous trigger generation of the TSnSTCI0 signal and the TSnSTCI1 signal when the value of the TSnTDC bit of the TSnOPT7 register is 1. 0: The triggers of the TSnSTCI0 signal and the TSnSTCI1 signal do not simultaneously occur 1: Detects the simultaneous occurrence of the TSnSTCI0 signal and TSnSTCI1 signal triggers. The TSnTDF is set to 1 when the simultaneously generated triggers of the TSnSTCI0 signal and the TSnSTCI1 signal are detected. Then the warning interrupt (INTTSnWN) is generated. The flag can be cleared by writing 0. **Note: 1.** The TSnTDF flag is valid when the TSnSOC bit of the TSnOPT4 register is set to 0, the TSnPOT bit of the TSnOPT5 register = 1 or the TSnTDC bit of the TSnOPT7 register is set to 1. **2.** The TSnTDF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.

Bit position	Bit name	Function
5	TSnNDF	This is a flag that detects noise generation at the TAPTSn2 to TAPTSn0 pins. 0: No noise generated due to simultaneous change of 2 or more pins of the TAPTSn2 to TAPTSn0 pins. 1: Detected noise generation due to simultaneous change of 2 or more pins of the TAPTSn2 to TAPTSn0 pins. The TSnNDF flag is set to 1 when simultaneous value changes of 2 or more pins of the TSPTSn2 to TAPTSn0 pins are detected. Then the warning interrupt (INTTSnWN) occurs. The flag can be cleared by writing 0. **Note:** The TSnNDF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.
4	TSnPRF	This is a flag that detects the inversion of the input order of the TAPTSn2 to TAPTSn0 pins. 0: The input order of TAPTSn2 to TAPTSn0 is not inverted. 1: The input order of TAPTSn2 to TAPTSn0 is inverted. The TSnPRF flag is set to 1 at the change timing of the TSnTSF bit of the TSnOPT5 register and a warning interrupt (INTTSnWN) then occurs. The flag can be cleared by writing 0. Detection is possible from the timing of the second TAPTSn2 to TAPTSn0 pin change after the TSnCE bit of the TSnCTL0 register is set to 1. **Note:** The TSnPRF flag is valid when using the 180° excitation control function in the high-accuracy T-PWM mode, 120° excitation mode, or 120° excitation mode
3	TSnPEF	This is a flag that detects [0, 0, 0] or [1, 1, 1] input to the TAPTSn2 to TAPTSn0 pins. 0: No input of [0, 0, 0] or [1, 1, 1] to the TAPTSn2 to TAPTSn0 pins. 1: [0, 0, 0] or [1, 1, 1] input to the TAPTSn2 to TAPTSn0 pins. The TSnPEF flag is set to 1 when [0, 0, 0] or [1, 1, 1] input to TAPTSn2 to TAPTSn0 is detected and a warning interrupt (INTTSnWN) then occurs. The flag can be cleared by writing 0. **Note:** The TSnPEF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.

Bit position	Bit name	Function
2	TSnTBF	This is a flag that detects both of the positive phase and negative phase turns to be active simultaneously when any bit from the TSnTBA2 to TSnTBA0 bits of the TSnIOC4 register is set to 1. 0: Positive phase and negative phase are not active simultaneously. 1: Positive phase and negative phase become active simultaneously. The TSnTBF flag is set to 1 when simultaneous active of any of positive phase (TOSn1, TOSn3, TOSn5) and any of negative phase (TOSn2, TOSn4, TOSn6) is detected. Then the error interrupt (INTTSnER) occurs. The flag can be cleared by writing 0. Simultaneous active is not detected when TSnTBA2 to TSnTBA0 = [0, 0, 0].
1	TSnSUF	This is a flag that detects count up/down of 16-bit sub-counter. 0: Sub-counter performing count-up operation. 1: Sub-counter performing count-down operation. For the TSnSUF flag detection, count-up is the count from 0000H to (TSnCCR0 register value – 0002H), and count-down is the count from the TSnCCR0 register value to 0002H of sub-counter count operation. **Note: 1.** TSnSUF flag is read only. **2.** TSnSUF flag is valid only in the high-accuracy T-PWM mode.
0	TSnRSF	This is a flag that shows whether or not there is a reload request. 0: No reload request, or reload completed Enable write access to the TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1 registers. 1: Reload request issued (Disable write access to the TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1 registers) The TSnRSF flag indicates that the next transfer data is held in registers TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1. The TSnRSF flag is set to 1 by writing to the TSnCCR1 register and cleared to 0 upon reload completion. **Caution:** The written value of the register may be undefined if writing to the TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1 registers and the reload timing conflict while TSnRSF = 1.

(13) Timer Sn option register 7 (TSnOPT7)

The TSnOPT7 register is an 8-bit register that detects the abnormal timer output and controls switching of the timer output.
This register can be read/written with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 00H.

Address: TS0OPT7: FFFFF587H, TS1OPT7: FFFFF5C7H

	7	6	5	4	3	2	1	0	Initial value
TSnOPT7	0	0	TSnIDC	TSnPPC	TSnTDC	TSnPTC1	TSnPTC0	TSnTOS	00H

Bit position	Bit name	Function
5	TSnIDC	Sets the output pattern of TOSn1 to TOSn6 by combination of the TSnTSF bit of the TSnOPT5 register, the TSnPSC bit of the TSnOPT4 register and the TSnARD signal. Refer to *"Timer output order in each mode and function" on page 721* and *"Timer output pattern in each mode and function" on page 739* for the output pattern. TSnSOC of TSnOPT4 Register / TSnPOT of TSnOPT5 Register / TSnPSS of TSnOPT5 Register / Rotation Direction 1 / - / - / - 0 / 0 / - / TSnTSF flag 0 / 1 / 0 / Input signal to TSnARD 0 / 1 / 1 / TSnPSC bit **Caution:** The TSnIDC bit is valid in the 120° excitation mode, special 120° excitation mode, 180° excitation control, and software output control. 0 must be set in any other mode.
4	TSnPPC	Enables/disables input/output pattern difference detection (TSnPPF flag of the TSnOPT6 register, warning interrupt (INTTSnWN)). 0: Disable input/output pattern difference detection 1: Enable input/output pattern difference detection **Note:** TSnPPC = 0 disables the TSnPPF bit detection and the warning interrupt (INTTSnWN). Also, input/output pattern difference detection flag (TSnPPF bit of the TSnOPT6 register) is not controlled.
3	TSnTDC	Controls the simultaneous trigger detection of the TSnSTCI0 and TSnSTCI1 signals (TSnTDF flag of the TSnOPT6 register, the warning interrupt (INTTSnWN)). 0: Disables simultaneous trigger detection of the TSnSTCI0 and TSnSTCI1 signals. 1: Enables simultaneous trigger detection of the TSnSTCI0 and TSnSTCI1 signals. **Note:** TSnTDC = 0 disables the warning interrupt (INTTSnWN). Also, simultaneous trigger detection flag (the TSnTDF bit of the TSnOPT6 register) is not controlled.

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2, 1</td><td>TSnPTC1,
TSnPTC0</td><td>Controls detection of abnormal toggle of the TAPTSn2 to TAPTSn0 pins (the TSnPTF flag of the TSnOPT6 register, warning interrupt (INTTSnWN)).
<table>
<tr><th>TSnPTC1</th><th>TSnPTC0</th><th>Operations</th></tr>
<tr><td>0</td><td>0/1</td><td>• Don’t detect abnormal toggle of the TAPTSn2 to TAPTSn0 pins.</td></tr>
<tr><td>1</td><td>0</td><td>• Detect abnormal toggle of the TAPTSn2 to TAPTSn0 pins.
• Don’t change timer output switch method by abnormal toggle detection (S/W from the TSnPOT bit of the TSnOPT5 register 1 to 0).</td></tr>
<tr><td>1</td><td>1</td><td>• Detect abnormal toggle of the TAPTSn2 to TAPTSn0 pins.
• Change the timer output switch method through abnormal toggle detection (S/W from the TSnPOT bit of the TSnOPT5 register 1 to 0).</td></tr>
</table>
When the TSnPTC1 bit is set to 1, a warning interrupt (INTTSnWN) occurs when three or more toggles (abnormal toggles) are detected by the TSnAPTS2 to TSnAPTS0 pin inputs during the trigger of the TSnSTCI1 or TSnSTCI0 signal input. Then the TSnAPTS2 to TSnAPTS0 pin abnormal toggle detection flag (TSnPTF) is set to 1.</td></tr>
<tr><td>0</td><td>TSnTOS</td><td>Controls the TOSn0 pin output switch.
0: Outputs count up/down flag of 16-bit counter to TOSn0 pin.
1: Outputs count up/down flag of 16-bit sub-counter to TOSn0 pin.

When the TSnTOS bit is 0, the TSnCUF bit status of the TSnOPT0 register is output to the TOSn0 pin. When the TSnTOS bit is 1, the TSnSUF bit status of the TSnOPT6 register is output to the TOSn0 pin.

Note: The TSnTOS bit is valid only in the high-accuracy T-PWM mode.</td></tr>
</table>

(14) Timer Sn pattern register 0 (TSnPAT0)

The TSnPAT0 register is a 16-bit register that controls output patterns in the special 120° excitation mode and special pattern output mode.
This register can be read/written with a 16-bit manipulation instruction. Do not read or write it with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 0000H.

Caution The TSnPAT0 register setting must be executed while the timer Sn operation is stopped (when the TSnCE bit of the TSnCTL0 register is set to 0). In order to change the value of the TSnPAT0 register during the timer Sn operation (when the value of the TSnCE bit of the TSnCTL0 register is 1), do so after setting the TSnCE bit to 1 and waiting for 1 cycle of the 16-bit counter to be completed.

Address: TS0PAT0: FFFFF596H, TS1PAT0: FFFFF5D6H

	15	14	13	12	11	10	9	8	Initial value
TSnPAT0	Special pattern 3				Special pattern 2				0000H
	7	6	5	4	3	2	1	0	
	Special pattern 1				Special pattern 0				

Bit position	Bit name	Function
15 to 12	-	Controls the output pattern of special pattern 3 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn3 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-4 on page 755* for the output pattern.
11 to 8	-	Controls the output pattern of special pattern 2 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn2 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-4 on page 755* for the output pattern.
7 to 4	-	Controls the output pattern of special pattern 1 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn1 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-4 on page 755* for the output pattern.
3 to 0	-	Controls output pattern of special pattern 0 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn0 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-4 on page 755* for the output pattern.

Table 17-4 Output Pattern Setting for TSnPAT0 Register

TSnPAT0 register bit setting: bits 15 to 12 (special pattern 3), 11 to 8 (special pattern 2), 7 to 4 (special pattern 1), 3 to 0 (special pattern 0)	Output level by match between TSnCCRm register and 16-bit counter			
	Match with TSnCCR3 register	Match with TSnCCR2 register	Match with TSnCCR1 register	Match with TSnCCR0 register
0, 0, 0, 0	INACT[Note 1]	INACT[Note 1]	INACT[Note 1]	INACT[Note 1]
0, 0, 0, 1	INACT	INACT	INACT	ACT
0, 0, 1, 0	INACT	INACT	ACT	INACT
0, 0, 1, 1	INACT	INACT	ACT	ACT
0, 1, 0, 0	INACT	ACT	INACT	INACT
0, 1, 0, 1	INACT	ACT	INACT	ACT
0, 1, 1, 0	INACT	ACT	ACT	INACT
0, 1, 1, 1	INACT	ACT	ACT	ACT
1, 0, 0, 0	ACT	INACT	INACT	INACT
1, 0, 0, 1	ACT	INACT	INACT	ACT
1, 0, 1, 0	ACT	INACT	ACT	INACT
1, 0, 1, 1	ACT	INACT	ACT	ACT
1, 1, 0, 0	ACT	ACT	INACT	INACT
1, 1, 0, 1	ACT	ACT	INACT	ACT
1, 1, 1, 0	ACT	ACT	ACT	INACT
1, 1, 1, 1	ACT[Note 2]	ACT[Note 2]	ACT[Note 2]	ACT[Note 2]

Note 1. Inactive level is output regardless of match with the TSnCCRm register.

2. Active level is output regardless of match with the TSnCCRm register.

3. ACT: Active level output

4. INACT: Inactive level output

5. m = 1 to 3

(15) Timer Sn pattern register 1 (TSnPAT1)

The TSnPAT1 register is a 16-bit register that controls output patterns in the special 120° excitation mode and special pattern output mode.
This register can be read/written with a 16-bit manipulation instruction. Do not read or write it with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 0000H.

Caution The TSnPAT1 register setting must be executed while the timer Sn operation is stopped (when the TSnCE bit of the TSnCTL0 register is set to 0). In order to change the value of the TSnPAT0 register during the timer Sn operation (when the value of the TSnCE bit of the TSnCTL0 register is 1), do so after setting the TSnCE bit to 1 and waiting for 1 cycle of the 16-bit counter to be completed.

Address: TS0PAT1: FFFFF594H, TS1PAT1: FFFFF5D4H

	15	14	13	12	11	10	9	8	Initial value
TSnPAT1	Special pattern7				Special pattern 6				0000H
	7	6	5	4	3	2	1	0	
	Special pattern 5				Special pattern 4				

Bit position	Bit name	Function
15 to 12	-	Controls the output pattern of special pattern 7 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. Output pattern of the TOSn7 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-5 on page 757* for the output pattern.
11 to 8	-	Controls the output pattern of special pattern 6 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. The output pattern of the TOSn6 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-5 on page 757* for the output pattern.
7 to 4	-	Controls the output pattern of special pattern 5 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. The output pattern of the TOSn5 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-5 on page 757* for the output pattern.
3 to 0	-	Controls the output pattern of special pattern 4 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. The output pattern of TOSn4 pin is controlled in the special pattern output mode. **Note:** Refer to *Table 17-5 on page 757* for the output pattern.

Table 17-5 Output pattern setting for TSnPAT1 register

TSnPAT1 register bit setting: bits 15 to 12 (special pattern 7), 11 to 8 (special pattern 6), 7 to 4 (special pattern 5), 3 to 0 (special pattern 4)	Output level based on match between the tsnccrm register and 16-bit counter			
	Match with TSnCCR3 register	Match with TSnCCR2 register	Match with TSnCCR1 register	Match with TSnCCR0 register
0, 0, 0, 0	INACT[Note 1]	INACT[Note 1]	INACT[Note 1]	INACT[Note 1]
0, 0, 0, 1	INACT	INACT	INACT	ACT
0, 0, 1, 0	INACT	INACT	ACT	INACT
0, 0, 1, 1	INACT	INACT	ACT	ACT
0, 1, 0, 0	INACT	ACT	INACT	INACT
0, 1, 0, 1	INACT	ACT	INACT	ACT
0, 1, 1, 0	INACT	ACT	ACT	INACT
0, 1, 1, 1	INACT	ACT	ACT	ACT
1, 0, 0, 0	ACT	INACT	INACT	INACT
1, 0, 0, 1	ACT	INACT	INACT	ACT
1, 0, 1, 0	ACT	INACT	ACT	INACT
1, 0, 1, 1	ACT	INACT	ACT	ACT
1, 1, 0, 0	ACT	ACT	INACT	INACT
1, 1, 0, 1	ACT	ACT	INACT	ACT
1, 1, 1, 0	ACT	ACT	ACT	INACT
1, 1, 1, 1	ACT[Note 2]	ACT[Note 2]	ACT[Note 2]	ACT[Note 2]

Note 1. Inactive level is output regardless of match with the TSnCCRm register.

2. Active level is output regardless of match with the TSnCCRm register.

3. ACT: Active level output

4. INACT: Inactive level output

5. m = 1 to 3

(16) Timer Sn extension control register (TSnXCTL)

The TSnXCTL register is an 8-bit register which controls the extension operation mode of the high-accuracy T-PWM mode.
This register can be read/written with an 8-bit or 1-bit manipulation instruction.
Reset input sets this register to 00H.

Caution Do not rewrite the TSnXCTL register during the timer Sn operation (when TSnCE of the TSnCTL0 register is set to 1). However, write access of the same value is possible.

Address: TS0XCTL: FFFFFB58H, TS1XCTL: FFFFFBD8H

	7	6	5	4	3	2	1	0	Initial value
TSnXCTL	0	0	0	0	0	0	TSnXSL	TSnXEN	00H

Bit position	Bit name	Function
1	TSnXSL	Specifies the extension operation mode. 0: Center method mode 1: Symmetric interchange method mode **Note:** The TSnXSL bit is valid only in the high-accuracy T-PWM mode.
0	TSnXEN	Enables the extension operation mode. 0: Extension operation mode disabled 1: Extension operation mode enabled **Note:** The TSnXEN bit is valid only in the high-accuracy T-PWM mode. In all other modes disable the extension operation mode (TSnXEN = 0).

17.4 Basic Operation

17.4.1 16-bit counter basic operation

The 16-bit counter basic operation is described below. Refer to *"Operation in Each Mode" on page 827* for details.

(1) Count start operation

The 16-bit counter of timer Sn starts counting from the initial value FFFFH[Note] in every mode except the high-accuracy T-PWM mode.
Count-up operation is performed such as FFFFH, 0000H, 0001H, 0002H, 0003H …
For the high-accuracy T-PWM mode, refer to *"16-bit counter operation in high-accuracy T-PWM mode" on page 873*.

Note FFFEH in high-accuracy T-PWM mode.

(2) Clear operation

The 16-bit counter is cleared to 0000H upon a match between the values of the 16-bit counter and compare register. Count operation from FFFFH to 0000H is not detected as clear operation immediately after count start or when overflow.

(3) Overflow operation

Overflow of the 16-bit counter occurs at the timing of the switch from FFFFH to 0000H in the free-running mode. If overflow occurs, the TSnOVF bit of the TSnOPT0 register is set to 1 and an overflow interrupt (INTTSnOV) occurs. The overflow interrupt (INTTSnOV) does not occur immediately after count operation start or when the compare value is cleared by detecting match at FFFFH.

(4) Counter read operation during count operation

Timer Sn can read the 16-bit counter value with the TSnCNT register during count operation.

(5) Interrupt operation

The following interrupt signals are generated in timer Sn.

- INTTSnCC0: Functions as a TSnCCRn0 buffer register match interrupt.
- INTTSnCC1: Functions as a TSnCCRn1 buffer register match interrupt.
- INTTSnCC2: Functions as a TSnCCRn2 buffer register match interrupt.
- INTTSnCC3: Functions as a TSnCCRn3 buffer register match interrupt.
- INTTSnCC4: Functions as a TSnCCRn4 buffer register match interrupt.
- INTTSnCC5: Functions as a TSnCCRn5 buffer register match interrupt.
- INTTSnOD0: Functions as a peak interrupt at the S/W timing from count-up operation to count-down operation of the 16-bit counter.
- INTTSnOD: Functions as a trough interrupt at the S/W timing from count-down operation to count-up operation of the 16-bit counter.
- INTTSnOV: Functions as an overflow interrupt.
- INTTSnER: Functions as a positive/negative phase simultaneous active detection interrupt.
- INTTSnWN: Functions as each type warning detection interrupt.

17.4.2 Compare register rewriting

The reload function is valid in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode (when the TSnCMS bit is 0), PWM mode with dead time, external trigger pulse output mode, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Reload setting in any other mode than above is invalid.
The compare registers and control registers with reload function are as follows.

- TSnCCR0 to TSnCCR5
- TSnOPT1
- TSnPAT0, TSnPAT1
- TSnDTC0, TSnDTC1

There are the following two modes for rewriting compare registers with the reload function.

- Anytime rewrite mode
  In the anytime rewrite mode, writing to each compare register during update operation by each independent compare register enables write access value updating anytime.
- Reload mode (batch rewrite function)
  Reload is enabled by writing to the TSnCCR1 register (reload request flag TSnRSF is set) and then the values of the target registers are updated simultaneously at the next reload timing. The reload timing is the peak/trough timing of a 16-bit counter. Control is executed by the TSnRTE and TSnRBE bits of the TSnOPT1 register, respectively. Reload is not enabled even if writing to any other register than TSnCCR1.

Rewriting of compare registers is possible with DMA transfer.

The DMA transfer can start at each register. The address is incremented after each DMA transfer and always stops after the DMA transfer to the TSnCCR1 register.

Table 17-6 shows the order of the DMA transfers.

Table 17-6 Timer S DMA transfer (1/2)

Timer	Address	Register name	DMA transfer order	
TMS0	FFFFFB40H	TS0OPT1A	↓	Start registers
	FFFFFB42H	TS0DTC1A	↓	
	FFFFFB44H	TS0DTC0A	↓	
	FFFFFB46H	TS0CCR5A	↓	
	FFFFFB48H	TS0CCR4A	↓	
	FFFFFB4AH	TS0CCR3B	↓	
	FFFFFB4CH/ FFFFF594H[a]	TS0CCR2B/ TS0PAT1[a]	↓	
	FFFFFB4EH/ FFFFF596[a]	TS0CCR1B/ TS0PAT0[a]	↓	
	FFFFFB50H	TS0CCR0A	↓	
	FFFFFB52H	TS0CCR3A	↓	
	FFFFFB54H	TS0CCR2A	↓	
	FFFFFB56H	TS0CCR1A	↓	End register

Table 17-6 Timer S DMA transfer (2/2)

Timer	Address	Register name	DMA transfer order	
TMS1	FFFFFBC0H	TS1OPT1A	|	Start registers
	FFFFFBC2H	TS1DTC1A	|	
	FFFFFBC4H	TS1DTC0A	|	
	FFFFFBC6H	TS1CCR5A	|	
	FFFFFBC8H	TS1CCR4A	|	
	FFFFFBCAH	TS1CCR3B	|	
	FFFFFBCCH/ FFFFF5D4[a]	TS1CCR2B/ TS1PAT1[a]	|	
	FFFFFBCEH/ FFFFF5D6[a]	TS1CCR1B/ TS1PAT0[a]	|	
	FFFFFBD0H	TS1CCR0A	|	
	FFFFFBD2H	TS1CCR3A	|	
	FFFFFBD4H	TS1CCR2A	|	
	FFFFFBD6H	TS1CCR1A	↓	End register

a) If Timer Sn is operated in high-accuracy T-PWM mode, TSnCCR[2:1]B is used, otherwise TSnPAT[1:0]

Refer to *"Interrupt Thinning out Function" on page 805* for details on the interrupt thinning out function through the TSnOPT1 register setting.

Table 17-7 Rewrite timing by mode

Mode	Rewrite timing
Interval timer mode	Anytime rewrite
External event count mode	Anytime rewrite
External trigger pulse output mode	Reload**Note 3**
One-shot pulse mode	Anytime rewrite
PWM mode	Reload
Free-running mode	Anytime rewrite
Triangular-wave PWM mode	Reload**Note 1**
High-accuracy T-PWM mode	Anytime rewrite/Reload**Note 2**
PWM mode with dead time	Reload
120° excitation mode	Reload
Special 120° excitation mode	Reload
Special pattern output mode	Reload

Note
1. Rewriting only in trough interrupt
2. Setting with the TSnCMS bit of the TSnOPT0 register
3. Reload thinning out function disabled

(1) Anytime rewrite mode

In the anytime rewrite mode, the value written to each compare register is immediately transferred to the internal buffer register as the comparison target for the counter value.
The value is written to the compare registers (such as TSnCCR0 to TSnCCR5) and then transferred to the internal buffer register in 4 clocks (f_{TMSn}). However, only for the TSnCCR1 register, transfer timing is in 5 clocks (f_{TMSn}) because of its 3-step buffer composition.

Note In the high-accuracy T-PWM mode, the mode is the anytime rewrite mode when the TSnCMS bit of the TSnOPT0 register is set to 1. The settings of the TSnIOE, TSnICE, TSnRDE, and TSnID4 to TSnID0 bits of the TSnOPT1 register are ignored.

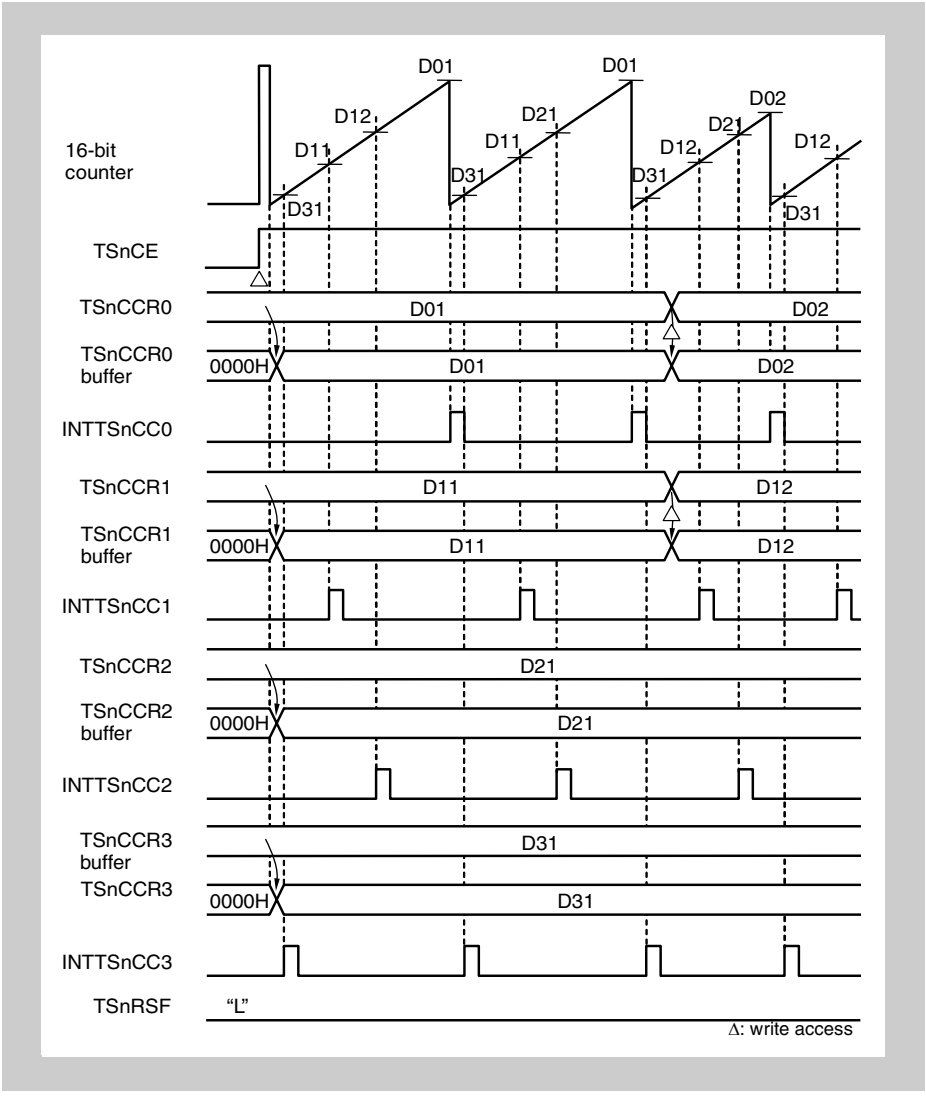


Figure 17-3 Anytime rewrite timing (example in PWM mode)

Note 1. D_{01}, D_{02}: TSnCCR0 register setting values (0000H to FFFFH)
D_{11}, D_{12}: TSnCCR1 register setting values (0000H to FFFFH)
D_{21}: TSnCCR2 register setting value (0000H to FFFFH)
D_{31}: TSnCCR3 register setting value (0000H to FFFFH)

2. Timing chart where interval timer mode is provided as an example

3. Δ: CPU write access

(a) Caution on TSnCCR0 register rewriting in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, transfer to the TSnCCR0 buffer register is not executed if the TSnCCR0 register is rewritten in the anytime rewrite mode. The timing is shown below.

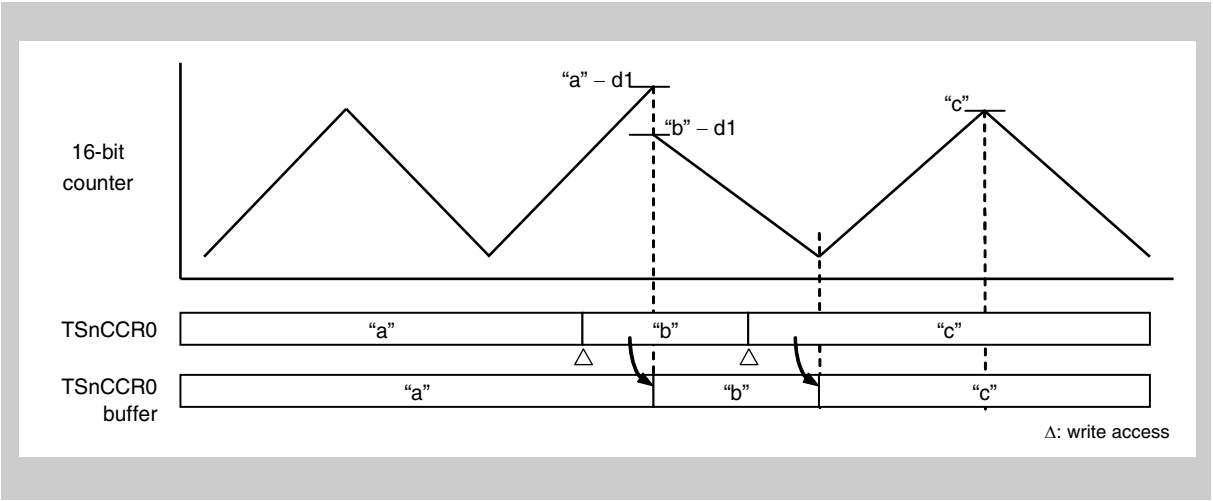
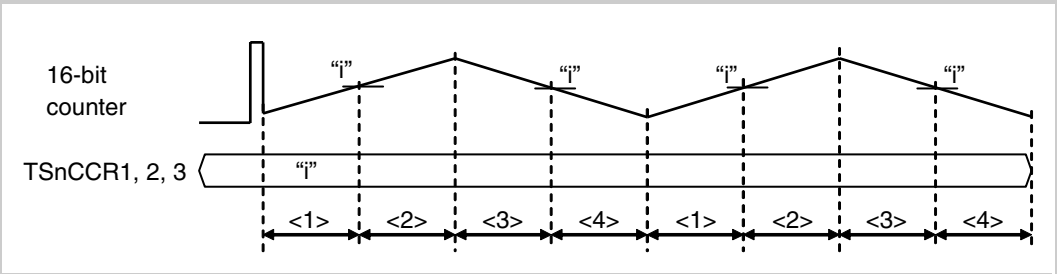


Figure 17-4 TSnCCR0 register rewrite timing in anytime rewrite mode (high-accuracy T-PWM mode)

Note d1: TSnDTC1 setting value

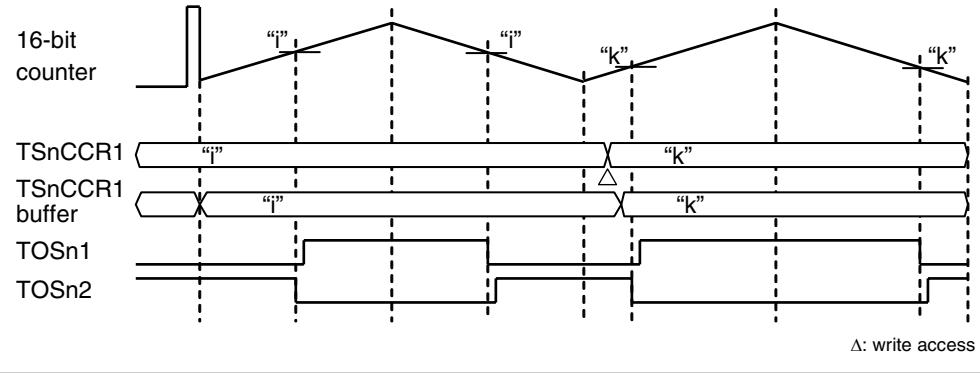
After rewriting of the TSnCCR0 register, the TSnCCR0 register value is transferred to the TSnCCR0 buffer register at the next peak/trough timing. The settings of the TSnIOE, TSnICE, TSnRDE, and TSnID4 to TSnID0 bits of the TSnOPT1 register are ignored because TSnCMS of TSnOPT0 is 1 (anytime rewrite mode).

(b) Caution on rewriting in high-accuracy T-PWM mode for TSnCCR1 to TSnCCR3 registers

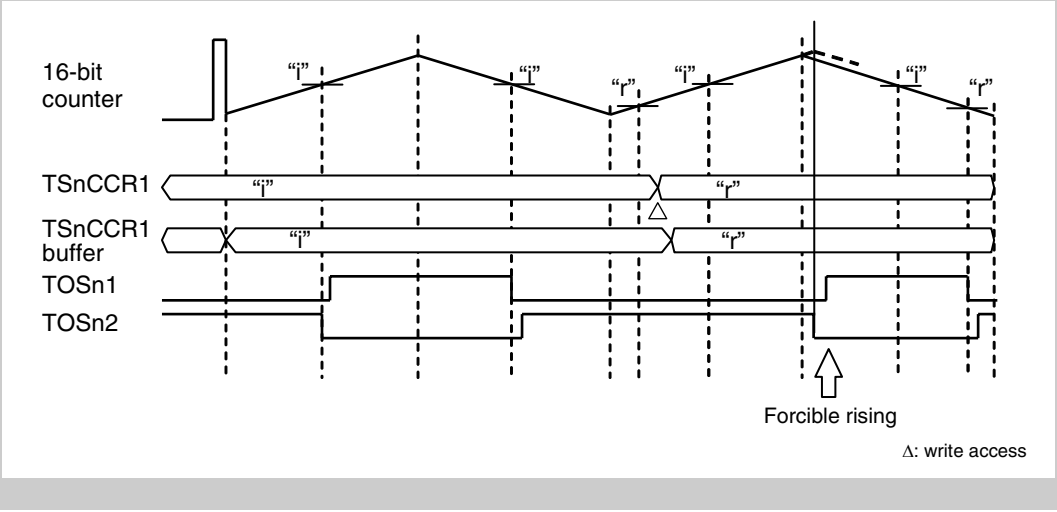


Rewriting during period <1> (rewriting before match detection)

When rewriting before match of the 16-bit counter value with the TSnCCR1 to TSnCCR3 values, the rewritten value is reflected immediately by detection of the match with the 16-bit counter value after rewriting.

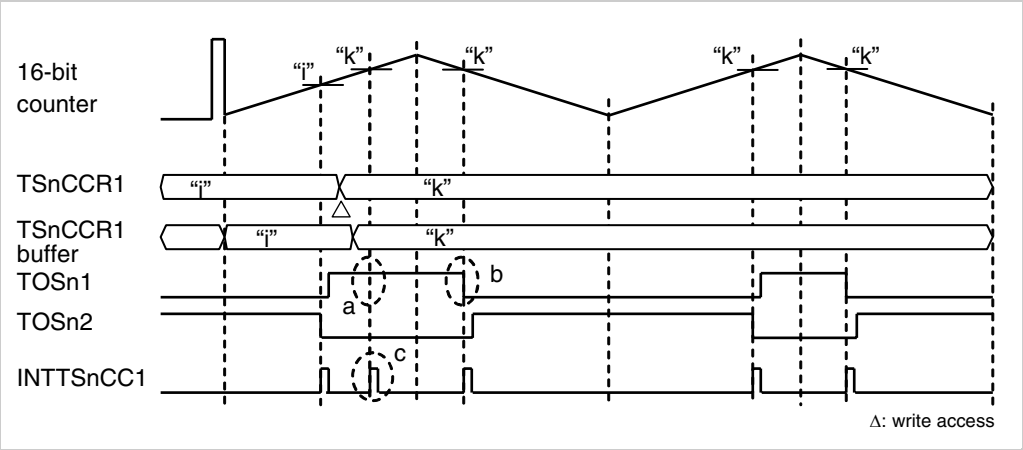


If a value smaller than the 16-bit counter value is rewritten before a match, the match is not detected, so that the output waveform is as follows. TOSn1, TOSn3, and TOSn5 forcibly change to active and TOSn2, TOSn4, and TOSn6 to inactive at the peak even without detection.



Rewriting at period <2> (rewriting after match detection)

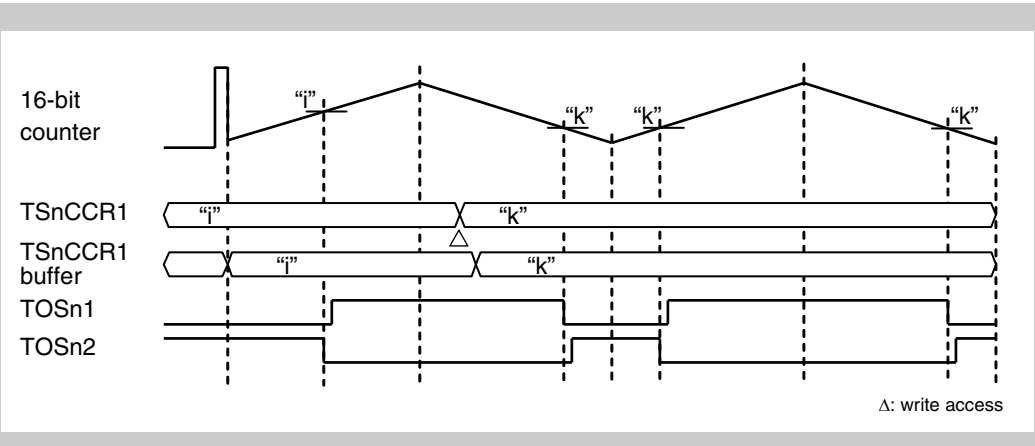
The timer output level does not change upon re-match after match detection if rewritten after match of the 16-bit counter value with the TSnCCR1 to TSnCCR3 register values.



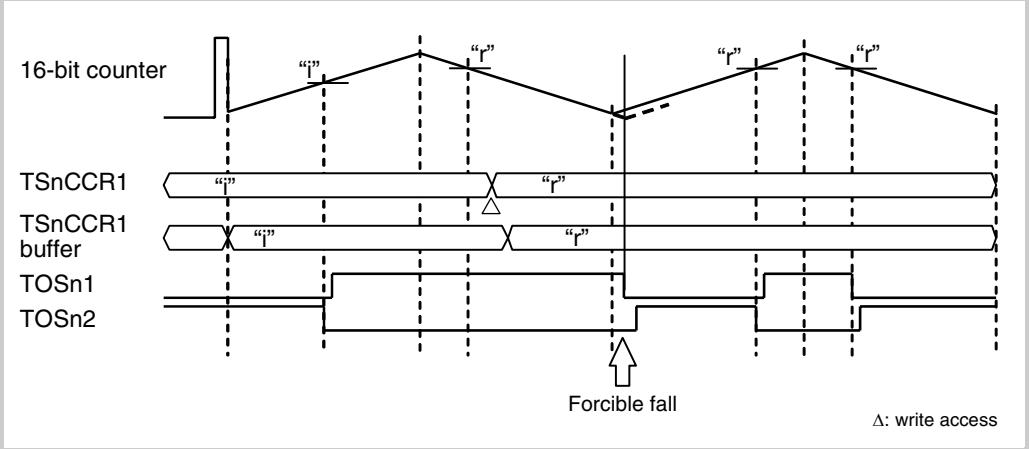
Note
1. The timer output level does not change upon rematch through rewriting after match detection.
2. The timer output level changes from the first match after up/down switch.
3. However, the interrupt output of INTTSnCC1 is match interrupt output.

Rewriting at period <3> (rewriting before match detection)

When rewriting before match of the 16-bit counter value with the TSnCCR1 to TSnCCR3 values, the rewritten value is reflected immediately by detection of the match with the 16-bit counter value after rewriting.

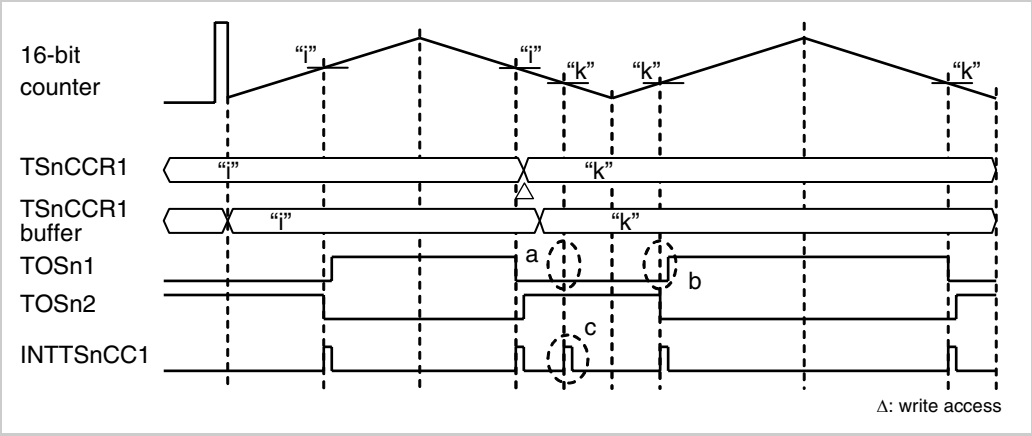


If a value larger than the 16-bit counter value is rewritten before match, match is not detected, so that the output waveform is as shown below. TOSn1, TOSn3, and TOSn5 forcibly change to inactive and TOSn2, TOSn4, and TOSn6 to active at the trough even without match detection.



Rewriting at period <4> (rewriting after match detection)

The timer output level does not change upon re-match after match detection if rewritten after match of the 16-bit counter value with the values of the TSnCCR1 to TSnCCR3 registers.



Note 1. The timer output level does not change even upon re-match through rewriting after match detection.

2. The timer output level changes from the first match after up/down switch.

3. However, the interrupt output of INTTSnCC1 is match interrupt output.

(2) Reload mode (batch rewrite function)

In the reload mode, the values written to the reload target registers (TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1) are transferred to each buffer register simultaneously at the reload timing. Reload target register value must be rewritten while the reload request flag (TSnRSF of the TSnOPT6 register) is 0.

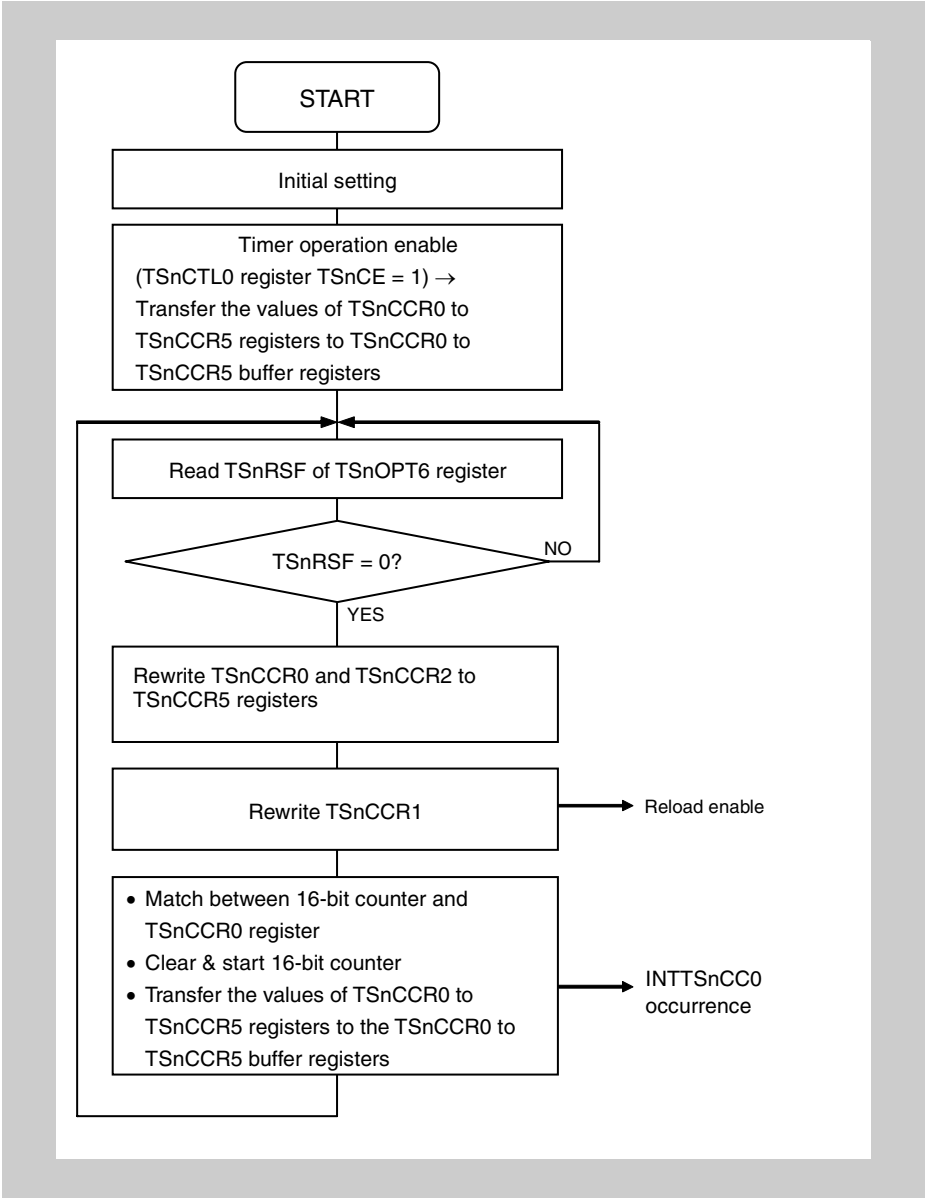


Figure 17-5 Basic operation in reload mode (batch rewrite function) <example in PWM mode>

Caution The write access to the TSnCCR1 register includes the operation that enables reload. Therefore, do not rewrite any TSnCCR1 register value before rewriting the other TSnCCR registers.

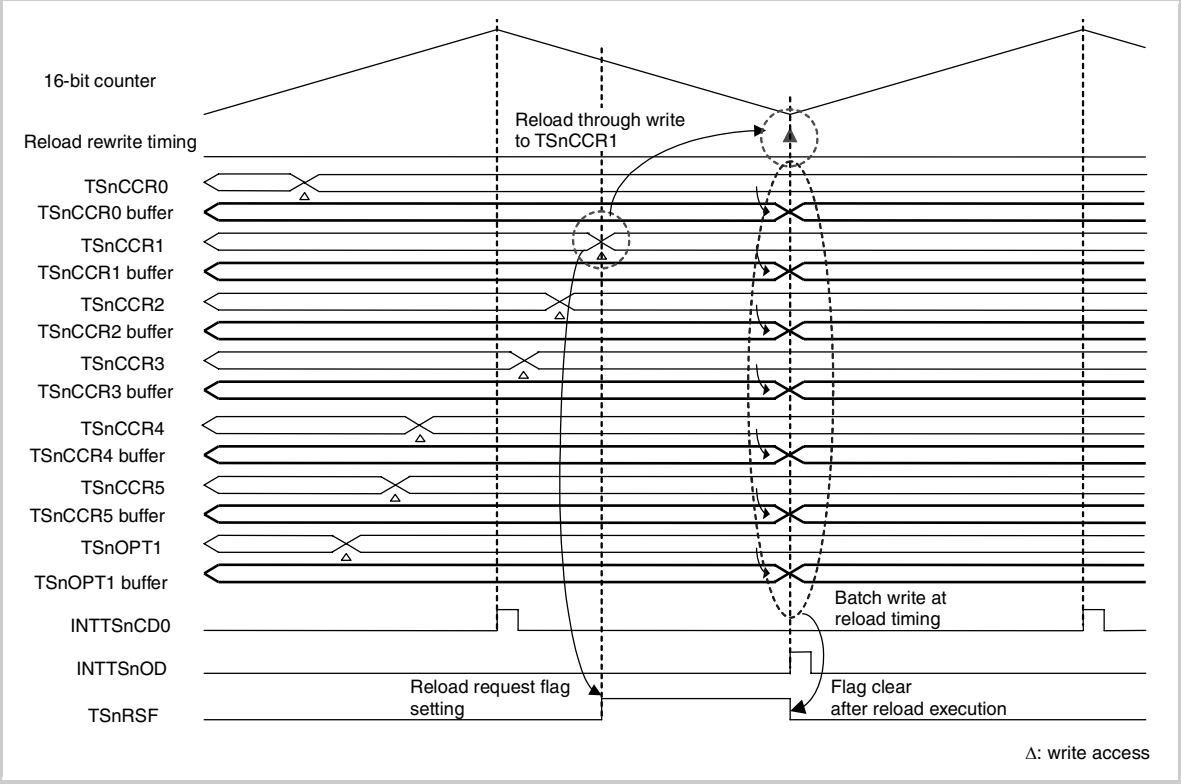


Figure 17-6 Batch rewrite timing (1/2) (high-accuracy T-PWM mode)

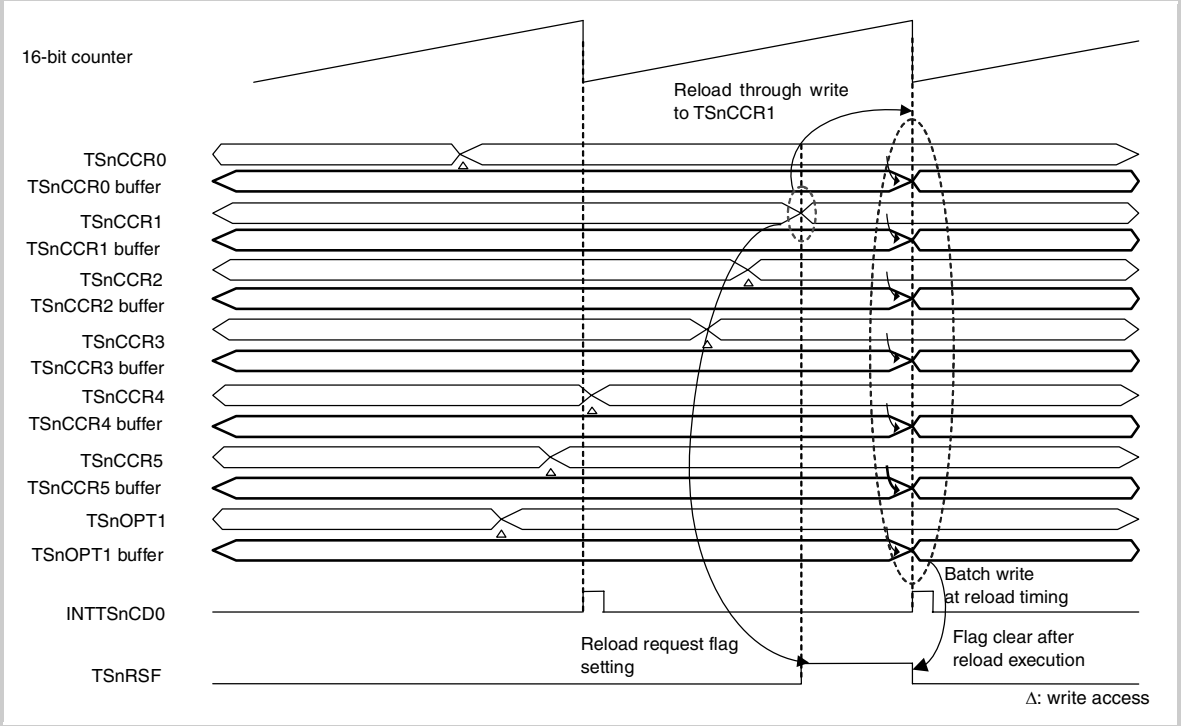


Figure 17-7 Batch rewrite timing (2/2) (PWM mode)

(a) Example of reload rewriting setting in each mode

The following is a setting example of the reload enable mode. Performing the settings listed in *Table 17-8 on page 770* results in reload. Reload does not occur in any other setting than this combination. Refer to *Table 17-3 on page 708* for details.

Table 17-8 Reload rewriting setting by mode

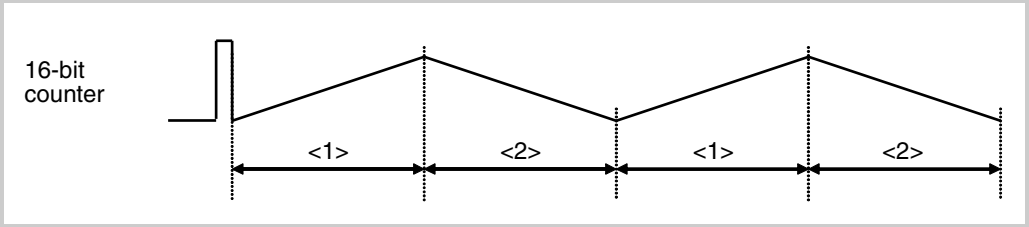
Setting bit mode	TSnCMS of TSnOPT0 register	TSnRDE of TSnOPT1 register	TSnICE of TSnOPT1 register	TS0nRTE of TSnOPT1 register	TSnIOE of TSnOPT1 register	TSnRBE of TSnOPT1 register
External trigger pulse output	0	0	×	1	×	×
PWM mode	0	0	×	1	×	×
	0	1	1	1	×	×
Triangular-wave PWM	0	0	×	×	×	1
	0	1	×	×	1	1
High-accuracy T-PWM	0	0	×	×Note 1	×	×Note 1
	0	1	×	×	1	1
	0	1	1	1	×	×
	0	1	1	1	1	1
PWM with dead time	0	0	×	1	×	×
	0	1	1	1	×	×
120° excitation	0	0	×	1	×	×
	0	1	1	1	×	×
Special 120° excitation	0	0	×	1	×	×
	0	1	1	1	×	×
Special pattern output	0	0	×	1	×	×
	0	1	1	1	×	×

Note 1. Except for TS0nRTE = 0 and TSnRBE = 0

2. × indicates 1 or 0.

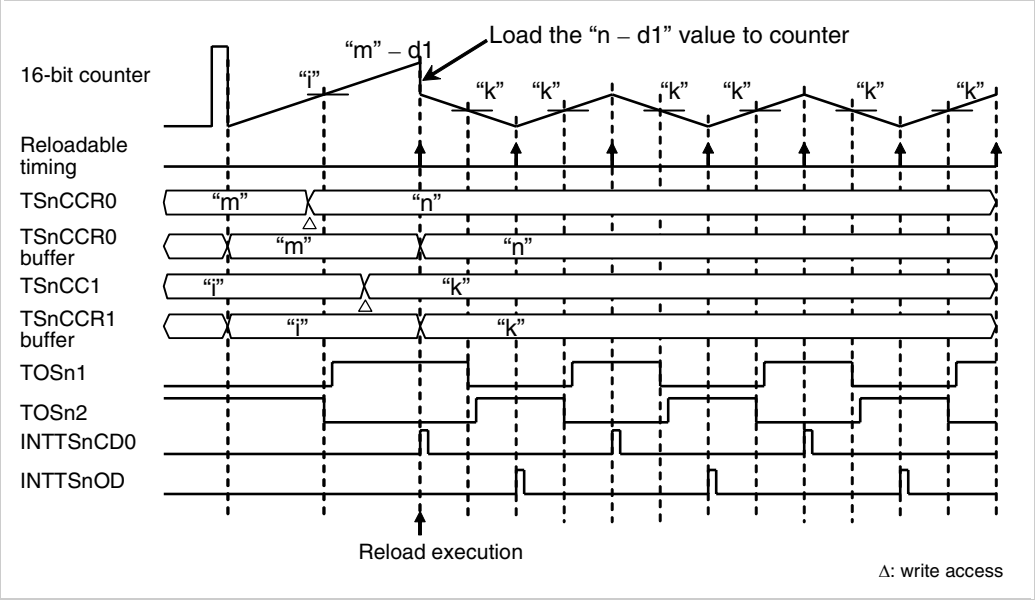
(b) Caution for TSnCCR0 register in high-accuracy T-PWM mode

The output waveform changes according to reload occurrence at peak/trough when rewriting the TSnCCR0 register in the reload mode (batch rewrite function) (when TSnCMS of the TSnOPT0 register is 0, TSnRTE is 1, TSnRBE is 1, TSnRDE is 0, TSnICE is 1, and TSnIOE is 1 in the TSnOPT1 register).

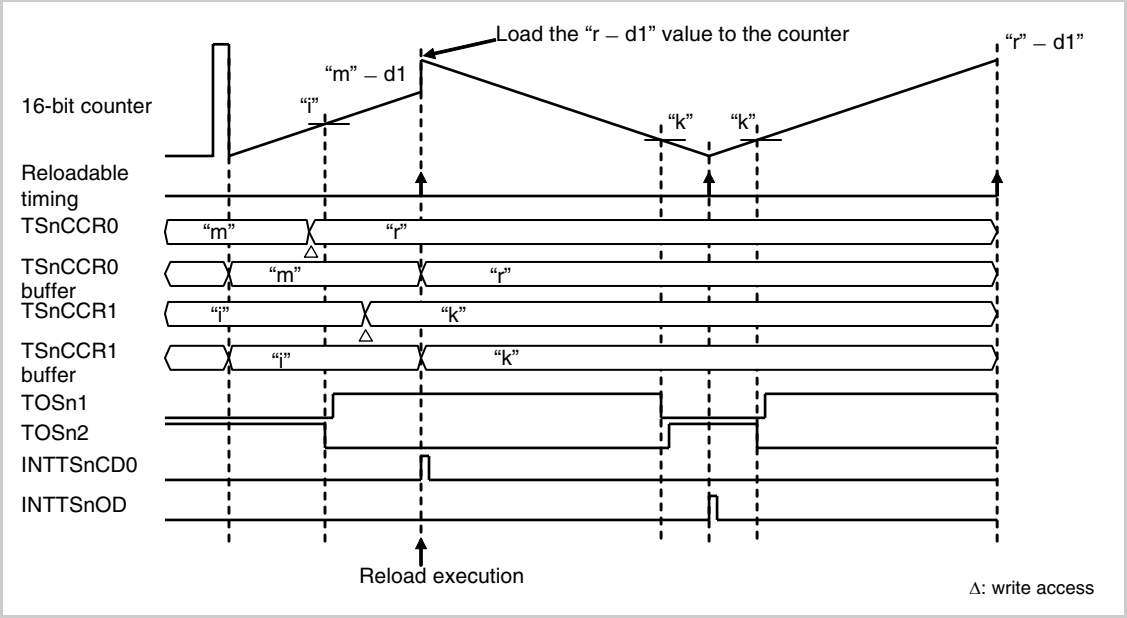


Rewriting in the period of <1> (rewriting during count-up operation)

The next reload timing causes cycle change and asymmetric triangular waveform output on the count-down side because of the peak point reload timing. The duty value must be set again due to cycle change.



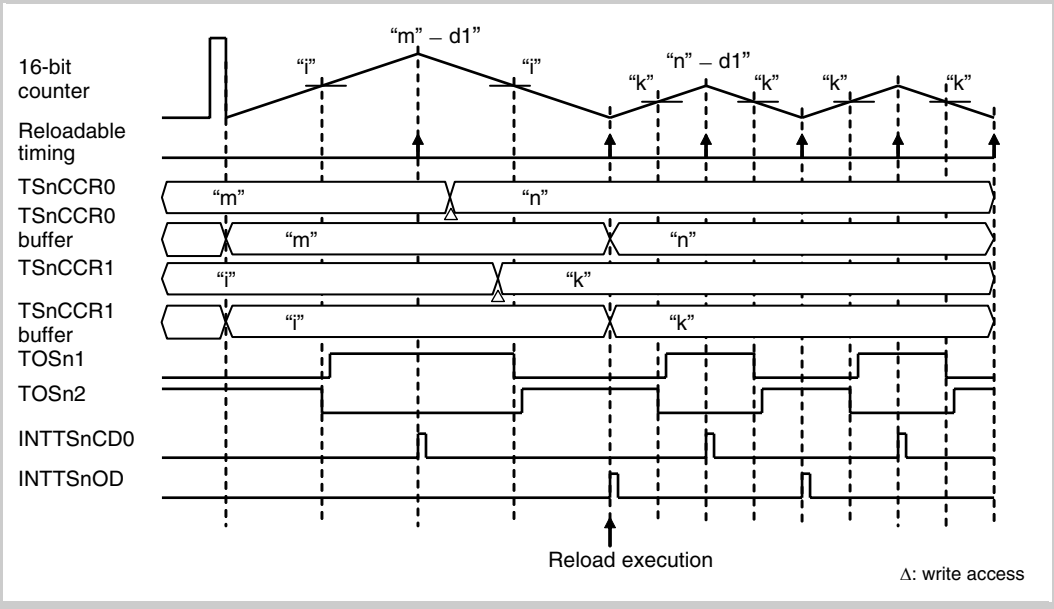
Note d1: TSnDTC1 setting value



Note d1: TSnDTC1 setting value

Rewriting at period <2> (Rewriting during count-down operation)

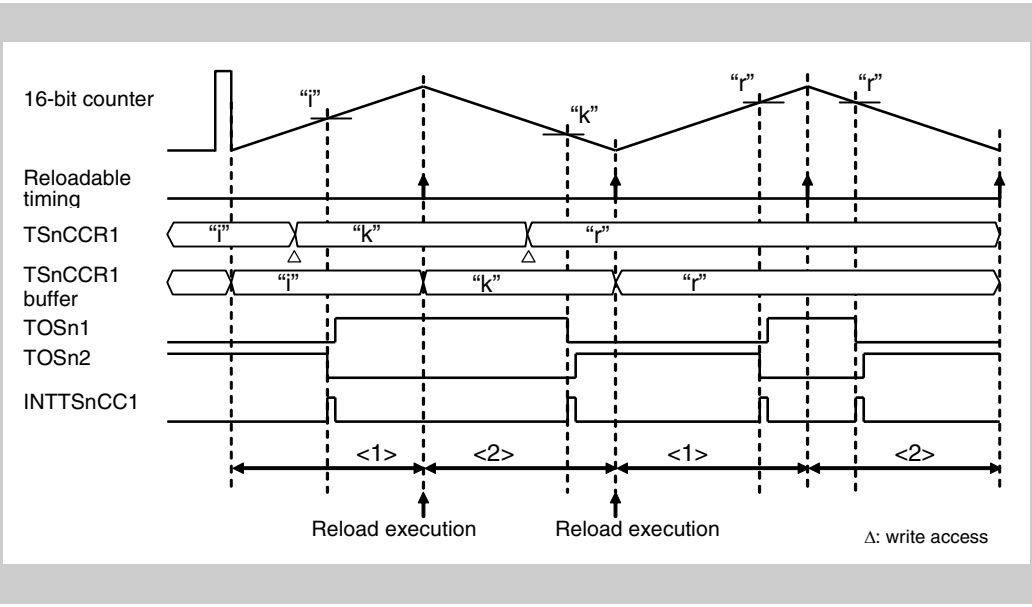
Because the next reload timing is at the trough point, the cycle value changes from the next cycle and the output remains a triangular waveform. The duty value must be set again due to cycle change.



Note d1: TSnDTC1 setting value

Δ: write access

(c) Cautions for TSnCCR1 to TSnCCR3 registers in high-accuracy T-PWM mode



Rewriting in the period <1> (rewriting during count-up operation)

An asymmetric triangular waveform is output because of reload at the peak interrupt timing.

Rewriting in the period <2> (rewriting during count-down operation)

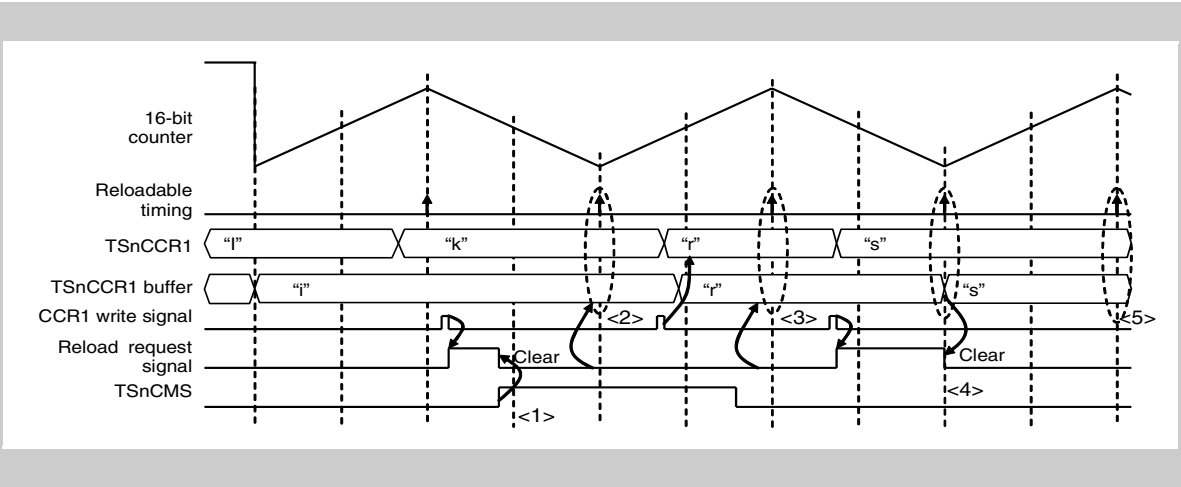
A symmetric triangular waveform is output because of reload at the trough interrupt timing.

(d) Caution on rewriting TSnOPT1

Writing to the TSnOPT1 or TSnOPT1L registers clears the internal interrupt thinning out counter. Then, the interrupt interval may be longer temporarily when the interrupt thinning out function is used.
To avoid this situation, rewriting must be executed by performing a setting such that reloading is done in synchronization with interrupt thinning out (TSnCMS of the TSnOPT0 register is 0, or TSnRDE of the TSnOPT1 register is 1).

(e) Caution on rewriting TSnCMS bit in high-accuracy T-PWM mode

Rewriting the TSnCMS bit of the TSnOPT0 register is possible during operation (TSnCE = 1), but care must be exercised in the TSnCCR1 register writing order.
Writing to the TSnCCR1 register under TSnCMS = 0 enables setting of the reload request signal (internal signal).
Setting of the reload request signal causes reload at the next reload timing, and the reload request signal is then cleared. The reload request signal is cleared by TSnCMS = 1 as well.



<1> Writing to the TSnCCR1 register while TSnCMS = 0 enables setting of the reload request signal (internal signal). The reload request signal is cleared by setting TSnCMS to 1 under the above status.

<2> Reload is not executed because the reload request signal is cleared by TSnCMS = 1.

<3> Writing to the TSnCCR1 register while TSnCMS = 1 does not set the reload request signal. If TSnCMS is set to 0 after that, reload is not executed at the next reload timing.

<4> Writing to the TSnCCR1 register while TSnCMS = 0 sets the reload request signal. Reload is executed at the next reload timing to clear the reload request signal.

<5> Once reload is executed, the reload request signal is cleared. Therefore, reload is not executed at the next reload timing until the TSnCCR1 register is written to.

17.4.3 Output list in each mode

(1) Timer output by each mode

The following list shows the timer outputs (TOSn0 to TOSn7 pins) in each mode.

Table 17-9 Timer Outputs in Each Mode (1/2)

Operation mode	TOSn0	TOSn1	TOSn2	TOSn3
Interval timer mode	Toggle output by TSnCCR0 compare match	Toggle output by TSnCCR1 compare match	Toggle output by TSnCCR2 compare match	Toggle output by TSnCCR3 compare match
External event count mode	Toggle output by TSnCCR0 compare match	Toggle output by TSnCCR1 compare match	Toggle output by TSnCCR2 compare match	Toggle output by TSnCCR3 compare match
External trigger pulse output mode	Toggle output by TSnCCR0 compare match or external trigger input	External trigger pulse waveform output	External trigger pulse waveform output	External trigger pulse waveform output
One-shot pulse mode	Active at count start Inactive by TSnCCR0 match	Active by TSnCCR1 match Inactive by TSnCCR0 match	Active by TSnCCR2 match Inactive by TSnCCR0 match	Active by TSnCCR3 match Inactive by TSnCCR0 match
PWM mode	Toggle output by TSnCCR0 compare match	PWM output by TSnCCR1 compare match	PWM output by TSnCCR2 compare match	PWM output by TSnCCR3 compare match
Free-running mode	Toggle output by TSnCCR0 compare match	Toggle output by TSnCCR1 compare match	Toggle output by TSnCCR2 compare match	Toggle output by TSnCCR3 compare match
Triangular-wave PWM mode	Up/down status output of 16-bit counter	PWM output by TSnCCR1 compare match	PWM output by TSnCCR2 compare match	PWM output by TSnCCR3 compare match
High-accuracy T-PWM mode	Up/down status output of 16-bit counter or 16-bit sub-counter	PWM output by TSnCCR1 compare match (with dead time)	Negative phase output against TOSn1	PWM output by TSnCCR2 compare match (with dead time)
PWM mode with dead time	Toggle output by TSnCCR0 compare match	PWM output by TSnCCR1 compare match (with dead time)	Negative phase output against TOSn1	PWM output by TSnCCR2 compare match (with dead time)
120° excitation mode	Toggle output by TSnCCR0 compare match	PWM output using TSnCCR1, TSnCCR2	PWM output using TSnCCR3, TSnCCR4	PWM output using TSnCCR1, TSnCCR2
Special 120° excitation mode	Toggle output by TSnCCR0 compare match	Pattern output by TSnPAT0	Pattern output by TSnPAT1	Pattern output by TSnPAT0
Special pattern output mode	Pattern output by TSnPAT0[3:0]	Pattern output by TSnPAT0[7:4]	Pattern output by TSnPAT0[11:8]	Pattern output by TSnPAT0[15:12]

Table 17-10 Timer Outputs in Each Mode (2/2)

Operation mode	TOSn4	TOSn5	TOSn6	TOSn7
Interval timer mode	Toggle output by TSnCCR4 compare match	Toggle output by TSnCCR5 compare match	-	-
External event count mode	Toggle output by TSnCCR4 compare match	Toggle output by TSnCCR5 compare match	-	-
External trigger pulse output mode	External trigger pulse waveform output	External trigger pulse waveform output	-	-
One-shot pulse mode	Active by TSnCCR4 match Inactive by TSnCCR0 match	Active by TSnCCR5 match Inactive by TSnCCR0 match	-	-
PWM mode	PWM output by TSnCCR4 compare match	PWM output by TSnCCR5 compare match	-	Pulse output by A/D conversion trigger**Note 1**
Free-running mode	Toggle output by TSnCCR4 compare match	Toggle output by TSnCCR5 compare match	-	-
Triangular-wave PWM mode	PWM output by TSnCCR4 compare match	PWM output by TSnCCR5 compare match	-	Pulse output by A/D conversion trigger**Note 1**
High-accuracy T-PWM mode	Negative phase output against TOSn3	PWM output by TSnCCR3 compare match (with dead time)	Negative phase output against TOSn5	Pulse output by A/D conversion trigger**Note 1**
PWM mode with dead time	Negative phase output against TOSn3	PWM output by TSnCCR3 compare match (with dead time)	Negative phase output against TOSn5	Pulse output by A/D conversion trigger**Note 1**
120° excitation mode	PWM output using TSnCCR3, TSnCCR4	PWM output using TSnCCR1, TSnCCR2	PWM output using TSnCCR3, TSnCCR4	Pulse output by A/D conversion trigger**Note 1**
Special 120° excitation mode	Pattern output by TSnPAT1	Pattern output by TSnPAT0	Pattern output by TSnPAT1	Pulse output by A/D conversion trigger**Note 1**
Special pattern output mode	Pattern output by TSnPAT1[3:0]	Pattern output by TSnPAT1[7:4]	Pattern output by TSnPAT1[11:8]	Pattern output by TSnPAT1[15:12]

Note 1. Refer to *"TOSn7 pin output control" on page 777*.

2. – indicates inactive level output.

(a) TOSn7 pin output control

The A/D conversion trigger signal can be output to the TOSn7 pin. The TOSn7 pin becomes active at the rising edge of the TSnADTRG0 signal, and inactive at the rising edge of the TSnADTRG1 signal. If the TSnADTRG0 signal is detected during the active status of the TOSn7 pin, the TOSn7 pin remains active level. If the TSnADTRG1 signal is detected during the inactive status of the TOSn7 pin, the TOSn7 pin remains inactive level. The inactive status of the TOSn7 pin has priority if the TSnADTRG0 and TSnADTRG1 signal triggers are simultaneously generated.

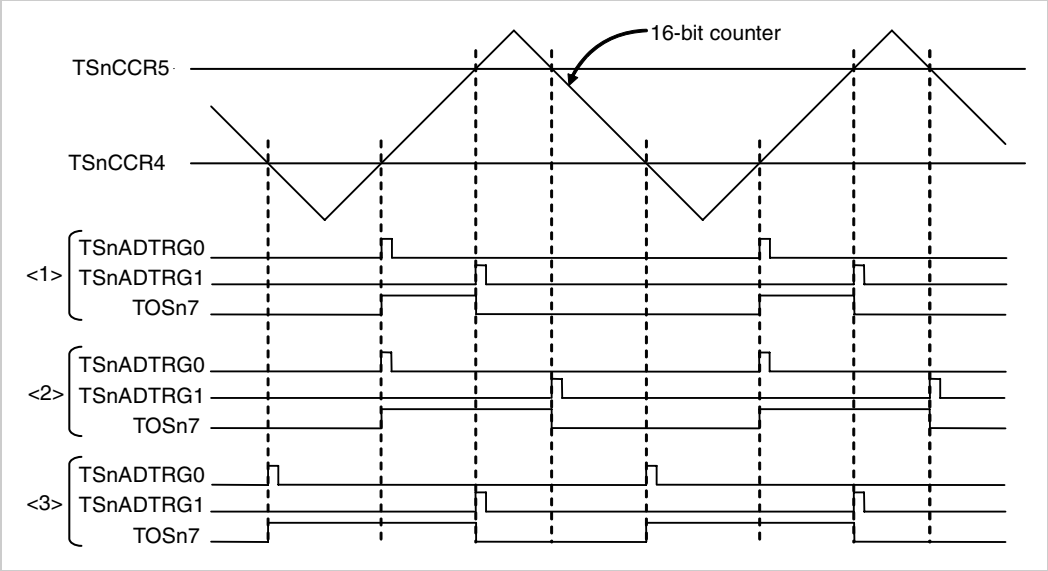


Figure 17-8 TOSn7 pin output timing

Note Case 1: When TSnCCR4 < TSnCCR5, TSnOPT2 = 04H, TSnOPT3 = 10H
Case 2: When TSnCCR4 < TSnCCR5, TSnOPT2 = 04H, TSnOPT3 = 20H
Case 3: When TSnCCR4 < TSnCCR5, TSnOPT2 = 08H, TSnOPT3 = 10H

(2) Interrupt in each mode

The following list shows the interrupts (INTTSnCC0 to INTTSnCC5, INTTSnOV, INTTSnER, INTTSnWN) in each mode.

Table 17-11 Interrupts in Each Mode (1/3)

Operation mode	INTTSnCC0	INTTSnCC1	INTTSnCC2	INTTSnCC3
Interval timer mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
External event count mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
External trigger pulse output mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
One-shot pulse mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
PWM mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
Free-running mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt	TSnCCR2 compare match interrupt	TSnCCR3 compare match interrupt
Triangular-wave PWM mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
High-accuracy T-PWM mode	TSnCCR0 compare match interrupt[Note 1]	TSnCCR1 compare match interrupt[Note 3]	TSnCCR2 compare match interrupt[Note 3]	TSnCCR3 compare match interrupt[Note 3]
PWM mode with dead time	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
120° excitation mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
Special 120° excitation mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]
Special pattern output mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt[Note 2]	TSnCCR2 compare match interrupt[Note 2]	TSnCCR3 compare match interrupt[Note 2]

Note
1. A compare match interrupt occurs only when 000H is set to the TSnDTC1 register. INTTSnCD0 can be used for peak interrupt.
2. Compare match interrupt does not occur when setting TSnCCR0 < TSnCCRm.
3. Compare match interrupt does not occur when setting is performed within the range of 0000H ≤ TSnCCRm < TSnDTC0, (TSnCCR0 − TSnDTC1) < TSnCCRm.
4. m = 1 to 5

Table 17-12 Interrupts in Each Mode (2/3)

Operation mode	INTTSnCC4	INTTSnCC5
Interval timer mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
External event count mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
External trigger pulse output mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
One-shot pulse mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
PWM mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
Free-running mode	TSnCCR4 compare match interrupt	TSnCCR5 compare match interrupt
Triangular-wave PWM mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
High-accuracy T-PWM mode	TSnCCR4 compare match interrupt[Note 2]	TSnCCR5 compare match interrupt[Note 2]
PWM mode with dead time	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
120° excitation mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
Special 120° excitation mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]
Special pattern output mode	TSnCCR4 compare match interrupt[Note 1]	TSnCCR5 compare match interrupt[Note 1]

Note 1. Compare match interrupt does not occur when setting TSnCCR0 < TSnCCRm.

2. Compare match interrupt is not generated when setting within the range of 0000H ≤ TSnCCRm < TSnDTC0, (TSnCCR0 − TSnDTC1) < TSnCCRm.

3. m = 1 to 5

Table 17-13 Interrupts in Each Mode (3/3)

Operation mode	INTTSnOV	INTTSnER	INTTSnWN
Interval timer mode	-	-	-
External event count mode	-	-	-
External trigger pulse output mode	-	-	-
One-shot pulse mode	-	-	-
PWM mode	-	Error interrupt	-
Free-running mode	Overflow interrupt	-	-
Triangular-wave PWM mode	-	Error interrupt	-
High-accuracy T-PWM mode	Overflow interrupt Note 1	Error interrupt	Warning interrupt Note 2
PWM mode with dead time	-	Error interrupt	-
120° excitation mode	-	Error interrupt	Warning interrupt
Special 120° excitation mode	-	Error interrupt	Warning interrupt
Special pattern output mode	-	Error interrupt	-

Note 1. An overflow interrupt (INTTSnOV) occurs when setting TSnCCR0, TSnDTC0, and TSnDTC1 incorrectly.

2. Only when the 180° excitation control function is used

(3) A/D conversion trigger, peak interrupt, trough interrupt in each mode

The A/D conversion triggers and peak/trough interrupts in each mode are listed below.

Table 17-14 A/D Conversion Trigger, Peak Interrupt and Trough Interrupt

Operation mode	TSnADTRG0	TSnADTRG1	INTTSnCD0	INTTSnOD
Interval timer mode	-	-	-	-
External event count mode	-	-	-	-
External trigger pulse output mode	-	-	-	-
One-shot pulse mode	-	-	-	-
PWM mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
Free-running mode	-	-	-	-
Triangular-wave PWM mode	Select from INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	-	Trough interrupt
High-accuracy T-PWM mode	Select from INTTSnCD0, INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnCD0, INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Peak interrupt	Trough interrupt
PWM mode with dead time	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
120° excitation mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
Special 120° excitation mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
Special pattern output mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts**Note 1**	Peak interrupt at the same timing with INTTSnCC0 interrupt	-

Note 1. The TSnCCR4 and TSnCCR5 registers execute output control of the TOSn4 and TOSn5 pins (TOSn2, TOSn4, and TOSn6 pins in the 120° excitation mode) as well as determine the INTTSnCC4 and INTTSnCC5 timings. Therefore, the output timings of the TOSn4 and TOSn5 pins must be considered when using INTTSnCC4 and INTTSnCC5.

2. INTTSnCD0 interrupt and INTTSnOD interrupt are the generation condition after interrupt thinning out.

17.5 Match Interrupt

Interrupt types include compare match interrupts (INTTSnCC0 to INTTSnCC5), peak interrupt (INTTSnCD0) and trough interrupt (INTTSnOD). Refer to *"Error/warning Interrupt" on page 823* for error/warning interrupts (INTTSNER, INTTSnWN).
Compare match interrupts (INTTSnCC0 to INTTSnCC5) occur regardless of the operation mode.
The peak interrupt (INTTSnCD0) occurs in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. This interrupt is output at the switch timing from count-up operation to count-down operation of the 16-bit counter in the high-accuracy T-PWM mode. In the modes with 16-bit counter sawtooth wave operation (PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode), the peak interrupt occurs after match of the 16-bit counter and the TSnCCR0 register values (at the same timing as an INTTSnCC0 interrupt).
The trough interrupt occurs at the switch timing from count-down operation to count-up operation of the 16-bit counter in the triangular-wave PWM output mode and high-accuracy T-PWM mode.

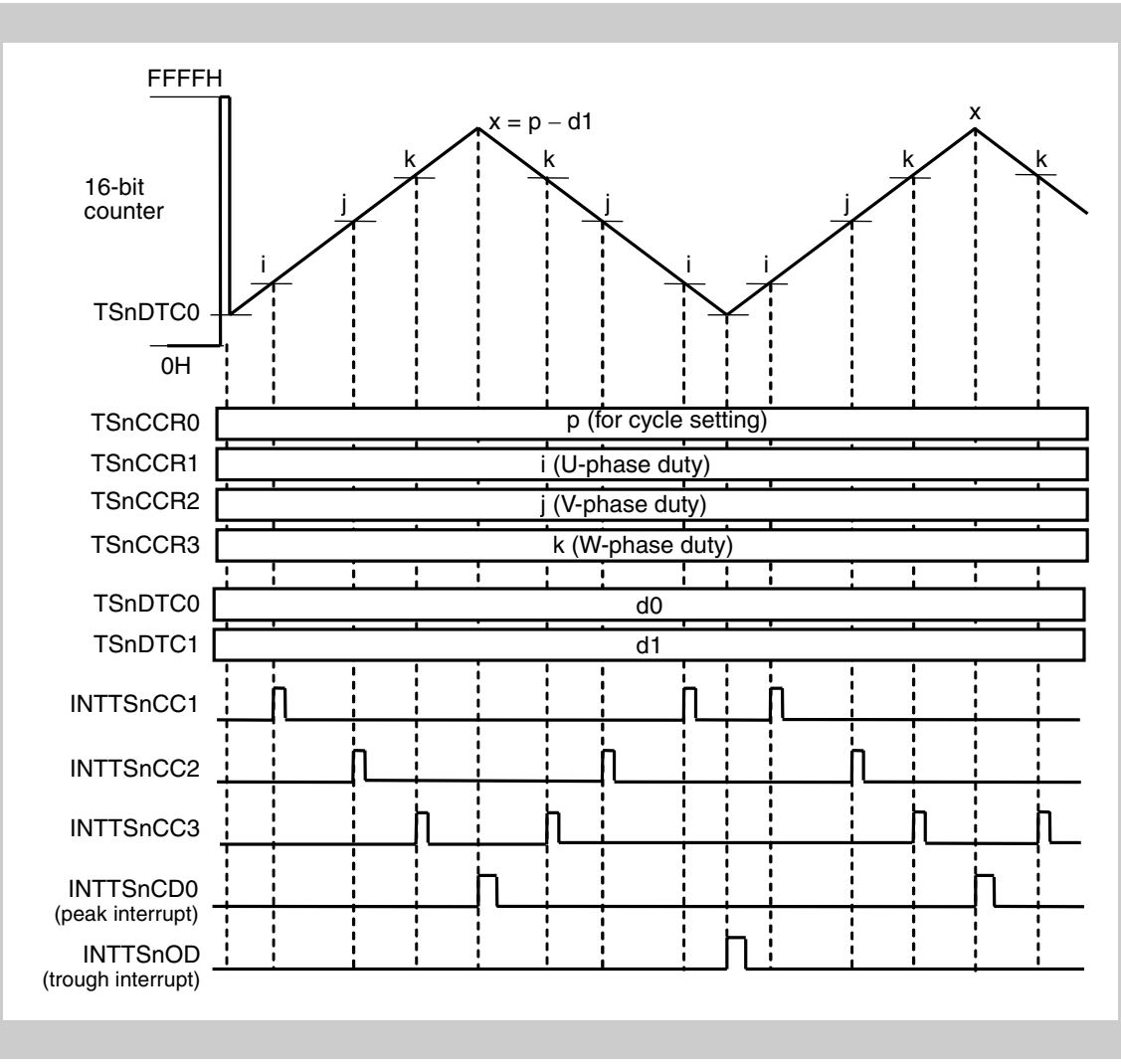


Figure 17-9 Interrupt signal output example (1/2) (high-accuracy T-PWM mode)

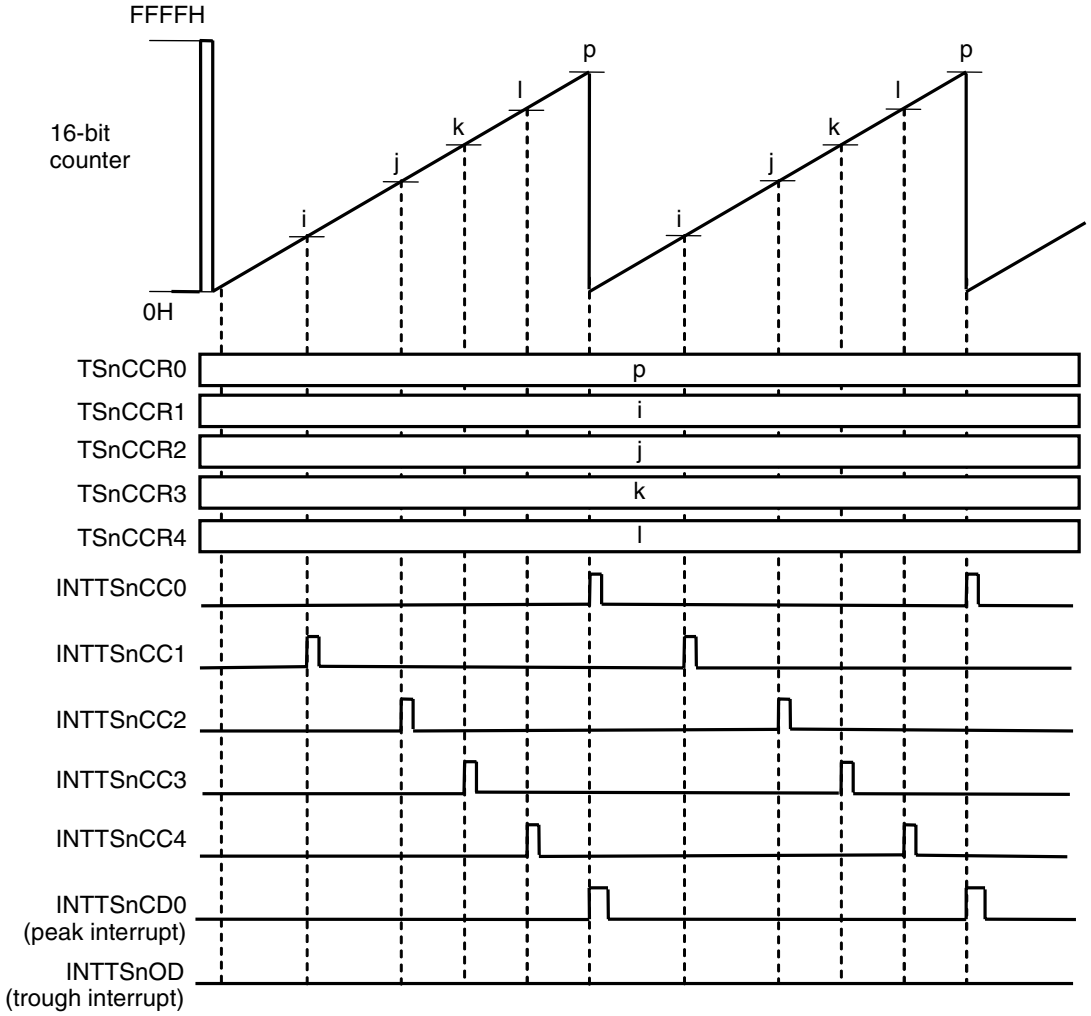


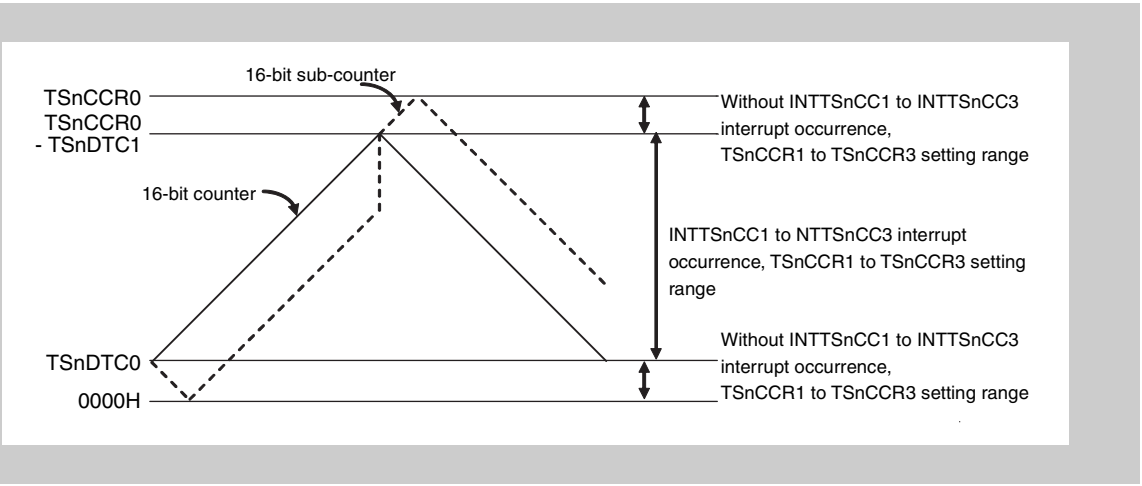
Figure 17-10 Interrupt signal output example (2/2) (example in PWM mode)

17.5.1 Caution on compare match interrupt

(1) Caution on high-accuracy T-PWM mode

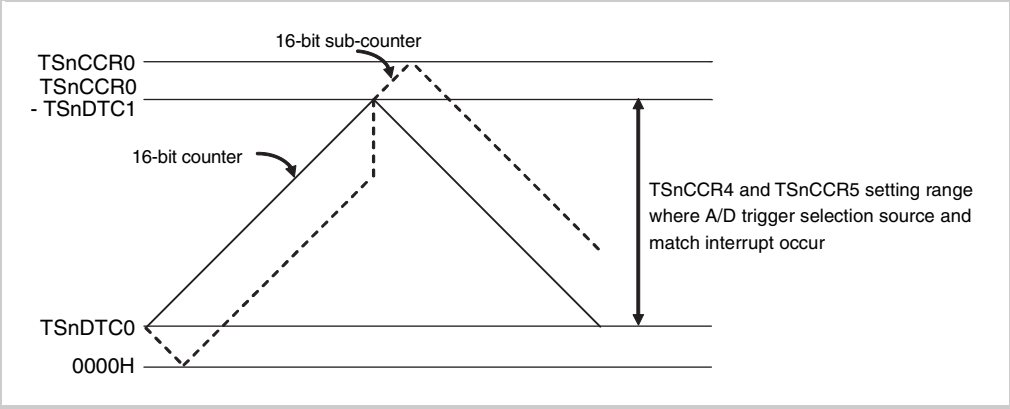
A compare match interrupt occurs upon a match between the 16-bit counter value and compare register (TSnCCR0 to TSnCCR5) values. Compare register setting beyond the 16-bit counter's count operation range is possible in the high-accuracy T-PWM mode. Therefore, compare match interrupts do not occur under the following conditions.

- Restriction concerning compare match interrupt of TSnCCR0 and the INTTSnCC0 register
  In the high-accuracy T-PWM mode, compare match interrupts (INTTSnCC0) does not occur while TSnDTC1 ≠ 0000H.
  Use INTTSnOD (trough interrupt)/INTTSnCD0 (peak interrupt) for the interrupt as a cycle.
- Restriction concerning the TSnCCR1 to TSnCCR3 registers
  A compare match interrupt of match between the compare value and the 16-bit counter value is not generated in the high-accuracy T-PWM mode if set within the following range.
  0000H ≤ TSnCCR1 to TSnCCR3 < TSnDTC0,
  TSnCCR0 to TSnDTC1 + 1 < TSnCCR1 to TSnCCR3 ≤ TSnCCR0



- Restriction concerning the TSnCCR4 and TSnCCR5 registers
  A compare match interrupt does not occur since a match between compare value and the 16-bit counter value is not generated in the high-accuracy T-PWM mode if set within the following range.
  0000H ≤ TSnCCR4, TSnCCR5 < TSnDTC0,
  TSnCCR0 to TSnDTC1 + 1 < TSnCCR4, TSnCCR5 ≤ TSnCCR0

The setting must be within the range of TSnDTC0 ≤ TSnCCR4, TSnCCR5 ≤ TSnCCR0 to TSnDTC1 because the TSnCCR4 and TSnCCR5 registers are used as a trigger source for A/D conversion triggers.

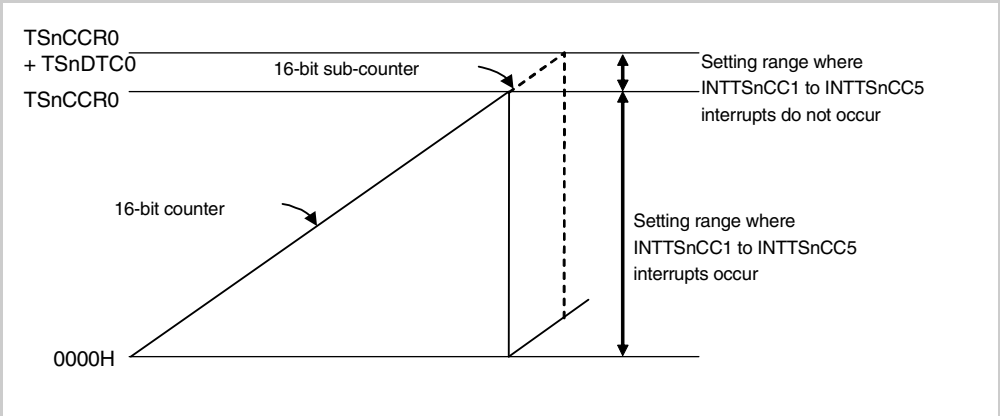


(2) Caution on PWM mode with dead time

A compare match interrupt occurs upon a match of the 16-bit counter and compare register (TSnCCR0 to TSnCCR5). Compare register setting beyond the count operation range of the 16-bit counter is possible in the T-PWM mode with dead time. Therefore, compare match interrupts do not occur under the following conditions.

- Restrictions concerning the TSnCCR1 to TSnCCR5 registers
  A compare match interrupt does not occur if set within the following range in the PWM mode with dead time because no match occurs between the compare value and the 16-bit counter value.

  TSnCCR0 < TSnCCR1 to TSnCCR5 ≤ TSnCCR0 + TSnDTC0
  The setting must be within the range of TSnCCR4, TSnCCR5 ≤ TSnCCR0 when using the TSnCCR4 and TSnCCR5 registers because the TSnCCR4 and TSnCCR5 registers are used as trigger sources for A/D conversion triggers.



17.6 Flags

Table 17-15 Flag list

No.	Flag name	Symbol	Register	Operation mode
(1)	Count-up flag	TSnCUF	TSnOPT0	Triangular-wave PWM mode, high-accuracy T-PWM mode
		TSnSUF	TSnOPT6	High-accuracy T-PWM mode
(2)	Positive/negative simultaneous active detection flag	TSnTBF	TSnOPT6	PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode
(3)	Reload request flag	TSnRSF	TSnOPT6	External trigger pulse output mode, PWM mode, high-accuracy T-PWM mode**Note 1**, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode
(4)	Noise detection flag	TSnNDF	TSnOPT6	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(5)	Pattern order detection flag	TSnTSF	TSnOPT5	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(6)	Pattern error detection flag	TSnPEF	TSnOPT6	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(7)	Pattern reverse detection flag	TSnPRF	TSnOPT6	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(8)	TAPTSn2 to TAPTSn0 pins abnormal toggle detection flag	TSnPTF	TSnOPT6	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(9)	TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag	TSnTDF	TSnOPT6	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(10)	Pattern phase difference detection flag	TSnPPF	TSnOPT6	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode
(11)	Overflow flag	TSnOVF	TSnOPT0	Free-running mode, and high-accuracy T-PWM mode**Note 3**
(12)	Timer output pattern flag	TSnOPF0 to TSnOPF2	TSnOPT5	High-accuracy T-PWM mode**Note 4**, PWM mode with dead time**Note 4**, 120° excitation mode, and special 120° excitation mode
(13)	Pattern switch detection flag (internal signal)	TSnESG	-	High-accuracy T-PWM mode**Note 2**, PWM mode with dead time**Note 2**, 120° excitation mode, and special 120° excitation mode

Note 1. Only in reload mode (when TSnCMS of the TSnOPT0 register is 0)

2. Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

3. Selectable only when setting conditions for the TSnDTC0 and TSnDTC1 registers are incorrectly set.

4. Selectable only when the 180° excitation function (when TSnADC of the TSnOPT5 register is 1) is used.

17.6.1 Count-up flags (TSnCUF, TSnSUF)

[Name]

Count-up flag (TSnCUF) - TSnOPT0 register
(TSnSUF) - TSnOPT6 register

[Description]

There are two kinds of count-up flags as follows.
TSnCUF is a count-up/down flag of 16-bit counter.
TSnSUF is a count-up/down flag of 16-bit sub-counter.
For both TSnCUF and TSnSUF, 0 indicates count-up operation and 1 indicates count-down operation.

[Example]

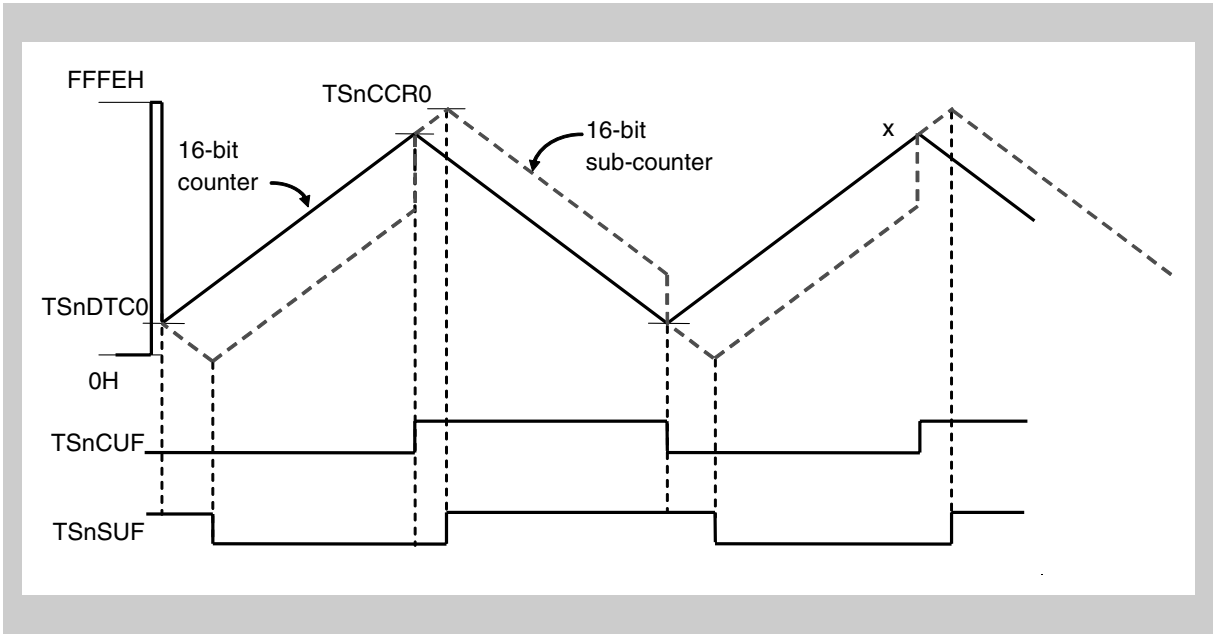


Figure 17-11 Count-up flag operation example

The TSnCUF value is as follows.

[In triangular-wave PWM mode]

0 ≤ 16-bit counter < TSnCCR0 + 1 ..... 0 (count-up)
TSnCCR0 + 1 ≥ 16-bit counter >0 ..... 1 (count-down)

[In high-accuracy T-PWM mode]

TSnDTC0 + 2 ≤ 16-bit counter < (TSnCCR0 - TSnDTC1) .... 0 (count-up)
(TSnCCR0 – TSnDTC1 - 2) ≥ 16-bit counter > TSnDTC0 .... 1 (count-down)

The TSnSUF value is as follows.

[In high-accuracy T-PWM mode]

0 ≤ 16-bit sub-counter < TSnCCR0 ..... 0 (count-up)
TSnCCR0 ≥ 16-bit sub-counter > 0 ..... 1 (count-down)

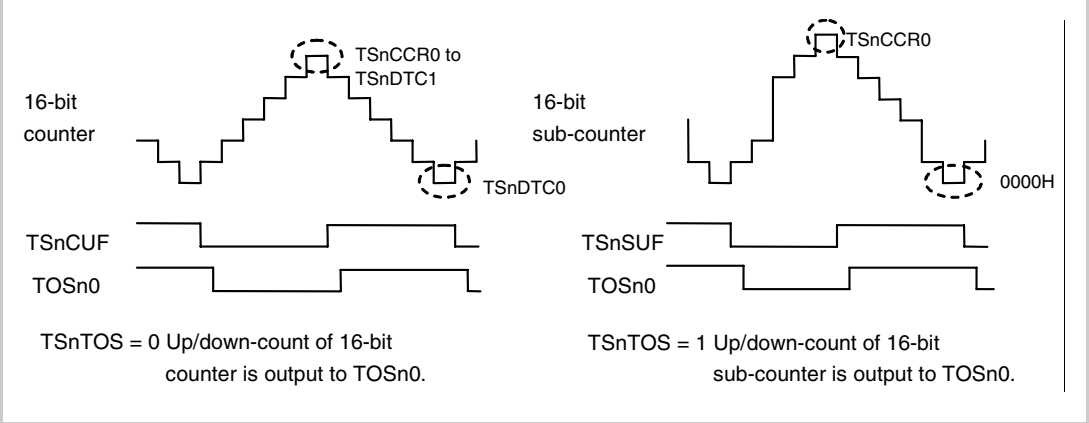


Figure 17-12 TOSn0 Output at switch of TSnTOS bit of TSnOPT7 register

[Operation mode]

TSnCUF Triangular-wave PWM mode, high-accuracy T-PWM mode
TSnSUF High-accuracy T-PWM mode

17.6.2 Positive/negative-phase simultaneous active detection flag (TSnTBF)

[Name]

Positive/negative phase simultaneous active detection flag (TSnTBF) - TSnOPT6 register

[Description]

Positive/negative phase simultaneous active of timer Sn can be detected by TSnTBF.

[Example]

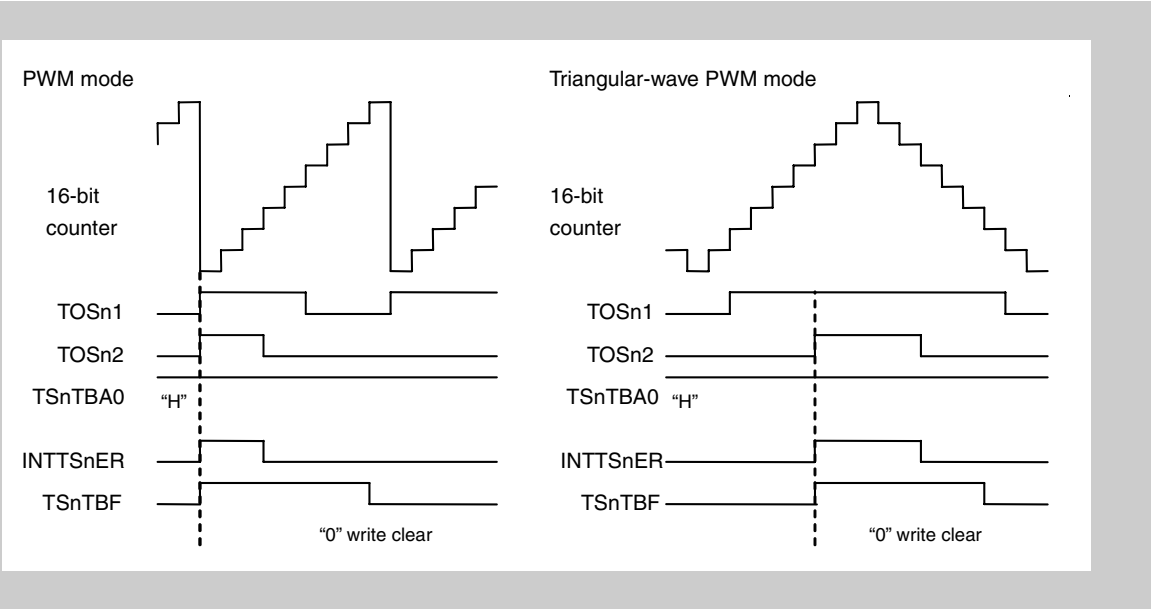


Figure 17-13 Positive/negative phase simultaneous active detection flag operation example

[Operation mode]

PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode

17.6.3 Reload request flag

[Name]

Reload request flag (TSnRSF) - TSnOPT6 register

[Description]

TSnRSF is set to 1 when a reload request is generated (when the TSnCCR1 register is written). The reload request flag is cleared to 0 when reload is generated and the value is transferred to all the buffer registers.

[Example]

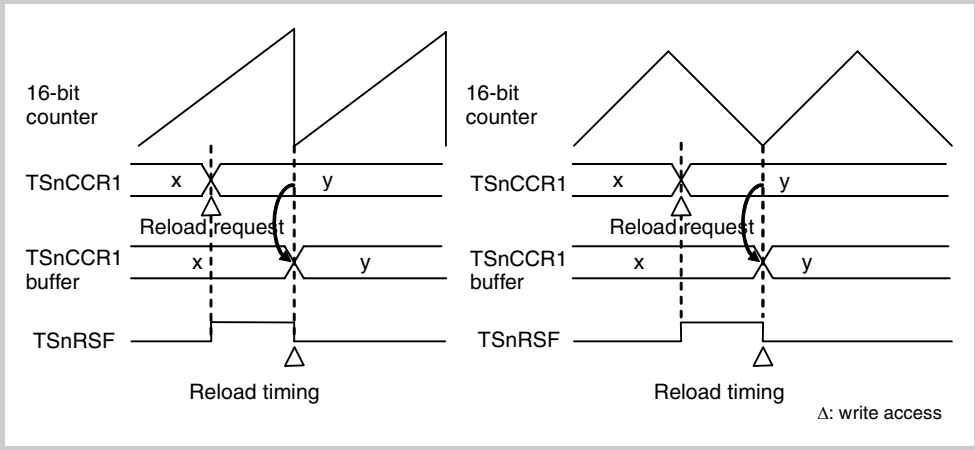


Figure 17-14 Reload request flag operation example

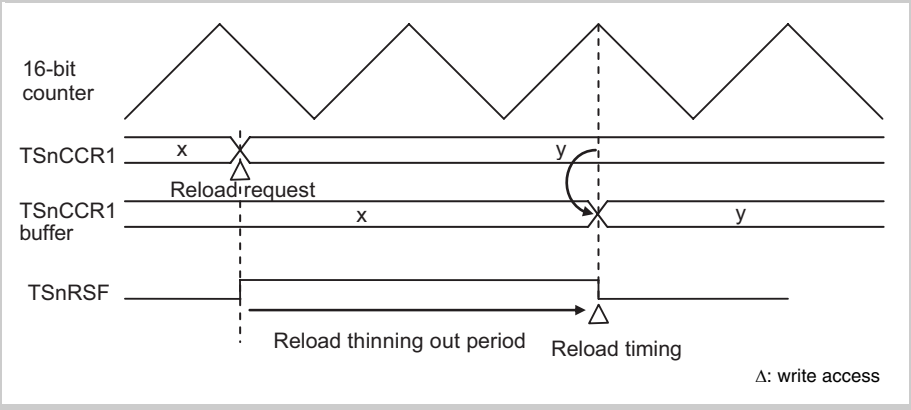


Figure 17-15 Reload request flag and reload thinning out period

[Operation mode]

External trigger pulse output mode, PWM mode, high-accuracy T-PWM mode[Note], triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode

Note Only in reload mode (only when TSnCMS of the TSnOPT0 register is 0)

17.6.4 Noise detection flag (TSnNDF)

[Name]

Noise detection flag (TSnNDF) - TSnOPT6 register

[Description]

TSnNDF can detect noise generation on the TAPTSn2 to TAPTSn0 pins. “Noise” means simultaneous change on two or more pins among the TAPTSn2 to TAPTSn0 pins.

The noise detection flag (TSnNDF) is set to 1 when simultaneous change is detected on two or more pins among TAPTSn2 to TAPTSn0 pins. Clear to 0 is executed when 0 is written to the TSnNDF flag, or TSnCE of the TSnCTL0 register is set to 0.

[Example]

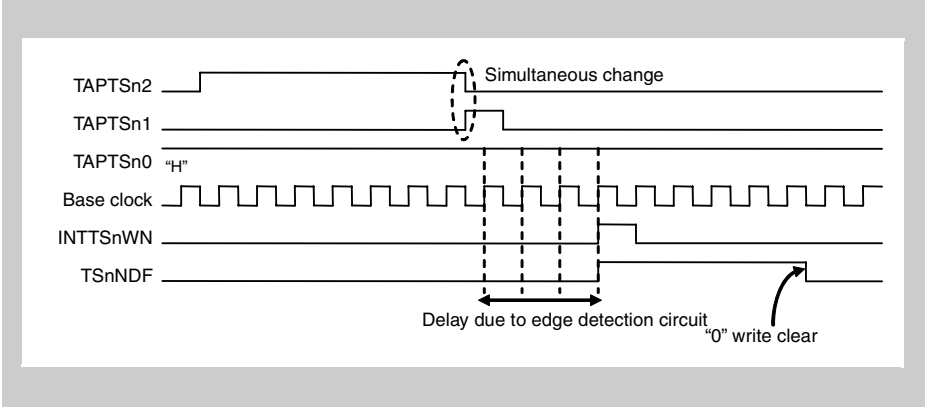


Figure 17-16 Noise detection flag operation example

[Operation mode]

High-accuracy T-PWM mode[Note] and PWM mode with dead time[Note]

Note Selectable only when the software output control function (when TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (when TSnADC of the TSnOPT5 register is 1) is used.

17.6.5 Pattern order detection flag (TSnTSF)

[Name]

TSnTSF detects the order of the pattern input to the TAPTSn2 to TAPTSn0 pins.

Pattern order detection flag (TSnTSF) - TSnOPT5 register

[Description]

The inputs the TAPTSn2 to TAPTSn0 pins result in the following operation.

- 0 is set when input to the TAPTSn2 to TAPTSn0 pins changes as follows.
  TAPTSn2 to TAPTSn0 =
  [1, 0, 1] → [1, 0, 0] → [1, 1, 0] → [0, 1, 0] → [0, 1, 1] → [0, 0, 1]
- 1 is set when input to TAPTSn2 to TAPTSn0 pins changes as follows.
  TAPTSn2 to TAPTSn0 =
  [1, 0, 1] ← [1, 0, 0] ← [1, 1, 0] ← [0, 1, 0] ←[0, 1, 1] ← [0, 0, 1]

[Example]

– When normal input is provided to the TAPTSn2 to TAPTSn0 pins
  As shown in *Figure 17-17 on page 793*, when the TAPTSn2 to TAPTSn0 pins are change in sequence, 0 or 1 is set according to the change order at the change timing.

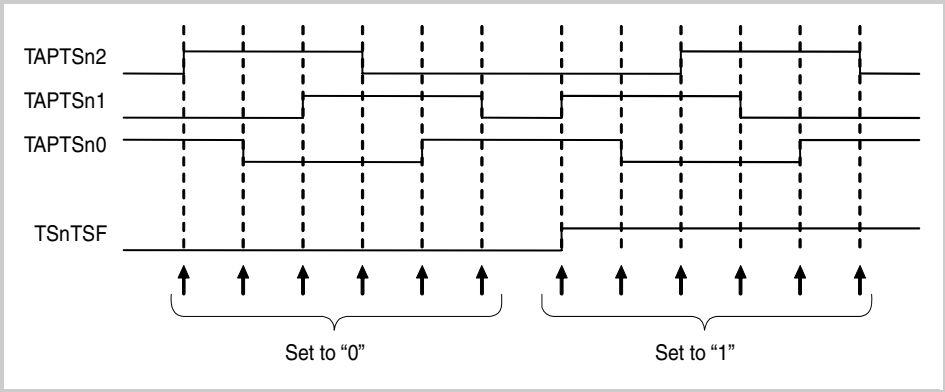


Figure 17-17 Pattern order detection flag operation example (in normal case)

- In exceptions occurrence

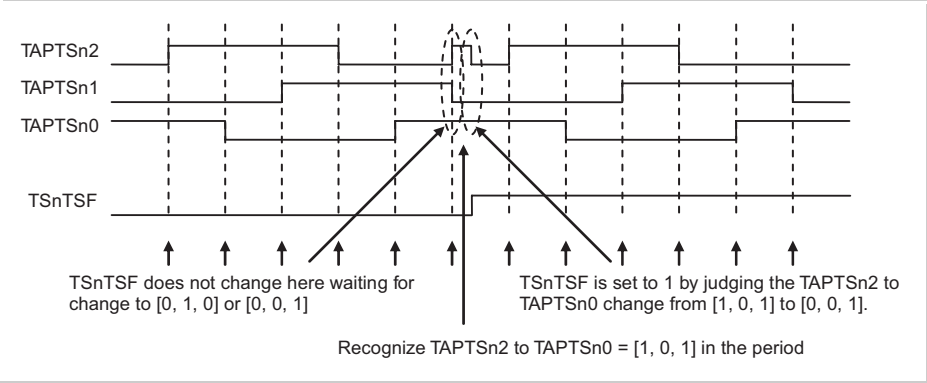


Figure 17-18 When inputs to pins TAPTSn2 to TAPTSn0 are changed by 2 bits

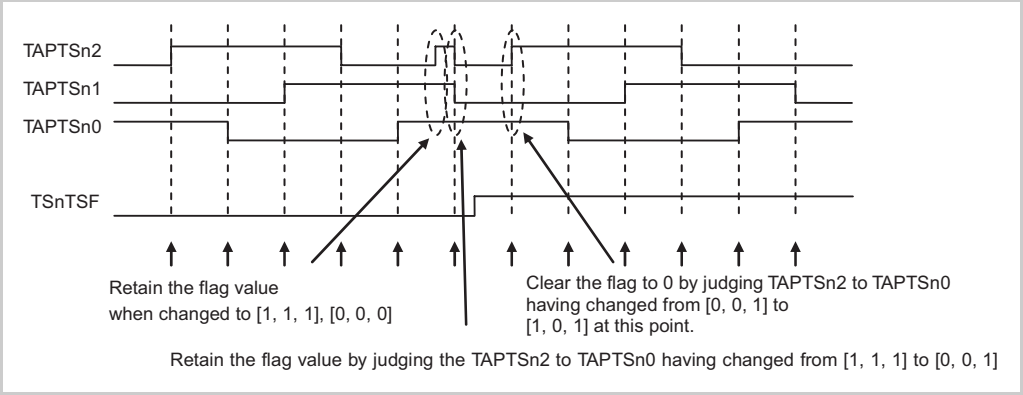


Figure 17-19 When inputs to pins TAPTSn2 to TAPTSn0 are changed to [0, 0, 0] or [1, 1, 1].

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode, and special 120° excitation mode

Note Selectable only when the software output control function (TSnSOC of the TSnOPT4 register = 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

17.6.6 Pattern error detection flag (TSnPEF)

[Name]

Pattern error detection flag (TSnPEF) - TSnOPT6 register

[Description]

TSnPEF can detect [0, 0, 0] or [1, 1, 1] input to the TAPTSn2 to TAPTSn0 pins.
TSnPEF is set to 1 when the pin level of the TAPTSn2 to TAPTSn0 pins is [0, 0, 0] or [1, 1, 1] level, and cleared when 0 is written to the TSnPEF flag, or while timer Sn operation is stopped (TSnCE of the TSnCTL0 register is set to 0).

[Example]

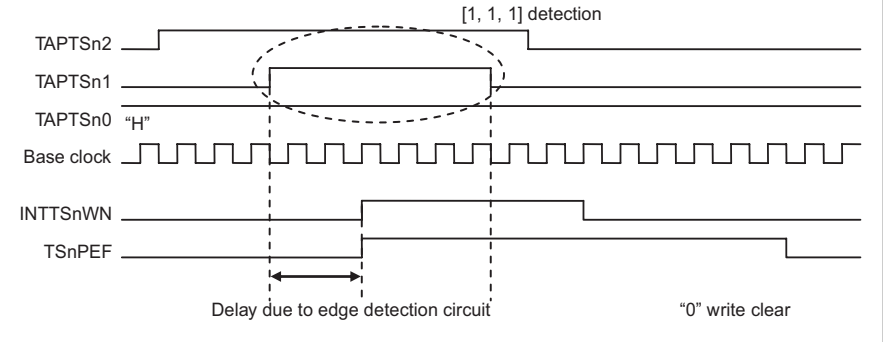


Figure 17-20 Pattern error detection flag operation example (TAPTSn2 to TAPTSn0 = [1, 1, 1])

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode and special 120° excitation mode

Note Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

17.6.7 Pattern reverse detection flag (TSnPRF)

[Name]

Pattern reverse detection flag (TSnPRF) - TSnOPT6 register

[Description]

TSnPRF can detect reversal of change order of the TAPTSn2 to TAPTSn0 pins.
TSnPRF switches at the change timing of pattern order detection flag (TSnTSF). The TSnPRF flag is valid with the 2nd or later change of the TAPTSn2 to TAPTSn0 pins immediately after TSnCE of the TSnCTL0 register is set to 1.

[Example]

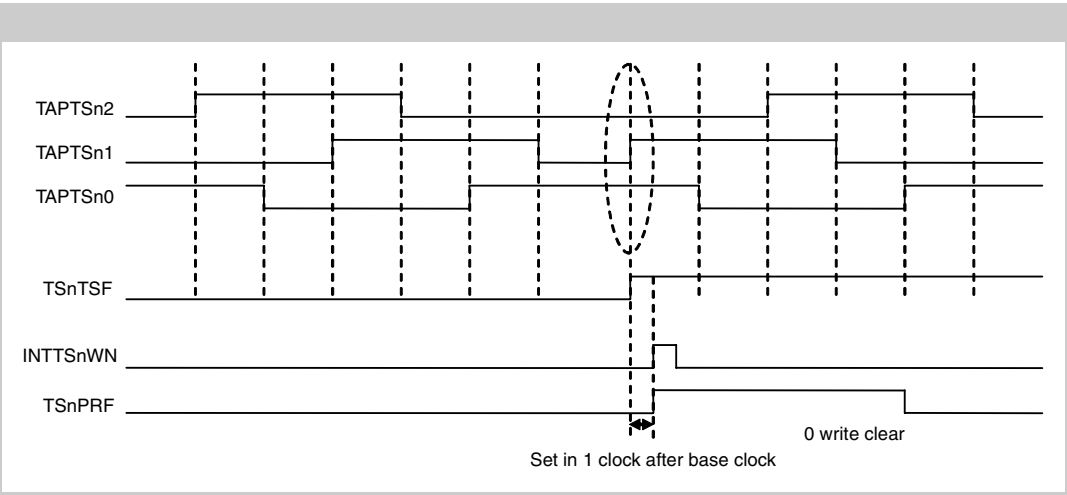


Figure 17-21 Pattern reverse rotation detection flag operation example

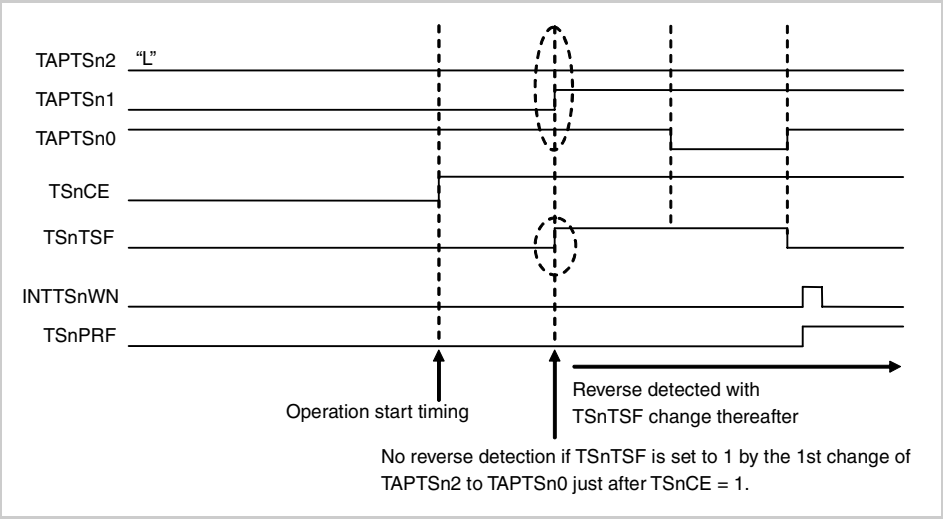


Figure 17-22 Operation example immediately after TSnCE of TSnCTL0 register is set to 1

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode, and special 120° excitation mode

Note Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

17.6.8 TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF)

[Name]

TAPTSn2 to TAPTSn0 pins abnormal toggle detection flag (TSnPTF) - TSnOPT6 register

[Description]

TSnPTF can detect that three or more changes of the TAPTSn2 to TAPTSn0 pins are input between the TSnSTCI0 signal triggers, or between the TSnSTCI1 signal triggers. If the third trigger of the TSnSTCI0 and TSnSTCI1 signals and change of the TAPTSn2 to TAPTSn0 pins are generated simultaneously, the TSnPTF flag is set to 1 and a warning interrupt (INTTSnWN) occurs.

[Example]

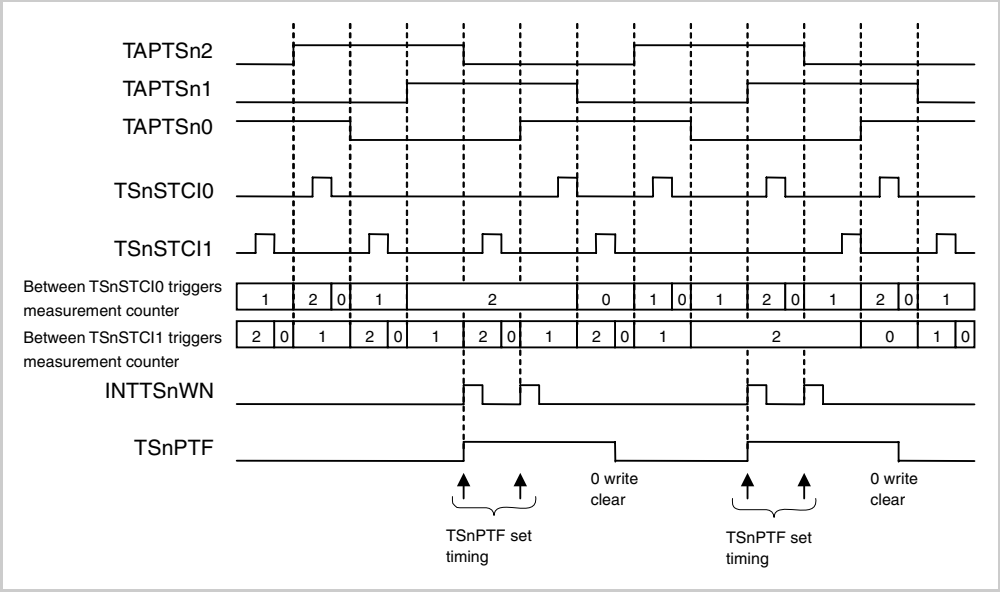


Figure 17-23 TAPTSn2 to TAPTSn0 pins abnormal toggle detection flag operation example

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode and special 120° excitation mode

Note Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

Caution

1. The TSnPTF flag is valid only when TSnPOT of the TSnOPT5 register = 1, TSnSOC of the TSnOPT4 register = 0, and TSnPTC1 of the TSnOPT7 register = 1.
2. When TSnPOT = 1, TSnPTC0 = 1 and TSnPTC1 = 1, abnormal toggle detection leads automatic switch to the pattern switch system (TSnPOT = 0) by output switch control of TOSn1 to TOSn6 pins.

Note If the third trigger of the TSnSTCI0 and TSnSTCI1 signals and change of the TAPTSn2 to TAPTSn0 pins are generated simultaneously, a warning interrupt (INTTSnWN) occurs and the TSnPTF flag is set.

17.6.9 TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF)

[Name]

TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) - TSnOPT6 register

[Description]

TSnTDF detects that the TSnSTCI0 and TSnSTCI1 signals become triggers simultaneously.

[Example]

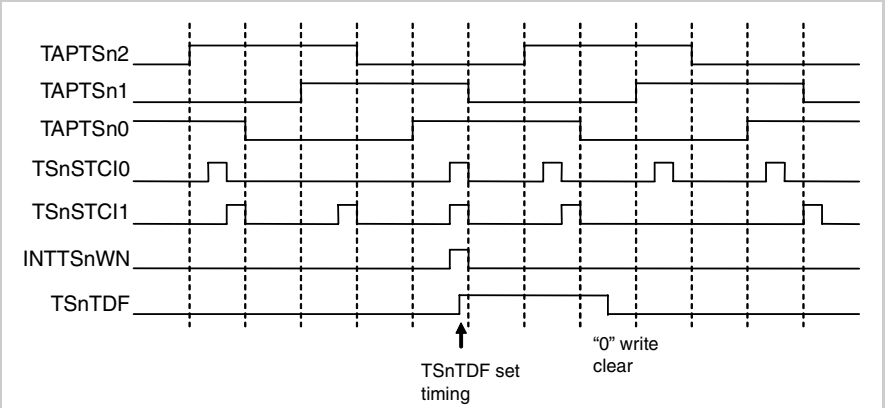


Figure 17-24 TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag operation example

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode and special 120° excitation mode

Note Selectable only when the software output control function (TSnSOC of the TSnOPT4 register = 1) or 180° excitation function (TSnADC of the TSnOPT5 register = 1) is used

Caution The TSnTDF flag is valid only when TSnPOT of the TSnOPT5 register is 1, TSnSOC of the TSnOPT4 register is 0, and TSnPTC1 of the TSnOPT7 register is 1.

17.6.10 Pattern phase difference detection flag (TSnPPF)

[Name]

Pattern phase difference detection flag (TSnPPF) - TSnOPT6 register

[Description]

TSnPPF can detect the phase difference between input pattern (TAPTSn2 to TAPTSn0 pins) and output pattern (TSnOPF2 to TSnOPF0 bits of TSnOPT5 register) when the semi-automatic driving system is used. The pattern phase difference is detected when the TSnSTCI0 or TSnSTCI1 signal is input, and the TSnPPF bit is then set. The TSnPPF bit remains 1 until it is cleared by software. When a phase difference is detected, the TSnPPF bit is set for each base clock (f_{TMSn}) of timer Sn. Clear the TSnPPF bit (0) at a timing at which no phase differences may occur.

Table 17-16 Normal input/output pattern

TAPTSn2 to TAPTSn0 (Input)	"1, 0, 1"	"1, 0, 0"	"1, 1, 0"	"0, 1, 0"	"0, 1, 1"	"0, 0, 1"
TSnOPF2 to TSnOPF0 (Output)	"0, 0, 1" "1, 0, 1" "1, 0, 0"	"1, 0, 1" "1, 0, 0" "1, 1, 0"	"1, 0, 0" "1, 1, 0" "0, 1, 0"	"1, 1, 0" "0, 1, 0" "0, 1, 1"	"0, 1, 0" "0, 1, 1" "0, 0, 1"	"0, 1, 1" "0, 0, 1" "1, 0, 1"

[Example]

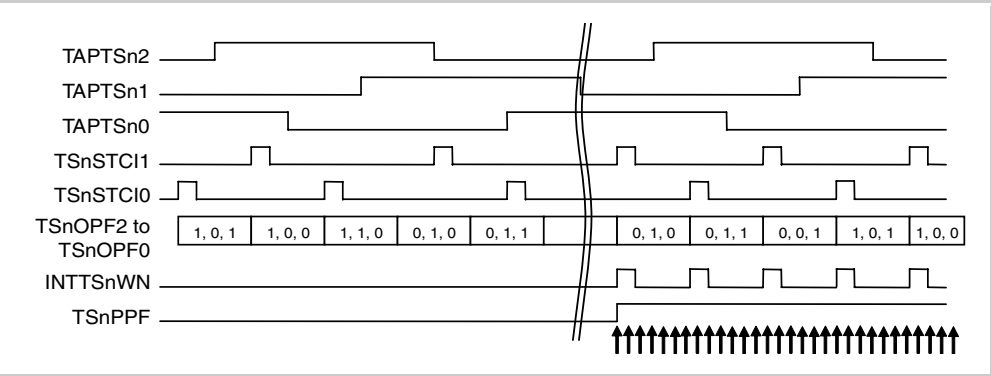


Figure 17-25 Pattern phase difference detection flag operation example

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode and special 120° excitation mode

Note Selectable only when the software output control function (TSnSOC of TSnOPT4 register is 1) or 180° excitation function (TSnADC of TSnOPT5 register is 1) is used

Caution The TSnPPF flag is valid only when TSnPOT of the TSnOPT5 register is 1, TSnSOC of the TSnOPT4 register is 0, and TSnPPC of the TSnOPT7 register is 1.

17.6.11 Overflow flag (TSnOVF)

[Name]

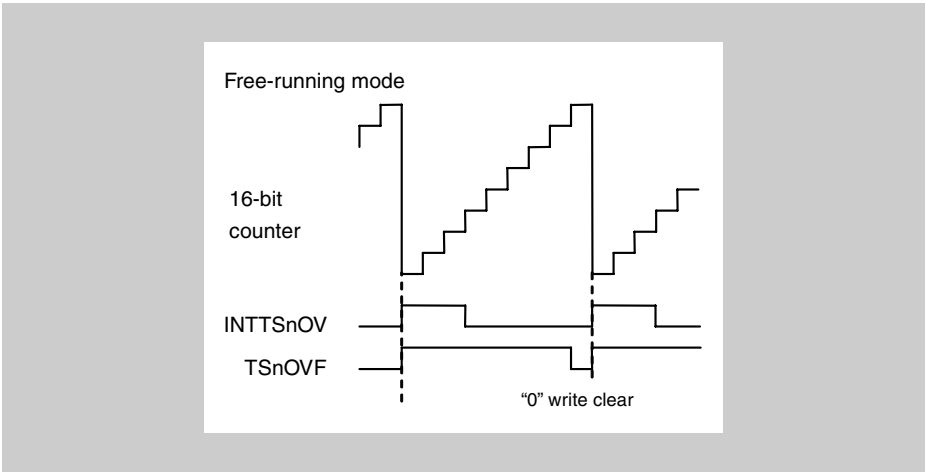
Overflow flag (TSnOVF) - TSnOPT0 register

[Description]

TSnOVF is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H.
This flag is cleared to 0 by writing 0 to it or the operation is stopped (TSnCE of the TSnCTL0 register = 0). If TSnOVF is set, an overflow interrupt (INTTSnOV) is generated simultaneously.

[Example]

Figure 17-26 Overflow flag operation example



[Operation mode]

Free-running mode and high-accuracy T-PWM mode[Note]

Note Only when the TSnDTC0, TSnDTC1 register values are incorrectly set

17.6.12 Timer output pattern flags (TSnOPF2 to TSnOPF0)

[Name]

Timer output pattern flags (TSnOPF2 to TSnOPF0) - TSnOPT5 register

[Description]

TSnOPF2 to TSnOPF0 are flags that indicate timer output patterns.
Refer to *"Timer output pattern in each mode and function" on page 739* for details.

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode and special 120° excitation mode

Note Selectable only when the 180° excitation function (TSnADC of TSnOPT5 register = 1) is used

17.6.13 Pattern switch detection flag (TSnESG)

[Name]

Pattern switch detection flag (TSnESG) - Internal signal

[Description]

TSnESG is toggled at the change timing of input pattern (TAPTSn2 to TAPTSn0).

TAPTSn2 to TAPTSn0 After Change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
TAPTSn2 to TAPTSn0 at present	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	Toggle	-	-	-	Toggle
	[1, 0, 0]	-	-	Toggle	-	Toggle	-	-	-
	[1, 1, 0]	-	-	-	Toggle	-	Toggle	-	-
	[0, 1, 0]	-	-	-	-	Toggle	-	Toggle	-
	[0, 1, 1]	-	-	-	-	-	Toggle	-	Toggle
	[0, 0, 1]	-	-	Toggle	-	-		Toggle	-

[Example]

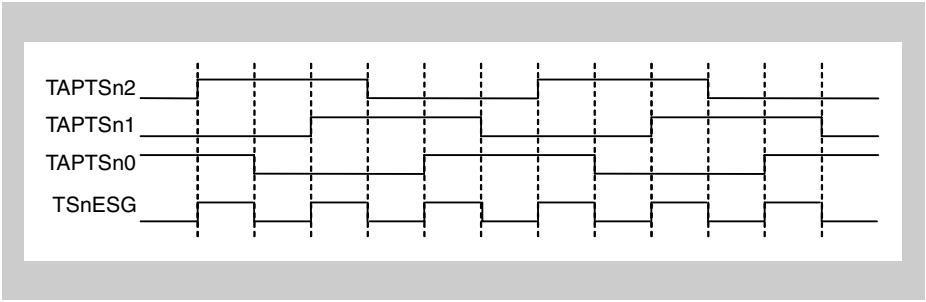


Figure 17-27 Pattern switch detection flag operation example

[Operation mode]

High-accuracy T-PWM mode[Note], PWM mode with dead time[Note], 120° excitation mode and special 120° excitation mode

Note Selectable only when the 180° excitation function (TSnADC of TSnOPT5 register = 1) is used

17.7 Interrupt Thinning out Function

The operations related to the interrupt thinning out function are as follows.

- The interrupts to be culled are peak interrupt (INTTSnCD0) and trough interrupt (INTTSnOD).
- The TSnICE bit of the TSnOPT1 register is used to enable the output of INTTSnCD0 interrupts and specify the interrupt thinning out count.
- The TSnIOE bit of the TSnOPT1 register is used to enable the output of INTTSnOD interrupts and specify the interrupt thinning out count.
- Whether or not reload thinning out is to be performed can be specified with the TSnRDE bit of the TSnOPT2 register.
  If the use of reload thinning out is specified, reloading is executed at the same time as an interrupt output after the thinning out.
  If the use of reload thinning out is not specified, reloading is executed at the reload timing after a write access is performed on the TSnCCR1 register.

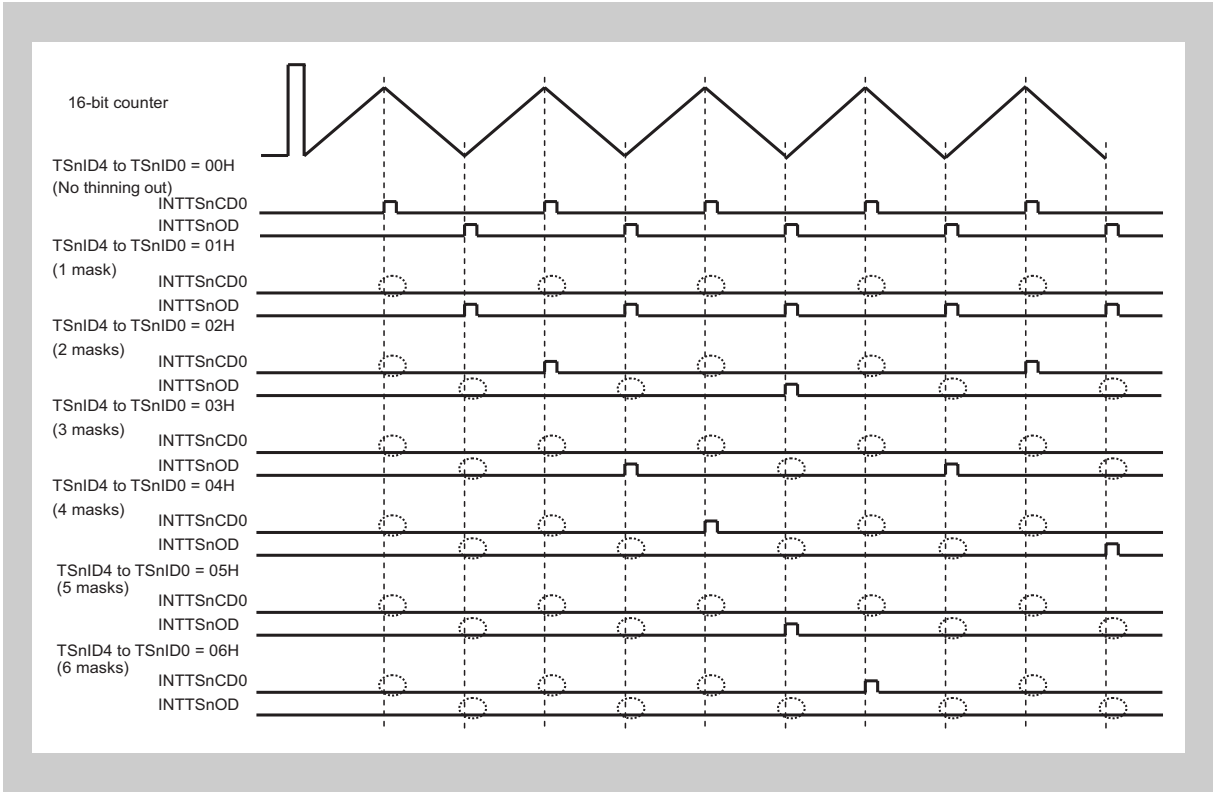
Caution

1. The interrupt thinning out function is enabled in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, or special pattern output mode.
2. If a write access is performed on the TSnOPT1 or TSnOPT1L register by using the anytime rewrite mode or reload mode (batch rewrite function), the internal thinning out counter is cleared. Therefore, the interrupt interval may be extended temporarily when the interrupt thinning out function is used. To avoid this, change the interrupt thinning out count so that the values are reloaded to the counter in synchronization with the interrupt thinning out (TSnCMS of the TSnOPT0 register = 0, or TSnRDE of the TSnOPT1 register = 1).

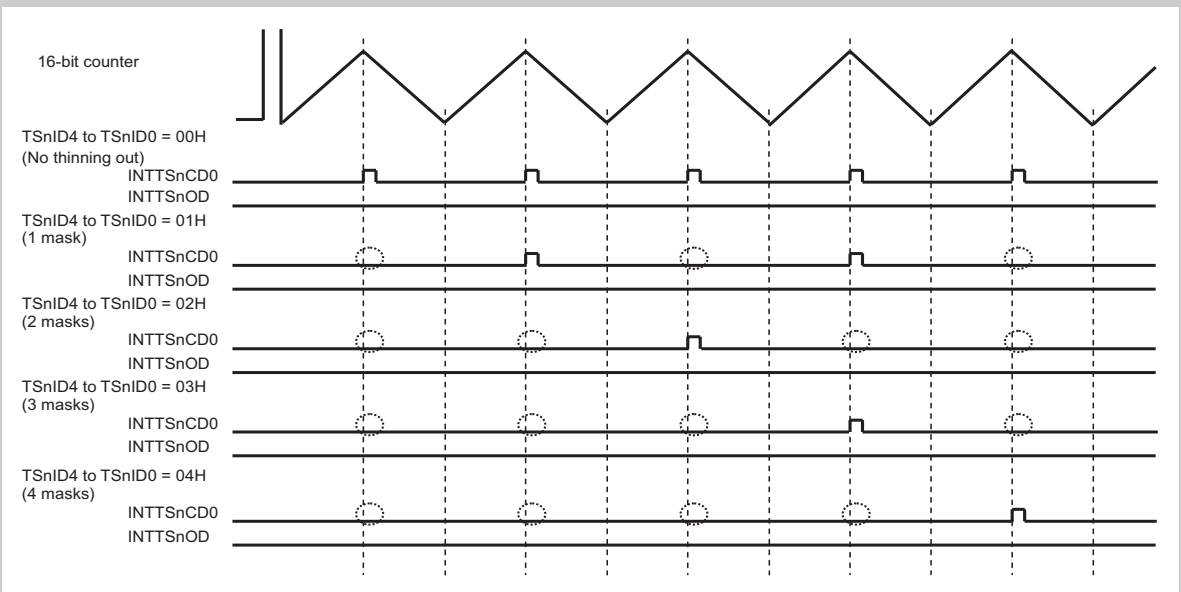
17.7.1 Interrupt thinning out function operation

rec

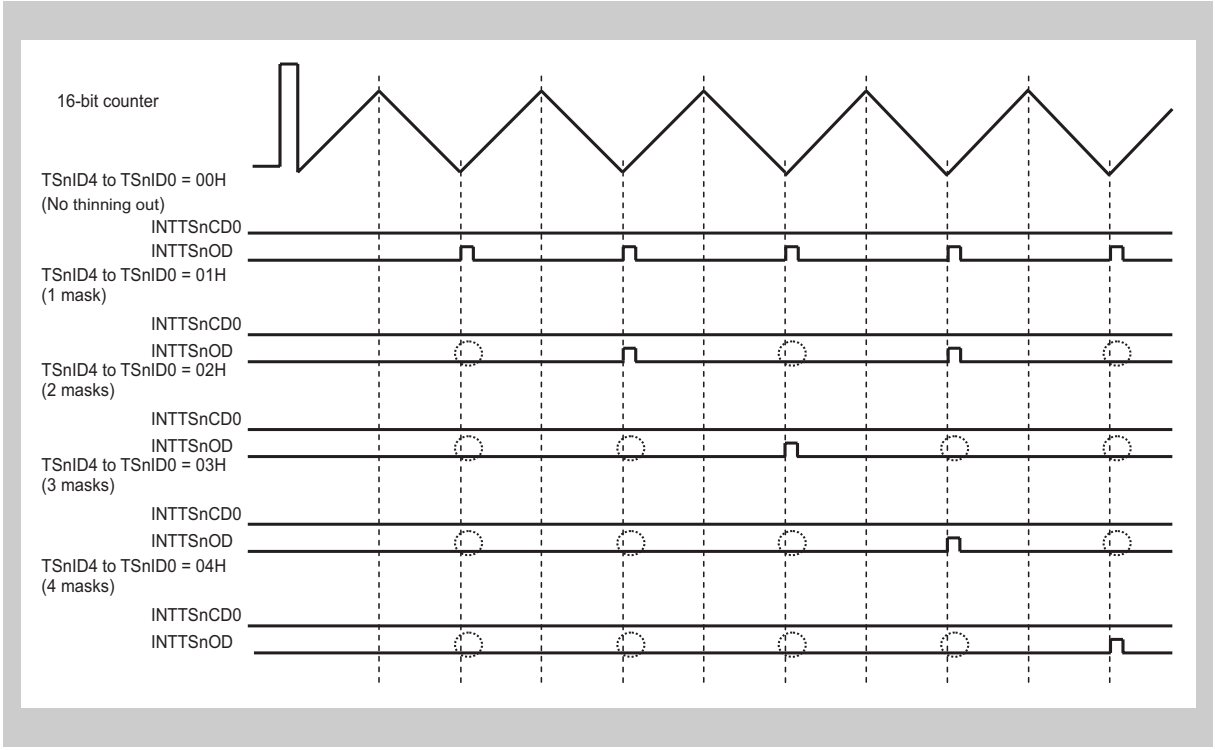
4. Interrupt thinning out operation when TSnICE and TSnIOE of the TSnOPT1 register are 1 (when a peak/trough interrupt occurs in high-accuracy T-PWM mode)



5. Interrupt thinning out operation when the TSnICE and TSnIOE of the TSnOPT1 register are 1 (when only a peak interrupt occurs in high-accuracy T-PWM mode)



6. Interrupt thinning out operation when the TSnOPT1 register's TSnICE is 0 and TSnIOE is 1 (when only a trough interrupt occurs in high-accuracy T-PWM mode)



17.7.2 Operation example when peak and trough interrupts occur alternatively (in high-accuracy T-PWM mode)

(1) Register setting

TSnICE = 1, TSnIOE = 1, TSnRBE = 1, and TSnRTE = 0 for the TSnOPT1 register

(2) Operation example

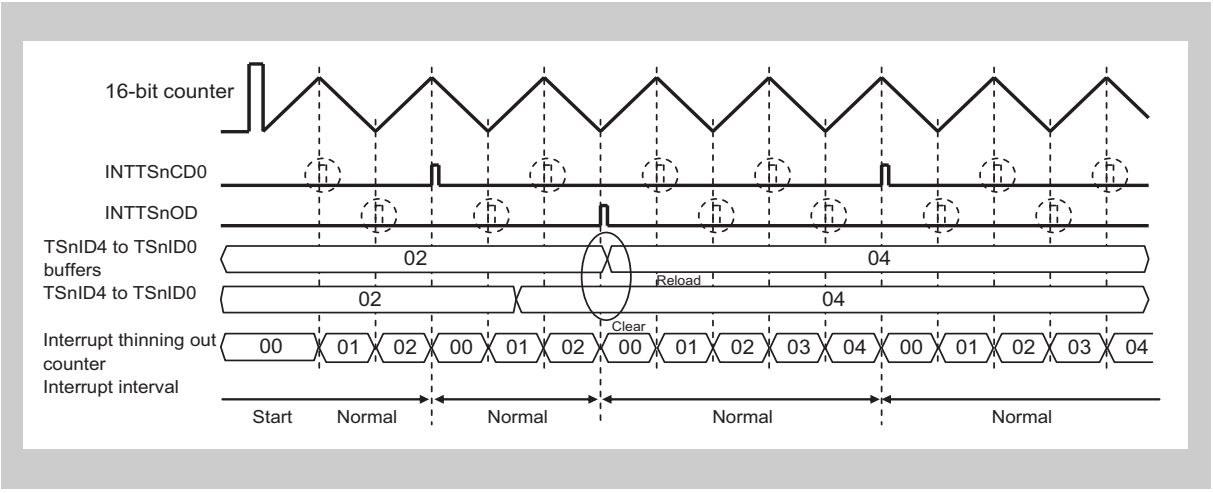


Figure 17-28 When TSnCMS of TSnOPT0 Register = 0 and TSnRDE of TSnOPT1 Register = 1 (Reload Thinning out Function Selected) (Recommended Setting)

Note Values are reloaded at the thinning out interrupt output timing, and other reload timings are ignored.

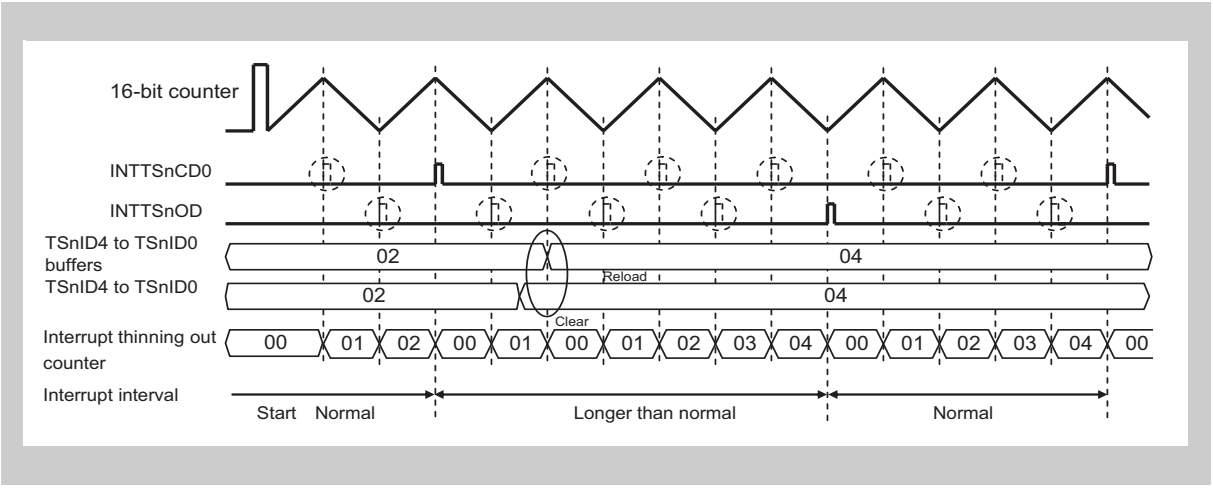


Figure 17-29 When TSnCMS of TSnOPT0 Register = 0 and TSnRDE of TSnOPT1 Register = 0 (Reload Function Selected)

Note Values are reloaded at the reload timing after rewriting.

Caution The interrupt interval may be extended.

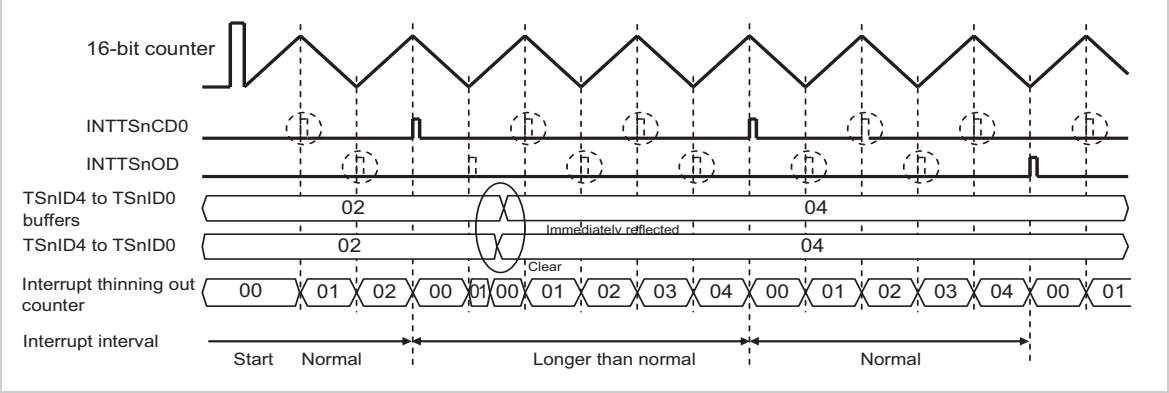


Figure 17-30 When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register is x (in anytime rewrite mode)

Note The reload timing is ignored after rewriting, and the value is immediately reflected.
The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

Caution The interrupt interval may be extended.

17.7.3 Operation example when only peak interrupt occurs (in high-accuracy T-PWM mode)

(1) Register setting

TSnICE = 1, TSnIOE = 0, TSnRBE = 0, and TSnRTE = 1 for the TSnOPT1 register

(2) Operation example

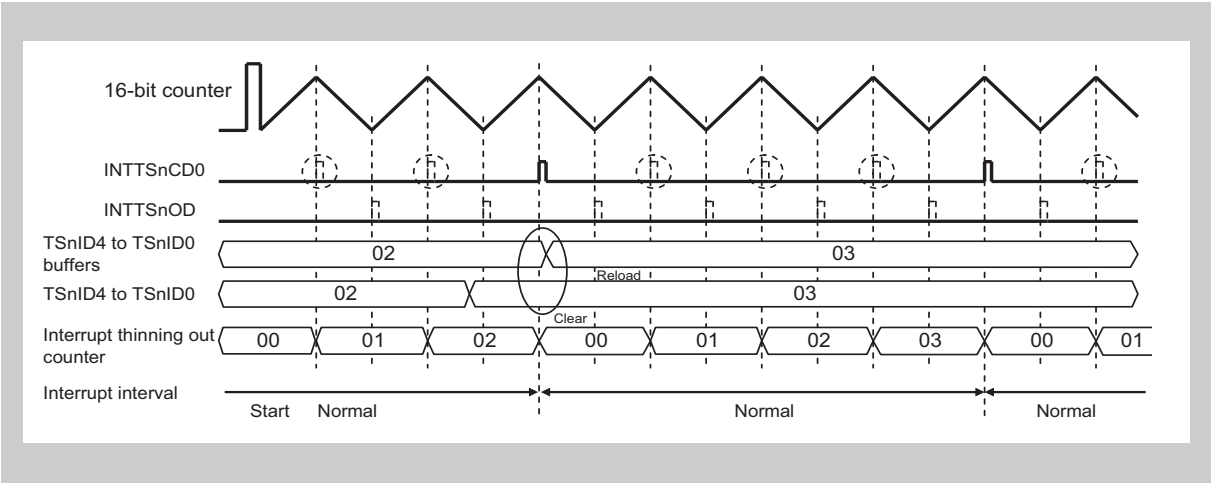


Figure 17-31 When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 1 (reload function selected) (recommended setting)

Note Values are reloaded at the thinning out interrupt output timing, and other reload timings are ignored.

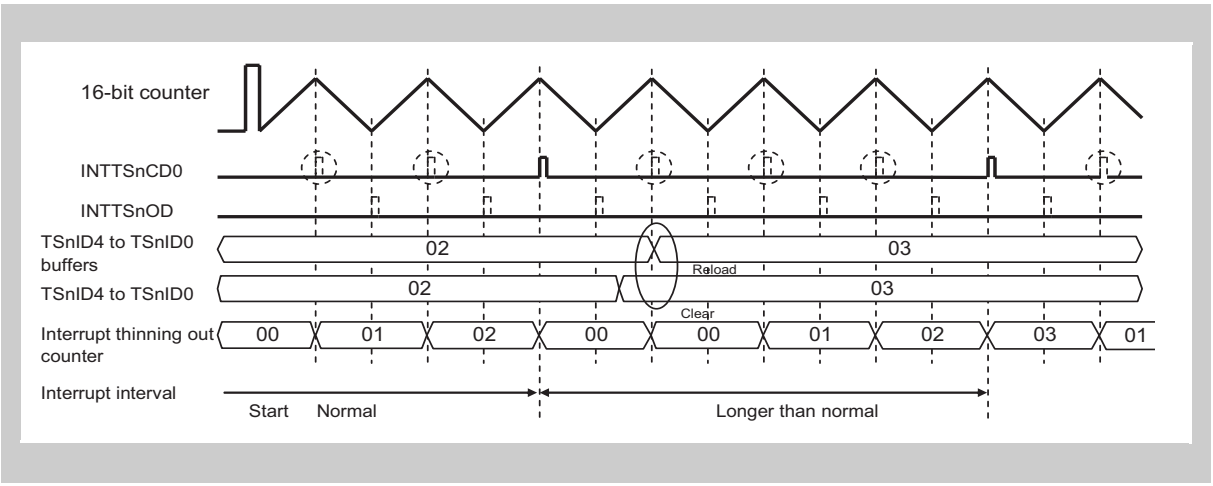


Figure 17-32 When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 0 (reload function selected)

Note Values are reloaded at the reload timing after rewriting.

Caution The interrupt interval may be extended.

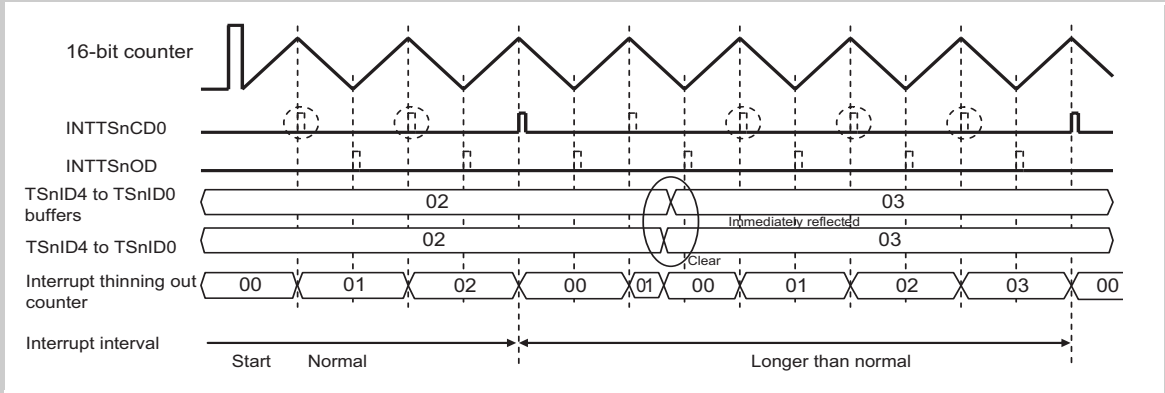


Figure 17-33 When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register = x (in anytime rewrite mode)

Note
1. The reload timing is ignored after rewriting, and then the value is immediately reflected.
2. The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

Caution The interrupt interval may be extended.

17.7.4 Operation example when only trough interrupt occurs (in high-accuracy T-PWM mode)

(1) Register setting

TSnICE = 0, TSnIOE = 1, TSnRBE = 1, and TSnRTE = 0 in the TSnOPT1 register

(2) Operation example

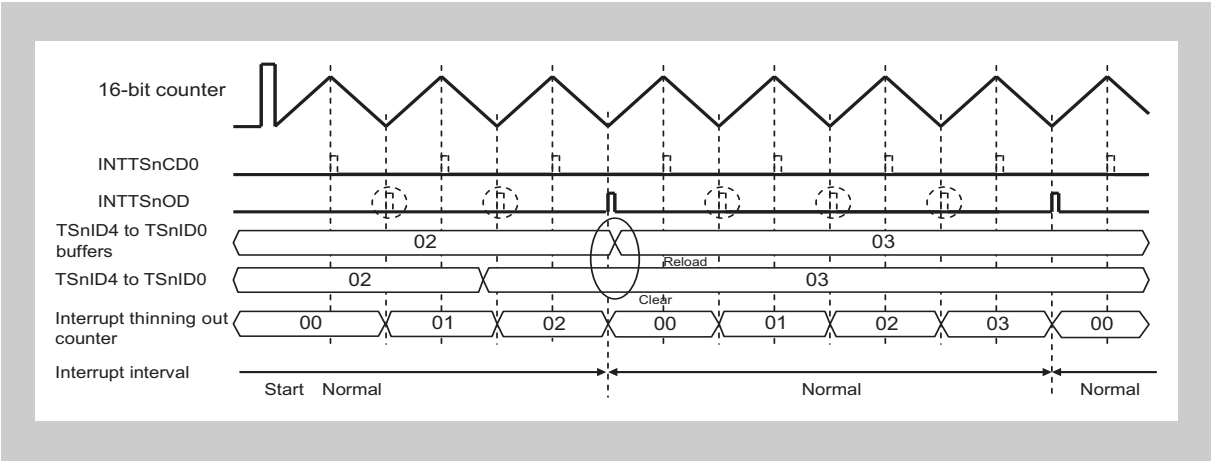


Figure 17-34 When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 1 (reload thinning out function) (recommended setting)

Note Values are reloaded at the thinning out interrupt output timing and other reload timings are ignored.

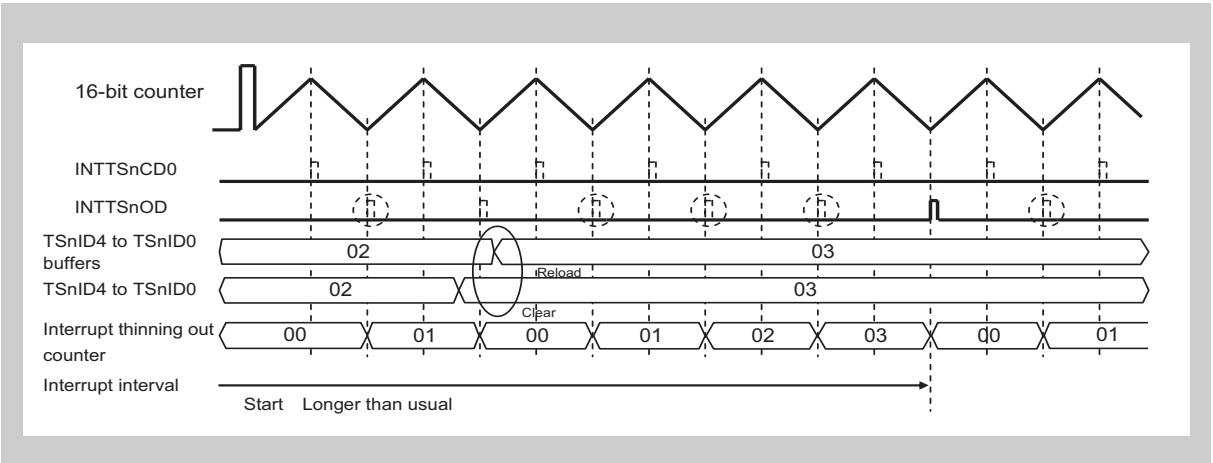


Figure 17-35 TSnOPT0 register TSnCMS = 0 and TSnOPT1 register TSnRDE = 0 (reload function)

Note Values are reloaded at the reload timing after rewriting.

Caution The interrupt interval may be extended.

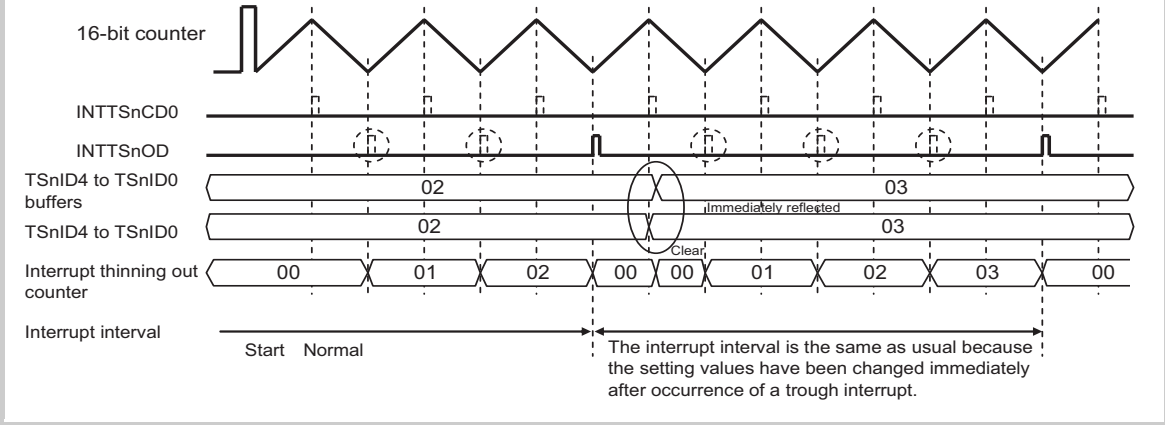


Figure 17-36 When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register = x (in anytime rewrite mode)

Note
1. The reload timing is ignored and the value is immediately reflected after rewriting.
2. The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

Caution The interrupt interval may be extended.

17.7.5 Operation example when only peak interrupt occurs (in PWM mode)

The operations related to the interrupt thinning out function in the PWM mode are as follows.

- The interrupt to be culled is the peak interrupt (INTTSnCD0). This interrupt is generated upon a match between the TSnCCR0 register and 16-bit counter during saw tooth wave operation.
- The TSnICE bit of the TSnOPT1 register is used to enable output of the INTTSnCD0 interrupt and specify the interrupt thinning out count.
- TSnIOE bit setting in the TSnOPT1 register is invalid. In this case, generation of INTTSnOD interrupts is disabled.
- Whether or not the reload thinning out is to be performed can be specified with the TSnRDE bit of the TSnOPT1 register.
  If the use of the reload thinning out is specified, reloading is executed at the same time as an interrupt output after the thinning out.
  If the use of the reload thinning out is not specified, reloading is executed at the peak timing after a write access is performed on the TSnCCR1 register.

Caution If a write access is performed on the TSnOPT1 or TSnOPT1L register by using the anytime rewrite mode or reload mode (batch rewrite function), the internal thinning out counter is cleared. Therefore, the interrupt interval may be extended temporarily when the interrupt thinning out function is used. To avoid this, change the interrupt thinning out count so that the values are reloaded to the counter in synchronization with the interrupt thinning out (TSnCMS of the TSnOPT0 register = 0, or TSnRDE of the TSnOPT1 register = 1).

(1) Operation example

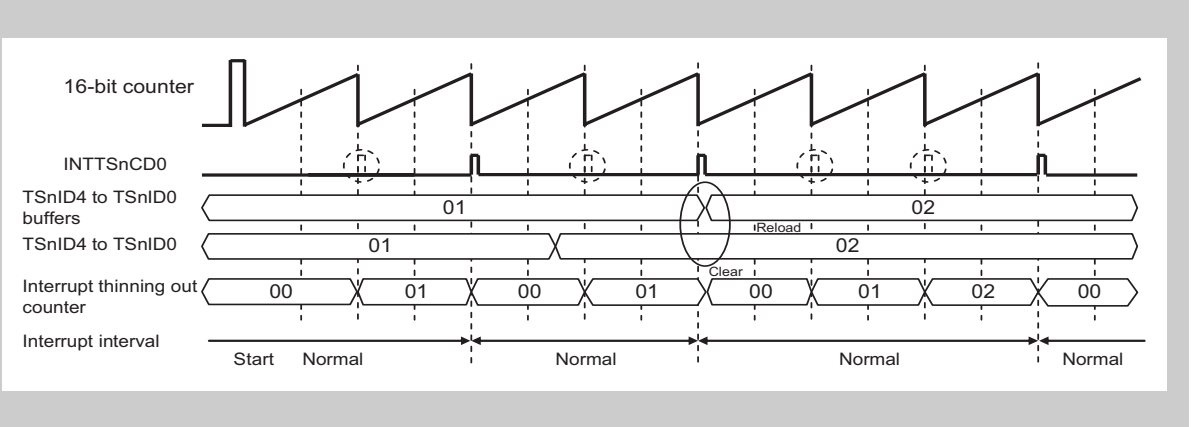


Figure 17-37 When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 1 (reload thinning out function) (recommended setting)

Note Values are reloaded at the thinning out interrupt output timing and other reload timings are ignored.

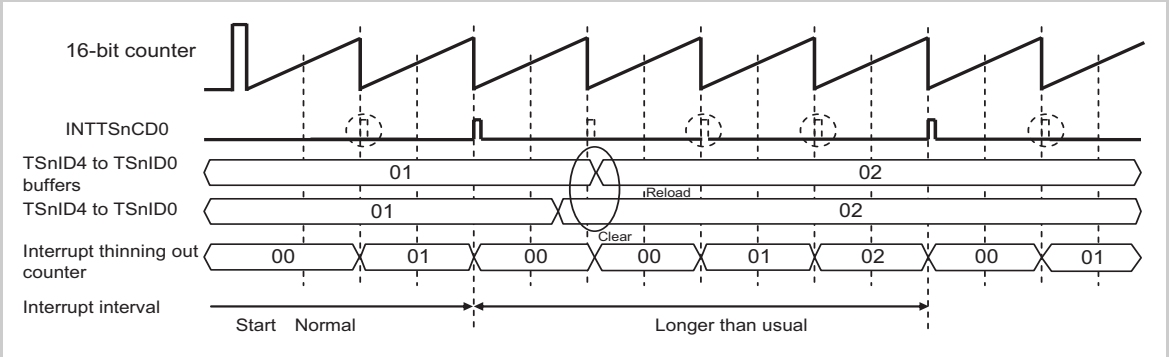


Figure 17-38 When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 0 (reload function)

Note Values are reloaded at the reload timing after rewriting.

Caution The interrupt interval may be extended.

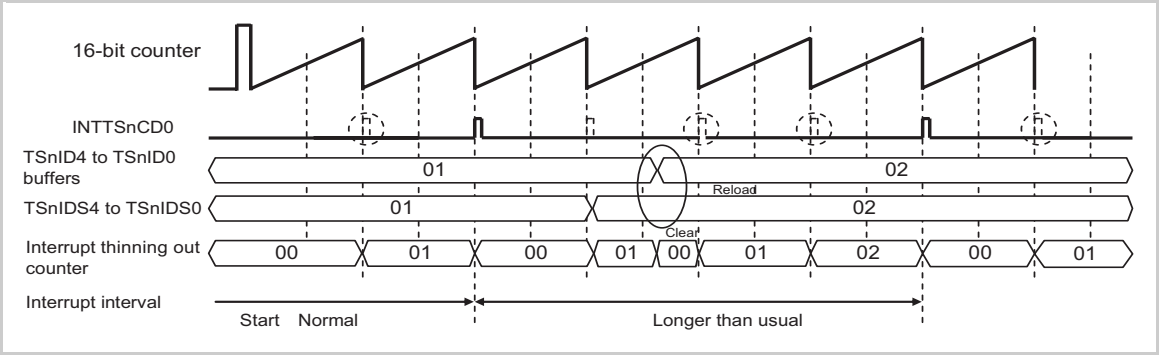


Figure 17-39 When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register = x (in anytime rewrite mode)

Note The reload timing is ignored and the value is immediately reflected after rewriting.
The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

Caution The interrupt interval may be extended.

17.8 A/D Conversion Trigger Function

The following describes the A/D conversion trigger output in PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode and special pattern output mode.
In the high-accuracy T-PWM mode, PWM mode with dead time, special 120° excitation mode, or special pattern output mode, the TSnCCR4 and TSnCCR5 registers do not affect the timer output as compare operation but are used for controlling match interrupts and A/D conversion triggers.
In the PWM mode, triangular-wave PWM mode, or 120° excitation mode, note that the setting of the TSnCCR4 and TSnCCR5 registers may affect the timer output.
Refer to *"A/D conversion trigger, peak interrupt, trough interrupt in each mode" on page 781* for details on A/D conversion triggers that can be output in each mode.

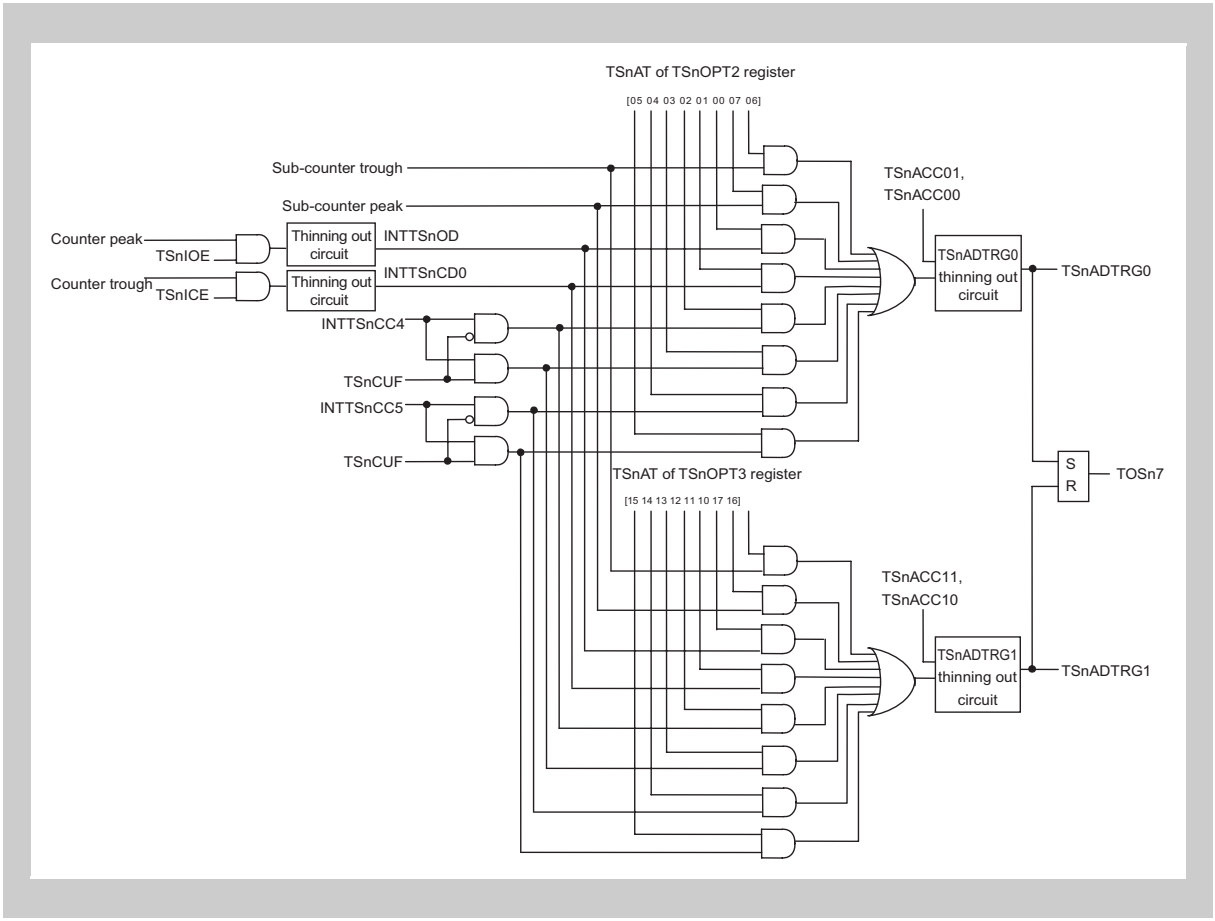


Figure 17-40 A/D conversion trigger output control circuit

The interrupt signals for compare match interrupts (INTTSnCC5, INTTSnCC4), peak interrupt (INTTSnCD0), trough interrupt (INTTSnOD), the 16-bit sub-counter peak timing, and the 16-bit sub-counter trough timing can be selected to provide OR output as shown in *Figure 17-40 on page 816*.
Timer Sn has two channels of the same A/D trigger control circuits. Each can be controlled independently.
The A/D trigger thinning out functions are equipped to enable setting with 1/2, 1/4, or 1/8 thinning out ratio, or without thinning out.

17.8.1 A/D conversion trigger operation

Timer Sn has a function that generates an A/D conversion start trigger (TSnADTRG0 or TSnADTRG1 signal) by arbitrarily selecting eight trigger sources.
The following eight trigger sources can be selected with the TSnAT07 to TSnAT00 bits of the TSnOPT2 register or the TSnAT17 to TSnAT10 bits of the TSnOPT3 register.
The following describes TSnADTRG0/TSnADTRG1 signal control using control bits TSnAT07 to TSnAT00/TSnAT17 to TSnAT10.

(1) Output control of TSnADTRG0/TSnADTRG1 (TSnOPT2 and TSnOPT3 registers)

<Trigger source>

- TSnAT00/TSnADT10 = 1
  An A/D conversion trigger is generated when a trough interrupt (INTTSnOD) occurs
- TSnAT01/TSnADT11 = 1
  An A/D conversion trigger is generated when a peak interrupt (INTTSnOD0) occurs.
- TSnAT02/TSnADT12 = 1
  An A/D conversion trigger generation is enabled when a compare match interrupt (INTTSnCC4) occurs during 16-bit counter count-up operation.
- TSnAT03/TSnADT13 = 1
  An A/D conversion trigger is enabled when a compare match interrupt (INTTSnCC4) occurs during 16-bit counter count-down operation.
- TSnAT04/TSnADT14 = 1
  An A/D conversion trigger generation is enabled when a compare match interrupt (INTTSnCC5) occurs 16-bit counter during count-up operation.
- TSnAT05/TSnADT15 = 1
  An A/D conversion trigger generation is enabled when a compare match interrupt (INTTSnCC5) occurs during 16-bit counter count-down operation.
- TSnAT06/TSnADT16 = 1
  An A/D conversion trigger generation is enabled at a peak of the 16-bit sub-counter (when the timing is switched from up to down).
- TSnAT07/TSnADT17 = 1
  A/D conversion trigger generation is enabled at a trough of the 16-bit sub-counter (when the timing is switched from down to up).

<Thinning out setting>

- TSnACC01, TSnACC00 bits/TSnACC11, TSnACC10 bits
  TSnADTRG0/TSnADTRG1 signal thinning out setting

A/D conversion start trigger signals are undergo ORing, and then culled as specified with the TSnACC01, TSnACC00/TSnACC11, TSnACC10 bits, and are output to the TSnADTRG0 or TSnADTRG1 signal.
The peak interrupt (INTTSnOD) and trough interrupt (INTTSnOD) selected by the TSnAT00 and TSnAT01/TSnAT10, TSnAT11 bits are signals that are output after interrupt thinning out.
Therefore, output is provided at the timing when it is subject to interrupt thinning out control. If interrupt output (TSnICE and TSnIOE bits of TSnOPT1 register) is not enabled, neither can the A/D conversion start trigger be output.

The TSnACC01, TSnACC00, TSnAT07 to TSnAT00/TSnACC11, TSnACC10, TSnAT17 to TSnAT10 bits can be rewritten during timer operation.
If a bit that sets an A/D conversion start trigger is rewritten during timer operation, the output status of the A/D conversion start trigger is reflected immediately.
These control bits can be written anytime regardless of the operation mode.
When TSnACC01, TSnACC00/TSnACC11, TSnACC10 are written, the A/D conversion trigger thinning out counter is cleared and starts counting from 0. (The thinning out counter is also cleared when the same value is rewritten.)

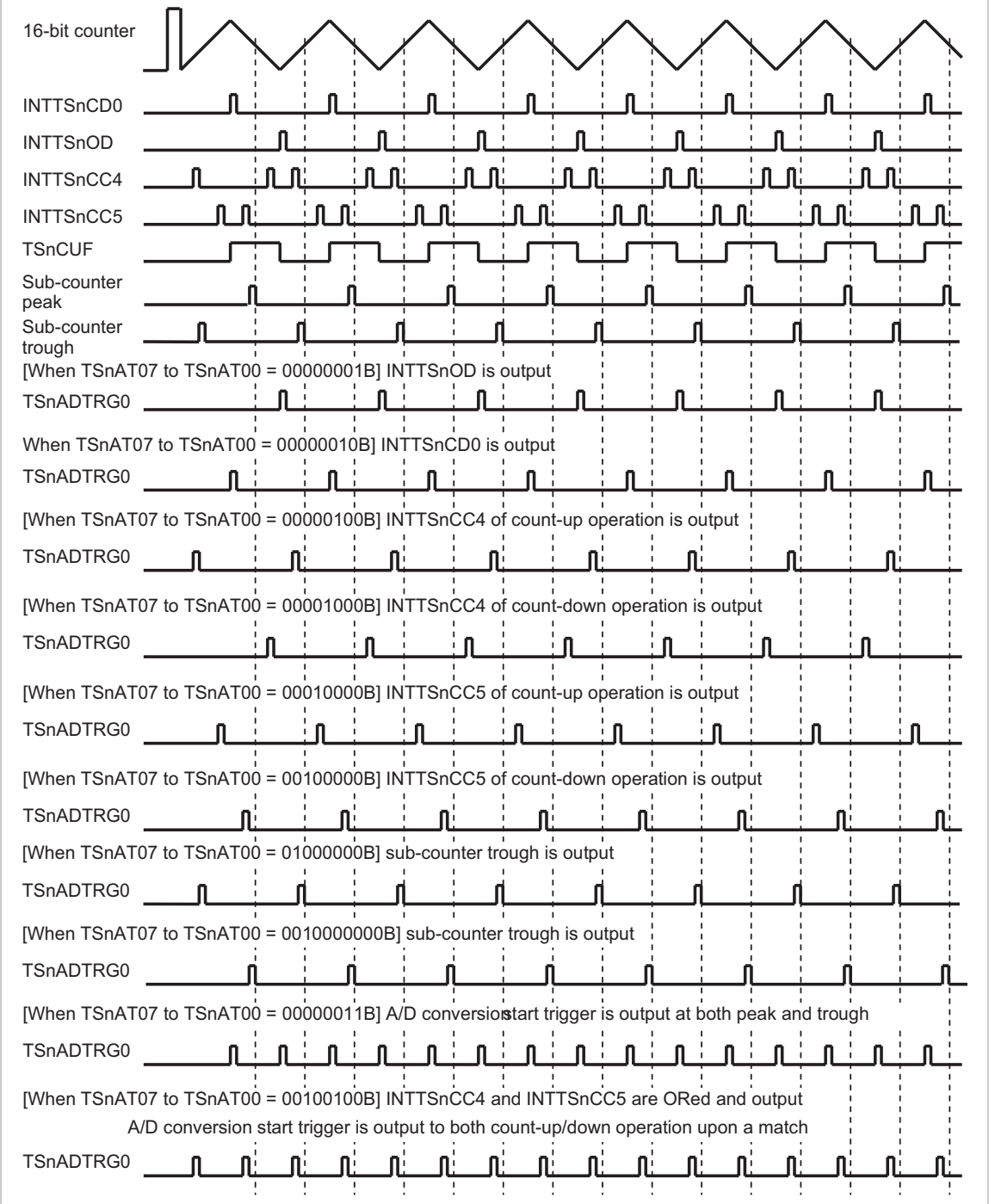


Figure 17-41 When TSnICE = 1, TSnIOE = 1, TSnID4 to TSnID0 = 00H for TSnOPT1 register and TSnACC01, TSnACC00 bits = 00H for TSnOPT2 register (in high-accuracy T-PWM mode)

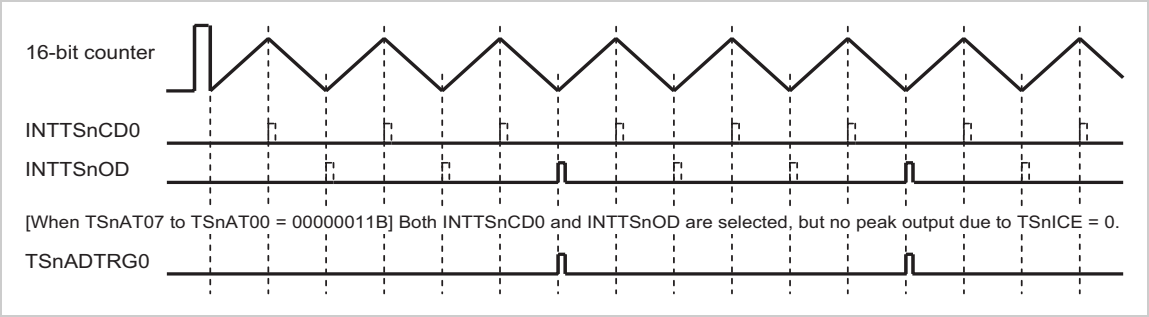


Figure 17-42 When TSnICE = 0, TSnIOE = 1, TSnID4/TSnID0 = 02H for TSnOPT1 Register and TSnACC01, TSnACC00 Bits = 00H for TSnOPT2 Register (in High-Accuracy T-PWM Mode)

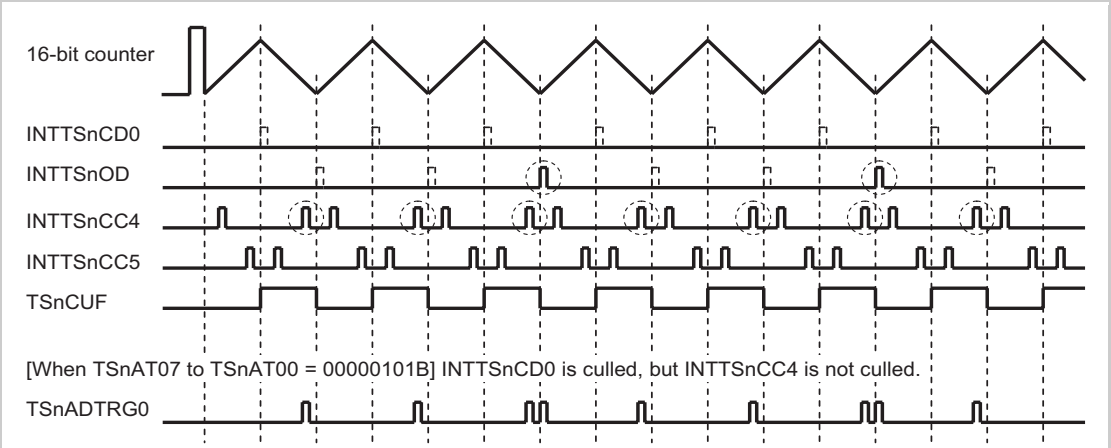


Figure 17-43 When TSnICE = 0, TSnIOE = 1, TSnID4 to TSnID0 = 02H for TSnOPT1 register and TSnACC01, TSnACC00 Bits = 00H for TSnOPT2 register (in high-accuracy T-PWM mode)

(2) A/D conversion trigger thinning out function

An operation example of the A/D conversion trigger thinning out function is shown below.

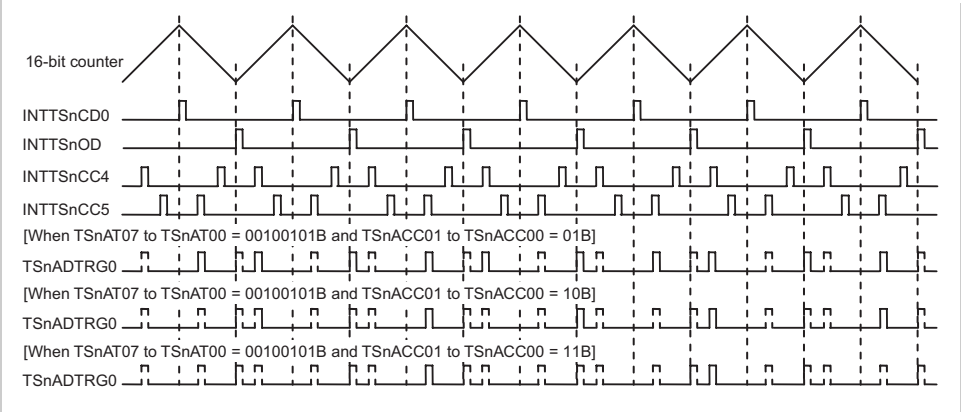


Figure 17-44 Operation example of A/D conversion trigger thinning out function (except TSnDTC = 000H)

Caution The AD trigger thinning out counter operation differs according to whether the dead time register (TSnDTC0) is “000H” or “other than 000H”.
When the TSnDTC0 register is “000H”, the 16-bit sub-counter trough of the first cycle from timer count start is not counted so that there is a difference in the AD trigger thinning out counter start timing as shown in *Figure 17-44 on page 821*.

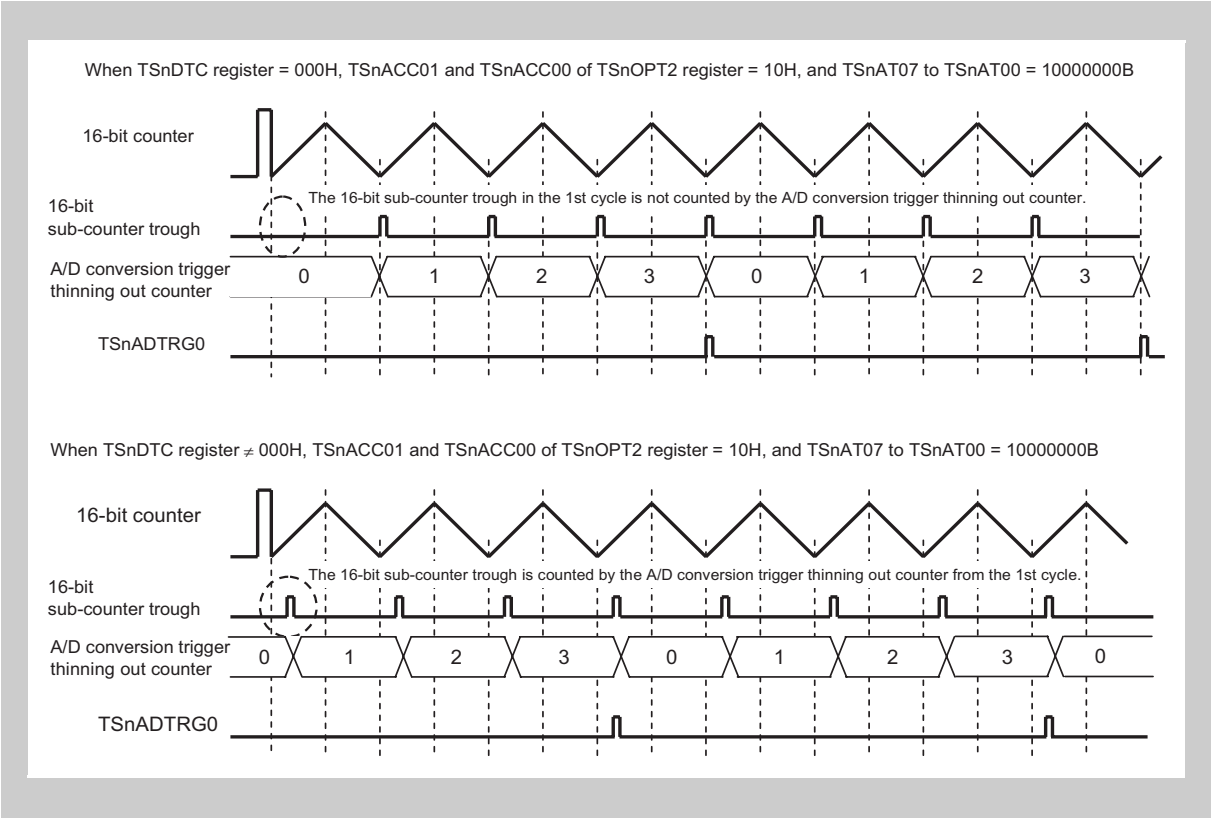


Figure 17-45 Operation example of A/D Conversion trigger thinning out function (TSnDTC = 000H)

(3) Cautions for A/D conversion trigger

- The count value of the A/D conversion trigger thinning out counter is incremented by 1 each time the same value is written (write access) to the TSnCCR4 register and TSnCCR5 register, and a valid A/D trigger is set under the same condition during count-up/down operation of the 16-bit counter. Output trigger is 1 pulse as well.
- The trough interrupt (INTTSnOD) does not occur in the PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Only the peak interrupt (INTTSnCD0) is valid.

17.9 Error/warning Interrupt

17.9.1 Error interrupt function

When a positive/negative phase simultaneous active is detected, the TSnTBF flag of the TSnOPT6 register is set and an error interrupt (INTTSnER) of timer Sn is generated. Switching ON/OFF of error detection in each phase (TOSn1/TOSn2, TOSn3/TOSn4, TOSn5/TOSn6) is possible with the TSnTBA2 to TSnTBA0 bits of the TSnIOC4 register after error interrupts are enabled (TSnEOC of TSnIOC4 register = 1).

The following table lists whether or not positive/negative simultaneous active error detection is possible in each mode.

Mode	Positive/negative phase simultaneous active error detection
Interval timer mode	-
External event count mode	-
External trigger pulse output mode	-
One-shot pulse mode	-
PWM mode	√
Free-running mode	-
Triangular-wave PWM mode	√
High-accuracy T-PWM mode	√
PWM mode with dead time	√
120° excitation mode	√
Special pattern output mode	√

Note √: Detection possible
–: Detection not possible

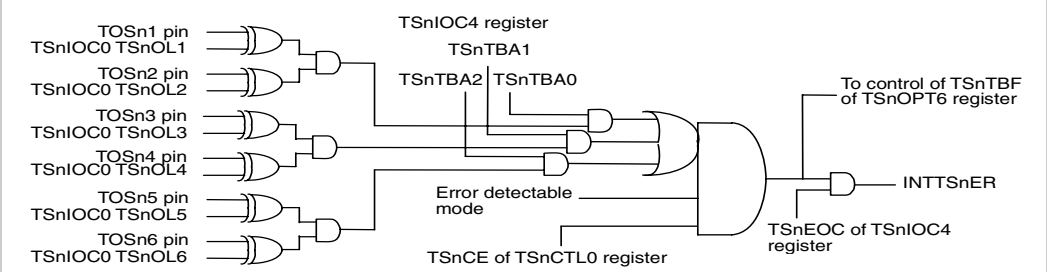


Figure 17-46 Error interrupt (INTTSnER) output control circuit

Caution If an error interrupt occurs, release the error status during error interrupt servicing. An error interrupt will not occur thereafter until the error status is released.

(1) In PWM/special pattern output mode

1. When the dead time control function is not used (TSnDSE of TSnOPT0 register = 0 in PWM mode)
   As shown in *Figure 17-47 on page 824*, an error interrupt (INTTSnER) occurs when the TSnCCR1 and TSnCCR2 registers are set so that the active level is output from the TOSn1 and TOSn2 pins simultaneously. An error interrupt (INTTSnER) also occurs when the TSnCCR3 and TSnCCR4 registers are set so that the active level is output from the TOSn3 and TOSn4 pins simultaneously.

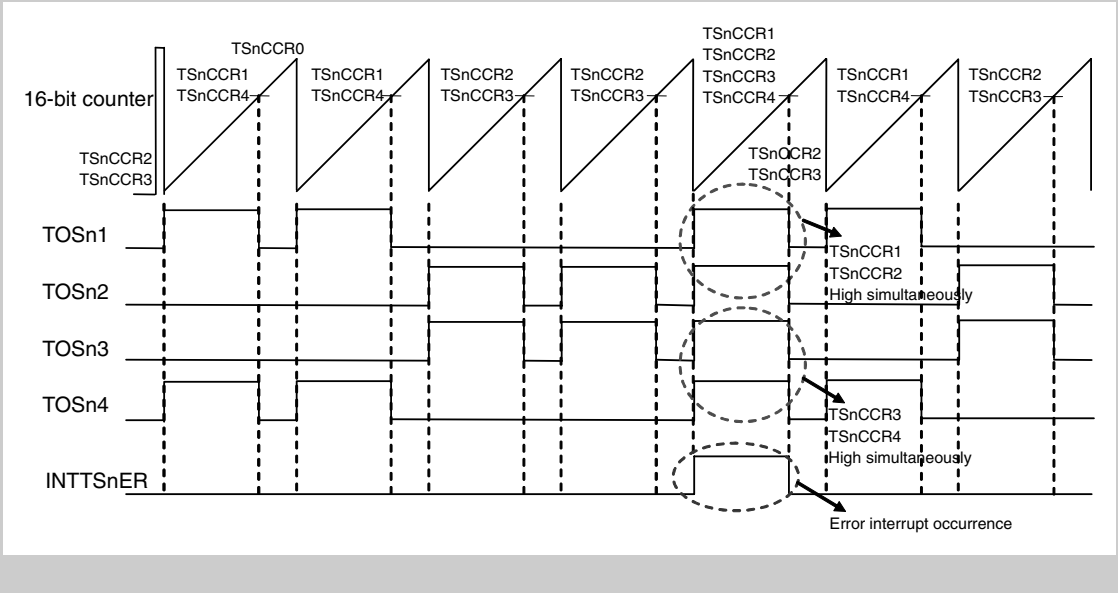


Figure 17-47 Generation of error interrupt (INTTSnER) (in PWM mode)

The following is the case where the output active level is switched by using the TSnOL1 and TSnOL2 bits of the TSnIOC0 register.

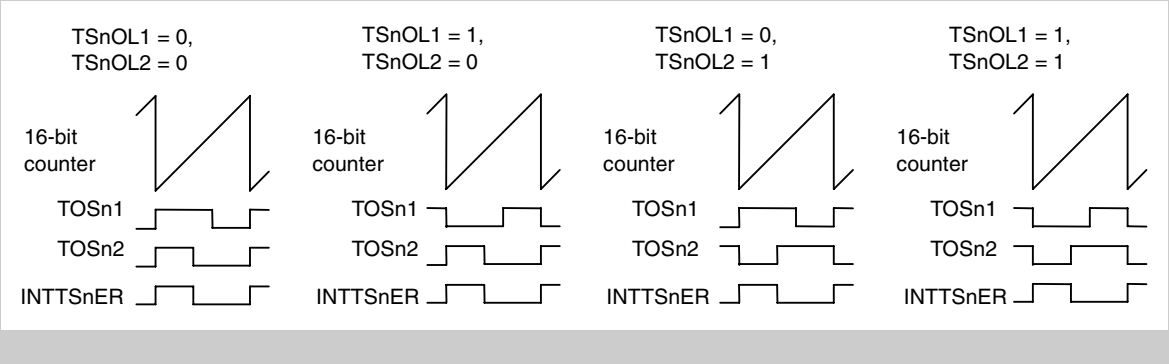


Figure 17-48 Generation of error interrupt in each active level (INTTSnER)

2. When the dead time control function is used (TSnDSE of the TSnOPT0 register = 1 in PWM mode, or in special pattern output mode)
   Even if the timer output is inactive, an error interrupt occurs when the set conditions occurs simultaneously. For details refer to
   – *"Dead time control in high-accuracy T-PWM mode" on page 900*
   – *"Dead time control in PWM mode with dead time" on page 922*
   – *"Dead time control in 120° excitation mode" on page 952*
   – *"Dead time control in special 120° excitation mode" on page 994*
   – *"Dead time control in special pattern output mode" on page 1007*

(2) Triangular-wave PWM mode

As shown in *Figure 17-49 on page 825*, an error interrupt (INTTSnER) occurs when the TSnCCR0 and TSnCCR1 registers are set so that the active level is output from the TOSn1 and TOSn2 pins simultaneously. An error interrupt (INTTSnER) also occurs when the TSnCCR3 and TSnCCR4 registers are set so that the active level is output from the TOSn3 and TOSn4 pins simultaneously.

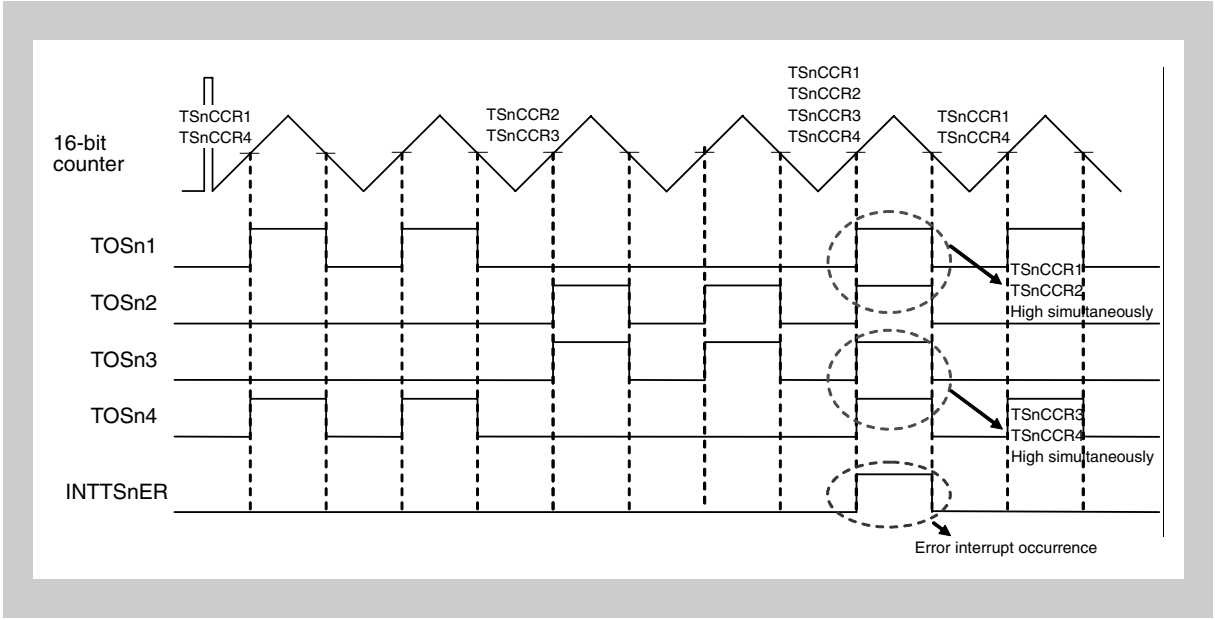


Figure 17-49 Generation of error interrupt (INTTSnER) (in triangular PWM mode)

(3) In high-accuracy T-PWM mode, PWM mode with dead time, or 120° excitation mode

Errors caused by setting values of the compare register, the dead time setting register in the high-accuracy T-PWM mode, PWM mode with dead time, or 120° excitation mode do not occur.

(4) In special 120° excitation mode

An error interrupt (INTTSnER) occurs when timer Sn pattern registers 0 and 1 (TSnPAT0, TSnPAT1) are set so that the active level is output simultaneously. Refer to the caution on setting of the TSnPAT0 and TSnPAT1 registers in the special 120° excitation mode in *"Difference between 120° excitation mode and special 120° excitation mode" on page 997* for details.

17.9.2 Warning interrupt function

Timer S has the warning interrupt (INTTSnWN) function.
A warning interrupt (INTTSnWN) is generated when any of the following conditions is detected.
Refer to *"Flags" on page 787* for details.

- When two or more simultaneous pin changes are detected at the TAPTSn2 to TAPTSn0 pins
  (Refer to *"Noise detection flag (TSnNDF) " on page 792*)
- When a reverse is detected at the TAPTSn2 to TAPTSn0 pins
  (Refer to *"Pattern reverse detection flag (TSnPRF) " on page 796*)
- When [0, 0, 0] or [1, 1, 1] is detected at the TAPTSn2 to TAPTSn0 pins
  (Refer to *"Pattern error detection flag (TSnPEF) " on page 795*)
- When the TSnPAT2 to TSnPAT0 pins are toggled three times or more during TSnSTCI0 and TSnSTCI1 signal triggers
  (Refer to *"TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF) " on page 798*)
- When the TSnSTCI0 and TSnSTCI1 signals are detected simultaneously
  (Refer to *"TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) " on page 800*)
- When the phase relation between the input pattern (TAPTSn2 to TAPTSn0) and the output pattern (TSnOPF2 to TSnOPF0) is displaced
  (Refer to *"Pattern phase difference detection flag (TSnPPF) " on page 801*)

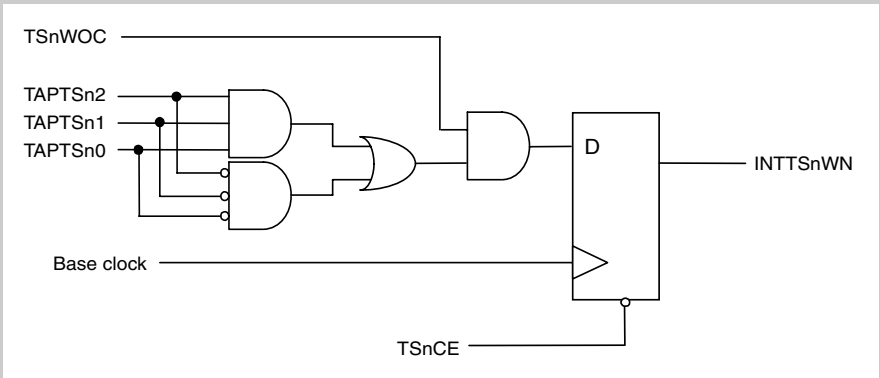


Figure 17-50 Error detection at TAPTSn2 to TAPTSn0 pins

17.10 Operation in Each Mode

Table 17-17 Mode list

TSnCTL1 Register				Timer Mode
TSnMD3	TSnMD2	TSnMD1	TSnMD0	
0	0	0	0	Interval timer mode
0	0	0	1	External event count mode
0	0	1	0	External trigger pulse output mode
0	0	1	1	One-shot pulse output mode
0	1	0	0	PWM mode
0	1	0	1	Free-running mode
0	1	1	1	Triangular-wave PWM mode
1	0	0	0	High-accuracy T-PWM mode[Notes 1, 2]
1	0	0	1	PWM mode with dead time[Note 1]
1	0	1	0	120° excitation mode[Notes 1, 2]
1	0	1	1	Special 120° excitation mode[Notes 1, 2]
1	1	0	0	Special pattern output mode
Other than above				Setting prohibited

Note 1. The software output function can be used by setting TSnSOC of the TSnOPT4 register to 1.

2. 180° excitation control can be used by setting TSnADC of the TSnOPT5 register to 1.

17.10.1 Interval timer mode

In the interval timer mode, a compare match interrupt (INTTSnCC0) is generated upon a match between the TSnCCR0 register and 16-bit counter, and the 16-bit counter is then cleared. The interval time means the INTTSnCC0 generation interval.
In the interval timer mode, the 16-bit counter is cleared only upon a match between the 16-bit counter and the TSnCCR0 register.
The 16-bit counter is not cleared by using the TSnCCR1 to TSnCCR5 registers.
Setting values of the TSnCCR1 to TSnCCR5 registers are transferred to the TSnCCR1 to TSnCCR5 buffer registers to be compared with the 16-bit counter value, and a compare match interrupt (INTTSnCC1 to INTTSnCC5) then occurs.
The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer operation (TSnCE bit values of TSnCTL0 register).
Toggle output is performed from the TOSn0 to TOSn7 pins by setting the TSnOE0 to TSnOE7 bits of the TSnIOC0 register to 1.

Note The interval timer mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 0, 0].

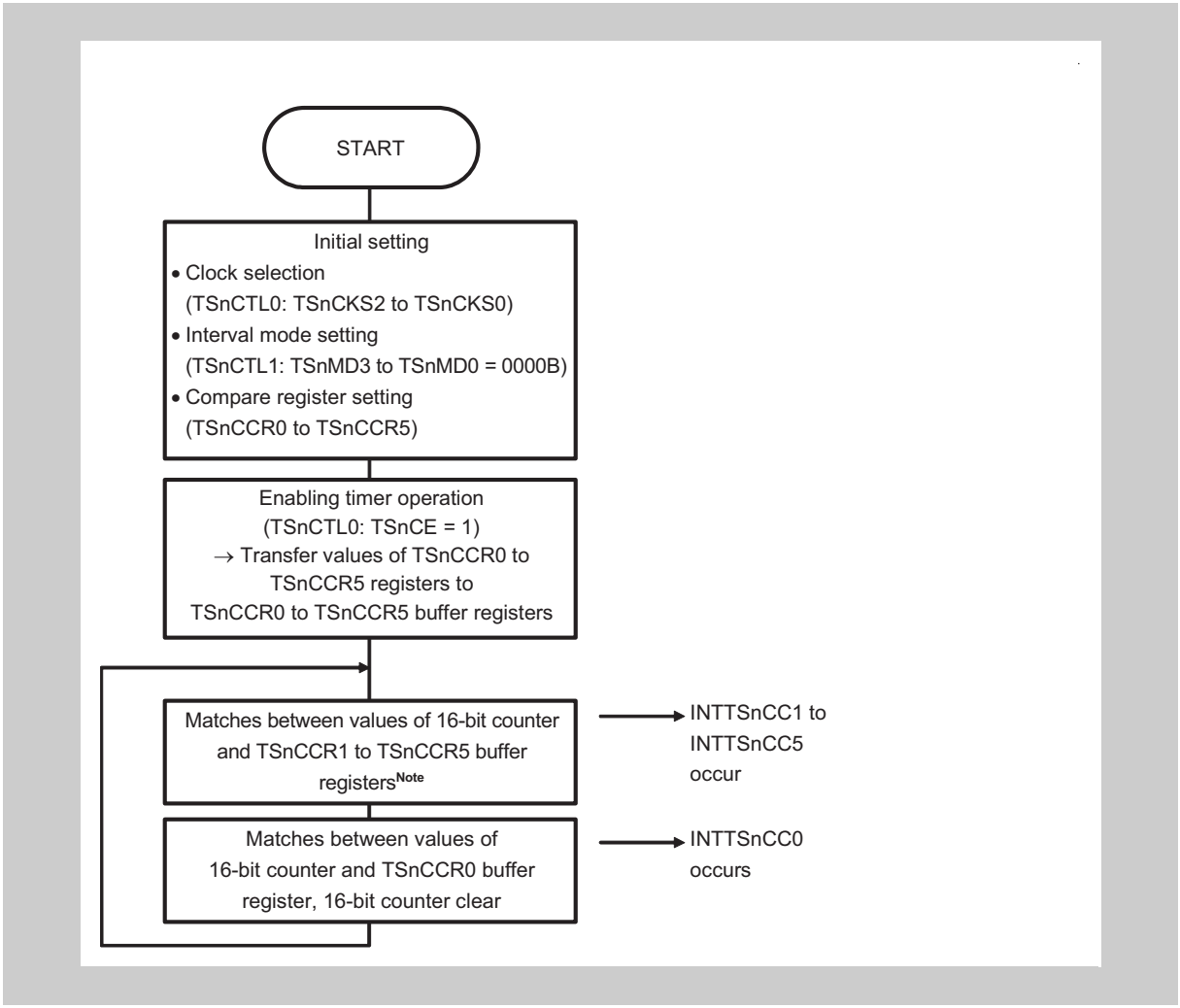


Figure 17-51 Basic operation flow in interval timer mode

Note The 16-bit counter is not cleared upon a match with the TSnCCR1 to TSnCCR5 registers.

(1) Interval timer mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Anytime rewrite	Possible	Compare value
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Compare value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pins

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pins

Pin	Function
TOSn0 to TOSn5	Toggle output upon a compare match with the TSnCCR0 to TSnCCR5 registers
TOSn6, TOSn7	-

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match with the TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from TSnCCR1 to TSnCCR5 to TSnCCR1 to TSnCCR5 + 1

Note "–" indicates an unused function in the interval timer mode.

In case of D1 > D2 > D3, only value of the TSnCCR0 register is rewritten, without output from TOSn0 and TOSn1

(TSnOE0 = 0, TSnOE1 = 0, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0 register)

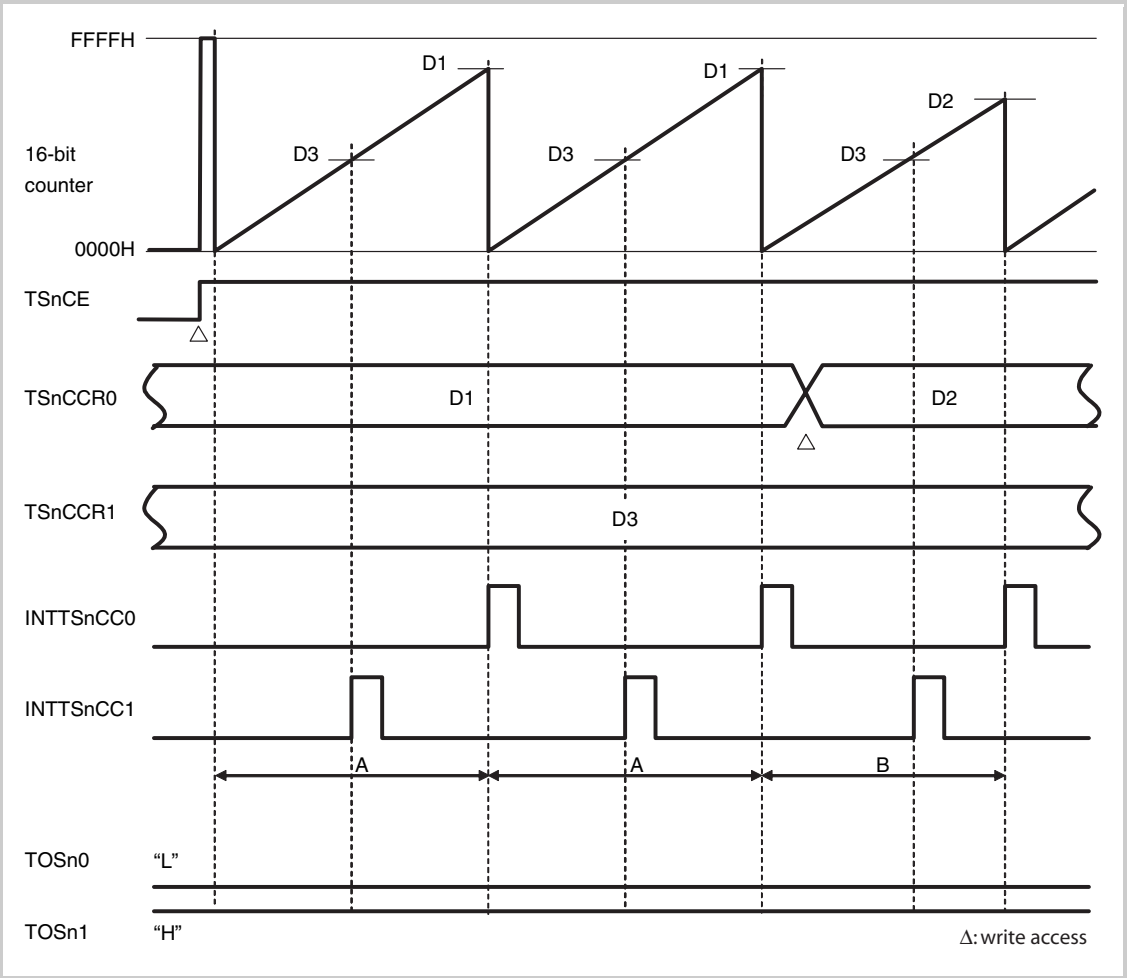


Figure 17-52 Basic timing in interval timer mode (1/2)

Note 1. D1, D2: TSnCCR0 register setting values (0000H to FFFFH)
D3: TSnCCR1 register setting value (0000H to FFFFH)

2. Interval time =A: interval time (D1 + 1) × count clock cycle
B: Interval time (D2 + 1) × count clock cycle

3. TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

In case of D1 = D2, without rewriting of TSnCCR0 and TSnCCR1 registers, with output from TOSn0 and TOSn1
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0 register)

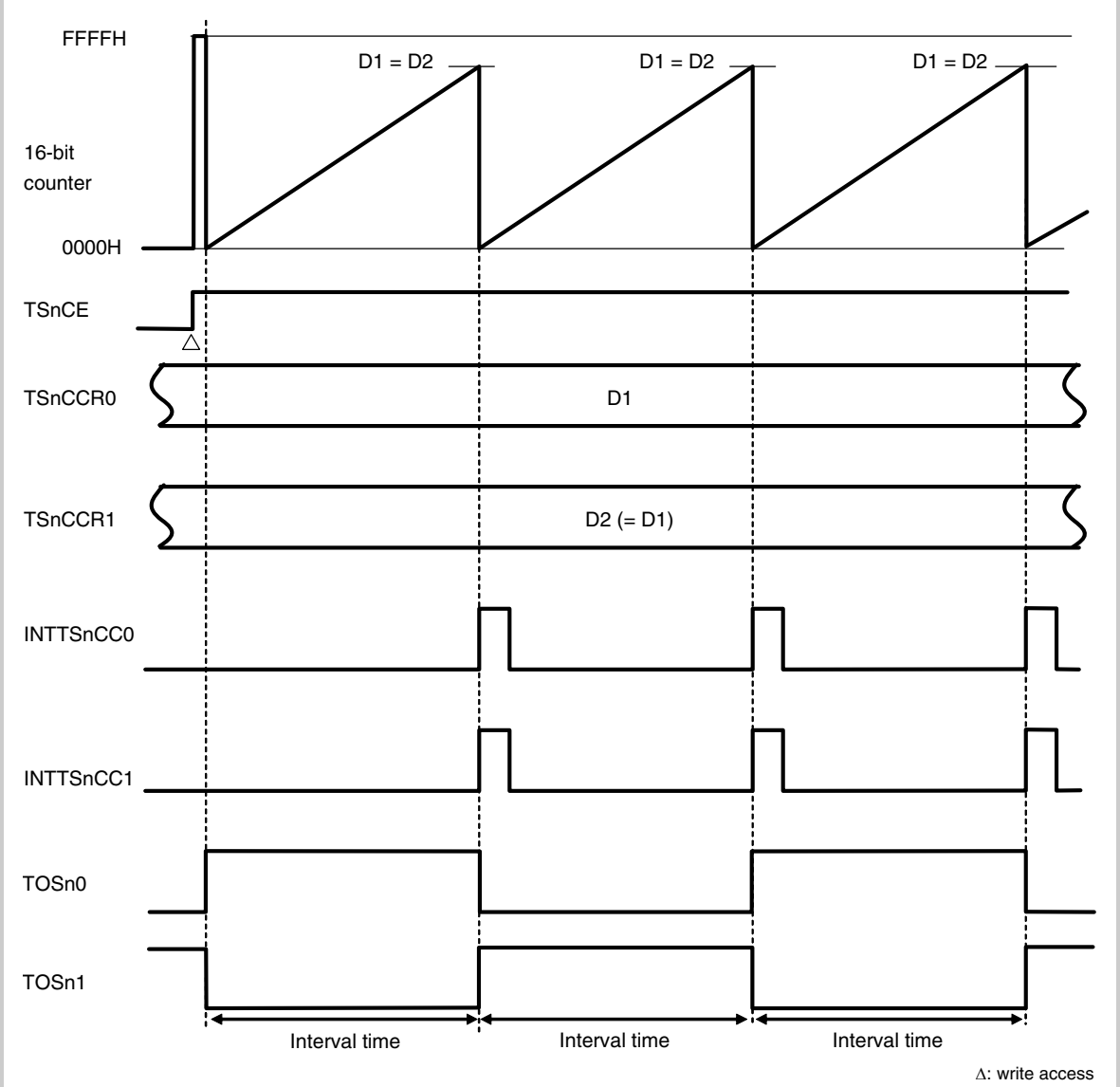


Figure 17-53 Basic timing in interval timer mode (2/2)

Note 1. D1: TSnCCR1 register setting value (0000H to FFFFH)
D2: TSnCCR1 register setting value (0000H to FFFFH)

2. Interval time = TOSn0, TOSn1 toggle time = (D1 + 1) × (count clock cycle)

3. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

17.10.2 External event count mode

In the external event count mode, the count-up operation starts with external event input (TEVTSn pin). At that time, the input signal from the TEVTSn pin is used as the count clock, regardless of the TSnEEE bit of the TSnCTL1 register. In the external event count mode, the 16-bit counter is cleared only upon a match between the 16-bit counter and the TSnCCR0 register. The 16-bit counter cannot be cleared by using the TSnCCR1 to TSnCCR5 registers. The setting values of the TSnCCR1 to TSnCCR5 registers are transferred to the TSnCCR1 to TSnCCR5 buffer registers to be compared with the 16-bit counter value, and a compare match interrupt (INTTSnCC1 to INTTSnCC5) then occurs.
The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer Sn operation (TSnCE bit value of TSnCTL0 register).
Toggle output is performed from the TOSn0 to TOSn7 pins by setting the TSnOE0 to TSnOE7 bits of the TSnIOC0 register to 1.

Note
1. In the external event count mode, the edge detection count of the TEVTSn pin input is m + 1, where the TSnCCR0 register setting value is m.
2. The external event count mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 0, 1].

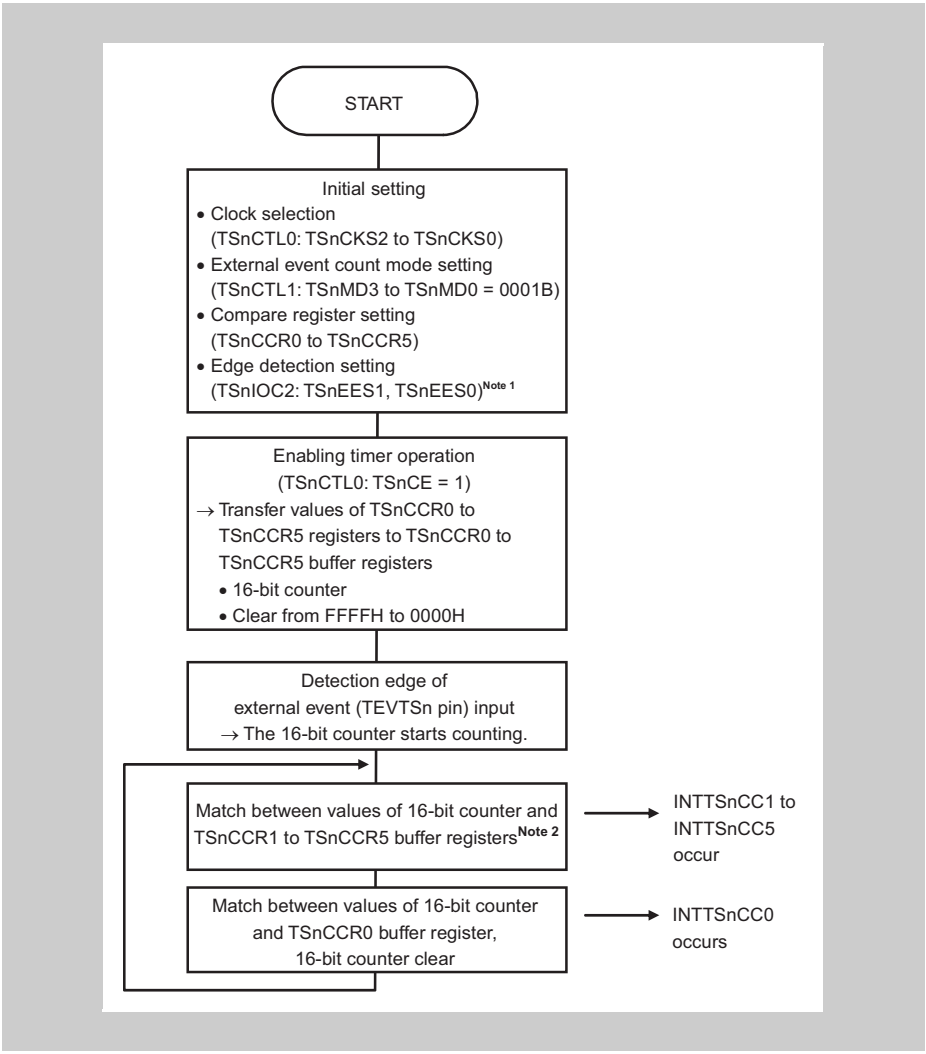


Figure 17-54 Basic operation flow in external event count mode

Note 1. Set other than TSnEES1 = 0 and TSnEES0 = 0.

2. The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

(1) External event count mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Anytime rewrite	Possible	Compare value
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Compare value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	Timer count through external event count signal

(c) Output pin

Pin	Function
TOSn0 to TOSn5	Toggle output upon a compare match with TSnCCR0 to TSnCCR5 registers
TOSn6, TOSn7	-

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match with TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from <TSnCCR1 to TSnCCR5> to TSnCCR1 to TSnCCR5 + 1

Note "–" indicates an unused function in the external event count mode.

In case of D1 > D2 > D3, rewriting of only TSnCCR0 register value, without output from TOSn0 and TOSn1
(TSnOE0 = 0, TSnOE1 = 0, TSnOL0 = 0, TSnOL1 = 1 for TSnIOC0 register
TSnEES1 = 1, TSnEES0 = 1 for TSnIOC register, both edges detected)

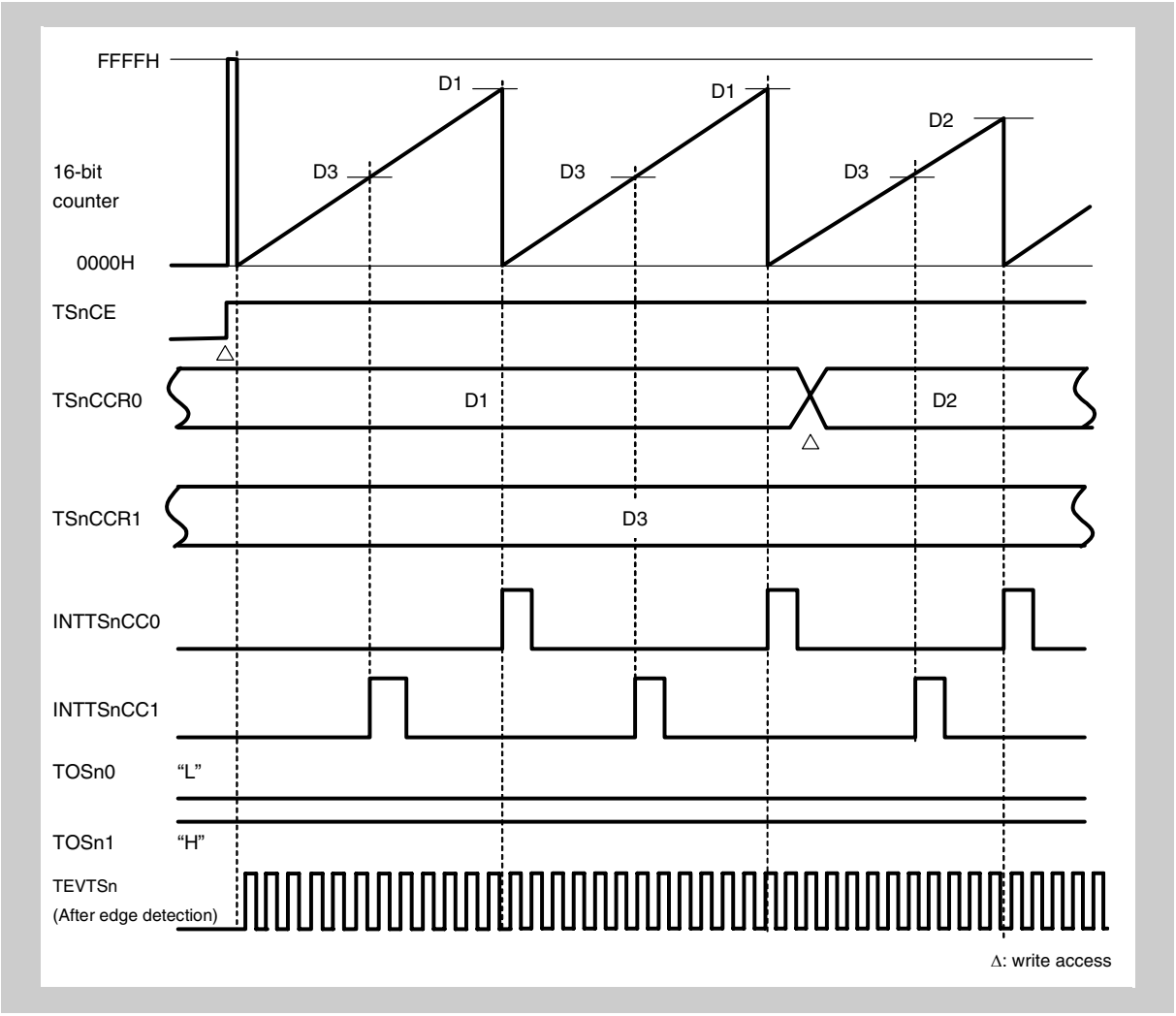


Figure 17-55 Basic operation timing in external event count mode (1/4)

Note
1. After the timer Sn operation starts (TSnCE = 1), the 1st cycle count value will be 1 less than the value after the 2nd cycle.
2. D1, D2: TSnCCR0 register setting values (0000H to FFFFH)
   D3: TSnCCR1 register setting value (0000H to FFFFH)
3. Event count = (Dk + 1)
4. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.
5. k = 1, 2

In case of D1 = D2, TSnCCR0, without rewriting of TSnCCR1 register value, with output from TOSn0 and TOSn1
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0 register)

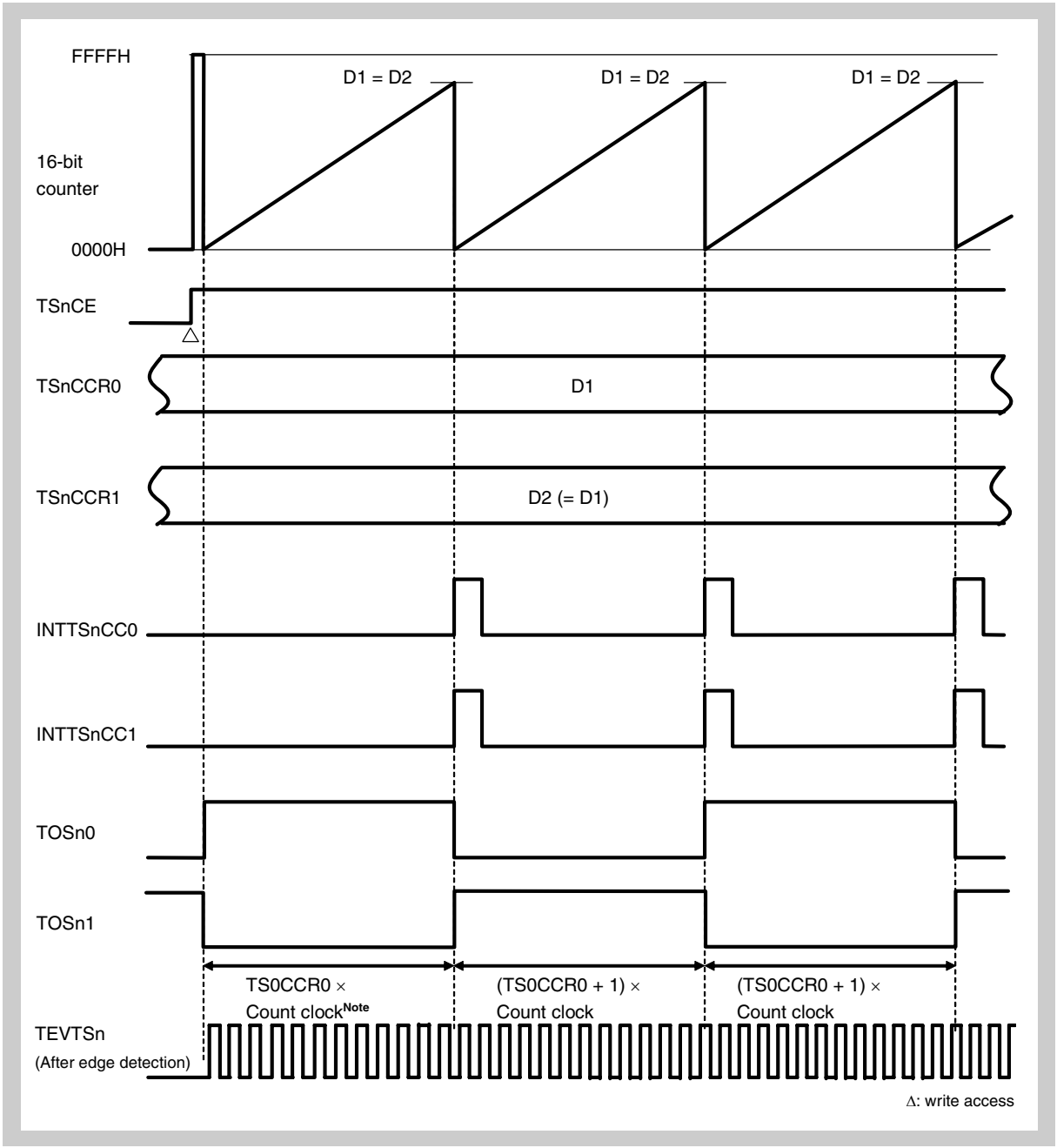


Figure 17-56 Basic operation timing in external event count mode (2/4)

Note 1. After the timer Sn operation starts (TSnCE = 1), the 1st cycle count value will be 1 less than the value after the 2nd cycle.

2. D1: TSnCCR0 register setting value (0000H to FFFFH)
D2: TSnCCR1 register setting value (0000H to FFFFH)

3. Event count = (D1 + 1)

4. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

In case of D1 = D2, TSnCCR0, without rewriting of TSnCCR1 register value, with output from TOSn0 and TOSn1
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0 register)
TSnEES1 = 1, TSnEES0 = 1 for TSnIOC register, both edges detected

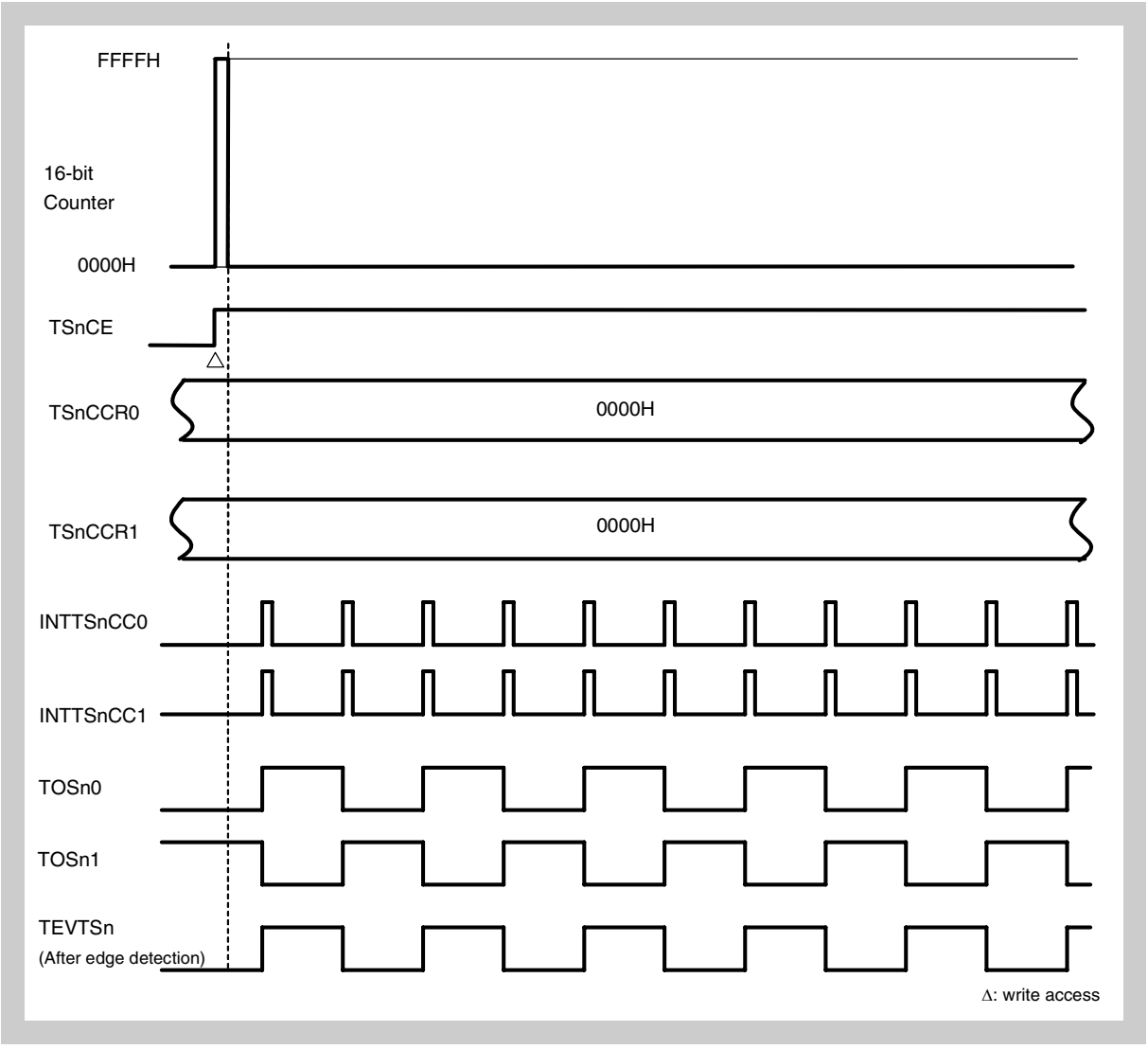


Figure 17-57 Basic operation timing in external event count mode (3/4)

Note 1. Event count = 1

2. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

In case of D1 = D2, without rewriting of TSnCCR0, TSnCCR1 register value, with output from TOSn0 and TOSn1
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, TSnOL1 = 1 for TSnIOC0 register)

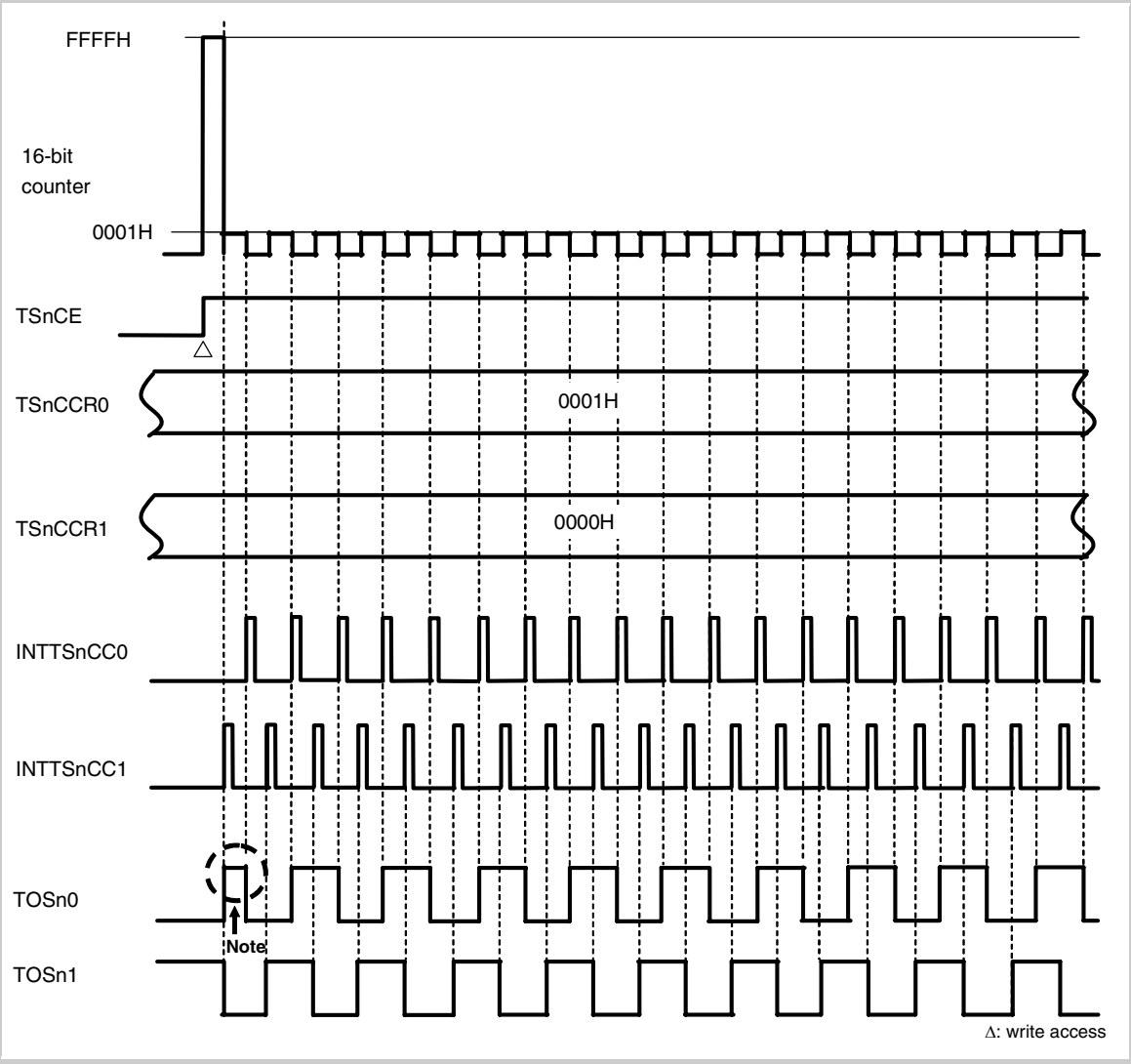


Figure 17-58 Basic operation timing in external event count mode (4/4)

Note
1. After the timer Sn operation starts (TSnCE = 1), the 1st cycle count value will be 1 less than the value after the 2nd cycle.
2. D1: TSnCCR0 register setting value (0001H)
   D2: TSnCCR1 register setting value (0001H)
3. Event count = (Dk + 1)
4. TOSn0 to TOSn5 are toggled when the 16-bit counter is set to 0001H.
5. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.
6. k = 1, 2

(2) Caution on timer output

In the external event count mode, the 16-bit counter is set to 0000H by setting the TSnCE bit to 1, regardless of TEVTSn pin edge detection. Output of the TOSn0 to TOSn5 pins will be active when the value is switched from 0000H to 0001H in the 1st cycle, and when the TSnCCR0 value is switched to 0000H in the 2nd and subsequent cycles. Therefore, the toggle width of the TOSn0 to TOSn5 pins differs between the 1st, 2nd and subsequent cycles.

Differences between 1st cycle and 2nd cycle toggle width of TOSn0 to TOSn5

- TOSn0

  Output is toggled at the TSnCCR0 count in the 1st counter cycle. In the 2nd and subsequent cycles, output is toggled at every TSnCCR0 + 1 count.

- TOSn1 to TOSn5

  – TSnCCR1 to TSnCCR5 = TSnCCR0

  Output is toggled at the TSnCCR0 count in the 1st counter cycle. In the 2nd and subsequent cycles, output is toggled at every TSnCCR0 + 1 count.

  – TSnCCR1 to TSnCCR5 = 0000H

  Output is toggled at every TSnCCR0 + 1 count in all counter cycles.

  – 0000H < TSnCCR1 to TSnCCR5 < TSnCCR0

  Output is toggled at TSnCCR1 to TSnCCR5 counts in the 1st counter cycle. In the 2nd and subsequent cycles, output is toggled at every TSnCCR0 + 1 count.

17.10.3 External trigger pulse output mode

In the external trigger pulse output mode, when a duty is set with the TSnCCR1 to TSnCCR5 registers and a cycle is set with the TSnCCR0 register, and TSnCE is then set to 1, the 16-bit counter remains stopped with FFFFH and waits for external trigger input (TTRGSn pin). Count-up operation starts when a TTRGSn pin valid edge is detected or when the TSnEST bit of the TSnCTL1 register is set to 1. An external trigger pulse is output from the TOSn1 to TOSn5 pins. Toggle output is performed from the TOSn0 pin upon a match with the TSnCCR0 register value. A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during count operation.
The TSnCCR0 to TSnCCR5 registers can be rewritten by reload operation during count operation. Compare register values are reloaded upon a match between the values of the 16-bit counter and the TSnCCR0 register.
If a TTRGSn pin valid edge is detected or if the TSnEST bit of the TSnCTL1 register is set to 1 during operation in the external trigger pulse output mode, the 16-bit counter is cleared and starts the count-up operation again. In this case, if the TOSn1 to TOSn5 pins are inactive, the output from the TOSn1 to TOSn5 pins is active when an external trigger is input. If the TOSn1 to TOSn5 pins are inactive, the output remains active even if an external trigger is input.
In the external trigger pulse output mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

Caution

1. Clear the TSnEEE bit of the TSnCTL1 register to 0 in the external trigger pulse output mode.
2. When the TSnRTE bit of the TSnOPT1 register is set to 1, reloading is executed by write to the TSnCCR1 register. In case of rewriting only the TSnCCR0 register value, the same value must also be written to the TSnCCR1 register. In this case, reloading is not performed if only the TSnCCR0 register is rewritten.

Note The external trigger pulse output mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 1, 0].

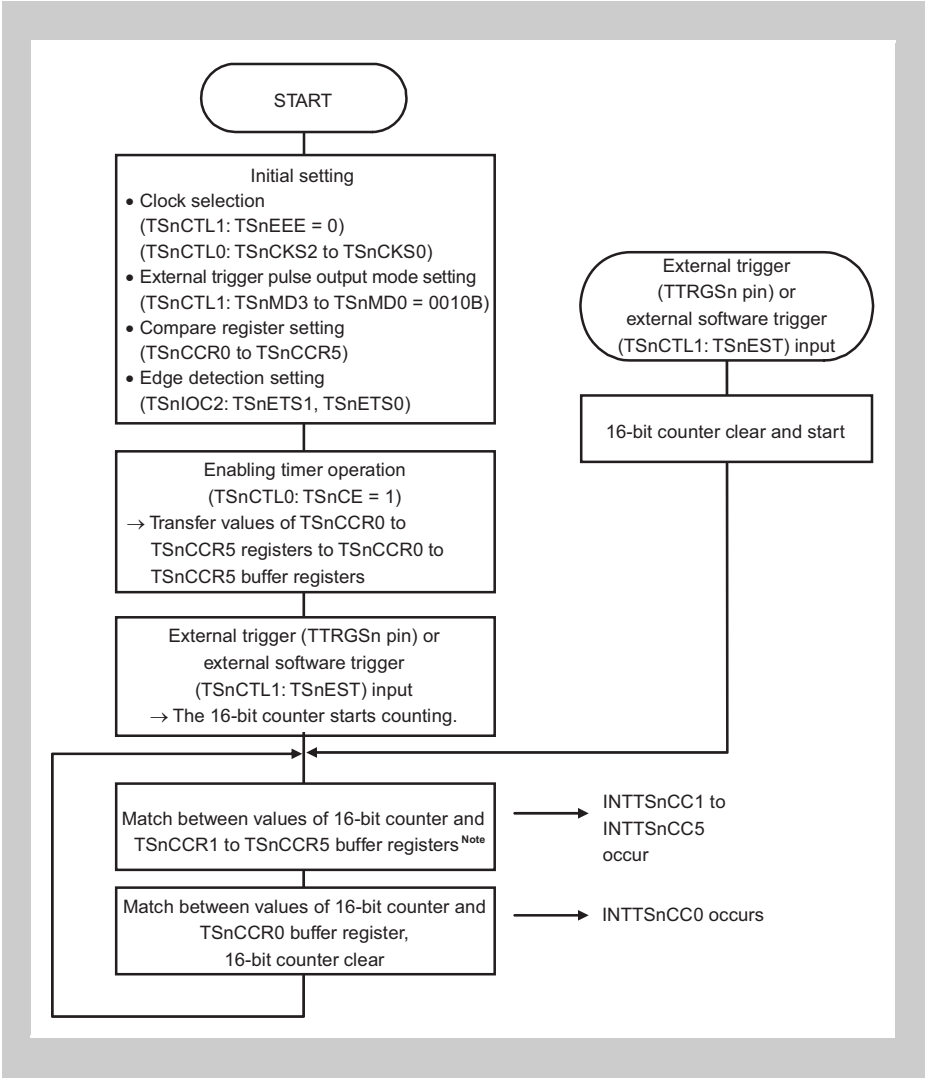


Figure 17-59 Basic operation flow in external trigger pulse output mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

(1) External trigger pulse output mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	Duty
TSnCCR4, TSnCCR5	Reload	Possible	Duty

(b) Input pin

Pin	Function
TTRGSn	16-bit counter clear and start by trigger input
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Toggle output by TSnCCR0 register compare match or external trigger input
TOSn1 to TOSn5	External trigger pulse waveform output
TOSn6, TOSn7	-

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match with TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After match detection between 16-bit counter and TSnCCR1 to TSnCCR5 values

Note "–" indicates an unused function in the external trigger pulse output mode.

In case of rewriting of TSnCCR0, TSnCCR1 register values, with output from TOSn0 and TOSn1
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

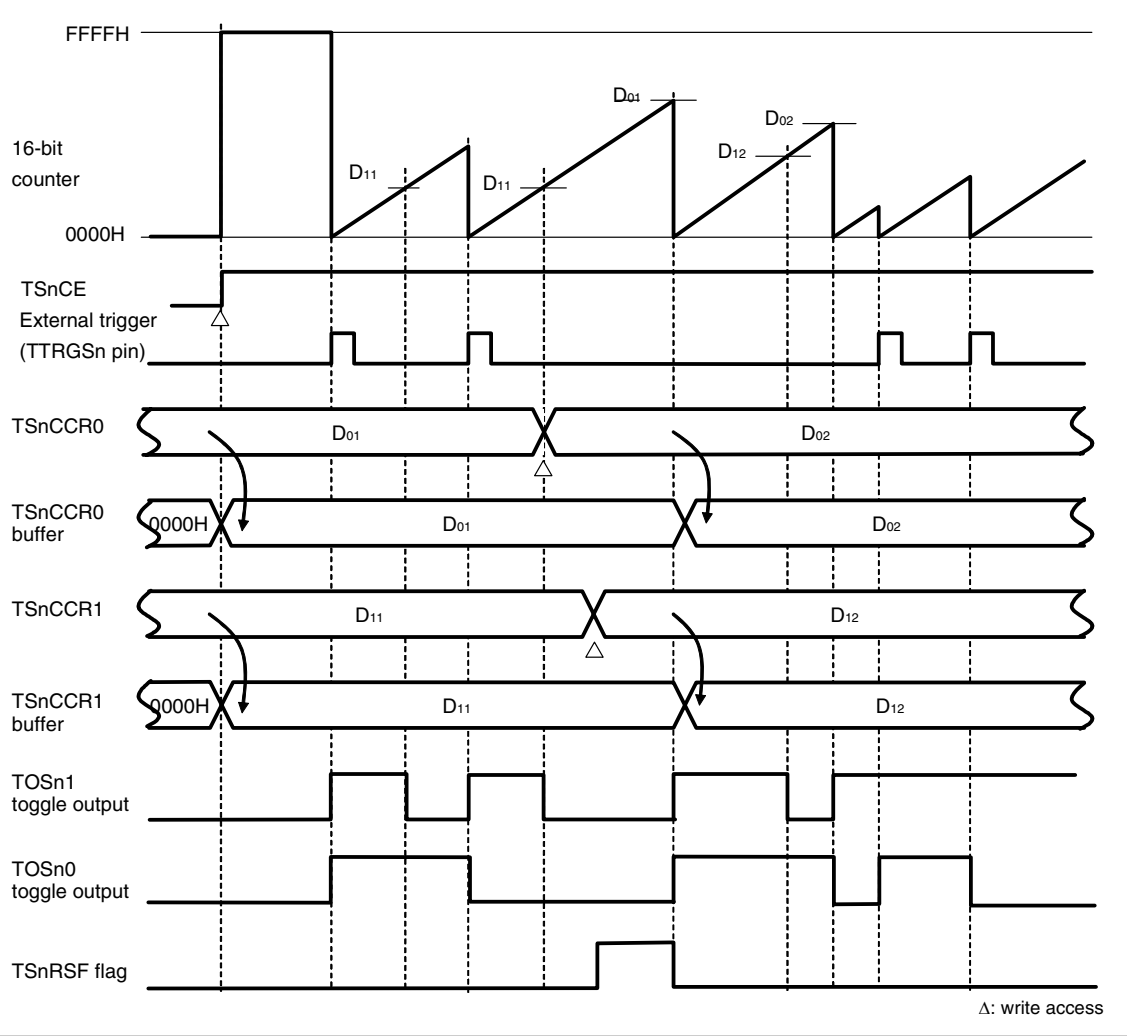


Figure 17-60 Basic operation timing in external trigger pulse output mode

Note

1. D_{01}, D_{02}: TSnCCR0 register setting values (0000H to FFFFH)
   D_{11}, D_{12}: TSnCCR1 register setting values (0000H to FFFFH)
2. TOSn1 (PWM) Duty = (TSnCCR1 register setting value) × (count clock cycle)
   Cycle = (TSnCCR0 register setting value + 1) × (count clock cycle)
3. The TOSn0 pin is toggled immediately after the count start and when the 16-bit counter is cleared.
4. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

17.10.4 One-shot pulse mode

In the one-shot pulse mode, a cycle is set with the TSnCCR0 register and the output compare value is set with the TSnCCR1 to TSnCCR5 registers. When the TSnCE bit of the TSnCTL0 register is set to 1, the 16-bit counter remains stopped with FFFFH and waits for external trigger input (TTRGSn pin).
Count-up operation starts when the valid edge of an external trigger input (TTRGSn pin) is detected or when the TSnEST bit of the TSnCTL0 register is set to 1.
The TOSn1 to TOSn5 pins are active upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 registers. The TOSn1 to TOSn5 pins are inactive upon a match between the values of the 16-bit counter and the TSnCCR0 register, and the 16-bit counter is cleared to 0000H and then stops.
TOSn0 provides toggle output upon a match between the values of the 16-bit counter and the TSnCCR0 buffer register during count operation. A compare match interrupt (INTTSnCC0) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 register, and a compare match interrupt (INTTSnCC1 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers during count operation.
The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer Sn operation (TSnCE bit value of TSnCTL0 register).
Trigger input is ignored during 16-bit counter operation. The second trigger must be input while the 16-bit counter is stopped with 0000H.
In the one-shot pulse mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

Caution Clear the TSnEEE bit of the TSnCTL1 register to 0 in the one-shot pulse mode.

Note The one-shot pulse mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 1, 1].

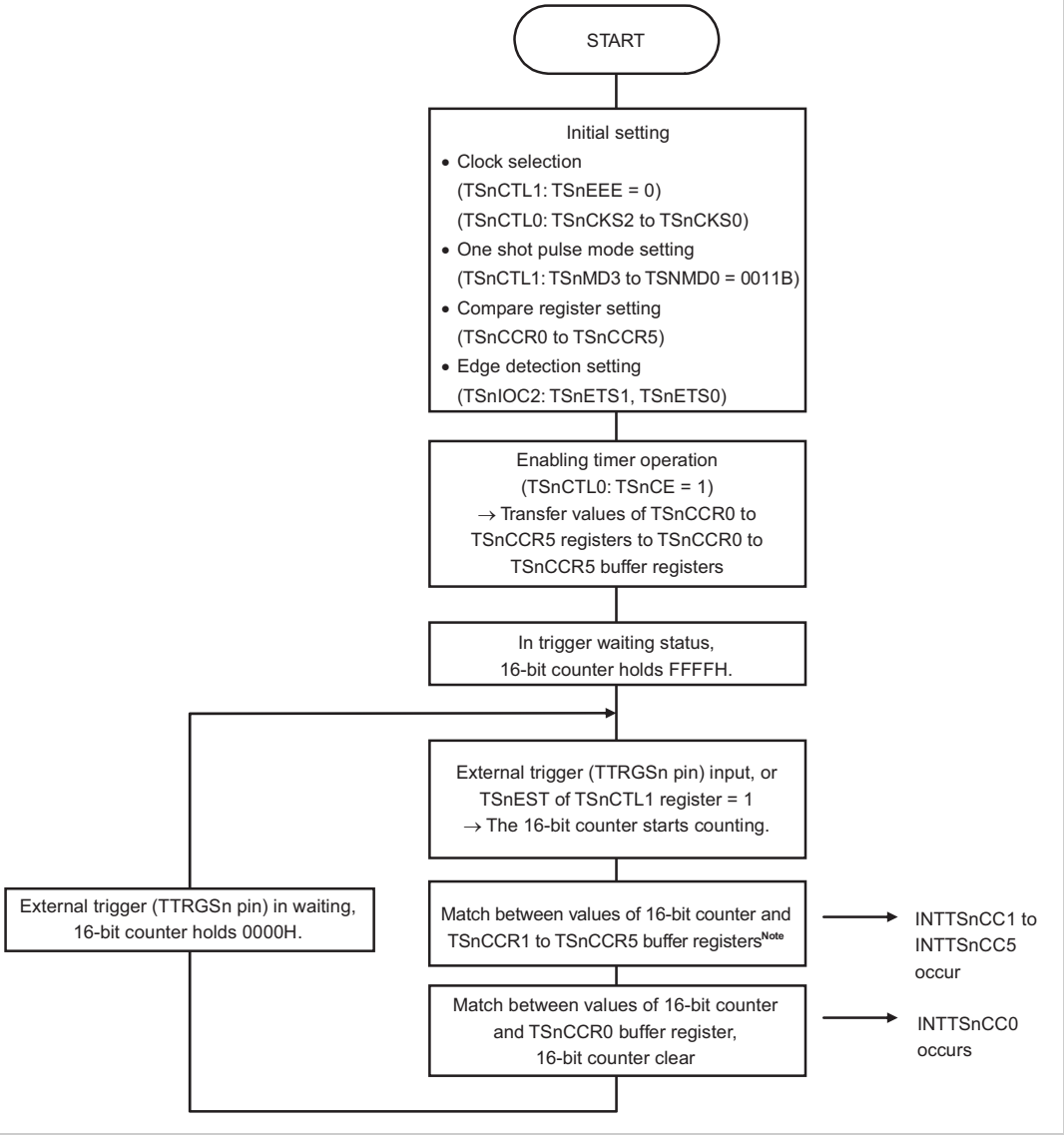


Figure 17-61 Basic operation flow in one-shot pulse mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

Caution The 16-bit counter is not be cleared and trigger input is ignored even if the trigger is input during count-up operation by the 16-bit counter.

(1) One-shot pulse mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting During Operation	Function
TSnCCR0	Anytime rewrite	Possible	Cycle
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Output delay value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Output delay value

(b) Input pin

Pin	Function
TTRGSn	16-bit counter start by trigger input
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Active at count start, inactive upon match of TSnCCR0 register
TOSn1 to TOSn5	Active upon match of TSnCCR1 to TSnCCR5 registers, inactive upon match of TSnCCR0 register
TOSn6, TOSn7	-

(d) Interrupt

Interrupt	Function
INTTSnCC1 to INTTSnCC5	Compare match of TSnCCR1 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between values of 16-bit counter and TSnCCR1 to TSnCCR5 registers

Note 1. "–" indicates an unused function in the one-shot pulse mode.

2. m = 1 to 5

(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 1, and TSnOL1 = 1 for TSnIOC0 register,
TSnEES1 = 0 and TSnEES0 = 1 for TSnIOC2 register, rising edge detected)

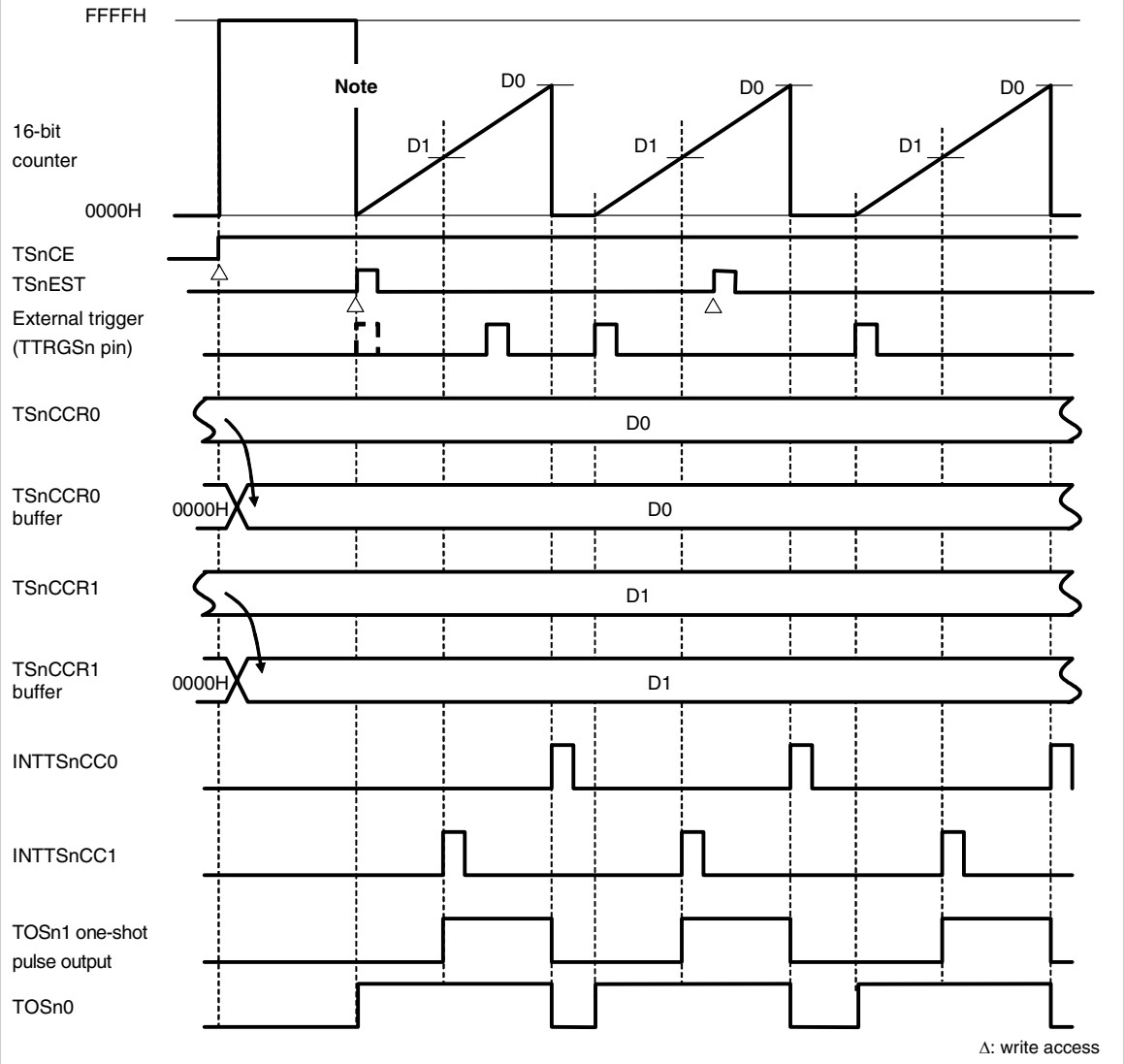


Figure 17-62 Basic operation timing in one-shot pulse mode

Note 1. The 16-bit counter starts count-up operation when the TSnEST bit of the TSnCTL1 register is set to 1 or when a pulse is input from the TTRGSn pin.

2. D0: TSnCCR0 register setting value (0000H to FFFFH)
D1: TSnCCR1 register setting value (0000H to FFFFH)

3. TOSn1 (output delay) = (TSnCCR1 register setting value) × (count clock cycle)
TOSn1 (output pulse width) = {(TSnCCR0 register setting value + 1) − (TSnCCR1 register setting value)} × (count clock cycle)

4. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

17.10.5 PWM mode

In the PWM mode, when a duty is set with the TSnCCR1 to TSnCCR5 registers and a cycle is set with the TSnCCR0 register, and the TSnCE bit of the TSnCTL0 register is then set to 1, duty variable type PWM is output from the TOSn1 to TOSn5 pins.
The TOSn1 to TOSn5 pins are active when count-up operation starts, and inactive upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 registers. Then, the TOSn1 to TOSn5 pins are active upon a match with the TSnCCR0 register value. The TOSn0 pin provides toggle output upon a match with the TSnCCR0 buffer register.
A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during count operation.
The TSnCCR0 to TSnCCR5 registers can be rewritten during count operation. Compare register values are reloaded upon a match between the 16-bit counter and TSnCCR0 register values.
The TSnCCR0 to TSnCCR5 registers can be rewritten with the reload write method, regardless of the timer Sn operation (TSnCE bit value of TSnCTL0 register).
In the PWM mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

Caution When the TSnRTE bit of the TSnOPT1 register is set to 1, reloading is executed by write to the TSnCCR1 register. In case of rewriting only the TSnCCR0 register value, the same value must also be written to the TSnCCR1 register. In this case, reloading is not performed if only the TSnCCR0 register is rewritten.

Note The PWM mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 1, 0, 0].

In case of TSnCCR0, TSnCCR1 to TSnCCR5 values not rewritten during timer operation

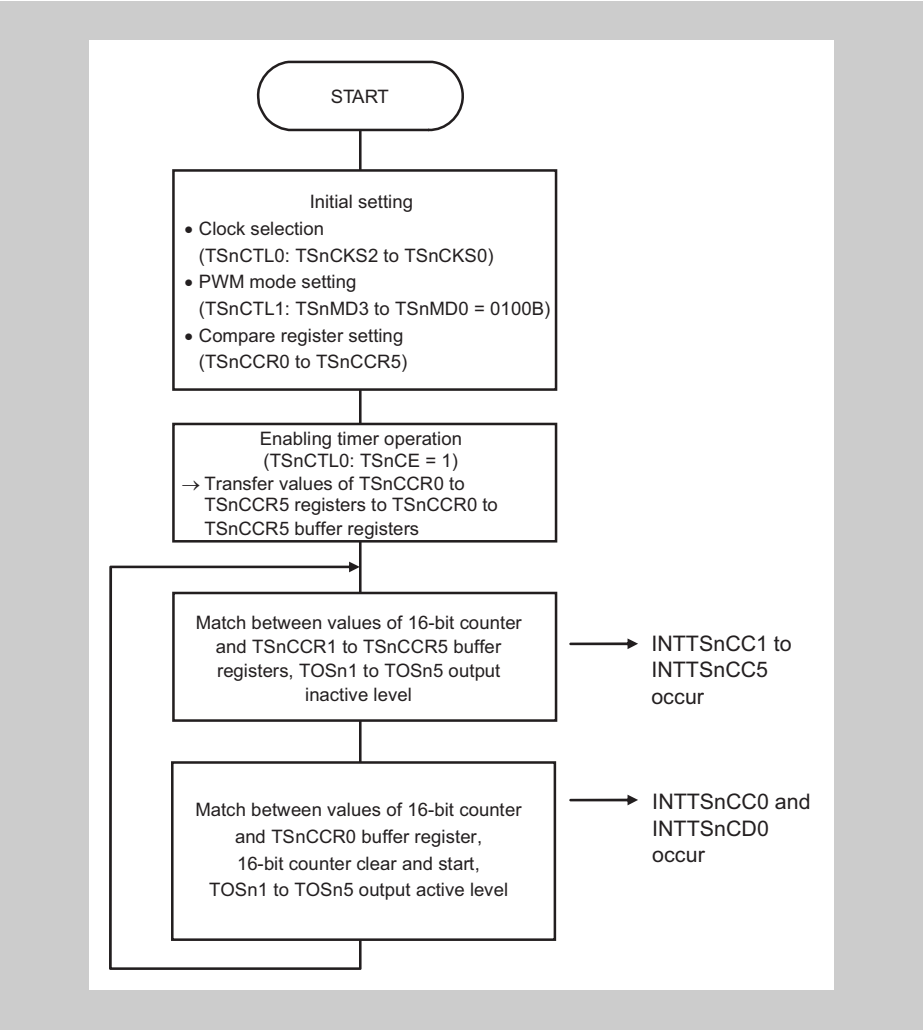


Figure 17-63 Basic operation flow in PWM mode (1/2)

In case of TSnCCR0, TSnCCR1 to TSnCCR5 values rewritten during timer operation

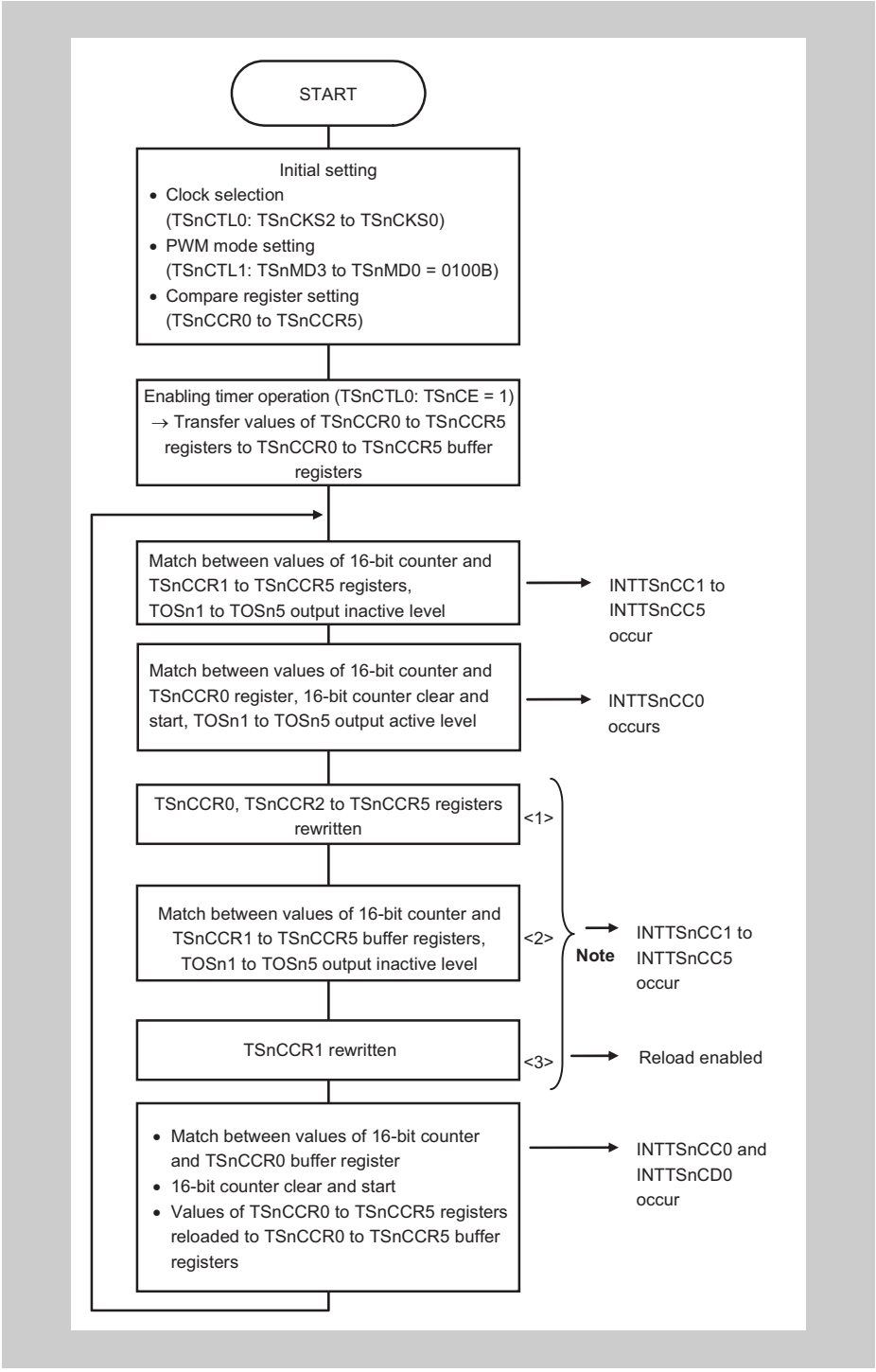


Figure 17-64 Basic operation flow in PWM mode (2/2)

Note Although the timing of <2> may differ according to the TSnCCR register value or the timing of rewriting <1> and <3>, be sure to perform <3> last.

(1) PWM mode operation list

(a) Register rewriting

Register	Rewriting Method	Rewriting During Operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	Duty
TSnCCR4, TSnCCR5	Reload	Possible	Duty
TSnDTC0, TSnDTC1	Reload	Conditionally possible[Note]	Cycle, dead time

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Toggle output by compare match of TSnCCR0 register
TOSn1 to TOSn5	PWM output by compare match of TSnCCR1 to TSnCCR5 registers
TOSn6	-
TOSn7	Pulse output by A/D conversion trigger

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (simultaneous occurrence with INTTSnCC0)
INTTSnWN	-

(e) Compare match timing

Compare Match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between values of 16-bit counter and TSnCCR1 to TSnCCR5 registers

Note Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* for details.

(f) Output condition

Output	Condition
Duty 0% output	TSnCCRm = 0000H
Duty 100% output	TSnCCRm = TSnCCR0 + 1
Dead time check	TSnDSE of TSnOPT0 register = 1

Note 1. "–" indicates an unused function in the PWM mode.

2. m = 1 to 5

Only TSnCCR1 value rewritten, with output from TOSn0 and TOSn1 (TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

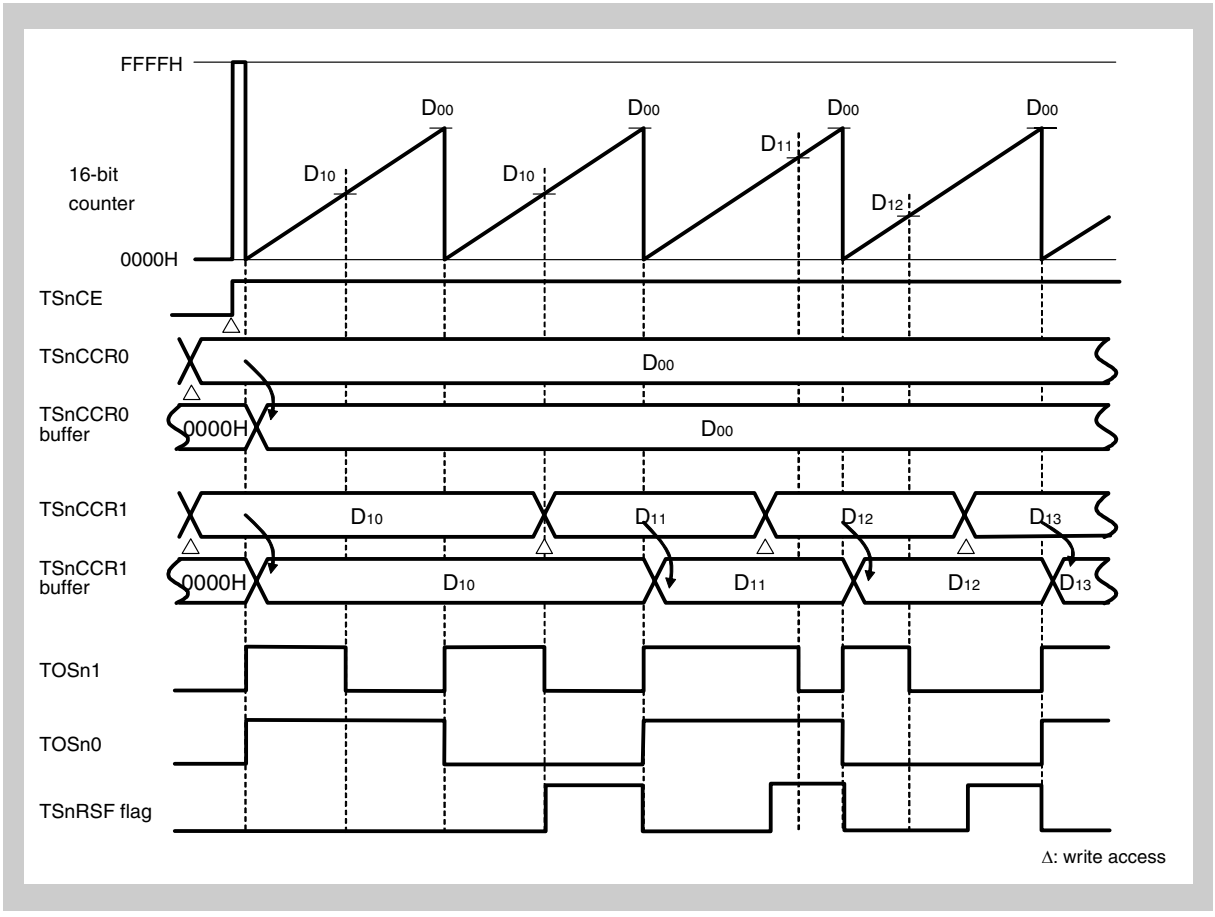


Figure 17-65 Basic operation timing in PWM mode (1/2)

Note 1. D_{00}: TSnCCR0 register setting value (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: TSnCCR1 register setting values (0000H to FFFFH)

2. TOSn1 (PWM) duty = (TSnCCR1 register setting value) × (count clock cycle)
TOSn1 (PWM) cycle = (TSnCCR0 register setting value + 1) × (count clock cycle)

3. TOSn0 is toggled immediately after count start and with (TSnCCR0 register setting value + 1) × (count clock cycle)

4. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

Only TSnCCR1 value rewritten, with output from TOSn0 and TOSn1 (TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

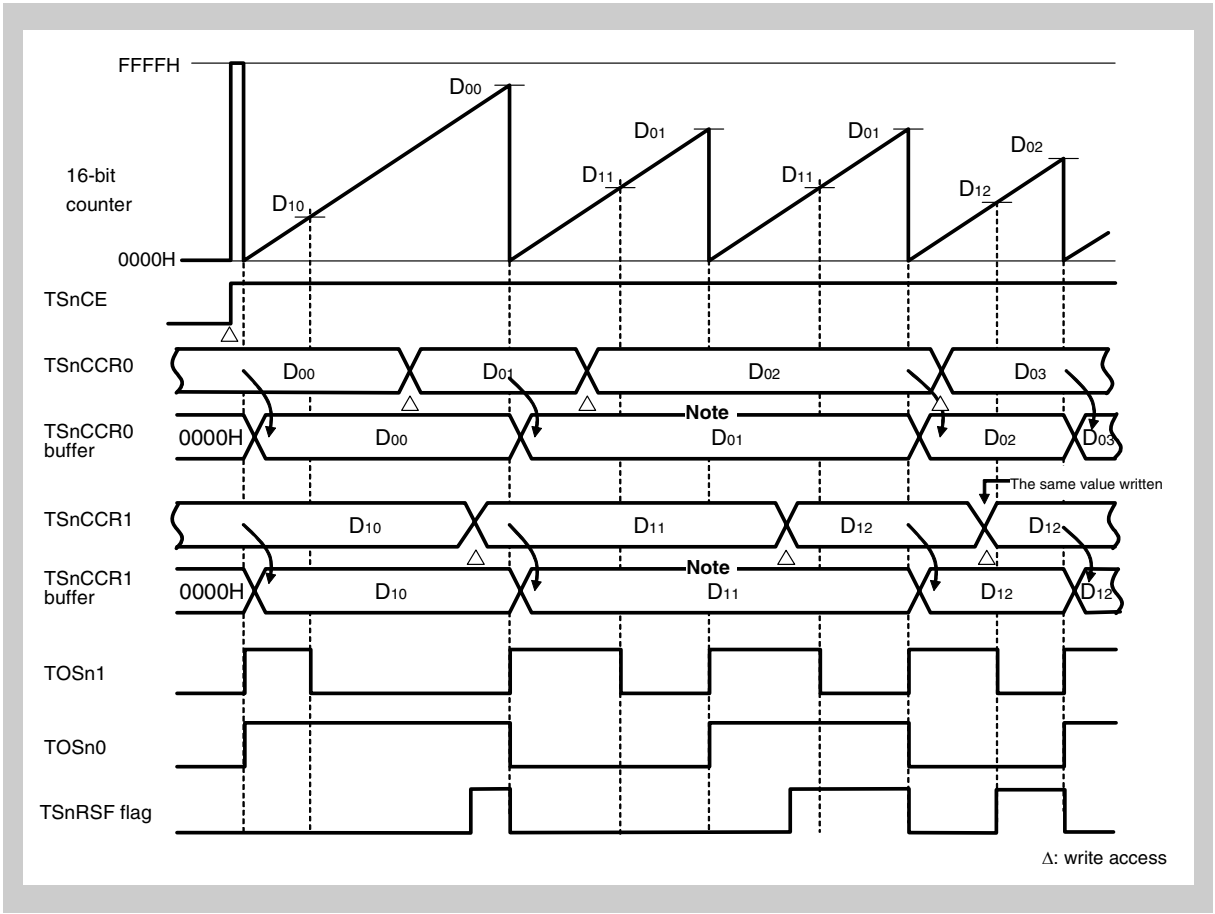


Figure 17-66 Basic operation timing in PWM mode (2/2)

Note

1. Since the TSnCCR1 register has not been written, this value is not reloaded to the TSnCCR0 buffer register.
2. D_{00}, D_{01}, D_{02}, D_{03}: TSnCCR0 register setting values (0000H to FFFFH)
   D_{10}, D_{11}, D_{12}, D_{13}: TSnCCR1 register setting values (0000H to FFFFH)
3. The TOSn0 and TOSn1 pins become active level when the timer starts counting.
4. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

(2) Reload/interrupt thinning out functions in PWM mode

The reload thinning out function and interrupt thinning out function are enabled by setting TSnRTE = 1, TSnICE = 1, and TSnRDE = 1 for the TSnOPT1 register, and TSnID4 to TSnID0.
The interrupt thinning out function is enabled by setting TSnRTE = 1, TSnRDE = 0, and TSnID4 to TSnID0.

(3) Dead time control in PWM mode

In the PWM mode, dead time control is enabled by setting the dead time value to the TSnDTC0 and TSnDTC1 registers and setting TSnDSE of the TSnOPT0 register to 1. The dead time can be controlled at the timing of switching the TOSn1, TOSn2, TOSn3 and TOSn4 pins.

Table 17-18 Dead time in PWM mode

Switch timing	Dead time
TOSn1: High level to low level TOSn2: Low level to high level	TSnDTC1 register value
TOSn2: High level to low level TOSn1: Low level to high level	TSnDTC0 register value
TOSn3: High level to low level TOSn4: Low level to high level	TSnDTC1 register value
TOSn4: High level to low level TOSn3: Low level to high level	TSnDTC0 register value

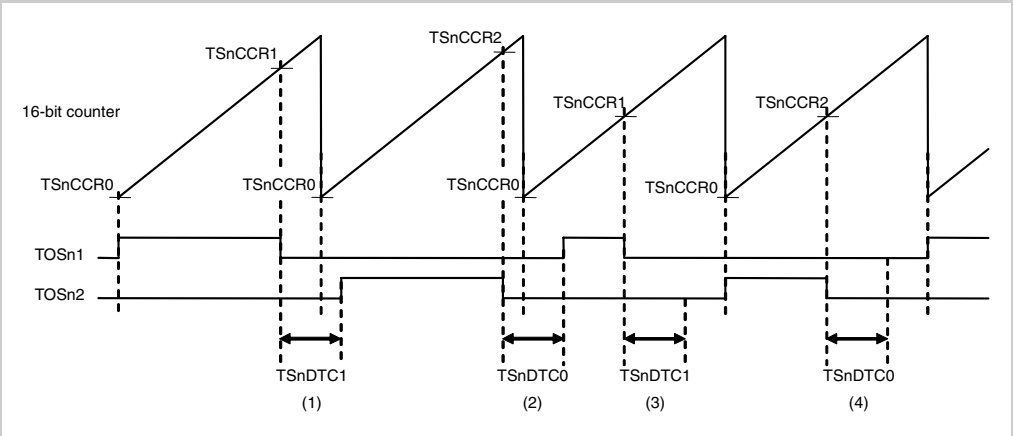


Figure 17-67 Dead time control between TOSn1 and TOSn2 pins (1/2)

The dead time counter starts at the falling edge of TOSn1 during the period of (1). Then, TOSn2 normally becomes active when the 16-bit counter becomes 0000H, but since the dead time counter is operating, it becomes active when the dead time count operation is finished.
The dead time counter starts at the falling edge of TOSn2 during the period of (2). Then, TOSn1 normally becomes active when the 16-bit counter becomes 0000H, but since the dead time counter is operating, it becomes active when the dead time count operation is finished.
The dead time counter starts at the falling edge of TOSn1 during the period of (3). Then, TOSn2 becomes active when the 16-bit counter becomes 0000H

after the dead time counter finishes the count operation.
The dead time counter starts at the falling edge of TOSn2 during the period of (4). Then, TOSn1 becomes active when the 16-bit counter becomes 0000H after the dead time counter finishes the count operation.

Note 1. The active level of TOSn1 and TOSn2 is high level.

2. The active level of TOSn3 and TOSn4 is high level.

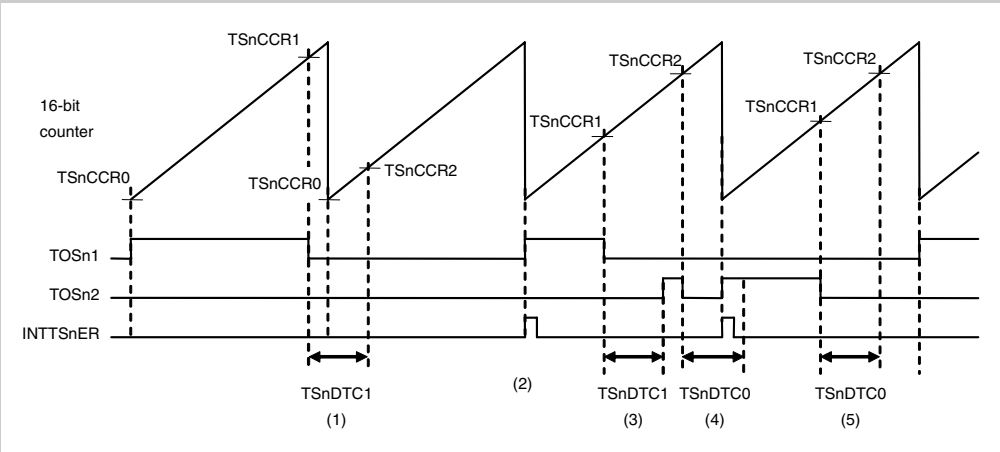


Figure 17-68 Dead time control between TOSn1 and TOSn2 pins (2/2)

The dead time counter starts at the falling edge of TOSn1 during the period of (1). Then, TOSn2 normally becomes active when the 16-bit counter becomes 0000H, but since the dead time counter is operating, it becomes active when the dead time count operation is finished. In this case, however, TOSn2 becomes inactive upon a compare match, because a TSnCCR2 compare match occurs before the dead time counter operation finishes.

TSnCCR1 + TSnDTC1 ≥ TSnCCR0 + TSnCCR2 (TOSn2 remains inactive).
TSnCCR2 + TSnDTC0 ≥ TSnCCR0 + TS0CCR1 (TOSn1 remains inactive).

TOSn1 becomes active during the period of (2) because dead time control is not in operation. In this case, however, an INTTSnER interrupt occurs because TSnCCR1 and TSnCCR2 are set so that TOSn1 and TOSn2 rise simultaneously.
The dead time counter starts at the falling edge of TOSn1 during the period of (3). TOSn2 becomes active when the dead time counter operation is finished. TOSn2 becomes inactive upon a match of TSnCCR2.
The dead time counter starts at the falling edge of TOSn2 during the period of (4). Then, the 16-bit counter is cleared and the next PWM cycle is output. TOSn2 becomes active because TOSn1 is in the dead time period. In this case, however, an INTTSnER interrupt occurs because TSnCCR1 and TSnCCR2 are set so that TOSn1 and TOSn2 rise simultaneously.
The dead time counter starts at the falling edge of TOSn2 during the period of (5). TOSn1 does not become active even after the dead time counter operation is finished, because TSnCCR1 is smaller than TOSn2 and is already inactive.

Note 1. The active level of TOSn1 and TOSn2 is high level.

2. The active level of TOSn3 and TOSn4 is high level.

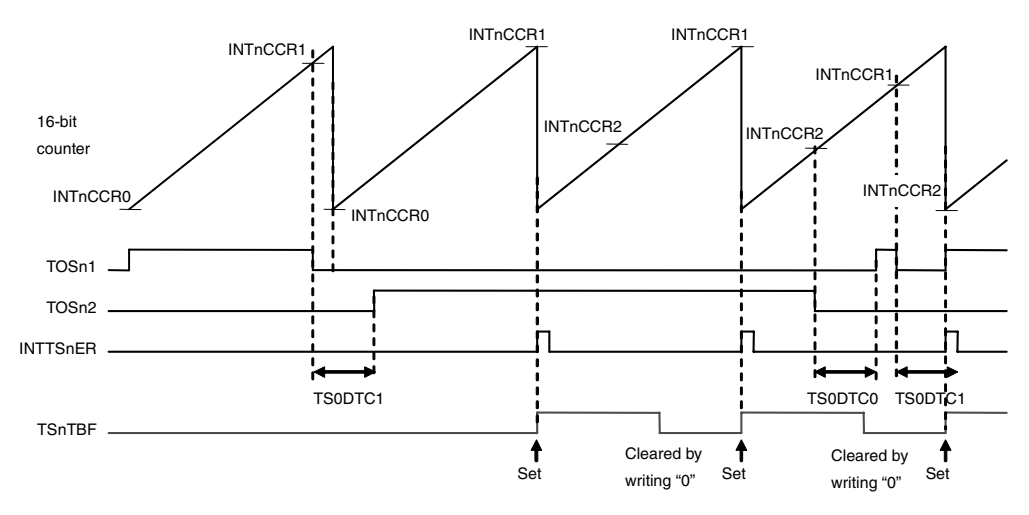


Figure 17-69 Duty 100% output

As shown in *Figure 17-69 on page 856*, when TOSn2 is set to make duty 100% output (TSnCCR2 ≥ TSnCCR0 + 1), the TOSn1 output is fixed at the low level. This means that the control masks the TOSn1 active condition, because TOSn2 is already active when TOSn1 becomes active. In this case, an INTTSnER interrupt occurs because of the setting to make TOSn1 and TOSn2 high level simultaneously.

Note 1. The active level of TOSn1 and TOSn2 is high level.

2. The active level of TOSn3 and TOSn4 is high level.

(4) Dead time rewriting by PWM mode operation

In the PWM mode, timer Sn dead time setting registers (TSnDTC0, TSnDTC1) can be rewritten during count operation. The new setting for the dead time becomes valid when the value is reloaded. The value cannot be changed with the anytime rewrite method.
Change of the dead time will be valid at the culled reload timing when the reload thinning out function (TS0RDE bit of TSnOPT1 register = 1) is used in the PWM mode.
TS0CCR1 must be written while the TS0RDE bit of the TSnOPT1 register is 1 in order to enable the reload timing.

(5) Other timer outputs in PWM mode

(a) TOSn5 pin

PWM output can be performed in the same manner as the TOSn1 to TOSn4 pins by setting the TSnCCR5 register.
The dead time cannot be set.

(b) TOSn0 and TOSn6 pins

The TOSn0 pin repeats toggle operation in every PWM cycle.
The TOSn6 pin outputs the inactive level.

(c) INTTSnER interrupt

The INTTSnER interrupt occurrence conditions vary depending on the TSnDSE bit setting in the TSnOPT0 register. TSnTBF is set to 1 when an interrupt occurs.

TSnDSE of TSnOPT0 register	TSnTBF of TSnOPT6 register	INTTSnER
0 (Dead time control disabled)	Set when timer outputs become active simultaneously	An error interrupt is output while timer outputs become active simultaneously and output active simultaneously.
1 (Dead time control enabled)	Set when timer outputs become active simultaneously	An error interrupt is output when a condition that sets timer outputs simultaneously occurs.

When the TSnDSE bit of the TSnOPT0 register is 1, or the TSnTBA0 and TSnTBA1 bits of the TSnIOC4 register are 1, occurrence of a condition that sets positive/negative phase simultaneously for the TOSn1 and TOSn3 pins and the TOSn2 and TOSn4 pins is the error interrupt occurrence condition. Therefore, an error interrupt occurs when a positive/negative phase simultaneous setting condition occurs even when TSnOE1 and TSnOE2 of the TSnIOC0 register are set to 0 and the TOSn1 and TOSn2 pins are fixed at the inactive level. Moreover, the positive/negative phase simultaneous active detection flag is also set.
Clear TSnTBA0 and TSnTBA1 to 0 to prevent the above situation.

17.10.6 Free-running mode

In the free-running mode, the 16-bit counter counts from 0000H to FFFFH when the TSnCE bit of the TSnCTL0 register is set to 1. When the 16-bit counter overflows from FFFFH to 0000H, the overflow flag (TSnOVF) is set to 1, an overflow interrupt (INTTSnOV) occurs, and the count is then cleared. Count operation in the free-running mode continues until TSnCE is cleared to 0. A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during the count operation.
The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer Sn operation (TSnCE bit value).
TOSn0 to TOSn5 become active immediately after the operation starts and perform toggle output when a match interrupt with the compare register occurs.

Caution The overflow flag (TSnOVF) is not cleared only by being read, but cleared when 0 is written or the operation stops (TSnCE = 0).

Note The free-running mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 1, 0, 1].

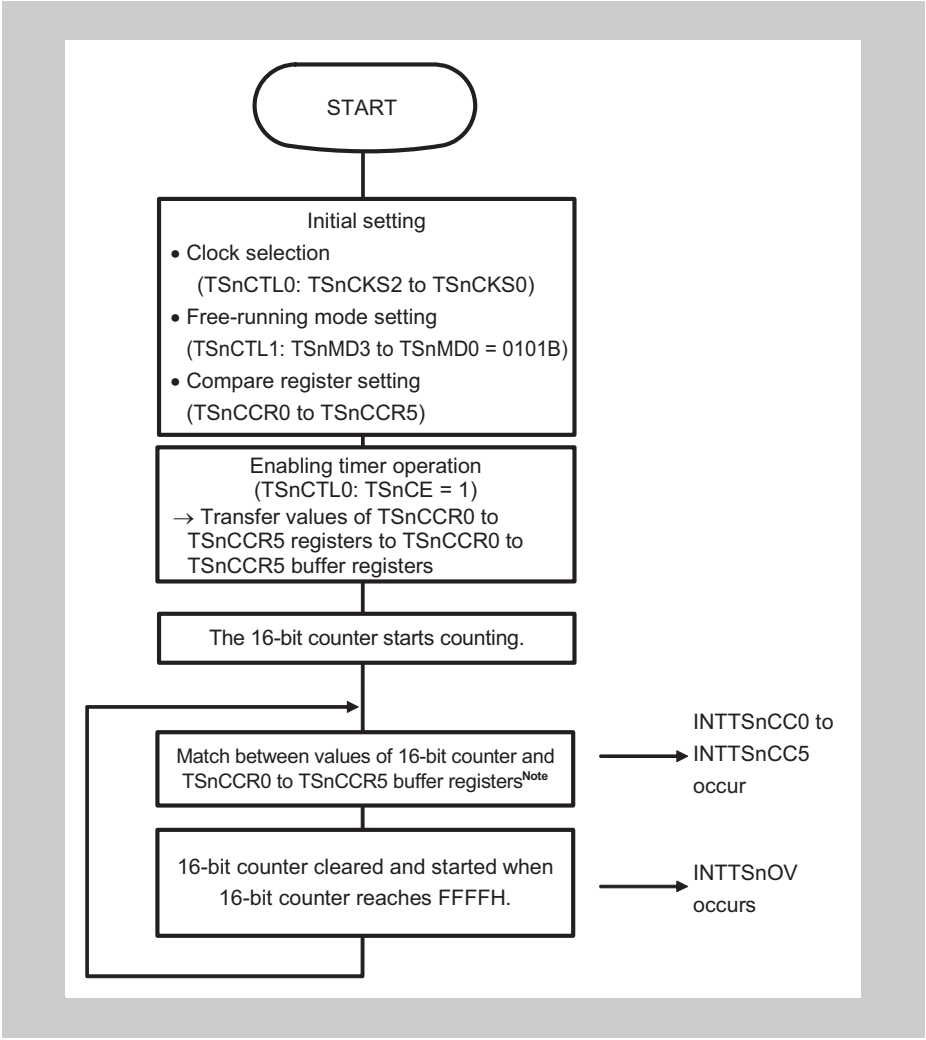


Figure 17-70 Basic operation flow in free-running mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 buffer registers.

(1) Free-running mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Anytime rewrite	Possible	Compare value
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Compare value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pin

PIn	Function
TOSn0 to TOSn5	Toggle output by compare match of TSnCCR0 to TSnCCR5 registers
TOSn6, TOSn7	-

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 register
INTTSnOV	Overflow
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

Note “–” indicates an unused function in the free-running mode.

TSnCCR0, TSnCCR1 register values rewritten, with output from TOSn0 and TOSn1

(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

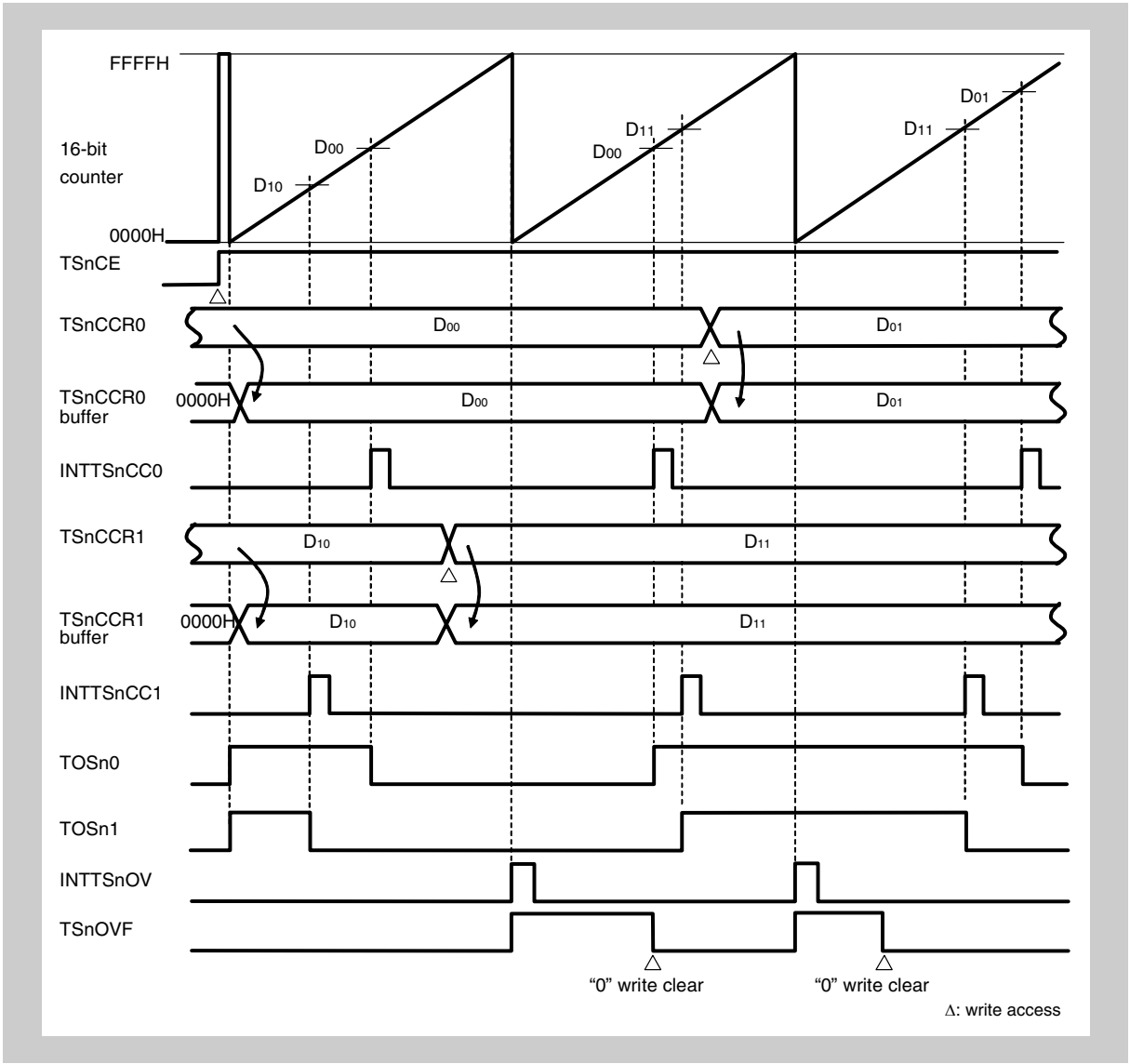


Figure 17-71 Basic operation flow in free-running mode

Note 1. D_{00}, D_{01}: TSnCCR0 register setting values (0000H to FFFFH)
D_{10}, D_{11}: TSnCCR1 register setting values (0000H to FFFFH)

2. The TOSn0 and TOSn1 pins become active level when the timer starts counting.

3. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

17.10.7 Triangular-wave PWM mode

In the triangular-wave PWM mode, when a duty is set with the TSnCCR1 to TSnCCR5 registers, a cycle is set with the TSnCCR0 register and the TSnCE bit of the TSnCTL0 register is then set to 1, duty variable type triangular-wave PWM output is provided from the TOSn1 to TOSn5 pins. TOSn0 performs toggle output upon a match of the TSnCCR0 buffer register and when a 16-bit counter underflow occurs. A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during count operation. Also, a trough interrupt (INTTSnOD) occurs when a 16-bit counter underflow occurs. The TSnCCR0 to TSnCCR5 registers can be rewritten during count operation. Compare register values are reloaded when a 16-bit counter underflow occurs.

Writing to the TSnCCR1 register enables the next reload timing.
In the triangular-wave PWM mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

Note

1. Set a value of $0 \leq$ TSnCCR0 $\leq$ FFFEH to the TSnCCR0 register in the triangular-wave PWM mode.
2. Set the TSnRBE bit of the TSnOPT1 register to 1 in the triangular-wave PWM mode. Reload does not occur by clearing TSnRBE to 0.
3. The triangular-wave PWM mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 1, 1, 1].

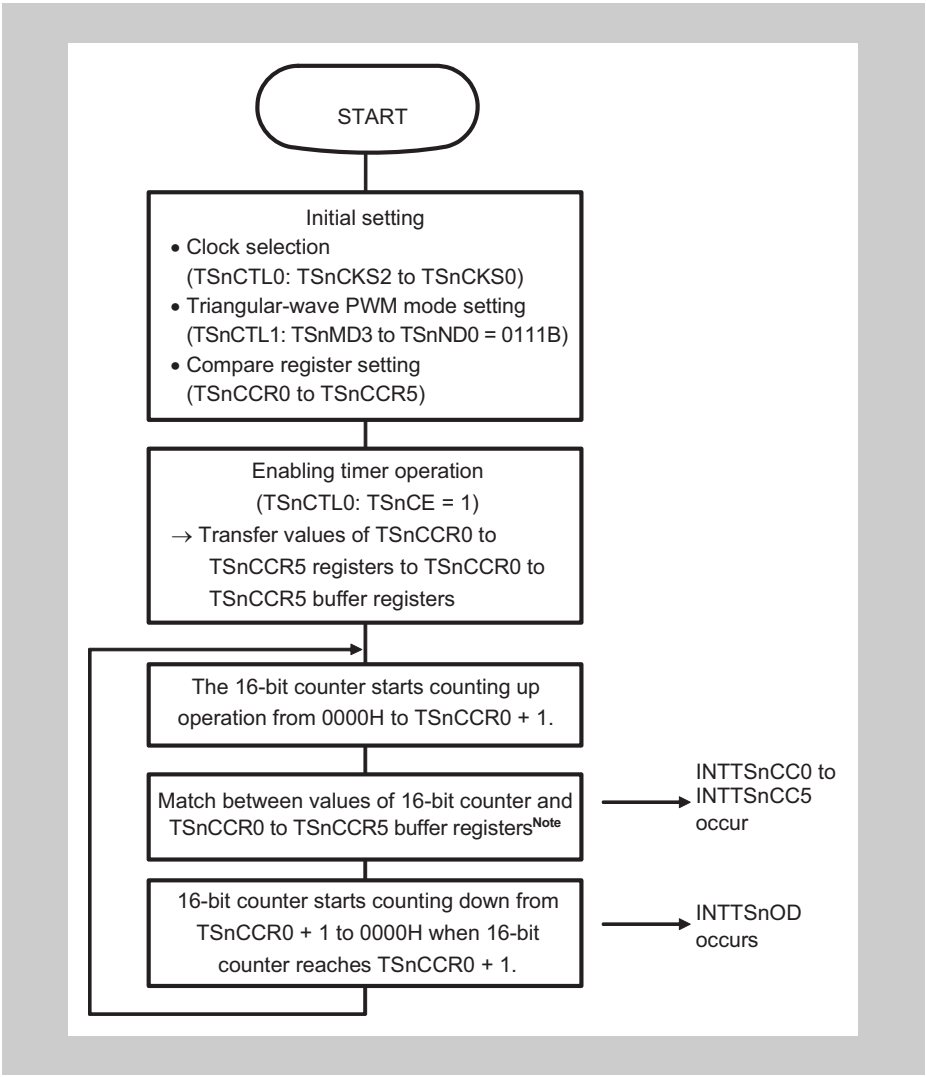


Figure 17-72 Basic operation flow in triangular-wave PWM mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 buffer registers.

(1) Triangular-wave PWM mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	1/2 of cycle
TSnCCR1 to TSnCCR3	Reload	Possible	1/2 of duty
TSnCCR4, TSnCCR5	Reload	Possible	1/2 of duty

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Inactive during 16-bit counter count-up operation Active during count-down operation.
TOSn1 to TOSn5	PWM output by compare match of TSnCCR1 to TSnCCR5 registers
TOSn6	-
TOSn7	Pulse output by A/D conversion trigger

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	Trough interrupt
INTTSnCD0	-
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 toTSnCCR0 − 1
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

(f) Output condition

Output	Condition
Duty 0% output	TSnCCRm = TSnCCR0 + 1
Duty 100% output	TSnCCRm = 0000H
Toggle at peak/trough	TSnCCRm ≥ TSnCCR0 + 2

Note 1. “–” indicates an unused function in the triangular-wave PWM mode.

2. m = 1 to 5

With output from TOSn0 and TOSn1
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

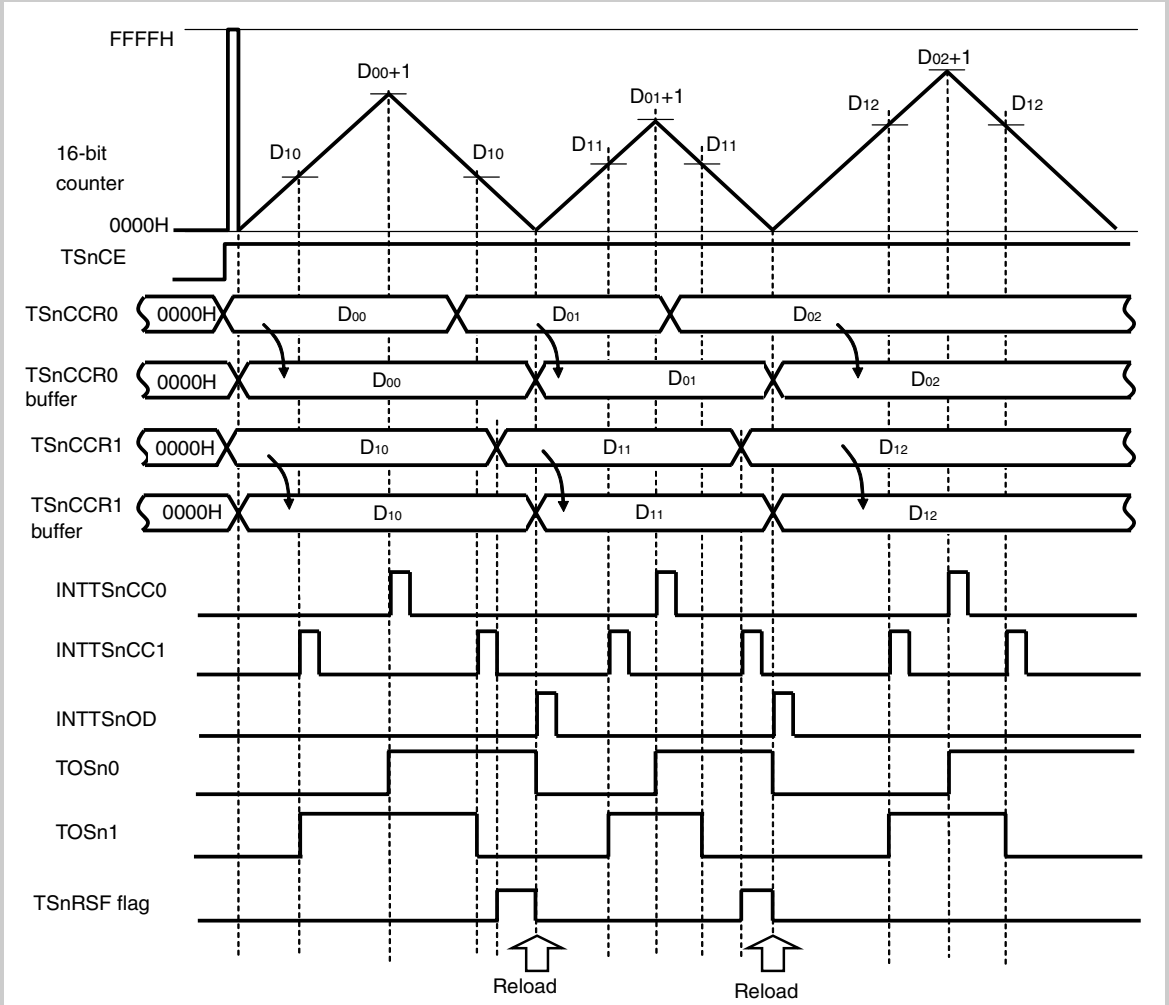


Figure 17-73 Basic operation timing in triangular-wave PWM mode

Note 1. D_{00}, D_{01}, D_{02}: Triangular-wave cycle (TSnCCR0 + 1) × 2
D_{10}, D_{11}, D_{12}: PWM duty (TSnCCR0 − TCnCCR1 + 1) × 2
Count-up: 0000H to D_{00}
Count-down: (D_{00}– 0001H

2. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

(2) Reload/interrupt thinning out function in triangular-wave PWM mode

The reload thinning out function and interrupt thinning out function are enabled by setting TSnRTE = 1, TSnICE = 1, and TSnRDE = 1 in the TSnOPT1 register, and TSnID4 to TSnID0.
The interrupt thinning out function is enabled by setting TSnRTE = 1, TSnRDE = 0, and TSnID4 to TSnID0.

17.10.8 High-accuracy T-PWM mode

In the high-accuracy T-PWM mode, 6-phase PWM is generated by using the 16-bit counter (up/down, ±2 counts, real 15-bit) and four 16-bit compare registers (LSB = additional pulse control bit).
The carrier wave value calculated with "TSnCCR0 – TSnDTC0 – TSnDTC1" is set to the TSnCCR0 register. A duty of U, V, and W-phase voltage data signal is set with the TSnCCR1 to TSnCCR3 registers. A dead time is set with the TSnDTC0 and TSnDTC1 registers. The TSnDTC0 register can set the dead time of negative phase (OFF) to positive phase (ON). TSnDTC1 can sets the dead time of positive phase (OFF) to negative phase (ON).
The 16-bit counter performs count-up operation using the TSnDTC0 register value as the minimum value, and count-down operation upon a match with the maximum value indicated by TSnCCR0 –TSnDTC1.
The 10-bit dead time counters (TSnDTT1 to TSnDTT3) reloads the value set to the TSnDTC0 and TSnDTC1 registers upon a match between the values of the TSnDTT1 to TSnDTT3 and TSnCCR1 to TSnCCR3 registers to perform count-down operation.
A compare match interrupt signal (INTTSnCC1 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 registers.

Note The high-accuracy T-PWM mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 0, 0].

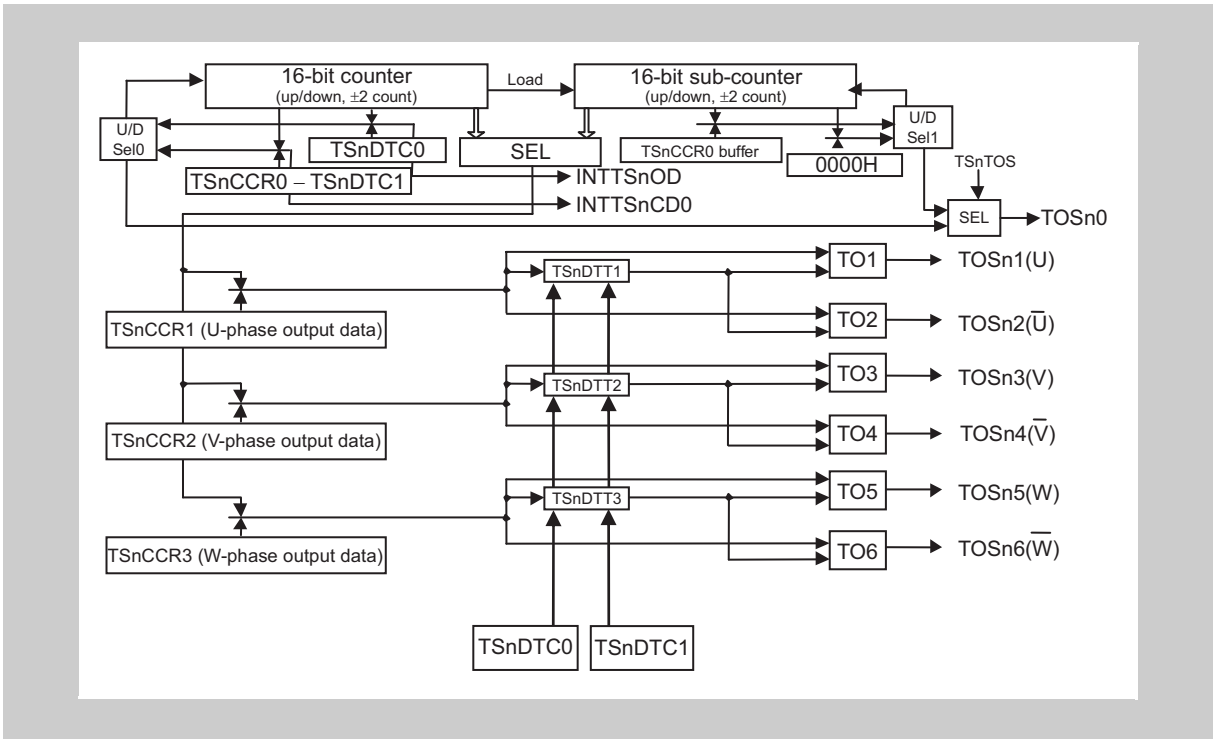


Figure 17-74 Block diagram of high-accuracy T-PWM mode

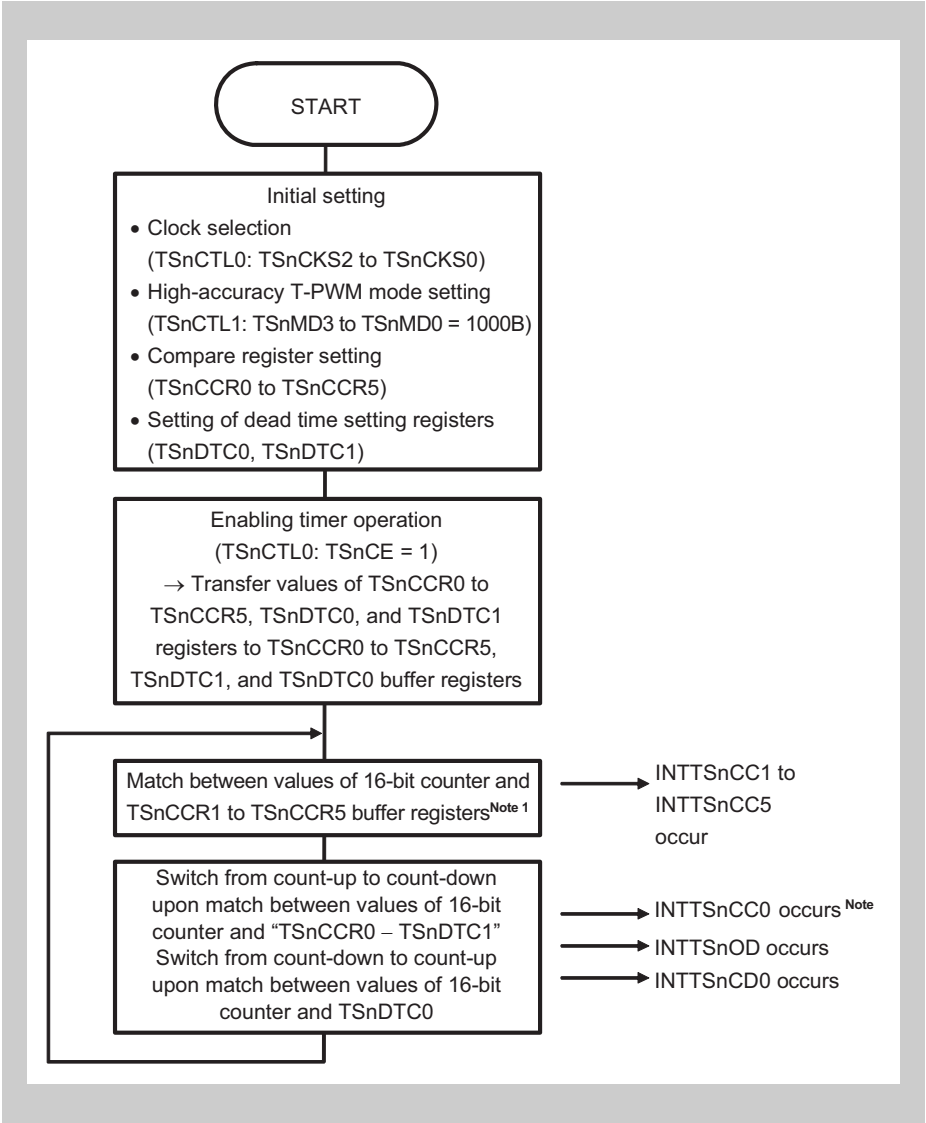


Figure 17-75 Basic operation flow in high-accuracy T-PWM mode

Note 1. The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 buffer registers.

2. Only when TSnDTC = 0000H

(1) High accuracy T-PWM mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting During Operation	Function
TSnCCR0	Reload/anytime rewrite	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload/anytime rewrite	Possible	PWM duty
TSnCCR4, TSnCCR5	Reload/anytime rewrite	Possible	PWM duty (selectable as A/D conversion trigger)
TSnDTC0, TSnDTC1	Reload	Conditionally possible[Note 1]	Cycle, dead time

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Inactive during counting up, active during counting down by 16-bit counter or 16-bit sub-counter
TOSn1	PWM output (with dead time) upon a TSnCCR1 compare match
TOSn2	Negative phase output for TOSn1[Note 2]
TOSn3	PWM output (with dead time) upon a TSnCCR2 compare match
TOSn4	Negative phase output for TOSn3[Note 2]
TOSn5	PWM output (with dead time) upon a TSnCCR3 compare match
TOSn6	Negative phase output for TOSn5[Note 2]
TOSn7	Pulse output by A/D conversion trigger

Note 1. Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* for details.

2. The output level of TOSn2, TOSn4, and TOSn6 will be switched to active level by setting TSnCE of the TSnCTL0 register to 1 while TSnOE2, TSnOE4, and TSnOE6 are 1, and TSnOL2, TSnOL4, and TSnOL6 are 0 in the TSnIOC0 register.

3. "–" indicates an unused function in the high-accuracy T-PWM mode.

(d) Interrupt

Interrupt	Function
INTTSnCC0	INTTSnCC0 compare match[Note 1]
INTTSnCC1 to INTTSnCC5	TSnCCR1 to TSnCCR5 compare match
INTTSnOV	Overflow[Note 2]
INTTSnER	Error
INTTSnOD	Trough interrupt
INTTSnCD0	Peak interrupt
INTTSnWN	Warning interrupt

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to TSnCCR0 − 2
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

(f) Output condition

Output	Condition
Duty 0% output	TSnCCRm = TSnCCR0
Duty 100% output	TSnCCRm = 0000H

Note
1. Only when TSnDTC1 = 0000H
2. When the TSnCCR0, TSnDTC0, and TSnDTC1 registers are incorrectly set
3. “–” indicates an unused function in the high-accuracy T-PWM mode.
4. m = 1 to 3

(2) Various setting in high-accuracy T-PWM mode

(a) Mode setting

The high-accuracy T-PWM mode is set by setting TSnMD3 to TSnMD0 of the TSnCTL1 register to [1, 0, 0, 0].

(b) Output level/output enable setting

Output level/output is enabled by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register.
The TOSn0 pin outputs the status of count-up/down operation by the 16-bit counter and 16-bit sub-counter. The 16-bit counter and 16-bit sub-counter is switched with the TSnTOS bit of the TSnOPT7 register.
The TOSn7 pin outputs pulses by A/D conversion trigger. Set this as necessary.

(c) Enabling error interrupt occurrence

Occurrence of an error interrupt (INTTSnER) when positive/negative phase simultaneous active is detected is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1.
In the high-accuracy T-PWM mode, the positive/negative phases do not become active simultaneously by setting any value to the TSnCCR0 to TSnCCR3 registers.

(d) Register rewrite timing setting with reload function

Reload (batch rewrite) or anytime rewrite (default “0” reload) will be executed for registers with the reload function by using the TSnCMS bit of the TSnOPT0 register. Be sure to set the TSnRTE or TSnRBE bit of the TSnOPT1 register to 1 to execute the reload operation.
No reload timings will occur when both the TSnRTE and TSnRBE bits are 0.
In case of anytime rewrite, unexpected output may occur depending on the rewrite timing.
Refer to (a) to (c) in *“Anytime rewrite mode” on page 763* when using the anytime rewrite function.

(e) Interrupt and thinning out function setting

The interrupt and thinning out functions can be set with the TSnOPT1 register. Set the TSnICE bit to 1 to generate peak interrupts (INTTSnCD0), and set the TSnIOE bit to 1 to generate trough interrupts (INTTSnOD).
To use the thinning out function for peak/trough interrupts, set the TSnID4 to TSnID0 bits.

(f) Reload thinning out function setting

Reload timings can be set to the same timing as interrupt timings by setting the TSnRDE bit of the TSnOPT1 register to 1.

(g) A/D conversion trigger output setting

A/D conversion trigger 0 (TSnADTRG0 signal) can be set using the TSnAT07 to TSnAT00 bits of the TSnOPT2 register.
Enabling/disabling of A/D conversion trigger output upon a match with the TSnCCR5 register (16-bit counter count-up/down operation), a match with the TSnCCR4 register (16-bit counter count-up/down operation), at a peak interrupt (INTTSnCD0), or at a trough interrupt of the 16-bit counter or 16-bit sub-counter, is set with the TSnAT07 to TSnAT00 bits.
A/D conversion trigger 1 (TSnADTRG1 signal) can be set using the TSnAT17 to TSnAT10 bits of the TSnOPT3 register.
Enabling/disabling of A/D conversion trigger output upon a match with the

TSnCCR5 register (16-bit counter count-up/down operation), a match with TSnCCR4 register (16-bit counter count-up/down operation), at a peak interrupt (INTTSnCD0), or at a trough interrupt of the 16-bit counter or 16-bit sub-counter, is set with the TSnAT17 to TSnAT10 bits.
To set match timings for the TSnCCR4 and TSnCCR5 register, set the compare values to each register.
The thinning out function can be used for the TSnADTRG0 and TSnADTRG1 signals. No thinning out, 1-thinning out, 3-thinning out and 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

Caution Set the TSnOPT2, TSnOPT3, TSnCCR4, and TSnCCR5 registers correctly in order to output A/D conversion trigger timing pulse to the TOSn7 pin.

(h) Dead time setting

A dead time is set with the TSnDTC0 and TSnDTC1 registers.
A dead time can be calculated with the following expressions.
16-bit counter operation clock cycle × TSnDTC0
16-bit counter operation clock cycle × TSnDTC1

The time from inactive change of the TOSn2, TOSn4, and TOSn6 pins to active change of the TOSn1, TOSn3, and TOSn5 pins can be set with the TSnDTC0 register.
The time from inactive change of the TOSn1, TOSn3, and TOSn5 pins to active change of the TOSn2, TOSn4, and TOSn6 pins can be set with TSnDTC1 register.

(i) Carrier wave cycle

Set the carrier wave cycle with the TSnCCR0 register based on the following expressions.
TSnCCR0 = (Carrier-wave cycle/16-bit counter operation clock cycle) + TSnDTC1 + TSnDTC0

The value set to the TSnCCR0 register must satisfy the following conditions in view of the dead time.
TSnCCR0 > 3 × MAX (TSnDTC0, TSnDTC1) + MIN (TSnDTC0, TSnDTC1)
0002H ≤ TSnCCR0 ≤ FFFEH
TSnCCR0 must be an even number.

Note MAX (A, B) indicates the greater value of A and B, and MIN (A, B) indicates the smaller value of A and B.

(j) Duty (PWM width) setting

U, V and W-phase duties can be set with the TSnCCR1 to TSnCCR3 registers, respectively. The setting range of the TSnCCR1 to TSnCCR3 registers are as follows.
0000H ≤ TSnCCR1 to TSnCCR3 ≤ TSnCCR0 + 1
LSB (Least Significant Bit) of the TSnCCR1 to TSnCCR3 registers means the additional pulse setting.
When TSnCCR1 = 0003H, a negative phase (TOSn2 pin) change delays by one count clock compared with that when TSnCCR1 = 0002H (during count-up operation by the 16-bit counter).

Caution Do not perform setting such that TSnCCR0 + 2 < TSnCCR1 to TSnCCR3.

(3) 16-bit counter operation in high-accuracy T-PWM mode

The initial value of the 16-bit counter is FFFEH, the value “TSnDTC0 value + 2” is loaded immediately after the timer Sn operation is set to start (TSnCE of TSnCTL0 register = 1), and the 16-bit counter counts up in units of +2. Then, the 16-bit counter counts down in units of –2 at the match timing with “TSnCCR0 – TSnDTC1”.
The 16-bit counter operates as follows.

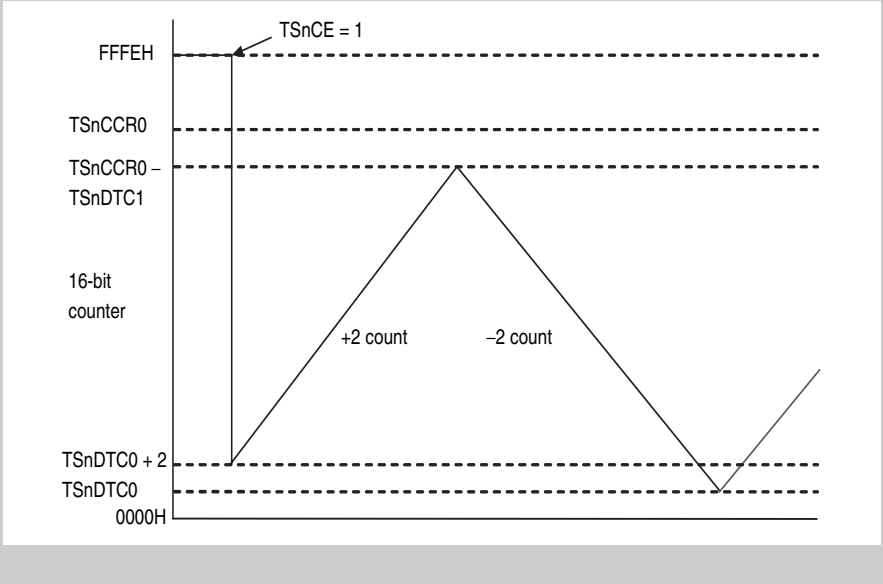


Figure 17-76 16-Bit counter operation in high-accuracy T-PWM mode

Note 16-bit counter minimum value: TSnDTC0
16-bit counter maximum value: TSnCCR0 – TSnDTC1
Carrier-wave cycle: (TSnCCR0 – TSnDTC0 – TSnDTC1) × count clock cycle

The initial value of the 16-bit sub-counter is FFFEH, and the value “TSnDTC0 value − 2” is loaded immediately after the timer Sn operation is set to start (TSnCE of TSnCTL0 register = 1). Then, the 16-bit counter counts down in units of −2 until the value matches with 0000H and starts counting up in units of +2 at the match timing.
The 16-bit counter value is loaded to the 16-bit sub-counter when the 16-bit counter operation is switched from counting up to counting down. The 16-bit sub-counter continues the count-up operation and starts counting down in units of −2 at the match timing with the TSnCCR0 register. The 16-bit counter value is loaded at the match timing with the TSnDTC0 register, and counting down is continued.
The 16-bit sub-counter operates as follows.

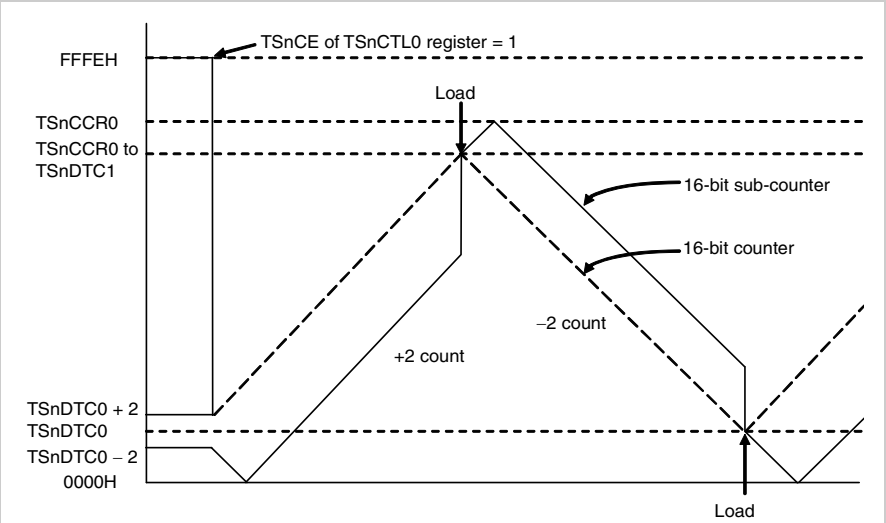


Figure 17-77 16-Bit sub-counter operation in high-accuracy T-PWM mode

Note 16-bit sub-counter minimum value: 0000H
16-bit sub-counter maximum value: TSnCCR0

(4) Basic operation in high-accuracy T-PWM mode

(a) Timer output example immediately after timer Sn operation starts

The timing diagram when TSnCCR0 = 0010H, TSnDTC0 = 0002H, TSnDTC1 = 0004H, and the TSnCCR1 register = 0000H to 0010H (partly extracted) is shown below. In this example, TSnOL1 to TSnOL6 of the TSnIOC0 register are set to 000000B.
When TSnCCR1 = TSnDTC0, the TOSn2 pin output level changes after compare match. In case of TSnCCR1 ≤ TSnDTC0 − 0002H, the TOSn2 pin output level varies to the active level when the initial value of the count is loaded after the timer Sn operation starts (TSnCE of TSnCTL0 register = 1). Change of the TOSn2 pin output level delays by one count clock compared with the case of TSnCCR1 = TSnDTC0 − 0002H, because TSnCCR1 = TSnDTC0 − 0001H is an additional pulse.

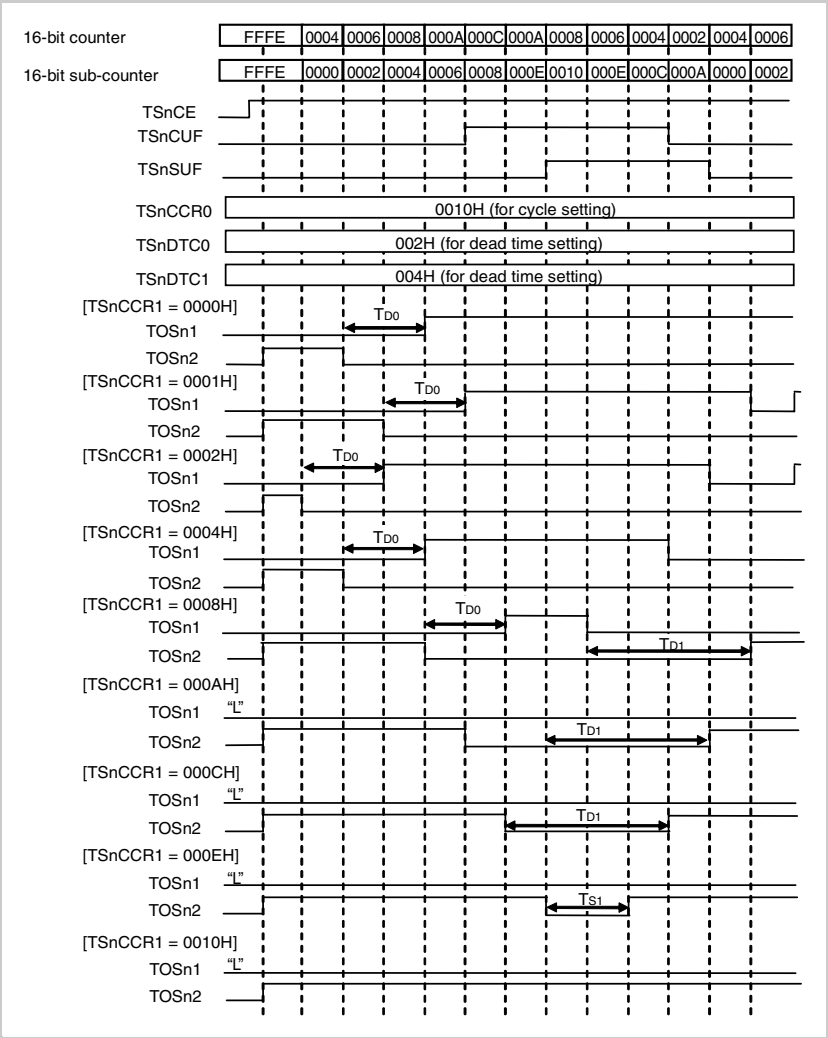


Figure 17-78 Timer output example when TSnCE = 1 (in high-accuracy T-PWM mode)

Note 1. TSnCCR0 = 0010H, TSnDTC0 = 0002H, TSnDTC1 = 0004H

2. T_{D0}:Time depending on TSnDTC0 register dead time setting
T_{D1}:Time depending on TSnDTC1 register dead time setting
T_{S1}:Time determined by result of compare between 16-bit sub-counter and TSnCCR1 when TSnCCR1 > 16-bit counter maximum value

(b) Timer output example during timer Sn operation

The timing diagram when TSnCCR0 = 0010H, TSnDTC0 = 0002H, TSnDTC1 = 0004H, and the TSnCCR1 register = 0000H to 0010H (partly extracted) is shown below. In this example, TSnOL1 to TSnOL6 of the TSnIOC0 register are set to 000000B.
The active (high level) range output by the positive phase (TOSn1 pin) is 0000H ≤ TSnCCR1 < TSnCCR0 − TSnDTC1 − TSnDTC0 (additional pulse).
The active (high level) range output by the negative phase (TOSn0 pin) is TSnDTC0 + TSnDTC1 < TSnCCR1 ≤ TSnCCR0.

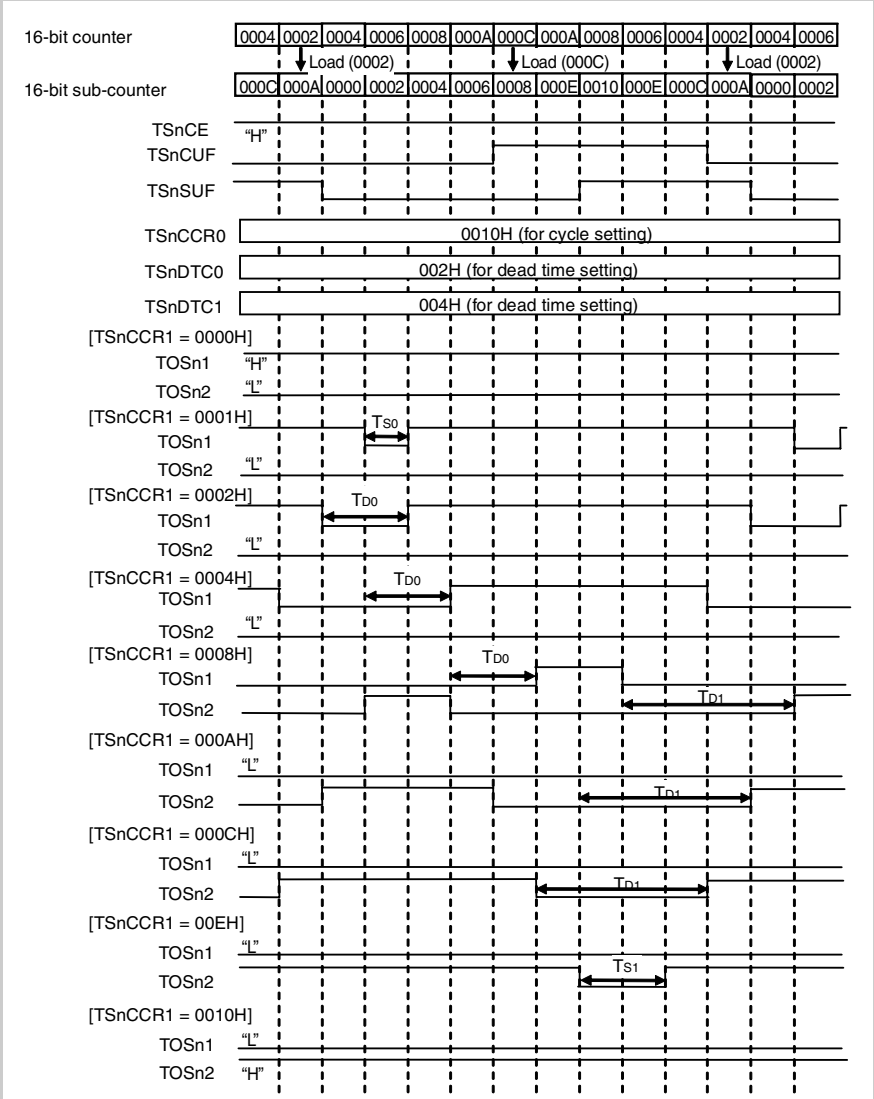


Figure 17-79 Timer output example during operation (in high-accuracy T-PWM mode)

Note 1. TSnCCR0 = 0010H, TSnDTC0 = 0002H, TSnDTC1 = 0004H

2. T_{D0}:Time depends on TSnDTC0 register dead time setting
T_{D1}:Time depends on TSnDTC1 register dead time setting
T_{S0}:Time determined by compare between 16-bit sub-counter and TSnCCR1 when TSnCCR1 < 16-bit counter minimum value.
T_{S1}:Time determined by result of compare between 16-bit sub-counter and TSnCCR1 when TSnCCR1 > 16-bit counter maximum value.

(5) Additional pulse control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, the additional pulse can be set by setting LSB of the duty setting registers (TSnCCR1 to TSnCCR3) to 1. The additional pulse control function enables finer duty control (with higher accuracy).
The examples of the TOSn1 pin output with/without additional pulse control are shown below, with the setting of TSnCCR0 = 12, and TSnDTC0, TSnDTC1 = 0.

(a) Pulse output with additional pulse control

In *Figure 17-80 on page 877*, additional pulse control is performed when an odd value is set to the TSnCCR1 register. Arrows and figures indicate the duty range of the TOSn1 pin output in one cycle.
As shown in *Figure 17-80 on page 877*, the TOSn1 pin output range (duty ratio) can be controlled by 1 count clock from 12-clock to 0-clock range when additional pulse control is performed.

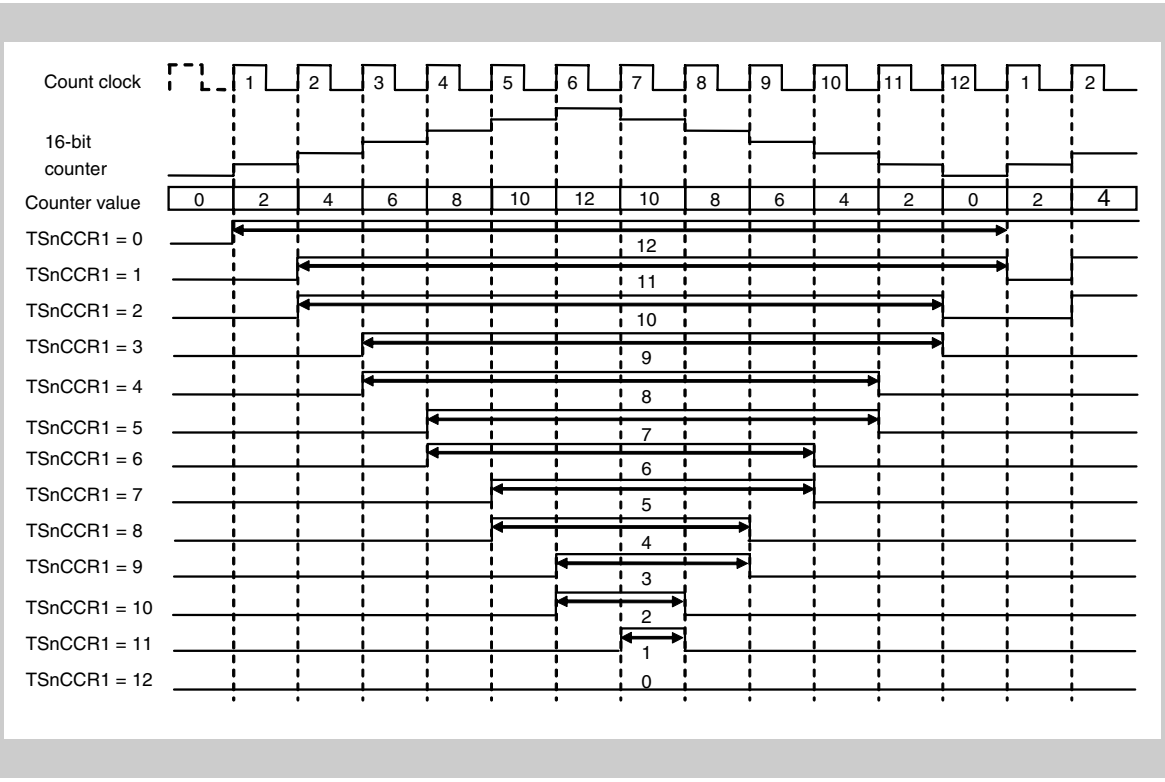


Figure 17-80 TOSn1 pin output example with additional pulse control

Note TSnCCR0 = 12, TSnDTC0 = 0, TSnDTC1 = 0

(b) Pulse output without additional pulse control

In *Figure 17-81 on page 878*, arrows and figures indicate the duty range of the TOSn1 pin output in one cycle.
The TOSn1 pin output range is controlled by 2 count clocks from 12-clock to 0-clock range if additional pulse control is not performed. In this case, duty variation volume becomes larger than that when additional pulse control is performed.

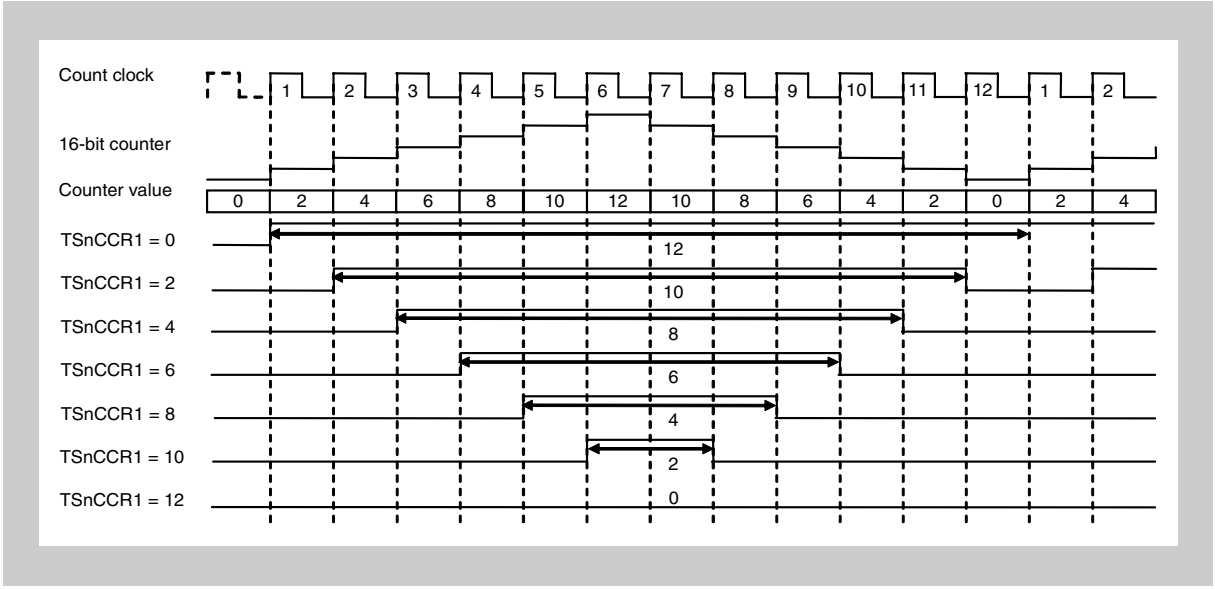


Figure 17-81 TOSn1 pin output example without additional pulse control

Note TSnCCR0 = 12, TSnDTC0 = 0, TSnDTC1 = 0

(6) Caution on timer output in high-accuracy T-PWM mode

There are cautions for TSnCCR1 to TSnCCR3 as follows when varying 6-phase PWM duty by using reload (batch rewrite).

(a) In case of TSnCCR0 + 2 ≤ TSnCCRm (Setting prohibited)

Figure 17-82 on page 879 shows the case when the value of "TSnCCR0 + 2 or more" is set to the TSnCCR1 register. When the TSnCCR1 register setting is changed like this, a match between the 16-bit counter and TSnCCR1 register does not occur thereafter. Therefore, the TOSn1 pin output level is forcibly changed to inactive level at the following 16-bit sub-counter trough timing. Output will be switched at 16-bit sub-counter peak/trough timing after that.

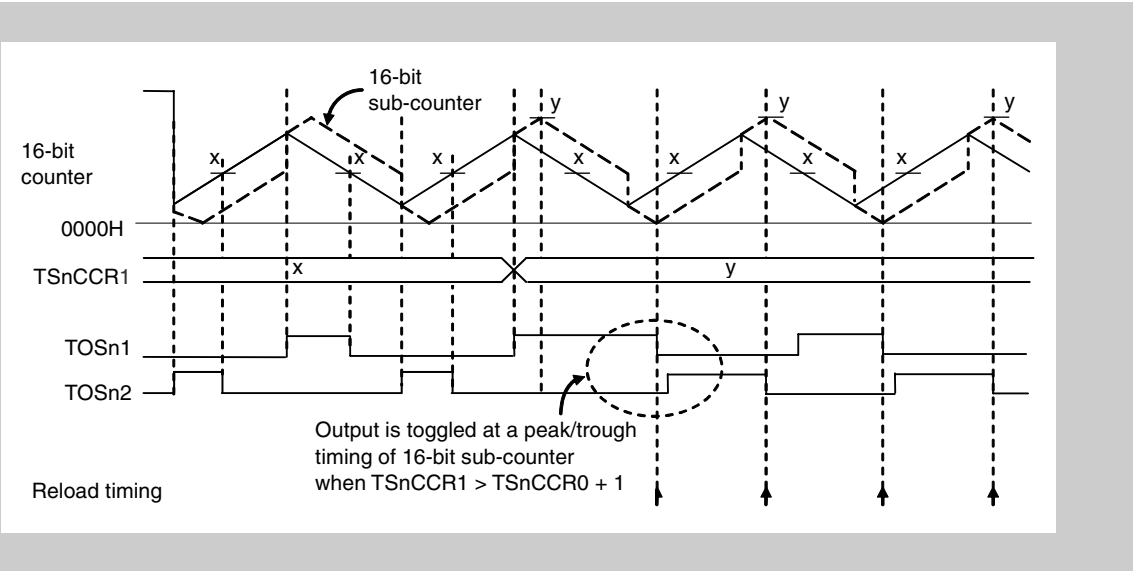


Figure 17-82 Output when TSnCCR0 + 2 ≤ TSnCCR1

Note m = 1 to 3

(b) In case of rewriting from TSnCCRm = 0000H to TSnCCRm = TSnCCR0

Figure 17-83 shows the output waveform where the TSnCCR1 register setting is changed from 100% output to 0% output.
The TOSn1 pin output is inverted upon a match between the TSnCCR1 register and 16-bit sub-counter, and the TOSn2 pin output is inverted after the dead time count.

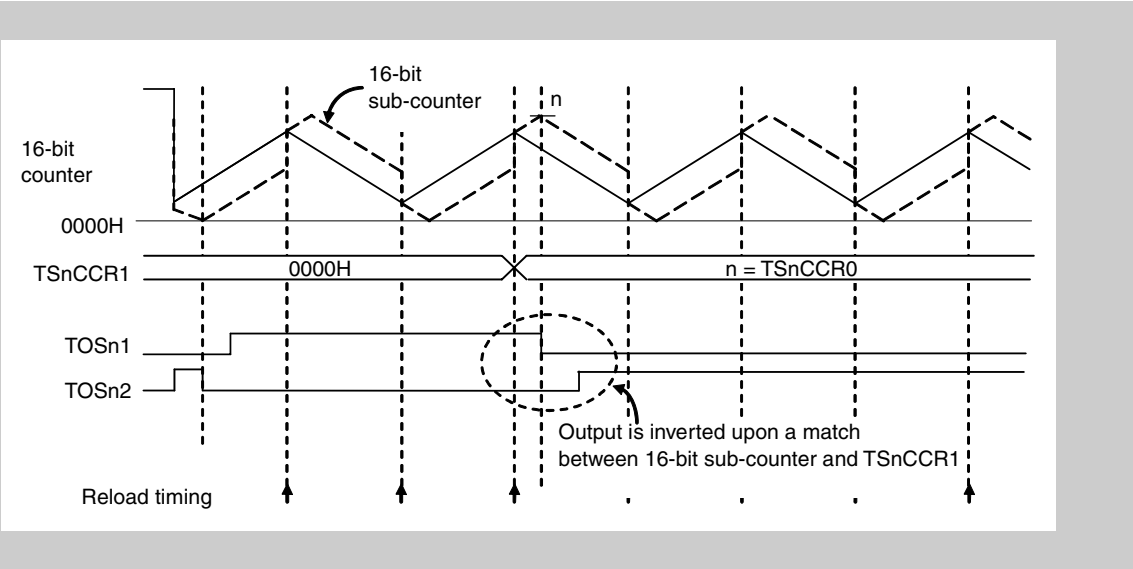


Figure 17-83 Output when rewriting from TSnCCR1 = 0000H to TSnCCR1 = TSnCCR0

Note m = 1 to 3

(c) In case of rewriting from "TSnDTC0 + TSnDTC1 < TSnCCRm < TSnCCR0 − TSnDTC0 − TSnDTC1" to "TSnCCRm < TSnDTC0 + TSnDTC1"

Figure 17-84 shows the output waveform when rewriting the TSnCCR1 register from x (TSnDTC0 + TSnDTC1 < x < TSnCCR0 − TSnDTC0 − TSnDTC1) to y (y < TSnDTC0 + TSnDTC1).
In this case, the TOSn1 pin output becomes active when the TOSn1 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and the TSnCCR1 register immediately after reload (batch rewrite).

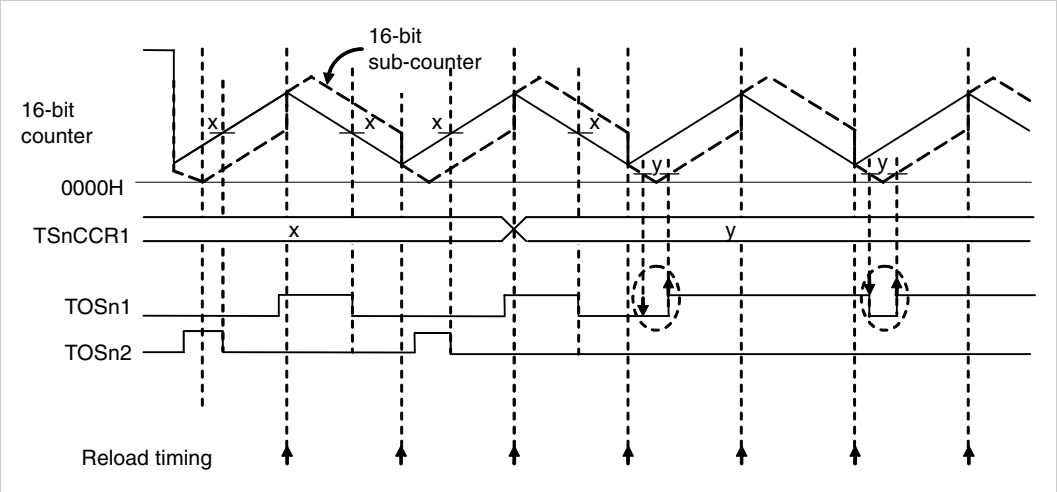


Figure 17-84 Output when rewriting from TSnDTC0 + TSnDTC1 < TSnCCR1 < TSnCCR0 − TSnDTC0 − TSnDTC1 to TSnCCR1 < TSnDTC0 + TSnDTC1

Note m = 1 to 3

(d) In case of rewriting from "TSnDTC0 + TSnDTC1 < TSnCCRm < TSnCCR0 – TSnDTC0 – TSnDTC1" to "TSnCCR0 – TSnDTC1 + 1 < TSnCCRm < TSnCCR0"

Figure 17-85 shows the output waveform when rewriting the TSnCCR1 register from x (TSnDTC0 + TSnDTC1 < x < TSnCCR0 – TSnDTC0 – TSnDTC1) to y (TSnCCR0 – TSnDTC0 – TSnDTC1 < TSnDTC0 < TSnCCR0). In this case, the TOSn2 pin output becomes inactive (high level) when the TOSn2 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and TSnCCRm register immediately after batch rewrite.

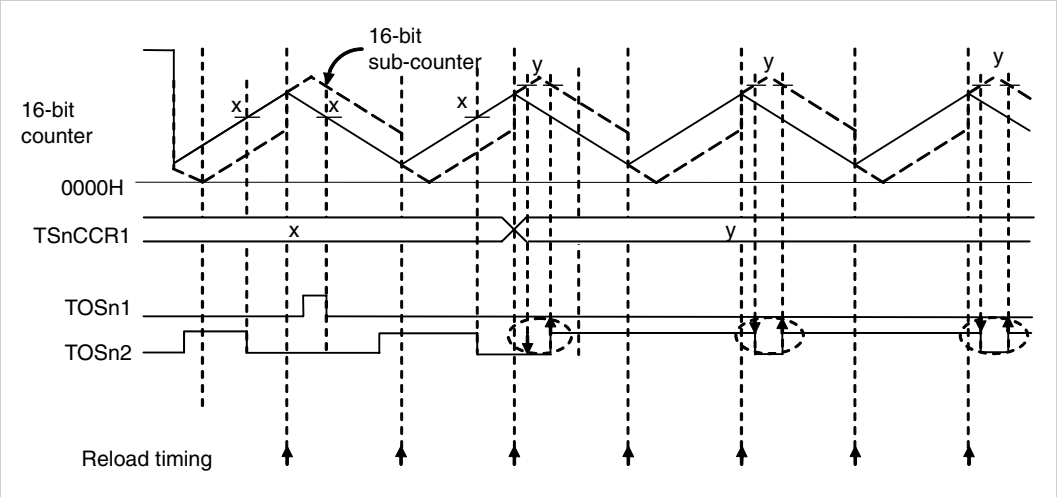


Figure 17-85 Output when rewriting from "TSnDTC0 + TSnDTC1 < TSnCCR1 < TSnCCR0 – TSnDTC0 – TSnDTC1" to "TSnCCR0 – TSnDTC1 + 1 < TSnCCR1 < TSnCCR0"

Note m = 1 to 3

(7) Timer output change after compare register updating

Timer output is affected when the compare register value is updated during reload execution. The timer output level is changed at any timing listed in *Table 17-18 on page 854* and *Table 17-19 on page 883*.

Table 17-19 Positive phase operation condition list

Operation	Symbol	Condition
Set	ST1	Match between counting up near the 16-bit sub-counter trough and compare register values
Clear	RT1	Match between counting down near the 16-bit sub-counter trough and compare register values
Set	ST2	At completion of dead time counter (TSnDTC0) operation
Clear	RT2	When 16-bit counter value matches with compare register value during count-down operation
Set	ST3	100% output for PWM duty
Clear	RT3	When no match occurs until 16-bit sub-counter counts down to 0000H
Clear	RT4	TSnCCR0 and TSnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TSnCCR0 and TSnDTC0, the operation is cleared by special processing.
Clear	RT5	The operation is cleared upon a match between peripheral 16-bit sub-counter peak and compare register values in positive phase active level.

Table 17-20 Negative phase operation condition list

Operation	Symbol	Condition
Set	SB1	Match between counting down near the 16-bit sub-counter peak and compare register values
Clear	RB1	Match between counting up near the 16-bit sub-counter peak and compare register values
Set	SB2	At completion of dead time counter (TSnDTC1) operation
Clear	RB2	When 16-bit counter value matches with compare register value during count-up operation
Set	SB3	100% output for PWM duty
Clear	RB3	When no match occurs until 16-bit sub-counter counts up to TSnCCR0
Clear	RB4	TSnCCR0 and TSnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TSnCCR0 and TSnDTC1, the operation is cleared by special processing.
Clear	RB5	The operation is cleared upon a match between peripheral 16-bit sub-counter trough and compare register values in negative phase active level.

Compare register value immediately before trough reload	Compare register value after trough reload (TSnDTC0 < TSnDTC1)	Figure No.
0000H	0000H < TSnCCR1 to TSnCCR3 < TSnDTC0	*Figure 17-86 on page 884*
	TSnCCR1 to TSnCCR3 = 0000H, TSnDTC0 + 1	*Figure 17-87 on page 885*
	TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 × 2	*Figure 17-88 on page 885*
	TSnDTC0 × 2 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC0 − TSnDTC1	*Figure 17-89 on page 886*
	TSnCCR0 − TSnDTC0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1	*Figure 17-90 on page 886*
	TSnCCR0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0	*Figure 17-91 on page 887*
	TSnCCR1 to TSnCCR3 = TSnCCR0	*Figure 17-92 on page 887*

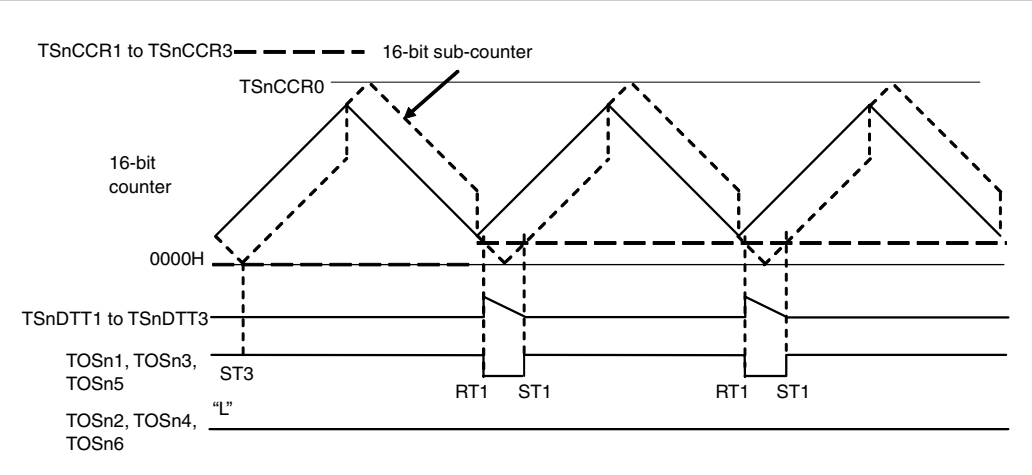


Figure 17-86 TSnCCR1 to TSnCCR3 = 0000H → 0000H < TSnCCR1 to TSnCCR3 < TSnDTC0

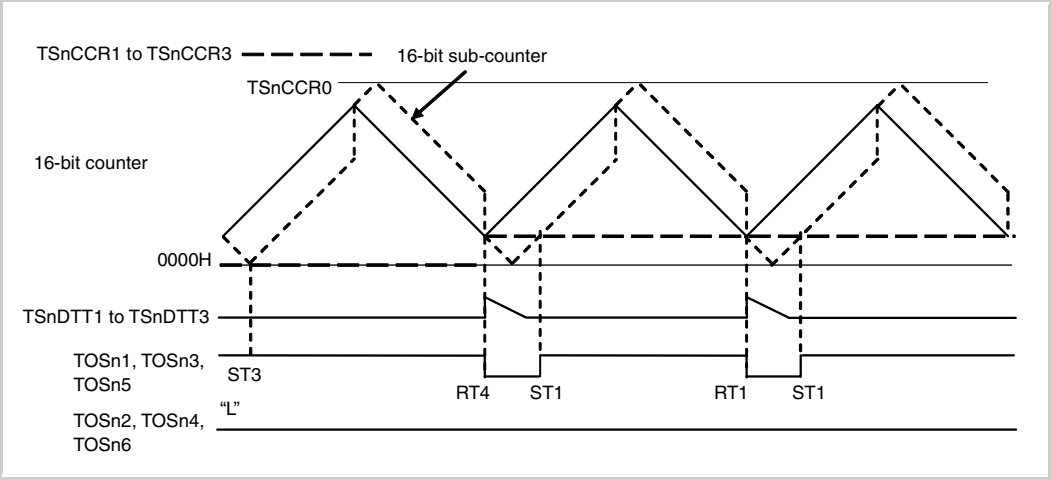


Figure 17-87 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR1 to TSnCCR3 = TSnDTC0, TSnDTC0 + 1

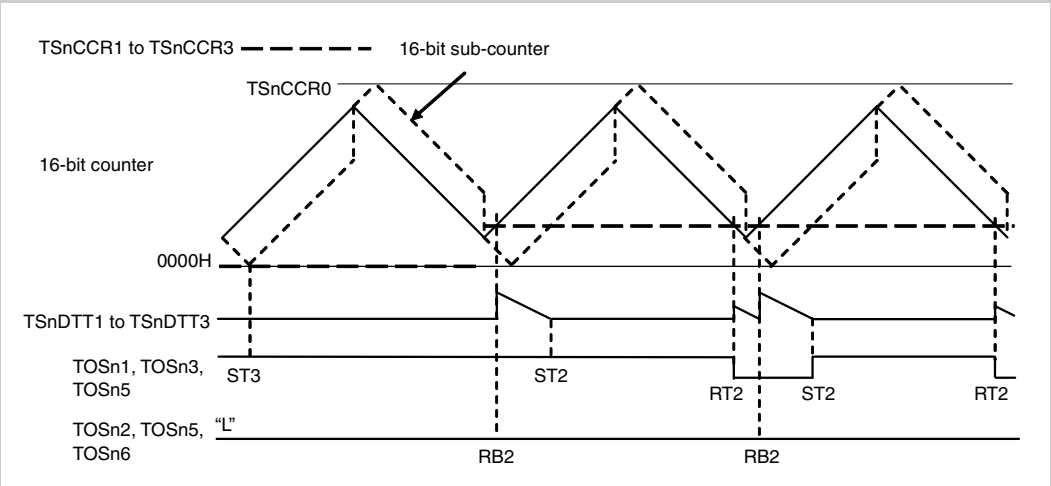


Figure 17-88 TSnCCR1 to TSnCCR3 = 0000H → TSnDTC0 < TSnCCR1 to TSnCCR3 < TSnDTC0 × 2

When the values of TSnCCR1 to TSnCCR3 are changed from "0000H ≤ TSnCCR1 to TSnCCR3 < TSnDTC0" to "TSnDTC0 < TSnCCR1 to TSnCCR3 < TSnDTC0 × 2", the positive phase will be 100% output for one cycle, as shown in *Figure 17-88*.
To prevent this phenomenon, change "0000H ≤ TSnCCR1 to TSnCCR3 < TSnDTC0" to "TSnDTC0 < TSnCCR1 to TSnCCR3 < TSnDTC × 2" through TSnDTC0, or directly change "0000H ≤ TSnCCR1 to TSnCCR3 < TSnDTC0" to "TSnDTC0 × 2 ≤ TSnCCR1 to TSnCCR3".

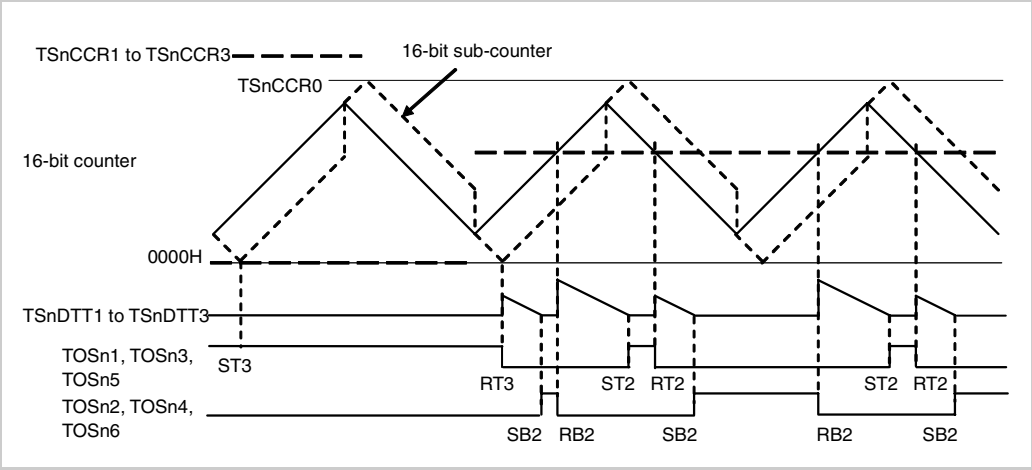


Figure 17-89 TSnCCR1 to TSnCCR3 = 0000H → TSnDTC0 × 2 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1 − TSnDTC0

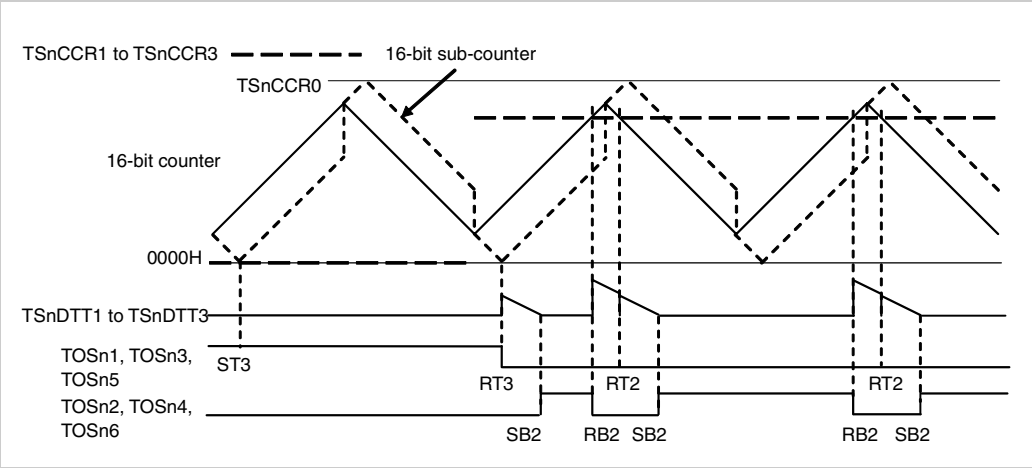


Figure 17-90 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR0 − TSnDTC1 − TSnDTC0 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1

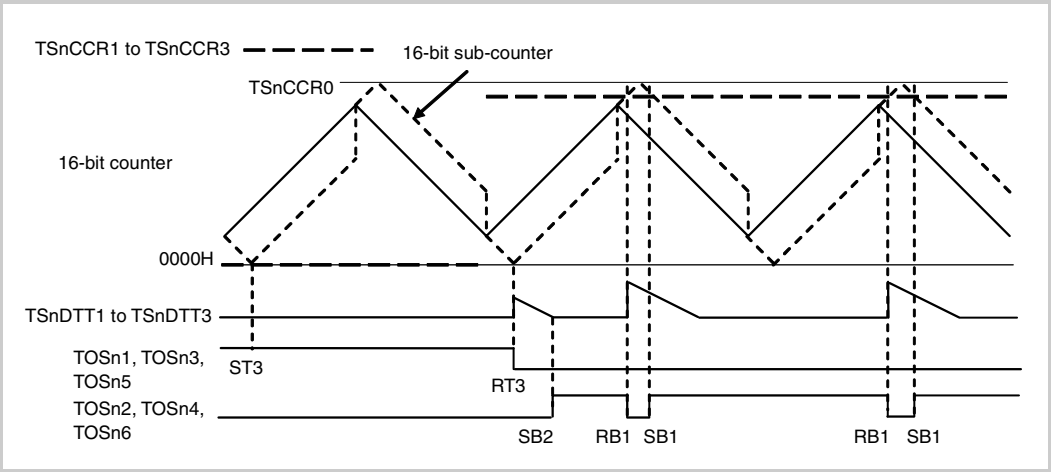


Figure 17-91 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR0 – TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0

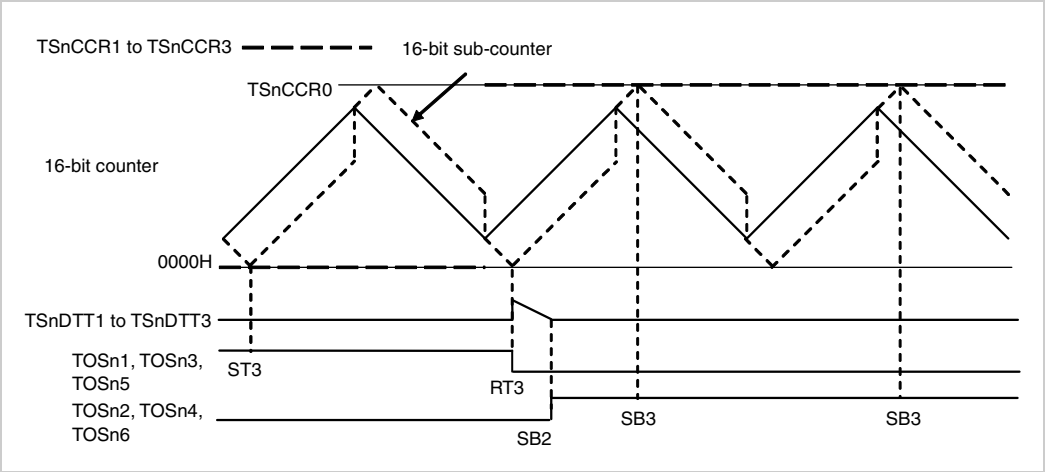


Figure 17-92 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR1 to TSnCCR3 < TSnCCR0

Compare register value immediately before trough reload	Compare register value after trough reload	Figure No.
TSnCCR0	TSnCCR1 to TSnCCR3 = 0000H	*Figure 17-93 on page 888*
	0000H < TSnCCR1 to TSnCCR3 < TSnDTC0	*Figure 17-94 on page 889*
	TSnCCR1 to TSnCCR3 = TSnDTC0, TSnDTC0 + 1	*Figure 17-95 on page 889*
	TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 < TSnDTC0 + TSnDTC1	*Figure 17-96 on page 890*
	TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC0 − TSnDTC1	*Figure 17-97 on page 890*
	TSnCCR0 − TSnDTC0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1	*Figure 17-98 on page 891*
	TSnCCR0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0	*Figure 17-99 on page 891*

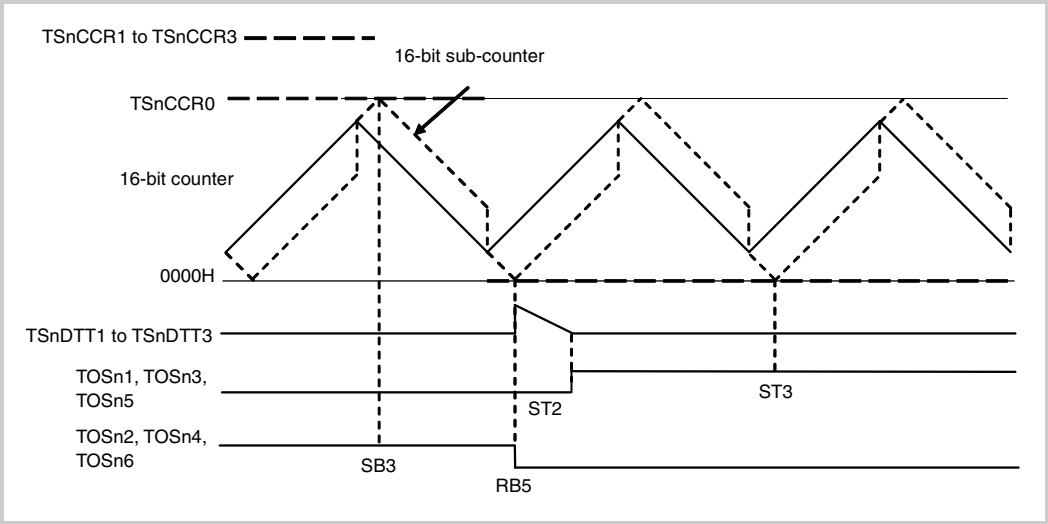


Figure 17-93 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR1 to TSnCCR3 = 0000H

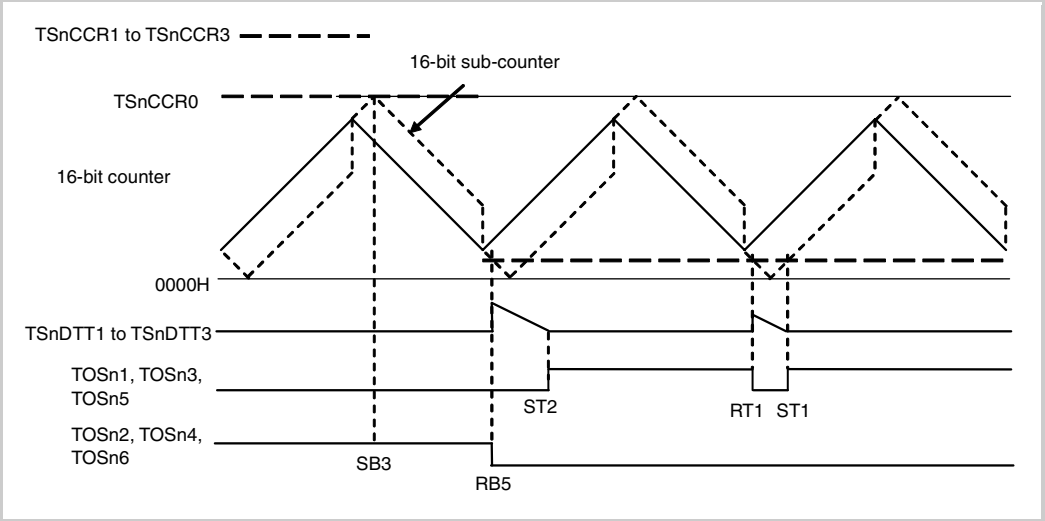


Figure 17-94 TSnCCR1 to TSnCCR3 = TSnCCR0 → 0000H < TSnCCR1 to TSnCCR3 < TSnDTC0

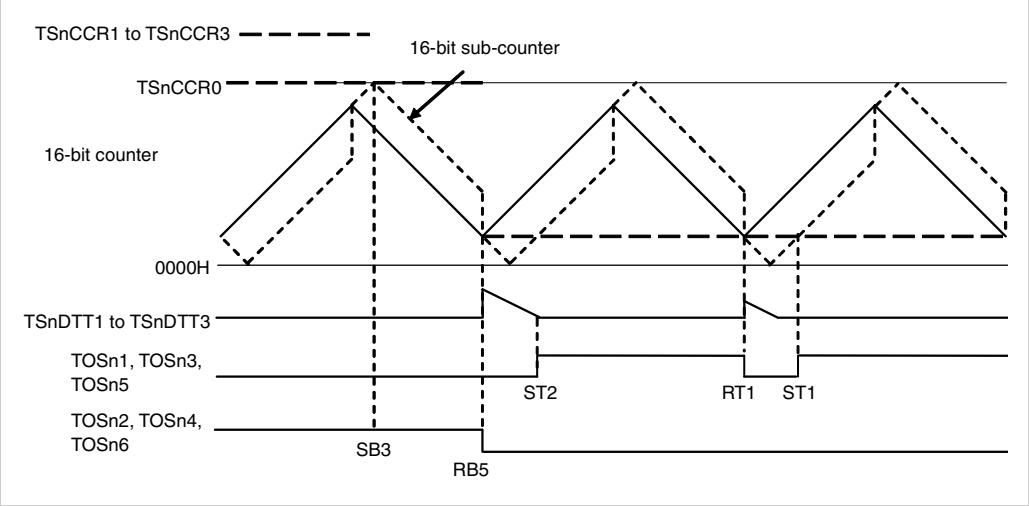


Figure 17-95 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR1 to TSnCCR3 = TSnDTC0, TSnDTC0 + 1

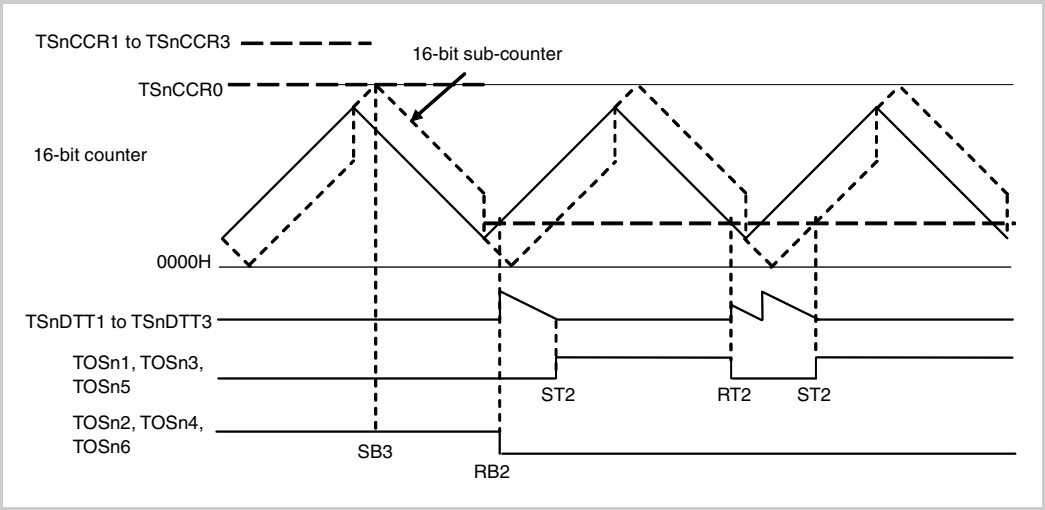


Figure 17-96 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + TSnDTC1

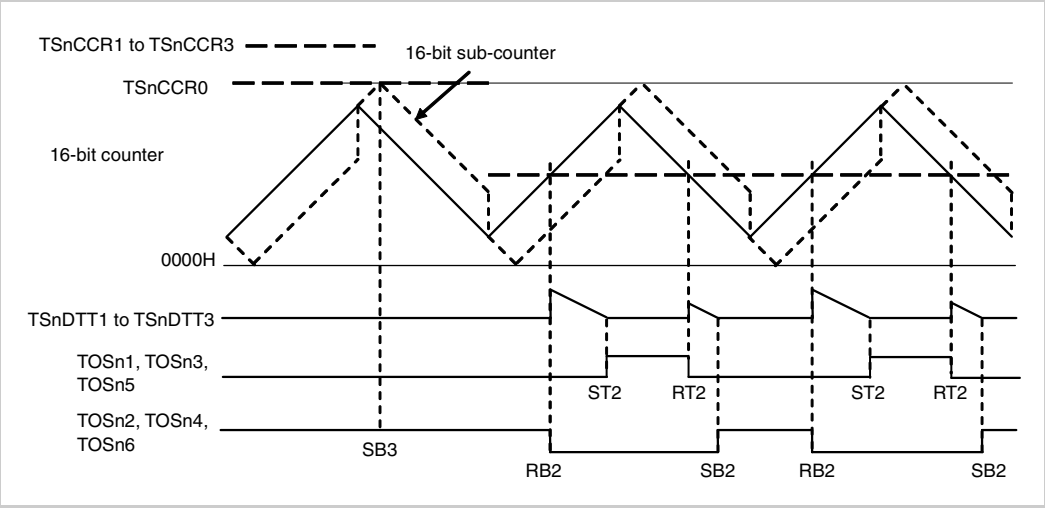


Figure 17-97 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1 − TSnDTC0

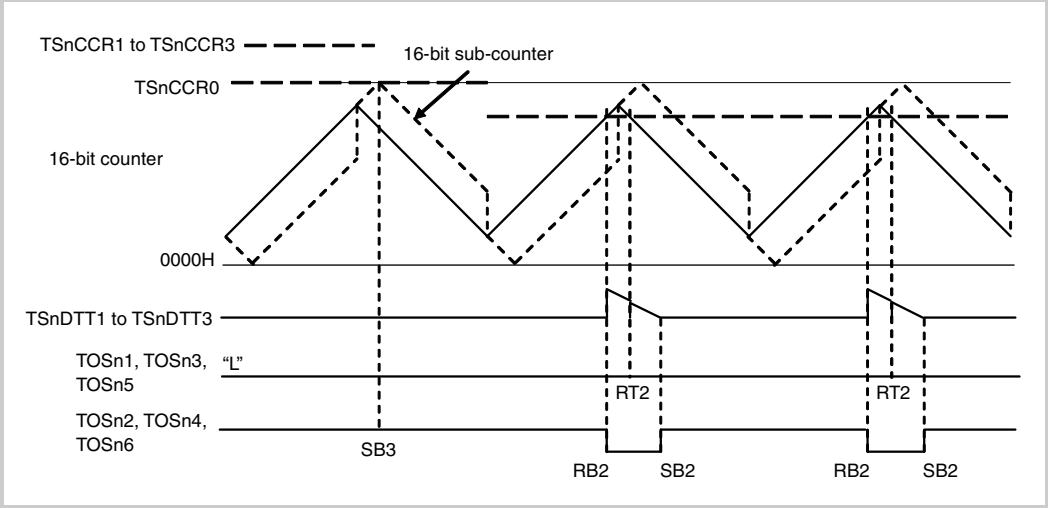


Figure 17-98 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR0 − TSnDTC1 − TSnDTC0 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1

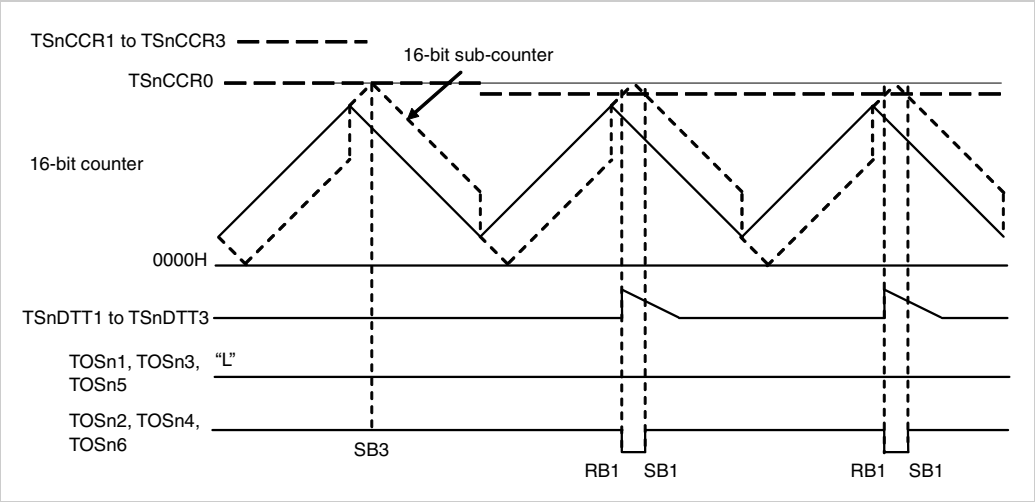


Figure 17-99 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0

Compare register value immediately before peak reload	Compare register value after peak reload (TSnDTC1 < TSnDTC0)	Figure No.
TSnCCR0	TSnCCR0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0	*Figure 17-100 on page 892*
	TSnCCR1 to TSnCCR3 = TSnCCR0 − TSnDTC1	*Figure 17-101 on page 893*
	TSnCCR0 − TSnDTC1 × 2 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1	*Figure 17-102 on page 893*
	TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1 × 2	*Figure 17-103 on page 894*
	TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 < TSnDTC0 + TSnDTC1	*Figure 17-104 on page 894*
	0000H < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + 1	*Figure 17-105 on page 895*
	TSnCCR1 to TSnCCR3 = 0000H	*Figure 17-106 on page 895*

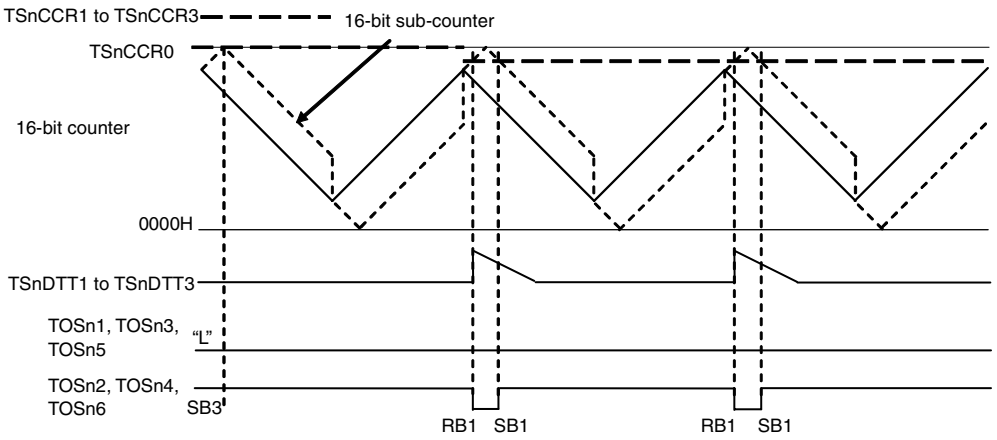


Figure 17-100 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR0 − TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0

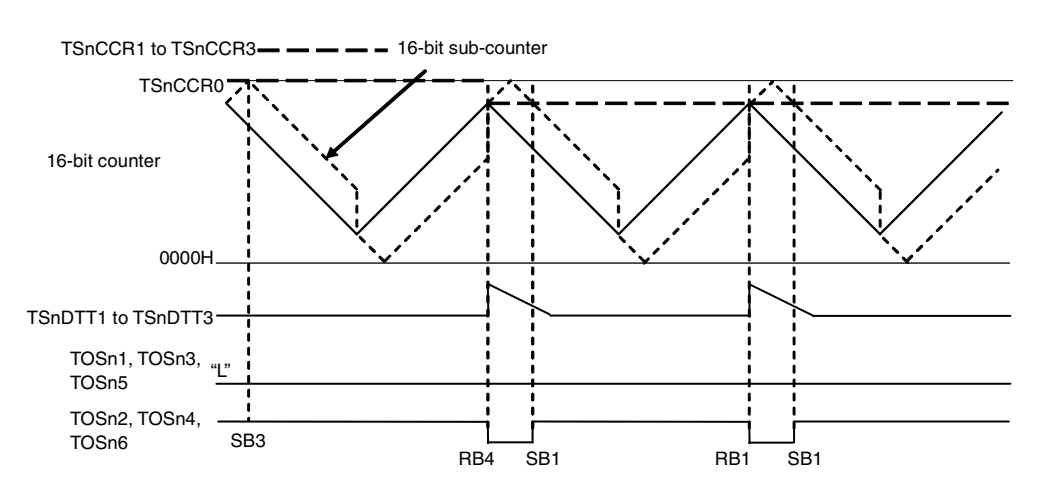


Figure 17-101 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR1 to TSnCCR3 = TSnDTC0 − TSnDTC1

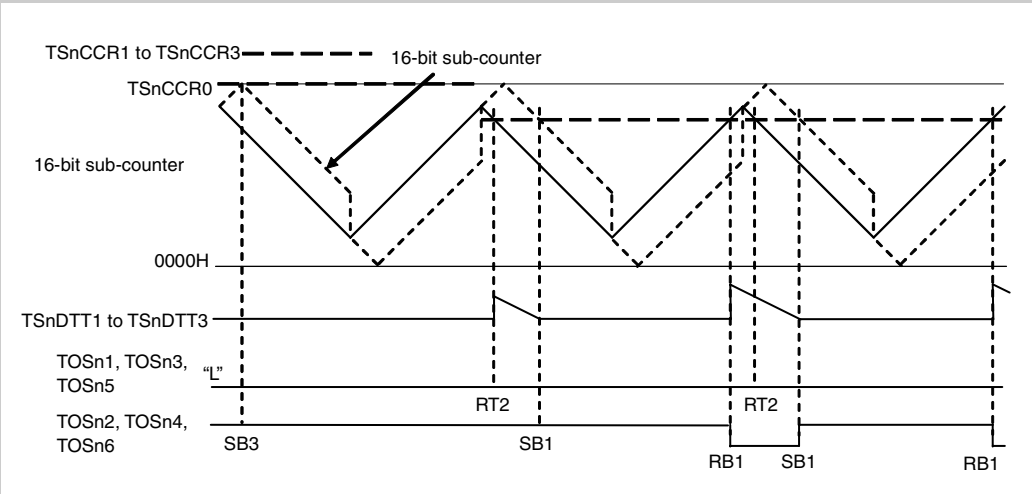


Figure 17-102 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR0 − TSnDTC1 x 2 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1

When the values of TSnCCR1 to TSnCCR3 are changed from "TSnCCR0 − TSnDTC1 < TSnCCR1 to TSnCCR3 ≤ TSnCCR0" to "TSnCCR0 − TSnDTC1 × 2 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1", the negative phase will be 100% output for one cycle, as shown in *Figure 17-102*.
To prevent this phenomenon, change "TSnCCR0 − TSnDTC1 < TSnCCR1 to TSnCCR3 ≤ TSnCCR0" to "TSnDTC0 < TSnCCR1 to TSnCCR3 < TSnDT1 × 2" through "TSnCCR0 − TSnDTC1", o directly change "TSnCCR0 − TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0" to "TSnCCR1 to TSnCCR3 ≤ TSnCCR0 − TSnDTC1 × 2".

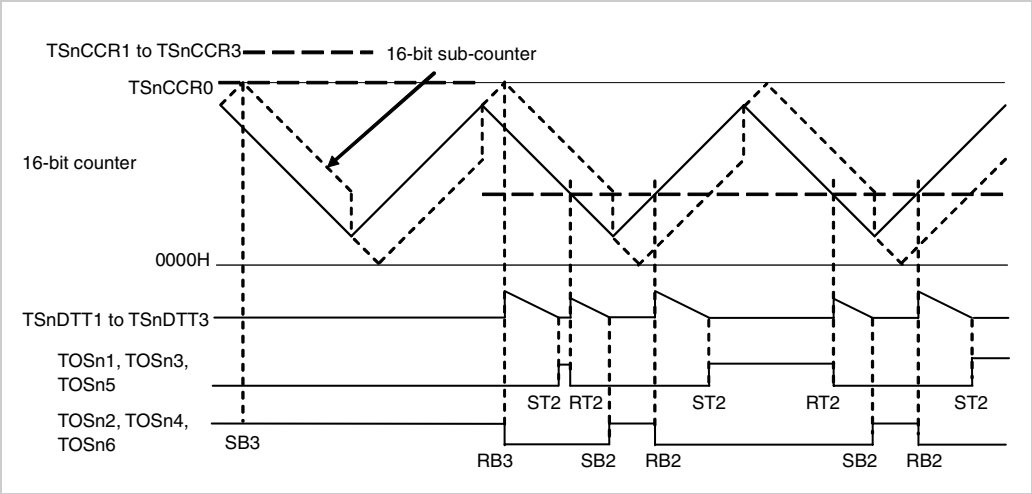


Figure 17-103 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1 x 2

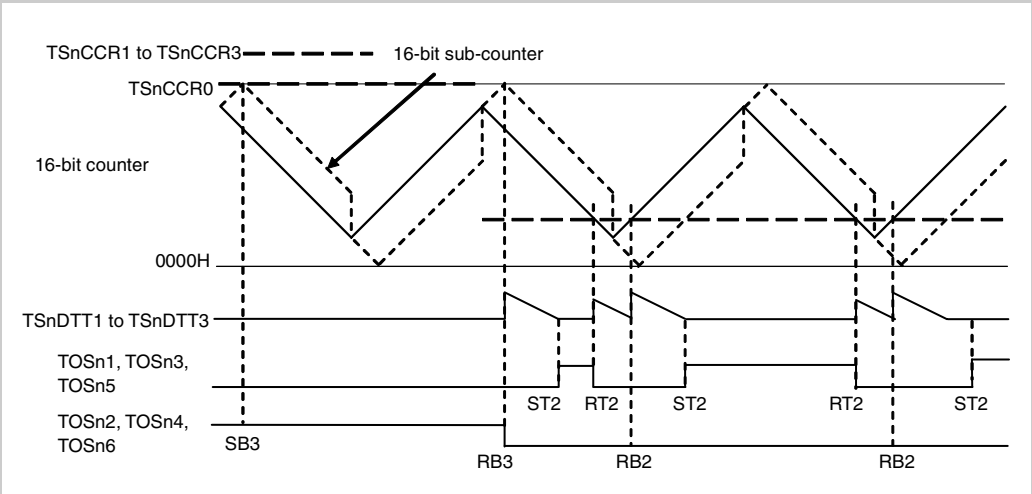


Figure 17-104 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + TSnDTC1

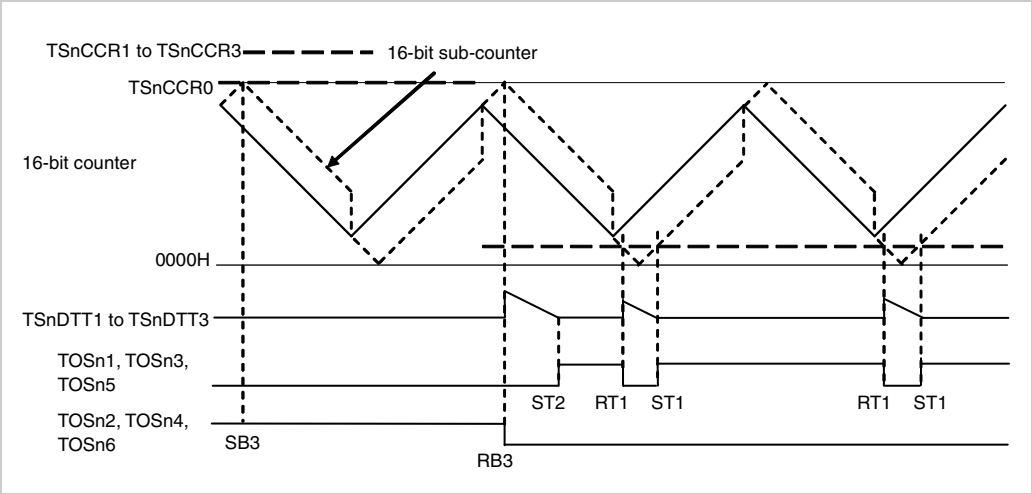


Figure 17-105 TSnCCR1 to TSnCCR3 = TSnCCR0 → 0000H < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + 1

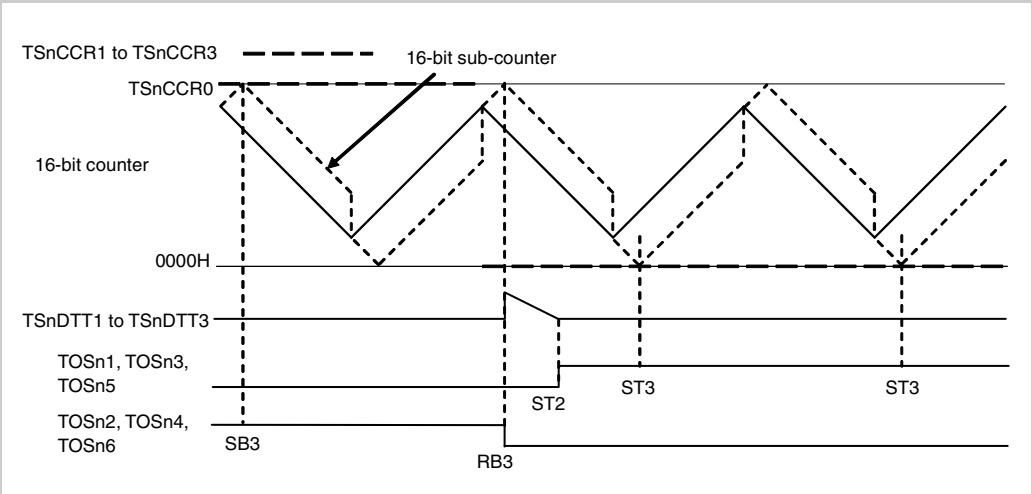


Figure 17-106 TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR1 to TSnCCR3 = 0000H

Compare register value immediately before peak reload	Compare register value after trough reload	Figure No.
0000H	TSnCCR1 to TSnCCR3 = TSnCCR0	*Figure 17-107 on page 896*
	TSnCCR0 − TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0	*Figure 17-108 on page 897*
	TSnCCR1 to TSnCCR3 = TSnCCR0 − TSnDTC1	*Figure 17-109 on page 897*
	TSnCCR0 − TSnDTC0 − TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC1	*Figure 17-110 on page 898*
	TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 − TSnDTC0 − TSnDTC1	*Figure 17-111 on page 898*
	TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + TSnDTC1	*Figure 17-112 on page 899*
	0000H < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + 1	*Figure 17-113 on page 899*

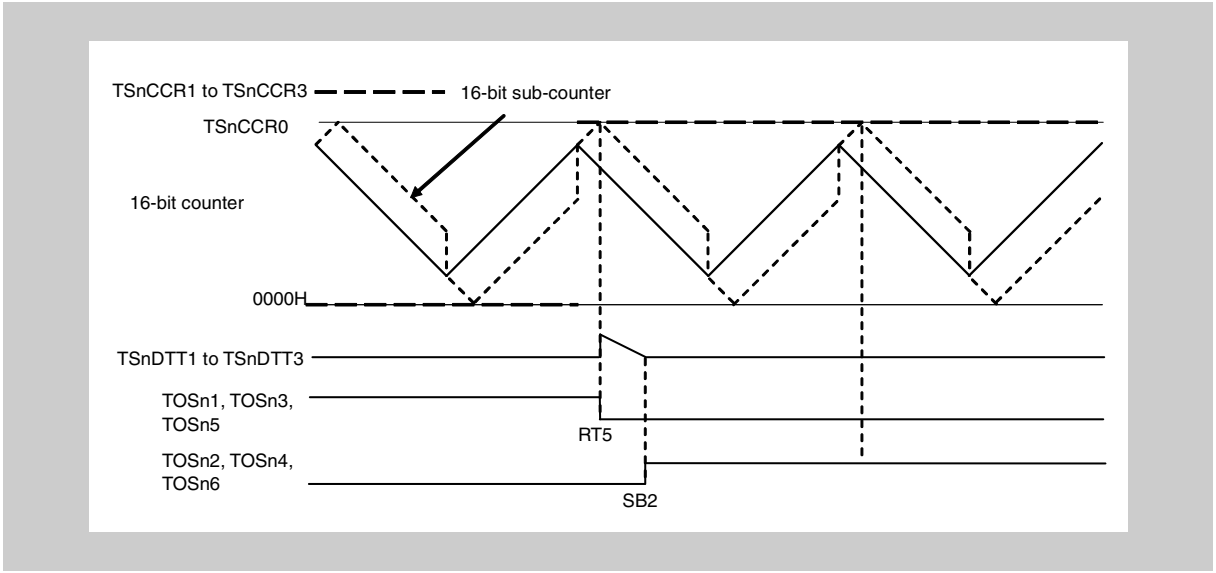


Figure 17-107 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR1 to TSnCCR3 = TSnCCR0

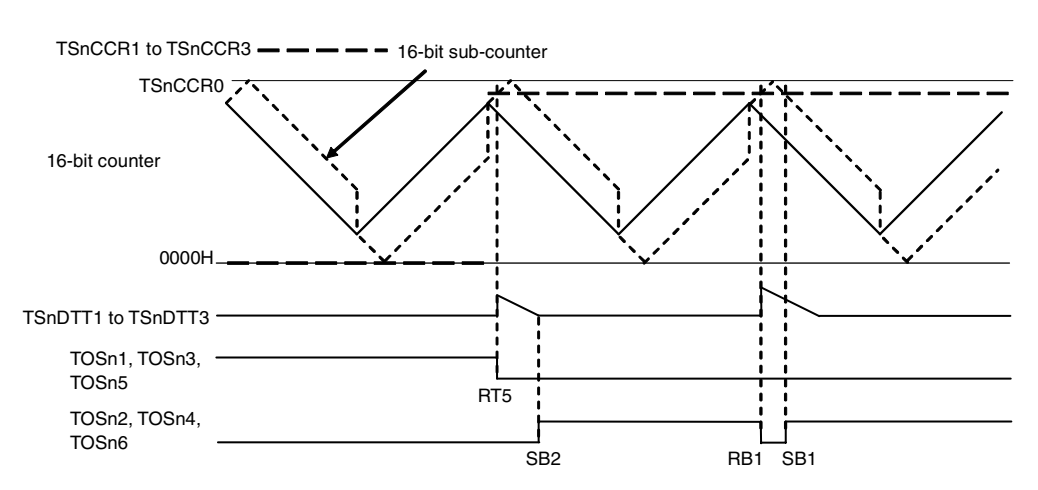


Figure 17-108 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR0 – TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0

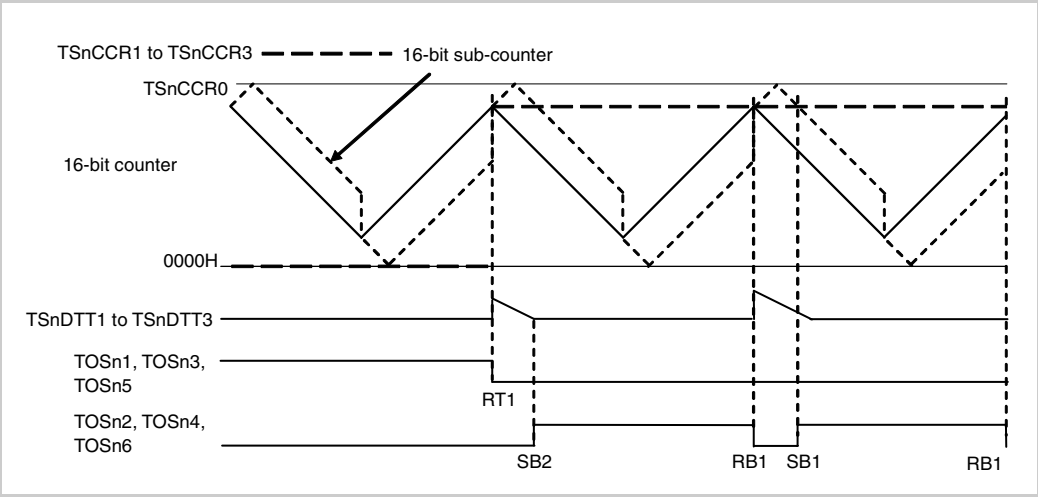


Figure 17-109 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR1 to TSnCCR3 = TSnCCR0 – TSnDTC1

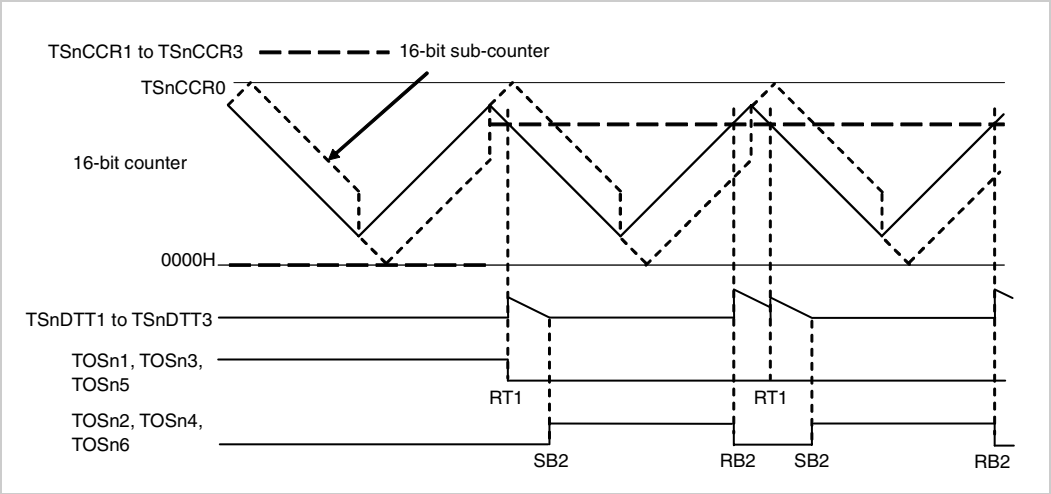


Figure 17-110 TSnCCR1 to TSnCCR3 = 0000H → TSnCCR0 – TSnDTC0 – TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 – TSnDTC1

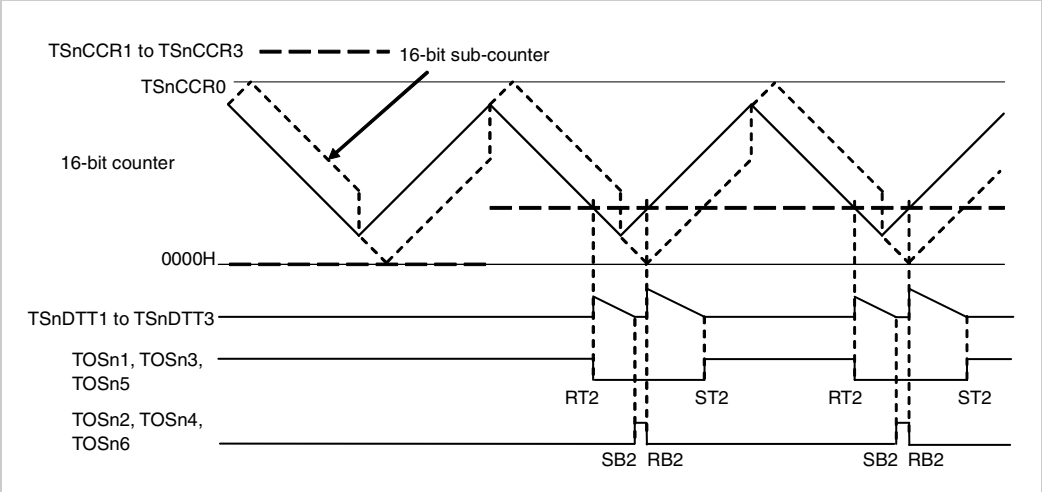


Figure 17-111 TSnCCR1 to TSnCCR3 = 0000H → TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 ≤ TSnCCR0 – TSnDTC0 – TSnDTC1

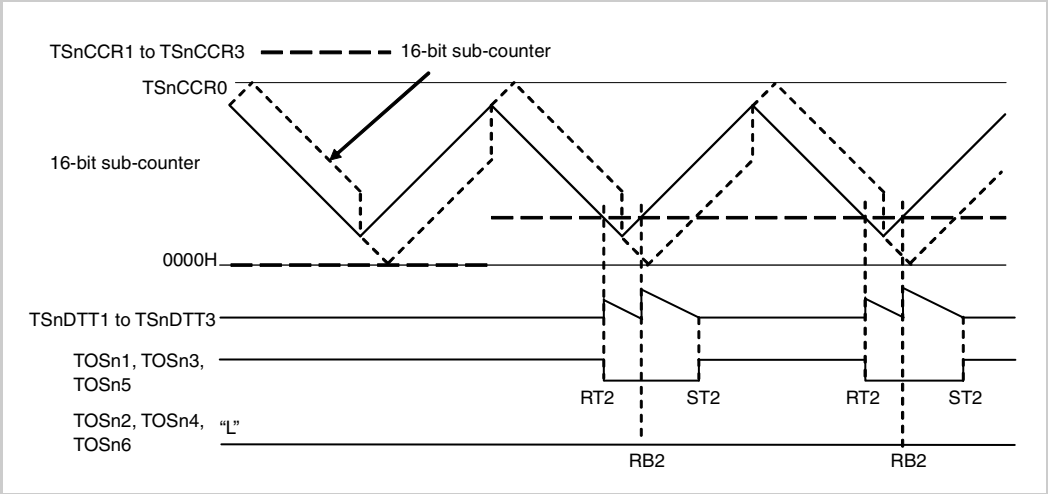


Figure 17-112 TSnCCR1 to TSnCCR3 = 0000H → TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 < TSnDTC0 + TSnDTC1

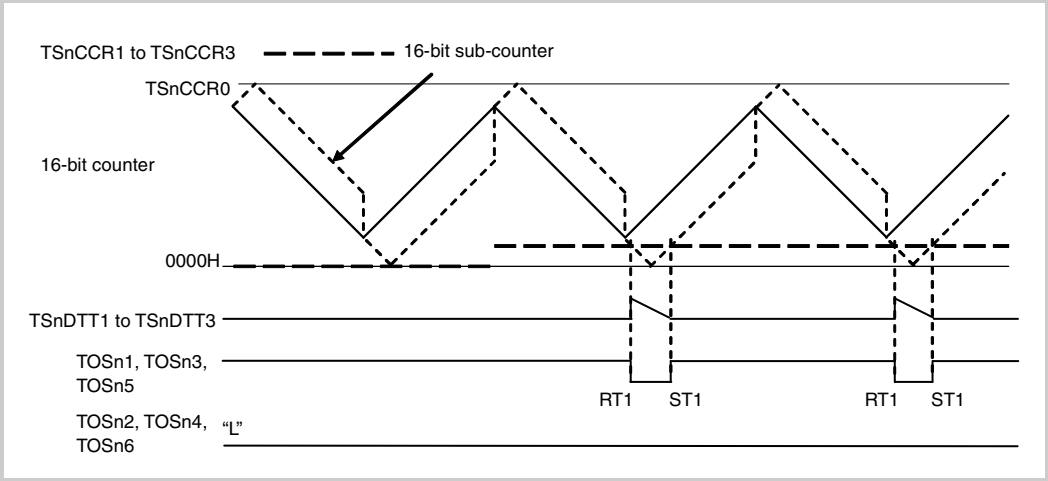


Figure 17-113 TSnCCR1 to TSnCCR3 = 0000H → 0000H < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 + 1

(8) Dead time control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, the TSnCCR1 to TSnCCR3 registers are used for duty setting and the TSnCCR0 register is used for cycle setting. By using these four registers, duty variable type 6-phase PWM waveform can be output. To implement dead time control, there are three 10-bit down-counters that synchronously operate with the count clock of the 16-bit counter, and two dead time setting registers (TSnDTC0, TSnDTC1).
The TSnDTC0 register is used to set the dead time from when a negative phase changes to inactive until a positive phase changes to active. The TSnDTC1 register is used to set the dead time from when a positive phase changes to inactive until a negative phase changes to active.
The output waveform in case of TSnDTC0 = x, TSnDTC1 = y is shown below.

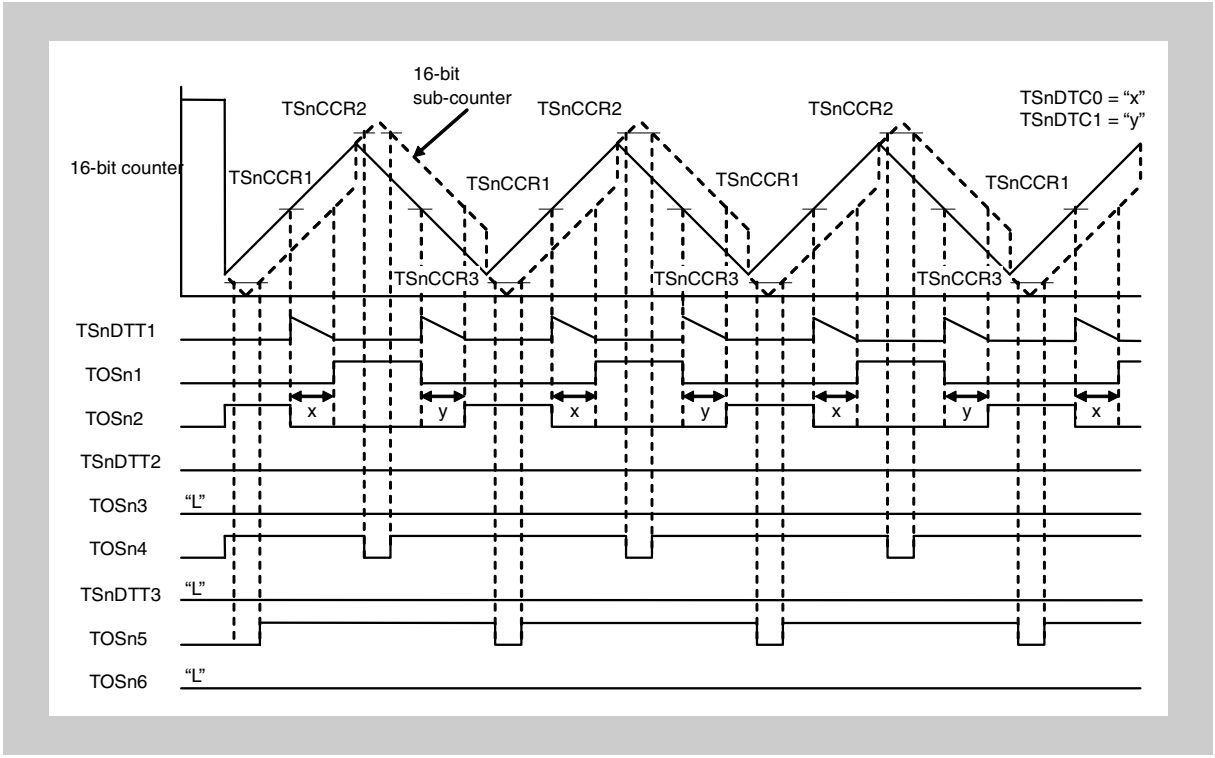


Figure 17-114 Output waveform example when dead time is set

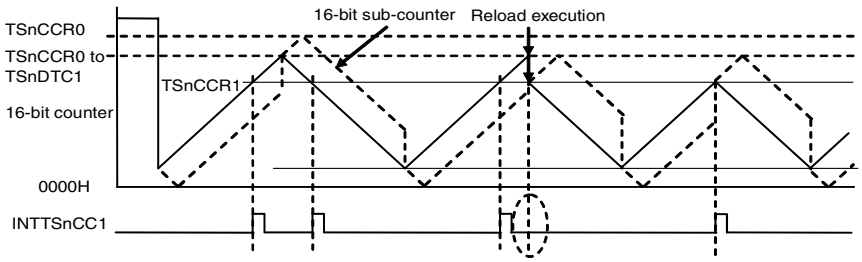
(9) Cautions on dead time control in high-accuracy T-PWM mode

(a) Rewriting of TSnDTC0 and TSnDTC1 registers

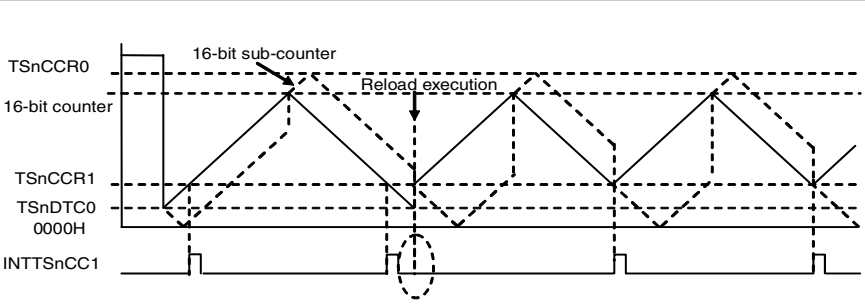
The setting of the dead time in the TSnDTC0, TSnDTC1 registers can be rewritten during operation. Note the following cautions when rewriting the dead time setting during operation.

Caution

1. Rewrite the TSnDTC0 and TSnDTC1 registers when using the reload function (TSnCMS = 0).
2. When the TSnDTC0 and TSnDTC1 registers are rewritten, carrier-wave cycles will be changed. In cases where carrier-wave cycles should not be changed, rewrite the TSnCCR0 register value at the same time as changing the TSnDTC0 and TSnDTC1 registers.
3. Rewriting is prohibited when TSnCMS = 1.
4. In case of changing TSnCCR0 and TSnCCR1 at a 16-bit counter peak: Match interrupts (INTTSnCC1 to INTTSnCC5) will not occur immediately after reload execution if the values set in the TSnCCR1 to TSnCCR5 register matches with and TSnCCR0 − TSnDTC1 (the new maximum value of main counter) after updating.



5. In case of changing TS0DTC0 at a 16-bit counter trough: Match interrupts (INTTSnCC1 to INTTSnCC5) will not occur immediately after reload execution if the values set in the TSnCCR1 to TSnCCR5 register match with TS0DTC0 (the new minimum value of main counter) after updating.



(10) Caution on rewriting cycles in high-accuracy T-PWM mode

In high-accuracy T-PWM mode, setting conditions for the TSnCCR0, TSnDTC0, and TSnDTC1 registers are as follows.

3 × MAX (TSnDTC0, TSnDTC1) + MIN (TSnDTC0, TSnDTC1) < TSnCCR0
0002H < TSnCCR0 ≤ FFFEH

MAX (A, B) indicates the greater value of A and B, and MIN (A, B) indicates the smaller value of A and B.

Figure 17-115 on page 902 shows an operation example when the setting range is exceeded.
This example shows the case where the TSnDTC0 register is set out of the range "TSnDTC0 ≥ TSnCCR0 − TSnDTC1". Though the 16-bit counter executes count-down operation, the count-down operation is executed from 0000H because no match occurs. In this case, the count operation continues by loading the TSnDTC0 register setting value. However, no match with TSnCCR0 − TSnDTC1 occurs in the count-up operation, thus the 16-bit counter overflows. In this case, the count operation continues by loading the TSnDTC0 register setting value again.
An overflow interrupt (INTTSnOV) occurs when the 16-bit counter loads the TSnDTC0 register setting value from 0000H or when an overflow occurs at FFFEH, and then the TSnOVF flag is set. An overflow interrupt (INTTSnOV) does not occur if the TSnCCR0, TSnDTC0, and TSnDTC1 registers are set correctly, so this can be used for detecting incorrect settings.

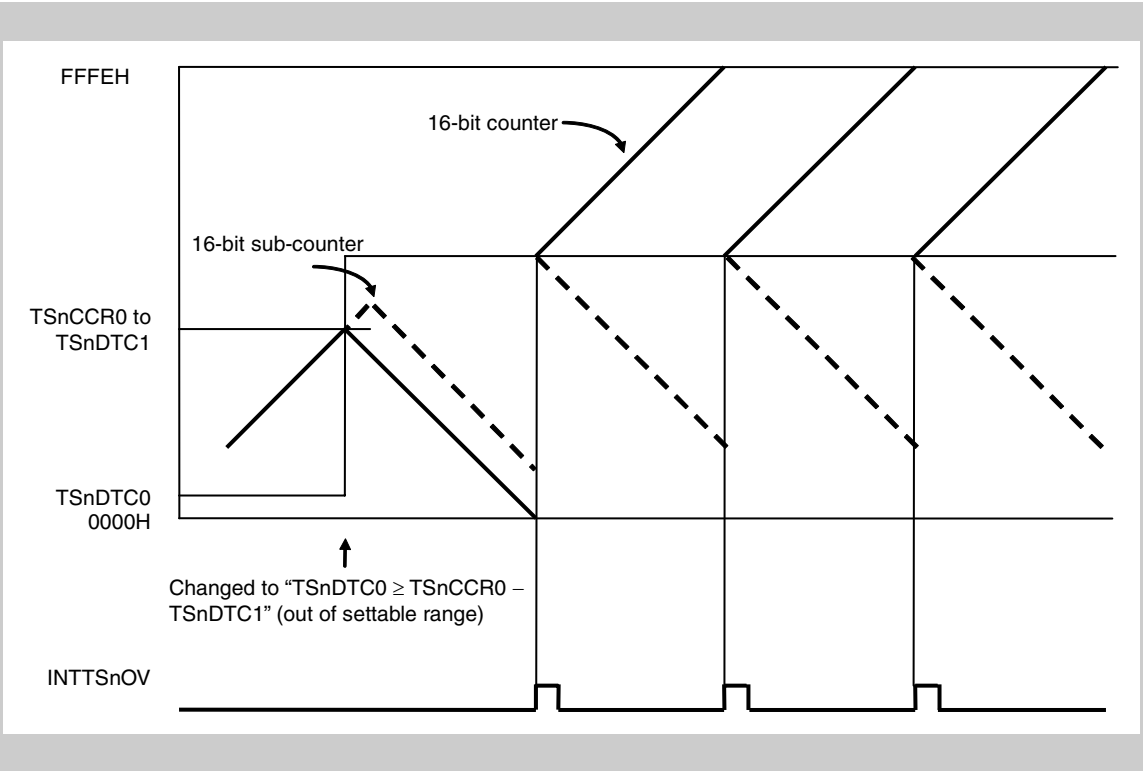


Figure 17-115 Operation example setting is out of range

(11) Error interrupt (INTTSnER) in high-accuracy T-PWM mode

The positive/negative simultaneous active detection function can be used in the high-accuracy T-PWM mode. Error interrupts (INTTSnER) do not occur in the high-accuracy T-PWM mode. In case of occurrence, the internal circuits may be damaged.

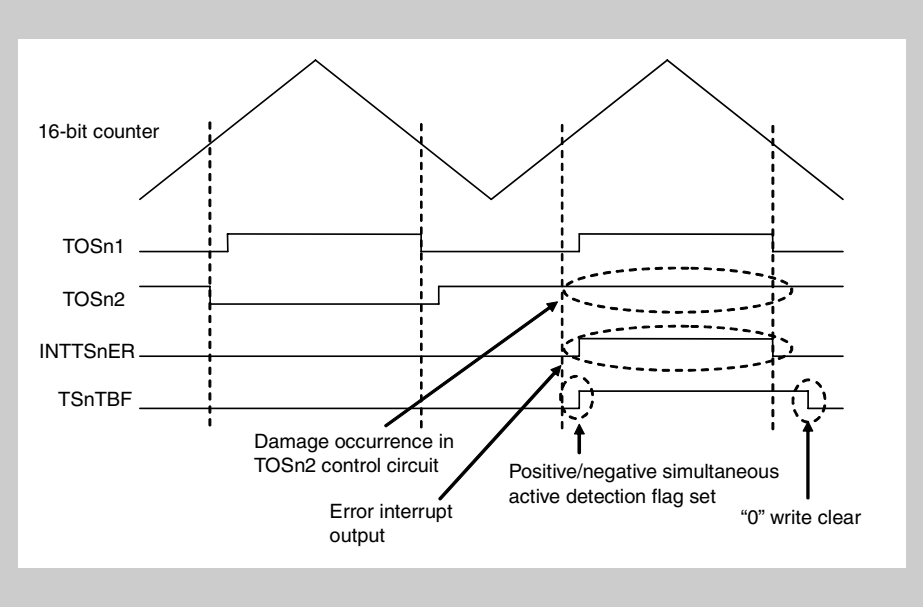


Figure 17-116 Error interrupt operation example

(12) Software output control function in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, timer output control can be performed through software control by using the TSnSOC and TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register. Output is switched immediately when TSnSOC is set to 1, as shown in *Figure 17-117 on page 904*. The dead time period is assured as long as the dead time is set. Output is retained until TSnSOC is cleared to 0. Then the function is switched to the output control by the high-accuracy T-PWM mode at the reload timing.
Refer to *"Software output function" on page 1014* for details on the software output control function.

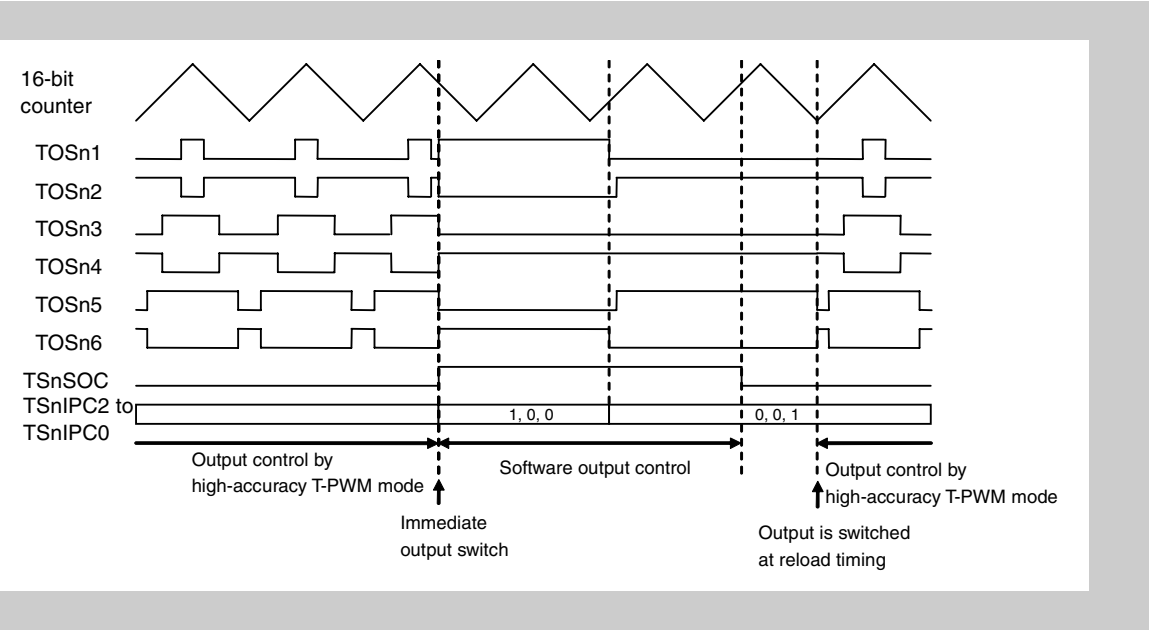


Figure 17-117 Software output control function switched from high-accuracy T-PWM mode

Caution Use the software output control function in the reload (batch rewrite) mode (TSnCMS of TSnOPT register = 0).

(a) Software output control procedure

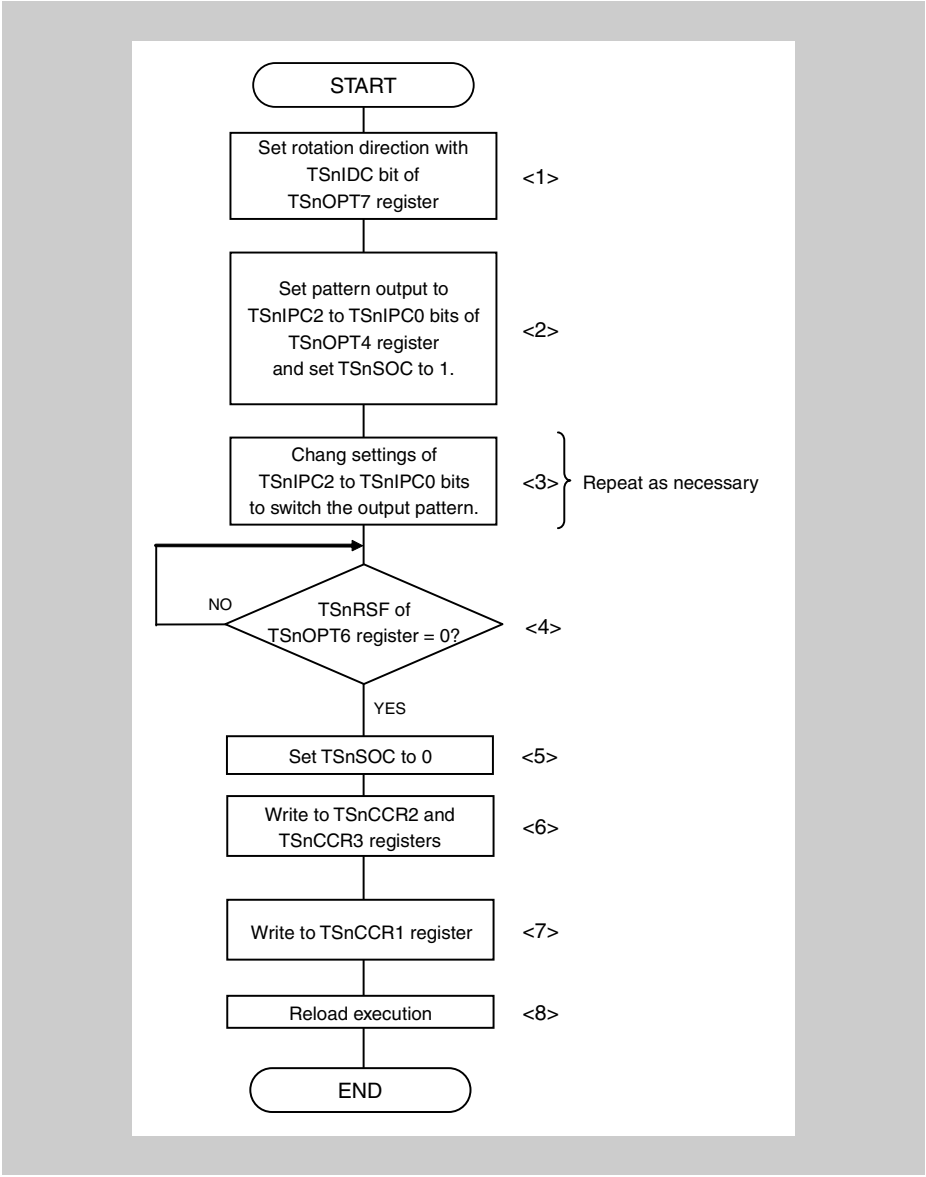


Figure 17-118 Software output control function processing flow (1/2)

The procedure for software output control processing is as follows.

<1> Set TSnIDC to determine the rotation direction. There is a 180° phase difference in the timer output between when TSnIDC = 0 and TSnIDC = 1. With the software output control function, the timer output is not changed only by rewriting of the bit.

<2> Set the output pattern so as to output to TSnIPC2 to TSnIPC0 and also set TSnSOC to 1 to provide software output.

<3> Change the output pattern setting of TSnIPC2 to TSnIPC0 to change the timer output. The settings of the following registers can be changed during software control.
TSnCE bit of TSnCTL0 register, TSnOPT1 to TSnOPT4 registers, TSnIDC and TSnTOS bits of TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, and TSnDTC1 register

<4> Check that the reload request flag (TSnRSF) is 0. If TSnRSF = 1, do not proceed to the next step before TSnRSF is cleared to 0.

<5> Software control release is started by clearing TSnSOC to 0 (Not yet released in this step).

<6> Set the compare register required after the release of software output control. Proceed to the next step if changing is not needed. Change the register that has the reload function in this step, if necessary.

<7> Write to the TSnCCR1 register to start reload.

<8> Reload is executed and software output is released.

Caution

1. Do not change TSnIPC2 to TSnIPC0 once after clearing TSnSOC to 0.

2. Be sure to execute reload after execution of procedures <4>, <5>, <6> and <7>. Software output cannot be released if reload cannot be executed.

(13) 180° excitation control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, the 180° excitation control function can be used by using the TSnADC and TSnPOT bits of the TSnOPT5 register, the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register and the TSnIDC bit of the TSnOPT7 register.
Output is switched immediately when TSnADC is set to 1, as shown in *Figure 17-119 on page 907*. The dead time period is assured as long as the dead time is set. Then, when a trigger is input to the output pattern switch trigger (TSnSTCI1 and TSnSTCI0 signals or TAPTSn2 to TAPTSn0 pins), the output is changed in accordance with the setting of the TSnPSC bit of the TSnOPT4 register. Output is retained until TSnADC is cleared to 0. Then the function is switched to the output control by the high-accuracy T-PWM mode, at the reload timing.
Be sure to use the 180° excitation control function in reload (batch rewrite) mode (TSnCMS of TSnOPT0 register = 0).
Refer to *"180° excitation control function" on page 1016* for details on 180° excitation control.

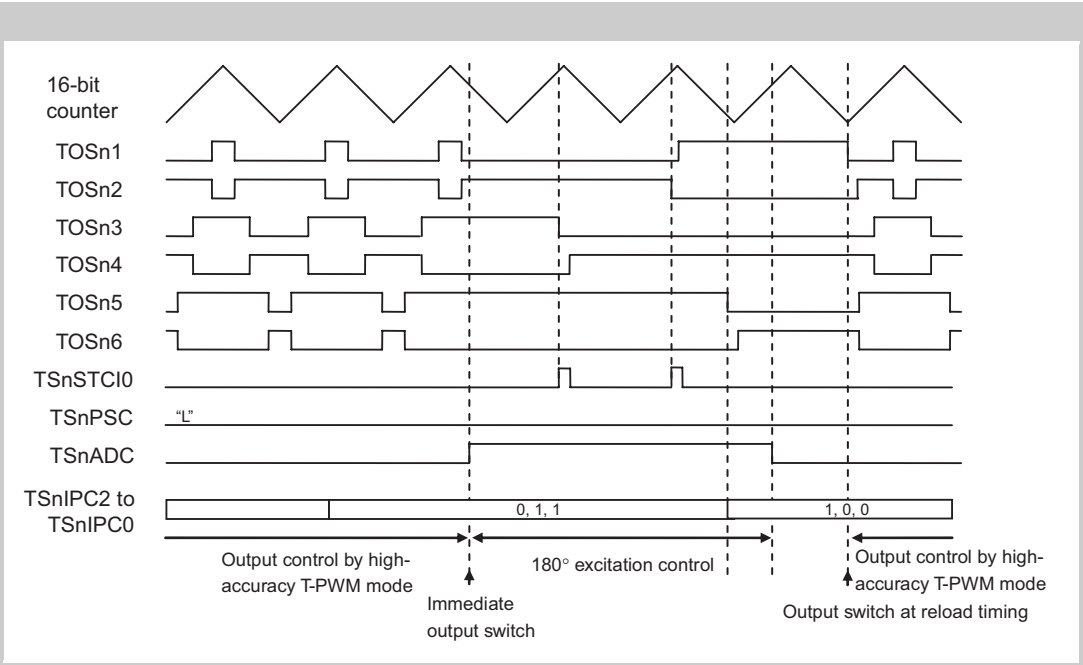


Figure 17-119 Example of switching 180° excitation control function in high-accuracy T-PWM mode

(a) 180° excitation control processing procedure

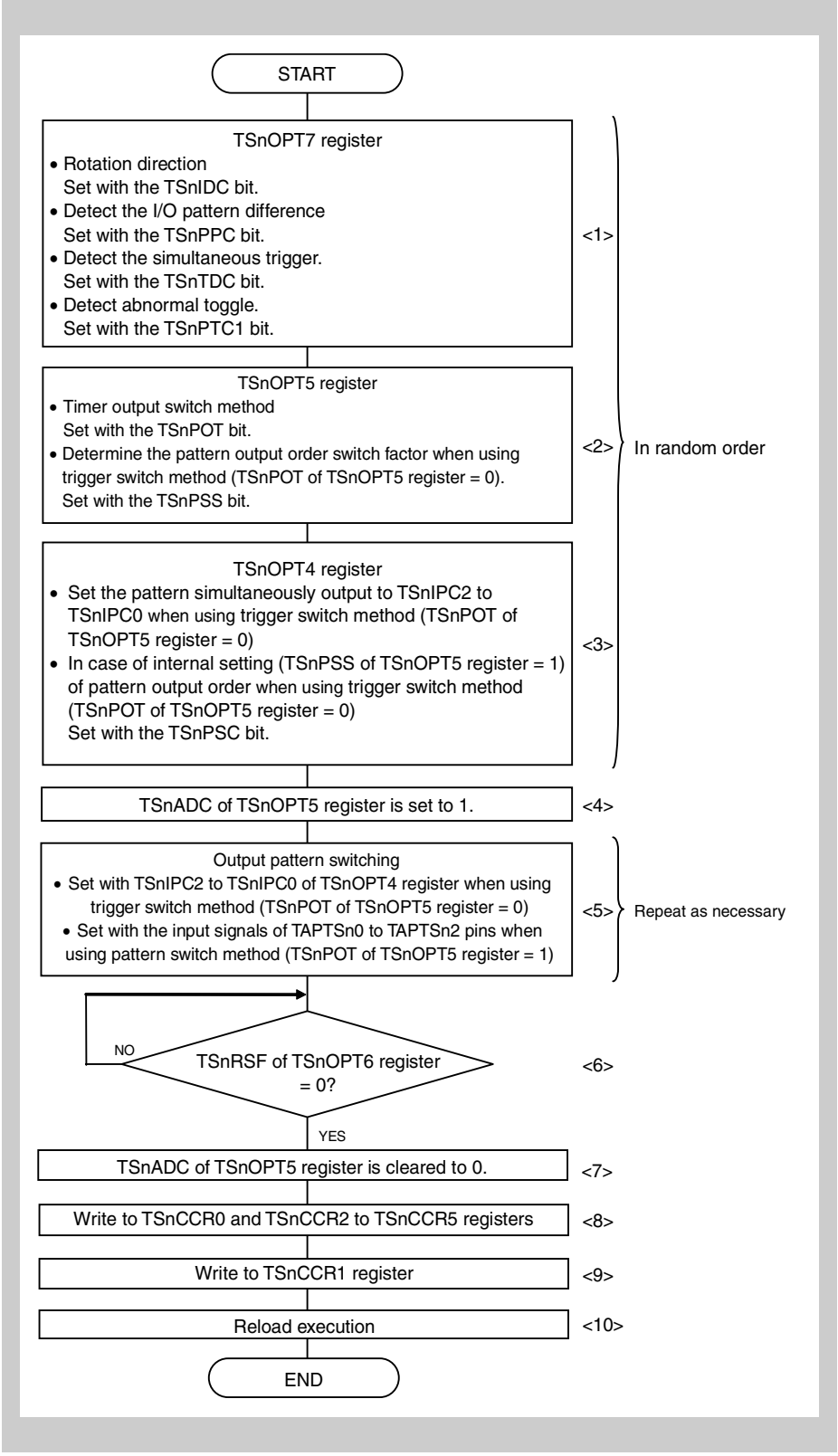


Figure 17-120 180° excitation control function processing flow

The procedure for 180° excitation control is as follows.

<1> TSnOPT7 register setting
Set a value to the TSnIDC bit to select the rotation direction.
Set a value to the TSnPPC bit to select enable (1) or disable (0) of the pattern phase difference detection flag (TSnPPF) of input (TAPTSn2 to TAPTSn0 pins) or output (TSnOPF2 to TSnOPF0 flags) with the trigger switch method (TSnPOT of TSnOPT5 register = 1).
Set a value to the TSnTDC bit to select enable (1) or disable (0) of the TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) with the trigger switch method.
Set values to the TSnPTC1 and TSnPTC0 bits to select enable (1) or disable (0) of the TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF).

<2> TSnOPT5 register setting
Set a value to the TSnPOT bit to select the pattern output trigger. The pattern switch method with which the output is switched with the input pattern of the TAPTSn2 to TAPTSn0 pins can be enabled by clearing TSnPOT to 0, and the trigger switch method with which the output is switched at a rising edge of the TSnSTCI0 and TSnSTCI1 signals can be enabled by setting TSnPOT to 1.
Set a value to the TSnPSS bit to select the pattern output order switch factor.

<3> TSnOPT4 register setting
Set a value to the TSnPSC bit to select the pattern output order for when TSnPSS of the TSnOPT5 register is 1 by using the trigger switch method.
Set values to the TSnIPC2 to TSnIPC0 to select the initial pattern for the trigger switch method.

<4> Start the 180° excitation control function by setting TSnADC to 1.

<5> Output pattern switching

– Pattern switch method
With the pattern switch method, the output is switched with the input pattern of TAPTSn2 to TAPTSn0 pins. However, the previous output level is held until [1, 1, 1]0 or [0, 0, 0] is input to the TAPTSn2 to TAPTSn0 pins.

– Trigger switch method
With the trigger switch method, the output is switched at a rising edge of the TSnSTCI0 and TSnSTCI1 signals. Set the appropriate value to TSnIPC2 to TSnIPC0 to forcibly change the timer output.
Writing to the TSnIPC2 to TSnIPC0 is prior to the rising edge of the TSnSTCI0 and TSnSTCI1 signals if both occur simultaneously.

– Switch method can be changed during operation.

– The registers that can be rewritten during 180° excitation control function operation are as follows.
TSnCE bit of TSnCTL0 register, TSnOPT1 to TSnOPT5 registers, TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register

Caution Setting TSnSOC of the TSnOPT4 register to 1 and TSnADC of TSnOPT5 register to 1 is prohibited.

Note <1> to <3> are in random order.

<6> Check that the reload request flag (TSnRSF) is 0. If TSnRSF = 1, do not proceed to the next step before TSnRSF is cleared to 0.

<7> Release of 180° excitation control is started by clearing TSnADC to 0 (Not yet released in this step).
After TSnADC is changed to 0, output is switched at the switch timing of the pattern/trigger switch method until reload is executed. The output level can be changed forcibly by writing to TSnIPC2 to TSnIPC0.

<8> Set the compare register required after the release of 180° excitation control. Proceed to the next step if changing is not needed. Change the register that has the reload function in this step if necessary.

<9> Write to the TSnCCR1 register to start reload.

<10>Reload is executed and 180° excitation control is released.

Caution Be sure to execute reload after execution of steps <6>, <7>, <8> and <9>. 180° excitation control cannot be released if reload cannot be executed.

(b) 180° excitation control switch timing from high-accuracy T-PWM mode

Output is switched immediately when switching from the high-accuracy T-PWM mode to 180° excitation control. Dead time control by hardware is inserted if the on-going output level is reversed. The output level is held until the reload is performed when switching from 180° excitation control to the high-accuracy T-PWM mode.

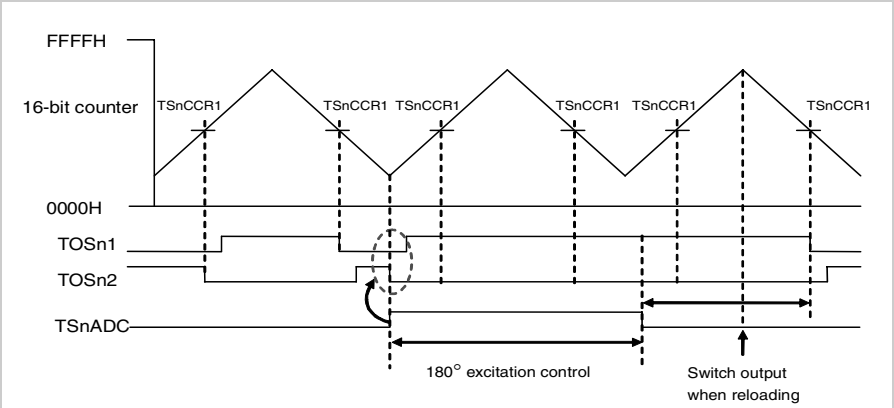


Figure 17-121 Output when switching between high-accuracy T-PWM Mode and 180° excitation control (in case of output reverse)

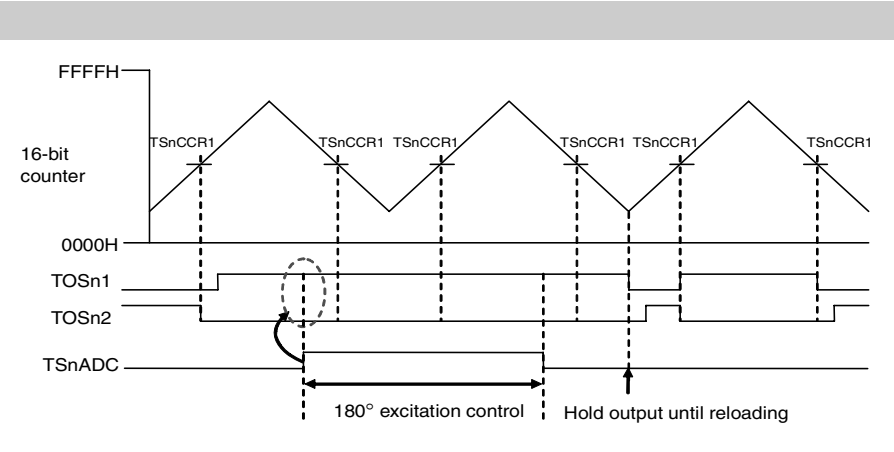


Figure 17-122 Output when switching between high-accuracy T-PWM Mode and 180° excitation control (in case of output not reversed)

The output switched during the dead time count is set without waiting for the dead time count to finish in case of the same phase, or is set after the dead time count finishes in case of the negative phase.

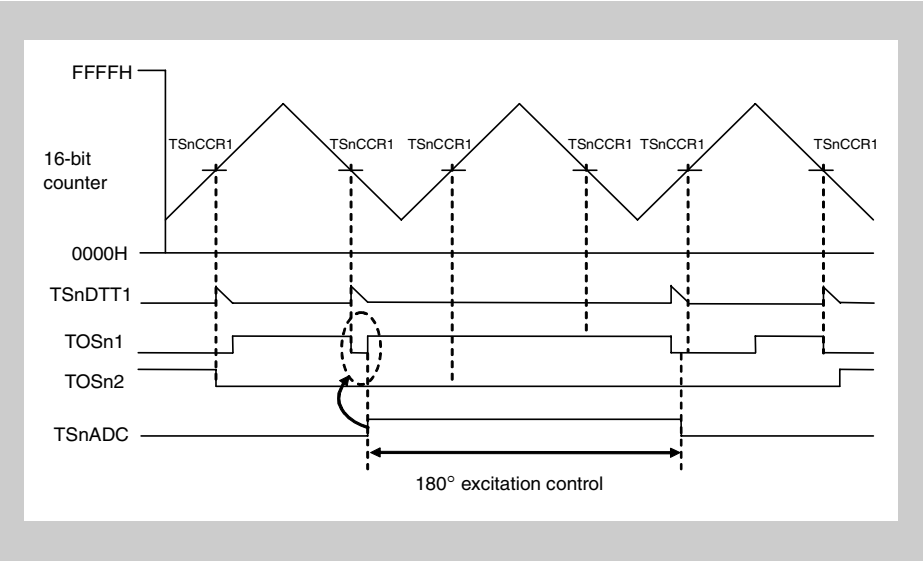


Figure 17-123 Output when switching between high-accuracy T-PWM mode and 180° excitation control (during dead time count)

(14) Semi-automatic driving system using high-accuracy T-PWM mode

This section explains the system in which the semi-automatic driving function is used in the high-accuracy T-PWM mode. *Figure 17-124 on page 913* shows the timing. The 180° excitation control function is automatically enabled by setting TSnADC of the TSnOPT5 register to 1. As shown here, the output pattern is switched by the TSnSTCI0 signal as a trigger in the 180° excitation control function, and the dead time set with the TSnDTC0 and TSnDTC1 registers is always added when switching.
Output is switched immediately when switching from the high-accuracy T-PWM mode to 180° excitation control, but control is switched after waiting for the reload timing when switching from 180° excitation control to the high-accuracy T-PWM mode.

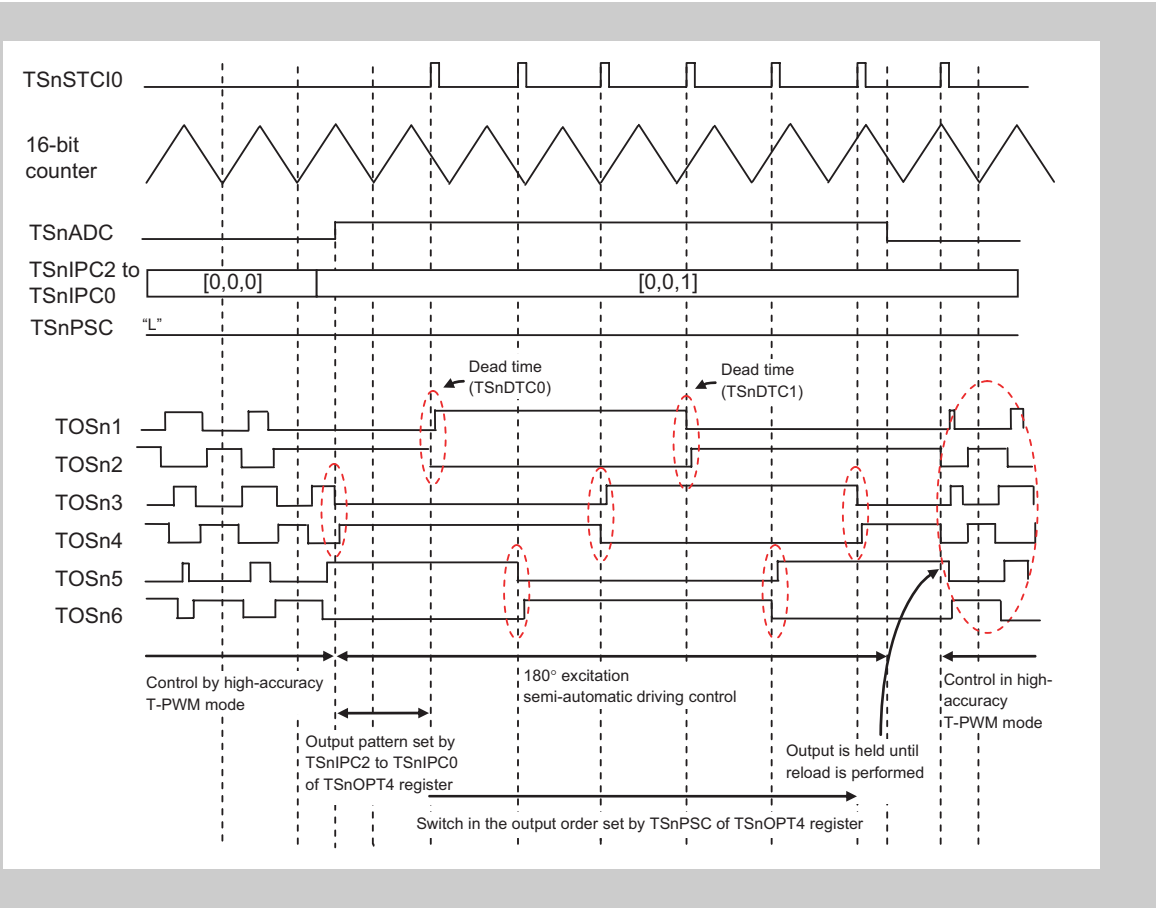


Figure 17-124 Output example of semi-automatic driving system using high-accuracy T-PWM mode

Note Setting example
TSnCTL1 register: TSnMD3 to 0 = [1, 0, 0, 0], TSnIOC0 register: TSnOE6 to 1 = [1, 1, 1, 1, 1, 1, 1, 1], TSnOL6 to 1 = [0, 0, 0, 0, 0, 0], TSnOPT4 register: TSnIPC2 to 0 = [0, 0, 1], TSnPSC = 0, TSnOPT5 register: TSnADC = 1

17.10.9 PWM mode with dead time

(1) Overview of PWM mode with dead time

In the PWM mode with dead time, 6-phase PWM is generated to output from the TOSn1 to OSn6 pins by using the saw-tooth wave operation of the 16-bit counter and four 16-bit compare registers. The maximum value of the 16-bit counter is set with the TSnCCR0 register. The duty of the U, V, and W-phase voltage data signal is set with the TSnCCR1 to TSnCCR3 registers. Dead time is set with the TSnDTC0 and TSnDTC1 registers. The dead time from the negative phase to the positive phase and the dead time from the positive phase to the negative phase can be set with TSnDTC0 and TSnDTC1, respectively. The 16-bit counter counts up by setting 0000H as the minimum value. When matching with the maximum value (cycle), the 16-bit counter is cleared (0000H), and continues counting up.
The 10-bit dead time counter (TSnDTT1 to TSnDTT3) reloads the setting value of the TSnDTC0 and TSnDTC1 registers upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR3 registers to count down.
Compare match interrupts (INTTSnCC0 to INTTSnCC3) occur upon respective match between the values of the 16-bit counter and TSnCCR0 to TSnCCR3 registers.

Note PWM mode with dead time is valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 0, 1].

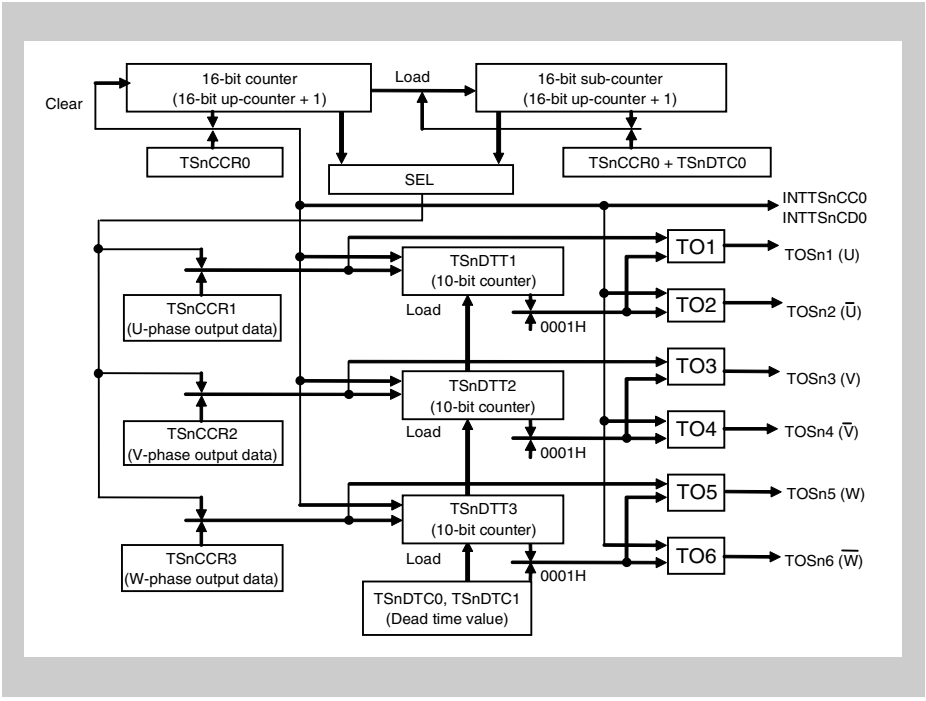


Figure 17-125 Block diagram in PWM mode with dead time

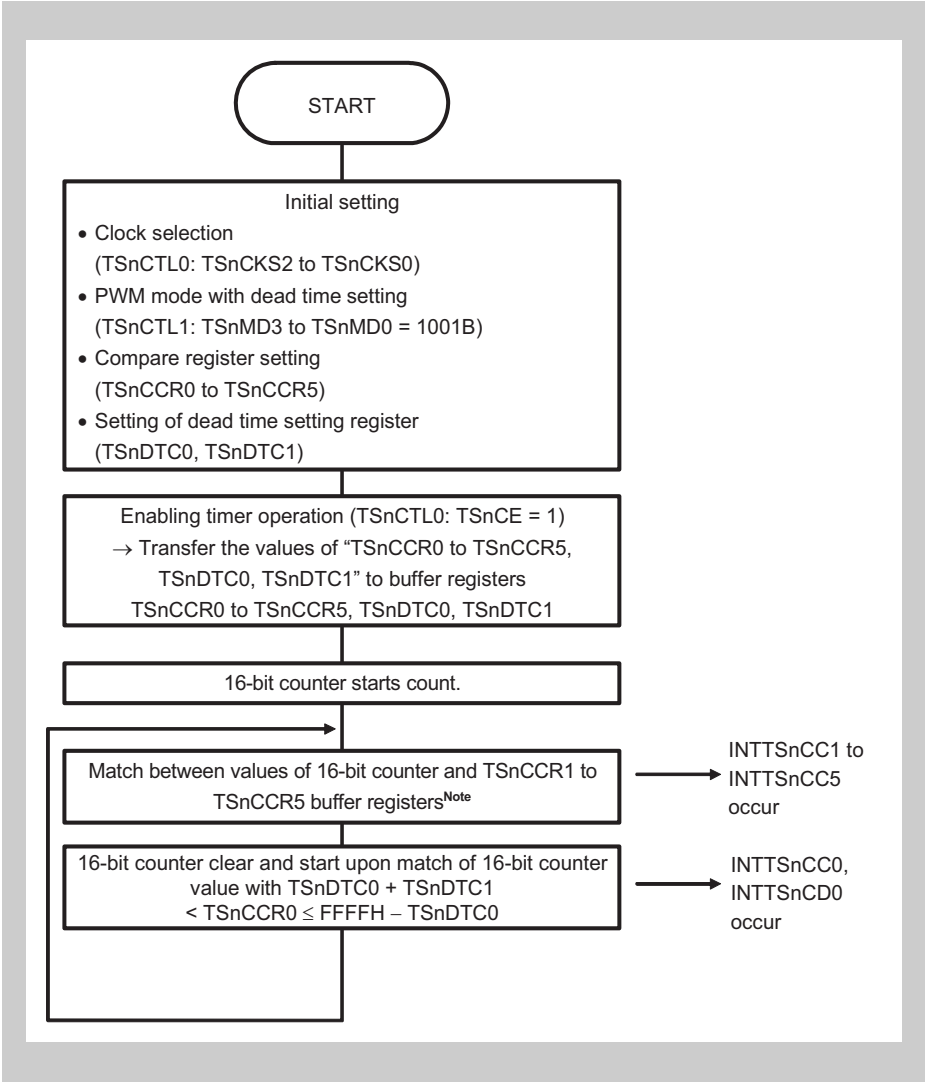


Figure 17-126 Basic operation flow in PWM mode with dead time

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

(2) PWM Mode with Dead Time Operation List

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	PWM duty
TSnCCR4, TSnCCR5	Reload	Possible	PWM duty (Selectable as A/D conversion trigger)
TSnDTC0, TSnDTC1	Reload	Conditionally possible[Note]	Cycle, dead time

Note Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* for details.

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Toggle output by TSnCCR0 compare match
TOSn1	PWM output (with dead time) by TSnCCR1 compare match
TOSn2	Negative phase output (with dead time) to TOSn1
TOSn3	PWM output (with dead time) by TSnCCR2 compare match
TOSn4	Negative phase output (with dead time) to TOSn3
TOSn5	PWM output (with dead time) by TSnCCR3 compare match
TOSn6	Negative phase output (with dead time) to TOSn5
TOSn7	Pulse output by A/D conversion trigger

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing with INTTSnCC0)
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	When 16-bit counter switches from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between 16-bit counter and TSnCCR1 to TSnCCR5

(f) Output condition

Output	Condition
Duty 0% output	TSnCCRm = 0000H
Duty 100% output	TSnCCR0 + TSnDTC0 + 1 = TSnCCRm
Dead time check	TSnDTC0, TSnDTC1

Note 1. "–" indicates an unused function in the PWM mode with dead time.

2. m = 1 to 3, n = 0,1

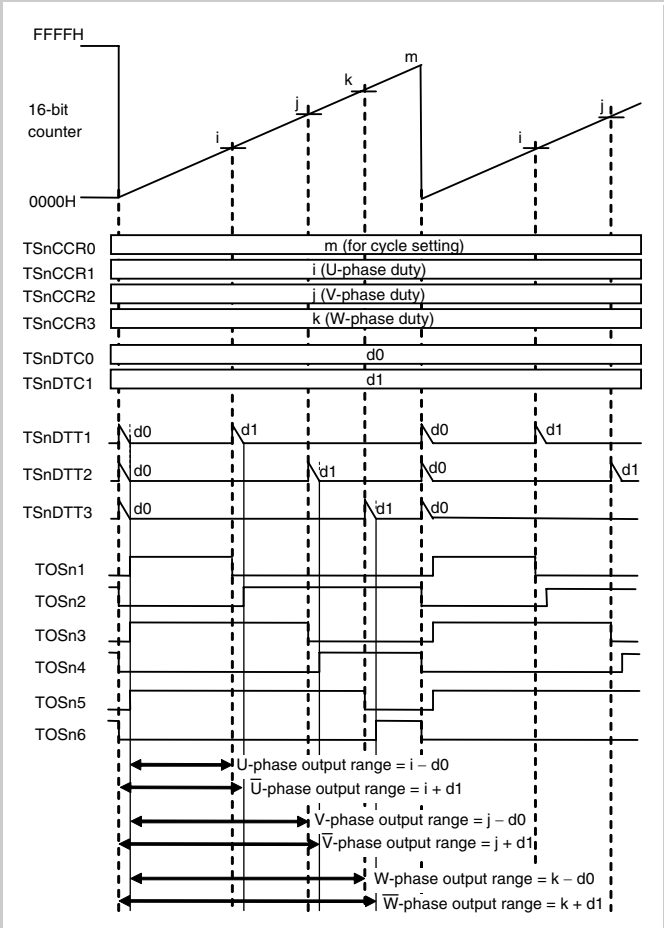


Figure 17-127 Output waveform example in PWM mode with dead time

Caution 1. The maximum value settable to the TSnCCR1 to TSnCCR3 registers is "TSnCCR0 + TSnDTC0 + 1".

2. PWM output is provided with duty 0% when 0000H is set to the TSnCCR1 to TSnCCR3 registers.

3. PWM output is provided with duty 100% when “TSnCCR0 + TSnDTC0 + 1” is set to the TSnCCR1 to TSnCCR3 registers.
4. Set the TSnCCR0 register to satisfy the condition “TSnDTC0 + TSnDTC1 < TSnCCR0 ≤ FFFFH − TSnDTC0”.

(3) Setting of PWM mode with dead time

(a) Mode setting

The PWM mode with dead time is set by setting TSnMD4 to TSnMD0 of the TSnCTL1 register to [1, 0, 0, 1].

(b) Output level/output enable setting

Output level/output enable is set by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register.
Toggle output is provided from the TOSn0 pin upon the cycle match (match between the 16-bit counter and TSnCCR0 register).
Pulse output by A/D conversion trigger is provided from the TOSn7 pin. Set this pin as needed.

(c) Error interrupt occurrence enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1 when the simultaneous positive/negative phase state is detected.
A pin that detects simultaneous active state is set by setting the TSnTBA2 to TSnTBA0 bits of the TSnIOC4 register. In the PWM mode with dead time, positive and negative phases do not become active even if any value is set to the TSnCCR0 to TSnCCR3 registers.

(d) Interrupt and thinning out function setting

A peak interrupt (INTTSnCD0) occurs upon a match between the TSnCCR0 register and 16-bit counter. (Trough interrupts do not occur). Set TSnICE of the TSnOPT1 register to 1 to output peak interrupts. To use the thinning out function for peak interrupt, execute setting with the TSnID4 to TSnID0 bits of the TSnOPT1 register.

(e) Reload thinning out function setting

Set TSnRDE of the TSnOPT2 register to 1 to set reload timing to the same timing as interrupt timing. Reload is performed when TSnRTE of the TSnOPT1 register is set to 1.

(f) A/D conversion trigger output setting

A/D conversion trigger 0 (TSnADTRG0 signal) is set by setting the TSnAT04, TSnAT02, TSnAT01 bits of the TSnOPT2 register. The TSnAT04, TSnAT02, and TSnAT01 bits specify whether to enable/disable A/D conversion trigger output upon match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at the peak interrupt (INTTSnCD0) of the 16-bit counter or 16-bit sub-counter.
A/D conversion trigger 1 (TSnADTRG1 signal) is set by setting the TSnAT14, TSnAT12, and TSnAT11 bits of the TSnOPT3 register. The TSnAT14, TSnAT12, and TSnAT11 bits specify whether to enable/disable A/D conversion trigger output upon match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at the peak interrupt (INTTSnCD0) of the 16-bit counter or 16-bit sub-counter.
Compare values are set with the TSnCCR4 and TSnCCR5 registers.
Thinning out function is enabled for the TSnADTRG0 and TSnADTRG1 signals.
No thinning out, 1-thinning out, 3-thinning out and 7-thinning out setting can be executed with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

Caution

1. Setting TSnAT07, TSnAT06, TSnAT05, TSnAT03, TSnAT00 of the TSnOPT2 register to 1 is prohibited.
2. Setting TSnAT17, TSnAT16, TSnAT15, TSnAT13, TSnAT10 of the TSnOPT3 register to 1 is prohibited.

(g) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.
Dead time can be calculated with the following equation.
16-bit counter operation clock cycle × TSnDTC0
16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of the TOSn2, TOSn4, and TOSn6 pins to the active state of the TOSn1, TOSn3, and TOSn5 pins can be set with the TSnDTC0 register.
The time for the change from the inactive state of the TOSn1, TOSn3, and TOSn5 pins to the active state of the TOSn2, TOSn4, and TOSn6 pins can be set with the TSnDTC1 register.

(h) Duty (PWM width) setting

U, V and W-phase duties are set with the TSnCCR1 to TSnCCR3 registers respectively. The setting range of the TSnCCR1 to TSnCCR3 registers is as follows.

$0000H \leq TSnCCR1 - TSnCCR3 \leq TSnCCR0 + TSnDTC0 + 1$

Set the TSnCCR0 register to satisfy the condition “$TSnDTC0 + TSnDTC1 < TSnCCR0 \leq FFFFH - TSnDTC0$”.

(4) Operation in PWM mode with dead time

Figure 17-128 on page 921 shows the timing when setting TSnCCR0 = 0007H, TSnDTC0 = 0002H, TSnDTC1 = 0002H, and setting the TSnCCR0 register to 0000H to 0007H (in part).
The PWM width increases/decreases by 1 count clock if the compare value of the TSnCCR1 register increases/decreases. Arrows indicate the increase/decrease by the TSnDTC1 + 1 count clock in *Figure 17-128 on page 921*.
This occurs when the value is rewritten to “TSnCCR1 + 0001H” which will be “TSnDTC1 < TSnCCR1” because the dead time control is valid.

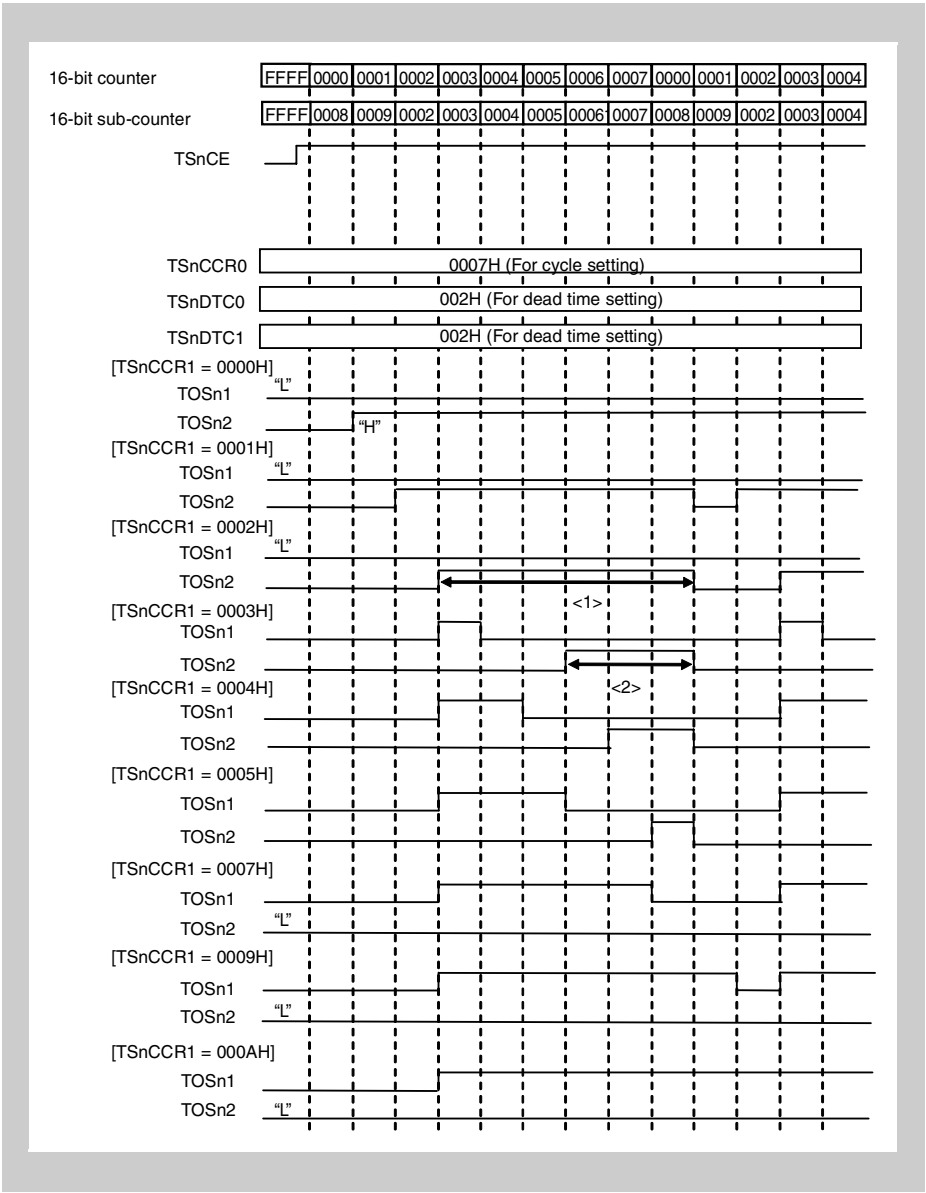


Figure 17-128 Timer output example in case of setting TSnCE = 1 (initial setting in PWM mode with dead time)

<1> PWM width without dead time

<2> PWM width decreases by TSnDTC1 + 1 because the PWM width with dead time is “TSnCCR1 = TSnDTC0 + 0001H”.

(5) Dead time control in PWM mode with dead time

In the PWM mode with dead time, the duty setting registers are TSnCCR1 to TSnCCR3, and the cycle setting register is TSnCCR0.
Duty variable type 6-phase PWM waveform output is provided by using these four registers.
There are three 10-bit down-counters that synchronously operate with the count clock of the 16-bit counter, and dead time setting registers (TSnDTC0, TSnDTC1) in order to achieve dead time control. Dead time from the negative phase changes to inactive until the positive phase changes to active is set by the TSnDTC0 register, and the dead time from the positive phase changes to inactive until the negative phase changes to active is set by the TSnDTC1 register.
Then, the output waveform in case of TSnDTC0 = x, TSnDTC1 = y is shown in *Figure 17-129 on page 922*.

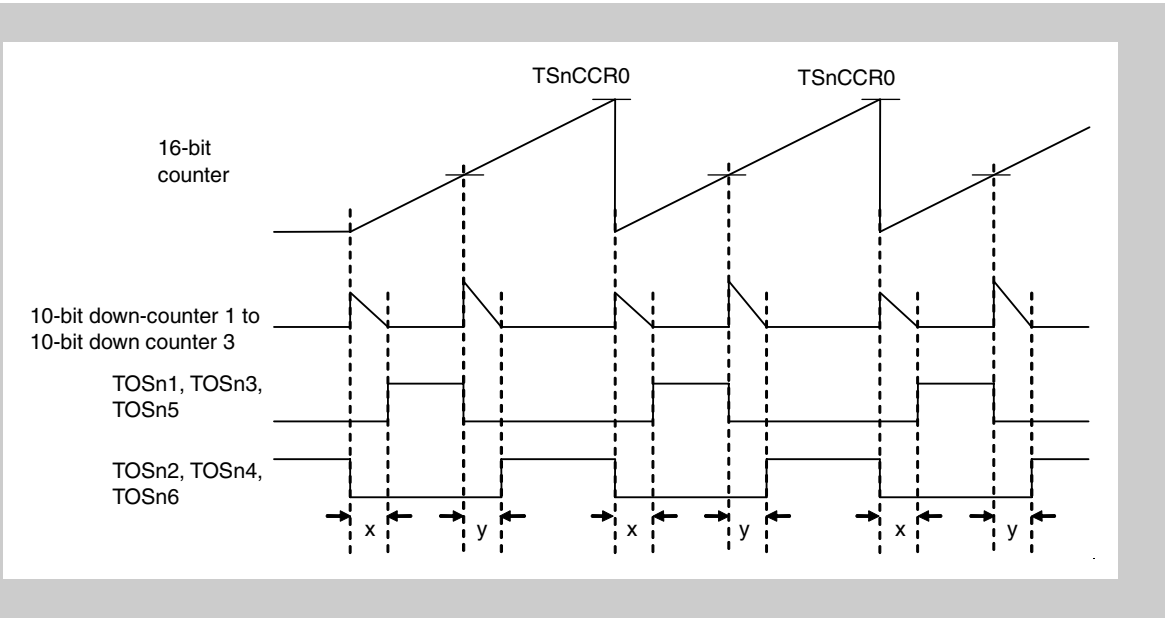


Figure 17-129 Output waveform example in PWM mode with dead time

(6) Error interrupt (INTTSnER) in T-PWM mode with dead time

The positive/negative phase simultaneous active state detection function is enabled in the T-PWM mode with dead time. Error interrupts (INTTSnER) do not occur in the high-accuracy T-PWM mode. In case of occurrence, the internal circuit may be damaged.

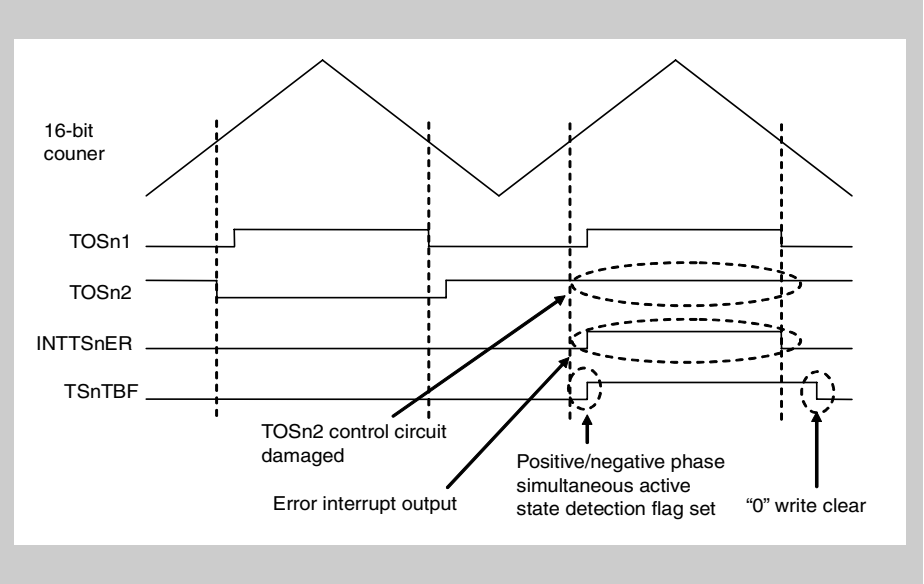


Figure 17-130 Error interrupt operation example

(7) Caution about interrupt timing in T-PWM mode with dead time

The interrupt output timing differs between “NTTSnCC0, INTTSnCC4, INTTSnCC5” and “INTTSnCC1, INTTSnCC2, INTTSnCC3”. Therefore, interrupt timing will not be the same even if the same value is set.

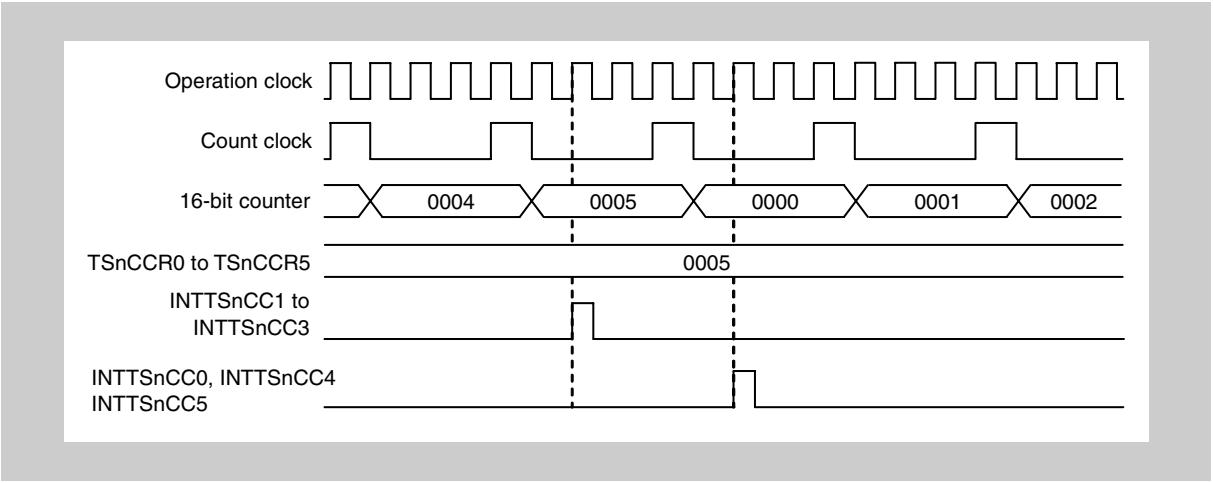


Figure 17-131 Interrupt timing example in T-PWM mode with dead time

(8) Software output control function in PWM mode with dead time

Timer output control can be executed with software control by using the TSnSOC and TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.
Output is switched immediately at the time of setting TSnSOC = 1 as shown in *Figure 17-132*. Dead time period is assured by setting the dead time.
Output is held at the time of setting TSnSOC = 0. Then the function is switched to the output control in the PWM mode with dead time at reload timing occurrence.
Refer to *"Software output function" on page 1014* for details on the software output control function.

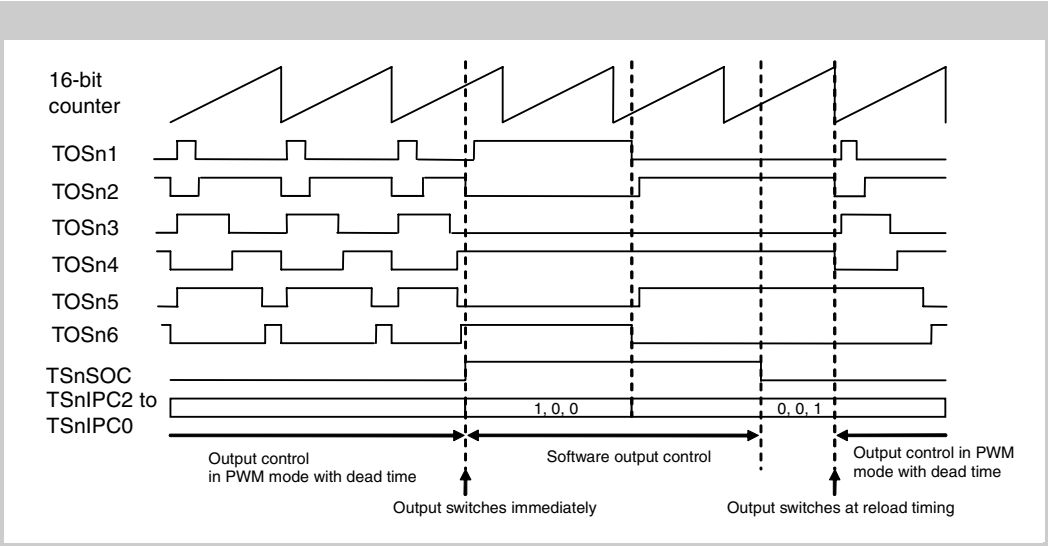


Figure 17-132 Software output control function switch example from PWM mode with dead time

(a) Software output control procedure

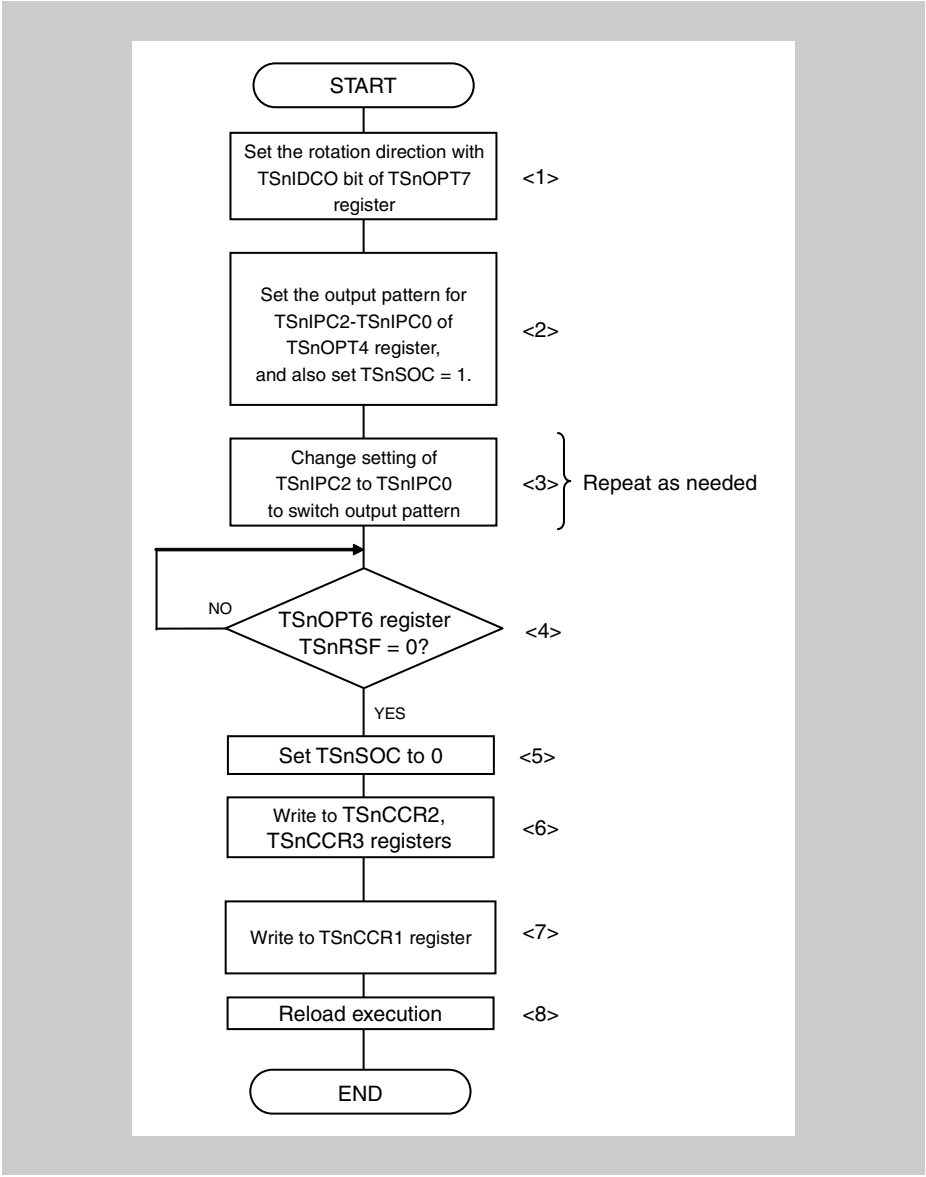


Figure 17-133 Processing flow in software output control

The software output control processing procedure is described below.

<1> Set TSnIDC in order to determine rotation direction. There is a 180° phase difference in the timer output between when TSnIDC = 0 and TSnIDC = 1. In the software output control function, the timer output is not changed only by rewriting the bit.

<2> Set the pattern to be output to TSnIPC2 toTSnIPC0 and also set TSnSOC to 1 in order to provide software output.

<3> Change the output pattern setting of TSnIPC2 to TSnIPC0 to change the timer output.
The registers that can be changed during the software control are as follows.
TSnCE bit of the TSnCTL0 register, TSnOPT1 to TSnOPT4 registers, TSnIDC, TSnTOS bits of TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register

<4> Check that the reload request flag (TSnRSF) is set to 0. In the case of TSnRSF = 1, do not move to the next procedure until the setting of TSnRSF = 0.

<5> Software control release is started by setting TSnSOC = 0 (Not yet released in this step).

<6> Set the compare register which is necessary after the release of software output control. Move to the next procedure when any change is not needed. Change the register with the reload function in this step if necessary.

<7> Write to the TSnCCR1 register to activate reload.

<8> Execute reload to release the software output.

Caution

1. Any change of TSnIPC2 to TSnIPC0 is prohibited after setting TSnSOC to 0.
2. Be sure to execute reloading after the execution of procedures <4>, <5>, <6> and <7>. Software output cannot be released if reload cannot be executed.

17.10.10 120° excitation mode

(1) Overview of 120° excitation mode

In the 120° excitation mode, 120° excitation control is performed by detecting the detection of external input (pattern: TAPTSn2 to TAPTSn0 pins, trigger: TSnSTCI0 signal) and by the output control of timer output (pins TOSn1 to TOSn6).
Motor systems that can be controlled in the 120° excitation mode are as follows.

- System where a hall sensor is connected at 120° intervals against the motor rotation
- System where normal/reverse rotation is possible by outputting, responding to the hall sensor input
  A concrete output example of normal/reverse rotation is shown in *Figure 17-134 on page 928*.

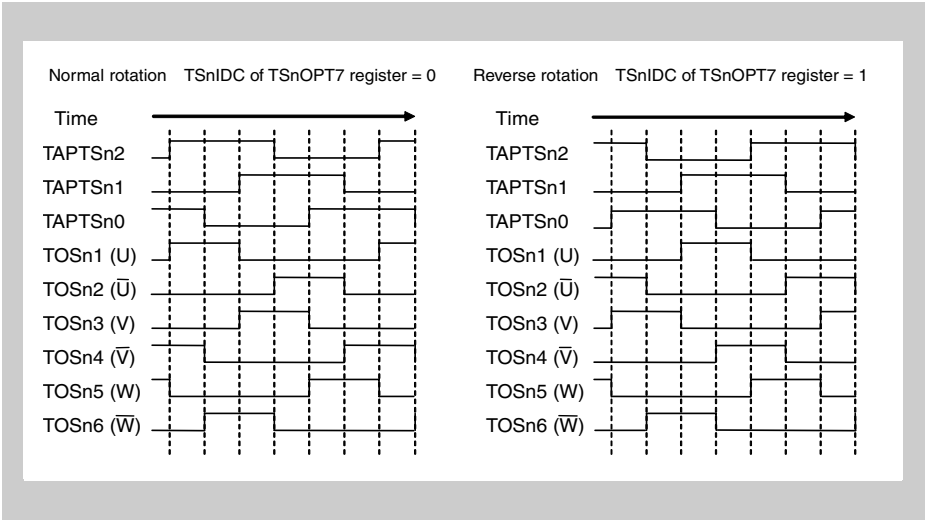


Figure 17-134 Normal/Reverse rotation example

Note
1. TSnOPT4 register: TSnSOC = 0,
   TSnOPT5 register: TSnADC, TSnPOT = 0
2. The 120° excitation mode is valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 1, 0].

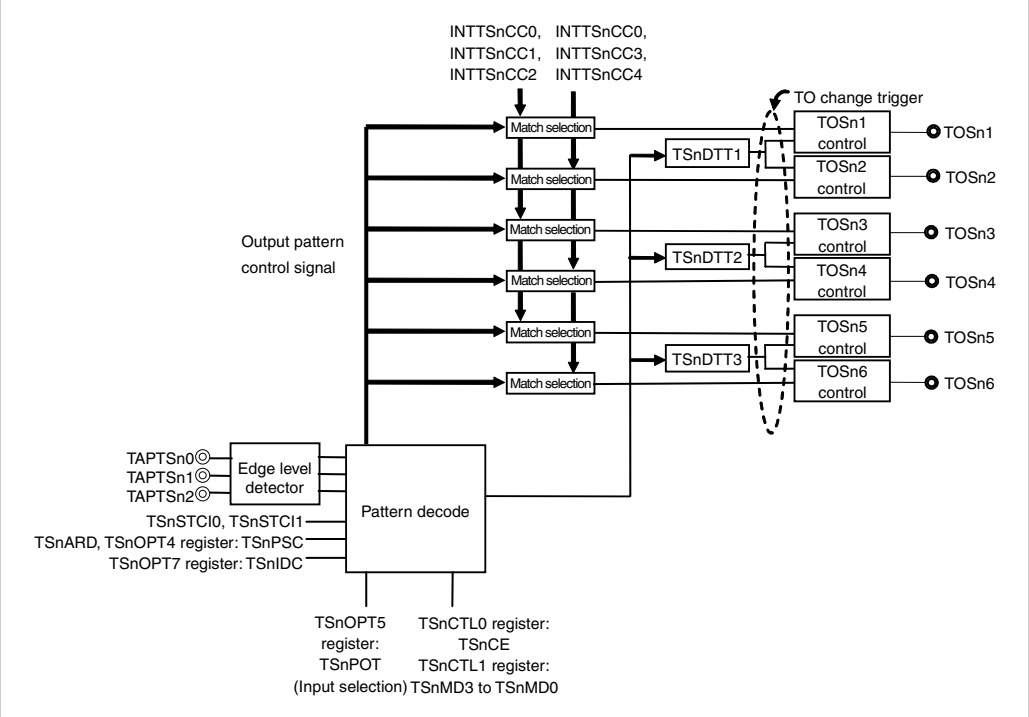


Figure 17-135 Configuration diagram in 120° excitation mode

Table 17-21 Output patten of TAPTSn2 to TAPTSn0 pins (normal rotation)

Output Pin	TAPTSn2 to TAPTSn0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 1	Note 1
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 1	Note 1
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 1	Note 1
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 1	Note 1
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 1	Note 1
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 1	Note 1

Note 1. The output level of the TOSn0 to TOSn6 pins remains the level before change when changed to [0, 0, 0] or [1, 1, 1].

2. PWM1 to PWM4 indicate the PWM output of the TSnCCR1 to TSnCCR4 registers.

3. NACT indicates the inactive level output.

There are the following two methods according to the external input used in the 120° excitation mode.

- Patten switch method: 3 patterns of input (hall sensor, etc.)
- Trigger input switch method: Trigger input (encoder circuit, offset, etc.)

(a) Pattern switch method (TSnPOT of TSnOPT5 register = 0)

Level detection is performed for the TAPTSn2 to TAPTSn0 pins (three patterns of input from hall sensor) to decode the signals after the level detection. PWM output (which is to be determined with the values of the TSnCCR1 to TSnCCR4 registers) of the TOSn1 to TOSn6 pins is selected by judging the decode result. PWM output is generated by decoded signals and compare match interrupts (INTTSnCC1 to INTTSnCC4) to control the output. Dead time control is performed by the operation of the dead time counter at respective phase signal fall timing to ensure dead time insertion.
With the pattern switch method, the decoded pattern is output by using the data from input pattern (TAPTSn2 to TAPTSn0 pins), current direction control bit (TSnIDC of the TSnOPT7 register), TAPTSn2 to TAPTSn0 order detection flag (TSnTSF of the TSnOPT5 register). *Figure 17-136 on page 931* and *Figure 17-137 on page 932* show the timer output when the TAPTSn2 to TAPTSn0 pins are changed. The output pattern is switched to the pattern corresponding to the input if any error occurs in the input pattern when switching from the input pattern 1 to the input pattern 4.
The pattern set with the input level of the TAPTSn2 to TAPTSn0 pins, the TSnIDC and TSnARD signals, (TSnPSS of TSnOPT5 register = 0), and the TSnPSC bit of the TSnOPT4 register (TSnPSS of TSnOPT5 register = 1) is output. Output pattern is determined with the TSnTSF flag instead of the TSnPSC bit if the value of the TSnTSF flag is determined.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0, TSnOPT5 register: TSnPOT = 0, and TSnOPT7 register: TSnIDC = 0.

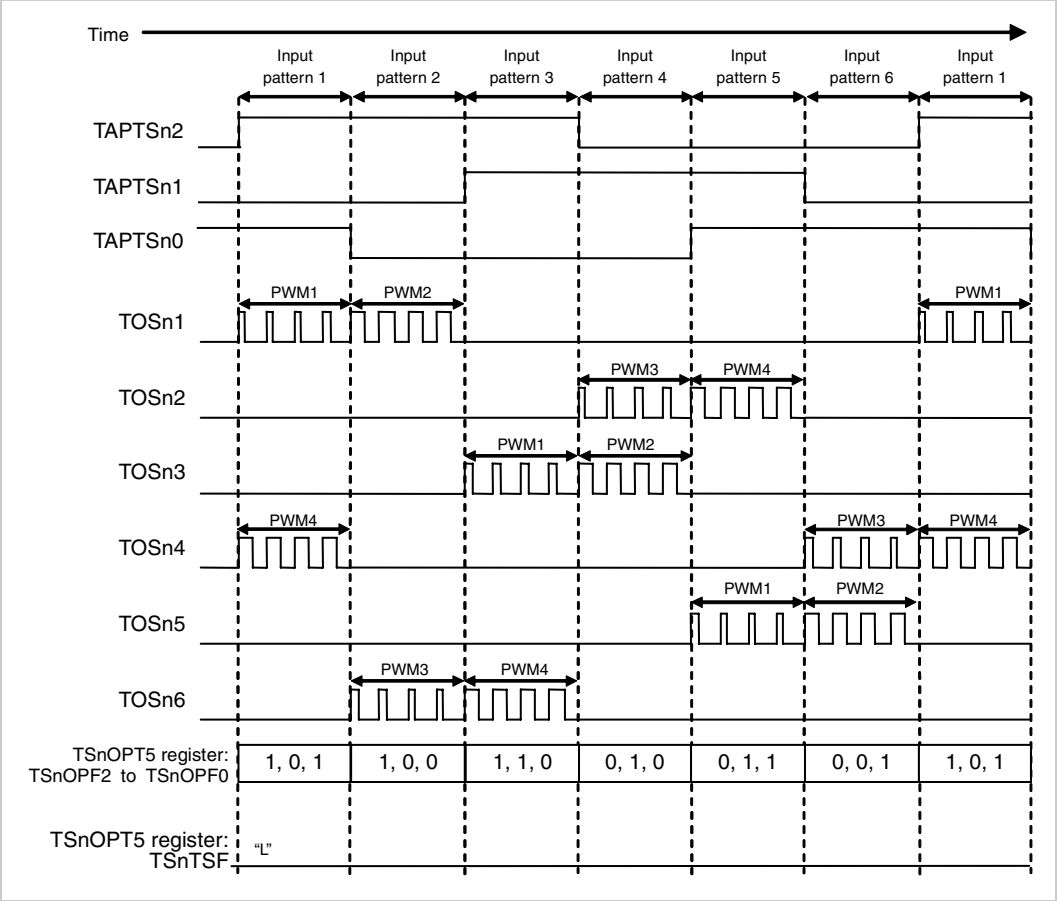


Figure 17-136 120° excitation mode operation example in use of pattern switch method (normal rotation)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 0, and TSnOPT7 register: TSnIDC = 1.

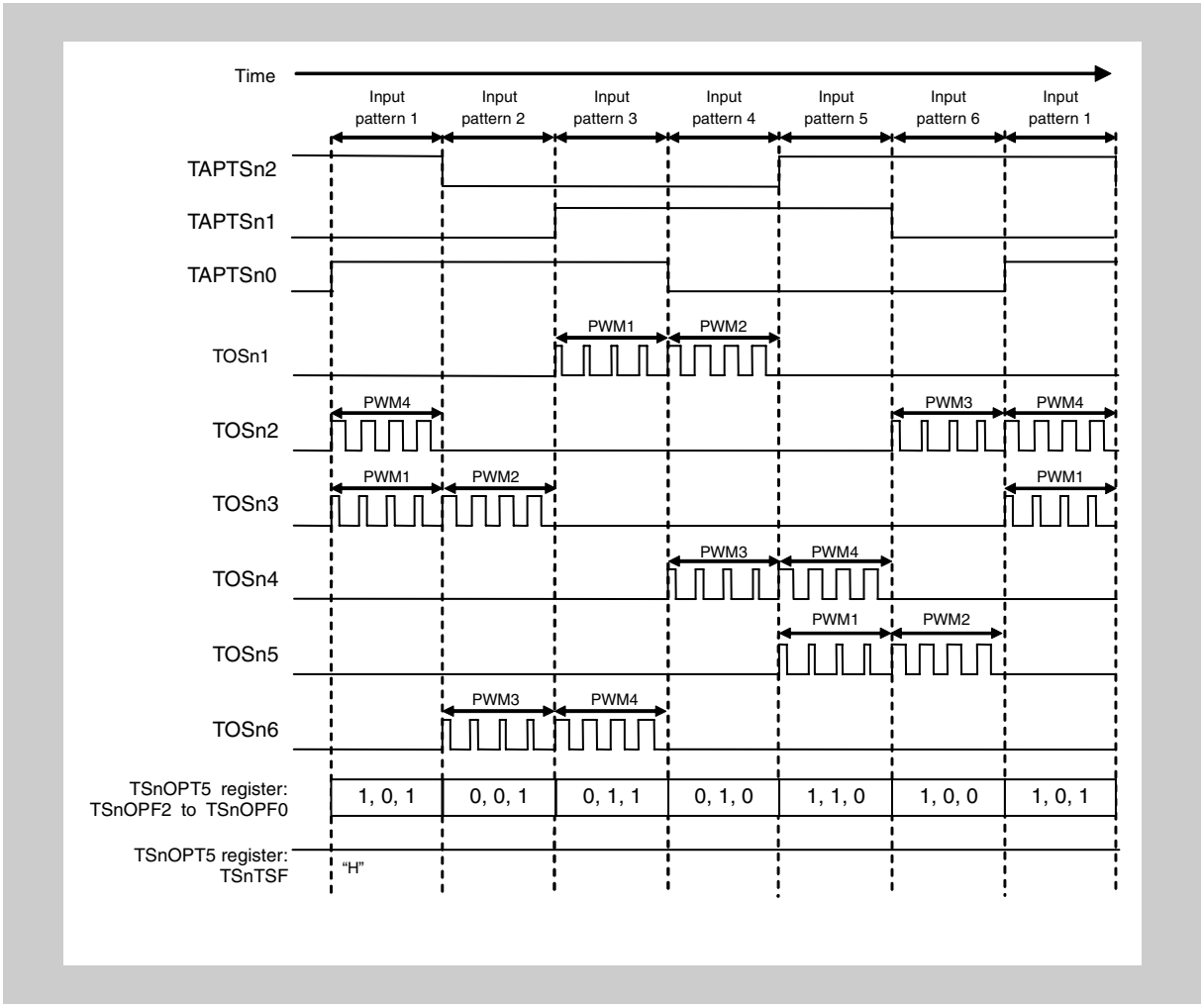


Figure 17-137 120° excitation mode operation example in use of pattern switch method (reverse rotation)

(b) Trigger switch method

With the trigger input switch method, rises of the TSnSTCI0 and TSnSTCI1 signals are detected to generate the output switch timing.
The timer output initial pattern is set with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. The output patterns after the initial pattern are determined with the TSnIPC2 to TSnIPC0 bits, rotation direction or the TSnIDC bit of the TSnOPT7 register. There are four patterns which can be output in combination of rotation direction and the TSnIDC bit.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
1	0	Determine rotation direction by using input signal to TSnARD
	1	Determine rotation direction by using TSnPSC bit

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1, TSnOPT7 register: TSnIDC = 0

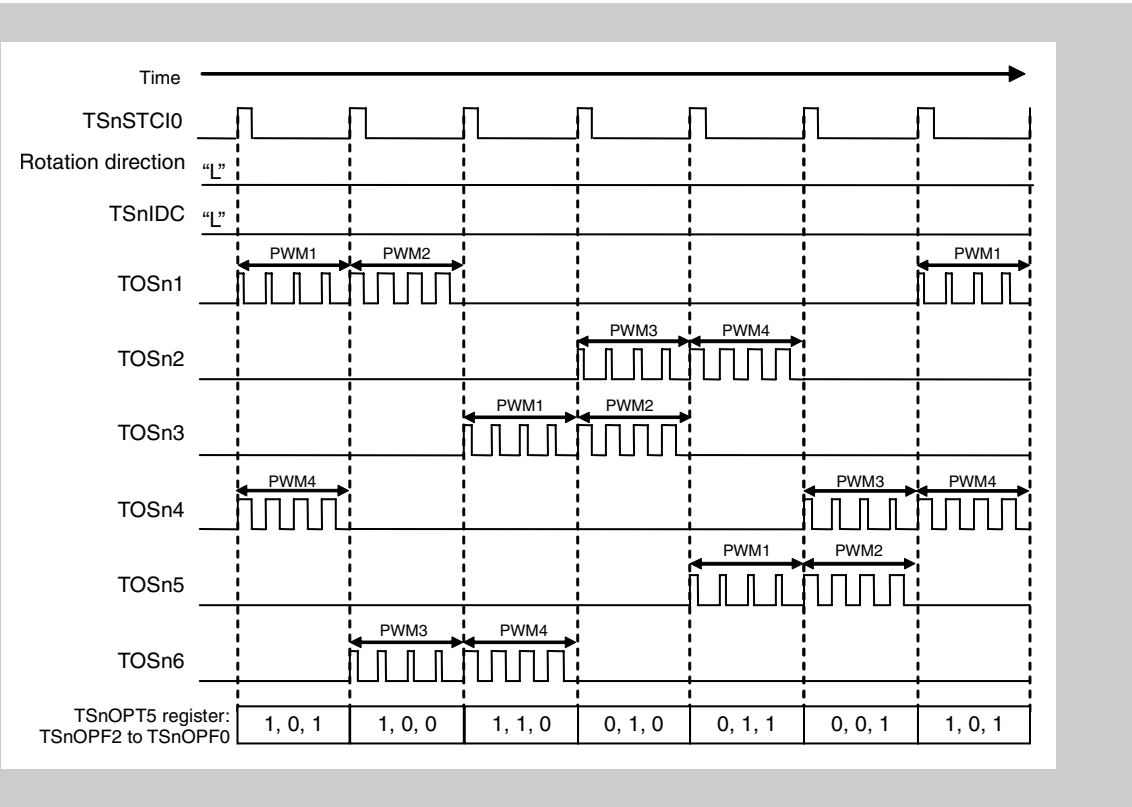


Figure 17-138 120° excitation mode operation example in use of trigger input switch method (normal rotation)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1, TSnOPT5 register: TSnPOT = 1, TSnPSS = 1, TSnOPT7 register: TSnIDC = 1

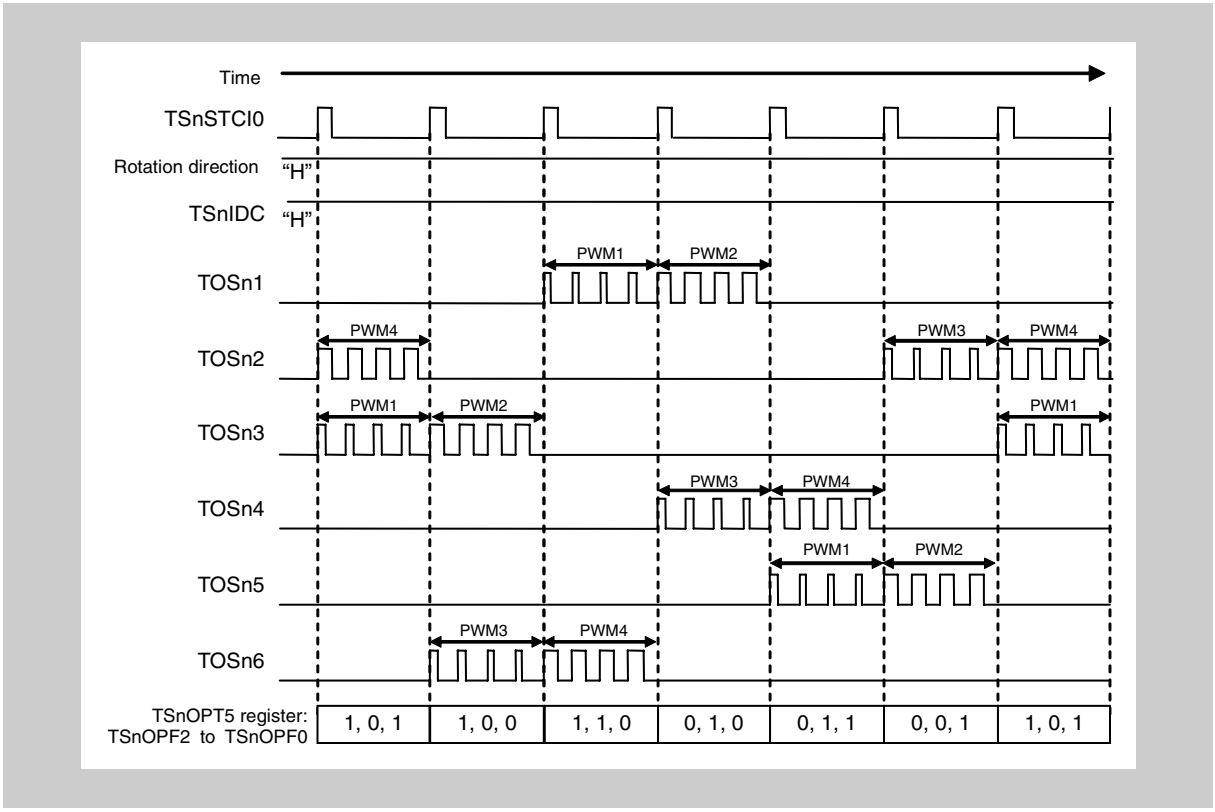


Figure 17-139 120° excitation mode operation example in use of trigger input switch method (reverse rotation)

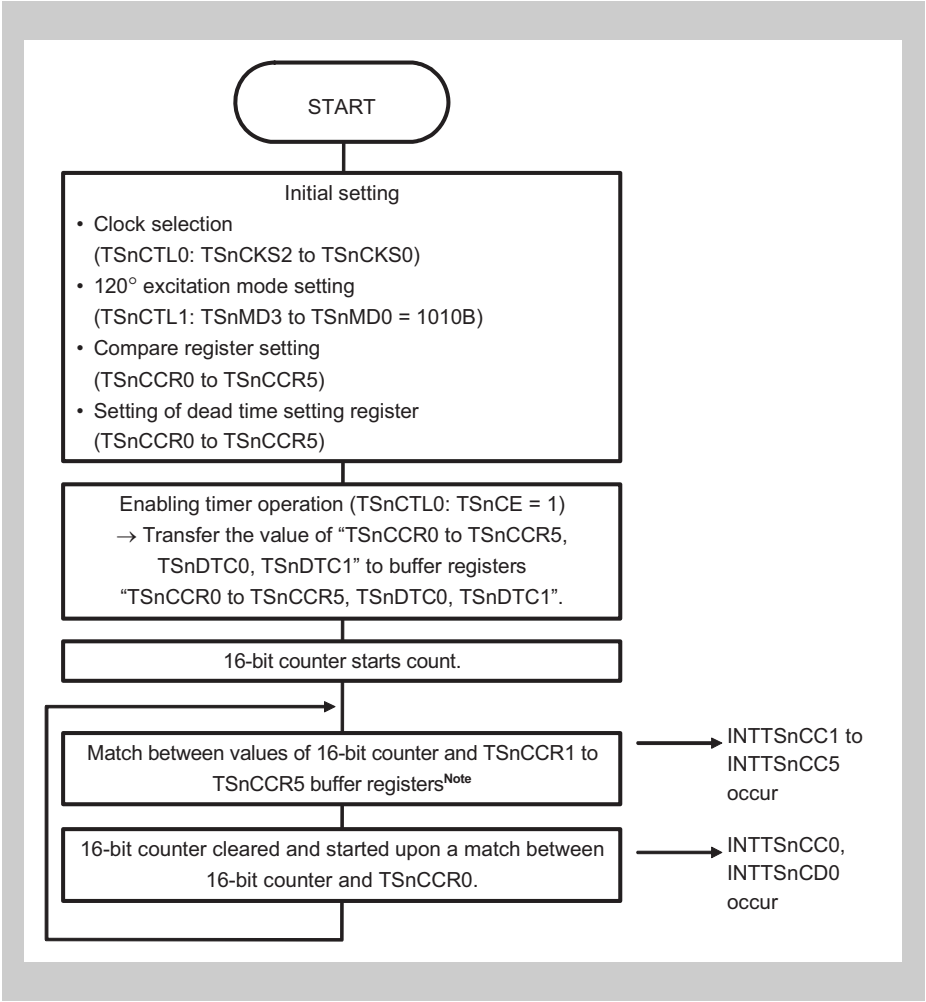


Figure 17-140 Basic operation flow in 120° excitation mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

(2) 120° excitation mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting During Operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	PWM duty
TSnCCR4	Reload	Possible	PWM duty (selectable as A/D conversion trigger)
TSnCCR5	Reload	Possible	Selectable as A/D conversion trigger
TSnDTC0, TSnDTC1	Reload	Conditionally possible[Note]	Dead time

Note Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* for details.

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-
TAPTSn2 to TAPTSn0	Pattern input
TSnSTCI0, TSnSTCI1 signals	Trigger input
TSnARD signal	Pattern change order input

(c) Output pin

Pin	Function
TOSn0	Toggle output by TSnCCR0 compare match
TOSn1	PWM output (with dead time) by TSnCCR1, TSnCCR2 compare match
TOSn2	PWM output (with dead time) by TSnCCR3, TSnCCR4 compare match
TOSn3	PWM output (with dead time) by TSnCCR1, TSnCCR2 compare match
TOSn4	PWM output (with dead time) by TSnCCR3, TSnCCR4 compare match
TOSn5	PWM output (with dead time) by TSnCCR1, TSnCCR2 compare match
TOSn6	PWM output (with dead time) by TSnCCR3, TSnCCR4 compare match
TOSn7	Pulse output by A/D conversion trigger

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of INTTSnCC0 to INTTSnCC5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing with INTTSnCC0).
INTTSnWN	Warning

(e) Compare match timing

Compare match	Timing
TSnCCR0	When 16-bit counter switches from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between values of 16-bit counter and TSnCCR1 to TSnCCR5

Note "–" indicates an unused function in the 120° excitation mode.

(3) Setting of 120° excitation mode

(a) Mode setting

The 120° excitation mode is set by setting TSnMD3 to TSnMD0 of the TSnCTL1 register to [1, 0, 1, 0].

(b) Output level/output enable setting

Output level/output enable is set by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register.
Toggle output is provided from the TOSn0 pin with cycle match (match between the 16-bit counter and TSnCCR0 register).
The TOSn7 pin is for A/D conversion output. Enable the output according to the need.

(c) Error interrupt/warning interrupt occurrence enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1 when the positive/negative phase simultaneous active state is detected.
A warning interrupt (INTTSnWN) is enabled by setting TSnWOC of the TSnIOC4 register to 1.

(d) Interrupt and thinning out function setting

A peak interrupt (INTTSnCD0) occurs upon a match between the TSnCCR0 register and 16-bit counter. (A trough interrupt does not occur).
Set TSnICE of the TSnOPT1 register to 1 to output a peak interrupt.
To use the thinning out function for peak interrupt, set the TSnID4 to TSnID0 bits of the TSnOPT1 register.

(e) Reload thinning out function setting

Set TSnRDE of the TSnOPT1 register to 1 to set reload timing to the same timing with the interrupt timing.

(f) A/D conversion trigger output setting

To set A/D conversion trigger 0 (TSnADTRG0 signal), set whether to enable/disable at the match timing with the TSnCCR5 register, at the match timing with the TSnCCR4 register, and at the peak interrupt (INTTSnCD0) by using the TSnAT04, TSnAT02, and TSnAT01 bits of the TSnOPT2 register.
To set A/D conversion trigger 1 (TSnADTRG1 signal), set whether to enable/disable at match timing with TSnCCR5 register, at match timing with the TSnCCR4 register, and at peak interrupt (INTTSnCD0) by using the TSnAT14, TSnAT12, and TSnAT11 bits of the TSnOPT3 register.
No thinning out, 1-thinning out, 3-thinning out, or 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

Caution

1. When using the TOSn7 pin, set the TSnOPT2 and TSnOPT3 registers and TSnCCR4 and TSnCCR5 registers appropriately according to the system.
2. Trough interrupts (INTTSnOD) do not occur in the 120° excitation mode. Be sure to set TSnAT00, 10 bits of the TSnOPT2 and TSnOPT3 registers to 0.
3. The 16-bit sub-counter does not operate in the 120° excitation mode. Be sure to set the TSnAT07, TSnAT06, TSnAT17, and TSnAT16 bits of the TSnOPT2, TSnOPT3 registers to 0.
4. The 16-bit counter does not enter the count-down status in the 120° excitation mode. Be sure to set the TSnAT05, TSnAT03, TSnAT15, and TSnAT13 bits of the TSnOPT2 and TSnOPT3 registers to 0.
5. Both PWM duty and A/D0 conversion trigger are set with the TSnCCR4 register in the 120° excitation mode. Note that both the PWM duty and A/D0 conversion trigger timing will be changed if the TSnCCR4 register setting value is changed.

(g) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.
Dead time can be calculated with the following equation.
16-bit counter operation clock cycle × TSnDTC0
16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of TOSn2, TOSn4, TOSn6 pins to the active state of TOSn1, TOSn3, TOSn5 pins can be set with the TSnDTC0 register.
The time for the change from the inactive state of TOSn1, TOSn3, TOSn5 pins to the active state of TOSn2, TOSn4, TOSn6 pins can be set with the TSnDTC1 register.

(h) Output PWM setting

In the 120° control, the output of the TOSn1, TOSn3, and TOSn5 pins is controlled with the TSnCCR1 and TSnCCR2 registers, and the output the TOSn2, TOSn4, and TOSn6 pins is controlled with the TSnCCR3 and TSnCCR4 registers. The duty ratio can be set with the TSnCCR1 to TSnCCR4 registers for the PWM cycle (TSnCCR0 register). 0000H must be set to the TSnCCR1 to TSnCCR4 registers to set the duty ratio of 0%, and “TSnCCR0 + 1” should be set to the TSnCCR1 to TSnCCR4 registers to set the duty ratio of 100%. These settings enable chopping output control and rectangular output control.

(i) Pattern output switch timing, pattern output order, and initial output pattern setting

[Pattern switch method]

The pattern switch method is enabled by setting TSnPOT of the TSnOPT5 register to 0.
Output of the TOSn1 to TOSn6 pins is switched when the TAPTSn2 to TAPTSn0 pins change.
Output order at operation start is set with the TSnIDC bit of the TSnOPT7 register. Initial output pattern is set with the TSnPSC bit of the TSnOPT4 register. However, setting of the TSnPSC bit is invalid after rotation direction is determined (after a value is set in the TSnTSF bit of the TSnOPT5 register).

[Trigger switch method]

Trigger switch method is enabled by setting TSnPOT of the TSnOPT5 register to 1. Output of the TOSn1 to TOSn6 pins is switched with the rise of the external input (TSnSTCI1 and TSnSTCI0 signals).
Output order is set with the combination of TSnARD signal, TSnPSC bit of the TSnOPT4 register, or the TSnIDC bit of the TSnOPT7 register. Selection of the TSnARD signal or the TSnPSC bit of the TSnOPT4 register is set with the TSnPSS bit of the TSnOPT4 register. The output order is controlled with the TSnARD input signal when TSnPSS is set to 0, and controlled with the TSnPSC bit when TSnPSS is set to 1. Refer to *"Timer output pattern in each mode and function" on page 739* for the pattern output order.
The initial output pattern can be controlled with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. The Initial pattern is output at the timer Sn operation start setting timing (when TSnCE of the TSnCTL0 register to 1) following setting with the TSnIPC2 to TSnIPC0 bit. Refer to *"Timer output initial pattern setting in each mode and function" on page 731* for details.

Caution Set the initial pattern by reading the input level of the port connected to the TAPTSn2 to TAPTSn0 pins.

(4) Operation in 120° excitation mode

Figure 17-141 on page 941 and *Figure 17-142 on page 942* show operation examples in the 120° excitation mode. TOSn1 to TOSn6 detect input level change timing of the TAPTSn2 to TAPTSn02 pins to change output pattern. The 16-bit counter operation is saw-tooth wave operation to provide PWM output by the SnCCR0 to TSnCCR4 registers. The 16-bit counter is not cleared to 0000H until the match with the TSnCCR0 register value in the PWM cycle even if change of the TAPTSn2 to TAPTSn0 pins is detected. The timer output pattern is switched from the cycle following the PWM cycle where the change of the TAPTSn2 to TAPTSn0 pins is detected.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0, TSnOPT5 register: TSnADC = 0, TSnPOT = 0, TSnOPT7 register: TSnIDC = 0

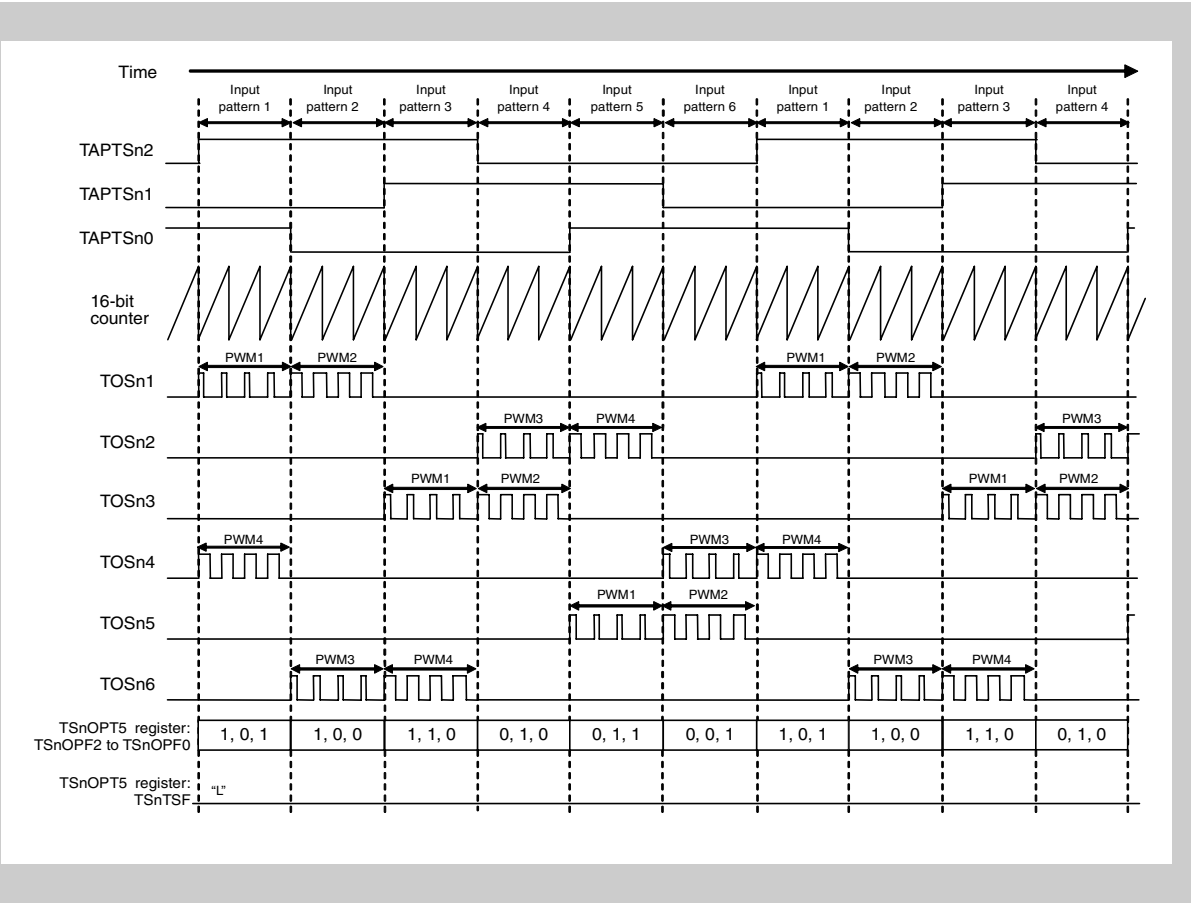


Figure 17-141 120° excitation mode basic operation timing example (pattern switch method, normal rotation)

Note PWM1 to PWM4 indicate the PWM operation set with the TSnCCR1 to TSnCCR4 registers.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1, TSnOPT5 register: TSnADC = 0, TSnPOT = 0, TSnOPT7 register: TSnIDC = 1

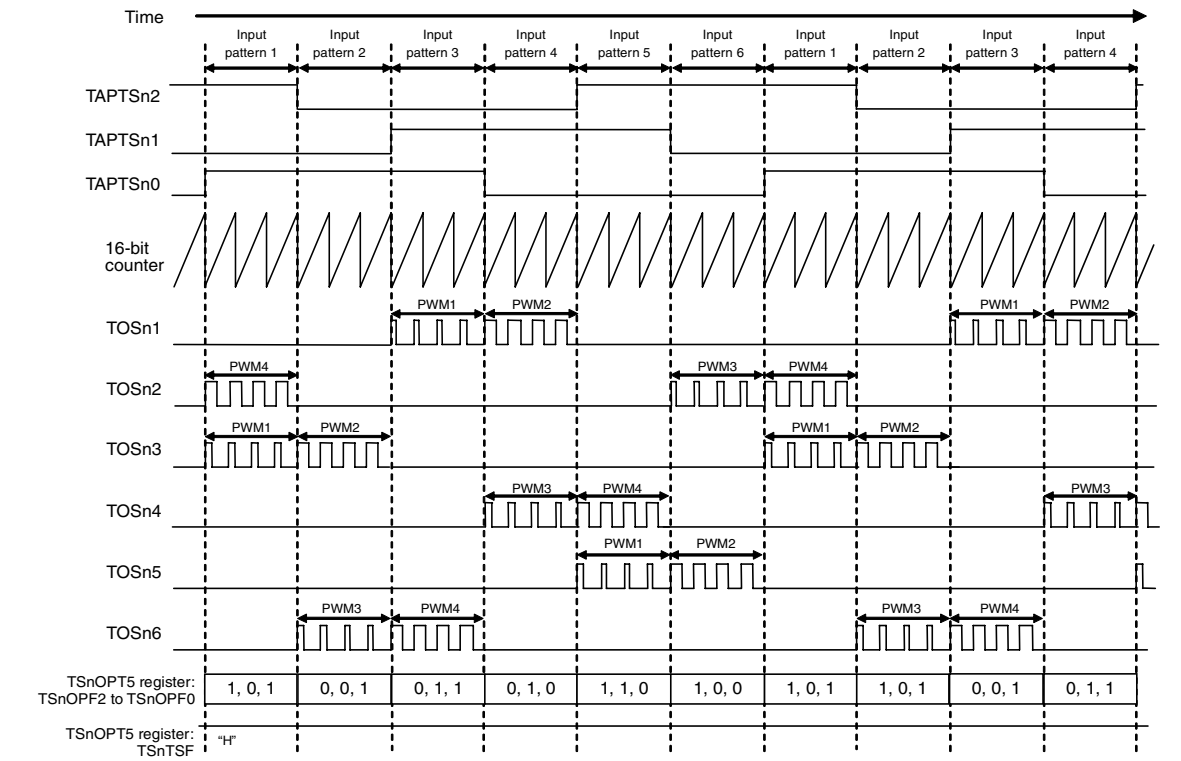


Figure 17-142 120° excitation mode basic operation timing example (pattern switch method, reverse rotation)

Note PWM1 to PWM4 indicate the PWM operation set with the TSnCCR1 to TSnCCR4 registers.

(5) Output pattern list in 120° excitation mode

In the 120° excitation mode, the output pattern is determined by the rotation direction and the TSnIDC bit of the TSnOPT7 register.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
0	-	TSnTSF flag
1	0	Input signal to TSnARD
1	1	TSnPSC bit

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,
and TSnOPT7 register: TSnIDC = 0

⟶ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0 of TSnOPT4 register[Note 1]/TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 2	Note 3
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 2	Note 3
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 2	Note 3
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 2	Note 3
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 2	Note 3

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,
and TSnOPT7 register: TSnIDC = 1

⟶ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0 of TSnOPT4 register[Note 1]/TSnOPF2 to TSnOPF0 of TSnOPT5 mode							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM1	PWM2	INACT	Note 2	Note 3
TOSn2	PWM3	PWM4	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn3	PWM2	INACT	INACT	INACT	INACT	PWM1	Note 2	Note 3
TOSn4	INACT	INACT	PWM3	PWM4	INACT	INACT	Note 2	Note 3
TOSn5	INACT	PWM1	PWM2	INACT	INACT	INACT	Note 2	Note 3
TOSn6	INACT	INACT	INACT	INACT	PWM3	PWM4	Note 2	Note 3

Note
1. The output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1. Then, output is switched according to the pattern switch order when the pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 do not change even if the output pattern is switched.
2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
3. Output remains the level before the change.
4. PWM1 to PWM4: PWM output of the TSnCCR1 to TSnCCR4 registers

5. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,
and TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output Pin	TSnIPC2 to TSnIPC0 of TSnOPT4 Register[Note 1]/TSnOPF2 to TSnOPF0 of TSnOPT5 Register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM2	PWM1	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn2	INACT	INACT	INACT	PWM4	PWM3	INACT	Note 2	Note 3
TOSn3	INACT	INACT	PWM2	PWM1	INACT	INACT	Note 2	Note 3
TOSn4	PWM3	INACT	INACT	INACT	INACT	PWM4	Note 2	Note 3
TOSn5	INACT	INACT	INACT	INACT	PWM2	PWM1	Note 2	Note 3
TOSn6	INACT	PWM4	PWM3	INACT	INACT	INACT	Note 2	Note 3

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,
and TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output Pin	TSnIPC2 to TSnIPC0 of TSnOPT4 Register[Note 1]/TSnOPF2 to TSnOPF0 of TSnOPT5 Register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM2	PWM1	INACT	Note 2	Note 3
TOSn2	PWM4	PWM3	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn3	PWM1	INACT	INACT	INACT	INACT	PWM2	Note 2	Note 3
TOSn4	INACT	INACT	PWM4	PWM3	INACT	INACT	Note 2	Note 3
TOSn5	INACT	PWM2	PWM1	INACT	INACT	INACT	Note 2	Note 3
TOSn6	INACT	INACT	INACT	INACT	PWM4	PWM3	Note 2	Note 3

Note

1. The output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1. Then, output is switched according to the pattern switch order when the pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 do not change even if the output pattern is switched.
2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
3. Output remains the level before the change.
4. PWM1 to PWM4: PWM output of the TSnCCR1 to TSnCCR4 registers
5. INACT: Inactive level output

(6) Operation start timing in 120° excitation mode

In the case of the trigger switch control in the 120° excitation mode, the pattern that is set with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, rotation direction, and the TSnIDC bit of the TSnOPT7 register can be output. In the case of the pattern switch control (TSnPOT = 0), however, the pattern of the TAPTSn2 to TAPTSn0 pins can be detected, but the rotation direction (TSnTSF bit of the TSnOPT5 register) cannot be determined. Therefore, the initial pattern is determined with TSnPSC of the TSnOPT4 register (TSnPSS = 1 of the TSnOPT5 register) or the TSnARD signal level (TSnPSS = 0 of the TSnOPT5 register) until the rotation direction is determined.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0 (TSnARD = 0),
TSnOPT5 register: TSnPOT = 0,
and TSnOPT7 register: TSnIDC = 0.

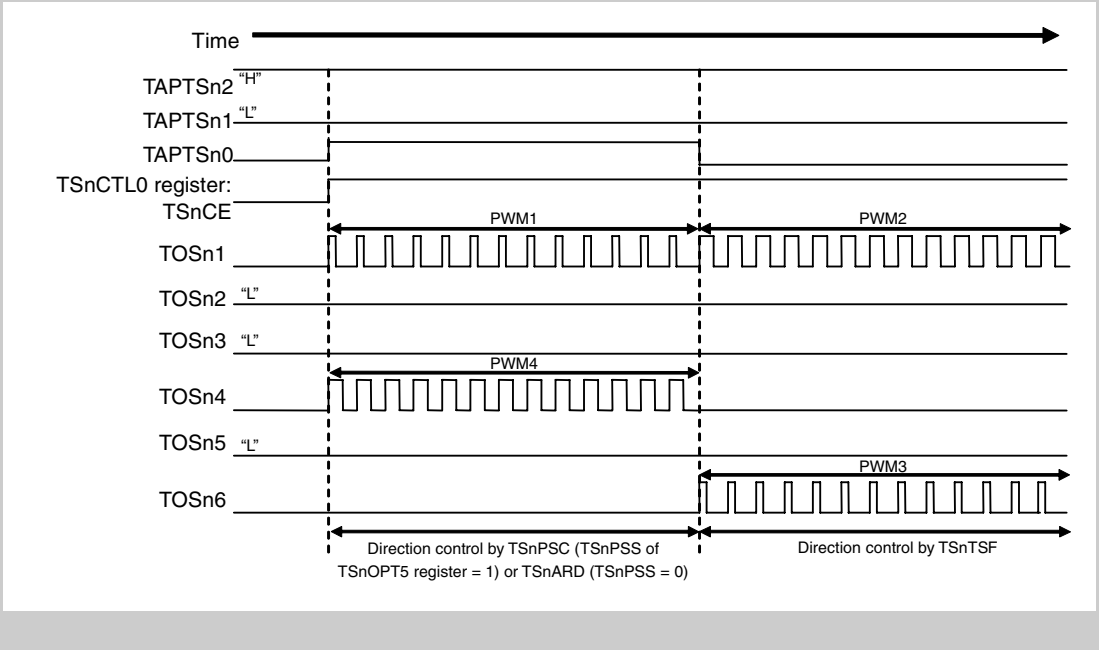


Figure 17-143 Control at timer output start during normal rotation (when correct pattern is input)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1 (TSnARD = 1),
TSnOPT5 register: TSnPOT = 0,
and TSnOPT7 register: TSnIDC = 0.

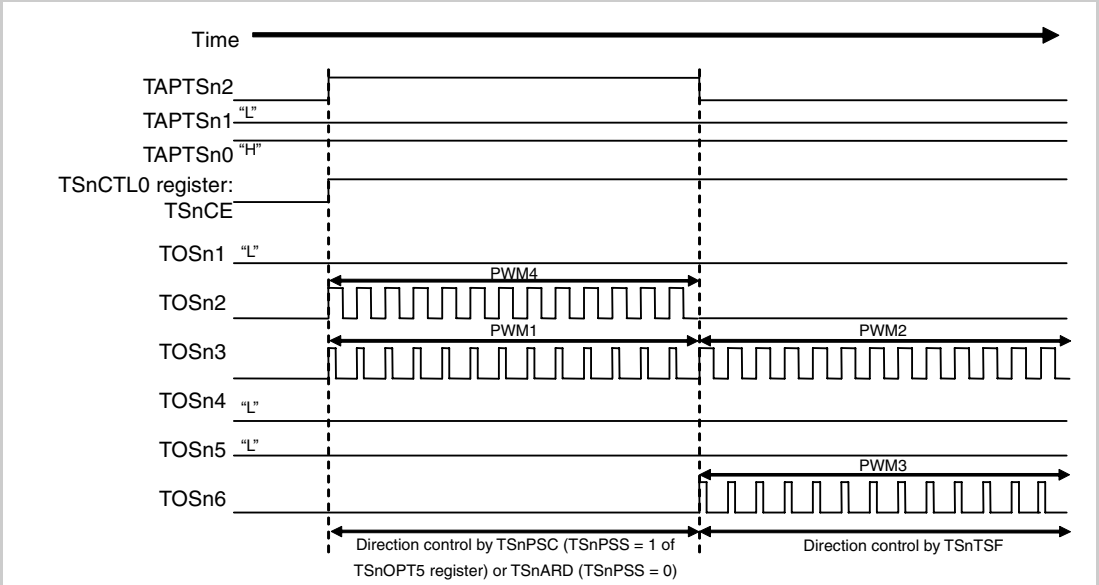


Figure 17-144 Control at timer output start during reverse rotation (when correct pattern is input)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0 (TSnARD = 0),
TSnOPT5 register: TSnPOT = 0,
and TSnOPT7 register: TSnIDC = 0.

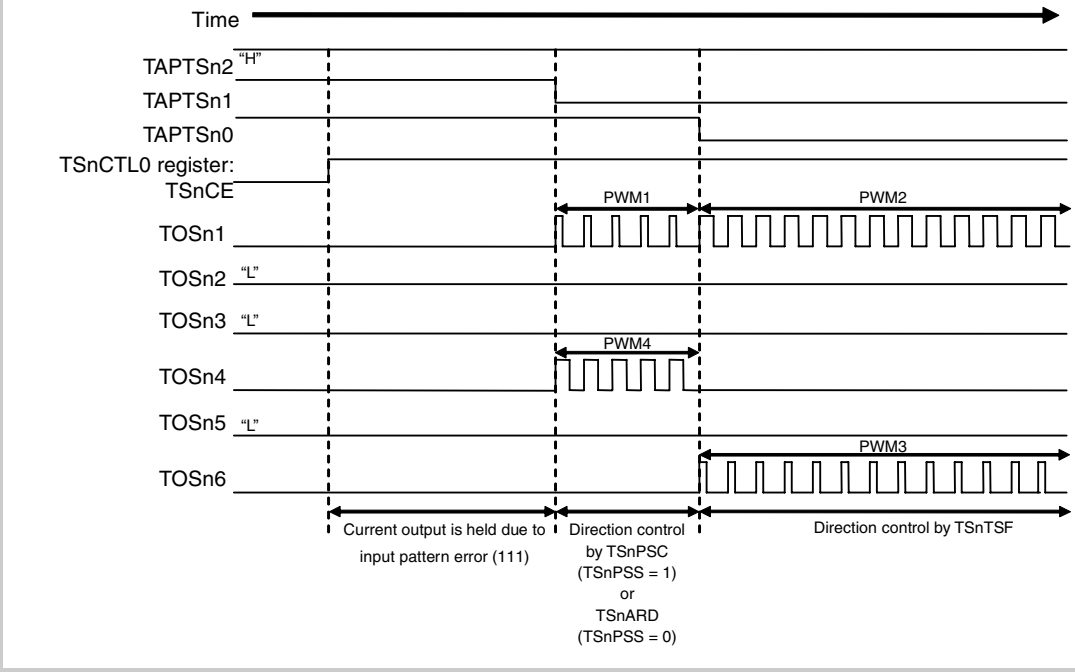


Figure 17-145 Control at timer output start during normal rotation (when error pattern is input)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1 (TSnARD = 1),
TSnOPT5 register: TSnPOT = 0,
and TSnOPT7 register: TSnIDC = 0.

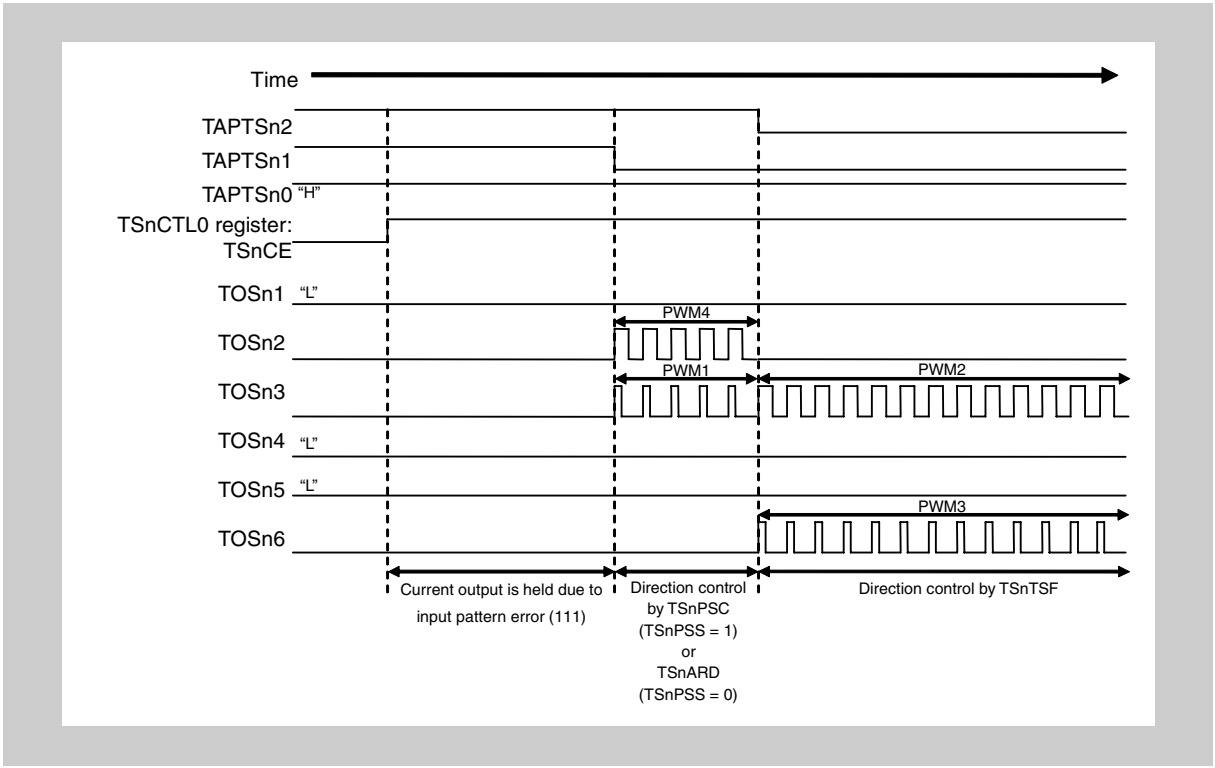


Figure 17-146 Control at timer output start during reverse rotation (when error pattern is input)

(7) Output switch timing in 120° excitation mode

In the 120° excitation mode, the external output pattern switch timing (TSnSTCI0 and TSnSTCI1 signals, TAPTSn2 to TAPTSn0 pins) is input regardless of the 16-bit counter operation. The output switch timing is from the cycle following the cycle of external input switch timing. Therefore, PWM output is held until the current cycle is completed even if the switch timing is input from the outside.
When the TAPTSn2 to TAPTSn0 pins are changed serially within the 1st cycle in the pattern switch method, the value edge-detected just before cycle match is used and reflected to PWM output from the following cycle. If the TSnSTCI0 and TSnSTCI1 signal triggers are input several times within a cycle in the trigger switch method, the pattern is switched to the adjacent output pattern due to only once acceptance.
Rewriting of TSnIPC2 to TSnIPC0 is prior to the TSnSTCI0 and TSnSTCI1 triggers in case of both occurrences within a cycle in the trigger switch method. The value rewritten just before cycle match is reflected if TSnIPC2 to TSnIPC0 are rewritten several times in a cycle.

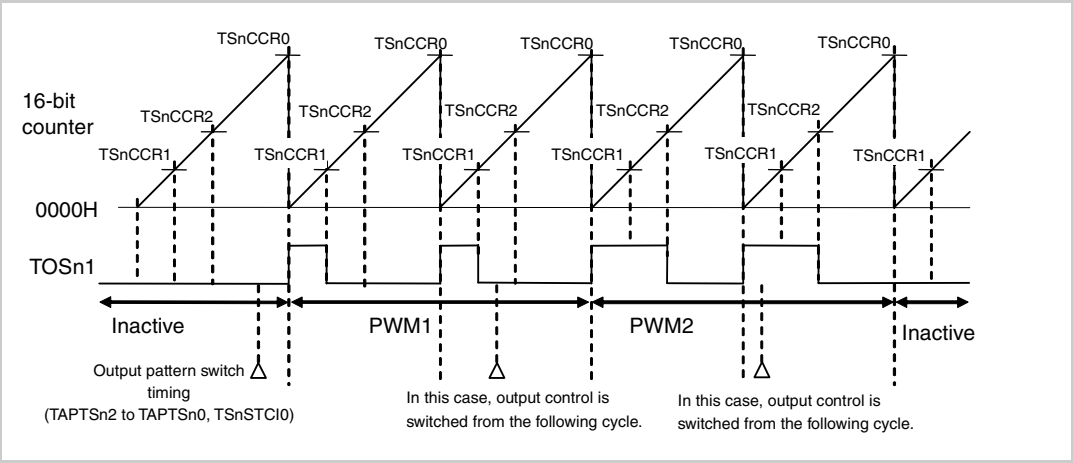


Figure 17-147 Output switch example (TAPTSn2 to TAPTSn0 Pins, TSnSTCI0, TSnSTCI1 signal trigger input)

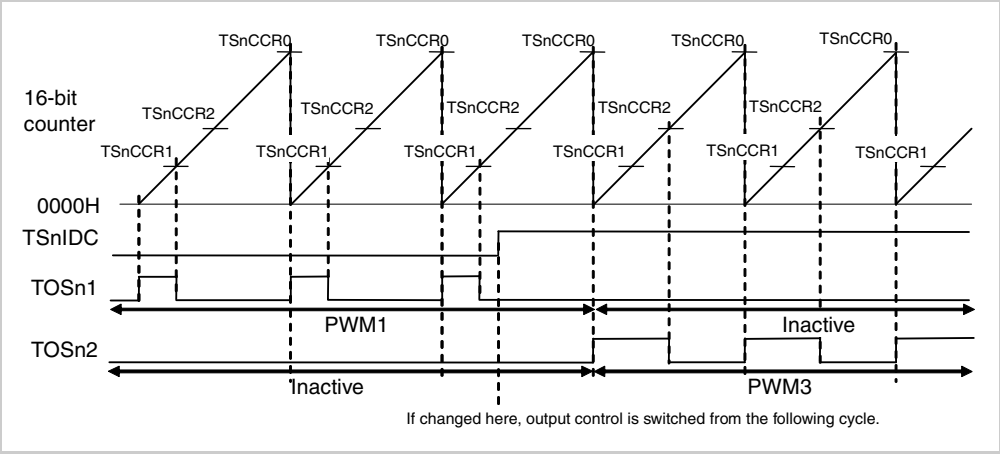


Figure 17-148 Output switch example (switch with TSnIDC Bit of TSnOPT7 register)

(TSnOPT5 register: TSnPOT bit = 1)

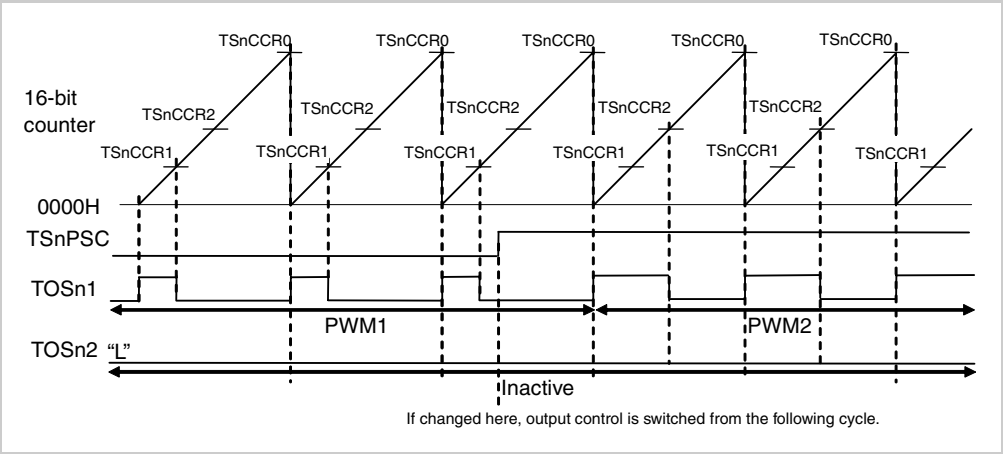


Figure 17-149 Output switch example (switch with TSnPSC Bit of TSnOPT4 register)

TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 1

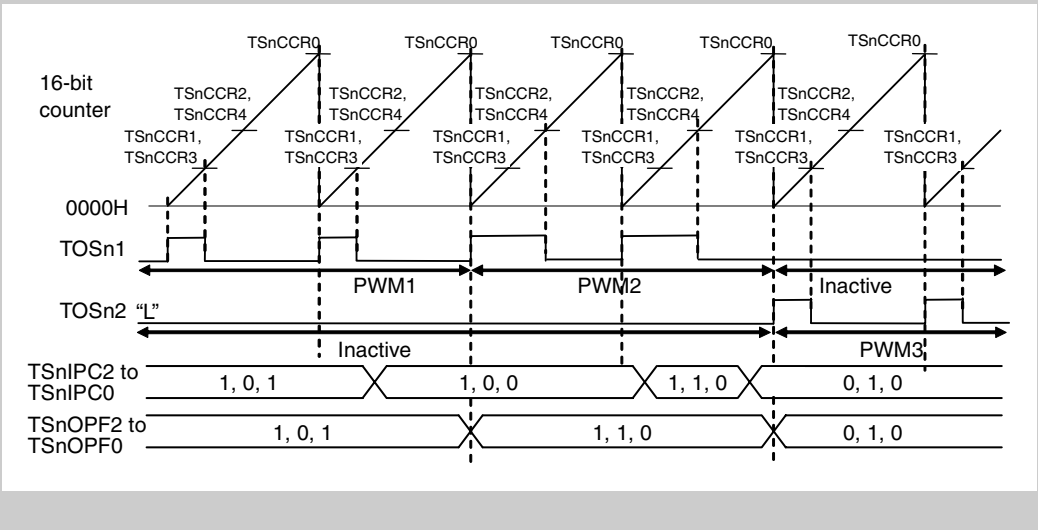


Figure 17-150 Output switch example (switch with TSnIPC2 to TSnIPC0 bits of TSnOPT4 register)

(8) Compare register rewrite timing in 120° excitation mode

The operation example when the TSnCCR1 register is reloaded (rewritten) is described below.
Figure 17-151 on page 951 shows an output example in case of rewriting the TSnCCR1 register. After change of the TSnCCR1 register, data is not transferred to the TSnCCR1 buffer register until the next reload timing (the changed data will not be valid), so that the output waveform can be obtained as set. However, rewriting the TSnCCR1 register again is prohibited during the reload is kept pending (in the period from the change of the TSnCCR1 register to the reload overwriting). Be sure to read the reload request flag (TSnRSF) to confirm 0 status before writing to the TSnCCR1 register.

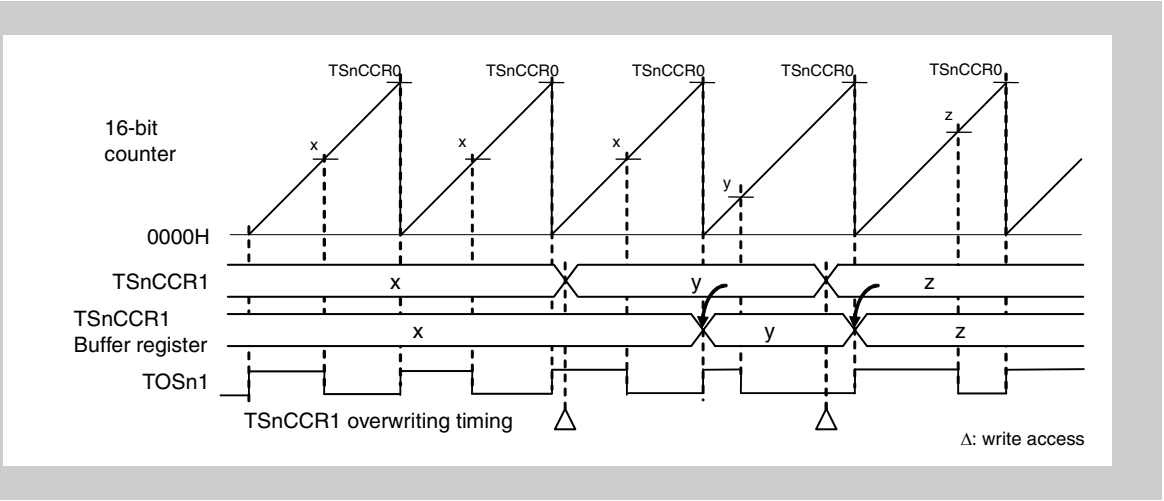


Figure 17-151 Output example in case of rewriting of TSnCCR1 register

(9) Dead time control in 120° excitation mode

In the 120° excitation mode, dead time is added by the dead time control operation at the fall of each phase.
The dead time that is set to the TSnDTC1 register is inserted at the fall of the positive phase, and the dead time that is set to the TSnDTC0 register is inserted at the fall of the negative phase.

(TSnOPT5 register: TSnPOT bit = 1)

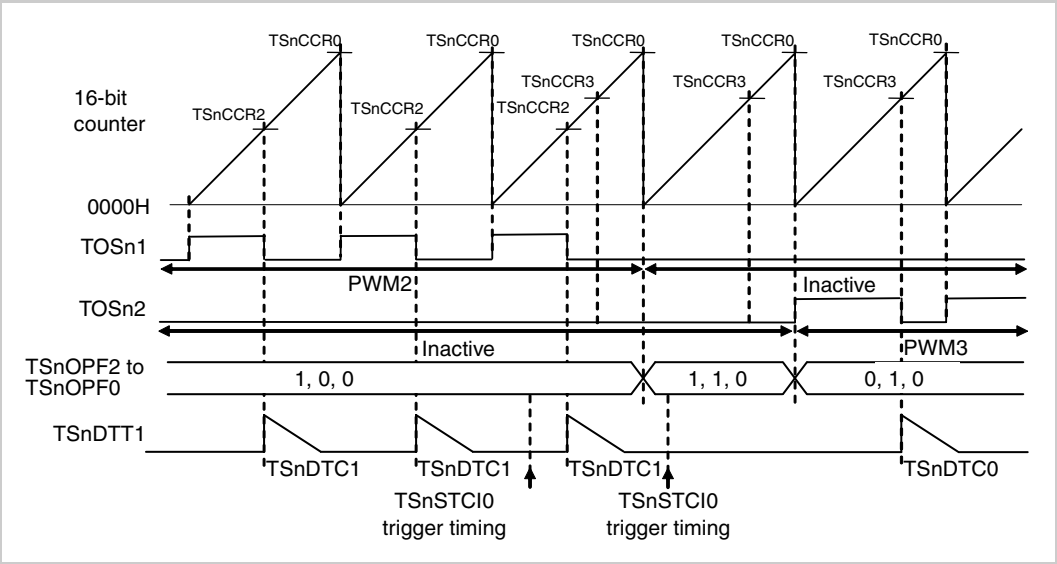


Figure 17-152 Output switch example (switch with TSnIPC2 to TSnIPC0 bits of TSnOPT4 register)

Caution Dead time control has little effect on the timer output during normal operation. However, dead time control may affect the timer output under the following conditions.

- When noise occurs in input pattern during use of the pattern switch method
- When input pattern changes earlier than PWM cycle during use of the pattern switch method
- When the output pattern is forcibly changed by changing the TSnIPC2 to TSnIPC0 bit of the TSnOPT4 register in the trigger switch method
- When the switch method is changed
- When the current direction control bit (TSnIDC bit of TSnOPT7 register) is changed
- When software output control function is used
- When 180° excitation control is used

(10) Output switch in 120° excitation mode

In the 120° excitation mode, the output pattern can be controlled by writing the value to the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register when the trigger switch method is used (TSnPOT of TSnOPT5 register is set to 1) and also the TSnADC bit of the TSnOPT5 register is set to 0 to control the output patterns. Dead time is secured by using hardware at the switch timing.

Caution

1. The output level of the TOSn1 to TOSn6 pins before writing is kept when [1, 1, 1] or [0, 0, 0] is written to the TSnIPC2 to TSnIPC0 bits.
2. Do not rewrite more than once to the TSnIPC2 to TSnIPC0 bits in a PWM cycle. If rewritten twice or more, the value just before PWM cycle match occurrence is reflected.

(11) Operation when noise is generated in TAPTSn2 to TAPTSn0 pins in 120° excitation mode

Brushless DC motor hall sensor input is assumed for the TAPTSn2 to TAPTSn0 pins.
Noise may occur in the TAPTSn2 to TAPTSn0 pins in some systems. The operation in case of noise occurrence is described here.
Be sure to insert the noise filter circuit between the hall sensor and the TAPTSn2 to TAPTSn0 pins when designing system products.
Figure 17-154 on page 954 illustrates the case when noise is generated in the TAPTSn2 to TAPTSn0 pins during operation in the pattern switch method.

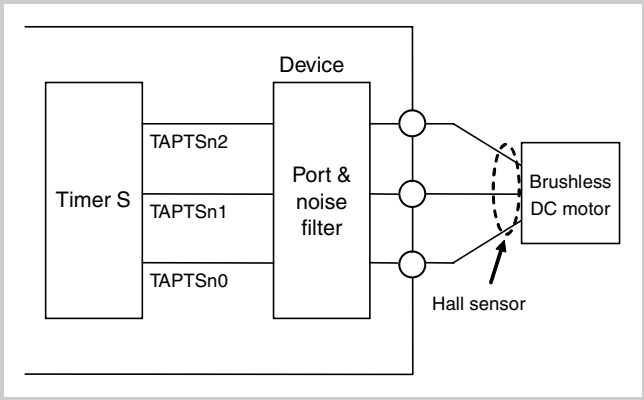


Figure 17-153 Noise filter circuit connection example

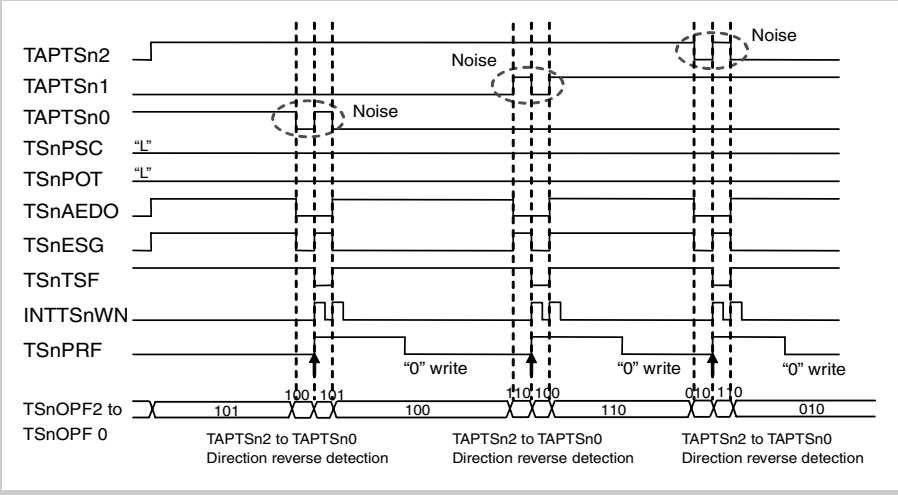


Figure 17-154 Noise occurrence example upon level change of TAPTSn2 to TAPTSn0 pins (pattern switch method)

(a) Change timing of input pattern change detection signal (TSnAEDO)

- TSnAEDO toggles at the input pattern (TAPTSn2 to TAPTSn0) change timing

Caution Be sure to specify the rotation direction with the TSnARD signal (TSnPSS of TSnOPT5 register set to 0) or the TSnPSC bit of the TSnOPT4 register (TSnPSS of TSnOPT5 register is set to 1).

(When TSnPSC = 0, or TSnARD = 0)

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	Toggle	-	-	-	-
	[1, 0, 0]	-	-	-	-	Toggle	-	-	-
	[1, 1, 0]	-	-	-	-	-	Toggle	-	-
	[0, 1, 0]	-	-	-	-	-	-	Toggle	-
	[0, 1, 1]	-	-	-	-	-	-	-	Toggle
	[0, 0, 1]	-	-	Toggle	-	-	-	-	-

(When TSnPSC = 1, or TSnARD = 1)

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	-	-	-	-	Toggle
	[1, 0, 0]	-	-	Toggle	-	-	-	-	-
	[1, 1, 0]	-	-	-	Toggle	-	-	-	-
	[0, 1, 0]	-	-	-	-	Toggle	-	-	-
	[0, 1, 1]	-	-	-	-	-	Toggle	-	-
	[0, 0, 1]	-	-	-	-	-	-	Toggle	-

(b) 3-phase encode signal (TSnESG) change timing

- TSnESG toggles at the input pattern (TAPTSn2 to TAPTSn0) change timing

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	Toggle	-	-	-	Toggle
	[1, 0, 0]	-	-	Toggle	-	Toggle	-	-	-
	[1, 1, 0]	-	-	-	Toggle	-	Toggle	-	-
	[0, 1, 0]	-	-	-	-	Toggle	-	Toggle	-
	[0, 1, 1]	-	-	-	-	-	Toggle	-	Toggle
	[0, 0, 1]	-	-	Toggle	-	-	-	Toggle	-

(c) Change timing of TOSn1 to TOSn6 pins

- In the case of the pattern switch method, the output pattern is changed when the input signal of the TAPTSn2 to TAPTSn0 pins[Note] is changed. The output switches in case of the simultaneous change of two or more pins.
- Output pattern changes at the TSnSTCI0 and TSnSTCI1 rising edges in the trigger switch method. Output also changes when writing to the TSnIPC2 to TSnIPC0[Note] bits of the TSnOPT4 register.

Note The output level of the TOSn0 to TOSn6 pins remains the level before change when changed to [0, 0, 0] or [1, 1, 1].

(d) TSnTSF flag change timing

- TSnTSF toggles at the input pattern (TAPTSn2 to TAPTSn0) change timing.

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	0	-	-	-	1
	[1, 0, 0]	-	-	1	-	0	-	-	-
	[1, 1, 0]	-	-	-	1	-	0	-	-
	[0, 1, 0]	-	-	-	-	1	-	0	-
	[0, 1, 1]	-	-	-	-	-	1	-	0
	[0, 0, 1]	-	-	0	-	-	-	1	-

(e) TSnNDF flag setting timing

- The setting timing is when two or more pins of TAPTSn2 to TAPTSn0 change simultaneously. The setting is cleared when writing 0.

(f) TSnPRF flag setting timing

- The setting timing is when TSnTSF changes. The setting is cleared when writing 0.

(g) TSnPEF flag setting timing

- The setting timing is when the value [0, 0, 0] or [1, 1, 1] is input to the TAPTSn2 to TAPTSn0 pins. The setting is cleared when writing 0.

(12) Basic control flow in 120° excitation mode

The 120° excitation mode includes 8 control statuses shown in *Table 17-22 on page 958*.
120° excitation control with the pattern switch method is enabled when setting TSnPOT of the TSnOPT5 register to 0.
This is defined as the phase fix control. In the phase fix control, delay from hall sensor or delay from sensor level detection to timer output needs to be considered. However, acceleration/deceleration is possible only by PWM duty change.
The 120° excitation control with the trigger switch method is used when TSnPOT = 1.
This is defined as phase variable control. In phase variable control, timer output pattern is set prior to hall sensor and acceleration/deceleration can be controlled by using the phase difference. However, phase variable control is more complex to control than phase fix control because the offset width for the hall sensor and the predicted value for the hall sensor must be considered. Refer to *"Timer Interconnection" on page 1105* for details.

The motor rotation direction can be set with the TSnPSC bit of the TSnOPT4 register when TSnPOT = 1, TSnPSS = 1 of TSnOPT5 register. Set TSnPSC = 0 in normal rotation, and TSnPSC = 1 in reverse rotation.
Control direction (acceleration/deceleration) is set with the TSnIDC bit of the TSnOPT7 register. Acceleration control is performed when the setting value is the same as the motor rotation direction (TSnPSC setting value), and deceleration control is performed when the setting value is different from the motor rotation direction.

Table 17-22 Timer Control Status

Status	TSnOPT4 register: TSnPSC	TSnOPT5 Register: TSnTSF	TSnOPT7 register: TSnIDC	TSnOPT5 register: TSnPOT	Control
A	-	0	0	0	Normal rotation/acceleration/phase fix
B	0	-	0	1	Normal rotation/acceleration/phase variable
C	0	-	1	1	Normal rotation/deceleration/phase variable
D	-	0	1	0	Normal rotation/deceleration/phase fix
E	-	1	1	0	Reverse rotation/acceleration/phase fix
F	1	-	1	1	Reverse rotation/acceleration/phase variable
G	1	-	0	1	Reverse rotation/deceleration/phase variable
H	-	1	0	0	Reverse rotation/deceleration/phase fix

Normally, control starts under the start status in which motor rotation stops. First, rotate the motor by using phase fix control to rotate from the stop status. Second, switch to the phase variable control to accelerate up to the high speed rotation. In the phase variable control, change the timer output earlier than the hall sensor change point in combination with timer T.
When decelerating from high speed rotation, change the control to the deceleration control by rewriting only the TSnIDC bit of the TSnOPT7 register. After decelerating revolution to the low-speed rotation, make the PWM duty smaller up to the stop status. This is the overall motor control flow.
Figure 17-155 and *Figure 17-156* show the status changes.

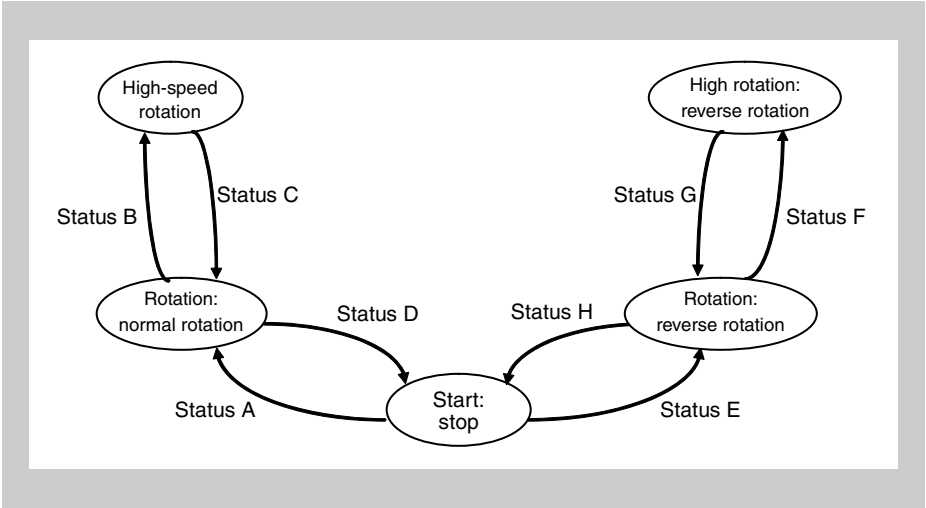


Figure 17-155 Status transition diagram

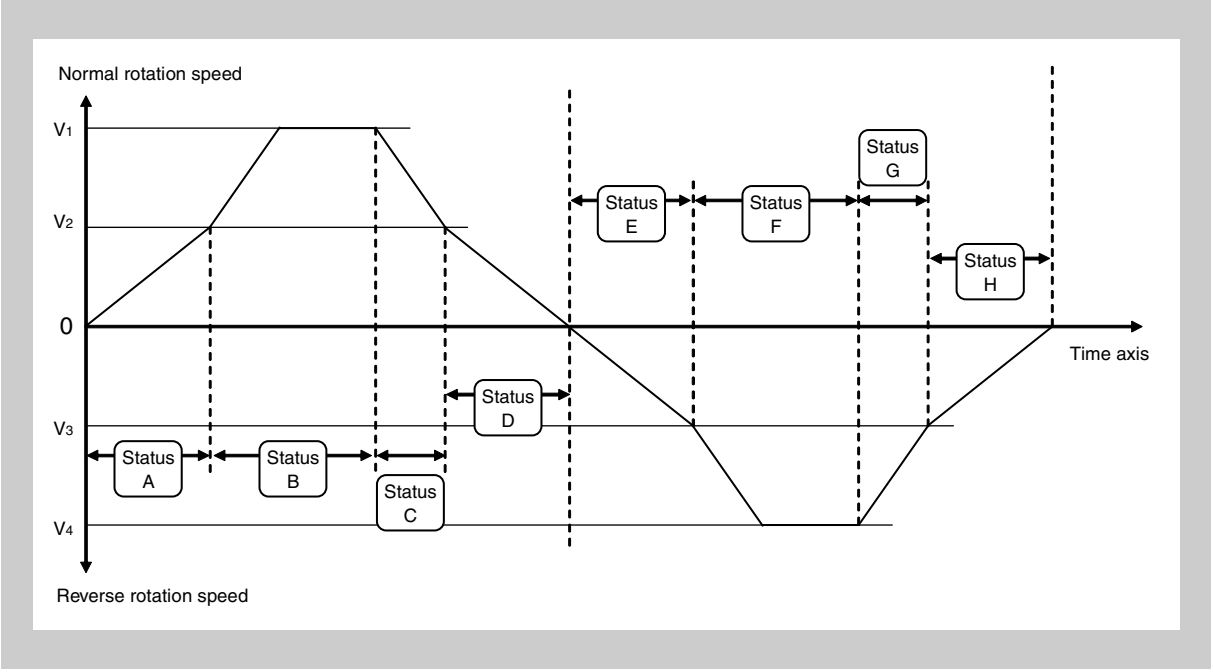


Figure 17-156 Relationship between status transition and motor rotation speed

Note V_1, V_4: Normal rotation/reverse rotation high-speed rotational speed
V_2, V_3: Normal rotation/reverse rotation low-speed rotational speed

(13) Software output control function in 120° excitation mode

In the 120° excitation mode, timer output is controlled with software control by using the TSnSOC and TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.
Output is switched immediately when setting TSnSOC = 1 as shown in *Figure 17-157*. The dead time period is assured by setting the dead time. Then, set TSnSOC to 0 in order to switch from the software output control to 120° excitation control. Output is held at this timing and is switched to be controlled by using the 120° excitation mode when reloading.
Refer to *"Software output function" on page 1014* for details on the software output control function.

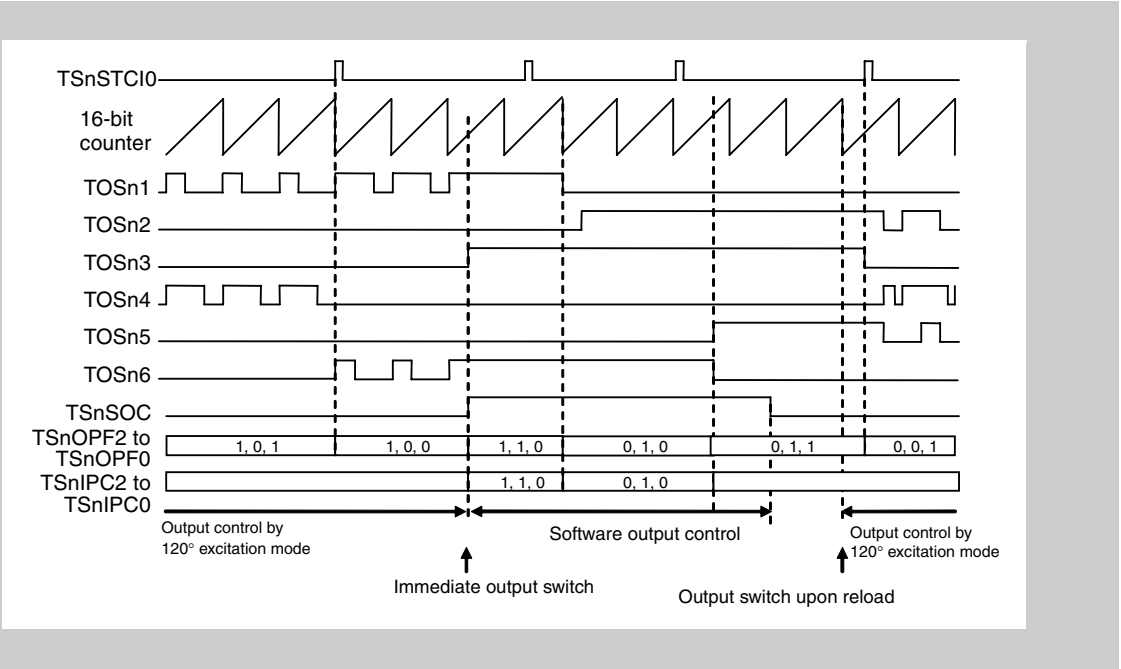


Figure 17-157 Example of software output control function switched from 120° excitation mode

(a) Software output control procedure

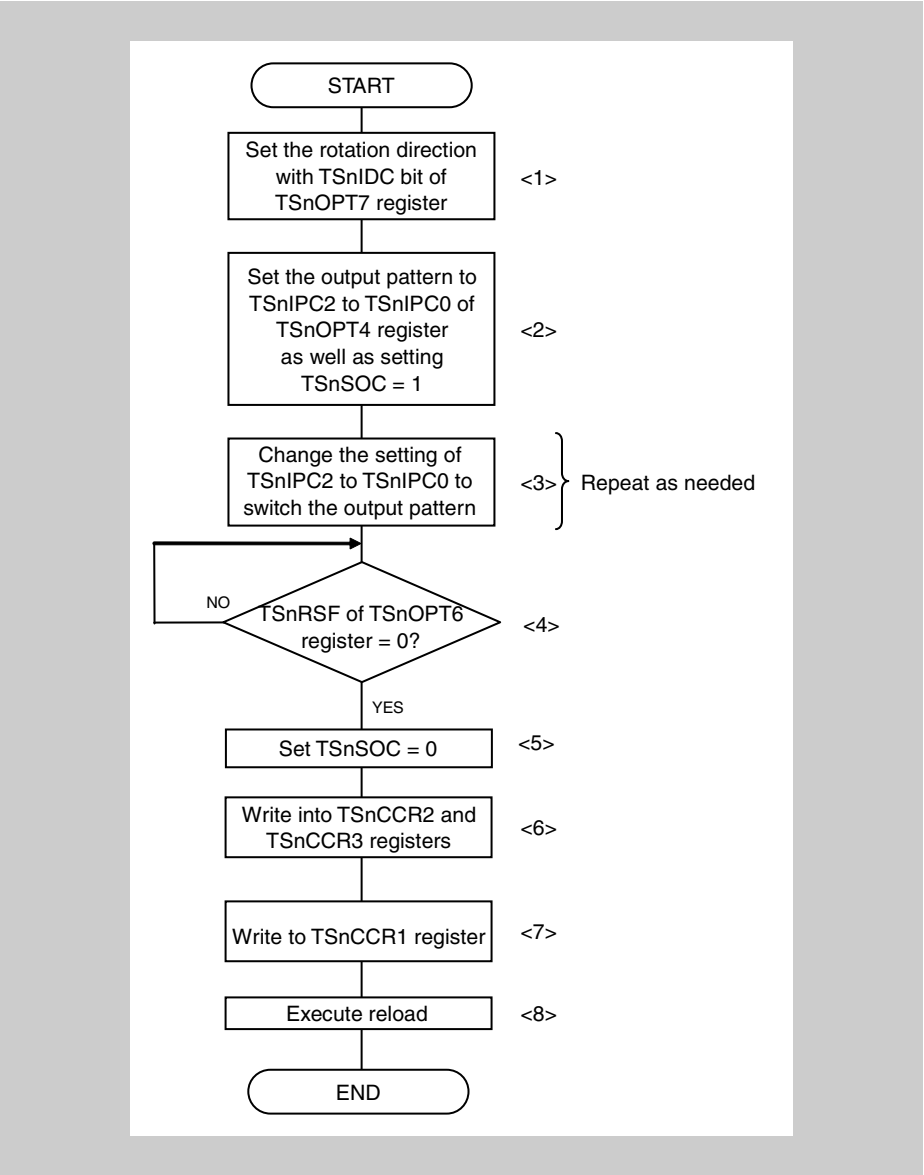


Figure 17-158 Processing flow in software output control

The software output control processing procedure is described below.

<1> Set TSnIDC to determine the rotation direction. There is a 180° phase difference in the timer output between when TSnIDC = 0 and TSnIDC = 1. In the software output control function, timer output is not changed only by rewriting the bit. However, 120° excitation control output pattern is changed if a cycle match occurs before <2>, so prevent cycle match before <2> in scheduling.

<2> Set the pattern to be output to TSnIPC2 to TSnIPC0 and also set TSnSOC to 1 in order to provide software output.

<3> Change the output pattern setting of TSnIPC2 to TSnIPC0 to change the timer output.
The registers that can be changed during software control are as follows.
TSnCTL0 register: TSnCE bit, TSnOPT1 to TSnOPT4 registers,
TSnOPT7 register: TSnIDC, TSnTOS bits, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register

<4> Check that the reload request flag (TSnRSF) is set to 0. In case of TSnRSF = 1, do not proceed to the next step before setting TSnRSF = 0.

<5> Software control release is started by setting TSnSOC = 0 (Not yet released in this step).

<6> Set the compare register which is necessary after the release of software output control. Proceed to the next step when there is no need to change. Change the register with the reload function in this step if necessary.

<7> Write to the TSnCCR1 register to perform reloading.

<8> Reload is executed and software output is released.

Caution

1. Any change of TSnIPC2 to TSnIPC0 is prohibited after setting TSnSOC to 0.
2. Be sure to perform reloading after the execution of steps <4>, <5>, <6> and <7>. Software output cannot be released if reload cannot be performed.

(14) 180° excitation control function in 120° excitation mode

In the 120° excitation mode, the 180° excitation control function can be used with TSnADC and TSnPOT of the TSnOPT5 register, the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register. Output is switched immediately when setting TSnADC = 1 as shown in *Figure 17-159 on page 963*. The dead time period is assured by setting the dead time. Then, trigger input to the output pattern switch trigger (TSnSTCI1 and TSnSTCI0 signals or TAPTSn2 to TAPTSn0 pins) leads to the output change according to the TSnPSC bit setting of the TSnOPT4 register. Output is held when clearing TSnADC to 0. Then the function is switched to the 120° excitation mode output control upon reload.
Refer to *"180° excitation control function" on page 1016* for details on 180° excitation control.

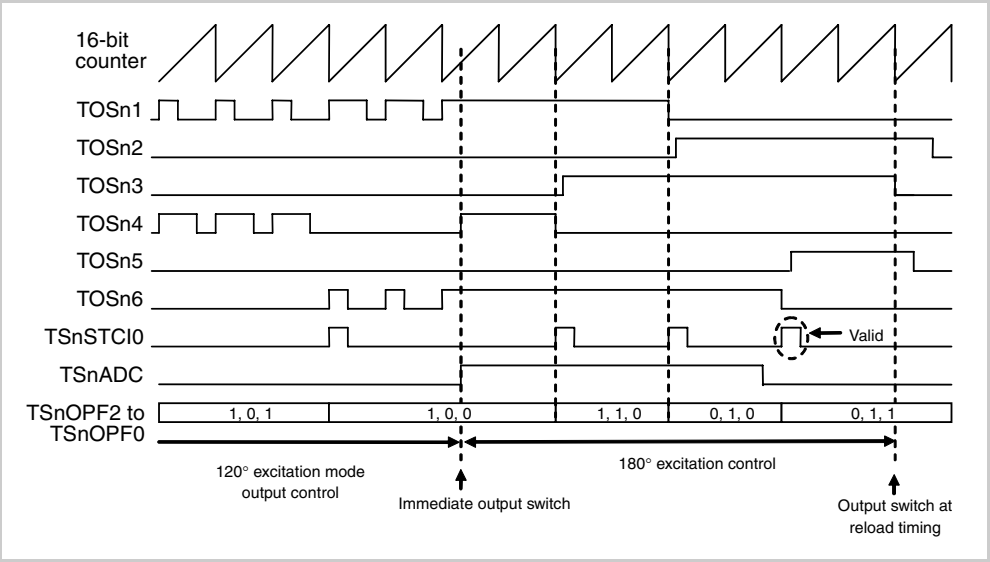


Figure 17-159 180° excitation control function switch example in 120° excitation mode

(a) 180° excitation control processing procedure

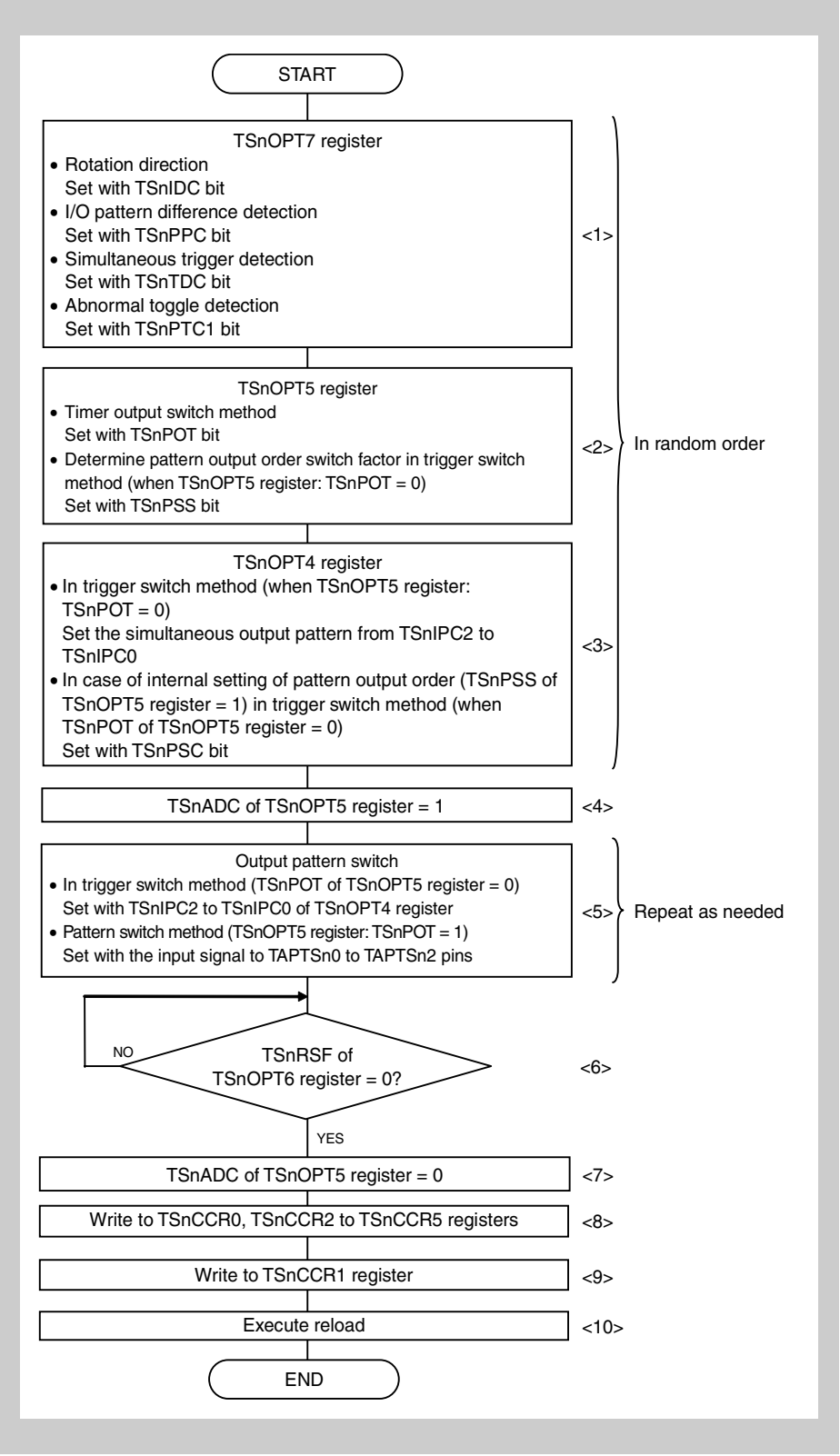


Figure 17-160 180° excitation control function processing flow

The following is the procedure for transition to 180° excitation control.

<1> TSnOPT7 register setting
Set a value to the TSnIDC bit to select the rotation direction.
Set a value to the TSnPPC bit to select to enable (1) or disable (0) the pattern phase difference detection flag (TSnPPF) of input (TAPTSn2 to TAPTSn0 pins) or output (TSnOPF2 to TSnOPF0 flags) with the trigger switch method (TSnPOT of TSnOPT5 register = 1).
Set a value to the TSnTDC bit to select either enable (1) or disable (0) the TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) with the trigger switch method.
Set a value to the TSnPTC1, TSnPTC0 bits to select either enable (1) or disable (0) for the TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF).

<2> TSnOPT5 register setting
Set a value to the TSnPOT bit to select the pattern output trigger. The pattern switch method where output is switched with the input pattern of TAPTSn2 to TAPTSn0 pins can be enabled by setting TSnPOT = 0, and the trigger switch method where output is switched with rising edge of TSnSTCI0, TSnSTCI1 signals can be enabled by setting TSnPOT = 1.
Set a value to the TSnPSS bit to select the pattern output order switch factor.

<3> TSnOPT4 register setting
Set a value to the TSnPSC bit to select the pattern output order for TSnPSS = 1 of the TSnOPT5 register in the trigger switch method.
Set a value to TSnIPC2 to TSnIPC0 to select the initial pattern for trigger switch method.

<4> Start the 180° excitation control function by setting TSnADC = 1.

<5> Output pattern switch

– Pattern switch method
In the pattern switch method, output is switched with the input pattern of the TAPTSn2 to TAPTSn0 pins. However, the previous output level is held if [1, 1, 1] or [0, 0, 0] is input to the TAPTSn2 to TAPTSn0 pins.

– Trigger switch method
In the trigger switch method, output is switched with the rising edge of the TSnSTCI0 and TSnSTCI1 signals.
Set the appropriate value to TSnIPC2 to TSnIPC0 if forcible change is needed for timer output. Writing to TSnIPC2 to TSnIPC0 is prior to the rising edge of the TSnSTCI0, TSnSTCI1 signals in case of these simultaneous occurrences.

– Switch method can be changed during operation.
The registers that can be rewritten during the 180° excitation control function operation are as follows.
TSnCTL0 register: TSnCE bit, TSnOPT1 to TSnOPT5 registers, TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register

Caution Setting of TSnSOC of the TSnOPT4 register = 1 and TSnADC of TSnOPT5 register = 1 is prohibited.

Note <1> to <3> can be performed in random order.

<6> Check that the reload request flag (TSnRSF) is set to 0. If TSnRSF = 1, do not proceed to the next step before setting TSnRSF = 0.

<7> Start the 180° excitation control function by setting TSnADC = 0. (Not yet released in this step).
After the change of TSnADC to 0, the output is switched upon the switching using the pattern/trigger switch method until the reload is performed. The output can be changed by writing to TSnIPC2 to TSnIPC0.

<8> Set the compare register that is necessary after the release of 180° excitation control. Proceed to the next step if there is no need for change. Change the register with the reload function in this step if necessary.

<9> Write to the TSnCCR1 register to perform reloading.

<10>Reload is executed and software output is released.

Caution Be sure to execute reload after execution of steps <6>, <7>, <8> and <9>. Software output cannot be released if reload cannot be executed.

(b) 180° excitation control switch timing from 120° excitation mode

Output is switched immediately when switching from the 120° excitation mode to 180° excitation control. Dead time control by hardware is inserted if the output is reverse to the current output level. The output level is held until the reload timing when switching from 180° excitation control to the 120° excitation mode.

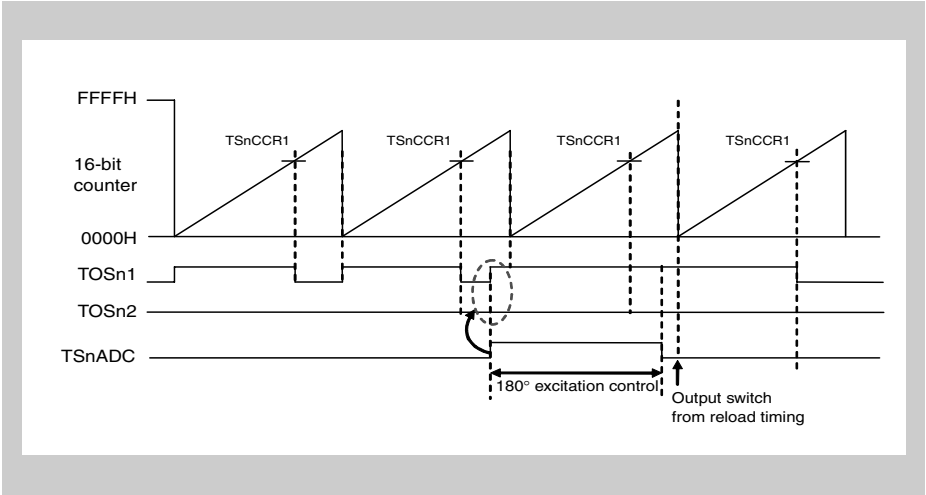


Figure 17-161 Output at switch timing between 120° excitation mode and 180° excitation control (in case of output reverse)

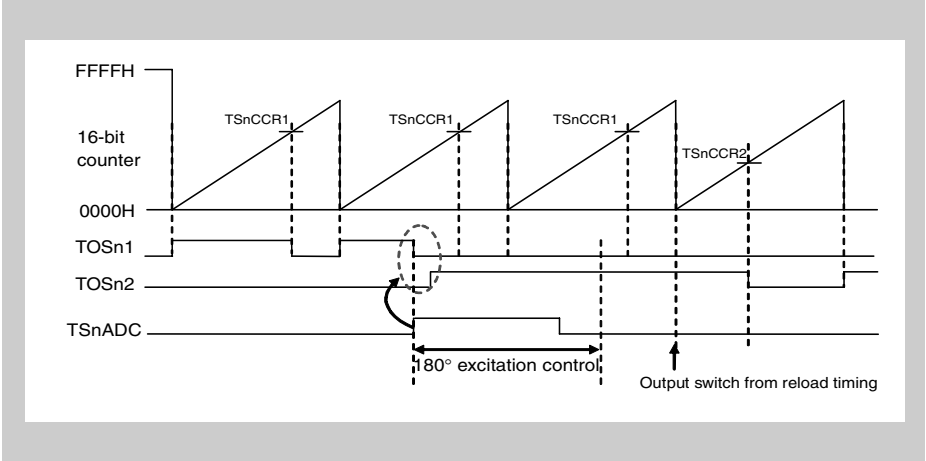


Figure 17-162 Output at switch timing between 120° excitation mode and 180° excitation control (in case of output not reversed)

The output switched during dead time count is set without waiting for the dead time count finish in case of the same phase, or is set after the dead time count finish in case of negative phase.

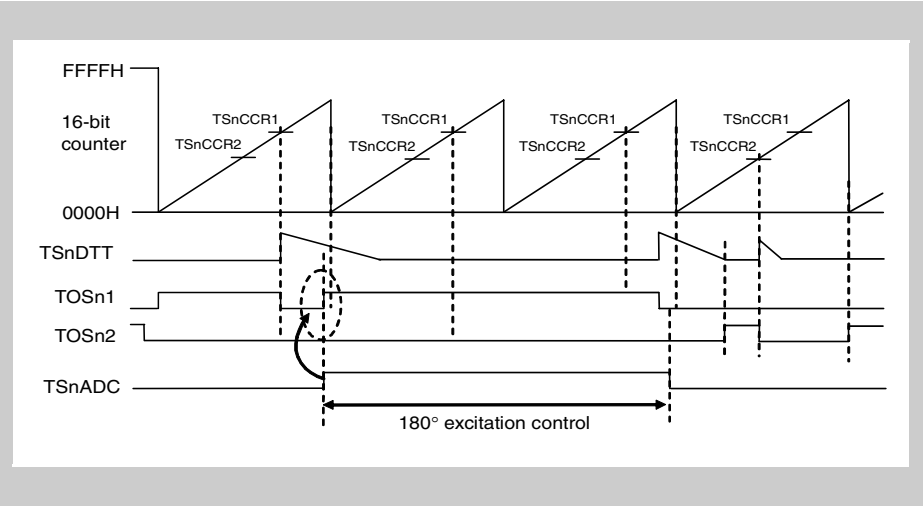


Figure 17-163 Output at switch timing between 120° excitation mode and 180° excitation control (during dead time count)

17.10.11 Special 120° excitation mode

(1) Overview of special 120° excitation mode

In the special 120° excitation mode, 120° excitation control is performed by detecting the external input (pattern: TAPTSn2 to TAPTSn0 pins, trigger: TSnSTCI0 signal) and by switching the timer output (TOSn1 to TOSn6 pins). Motor systems that can be controlled in the special 120° excitation mode are as follows.

- System where a hall sensor is connected at 120° intervals against the motor rotation
- System where normal/reverse rotation is possible by outputting as follows, in response to the hall sensor input

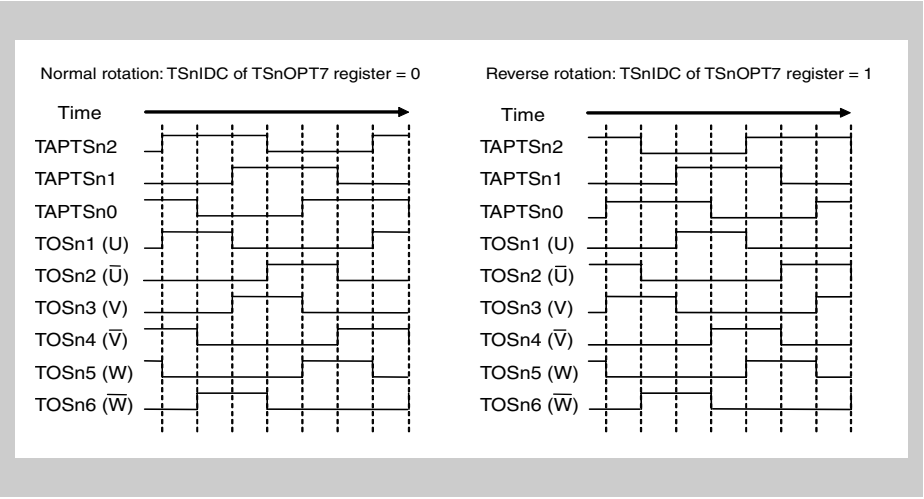


Figure 17-164 Normal/reverse rotation example

Note
1. TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 0
2. The special 120° excitation mode is valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 1, 1].

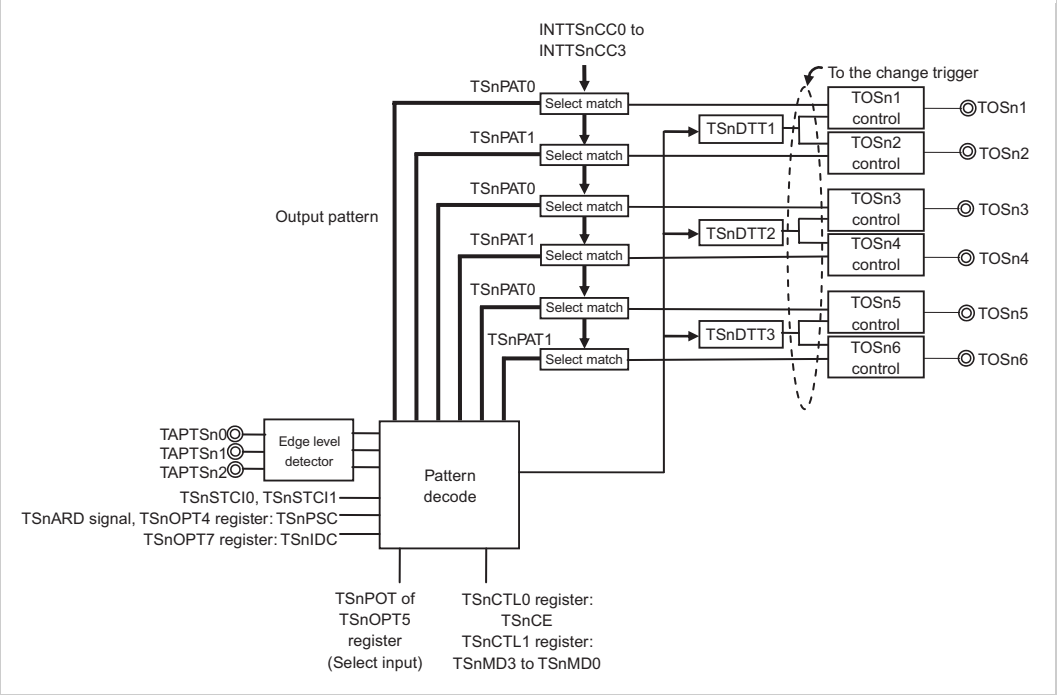


Figure 17-165 Configuration diagram in special 120° excitation mode

Table 17-23 Output Pattern of TAPTSn2 to TAPTSn0 Pins (Normal Rotation)

Output pin	TAPTSn2 to TAPTSn0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note	Note
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note	Note
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note	Note
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note	Note
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note	Note
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note	Note

Note The output level of the TOSn0 to TOSn6 pins remains the level before change when changed to [0, 0, 0] or [1, 1, 1].

Caution Special pattern 3 and special pattern 4 cannot be set to be active simultaneously. Special pattern 0 and special pattern 7 cannot be set to be active simultaneously either. The simultaneous active setting results in the following operations.

1. When TOSn1, TOSn3 and TOSn5 are set to be active prior to TOSn2, TOSn4, TOSn6
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
2. When TOSn2, TOSn4, and TOSn6 are set to be active prior to TOSn1, TOSn3 and TOSn5
   Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set to be active simultaneously
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

There are two methods based on the external input in the special 120° excitation mode.

- Pattern switch method: Pattern input of 3-input (hall sensor, etc.)
- Trigger switch method: Trigger input (encoder circuit, offset, etc.)

(a) Pattern switch method (TSnPOT of TSnOPT5 register = 0)

Level detection is performed for the TAPTSn2 to TAPTSn0 pins (three patterns of input from hall sensor) to decode the signals after level detection. Pattern output of the TOSn1 to TOSn6 pins (the output determined with the values of the TSnCCR1 to TSnCCR3, TSnPAT0, and TSnPAT1 registers) is selected by judging the decode result. Dead time control is performed by the operation of the dead time counter at respective phase signal fall timing to ensure dead time insertion.
With the pattern switch method, the decoded pattern is output provided by using data from the input pattern (TAPTSn2 to TAPTSn0 pins), current direction control bit (TSnIDC of the TSnOPT7 register), and TAPTSn2 to TAPTSn0 order detection flag (TSnTSF of the TSnOPT5 register). *Figure 17-166 on page 972* and *Figure 17-167 on page 973* show the timer output when the TAPTSn2 to TAPTSn0 pins are changed. The output pattern is switched to the pattern corresponding to the input if any error occurs in the input pattern when switching from input pattern 1 to input pattern 4.
The pattern set with the input level of the TAPTSn2 to TAPTSn0 pins, the TSnIDC and TSnARD signals, (TSnPSS of TSnOPT5 register = 0), and the TSnPSC bit of the TSnOPT4 register (TSnPSS of TSnOPT5 register =1) are output immediately after the operation starts (TSnCE of TSnCTL0 register = 1TSnCE = 1 of TSnCTL0 register). The output pattern is determined with the TSnTSF flag instead of the TSnPSC bit after the value of the TSnTSF flag is determined.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 0,
and TSnOPT7 register: TSnIDC = 0.

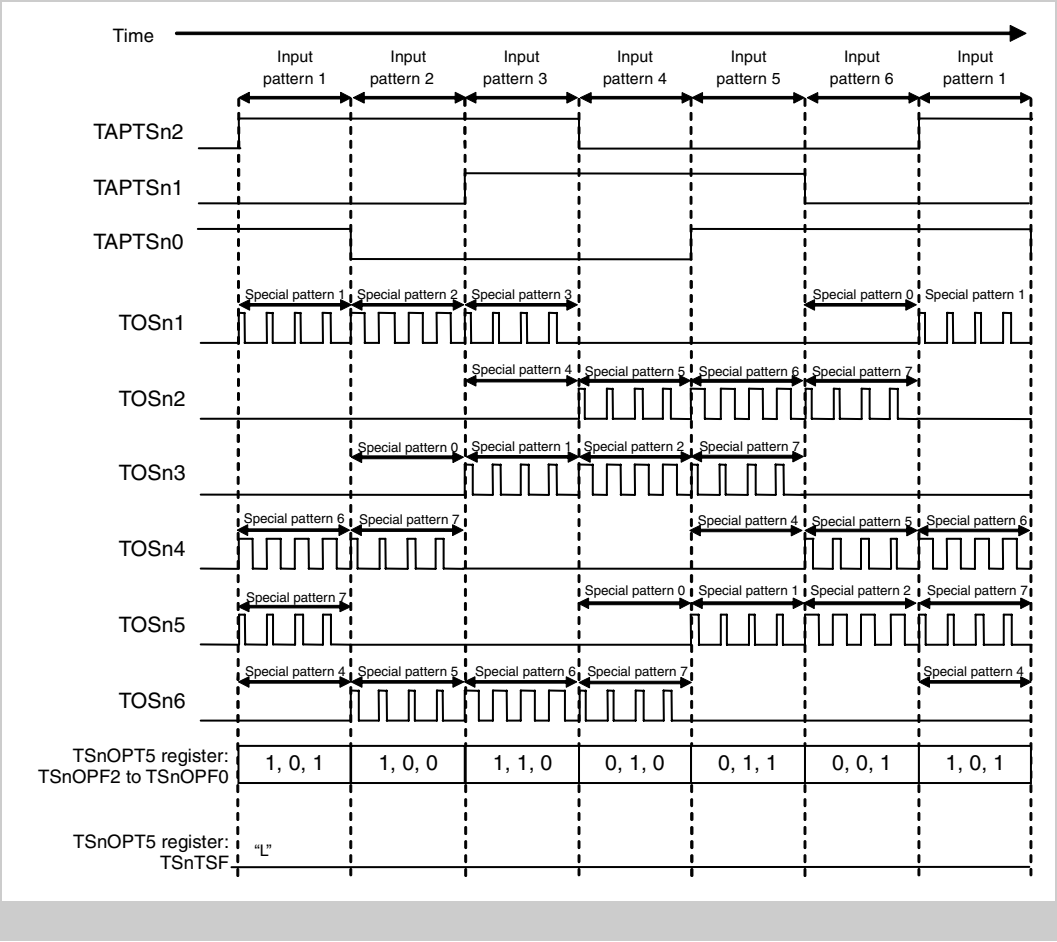


Figure 17-166 Special 120° excitation mode operation example in use of pattern switch method (normal rotation)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 0,
and TSnOPT7 register: TSnIDC = 1.

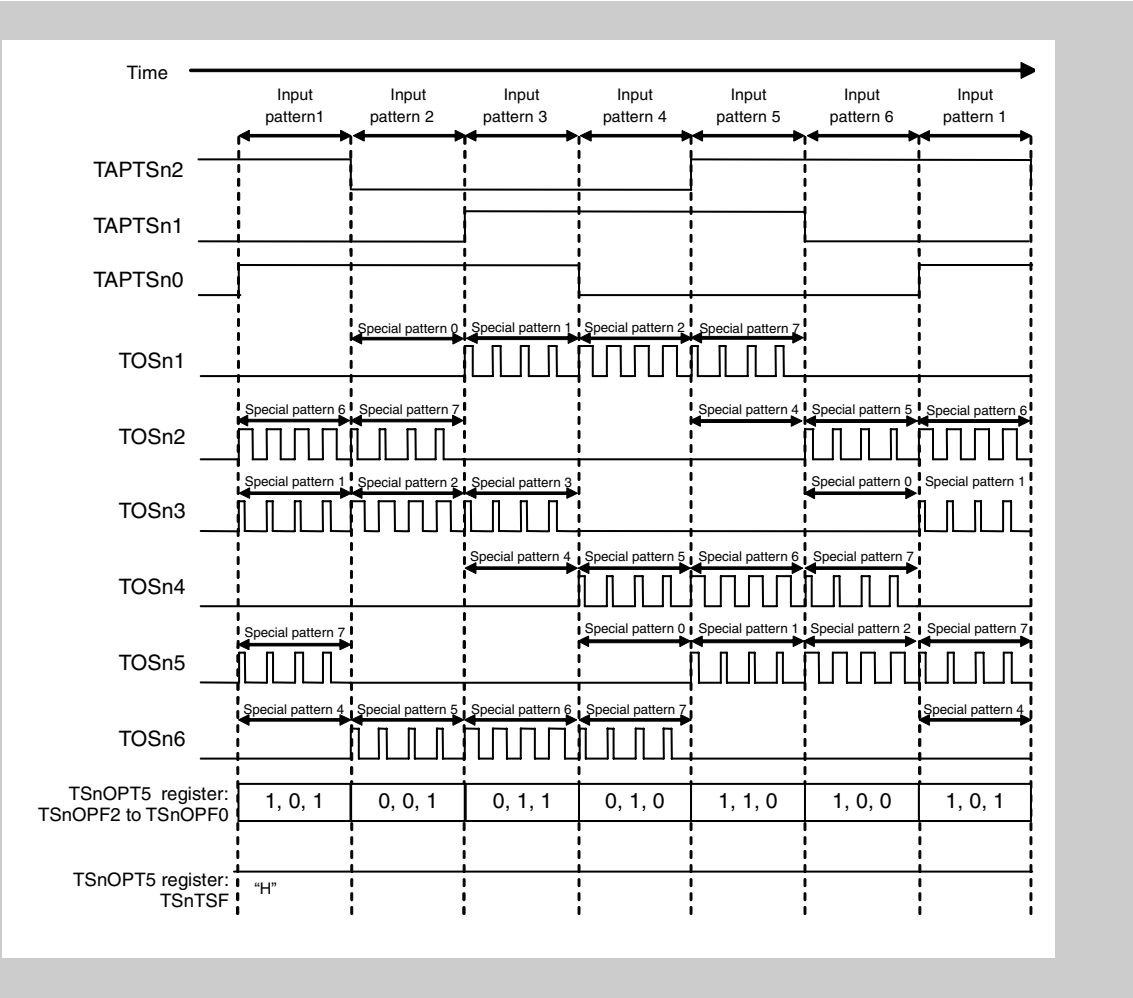


Figure 17-167 Special 120° excitation mode operation example in use of pattern switch method (reverse rotation)

(b) Trigger input switch method (TSnPOT of TSnOPT5 register = 1)

With the trigger input switch method, rise of theTSnSTCI0 and TSnSTCI1 signals is detected to generate the output switch timing. Timer output initial pattern is set with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. The output patterns after the initial pattern are determined with the TSnIPC2 to TSnIPC0 bits, rotation direction or the TSnIDC bit of the TSnOPT7 register. There are four patterns that can be output according to the combination of rotation direction and the TSnIDC bit.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
1	0	Determine rotation direction by using input signal to TSnARD
	1	Determine rotation direction by using TSnPSC bit

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

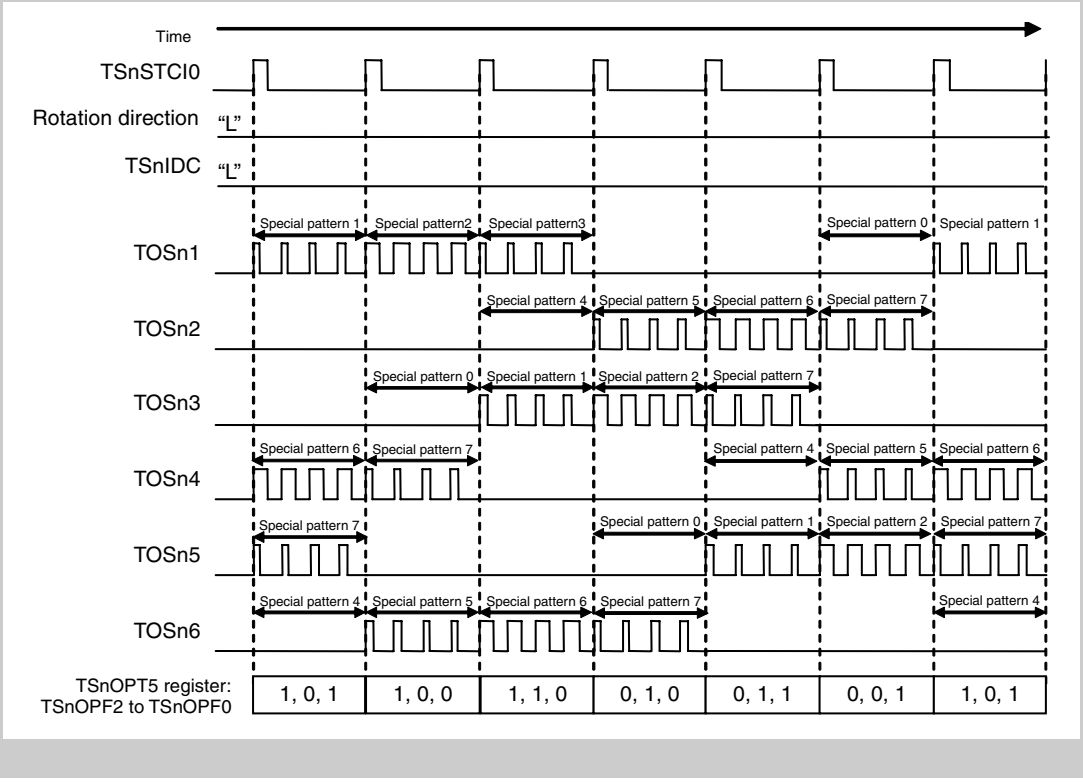


Figure 17-168 Special 120° excitation mode operation example in use of trigger input switch method (normal rotation)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

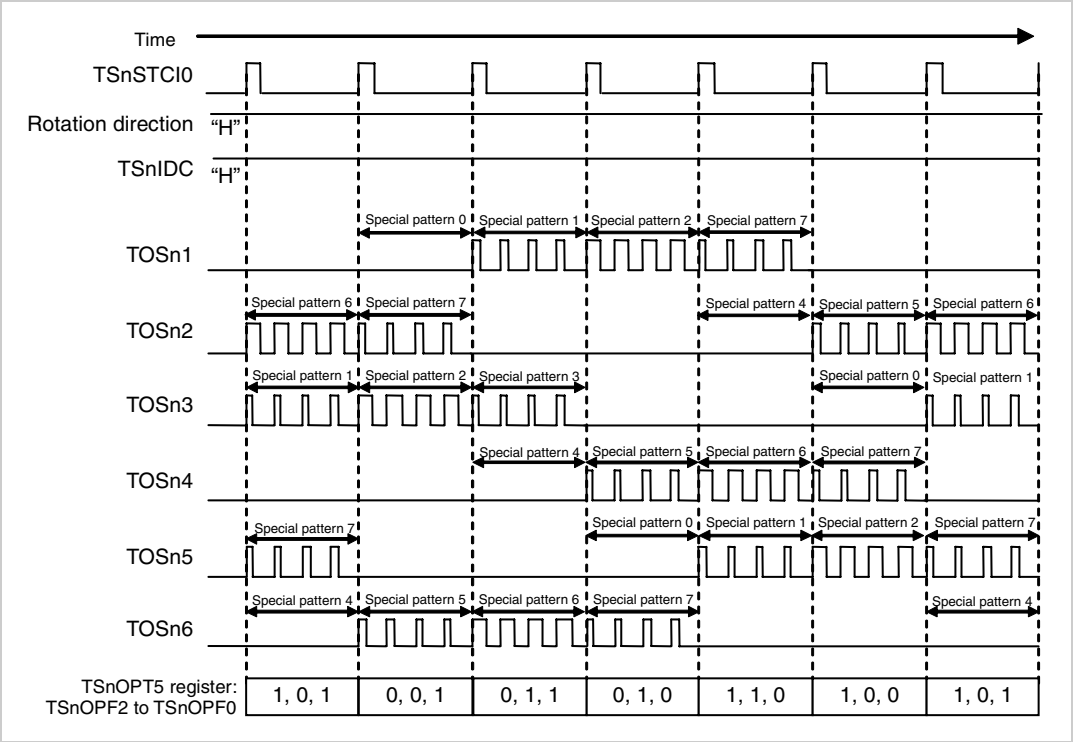


Figure 17-169 Special 120° excitation mode operation example in use of trigger input switch method (reverse rotation)

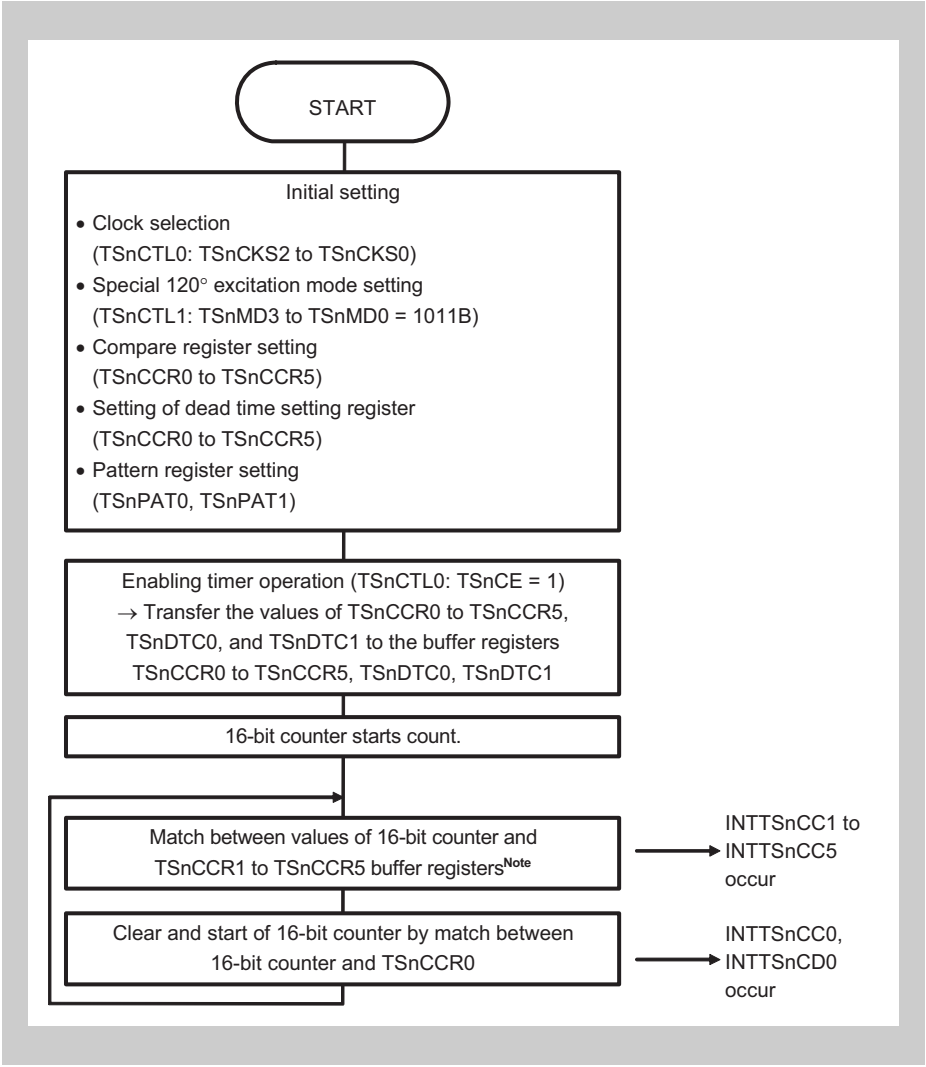


Figure 17-170 Basic operation flow in special 120° excitation mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

(2) Special 120° excitation mode operation List

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	PWM duty
TSnCCR4, TSnCCR5	Reload	Possible	Possible to select as A/D trigger
TSnDTC0, TSnDTC1	Reload	Conditionally possible[Note]	Dead time
TSnPAT0, TSnPAT1	Reload	Possible	Special pattern setting

Note Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* for details.

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-
TAPTSn2 to TAPTSn0	Pattern input
TSnSTCI0, TSnSTCI1 signal	Trigger input
TSnARD signal	Pattern change order input

(c) Output pin

Pin	Function
TOSn0	Toggle output by TSnCCR0 compare match
TOSn1	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT0
TOSn2	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT1
TOSn3	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT0
TOSn4	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT1
TOSn5	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT0
TOSn6	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT1
TOSn7	Pulse output by A/D conversion trigger

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of INTTSnCC0 to INTTSnCC5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing with INTTSnCC0)
INTTSnWN	Warning

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

Note “–” indicates an unused function in the special 120° excitation mode.

(3) Setting of special 120° excitation mode

(a) Mode setting

The special 120° excitation mode is set by setting TSnMD3 to TSnMD0 of the TSnCTL1 register to [1, 0, 1, 1].

(b) Output level/output enable setting

Output level/output enabling is performed by setting the TSnOL0 to TSnOL7, TSnOE0 to TSnOE7 bits of the TSnIOC0 register.
Toggle output is performed from the TOSn0 pin with the cycle match (match between 16-bit counter and TSnCCR0 register).
The TOSn7 pin output is for A/D conversion. Enable the output if necessary.

(c) Error interrupt/warning interrupt occurrence enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1 when positive/negative phase simultaneous active state is detected.
A warning interrupt (INTTSnWN) is enabled by setting TSnWOC of the TSnIOC4 register to 1.

(d) Interrupt and thinning out function setting

A peak interrupt (INTTSnCD0) occurs upon a match between the TSnCCR0 register and 16-bit counter. (Control of the TSnIOE bit of the TSnOPT1 register is invalid). Set TSnICE of the TSnOPT1 register to 1 to output a peak interrupt.
To use the thinning out function or peak interrupt, execute setting with the TSnID4 to TSnID0 bits of the TSnOPT1 register.

(e) Reload thinning out function setting

Set TSnRDE of the TSnOPT1 register to 1 to set reload timing to the same timing with the interrupt timing.

(f) A/D conversion trigger output setting

To set A/D conversion trigger 0 (TSnADTRG0 signal), set whether to enable/disable upon match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at peak interrupt (INTTSnCD0) by using the TSnAT04, TSnAT02, and the TSnAT01 bits of the TSnOPT2 register.
To set A/D conversion trigger 1 (TSnADTRG1 signal), set whether to enable/disable upon the match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at peak interrupt (INTTSnCD0) by using the TSnAT14, TSnAT12, and TSnAT11 bits of the TSnOPT3 register.
Set the compare value setting for the TSnCCR4 and TSnCCR5 registers.
Thinning out function is enabled for the TSnADTRG0 and TSnADTRG1 signals.
No thinning out, 1-thinning out, 3-thinning out and 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

Caution

1. When using the TOSn7 pin, set the TSnOPT2 and TSnOPT3 registers and the TSnCCR4 and TSnCCR5 registers appropriately according to the system.
2. Trough interrupts (INTTSnOD) do not occur in the special 120° excitation mode. Be sure to set TSnAT00, 10 bits of TSnOPT2 and TSnOPT3 registers to 0.
3. The 16-bit sub-counter does not operate in the special 120° excitation mode.
   Be sure to set TSnAT07, 06, TSnAT17, 16 bits of TSnOPT2, TSnOPT3 registers to 0.
4. Count-down status of the 16-bit counter does not occur in the special 120° excitation mode. Be sure to set TSnAT05, 03, TSnAT15, 13 bits of TSnOPT2, TSnOPT3 registers to 0.

(g) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.
Dead time can be calculated with the following equation.
16-bit counter operation clock cycle × TSnDTC0
16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of the TOSn2, TOSn4, TOSn6 pins to the active state of the TOSn1, TOSn3, TOSn5 pins can be set with the TSnDTC0 register.

The time for the change from the inactive state of the TOSn1, TOSn3, TOSn5 pins to the active state of the TOSn2, TOSn4, TOSn6 pins can be set with TSnDTC1 register.

(h) Special pattern output setting

In the special 120° control, the output of the TOSn1, TOSn3, and TOSn5 pins is controlled with TSnCCR0 to TSnCCR3 registers, and the output of the TOSn2, TOSn4, and TOSn6 pins is controlled with TSnCCR0 to TSnCCR3, and TSnPAT1 registers. The setting enables special pattern output control.

(i) Pattern output switch timing, pattern output order, and initial output pattern setting

[Pattern switch method]

The pattern switch method is enabled by setting TSnPOT of the TSnOPT5 register to 0. Output of pins TOSn1 to TOSn6 is switched at the change timing of the TAPTSn2 to TAPTSn0 pins.
The output order at operation start is set with the TSnIDC bit of the TSnOPT7 register. The initial output pattern is set with the TSnPSC bit of the TSnOPT4 register. However, setting with the TSnPSC bit is invalid after the rotation direction is determined (any value is set in the TSnTSF bit of the TSnOPT5 register).

[Trigger switch method]

Trigger switch method will be enabled when setting TSnPOT of TSnOPT5 register to 1.
Output of the TOSn1 to TOSn6 pins is switched with the rise of the external input (TSnSTCI1, TSnSTCI0 signals).
Output order is set in combination of TSnARD signal, TSnPSC bit of the

TSnOPT4 register, or TSnIDC bit of TSnOPT7 register.
Selection of TSnARD signal or TSnPSC bit of the TSnOPT4 register shall be set with the TSnPSS bit of the TSnOPT4 register.
Output order is controlled with the TSnARD input signal when TSnPSS is set to 0, and controlled with the TSnPSC bit when TSnPSS is set to 1. Refer to *"Timer output pattern in each mode and function" on page 739* for the pattern output order.
Initial output pattern can be controlled with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register.
Initial pattern is output upon the timer Sn operation start setting (when TSnCE of TSnCTL0 register is set to 1) after setting with the TSnIPC2 to TSnIPC0 bit. Refer to *"Timer output initial pattern setting in each mode and function" on page 731* for details.

Caution Set the initial pattern by reading the input level of the port connected to the TAPTSn2 to TAPTSn0 pins.

(4) Operation in special 120° excitation mode

Figure 17-171 on page 982 and *Figure 17-172 on page 983* show operation examples in the special 120° excitation mode. TOSn1 to TOSn6 detect input level change timing of the TAPTSn2 to TAPTSn0 pins to change the output pattern.
The 16-bit counter operation is a saw-tooth wave operation to output a special pattern set with the TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers. The 16-bit counter is not cleared to 0000H until matched with PWM cycle TSnCCR0 register value even if the change of the TAPTSn2 to TAPTSn0 pins is detected. Special output pattern is switched from the following PWM cycle where the change of the TAPTSn2 to TAPTSn0 pins is detected.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

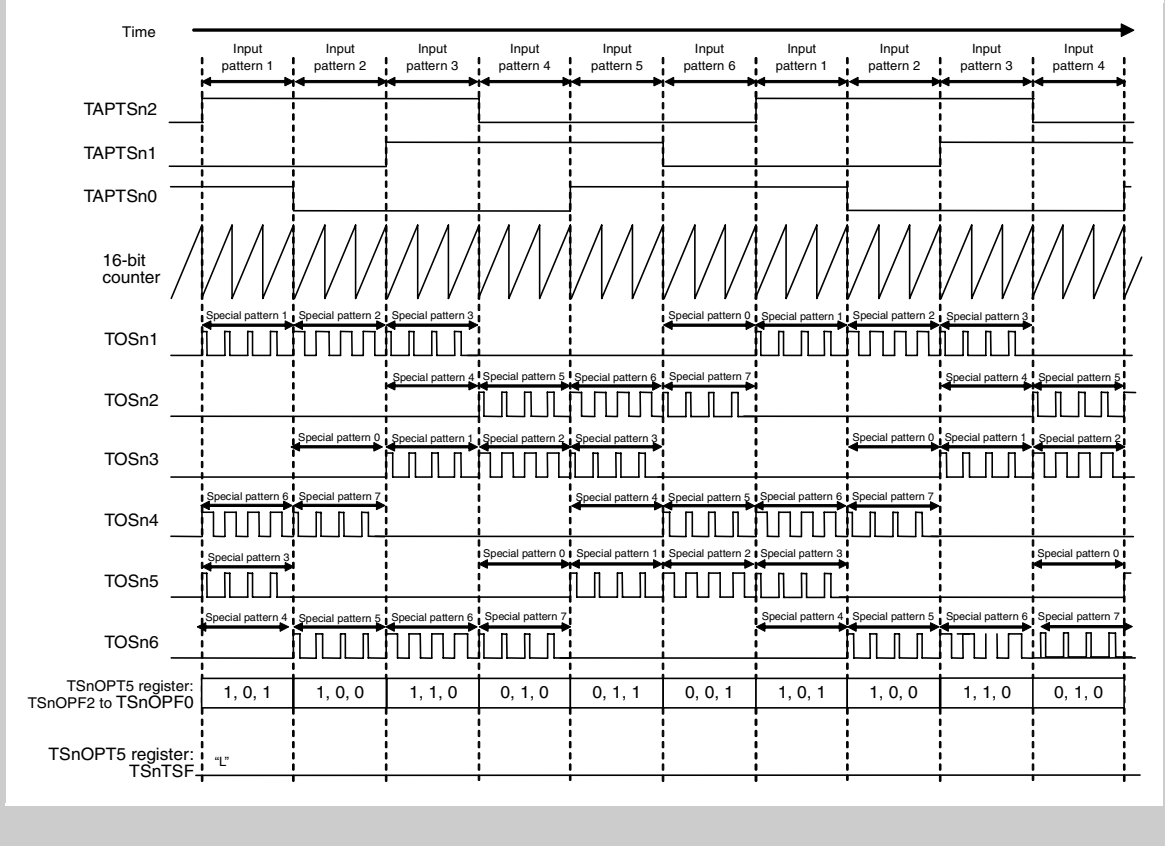


Figure 17-171 Special 120° excitation mode basic operation timing example (pattern switch method, normal rotation)

Note Refer to *"Description of special pattern of special 120° excitation mode" on page 987* for details on the special pattern.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

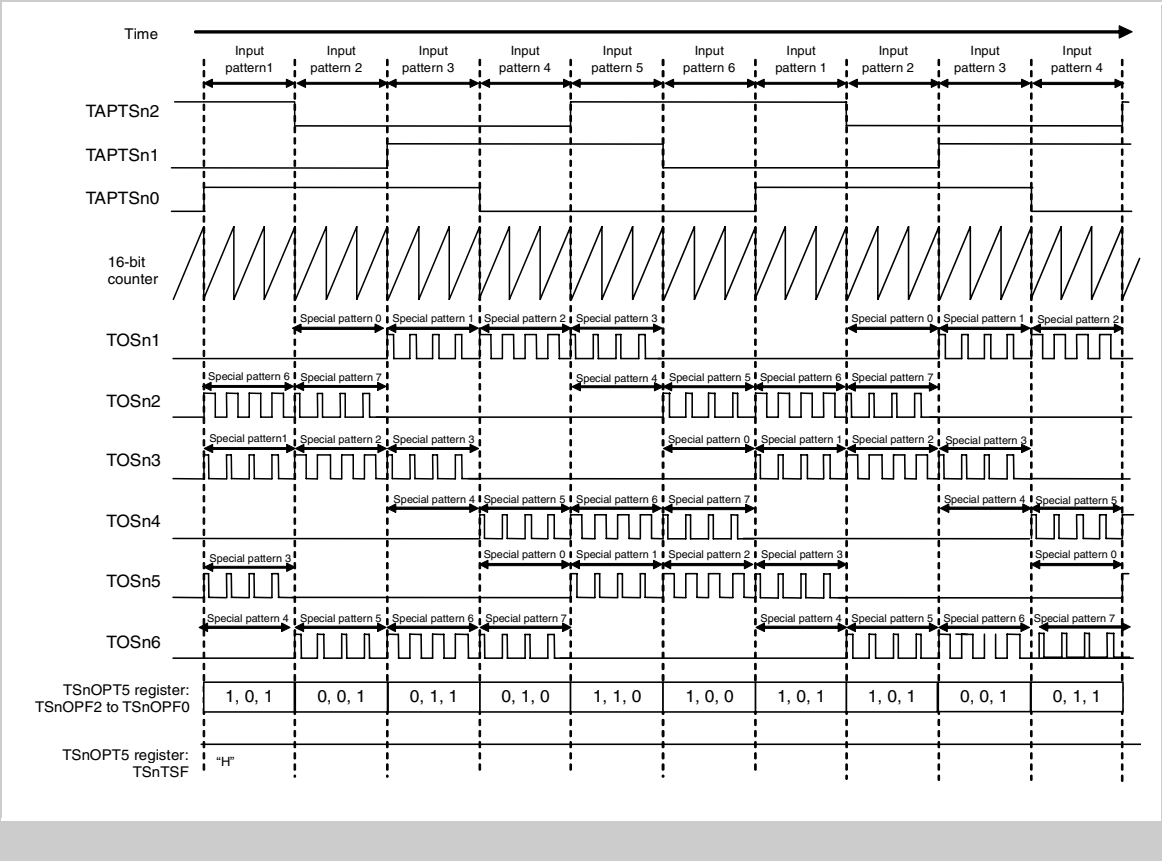


Figure 17-172 Special 120° excitation mode basic operation timing example (pattern switch method, reverse rotation)

Note Refer to *"Description of special pattern of special 120° excitation mode" on page 987* for details on the special pattern.

(5) Output pattern list in special 120° excitation mode

In the special 120° excitation mode, the output pattern is determined by rotation direction and the TSnIDC bit of the TSnOPT7 register.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
0	-	TSnTSF flag
1	0	Input signal to TSnARD
1	1	TSnPSC bit

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0[Note 1]/TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0[Note 1]/TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	Note 1	Note 2
TOSn2	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Note 1	Note 2
TOSn3	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Note 1	Note 2
TOSn4	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	Note 1	Note 2
TOSn5	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Note 1	Note 2

Note 1. The output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1. Then, output is switched in the pattern switch order when pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 are not changed even if the output pattern is switched.

2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.

3. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. When TOSn1, TOSn3 and TOSn5 are set active prior to TOSn2, TOSn4, and TOSn6
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
2. When TOSn2, TOSn4, and TOSn6 are set active prior to TOSn1, TOSn3 and TOSn5
   Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set active simultaneously
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

Note 1. Special pattern 0 to special pattern 7 indicate the special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers.

2. INACT indicates the inactive level output.

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0[Note 1]/TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	Note 1	Note 2
TOSn3	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	Note 1	Note 2
TOSn4	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Note 1	Note 2
TOSn5	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Note 1	Note 2
TOSn6	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,
TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0[Note 1]/TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	Note 1	Note 2
TOSn2	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Note 1	Note 2
TOSn3	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Note 1	Note 2
TOSn4	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	Note 1	Note 2
TOSn5	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Note 1	Note 2

Note
1. Output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1.
   Then, output is switched in the pattern switch order when a pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 are not changed even if the output pattern is switched.
2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
3. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be set active. Special pattern 0 and special pattern 7 cannot be set active, either. The simultaneous active setting leads to the following operations.

1. When TOSn1, TOSn3 and TOSn5 are set active prior to TOSn2, TOSn4, and TOSn6
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
2. When TOSn2, TOSn4, and TOSn6 are set active prior to TOSn1, TOSn3 and TOSn5
   Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set active simultaneously
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

Note
1. Special patterns 0 to 7 indicate the special pattern output by the TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers.
2. INACT indicates the inactive level output.

(6) Description of special pattern of special 120° excitation mode

Special pattern is set with the TSnCCR0 to TSnCCR3 registers and the TSnPAT0 and TSnPAT1 registers.
The functions of the special pattern registers (TSnPAT0, TSnPAT1) are classified in 4-bit units. A special pattern is set with each 4-bit and the 8 special patterns can be executed as follows.

Special pattern 0:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[3:0]
Special pattern 1:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[7:4]
Special pattern 2:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[11:8]
Special pattern 3:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[15:12]
Special pattern 4:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[3:0]
Special pattern 5:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[7:4]
Special pattern 6:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[11:8]
Special pattern 7:	Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[15:12]

In the special 120° excitation mode, output of 4 patterns can be set in the positive phase with TSnPAT0 and output of 4 patterns settable in negative phase with TSnPAT1 are possible.

In the special 120° excitation mode, priority is as follows when the same value is set to the TSnCCR0 to TSnCCR3 registers.
Output setting after TSnCCR3 match > output setting after TSnCCR2 match > output setting after TSnCCR1 match > output setting after TSnCCR0 match

Priority of every special pattern is as follows.

Special pattern 0:	TSnPAT0[3] > TSnPAT0[2] > TSnPAT0[1] > TSnPAT0[0]
Special pattern 1:	TSnPAT0[7] > TSnPAT0[6] > TSnPAT0[5] > TSnPAT0[4]
Special pattern 2:	TSnPAT0[11] > TSnPAT0[10] > TSnPAT0[9] > TSnPAT0[8]
Special pattern 3:	TSnPAT0[15] > TSnPAT0[14] > TSnPAT0[13] > TSnPAT0[12]
Special pattern 4:	TSnPAT1[3] > TSnPAT1[2] > TSnPAT1[1] > TSnPAT1[0]
Special pattern 5:	TSnPAT1[7] > TSnPAT1[6] > TSnPAT1[5] > TSnPAT1[4]
Special pattern 6:	TSnPAT1[11] > TSnPAT1[10] > TSnPAT1[9] > TSnPAT1[8]
Special pattern 7:	TSnPAT1[15] > TSnPAT1[14] > TSnPAT1[13] > TSnPAT1[12]

Setting and output examples are shown below.

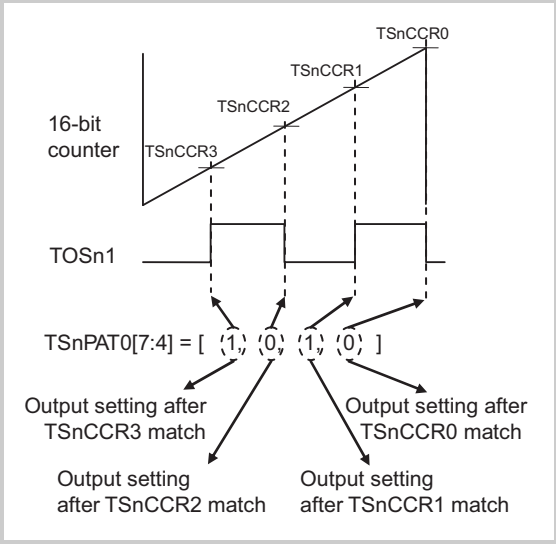


Figure 17-173 Special pattern setting example (1/2)

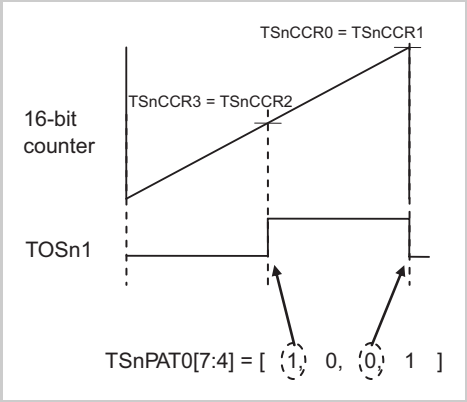


Figure 17-174 Special pattern setting example (2/2)

If the compare setting values of TSnPAT0[7] and TSnPAT0[6] are the same, TSnPAT0[7] has higher priority according to the priority order (TSnPAT0[7] > TSnPAT0[6] > TSnPAT0[5] > TSnPAT0[4]).

This applies to TSnPAT0[5] and TSnPAT0[4] as well.

(7) Output switch timing in special 120° excitation mode

In the special 120° excitation mode, external output pattern switch timing (TSnSTCI0 and TSnSTCI1 signals, TAPTSn2 to TAPTSn0 pins) is input regardless of the 16-bit counter operation, as shown in *Figure 17-175 on page 989*, *Figure 17-176 on page 990*, *Figure 17-177 on page 990*, and *Figure 17-178 on page 991*. The output switch timing starts from the following cycle of external input switch timing. Therefore, special pattern output will be held until the current cycle is completed even if the switch timing is input from the outside.
When the TAPTSn2 to TAPTSn0 pins are changed serially within the first cycle in the pattern switch method, the edge-detection value just before the cycle match is used and reflected to the special pattern output from the following cycle.
If the TSnSTCI0 and TSnSTCI1 signal triggers are input several times within a cycle in the trigger switch method, the pattern is switched to adjacent output pattern due to only once acceptance.
Rewriting of TSnIPC2 to TSnIPC0 is performed prior to the TSnSTCI0, TSnSTCI1 trigger if both occur within a cycle in the trigger switch method. The value rewritten just before cycle match is reflected if TSnIPC2 to TSnIPC0 are rewritten several times in a cycle.

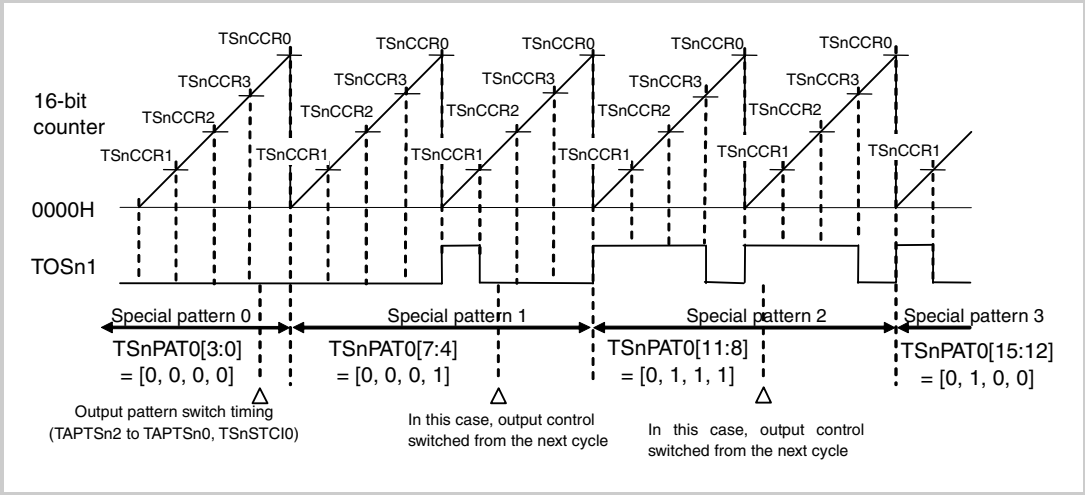


Figure 17-175 Output switch example (TAPTSn2 to TAPTSn0 pins, TSnSTCI0, TSnSTCI1 signal trigger input)

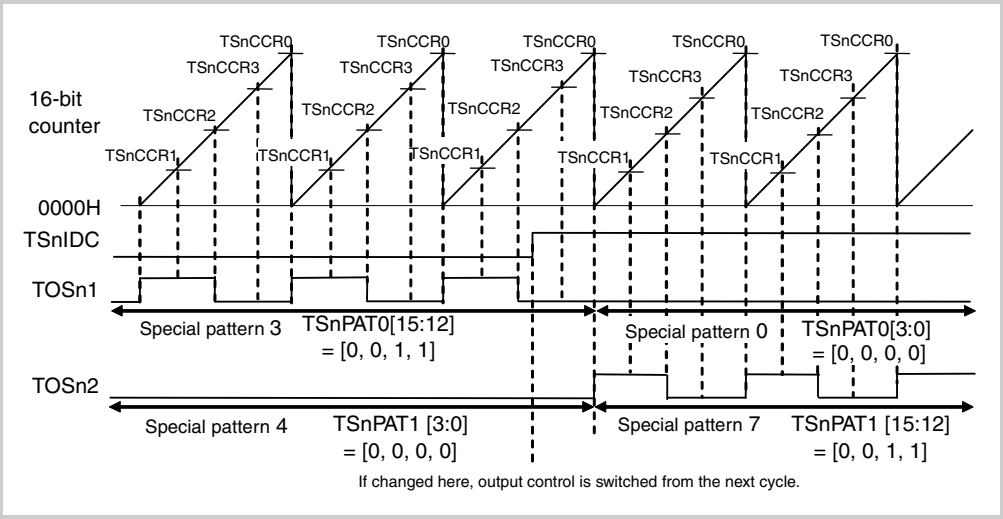


Figure 17-176 Output switch example (switch with TSnIDC bit of TSnOPT7 register)

(TSnOPT5 register: TSnPOT bit = 1)

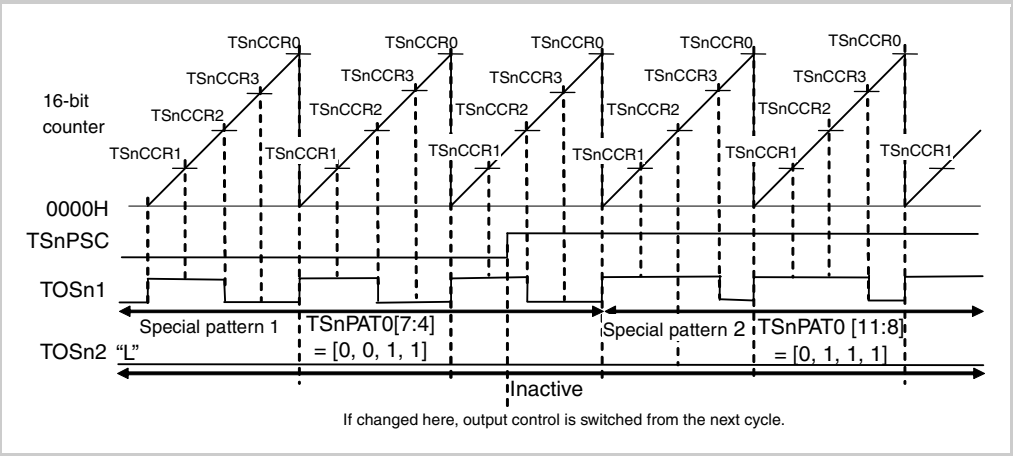


Figure 17-177 Output switch example (switch with TSnPSC bit of TSnOPT4 register)

TSnOPT4 register: TSnPSC = 0, TSnOPT5 register:
TSnPOT = 1, TSnPSS = 1

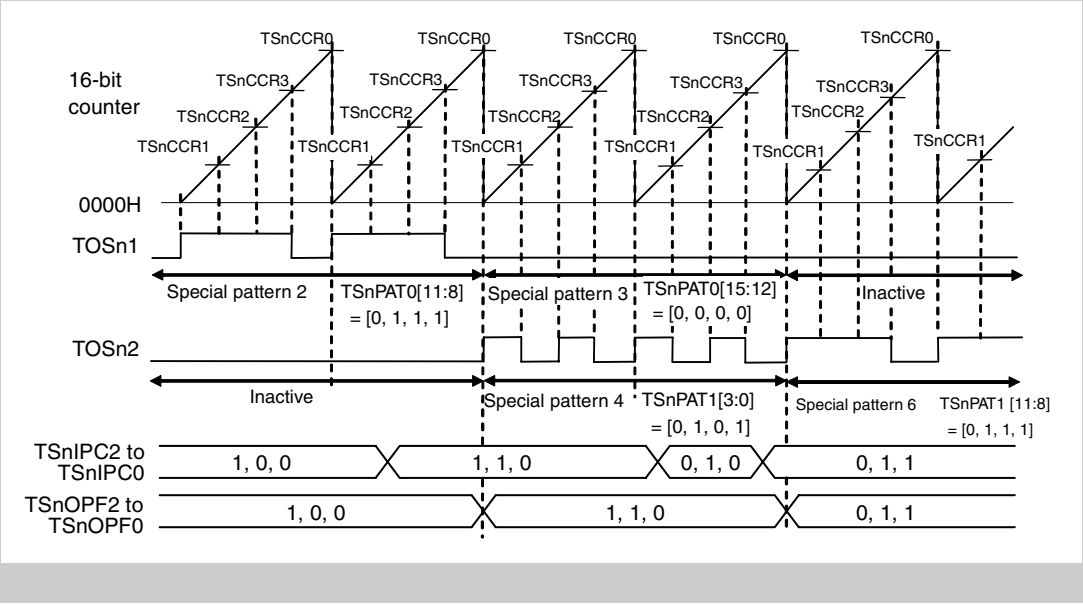


Figure 17-178 Output switch example (switch with TSnIPC2 to TSnIPC0 bits of TSnOPT4 register)

(8) Compare register rewrite timing in special 120° excitation mode

An operation example when the TSnCCR1 register is reloaded (batch rewriting) is described below.
Figure 17-179 on page 992 shows an output example in case of rewriting of the TSnCCR1 register.
After change of the TSnCCR1 register, data is not transferred to the TSnCCR1 buffer register until the next reload timing (the changed data will not be valid), so that the output waveform can be obtained as set. However, rewriting the TSnCCR1 register again is prohibited during the reload pending (in the period from changing the TSnCCR1 register to the batch rewriting).
Be sure to read the reload request flag (TSnRSF) to confirm 0 status when writing to the TSnCCR1 register.

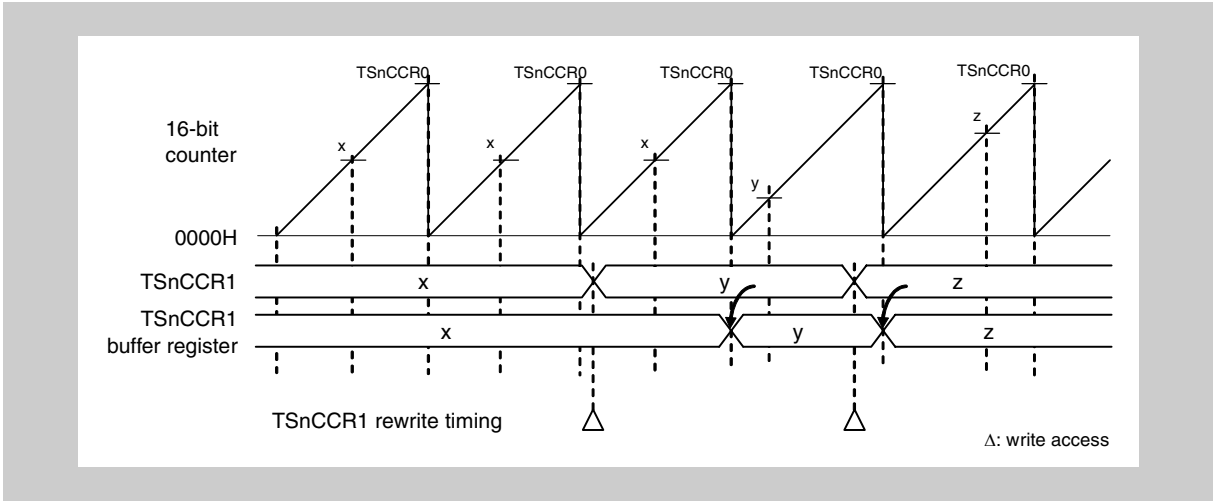


Figure 17-179 Output example in case of rewriting of TSnCCR1 register

(9) Pattern rewrite timing in special 120° excitation mode

The operation example in case of rewriting of the TSnPAT0 register is described below.
TSnPAT0 register setting value is transferred to the buffer register at reload timing after rewriting (at TSnCCR0 register match timing after writing to TSnCCR1 register). The setting is valid at the timing when the 16-bit counter is cleared. Then, the expected pattern can be output.

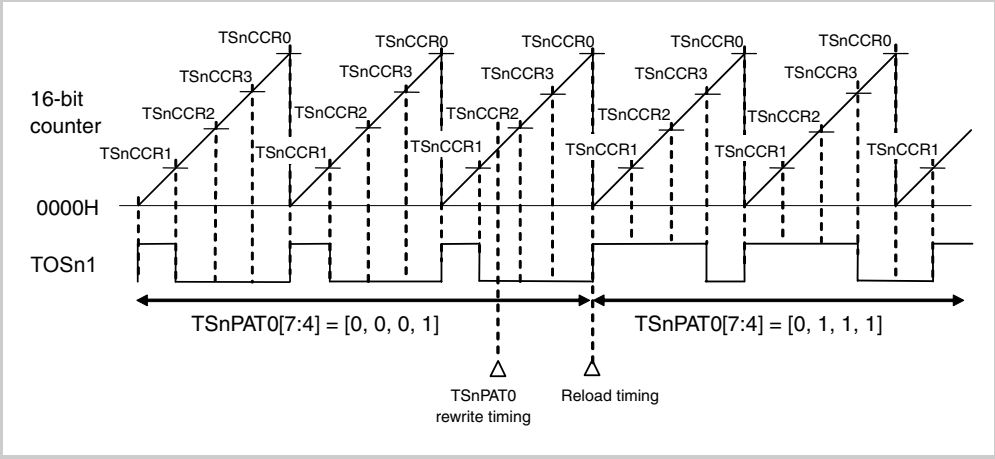


Figure 17-180 Pattern rewrite timing example

(10) Dead time control in special 120° excitation mode

In the special 120° excitation mode, dead time is added by the dead time control operation at the fall of each phase.
The dead time that is set to the TSnDTC1 register is inserted at the fall of the positive phase, and the dead time that is set to the TSnDTC0 register is inserted at the fall of the negative phase.

TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 1

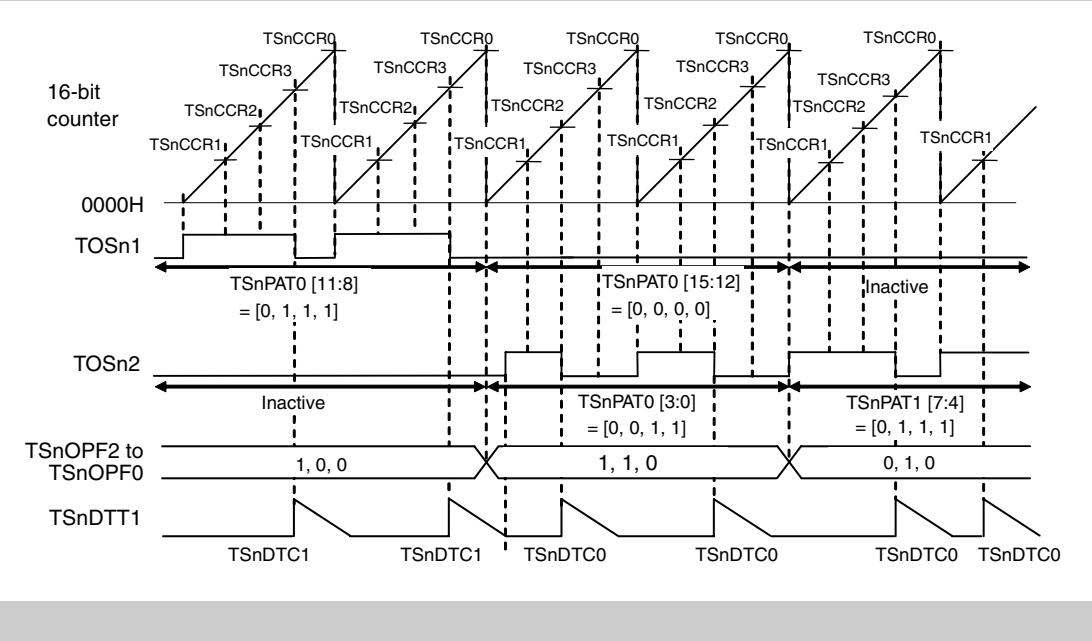


Figure 17-181 Dead time control example

Caution Dead time control has little effect on the timer output during normal operation. However, dead time control may affect the timer output under the following conditions.

- When noise occurs in input pattern during use of the pattern switch method
- When the input pattern changes earlier than PWM cycle during use of the pattern switch method
- When output pattern is forcibly changed by changing the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register in the trigger switch method
- When the switch method is changed
- When the current direction control bit (TSnIDC bit of TSnOPT7 register) is changed
- When software output control function is used
- When 180° excitation control is used
- When special pattern 3 and special pattern 4 are set active simultaneously
- When special pattern 0 and special pattern 7 are set active simultaneously

(11) Output switch in special 120° excitation mode

In the special 120° excitation mode, the output pattern can be controlled by writing the value to the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register when the trigger switch method is used (TSnPOT of TSnOPT5 register is set to 1) and also the TSnADC bit of the TSnOPT5 register is set to 0. Dead time is secured by using hardware at the switch timing.

Caution

1. The output level of TOSn1 to TOSn6 pins is kept when [1, 1, 1] or [0, 0, 0] is written to TSnIPC2 to TSnIPC0.
2. Do not rewrite more than once the TSnIPC2 to TSnIPC0 bits in a PWM cycle. If rewritten twice or more, the value just before PWM cycle match occurrence is reflected.

(12) Software output control function in special 120° excitation mode

In the special 120° excitation mode, timer output control can be executed with software control by using the TSnSOC, TSnIPC2, TSnIPC1, and TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.
Refer to *"Software output control function in 120° excitation mode" on page 960* for the operation and setting procedure.
Refer to *"Software output function" on page 1014* for details on the software output control function.

(13) 180° excitation control function in special 120° excitation mode

In the special 120° excitation mode, 180° excitation control function is enabled by using the TSnADC, TSnIPC2, TSnIPC1, and TSnIPC0 bits of the TSnOPT5 register, and the TSnIDC bit of the TSnOPT7 register.
Refer to *"180× excitation control function in 120° excitation mode" on page 963* for the operation and setting procedure.
Refer to *"180° excitation control function" on page 1016* for details on the 180° excitation control.

(14) Cautions about setting of TSnPAT0, TSnPAT1 registers in special 120° excitation mode

The setting where the simultaneous active level is output to positive and negative phases is possible in the special 120° excitation mode. Practically, this means the case when bits 3 to 0 of the TSnPAT0 register and bits 15 to 12 of the TSnPAT1 register are set to any value other than [0, 0, 0, 0], or the case when bits 15 to 12 of the TSnPAT0 register and bits 3 to 0 of the TSnPAT1 register are set to any value other than [0, 0, 0, 0]. However, the following circuit operation is used because it is risky to be simultaneous active level of positive and negative phases, taking the inverter control into account. *Figure 17-182 on page 996* shows an example in case of the setting to make the TOSn1 pin and TOSn2 pin active simultaneously. In this case, the TOSn1 pin is set by priority at the TOSn1, TOSn2 pins simultaneous rise timing. The TOSn3 pin is set by priority between the TOSn3 and TOSn4 pins, and the TOSn5 pin is set by priority between the TOSn5 and TOSn6 pins.

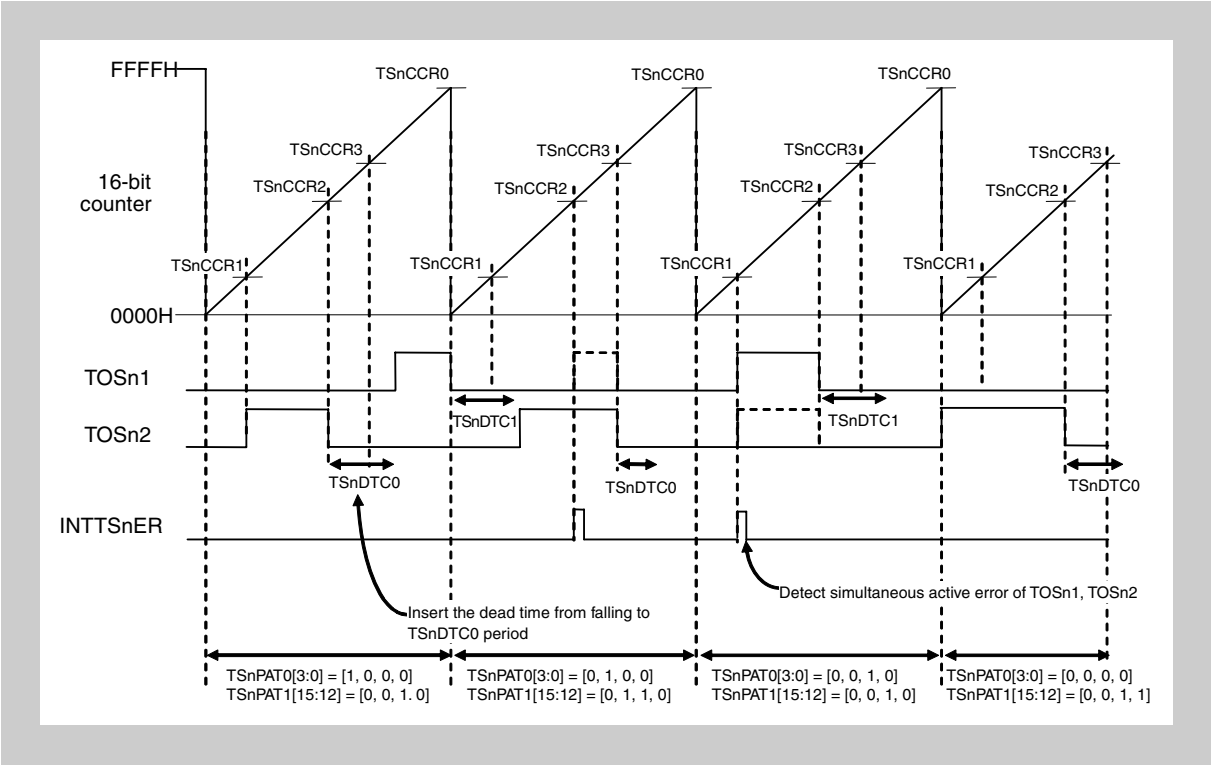


Figure 17-182 Output Control at Simultaneous Active Occurrence

(15) Difference between 120° excitation mode and special 120° excitation mode

The description of difference between the 120° excitation mode and the special 120° excitation mode are described below.

(a) Timer output difference

120° excitation mode

Positive phase:PWM1, PWM2 output
Negative phase:PWM3, PWM4 output

Special 120° excitation mode

Positive phase:Special pattern 0, special pattern 1, special pattern 2, special pattern 3
Negative phase:Special pattern 4 special pattern 5, special pattern 6, special pattern 7

(b) Difference of electric angle that can be excited

120° range of electric angle can be output during the PWM output period in the 120° excitation mode.

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 0
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0[Note 1]/TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 2	Note 3
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 2	Note 3
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 2	Note 3
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 2	Note 3
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 2	Note 3

In the special 120° excitation mode, output range can be increased by 60° forward/backward respectively for the electric angle with 120° excitation control output.

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,
TSnOPT5 register: TSnPOT = 0
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0[Note 1]/TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
2. Output remains the level before the change.

Caution Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously, either. The simultaneous active setting leads to the following operations.

1. When TOSn1, TOSn3 and TOSn5 are set active prior to TOSn2, TOSn4, and TOSn6
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
2. When TOSn2, TOSn4, and TOSn6 are set active prior to TOSn1, TOSn3 and TOSn5
   Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set active simultaneously
   Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

Note
1. Each column indicates 60° of electric angle.
2. PWM1 to PWM4 indicate PWM output by the TSnCCR1 to TSnCCR4 registers.
3. Special pattern 0 to special pattern 7 indicate the special pattern output by the TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers.
4. INACT indicates the inactive level output.

17.10.12 Special pattern output mode

(1) Special pattern output mode overview

16-bit counter saw-tooth wave operation and four 16-bit compare registers are used in special pattern output mode. The value that is set by the TSnPAT0 and TSnPAT1 registers is output to the TOSn0 to TOSn7 pins upon a match between the 16-bit counter and TSnCCR0 to TSnCCR3 register values.

TSnPAT0[3:0]	TOSn0 pin output control
TSnPAT0[7:4]	TOSn1 pin output control
TSnPAT0[11:8]	TOSn2 pin output control
TSnPAT0[15:12]	TOSn3 pin output control
TSnPAT1[3:0]	TOSn4 pin output control
TSnPAT1[7:4]	TOSn5 pin output control
TSnPAT1[11:8]	TOSn6 pin output control
TSnPAT1[15:12]	TOSn7 pin output control

Note Special pattern output mode is valid when TSnMD3 to TSnMD0 of TSnCTL1 register is set to [1, 1, 0, 0].

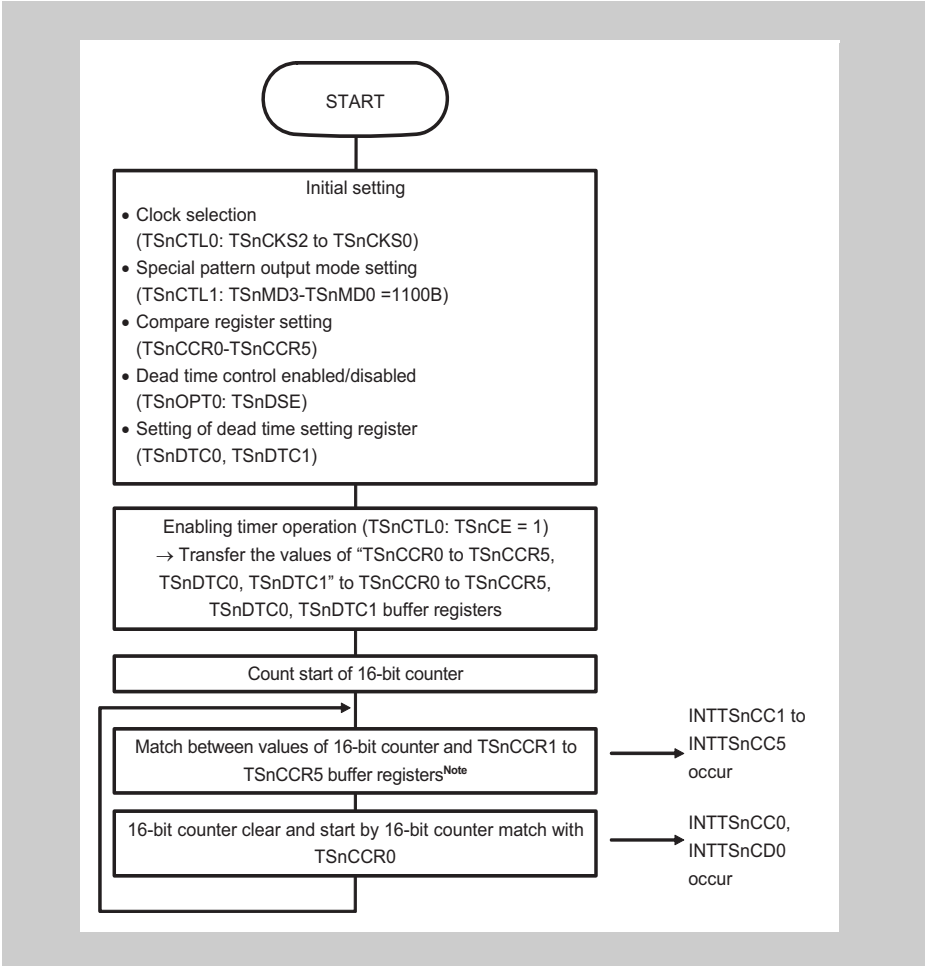


Figure 17-183 Basic operation flow in special pattern output mode

Note The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

(2) Special pattern output mode operation list

(a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle/compare value
TSnCCR1 to TSnCCR3	Reload	Possible	Compare value
TSnCCR4, TSnCCR5	Reload	Possible	Compare value (Selectable as A/D conversion trigger)
TSnDTC0, TSnDTC1	Reload	Conditionally possible[Note]	Compare value
TSnPAT0, TSnPAT1	Reload	Possible	Special pattern setting

Note Refer to *"Rewriting of TSnDTC0 and TSnDTC1 registers" on page 901* for details.

(b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	-

(c) Output pin

Pin	Function
TOSn0	Pattern output from TSnPAT0[3:0]
TOSn1	Pattern output from TSnPAT0[7:4]
TOSn2	Pattern output from TSnPAT0[11:8]
TOSn3	Pattern output from TSnPAT0[15:12]
TOSn4	Pattern output from TSnPAT1[3:0]
TOSn5	Pattern output from TSnPAT1[7:4]
TOSn6	Pattern output from TSnPAT1[11:8]
TOSn7	Pattern output from TSnPAT1[15:12]

(d) Interrupt

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing as INTTSnCC0 register)
INTTSnWN	-

(e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from TSnCCR1 to TSnCCR5 to TSnCCR1 to TSnCCR5 +1

Note "–" indicates an unused function in the special pattern mode.

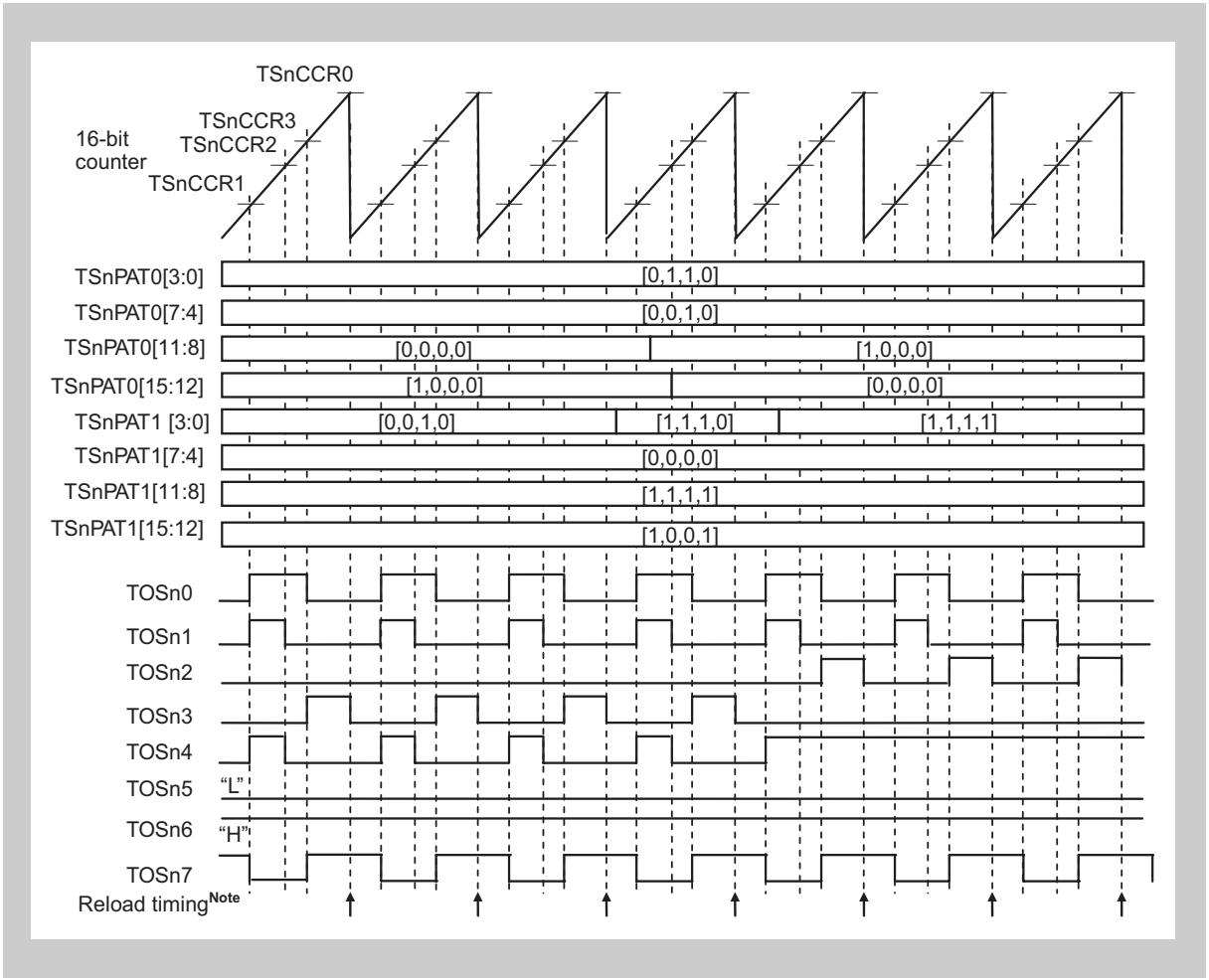


Figure 17-184 Special pattern output mode operation example

Note Write access to the TSnCCR1 register is necessary.

(3) Special pattern output mode setting

(a) Mode setting

The special pattern output mode is set by setting TSnMD4 to TSnMD0 of TSnCTL1 register to [1, 1, 0, 0].

(b) Output level/output enable setting

Output level/output enabling is performed by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register. Pattern is output from the TOSn0 to TOSn7 pins.

(c) Error interrupt enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC0 bit of the TSnIOC4 register to 1 when the positive/negative phase simultaneous active state is detected.

(d) Interrupt and thinning out function setting

Peak interrupt (INTTSnCD0) occurs by match between the TSnCCR0 register and 16-bit counter. (Trough interrupt does not occur). Set TSnICE of TSnOPT1 register to 1 if the peak interrupt is necessary. To use the thinning out function for peak interrupt, execute setting with the TSnID4 to TSnID0 bits of the TSnOPT1 register.

(e) Reload thinning out function setting

Set reload timing to the same timing as interrupt timing by setting the TSnRDE bit of the TSnOPT1 register to 1. Reload timing is generated by setting TSnRTE of TSnOPT1 register to 1.

Caution Write access to the TSnCCR1 register is required for reload execution.

(f) A/D conversion trigger output setting

To set A/D conversion trigger 0 (TSnADTRG0 signal), set enabling/disabling of match timing with the TSnCCR5 register, match timing with the TSnCCR4 register, and peak interrupt (INTTSnCD0) by using TSnAT04, TSnAT02, TSnAT01 bits of the TSnOPT2 register.
To set A/D conversion trigger 1 (TSnADTRG1 signal), set enabling/disabling of match timing with the TSnCCR5 register, match timing with the TSnCCR4 register, and peak interrupt (INTTSnCD0) by using the TSnAT14, TSnAT12, TSnAT11 bits of the TSnOPT3 register. Set the compare value of the TSnCCR4 and TSnCCR5 registers.
The thinning out function is enabled for the TSnADTRG0 and TSnADTRG1 signals. No thinning out, 1-thinning out,
3-thinning out and 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

Caution

1. When using the TOSn7 pin, set the TSnOPT2 and TSnOPT3 registers and TSnCCR4 and TSnCCR5 registers appropriately according to the system.
2. Trough interrupts (INTTSnOD) do not occur in special pattern output mode. Be sure to set TSnAT00, 10 bits of the TSnOPT2 andTSnOPT3 registers to 0.
3. Do not use the peaks/troughs of the 16-bit sub-counter in special pattern output mode. Be sure to set TSnAT07, TSnAT06, TSnAT17, and TSnAT16 bits of the TSnOPT2 and TSnOPT3 registers to 0.
4. Count-down status of the 16-bit counter does not occur in special pattern output mode. Be sure to set TSnAT05, TSnAT03, TSnAT15, TSnAT13 bits of the TSnOPT2 and TSnOPT3 registers to 0.

(g) Output pattern setting

In the output pattern, cycle setting is possible with the TSnCCR0 register, and duty setting with the TSnCCR1 to TSnCCR3 registers.
Timer output level after compare match (INTTSnCC0 to INTTSnCC3 occurrence) can be set with the TSnPAT0 and TSnPAT1 registers. In the special pattern output mode, the output pattern is controlled as follows.

TSnPAT0[3:0]:	TOSn0 pin output pattern setting
TSnPAT0[7:4]:	TOSn1 pin output pattern setting
TSnPAT0[11:8]:	TOSn2 pin output pattern setting
TSnPAT0[15:12]:	TOSn3 pin output pattern setting
TSnPAT1[3:0]:	TOSn4 pin output pattern setting
TSnPAT1[7:4]:	TOSn5 pin output pattern setting
TSnPAT1[11:8]:	TOSn6 pin output pattern setting
TSnPAT1[15:12]:	TOSn7 pin output pattern setting

(h) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.
Dead time can be calculated with the following equation.
16-bit counter operation clock cycle × TSnDTC0
16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of TOSn2, TOSn4, TOSn6 pins to the active state of the TOSn1, TOSn3, TOSn5 pins can be set with the TSnDTC0 register.
The time for the change from the inactive state of TOSn1, TOSn3, TOSn5 pins to the active state of the TOSn2, TOSn4, TOSn6 pins can be set with the TSnDTC1 register.

Caution TOSn1, TOSn3 and TOSn5 rise by priority in the setting of simultaneous rise of “TOSn1 and TOSn2”, “TOSn3 and TOSn4”, and “TOSn5 and TOSn6”.

(4) Special pattern output mode operation

Figure 17-185 on page 1005 and *Figure 17-186 on page 1006* show an operation example in the special pattern output mode. The TSnPAT0 and TSnPAT1 register values are rewritten by using the reload mode (batch rewrite function) in *Figure 17-185 on page 1005* and *Figure 17-186 on page 1006*. The output level is output as set in the TSnPAT0 and TSnPAT1 registers at match timing between the values of the TSnCCR0 to TSnCCR3 registers and the 16-bit counter.
Waveforms as expected are output because the batch rewrite is executed.
In the special pattern output mode, error interrupt (INTTSnER) occurs by setting TSnEOC of the TSnIOC4 register to 1 to detect the positive/negative simultaneous active state.
To generate a reload timing, write operation is necessary for the TSnCCR1 register. Therefore, write the same value to the TSnCCR1 register at last when rewriting the TSnPAT0 and TSnPAT1 registers.

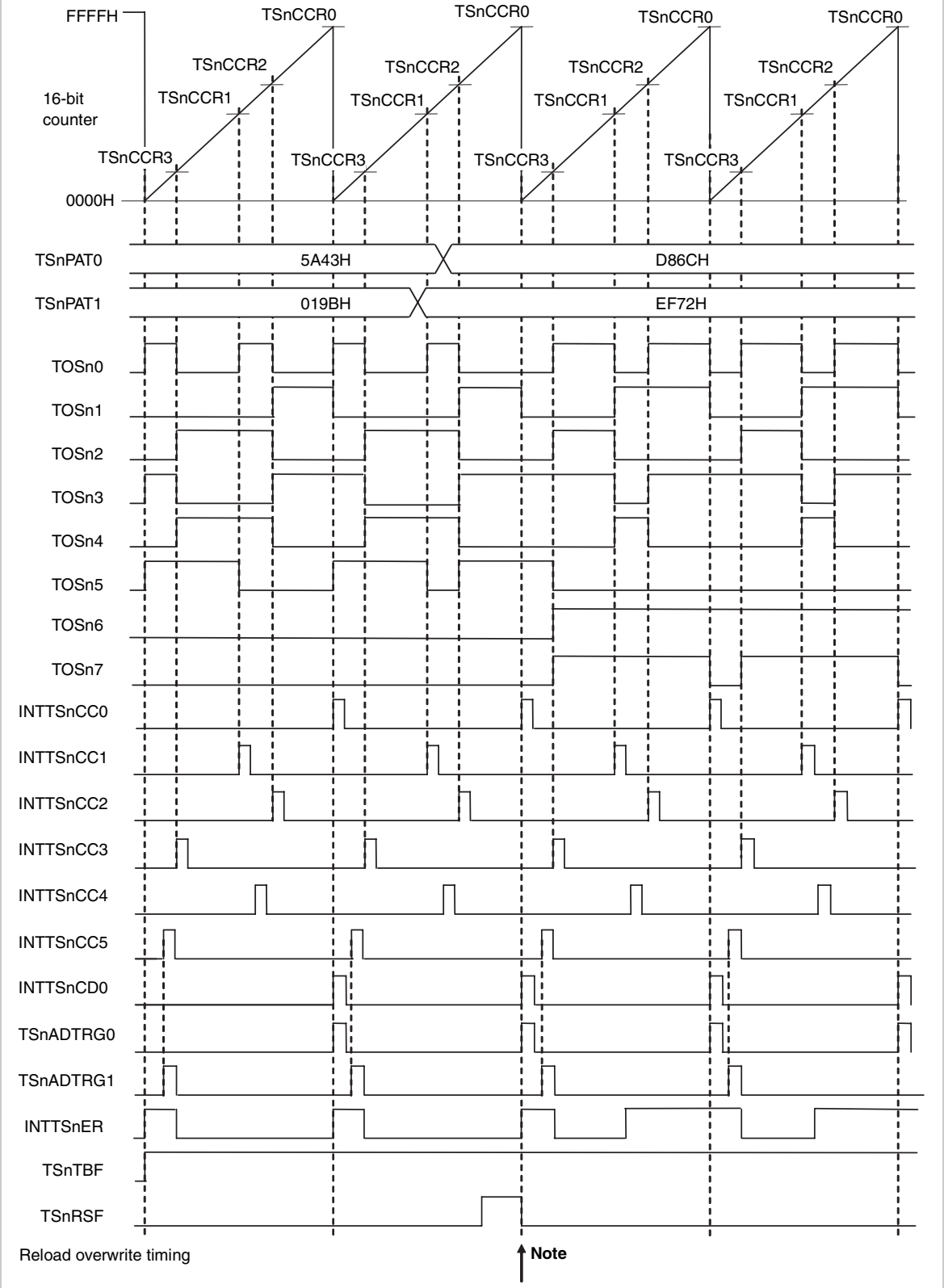


Figure 17-185 Special pattern output mode operation example (without dead time)

Note Write access to the TSnCCR1 register is required.

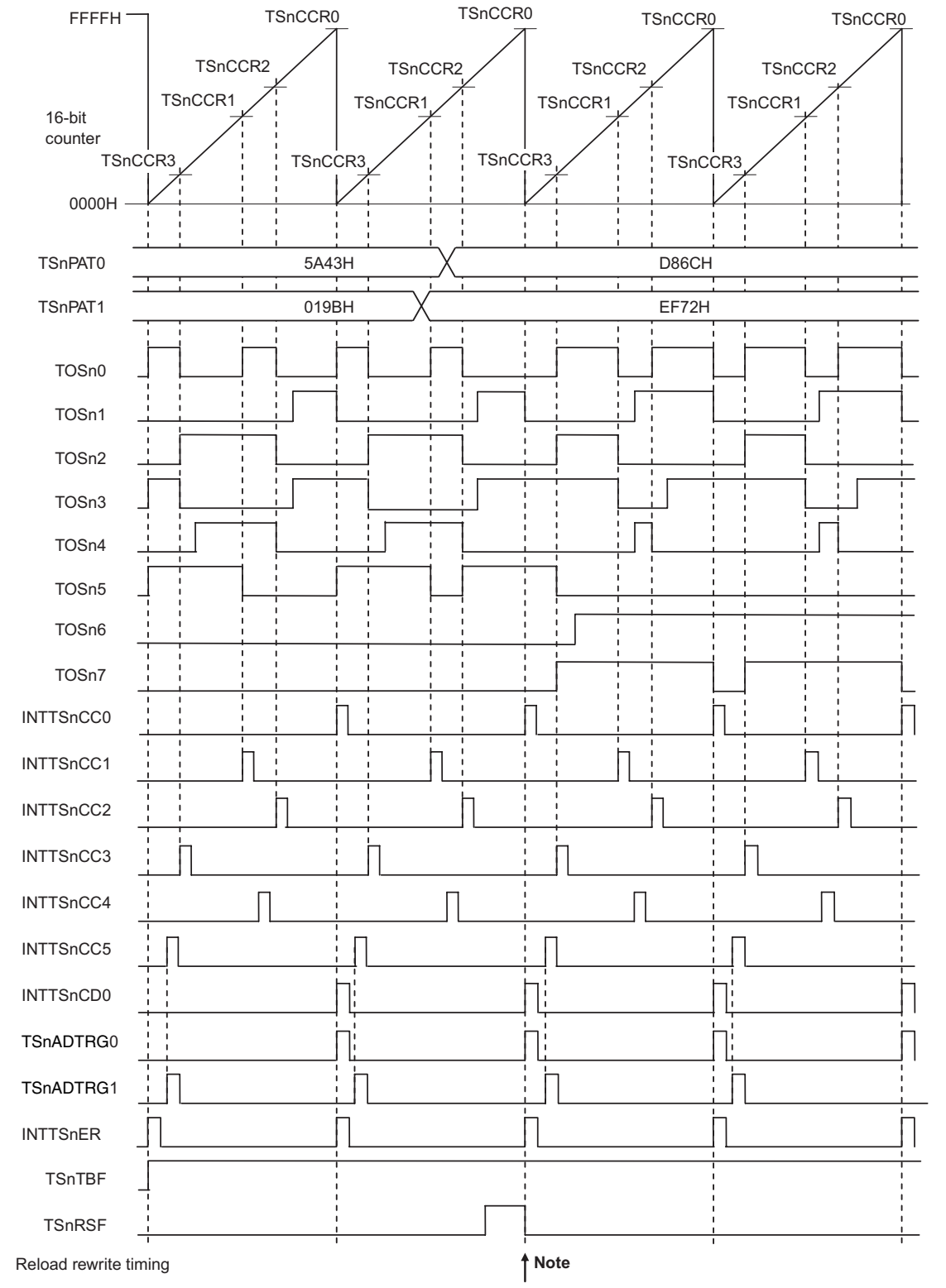


Figure 17-186 Special pattern output mode operation example (with dead time)

Note Write access to the TSnCCR1 register is required.

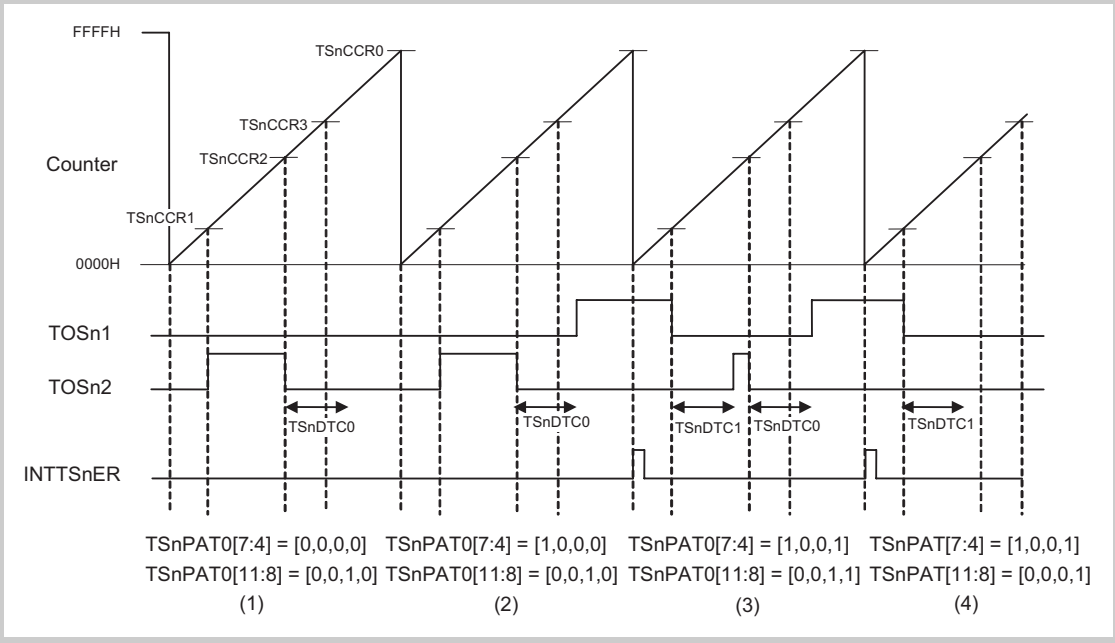
(5) Dead time control in special pattern output mode

In the special pattern output mode, dead time is added by the dead time control operation at the fall of each phase when TSnDSE of the TSnOPT0 register is set to 1.
The dead time that is set to the TSnDTC1 register is inserted at the fall of the positive phase, and the dead time that is set to the TSnDTC0 register is inserted at the fall of the negative phase.

Caution Dead time control of the TOSn0 and TOS7 pins is impossible.

[Control of TOSn1, TOSn2 pins]

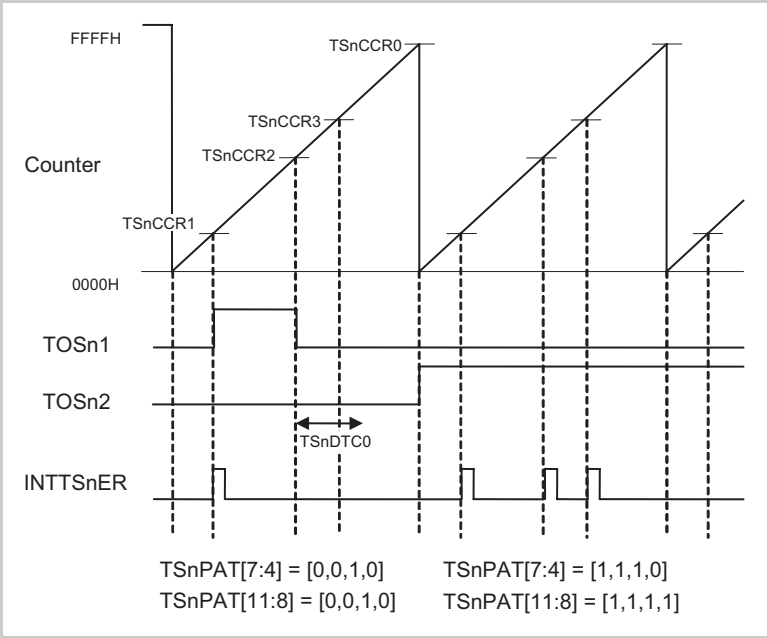
1. Dead time from the fall of the TOSn1 pin to the rise of the TOSn2 pin is assured.
2. The TOSn2 pin holds the inactive level when the TOSn1 pin is active level, and the TOSn1 pin holds the inactive level when the TOSn2 pin is active level.
3. Set the TOSn1 pin by priority if the TOSn1 and TOSn2 pins rise simultaneously.
4. The TOSn1 pin holds inactive level even if the TOSn1 pin setting condition is generated during the TOSn2 pin active level continued.



In cycle (1), dead time counter starts operation (with TSnDTC0 register value) from the fall of the TOSn2 pin, and controls the TOSn1 pin setting condition. In this example, there is nothing to occur because of no setting condition occurrence in the dead time period.

In cycle (2), dead time counter operates from the fall of the TOSn2 pin (with the TSnDTC0 register value). In this case, the TOSn1 pin setting condition is generated by the match with the TSnCCR3 register. However, the dead time counter is under operation, so the TOSn1 pin is set after the count completion of the dead time counter.

In cycle (3), setting conditions of the TOSn1 pin and TOSn2 pin are generated simultaneously at the start of the cycle. In this case, the TOSn1 pin is set by priority, the setting condition of TOSn2 pin is retained and an error interrupt (INTTSnER) occurs. The TOSn1 pin is cleared and the TOSn2 pin is set by the match between the TSnCCR1 register and counter. The TOSn2 pin is set after the dead time counter stops because dead time control is operated.
In cycle (4), the setting conditions of the TOSn1 pin and TOSn2 pin are generated simultaneously at the start of the cycle. In this case, the TOSn1 pin is set by priority, the setting condition of the TOSn2 pin is retained and an error interrupt (INTTSnER) occurs.
Clear conditions of the TOSn1 and TOSn2 pins are generated by match with the TSnCCR1 register, and then the TOSn1 pin is cleared. This time, the retained setting condition of TOSn2 pin is cleared.



When the condition in which TOSn1 and TOSn2 pins rise simultaneously, the TOSn1 pin becomes active. However, the TOSn1 pin retains the inactive level in case of the condition occurrence in which the TOSn1 and TOSn2 pins rise simultaneously during the active level output from the TOSn2 pin. Also, an error interrupt (INTTSnER) occurs at the same time.

[Control of TOSn3 and TOSn4 pins]

The control logic is same as the relation between TOSn1 and TOSn2. When the setting conditions for the TOSn3 and TOSn4 pins occur simultaneously, the TOSn3 pin is set by priority.

[Control of TOSn5 and TOSn6 pins]

The control logic is the same as the relation between TOSn1 and TOSn2. When the setting conditions for the TOSn5 and TOSn6 pins occur simultaneously, the TOSn5 pin is set by priority.

(6) Error interrupt in special pattern output mode (INTTSnER)

In the special pattern output mode, the positive/negative phase simultaneous active state detection function can be used by setting the TSnTBA2 to TSnTBA0 bits of TSnIOC4 register. There are the TSnTBF bit of TSnOPT6 register, or INTTSnER for detection method. The INTTTSnER operation differs according to the setting of the dead time control enabling/disabling (the TSnDSE bit of TSnOPT0 register).

Table 17-24 Detection Method List

TSnDSE of TSnOPT0 register	TSnTBF bit of TSnOPT6 register	INTTSnER
0 (Dead time control enabling)	Set with the timer output simultaneous active state	Error interrupt is output while active output is simultaneously provided.
1 (Dead time control disabling)	Set with the timer output simultaneous active state	Error interrupt is output at the timing when timer output simultaneous setting condition occurs.

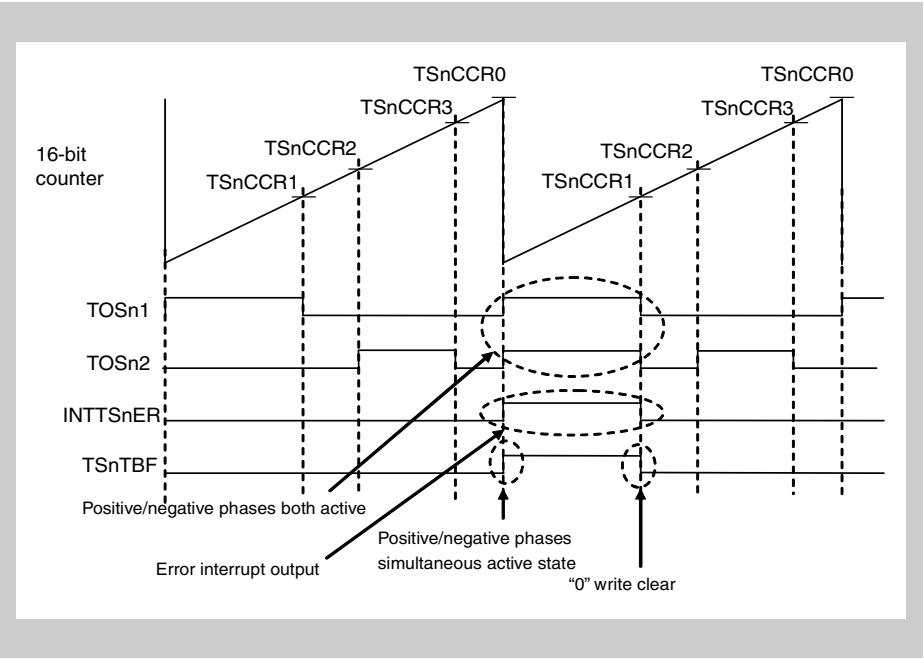


Figure 17-187 Error detection example (TSnDSE = 0) (1/2)

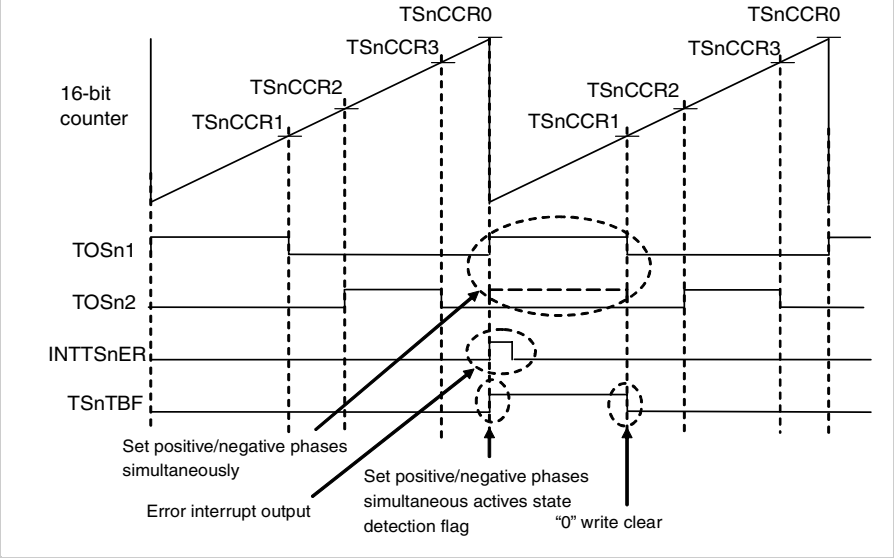


Figure 17-188 Error detection example (TSnDSE = 1) (2/2)

(a) Caution about error interrupt during dead time control

When the TSnDSE bit of the TSnOPT0 register is set to 1, or the TSnTBA0, TSnTBA1 and TSnTBA2 bits of the TSnIOC4 register are set to 1, positive/negative phase simultaneous setting condition occurrence of the TOSn1, TOSn3 and TOSn5 pins and the TOSn2, TOSn4, and TOSn6 pins is the error interrupt occurrence condition. Therefore, error interrupt occurs by occurrence of positive/negative phase simultaneous setting condition even when TSnOE1, TSnOE2 of the TSnIOC0 register is set to 0 and the TOSn1, TOSn2 pins are fixed in inactive level. The positive/negative phase simultaneous active state detection flag is also set. Set TSnTBA0, TSnTBA1 and TSnTBA2 to 0 in order to prevent the above situation.

(b) Error interrupt detail timing

Figure 17-189 on page 1011 and *Figure 17-190 on page 1011* show the INTTSnER trigger output timing. In case of TSnDSE = 1 of the TSnOPT0 register, 1 clock trigger is output to INTTSnER with the following operation clock when the setting condition of timer output 2 occurs. TSnTBF = 1 of the TSnOPT6 register is set simultaneously.

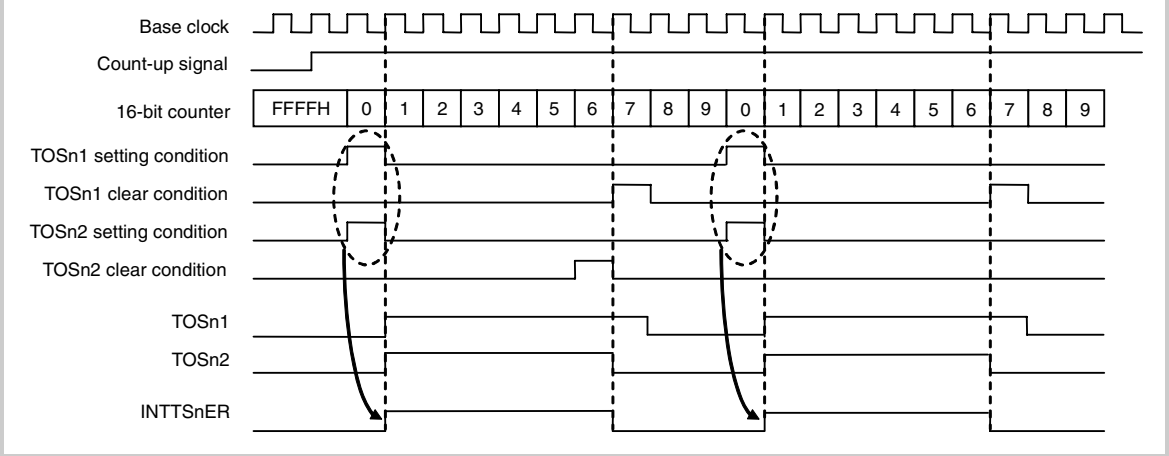


Figure 17-189 Error interrupt timing (TSnDSE = 0)

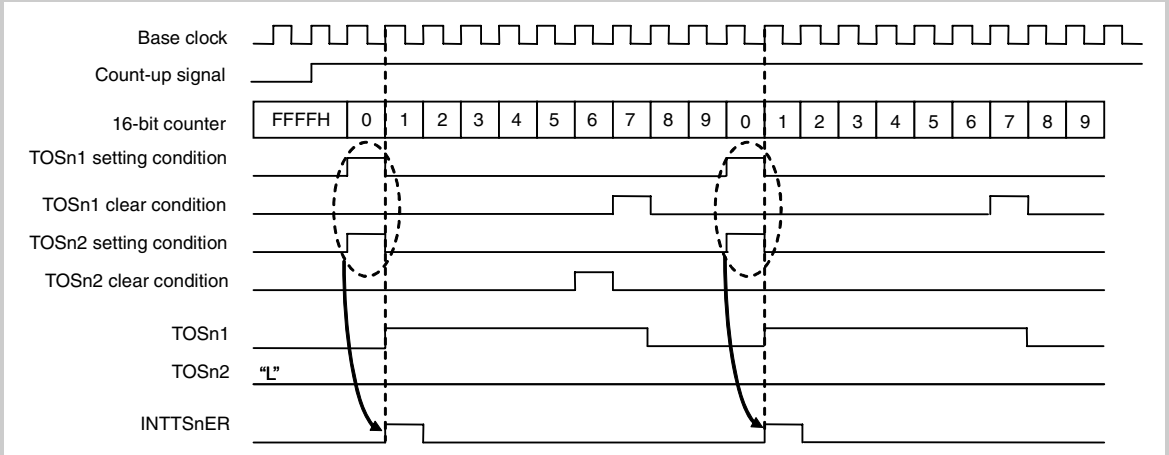


Figure 17-190 Error interrupt timing (TSnDSE = 1)

Note TOSn3 and TOSn4 and TOSn5 and TOSn6 are the same as well.

(7) Compare register rewriting operation in special pattern output mode

The compare register (TSnCCR0 to TSnCCR3) can be changed in the special pattern output mode. The change timing is the reload timing (the reload thinning out function is also enabled), so that the new setting is valid from the next reload timing after the TSnCCR1 register change.

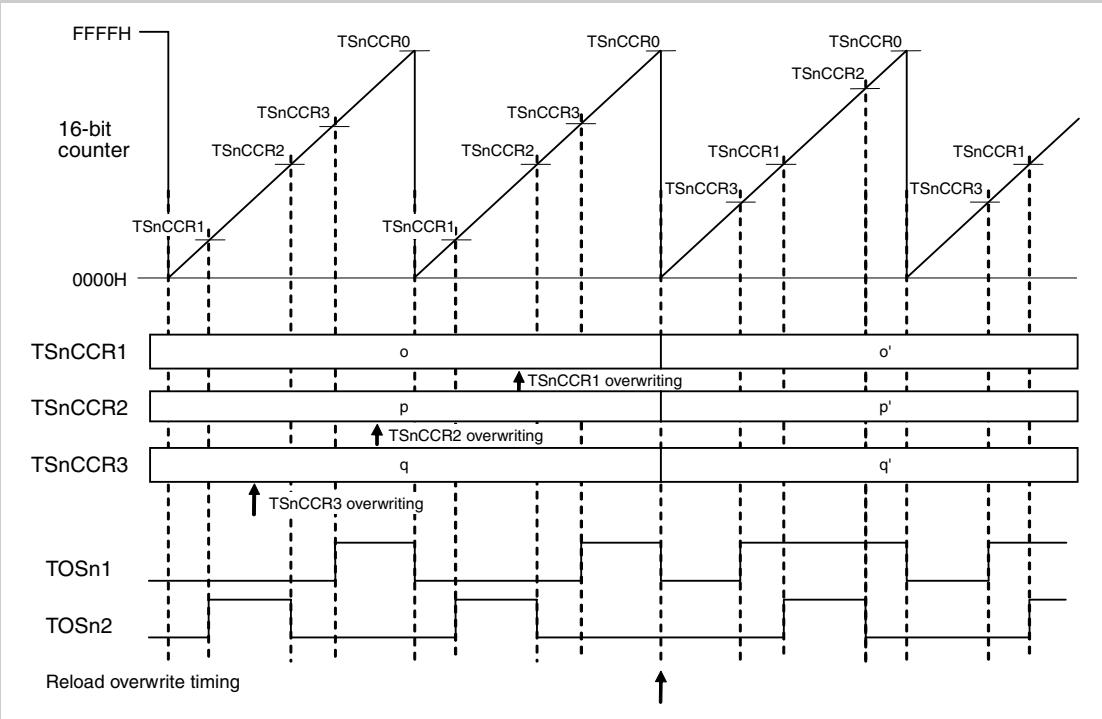


Figure 17-191 Compare register rewriting operation in special pattern output mode

Note TSnPAT0[7:4] = [1, 0, 0, 0]
TSnPAT0[11:8] = [0, 0, 1, 0]

(8) Pattern register rewriting operation in special pattern output mode

In the special pattern output mode, pattern setting registers (TSnPAT0, TSnPAT1) can be rewritten dynamically. The change valid timing is the reload timing (the reload thinning out function is also enabled), so that writing to the TSnCCR1 register is required in order to generate reload timing.

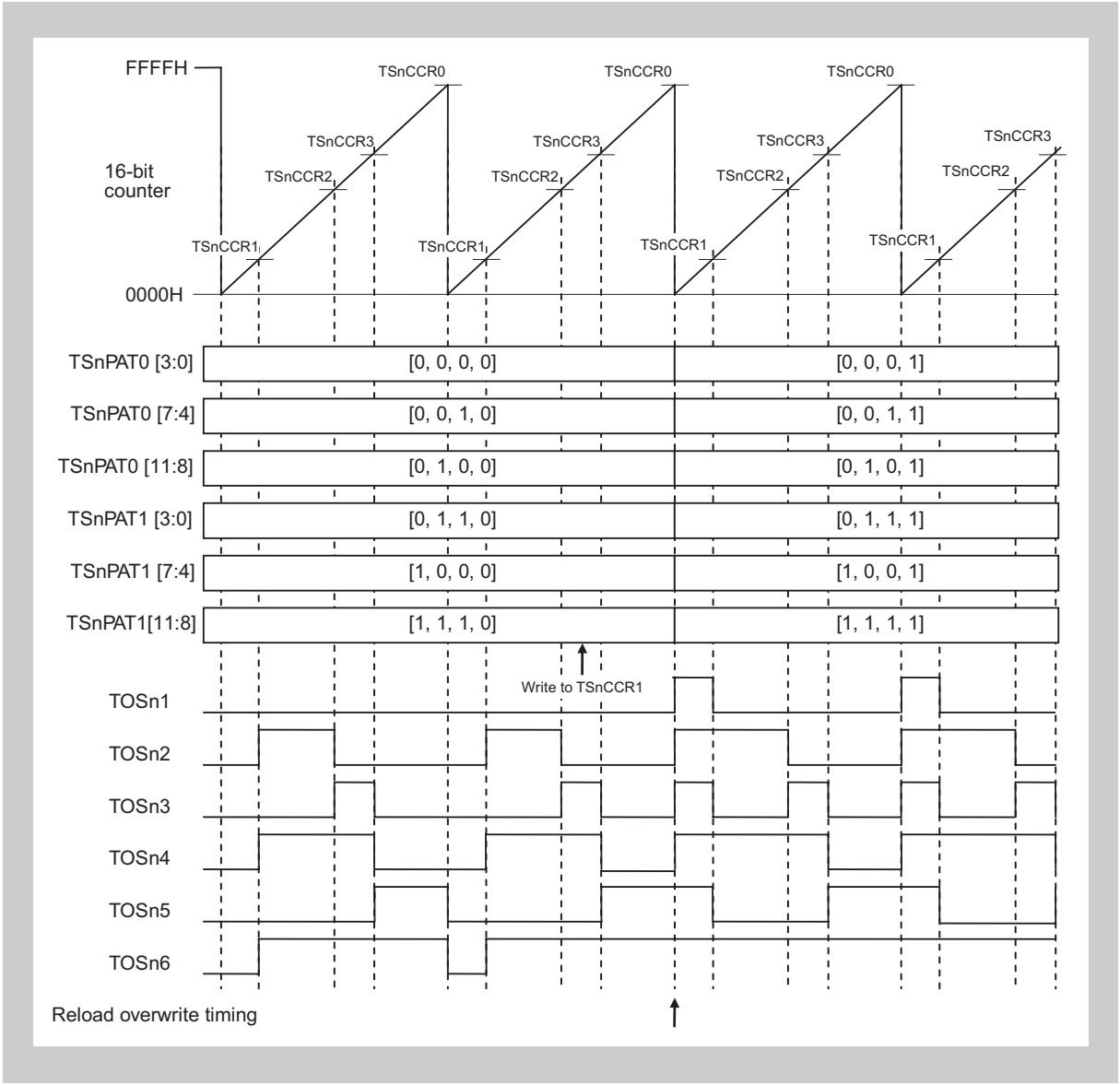


Figure 17-192 Pattern register rewriting operation in special pattern output mode

17.10.13 Software output function

The software output control function can be used in the high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode and special 120° excitation mode. The function enables the TOSn1 to TOSn6 pins to switch 6 patterns of output with software by using the TSnSOC bit of the TSnOPT4 register, TSnIPC2 to TSnIPC0 bits, and the TSnIDC bit of the TSnOPT7 register.
The TOSn1 to TOSn6 pins are immediately switched to the software output control by switching from TSnSOC = 0 to TSnSOC = 1.
Conversely, software output control is released at the reload timing if switched from TSnSOC = 1 to TSnSOC = 0.

Table 17-25 Register Description of Software Output Control Function

Register	Operation
TSnSOC	TSnSOC = 1
TSnIDC	Set the output pattern
TSnPOT, TSnPSS, TSnPSC	Setting invalid
TSnIPC0 to TSnIPC2	Set the output pattern shown in *Table 17-26 on page 1014* and *Table 17-27 on page 1014*

Table 17-26 Software output control output pattern (TSnIDC of TSnOPT7 register = 0)

TSnOPT4 register: TSnSOC = 1, TSnSOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 Register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
2. Output remains the level before the change.
3. ACT:Active level output
4. INACT:Inactive level output

Table 17-27 Software output control output pattern (TSnIDC = 1 of TSnOPT7 register)

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

Note 1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.

2. Output remains the level before the change.

3. ACT:Active level output

4. INACT:Inactive level output

17.10.14 180° excitation control function

The 180° excitation control function is enabled in the high-accuracy T-PWM mode, 120° excitation mode and special 120° excitation mode.
The function provides 6 patterns of output to the TOSn1 to TOSn6 pins by using the TSnADC bit of the TSnOPT5 register, the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, the TSnIDC bit of the TSnOPT7 register and external trigger (TSnSTCI0, TSnSTCI1 signals or TAPTSn2 to TAPTSn0 pins).
The TOSn1 to TOSn6 pins are immediately switched to 180° excitation control by switch from “TSnADC = 0” to “TSnADC = 1”.
Conversely, 180° excitation control is released at the reload timing if switched from TSnADC = 1 to TSnADC = 0.

Table 17-28 Register description of 180° excitation control function

Register	Operation
TSnSOC of TSnOPT4 register	TSnSOC = 0
TSnIPC2 to TSnIPC0 of TSnOPT4 register	Set the output pattern shown in the following *Table 17-29 on page 1016* and *Table 17-30 on page 1017*
TSnPSC of TSnOPT4 register	Setting invalid
TSnPSS, TSnTSF of TSnOPT5 register	
TSnPOT of TSnOPT5 register	Selection of pattern switch method or trigger switch method
TSnADC of TSnOPT5 register	180° excitation control function is enabled with TSnADC = 1.
TSnIDC of TSnOPT7 register	Set the output pattern (rotation direction).
TSnARD pin	Setting invalid

Table 17-29 Software output control output pattern (TSnIDC = 0 of TSnOPT7 register)

TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
2. Output remains the level before the change.
3. Active level output
4. INACT:Inactive level output

Table 17-30 Software Output Control Output Pattern (TSnIDC = 1 of TSnOPT7 Register)

TSnOPT4 register: TSnSOC = 0,
TSnOPT5 register: TSnADC = 1, TSnPOT = 1,
TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

Note
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
2. Output remains the level before the change.
3. ACT:Active level output
4. INACT:Inactive level output

17.11 Timer Synchronization

The Timer S synchronisation function is a synchron-start function.

The TMS1 will be the master and the TMS0 the slave.

To use this function the following procedure has to be used:

- TMS0 is stopped.
- Set the TS0CTL1.TS0SYE = 1
- Set the TSnCE-bit of TMS0 = 1
- TMS0 is waiting for start of TMS1. When TMS1 will be started, TMS0 will be started, too.

Both timers can be stopped separately, without influence to each other.

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

This microcontroller has two instances of the Timer T (TMT), TMT0 to TMT1.

Note Throughout this chapter, the individual instances of Timer T are identified by “n” (n = 0 or 1), for example TTnCTL1 for the Timer Tn control register 1.

18.1 Features

Timer T (TMT) is a 16-bit timer/event counter that provides general-purpose functions.

Timer T can perform the following operations:

- Interval timer function
- External event count function
- One-shot pulse output function
- External trigger pulse function
- 16-bit accuracy PWM output function
- Free-running function
- Pulse width measurement function
- 2-phase encoder function
- Triangular wave PWM output function
- Offset trigger generation function

18.2 Function Outline

- Capture trigger input signal × 2
- Encoder input signal × 2
- Encoder clear signal × 1
- External trigger input signal × 1
- External event input × 1
- Readable counter × 1
- Count write buffer × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer Output (TO) × 2
- Capture/compare match signal × 2
- Overflow interrupt × 1
- Encoder clear interrupt × 1

18.3 Configuration

Timer T is configured of the following hardware.

Table 18-1 Timer T configuration

Item	Configuration
Counter	16-bit counter
Registers	TMTn capture/compare registers 0, 1 (TTnCCR0, TTnCCR1) TMTn counter read buffer register (TTnCNT) TMTn counter write buffer register (TTnTCW) TMTn capture/compare buffer registers 0, 1
Timer input pins	7 (TITn0, TITn1, TEVTTn, TTRGTn, TENCTn0, TENCTn1, TECRTn)
Timer output pins	2 (TOTn0, TOTn1)
Timer input signals	TTTTI0, TTTTI1, TTTENC0, TTTENC1, TTTTRG, TTTEVT, TTTECR
Timer output signals	TTTTO0, TTTTO1, TTTICC0, TTTICC1, TTnEQC0, TTnEQC1
Control registers	TMTn control registers 0, 1 (TTnCTL0 to TTnCTL2) TMTn I/O control registers 0 to 3 (TTnIOC0 to TTnIOC3) TMTn option registers 0 to 2 (TTnOPT0 to TTnOPT2)
Interrupts	Compare match interrupt (INTTTnCC0, INTTTnCC1) Overflow interrupt (INTTTnOV) Encoder clear interrupt (INTTTnEC)

Table 18-2 List of Timer T registers (1/2)

Address	Register name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
FFFFF690H	TMT0 control register 0	TT0CTL0	R/W	×	×		00H
FFFFF691H	TMT0 control register 1	TT0CTL1	R/W	×	×		00H
FFFFF692H	TMT0 control register 2	TT0CTL2	R/W	×	×		00H
FFFFF693H	TMT0 I/O control register 0	TT0IOC0	R/W	×	×		00H
FFFFF694H	TMT0 I/O control register 1	TT0IOC1	R/W	×	×		00H
FFFFF695H	TMT0 I/O control register 2	TT0IOC2	R/W	×	×		00H
FFFFF696H	TMT0 I/O control register 3	TT0IOC3	R/W	×	×		00H
FFFFF697H	TMT0 option register 0	TT0OPT0	R/W	×	×		00H
FFFFF698H	TMT0 option register 1	TT0OPT1	R/W	×	×		00H
FFFFF699H	TMT0 option register 2	TT0OPT2	R/W	×	×		00H
FFFFF69AH	TMT0 capture/compare register 0	TT0CCR0	R/W			×	0000H
FFFFF69CH	TMT0 capture/compare register 1	TT0CCR1	R/W			×	0000H
FFFFF69EH	TMT0 counter read buffer register	TT0CNT	R			×	0000H[Note]
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
FFFFF6A4H	TMT1 I/O control register 1	TT1IOC1	R/W	×	×		00H
FFFFF6A5H	TMT1 I/O control register 2	TT1IOC2	R/W	×	×		00H

Table 18-2 List of Timer T registers (2/2)

Address	Register name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
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[Note]
FFFFF9A0H	TMT1 counter write buffer register	TT1TCW	R/W			×	0000H

Note When TTnCE = 0

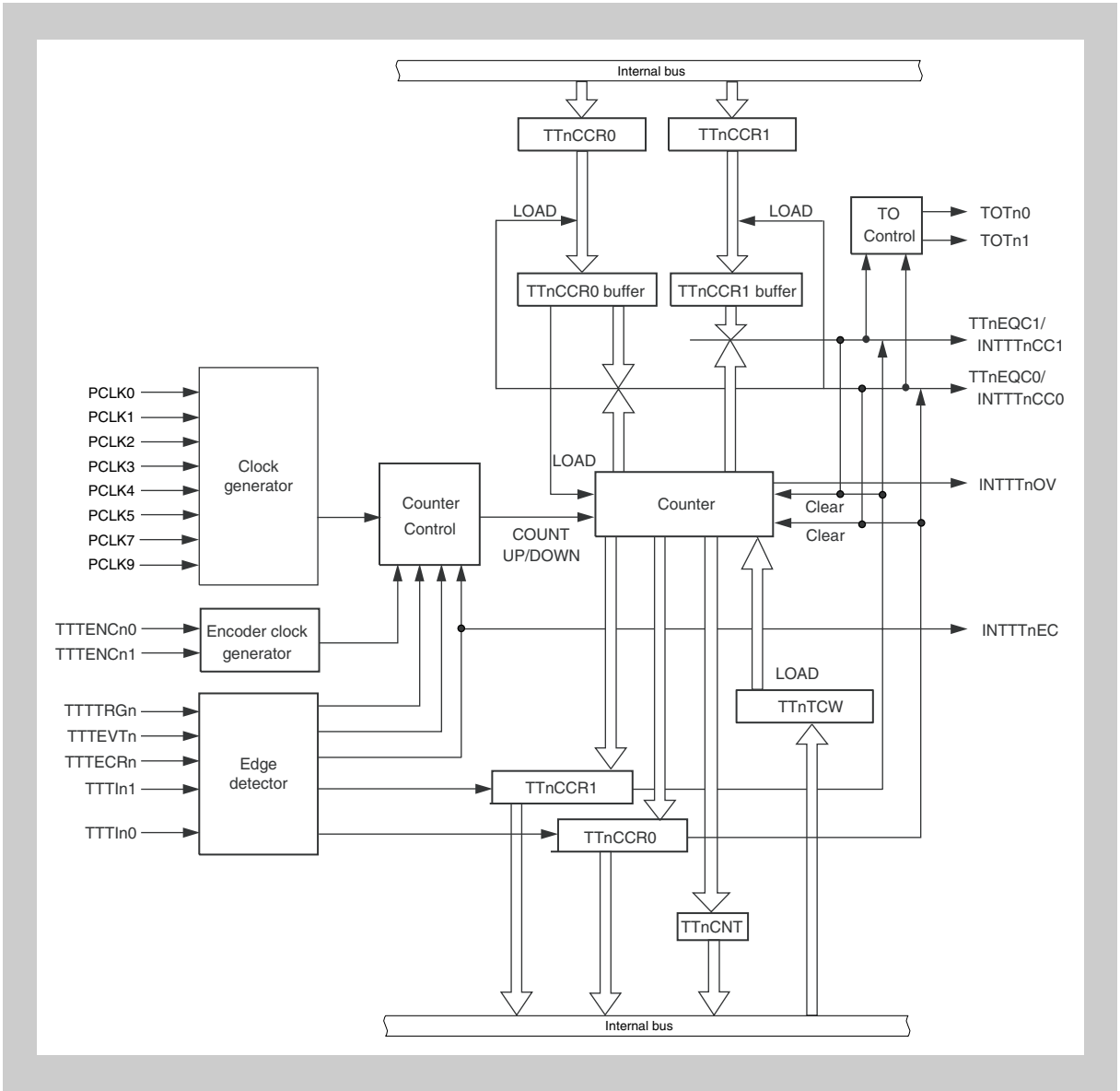


Figure 18-1 Block diagram of Timer T

Note 1. m = 0, 1

2. PCLKn: peripheral clocks (refer to *"Clock Generator" on page 179*):

PCLK	0	1	2	3	4	5	7	9
f [MHz]	32	16	8	4	2	1	0.25	0.0625

(1) TTnCCR0 - TMTn capture/compare register 0

The TTnCCR0 register is a 16-bit register that functions both as a capture register and as a compare register.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: TT0CCR0 FFFFF69AH, TT1CCR0 FFFFF6AAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR0																

The capture and compare functions are as follows in each mode.

Table 18-3 Capture/compare functions in each mode

Operation mode	Capture/compare setting of TTnCCR0 register	Rewriting method during compare	Counter clear function
Interval mode	Compare only	Anytime write	Compare match
External event count mode	Compare only	Anytime write	Compare match
External trigger pulse output mode	Compare only	Batch write (Reload)	Compare match
One-shot pulse mode	Compare only	Anytime write	Compare match
PWM mode	Compare only	Batch write (Reload)	Compare match
Free-running mode	Capture/compare selectable	Anytime write	-
Pulse width measurement mode	Capture only	-	External input (TITn0 pin)
Triangular wave PWM mode	Compare only	Batch write (Reload)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 TTnECM0 and TTnECM1 bits of the TTnCTL2 register.

- Use as compare register
  When TTnCE = 1, the TTnCCR0 register rewrite method differs according to the operation mode. Refer to *Table 18-3 on page 1023*.
  (For details about the compare register rewrite operation, refer to *"Method for writing to compare register" on page 1046*.)
- Use as capture register
  The counter value is saved to the TTnCCR0 register upon TITn0 pin input edge detection. The function to clear counters following capture differs according to the operation mode. Refer to *Table 18-3 on page 1023*.

(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.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: TT0CCR1 FFFFF69CH, TT1CCR1 FFFFF6ACH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR1																

The capture/compare functions in each operation mode are as follows.

Table 18-4 Capture/compare functions in each mode

Operation mode	Capture/compare setting of TTnCCR1 register	Rewriting method during compare	Counter clear function
Interval mode	Compare only	Anytime write	-
External event count mode	Compare only	Anytime write	-
External trigger pulse output mode	Compare only	Batch write (Reload)	-
One-shot pulse mode	Compare only	Anytime write	-
PWM mode	Compare only	Batch write (Reload)	-
Free-running mode	Capture/compare selectable	Anytime write	-
Pulse width measurement mode	Capture only	-	External input (TITn1 pin)
Triangular wave PWM mode	Compare only	Batch write (Reload)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 bits TTnECM0 and TTnECM1 of TTnCTL2 register.

3. The batch write reload timing is the counter's 0000H clear timing only.

- Use as compare register
  When TTnCE = 1, the write method of register TTnCCR1 differs according to the operation mode. Refer to *Table 18-4 on page 1024*.
  (For details about the compare register rewrite operation, refer to *"Method for writing to compare register" on page 1046*.)

- Use as capture register
  The counter value upon TITn1 pin input edge detection is saved to the TTnCCR1 register. The function to clear the counter following capture also differs according to the mode. Refer to *Table 18-4 on page 1024*.

(3) TTnCNT - TMTn counter read buffer register

The TTnCNT register is a read buffer register that can read the counter value.

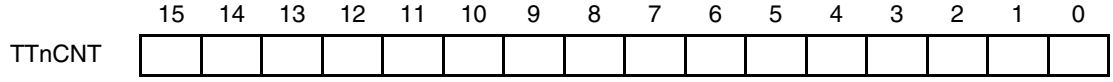
This register can be read in 16-bit units only.

Reset input clears this register to 0000H.

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, 0000H can be read.
- When bit TTnECC = 1, the value held when bit TTnCE was cleared to “0” can be read.

After reset: 0000H R Address: TT0CNT FFFFF69EH, TT1CNT FFFFF6AEH



(4) TTnTCW - TMTn counter write buffer register

The TTnTCW register is a write buffer register that can write the counter value.

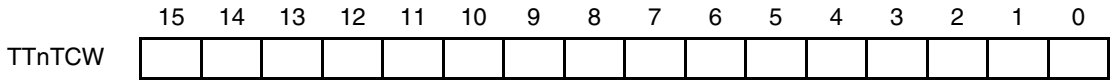
The setting value is valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. (In all other modes, the setting value is invalid.)

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

Reset input clears this register to 0000H.

Note When TTnECC of register TTnCTL2 = 0, the setting value is loaded to the counter when the TTnCE bit is set (to 1). (When TTnECC = 1, the counter holds its value, so it is not reloaded.)

After reset: 0000H R/W Address: TT0TCW FFFFF990H, TT1TCW FFFFF9A0H



18.4 Control Registers

(1) TTnCTL0 - TMTn control register 0

TTnCTL0 is an 8-bit register that controls the operation of TMTn.

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

Reset input clears this register to 00H.

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.

After reset: 00H R/W Address: TT0CTL0 FFFFF690H, TT1CTL0 FFFFF6A0H

	7	6	5	4	3	2	1	0
TTnCTL0 (n = 0, 1)	TTnCE	0	0	0	0	TTnCKS2	TTnCKS1	TTnCKS0

TTnCE	TMTn operation control
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 TMTn Count Clock and Time Until Count-Up.

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

TTnCKS2	TTnCKS1	TTnCKS0	Internal count clock selection
0	0	0	PCLK0
0	0	1	PCLK1
0	1	0	PCLK2
0	1	1	PCLK3
1	0	0	PCLK4
1	0	1	PCLK5
1	1	0	PCLK7
1	1	1	PCLK9

Table 18-5 TMTn Count Clock and Count Delay

Count clocks	TTnCKS2	TTnCKS1	TTnCKS0	Count delay	
				Minimum	Maximum
PCLK0	0	0	0	3 base clocks	4 base clocks
PCLK1	0	0	1		
PCLK2	0	1	0		
PCLK3	0	1	1	4 base clocks	5 base clocks + 1 count clock
PCLK4	1	0	0		
PCLK5	1	0	1		
PCLK7	1	1	0		
PCLK9	1	1	1		

Note f_{TMTn}: Base clock of TMTn ($f_{TMTn} = f_{PCLK0}$)

(2) TTnCTL1 - TMTn control register 1

The TTnCTL1 register is an 8-bit register that controls the operation of TMTn.

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

Reset input clears this register to 00H.

Set the TTnCTL1 register when TTnCE = 0. When TTnCE = 1, the bits other than bit TTnEST (TTnEEE, TTnMD3 to TTnMD0, TTnSYE) can be write accessed using the same value.

Caution In the one-shot pulse mode and external trigger pulse output mode, write access using "1", the same value as that of bit TTnEST, functions as one trigger.

After reset: 00H R/W Address: TR0CTL1 FFFFF691H, TR1CTL1 FFFFF6A1H

	7	6	5	4	3	2	1	0
TTnCTL1 (n = 0, 1)	0	TTnEST	TTnEEE	0	TTnMD3	TTnMD2	TTnMD1	TTnMD0

TTnEST	Software trigger control
0	No operation
1	Enable 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) **Note:** "0" is always read out from the TTnEST bit.	

TTnEEE	Count clock selection
0	Use of clock selected with bits TTnCKS2 to TTnCKS0 of TTnCTL0 register
1	Use of 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.

TTnMD3	TTnMD2	TTnMD1	TTnMD0	Timer mode
0	0	0	0	Interval mode
0	0	0	1	External event count mode
0	0	1	0	External trigger pulse output mode
0	0	1	1	One-shot pulse mode
0	1	0	0	PWM mode
0	1	0	1	Free-running mode
0	1	1	0	Pulse width measurement mode
0	1	1	1	Triangular wave PWM mode
1	0	0	0	Encoder compare mode
1	0	0	1	Encoder capture mode
1	0	1	0	Encoder capture compare mode
1	1	0	0	Offset trigger generation mode
Other than above				Setting prohibited

Caution Rewrite the TTnMD3 to TTnMD0 bits only when TTnCE = 0. (The same value can be written when TTnCE = 1.) The operation is not guaranteed if rewriting is performed when TTnCE = 1. If rewriting was mistakenly performed, set TTnCE = 0.

(3) TTnCTL2 - TMTn control register 2

The TTnCTL2 register is an 8-bit register that controls the operation of TMTn.

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

Reset input clears this register to 00H.

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.)

Set the TTnCTL2 register when TTnCE = 0. When TTnCE = 1, write access to the TTnCTL2 register can be performed with the same value.

After reset: 00H R/W Address: TR0CTL1 FFFFF692H, TR1CTL1 FFFFF6A2H

	7	6	5	4	3	2	1	0
TTnCTL2 (n = 0, 1)	TTnECC	0	0	TTnLDE	TTnECM1	TTnECM0	TTnUDS1	TTnUDS0

TTnECC	Selection of initialization/hold of counter value when TTnCE = 0
0	Initialize counter value when TTnCE = 0
1	Hold counter value when TTnCE = 0

When TTnECC = 0, setting TTnCE = 0 causes the counter to be reset to FFFFH, the capture registers (TTnCCR0/TTnCCR1) to be reset to 0000H, and the encoder-dedicated flags (TTnEOF/TTnEUF/TTnESF) to be reset to 0. When TTnECC = 0, the value of the TTnTCW register is loaded to the counter when TTnCE is set from 0 to 1. When TTnECC = 1, setting 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.)

TTnLDE	Encoder load enable
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

TTnECM1	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.)

TTnECM0	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 = 0)
Note: The setting of bit TTnECM0 is valid in the encoder compare mode and encoder capture compare mode. (It is invalid in the encoder capture mode.)	

TTnUDS1	TTnUDS0	Encoder operation mode
0	0	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.
0	1	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).
1	0	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”.
1	1	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.

Note When bits TTnUDS1 and TTnUDS0 are set to 10B or 11B, the settings of bits TTnEIS1 and TTnEIS0 of the TTnIOC3 register are invalid, and these bits are fixed to the setting for detection of both edges.

(4) TTnIOC0 - TMTn I/O control register 0

The TTnIOC0 register is an 8-bit register that controls timer output (TOTn0 and TOTn1 pins).

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

Reset input clears this register to 00H.

Set the TTnIOC0 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC0 register can be performed using the same value.

After reset: 00H R/W Address: TR0IOC0 FFFFF693H, TR1IOC0 FFFFF6A3H

	7	6	5	4	3	2	1	0
TTnIOC0 (n = 0, 1)	0	0	0	0	TTnOL1	TTnOE1	TTnOL0	TTnOE0

TTnOLm	Timer output level setting (TOTnm pin)
0	Normal output (Low level, when output is inactive.)
1	Inverted output (High level, when output is inactive.)

TTnOEm	Timer output control (TOTnm pin)
0	Timer output disabled (TOTnm pin output is fixed to inactive level.)
1	Timer output enabled (A pulse can be output from the TOTnm pin.)

Note m = 0, 1

(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).

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

Reset input clears this register to 00H.

Set the TTnIOC1 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC1 register can be performed using the same value.

After reset: 00H R/W Address: TR0IOC1 FFFFF694H, TR1IOC1 FFFFF6A4H

	7	6	5	4	3	2	1	0
TTnIOC1 (n = 0, 1)	0	0	0	0	TTnIS3	TTnIS2	TTnIS1	TTnIS0

TTnIS3	TTnIS2	Capture input (TITn1) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

Capture operation is performed and capture interrupt (INTTTnCC1) is output upon edge detection.

Note: The setting of bits TTnIS3 and TTnIS2 are valid in the free-running mode, pulse width measurement mode, encoder capture mode, and encoder capture compare mode.

TTnIS1	TTnIS0	Capture input (TITn0) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

Capture operation is performed and capture interrupt (INTTTnCC0) is output upon edge detection.

Note: The setting of bits TTnIS1 and TTnIS0 are valid in the free-running mode, pulse width measurement mode, encoder capture mode, and encoder capture compare mode.

(6) TTnIOC2 - TMTn I/O control register 2

The TTnIOC2 register is an 8-bit register that controls the valid edge of external event count input (TEVTTn pin) and external trigger input (TTRGTn pin).

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

Reset input clears this register to 00H.

Set the TTnIOC2 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC2 register can be performed using the same value.

After reset: 00H R/W Address: TR0IOC2 FFFFF695H, TR1IOC2 FFFFF6A5H

	7	6	5	4	3	2	1	0
TTnIOC2	0	0	0	0	TTnEES1	TTnEES0	TTnETS1	TTnETS0

(n = 0, 1)

TT1EES1	TT1EES0	External event counter input (TEVTTn) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection
Note: The settings of bits TTnEES1 and TTnEES0 are valid in the external event count mode, or when bit TTnEEE of the TTnCTL1 register = 1.		

TT1ETS1	TT1ETS0	External trigger input (TTRGTn) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection
Note: The settings of bits TTnETS1 and TTnETS0 are valid in the external trigger pulse output mode and the one-shot pulse mode.		

(7) TTnIOC3 - TMTn I/O control register 3

The TTnIOC3 register is an 8-bit register that controls the valid edge of encoder clear input (TECRTn pin) and encoder input (TENCTn1 and TENCTn0 pins).

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

Reset input clears this register to 00H.

Set the TTnIOC3 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC2 register can be performed using the same value.

After reset: 00H R/W Address: TR0IOC3 FFFFF696H, TR1IOC3 FFFFF6A6H

	7	6	5	4	3	2	1	0
TTnIOC3 (n = 0, 1)	TTnSCE	TTnZCL	TTnBCL	TTnACL	TTnECS1	TTnECS0	TTnEIS1	TTnEIS0

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 0000H 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.	

TTnZCL	Sets 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.	

TTnBCL	Sets 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.	

TTnACL	Sets the clear level for the A phase of encoder input (TENCTn0 pin)
0	Clear condition = Low level
1	Clear condition = High level
Note: The TTnACL bit is valid when TTnSCE = 1.	

TTnECS1	TTnECS0	Set the valid edge of encoder clear input (TECRTn pin)
0	0	No edge detection
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both rising and falling edge detection

The encoder clear interrupt (INTTTnEC) is output upon detection of the valid edge set with bits TTnECS1, TTnECS0.

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.

TTnEIS1	TTnEIS0	Set the valid edge of the encoder input signal (TENCTn1/TENCTn0 pins)
0	0	No edge detection
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both rising and falling edge detection

Note: Bits TTnEIS1 and TTnEIS0 are valid when bits TTnUDS1 and TTnUDS0 of register TTnCTL2 are 00_B or 01_B.

(8) TTnOPT0 - TMTn option register 0

The TTnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

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

Reset input clears this register to 00H.

Set the bits of the TTnOPT0 register other than TTnOVF when TTnCE = 0. When TTnCE = 1, write access of bits of the TTnOPT0 register other than TTnOVF can be performed using the same value.

After reset: 00H R/W Address: TR0IOC3 FFFFF697H, TR1IOC3 FFFFF6A7H

	7	6	5	4	3	2	1	0
TTnOPT0 (n = 0, 1)	0	0	TTnCCS1	TTnCCS0	0	0	0	TTnOVF

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.	

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.	

TTnOVF	Flag that indicates TMTn overflow
0	No overflow occurrence after timer restart or flag reset
1	Overflow occurrence

In the free-running mode, pulse width measurement mode, and offset trigger generation mode, if the counter value is counted up from FFFFH, overflow occurs, the OVF flag is set (1), and the counter is cleared to 0000H. The counter is also cleared by writing 0. At the same time that the TTnOVF flag is set (1), an overflow interrupt (INTTTnOV) occurs.
If 0 is written to the TTnOVF flag, or if TTnECC = 0 and TTnCE = 0 are set, the counter is cleared.

Note: Overflow does not occur during compare match & clear operation for counter value FFFFH and compare value FFFFH.

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.

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

Reset input clears this register to 00H.

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.))

After reset: 00H R/W Address: TR0IOC3 FFFFF698H, TR1IOC3 FFFFF6A8H

	7	6	5	4	3	2	1	0
TTnOPT1 (n = 0, 1)	0	0	0	0	0	TTnEUF	TTnEOF	TTnESF

TTnEUF	Indication of 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 0000H, underflow occurs, the OVF flag is set (to 1), and the counter is set to FFFFH. When the TTnEUF flag is set (to 1), an overflow interrupt (INTTTnOV) occurs at the same time.

The TTnEUF flag is cleared (to 0) under the following conditions.

- When 0 is written by CPU instruction
- When TTnCE = 0 is set while TTnECC = 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.

Note: When bit TTnECC of the TTnCTL2 register is 1, the flag status is held even if the value of bit TTnCE is changed from 1 to 0.

TTnEOF	Indication of 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 FFFFH, overflow occurs, the OVF flag is set (1), and the counter is cleared to 0000H. At the same time that the TTnEOF flag is set (1), an overflow interrupt (INTTTnOV) occurs.
However, the TTnOVF flag is not set (to 1).

The TTnEOF flag is cleared (0) under the following conditions.

- When 0 is written by CPU instruction
- When TTnCE = 0 is set while TTnECC = 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.

Note: When bit TTnECC of the TTnCTL2 register is 1, the flag status is held even if the value of bit TTnCE is changed from 1 to 0.

TTnESF	Indication of 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 TTnCE = 0 is set while TTnECC = 0

Note: When bit TTnECC of the TTnCTL2 register is 1, the flag status is held even if the value of bit TTnCE 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.

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

Reset input clears this register to 00H.

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.

After reset: 00H R/W Address: TR0IOC3 FFFFF699H, TR1IOC3 FFFFF6A9H

	7	6	5	4	3	2	1	0
TTnOPT2 (n = 0, 1)	0	0	0	0	0	0	0	TTnRSF

TTnRSF	Reload status flag
0	No reload request, or reload completed
1	Reload request was output

It 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.

18.5 Basic Operation

18.5.1 Basic counter operation

This section describes the basic operation of the counter. For details, refer to chapter *"Operation in Each Mode" on page 1050*.

(1) Counter start operation

(a) Encoder compare mode, encoder capture mode, encoder capture compare mode

The count operation is controlled by the phases of pins TENCTn0 and TENCTn1.

When TTnCE = 0 and TTnECC = 0, the counter is initialized by the TTnTCW register and the count operation is started. (The setting value of the TTnTCW register is loaded to the counter at the timing when TTnCE changes from 0 to 1.)

(b) Triangular wave PWM MODE

The counter starts counting from initial value FFFFH.

It counts up FFFFH, 0000H, 0001H, 0002H, 0003H…

Following count up operation, the counter counts down upon a match with the TTnCCR0 register.

(c) Modes other than the above

The counter starts counting from initial value FFFFH.

It counts up FFFFH, 0000H, 0001H, 0002H, 0003H…

(2) Counter clear operation

There are the following five counter clear causes.

- Clear through match between counter value and compare setting value.
- Capture and clear through capture input
- Counter clear through encoder clear input (TECRTn pin)
- Counter clear through match with clear condition level
- Clear through clear signal input (TTnSYCI) for synchronization function during slave operation

Table 18-6 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 bits TTnECM0 and TTnECM1 of the TTnCTL2 register.

(3) Counter reset and hold operations

In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, counter value hold is controlled with bit TTnECC of the TTnCTL2 register.

If TTnCE = 0 is set when TTnECC = 0, the counter is reset to 0000H. The setting value of the TTnTCW register is loaded to the counter when TTnCE = 1 is set next.

If TTnCE = 0 is set when TTnECC = 1, the counter value is held as is. Counting resumes from the held value when TTnCE = 1 is set next.

(4) Counter read operation during counter operation

In TMT, the counter value can be read during count operation using the TTnCNT register.

(5) Overflow operation

Counter overflow occurs in the free-running mode, pulse width measurement mode, encoder compare mode, encoder capture mode, encoder capture/compare mode, and offset trigger generation mode.

Overflow occurs when the counter value changes from FFFFH to 0000H.

In the free-running mode, pulse width measurement mode, offset trigger generation mode, the overflow flag (TTnOVF) is set to 1 and an overflow interrupt (INTTTnOV) is output. At this time, the TTnEOF flag is not set.

In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, the encoder dedicated overflow flag (TTnEOF) is set to 1 and an overflow interrupt (INTTTnOV) occurs. At this time, the TTnOVF flag is not set.

Under the following conditions, overflow does not occur.

- When the counter value changes from initial setting FFFFH to 0000H immediately after counting start
- When FFFFH is set to the compare register, and the counter is cleared to 0000H upon a match between the counter value and the compare setting value.
- When, in the pulse width measurement mode and offset trigger generation mode, capture operation is performed for counter value FFFFH, and the counter is cleared to 0000H.

(6) Underflow operation

Counter underflow occurs in the triangular wave PWM Mode, encoder compare mode, encoder capture mode, and encoder capture/compare mode.

Underflow occurs when the counter value changes from 0000H to FFFFH.

When underflow occurs in the triangular wave PWM mode, an overflow interrupt (INTTTnOV) occurs. At this time, the TTnOVF flag is not set.

In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, the encoder dedicated underflow flag (TTnEUF) is set to 1, and an overflow interrupt (INTTTnOV) occurs.

Underflow does not occur during count down immediately following counter start.

(7) Description of interrupt signal operation

In TMT, the following interrupt signals are output.

Name	Occurrence cause
INTTTnCC0	• 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.

18.5.2 Method for writing to compare register

The TTnCCR0 and TTnCCR1 registers can be rewritten during timer operation (TTnCE = 1). There are two write modes (anytime write, reload), depending on the mode.

(1) Anytime rewrite method

When the TTnCCR0 and TTnCCR1 registers are written during timer operation, the write value is immediately transferred to the TTnCCR0 buffer register and TTnCCR1 buffer register and is used as the value to be compared with the counter.

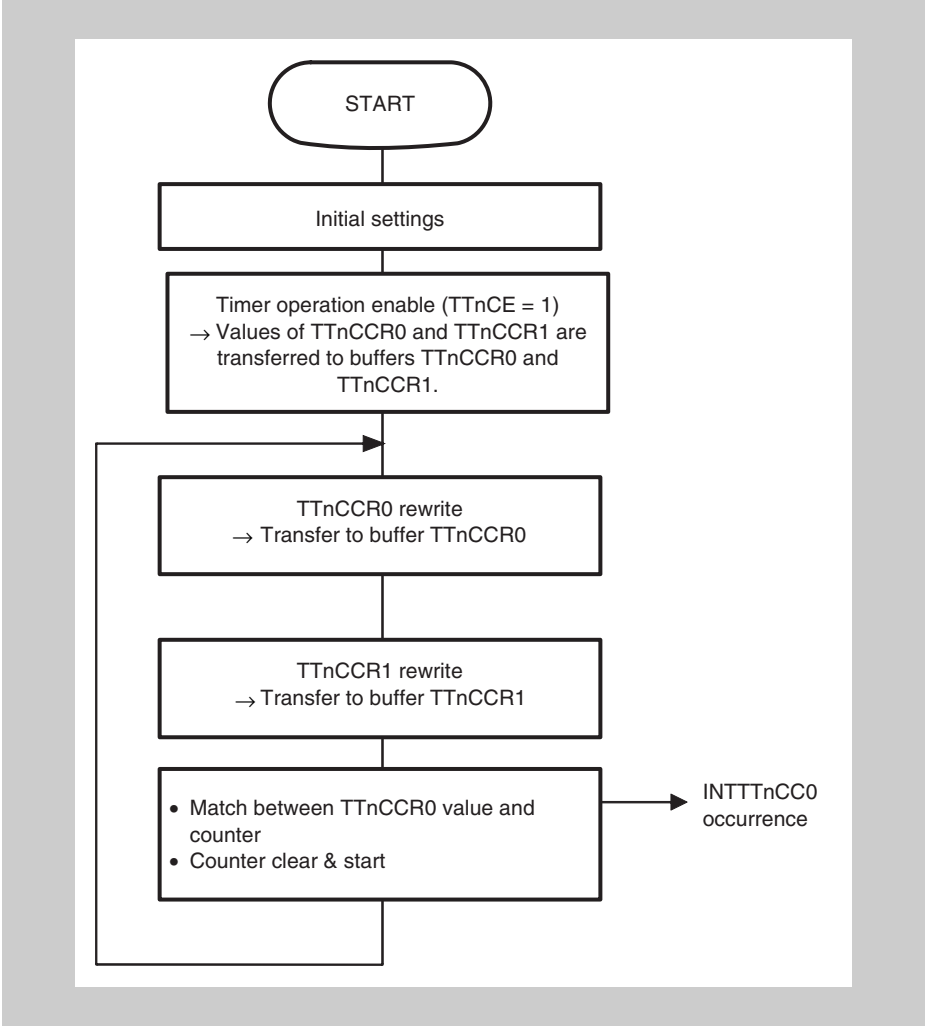


Figure 18-2 Basic operation flow for anytime rewrite

Note The interval mode is used as an example.

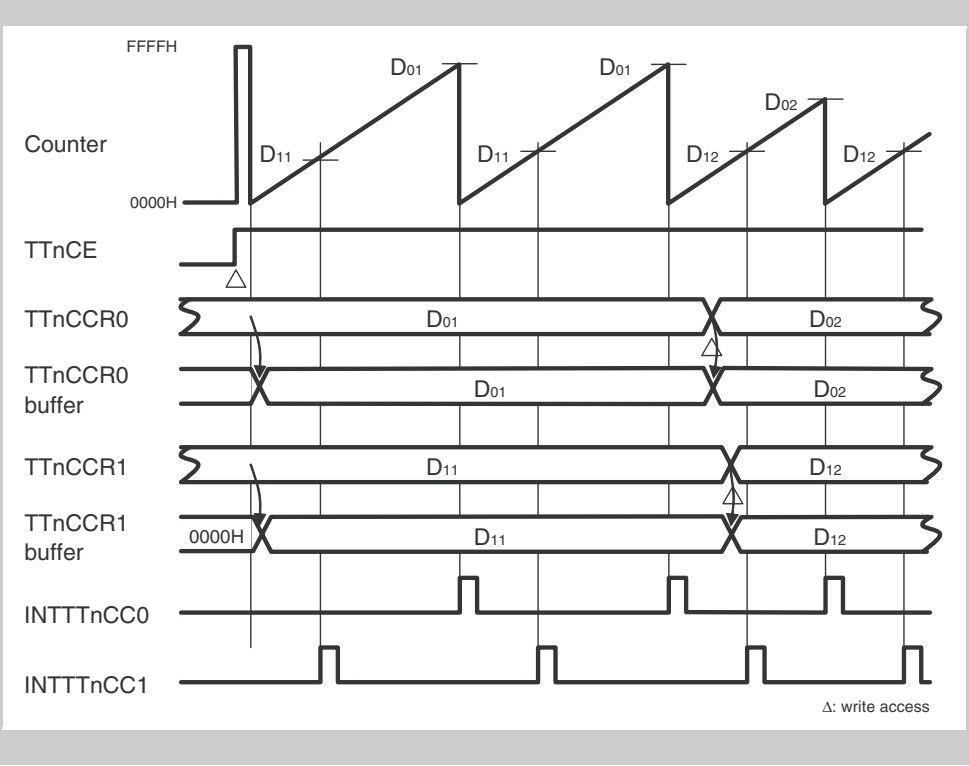


Figure 18-3 Basic anytime rewrite operation timing

Note 1. D_{01}, D_{02}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting values of TTnCCR1 register (0000H to FFFFH)

2. The interval mode is used as an example.

(2) Reload method (Batch rewrite)

When TTnCCR0, TTnCCR1 register write is performed during timer operation, the written value is used as the comparison value for the counter via the TTnCCR0 and TTnCCR1 buffer registers.

Under the reload method, rewrite the TTnCCR0 register before the TTnCCR0 register value is matched, and next, write to the TTnCCR1 register.

Then, when the TTnCCR0 register is matched or the counter is cleared to 0000H through external input, the values of the TTnCCR0 register and TTnCCR1 register are reloaded.

By writing to the TTnCCR1 register, the value becomes valid at the next reload timing.

Therefore, even if wishing to rewrite only the value of the TTnCCR0, rewrite the same value to the TTnCCR1 register to make the next reload valid.

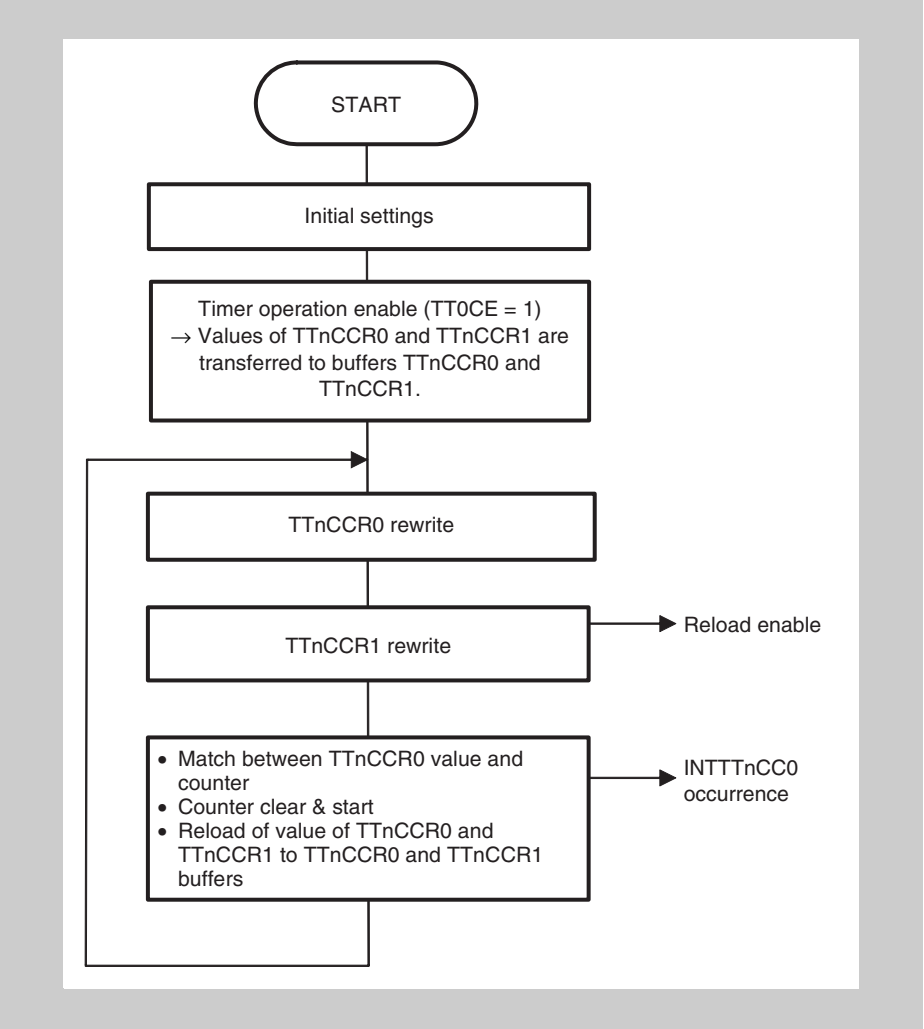


Figure 18-4 Basic operation flow for reload (batch rewrite)

Caution Rewrite to the TTnCCR1 register includes enabling reload. Therefore, rewrite the TTnCCR1 register after rewriting the TTnCCR0 register.

Note The PWM mode is used as an example.

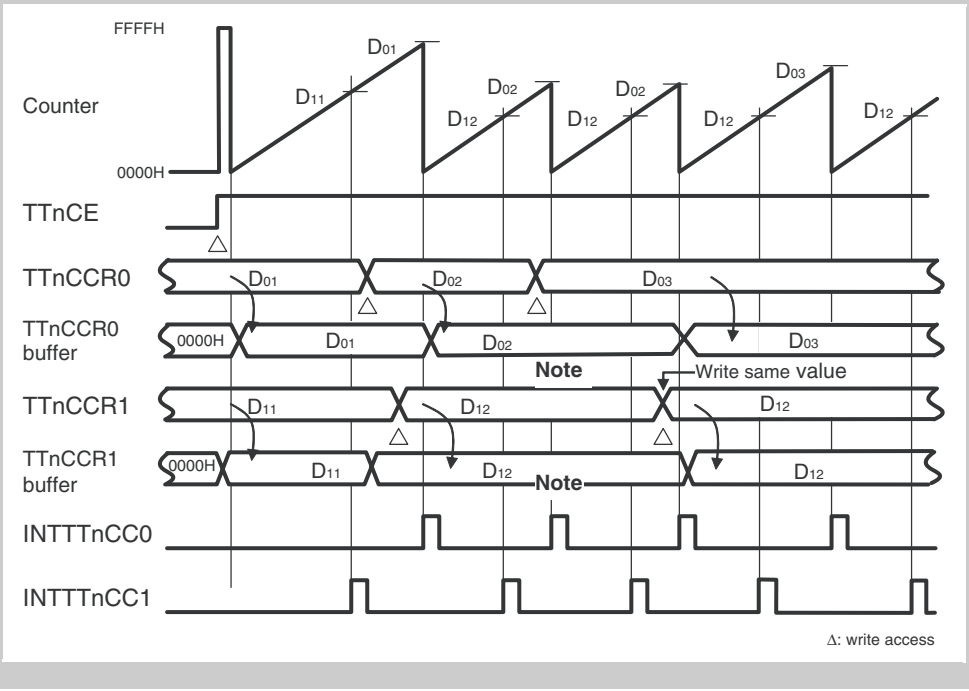


Figure 18-5 Basic reload operation timing

Note 1. Since the TTnCCR1 register is not written to, reloading is not performed even if TTnCCR0 is rewritten.

2. D_{01}, D_{02}, D_{03}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting values of TTnCCR1 register (0000H to FFFFH)

3. The PWM mode is used as an example.

Table 18-7 Capture/compare rewrite methods in each mode

Operation mode	Capture/compare rewrite method	
	TTnCCR0	TTnCCR1
Interval mode	Compare only (Anytime write type)	
External event count mode		
External trigger pulse output mode	Compare only (Reload type)	
One-shot pulse mode	Compare only (Anytime write type)	
PWM mode	Compare only (Reload type)	
Free-running mode	Capture/compare selectable (When compare is selected, anytime write type)	
Pulse width measurement mode	Capture only	
Triangular wave PWM mode	Compare only (Reload type)	
Encoder compare mode	Compare only (Anytime write type)	
Encoder capture mode	Capture only	
Encoder capture compare mode	Compare only (Anytime write type)	Capture only
Offset trigger generation mode	Capture only	Compare only (Reload type)

18.6 Operation in Each Mode

18.6.1 Interval timer mode

In the interval timer mode, a compare match interrupt (INTTTnCC0) occurs and the counter is cleared upon a match between the setting value of the TTnCCR0 register and the counter value. The occurrence interval for this counter and TTnCCR0 register match interrupt becomes the interval time.

In the interval timer mode, the counter is cleared only upon a match between the counter and the value of the TTnCCR0 register. Counter clearing using the TTnCCR1 register is not performed.
However, the setting value of the TTnCCR1 is compared to the counter value transferred to the TTnCCR1 buffer register and a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTnCE.

Pins TOTn0 and TOTn1 are toggle output controlled when bits TTnOE0 and TTnOE1 are set to 1.

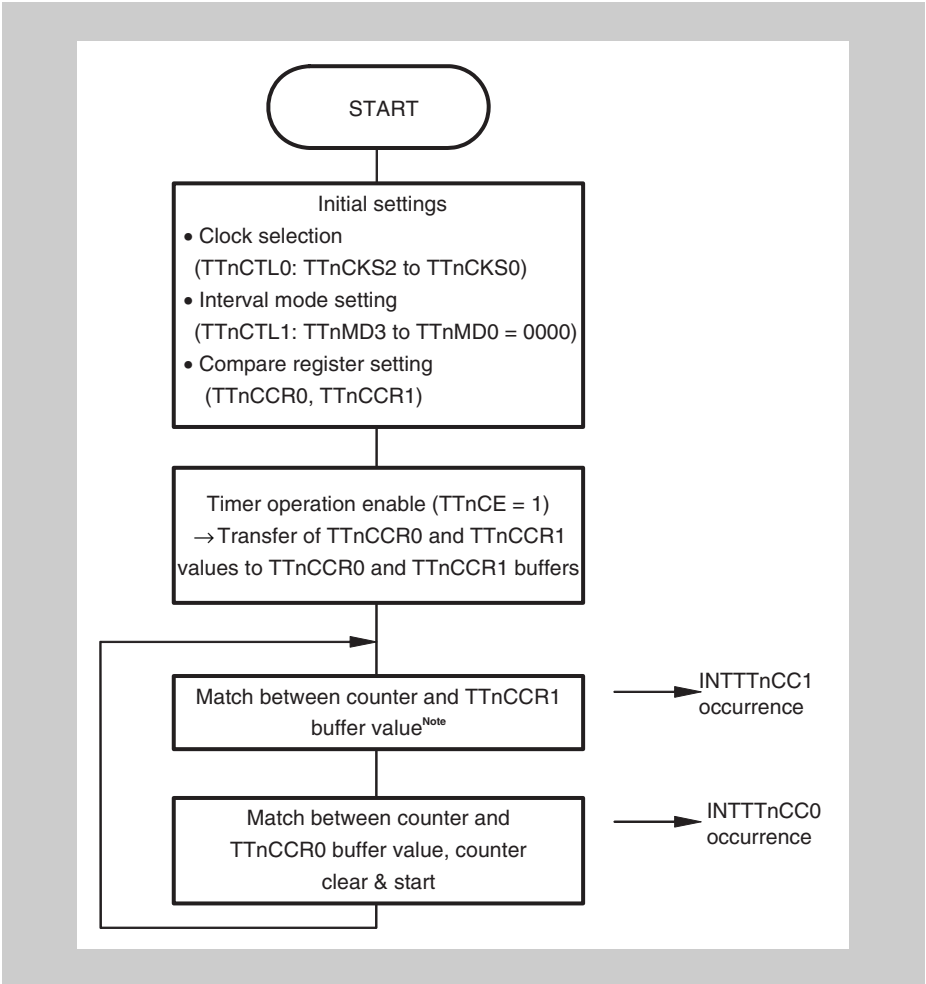


Figure 18-6 Basic operation flow in interval timer mode

Note In the case of a match between the counter and TTnCCR1 register, the counter is not cleared.

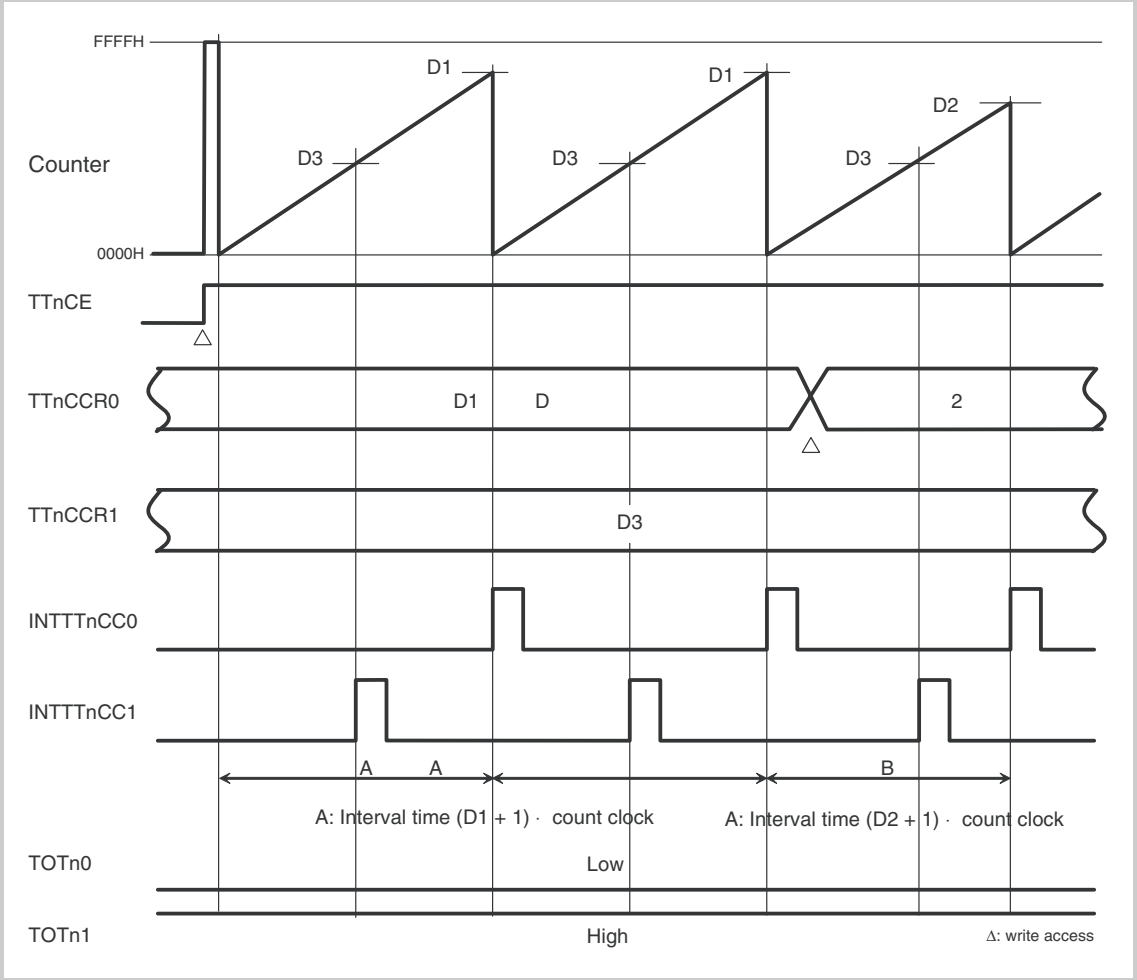


Figure 18-7 Basic timing in interval timer mode (1/2)
When D1>D2>D3, only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)

Note
1. D1, D2: Setting values of TTnCCR0 register (0000H to FFFFH)
   D3: Setting values of TTnCCR1 register (0000H to FFFFH)
2. Interval time = (Dm + 1) × (count clock cycle)
3. m = 1 to 3

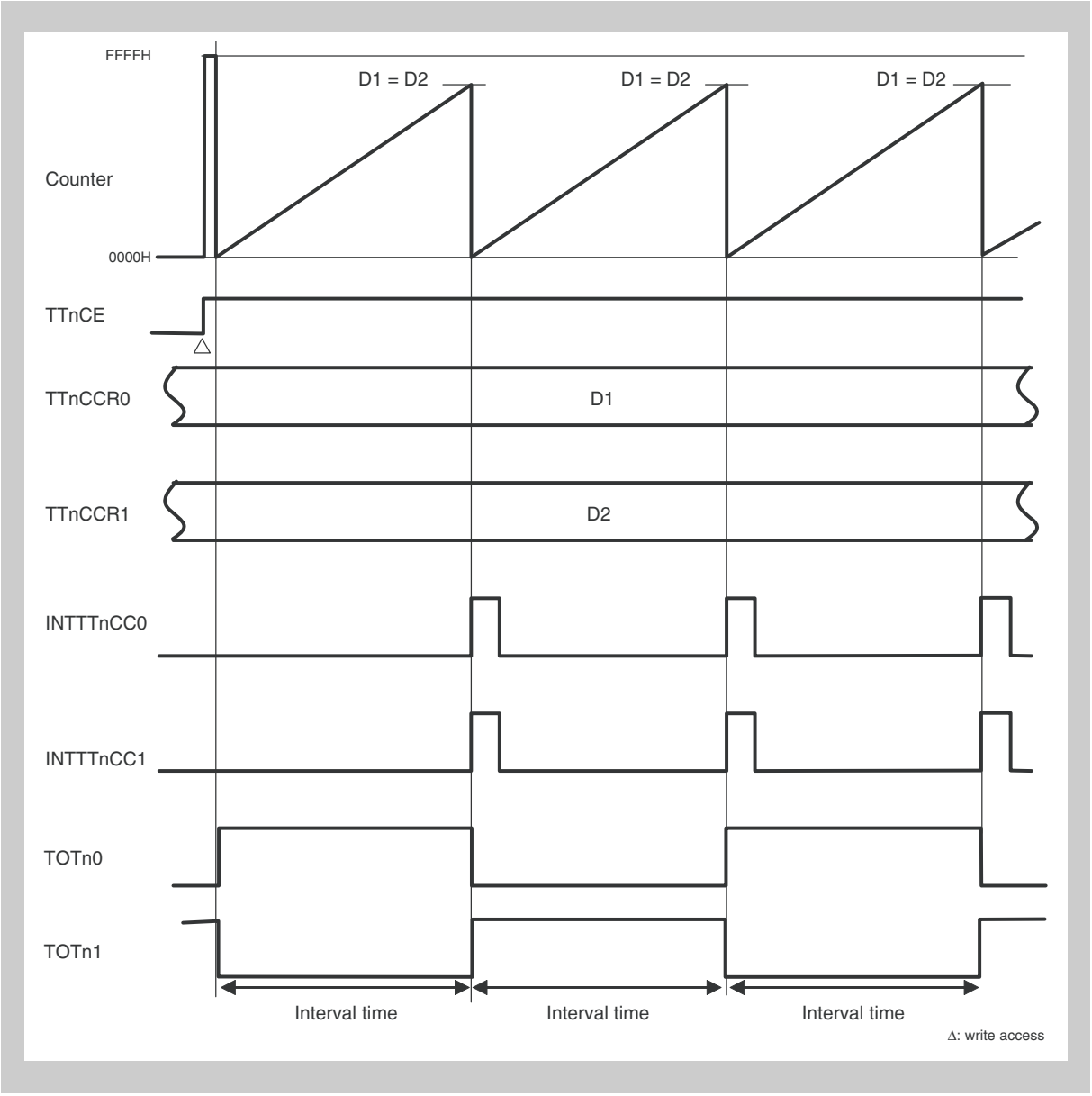


Figure 18-8 Basic timing in interval timer mode (2/2)
When D1 = D2, values of TTnCCR0 and TTnCCR1 registers not rewritten, TOTn1 output performed (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note
1. D1: Setting value of TTnCCR0 register (0000H to FFFFH)
   D2: Setting value of TTnCCR1 register (0000H to FFFFH)
2. Interval time = (Dm + 1) × (count clock cycle)
3. TOTn0, TOTn1 toggle time = (Dm + 1) × (count clock cycle)
4. m = 1, 2

18.6.2 External event count mode

In the external event count mode, count up starts upon external event input (TEVTTn pin). (The external event input (TEVTTn) is used as the count clock, regardless of bit TTnEEE of the TTnCTL1 register.)

In the external event count mode, the counter is cleared only upon a match between the counter and the value of the TTnCCR0 register. Counter clearing using the TTnCCR1 register does not work.

However, the value of the TTnCCR1 register is transferred to the TTnCCR1 buffer register, compared to the counter value, and a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten with the anytime write method, regardless of the value of bit TTnCE.

Pins TOTn0 and TOTn1 are toggle output controlled when bits TTnOE0 and TTnOE1 are set to 1.

When using only one compare register channel, it is recommended to set the TTnCCR1 register to FFFFH.

[External event count operation flow]

<1> TTnCTL1 register bits TTnMD3 to TTnMD0 = 0001B (mode setting) Edge detection set with TTnIOC2 register bits TTnEES1 and TTnEES0 (TTnEES1, TTnEES0 = setting other than 01B)

<2> TTnCTL0 register bit TTnCE = 1 (count enable)

<3> TEVTTn pin input edge detection (count-up start)

Caution

1. In external event count mode, when the setting value of the TTnCCR0 register is set to m, the number of TEVTTn pin input edge detection times is m+1.
2. In external event count mode, do not send the TTnCCR0 register to 0000H.

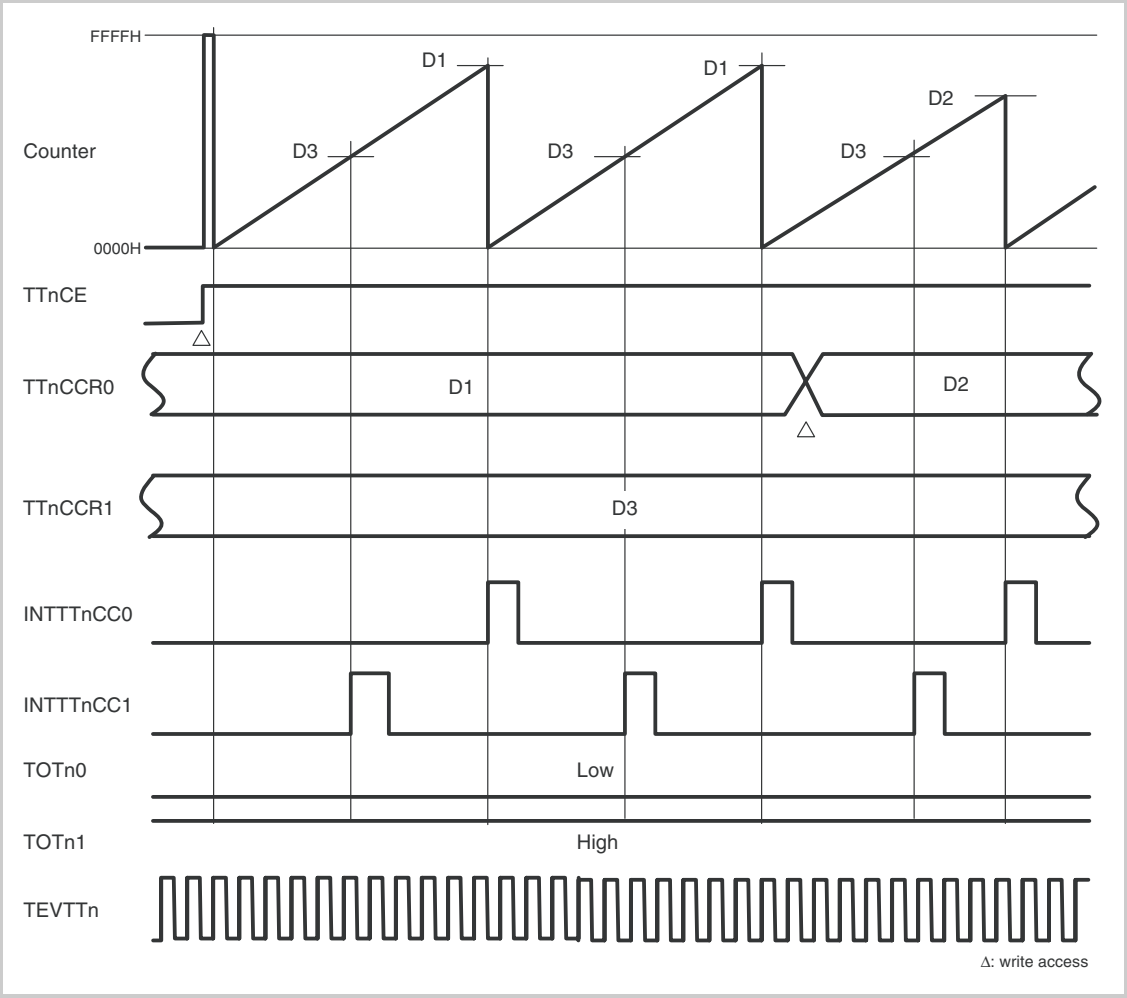


Figure 18-9 Basic operation timing in external event count mode (1/4)
When D1>D2>D3, only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output. The signal input from TEVTTn and internally synchronized is counted as the count clock (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)

Note
1. D1, D2: Setting values of TTnCCR0 register (0000H to FFFFH)
   D3: Setting value of TTnCCR1 register (0000H to FFFFH)
2. Number of event counts = (Dm + 1) (m = 1, 2)

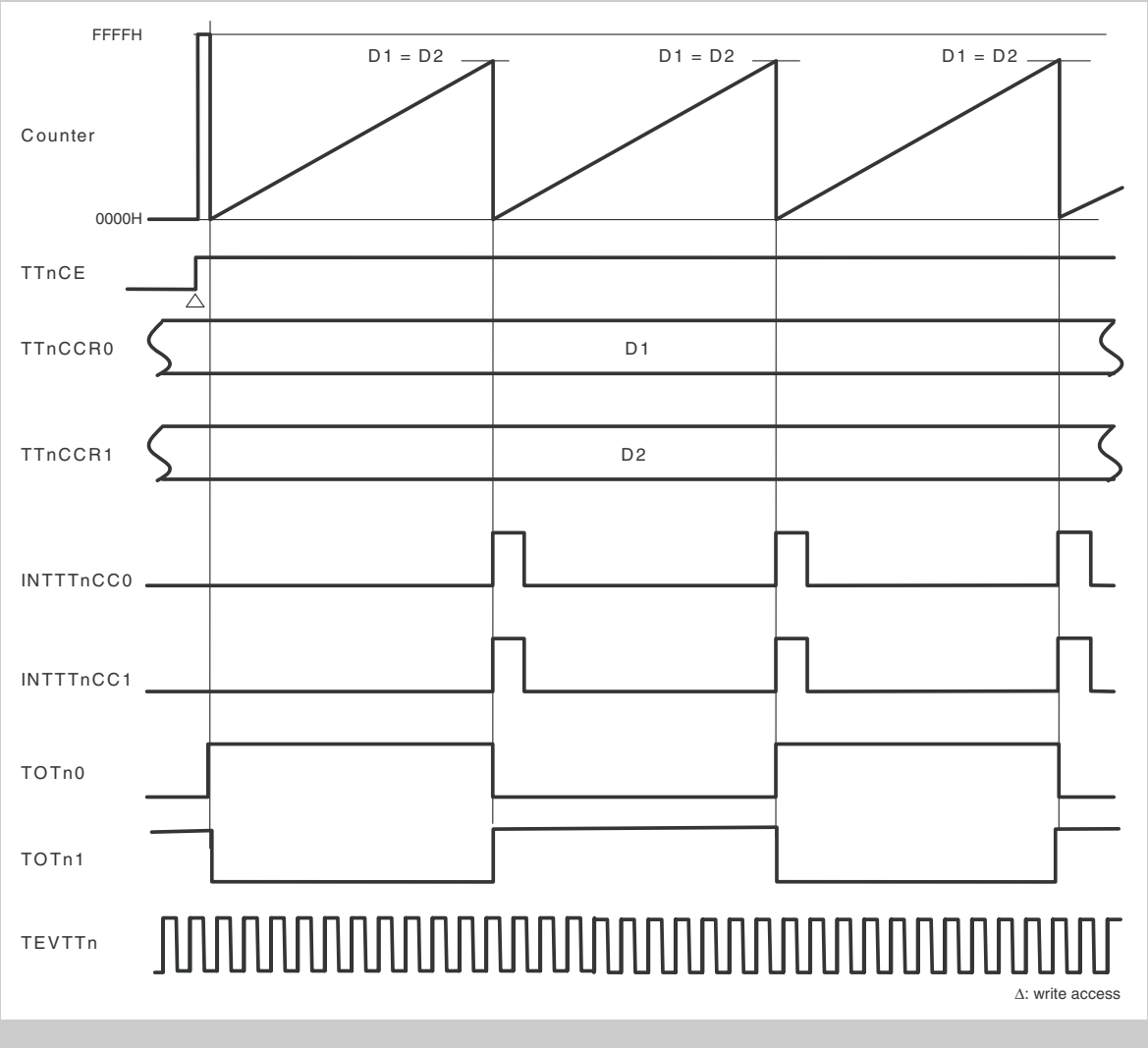


Figure 18-10 Operation timing in external event count mode (2/4)
When D1 = D2, TTnCCR0 and TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note 1. D1: Setting value of TTnCCR0 register (0000H to FFFFH)
D2: Setting value of TTnCCR1 register (0000H to FFFFH)

2. Number of event counts = (Dm + 1) (m = 1, 2)

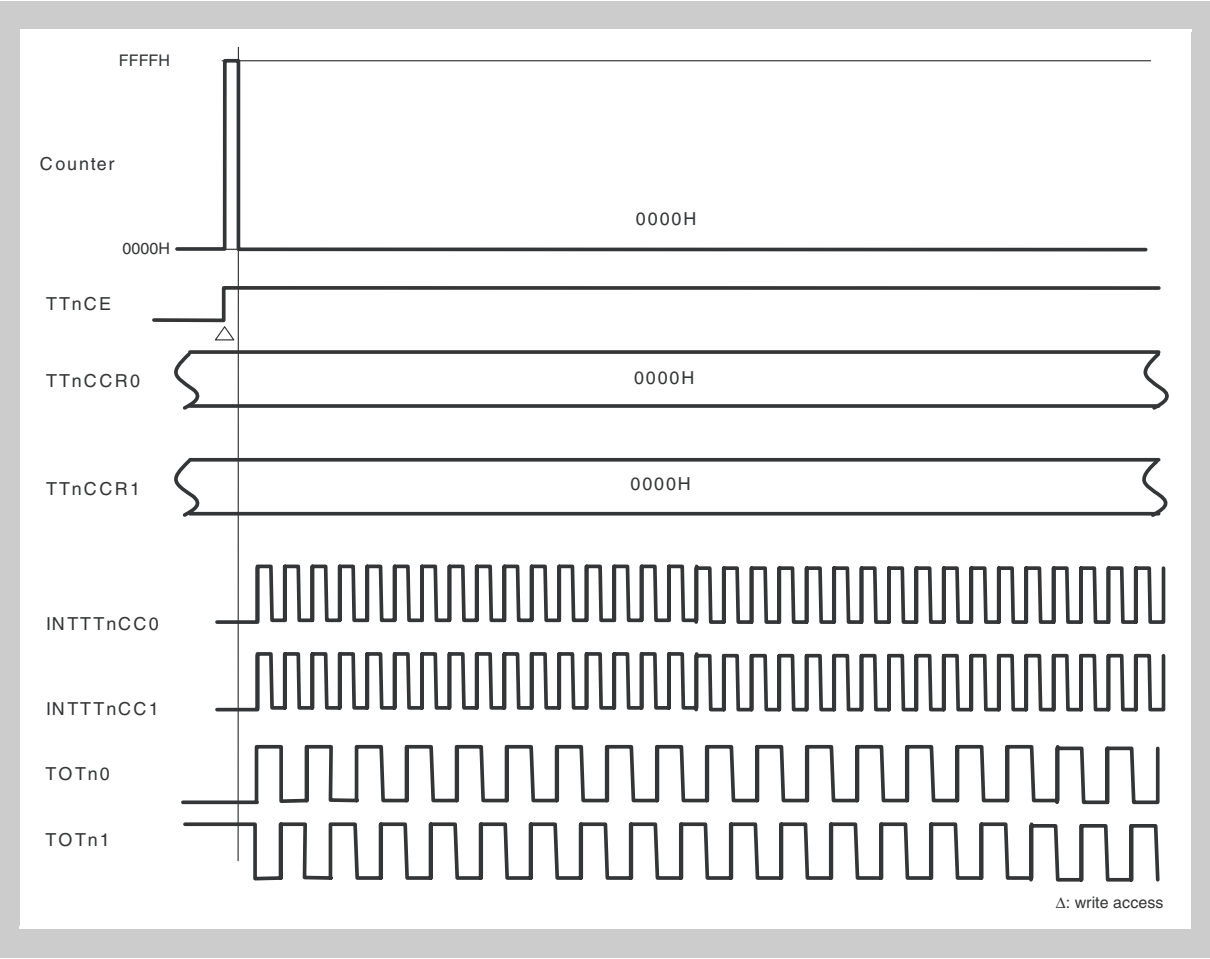


Figure 18-11 Operation timing in external event count mode (3/4)
When D1 = D2, TTnCCR0 and TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note 1. D1: Setting value of TTnCCR0 register (0000H)
D2: Setting value of TTnCCR1 register (0000H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

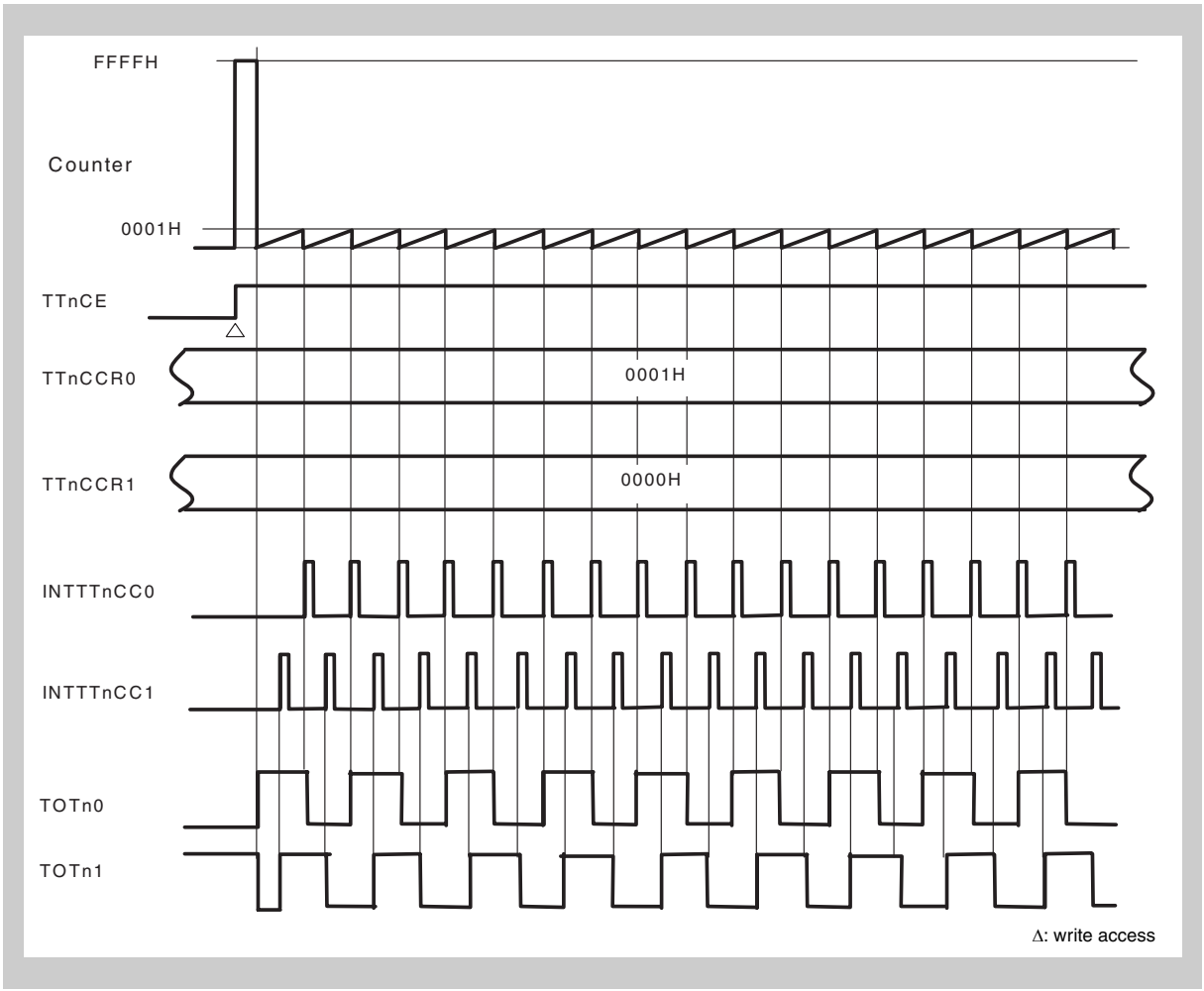


Figure 18-12 Basic operation timing in external event count mode (4/4)
When D1 = D2, TTnCCR0, TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note 1. D1: Setting value of TTnCCR0 register (0001H)
D2: Setting value of TTnCCR1 register (0000H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

18.6.3 External trigger pulse output mode

When, in the external trigger pulse mode, the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when the TTnEST bit of the TTnCTL1 register is set, count up starts.
An external trigger pulse is output from pin TOTn1, and toggle output is performed from pin TOTn0 upon a match with the TTnCCR0 register.
Moreover, during the count operation, upon a match between the counter and the TTnCCR0 register, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload is performed at the timing when the counter value and the TTnCCR0 register match.
However, when write access to the TTnCCR1 register is performed, the next reload timing becomes valid, so that even if wishing to rewrite only the value of the TTnCCR0 register, write the same value to the TTnCCR1 register. In this case, reload is not performed even if only the TTnCCR0 register is rewritten.

If, during operation in the external trigger pulse output mode, the external trigger (TTRGTn pin) edge is detected several times, or if the TTnEST bit of the TTnCTL1 register is set (to 1), the counter is cleared and count up is resumed.
Moreover, if at this time, the TOTn1 pin is in the low level status, the TOTn1 pin output becomes high level when an external trigger is input. If the TOTn1 pin is in the high level status, it remains high level even if external trigger input occurs.

In the external trigger pulse output mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

Caution In the external trigger pulse mode, set bit TTnEEE of the TTnCTL1 register to 0.

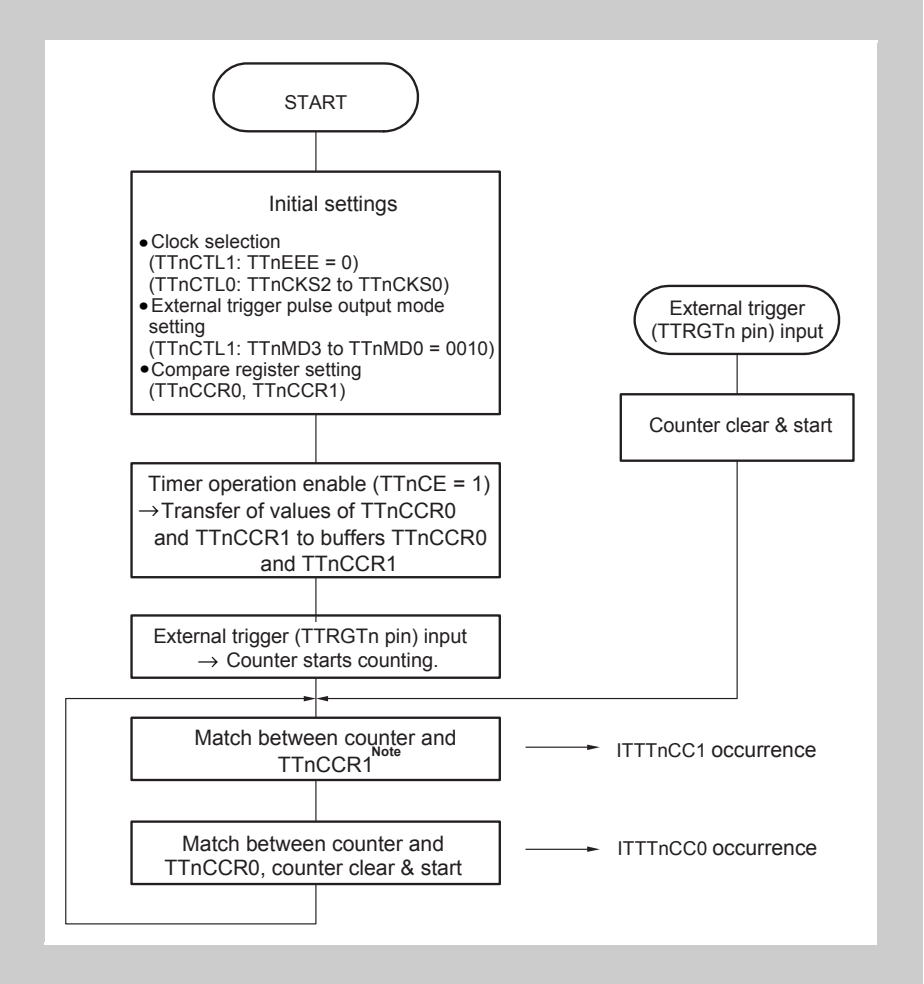


Figure 18-13 Basic operation flow in external trigger pulse output mode

Note The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.

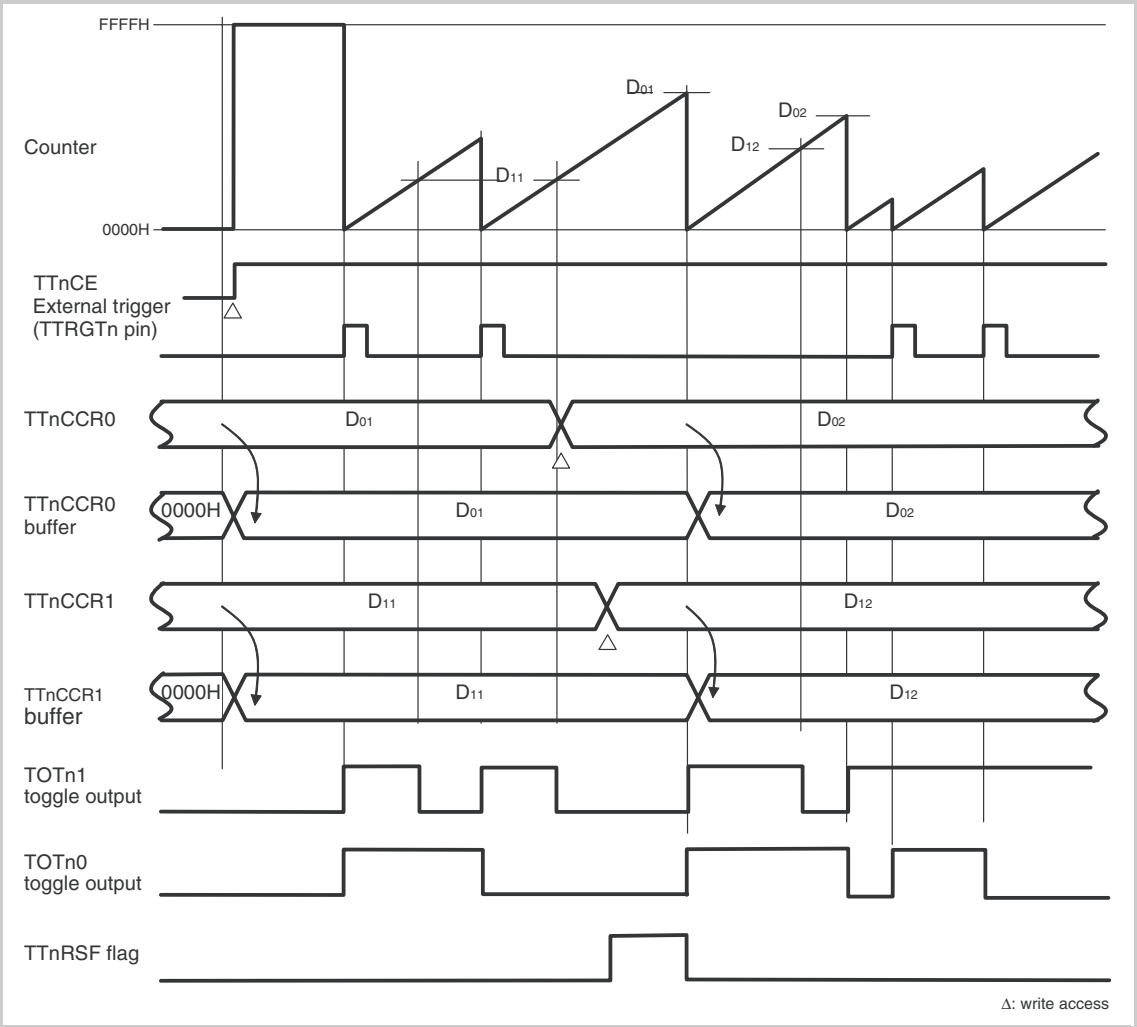


Figure 18-14 Basic operation timing in external trigger pulse output mode When values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note

1. D_{01}, D_{02}: Setting values of TTnCCR0 register (0000H to FFFFH)
   D_{11}, D_{12}: Setting values of TTnCCR1 register (0000H to FFFFH)
2. TOTn1 (PWM) duty = (setting value of TTnCCR1 register) × (count clock cycle) TOTn1 (PWM) cycle = (setting value of TTnCCR0 register + 1) × (count clock cycle)
3. Pin TOTn0 is toggled when the counter is cleared immediately following count start.

18.6.4 One-shot pulse mode

When, in the one-shot pulse mode, the duty is set to the TTnCCR0 register, the output duty delay value is set to the TTnCCR1 register, and bit TTnCE of the TTnCTL0 register is set to 1, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when bit TTnEST of the TTnCTL0 register is set to 1, count up starts. The TOTn1 pin becomes high level upon a match between the counter and TTnCCR1 register.
Moreover, upon a match between the counter and TTnCCR0 register, the TOTn1 pin becomes low level, and the counter is cleared to 0000H and then stops. The TOTn0 pin performs toggle output during the count operation upon a match between the counter and the TTnCCR0 buffer register.
Moreover, upon a match between the counter and TTnCCR0 register during count operation, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 buffer register, a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTnCE.

Even if a trigger is input during the counter operation, it is ignored. Be sure to input the second trigger when the counter is stopped at 0000H.

In the one-shot pulse mode, registers TTnCCR0 and TTnCCR1 have their function fixed as compare registers, so the capture function cannot be used.

[One-shot pulse operation flow]

<1> TTnCTL1 register bits TTnMD3 to TTnMD0 = 0011B (One-shot pulse mode)

<2> TTnCCR0 register setting (duty setting),
TTnIOC0 register bit TTnOE1 = 1 (TOTn1 pin output enable)

<3> TTnCTL0 register bit TTnCE = 1 (counter operation enable):
TOTn1 = Low-level output

<4> TTnCTL1 register bit TTnEST = 1 or TTRGTn pin edge detection (count-up start):
TOTn1 = Low-level output

<5> Match between counter value and TTnCCR1 buffer register:
TOTn1 = High-level output

<6> Match between counter value and TTnCCR0 buffer register:
TOTn1 = Low-level output, count clear

<7> Count stop: TOTn1 = Low-level output

<8> TTnCE = 0 (operation reset)

<1> to <2> can be in any order.

Caution In the one-shot pulse mode, set bit TTnEEE of the TTnCTL1 register to 0.

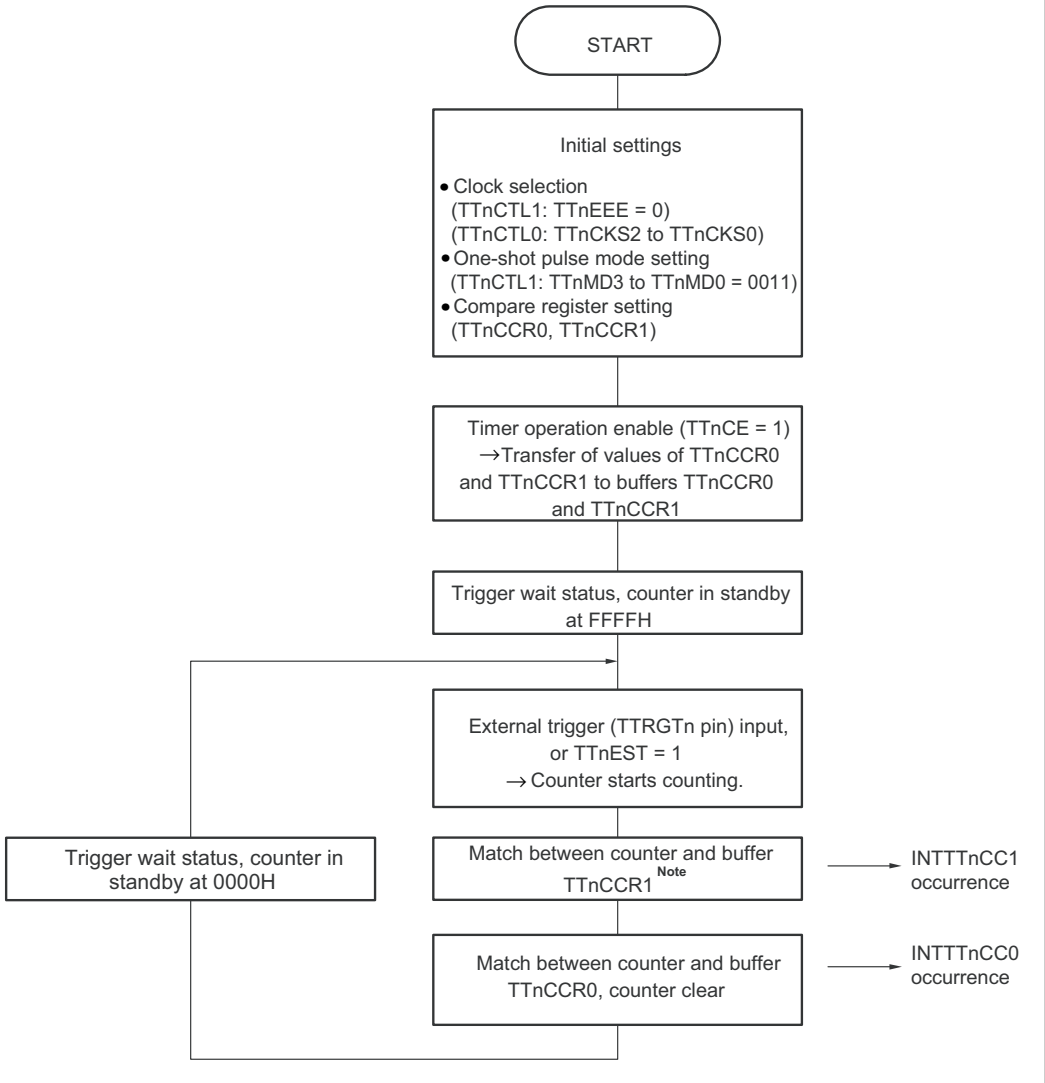


Figure 18-15 Basic operation flow in one-shot pulse mode

Note The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.

Caution The counter is not cleared even if trigger input is realized while the counter counts up, and the trigger input is ignored.

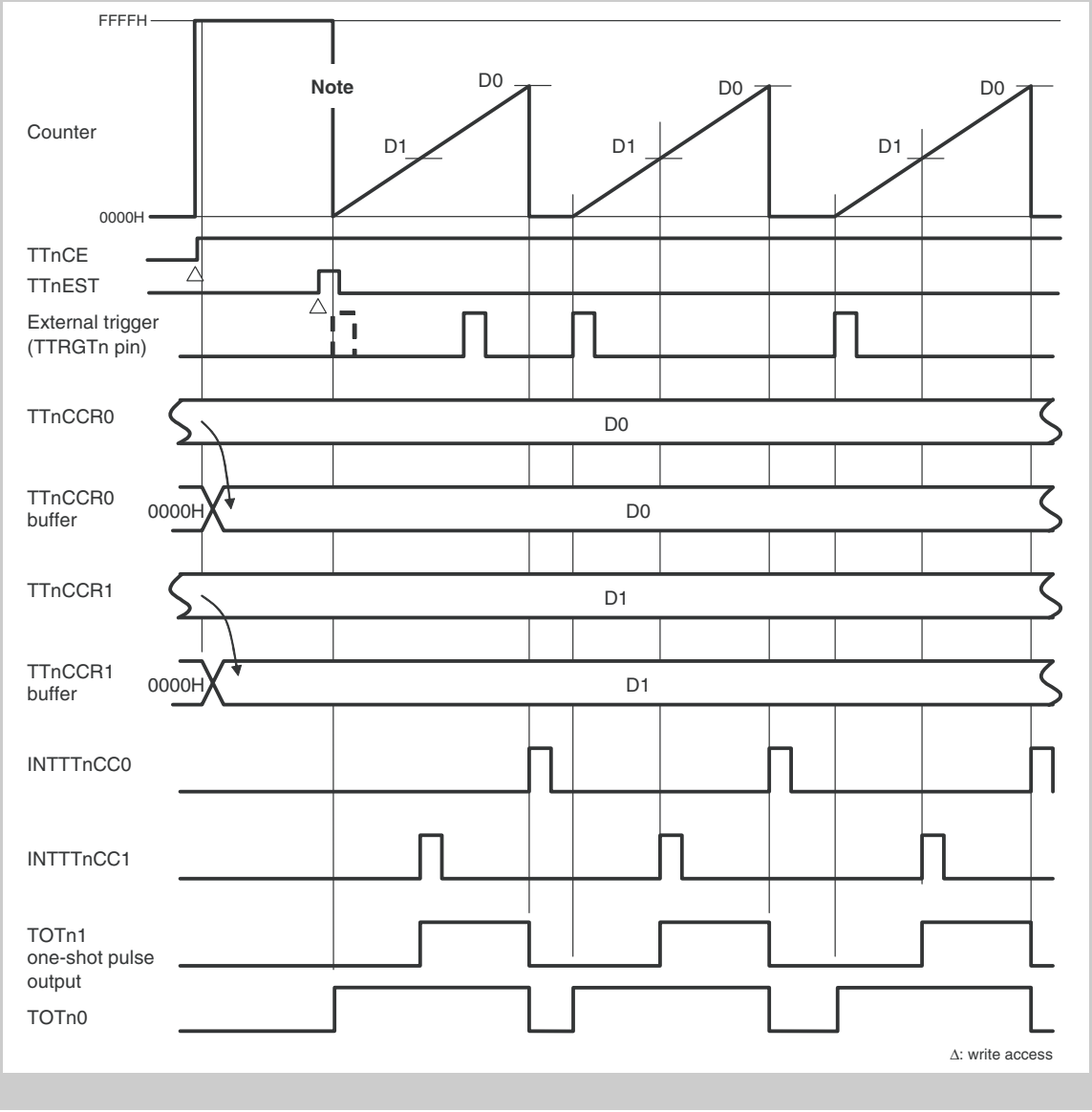


Figure 18-16 Basic operation timing in one-shot pulse mode (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note

1. Count up starts when the value of TTnEST becomes 1 or TTRGTn is input.
2. D0: Setting value of TTnCCR0 register (0000H to FFFFH)
   D1: Setting value of TTnCCR1 register (0000H to FFFFH)
3. TOTn1 (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)

18.6.5 PWM mode

When, in the PWM mode, the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, variable duty PWM output is performed from pin TOTn1.

Simultaneously with the start of count up operation, pin TOTn1 becomes 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.

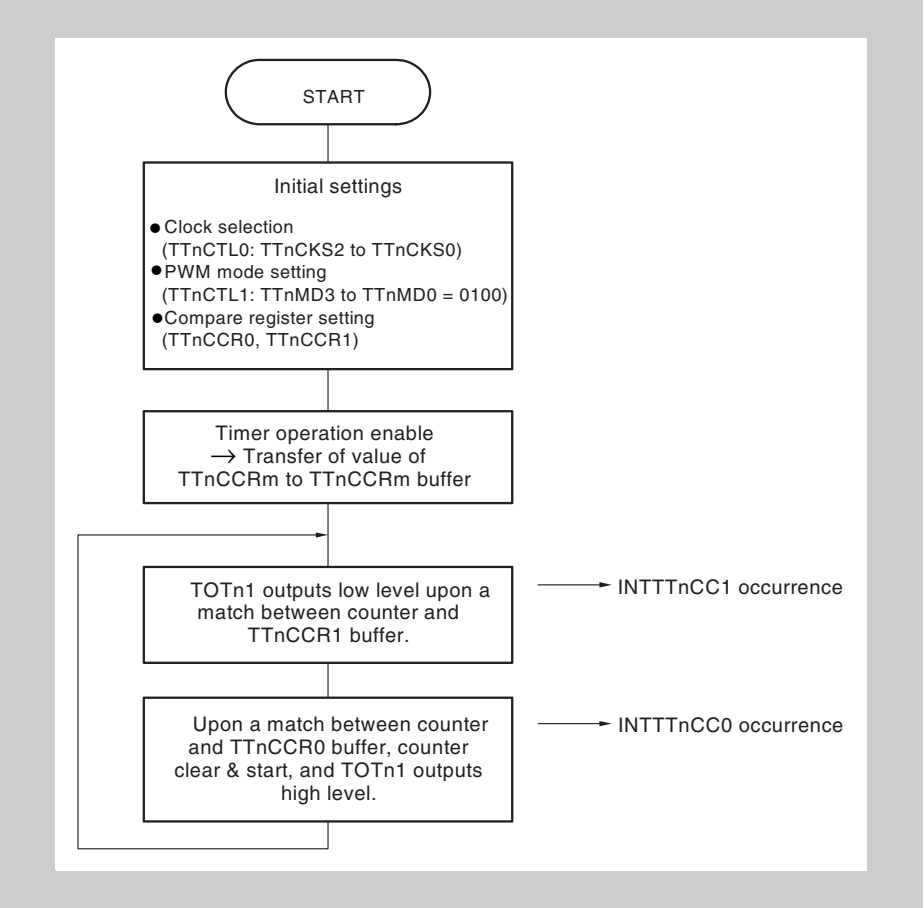


Figure 18-17 Basic operation mode in PWM mode (1/2)
When values of TTnCCR0 and TTnCCR1 registers are rewritten during timer operation

Note m = 0, 1

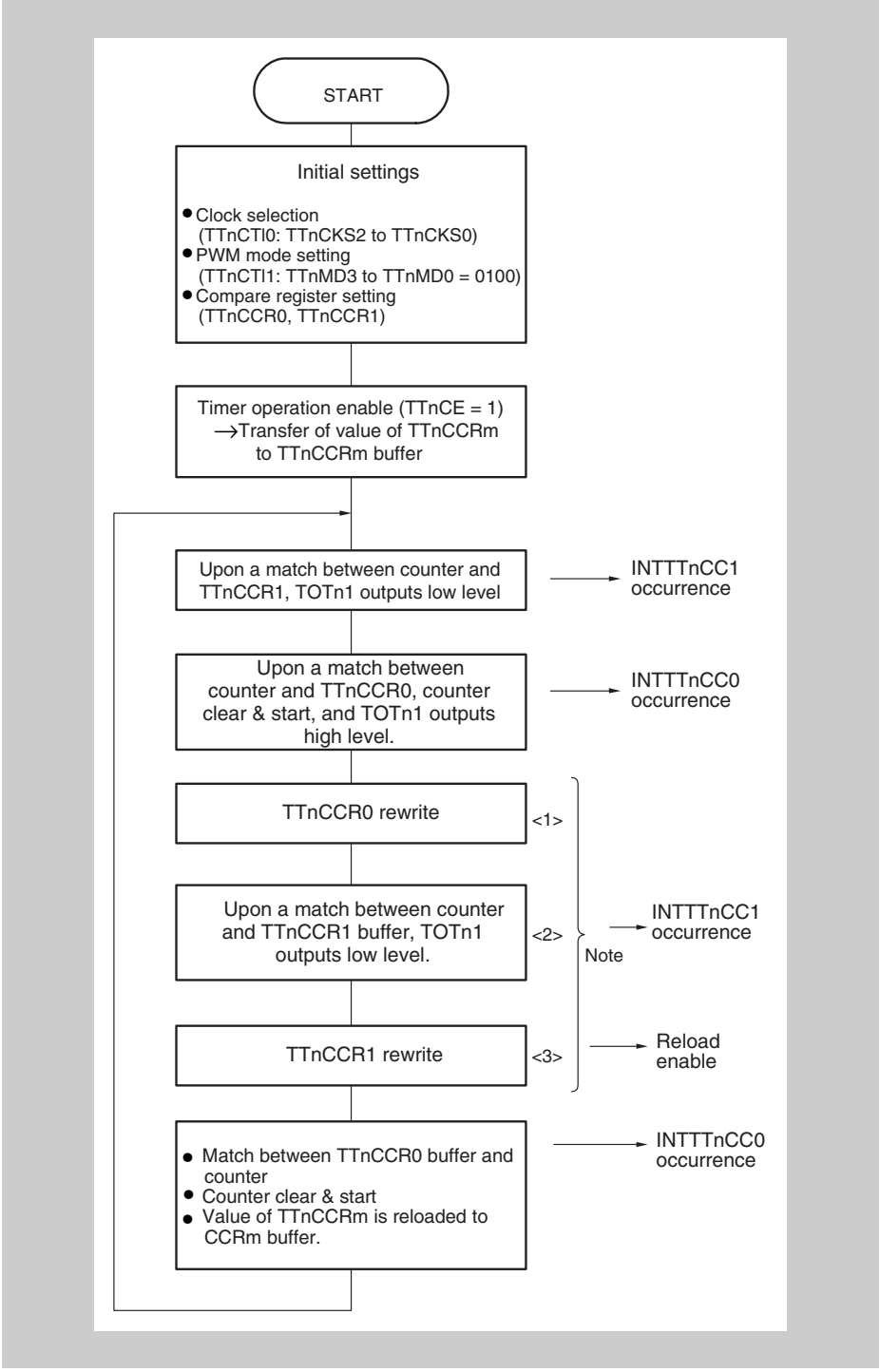


Figure 18-18 Basic operation flow in PWM mode (2/2)
When values of TTnCCR0 and TTnCCR1 registers are rewritten during timer operation

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

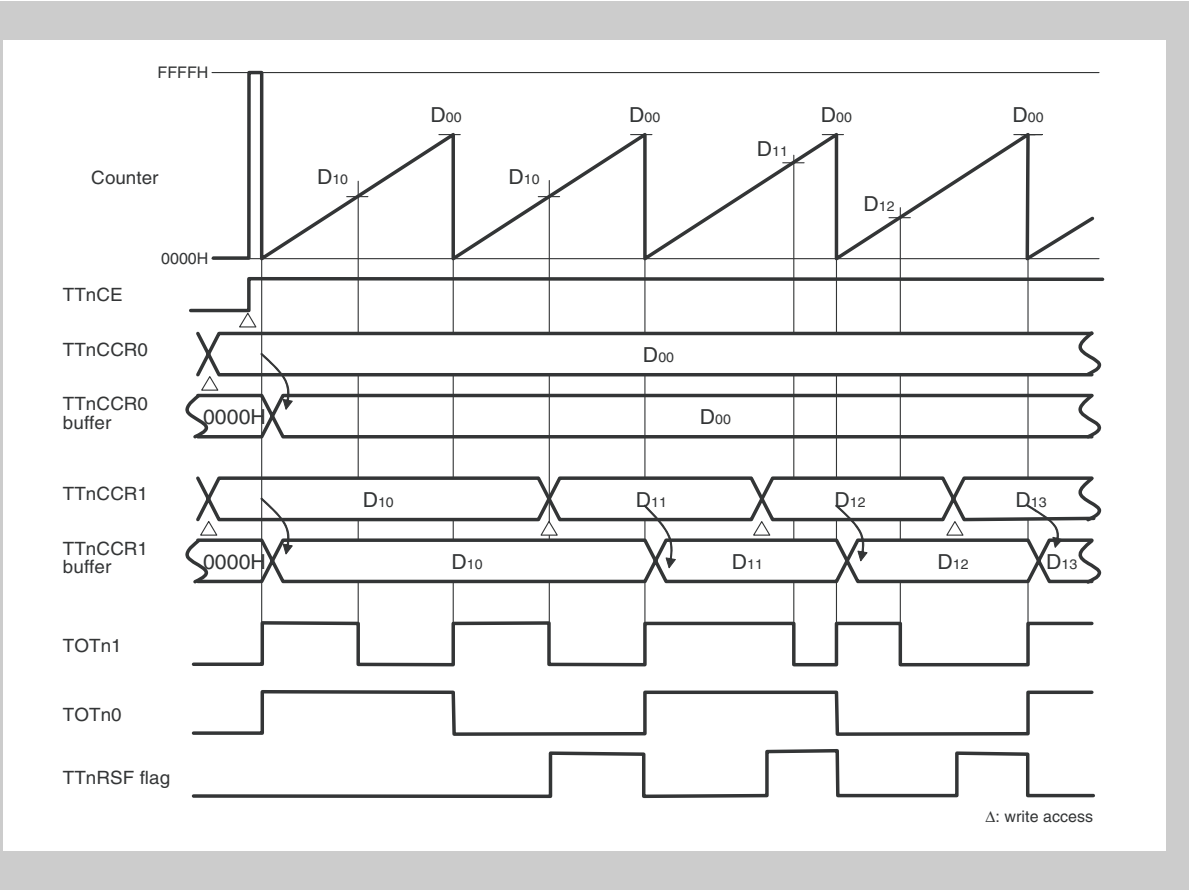


Figure 18-19 Basic operation timing in PWM mode (1/2)
When only value of TTnCCR1 is rewritten, and TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note

1. D_{00}: Setting value of TTnCCR0 register (0000H to FFFFH)
   D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TTnCCR1 register (0000H to FFFFH)
2. TOTn1 (PWM) duty = (setting value of TTnCCR1 register) × (count clock cycle)
   TOTn1 (PWM) cycle = (setting value of TTnCCR0 register + 1) × (count clock cycle)
3. TOTn0 is toggled immediately following counter start and at (setting value of TTnCCR0 register + 1) × (count clock cycle)

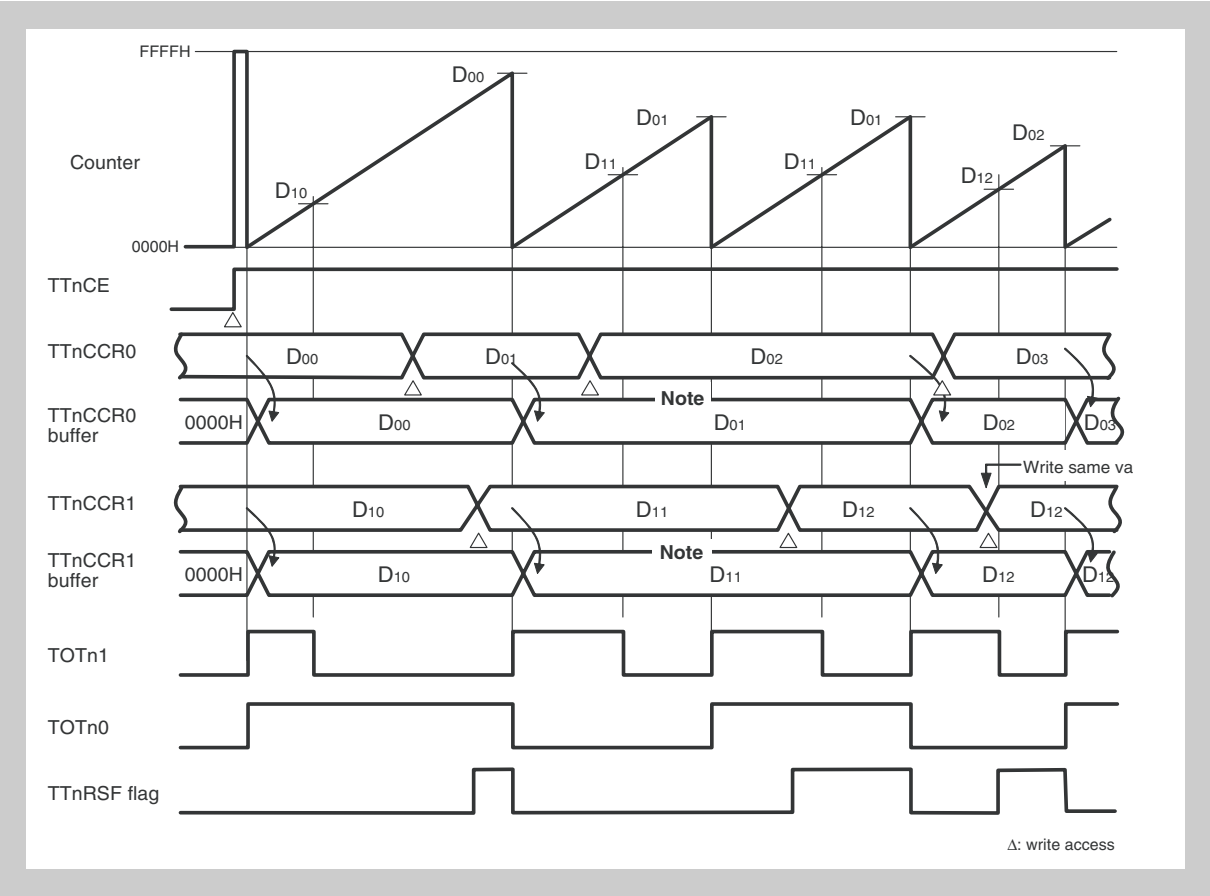


Figure 18-20 Basic operation timing in PWM mode (2/2)
When values of TTnCCR0 and TTnCCR1 register are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note 1. The TTnCCR1 register was not written to, so transfer to the TTnCCR0 buffer register was not performed. Held until the next reload timing.

2. D_{00}, D_{01}, D_{02}, D_{03}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TTnCCR1 register (0000H to FFFFH)

3. The TOTn0 and TOTn1 pins become high level at timer count start.

18.6.6 Free-running mode

The operation timing of the free-running mode is shown below.

The operation for bits TTnCCS1 and TTnCCS0 of register TTnOPT0 is specified.

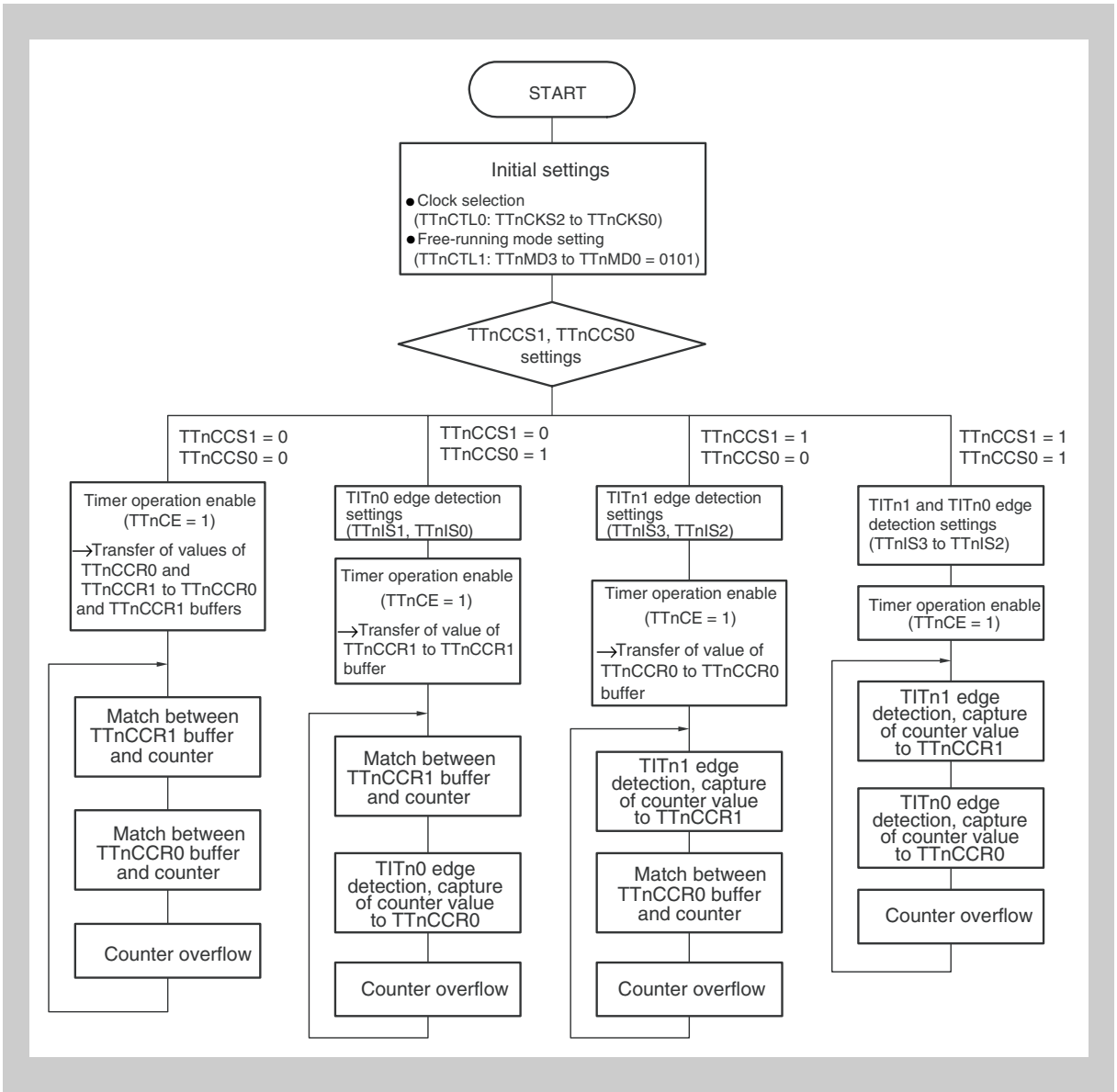


Figure 18-21 Basic operation flow in free-running mode

(1) Compare function (TTnCCS1 = 0, TTnCCS0 = 0)

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTTnOV) is output when the counter value changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. Moreover, during count operation, a compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 buffer register, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 buffer register.

The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of the TTnCE bit.

The TOTn0 and TOTn1 pins are toggle output controlled when bits register TTnOE0 and TTnOE1 of the TTnIOC0 register are set to 1.

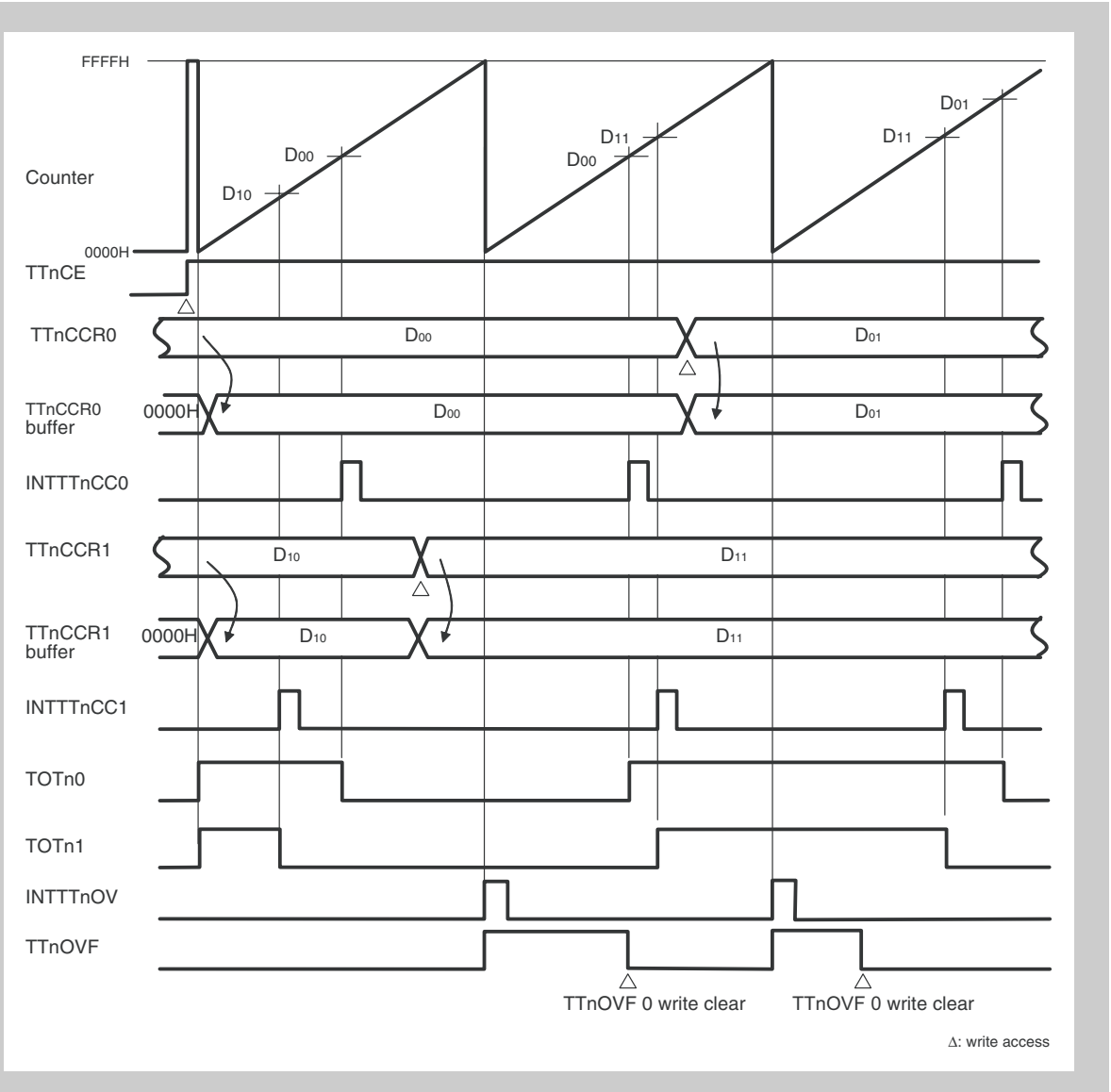


Figure 18-22 Basic operation timing in free-running mode (compare function) When values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note 1. D_{00}, D_{01}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Setting values of TTnCCR1 register (0000H to FFFFH)

2. TOTn0 (toggle) width = (setting value of TTnCCR0 register + 1) × (count clock cycle)

3. TOTn1 (toggle) width = (setting value of TTnCCR1 register + 1) × (count clock cycle)

4. Pins TOTn0 and TOTn1 become high level at count start.

(2) Capture function (TTnCCS1 = 1, TTnCCS0 = 1)

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTTnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. When, during count operation, the counter value is captured to the TTnCCR0 and TTnCCR1 registers through detection of the valid edge of capture input (TITn1, TITn0), a capture interrupt (INTTTnCC0, INTTTnCC1) is output.

Regarding capture in the vicinity of overflow (FFFFH), judgment is possible with the overflow flag (TTnOVF). However, judgment with the TTnOVF flag is not possible when the capture trigger interval is such that it includes two overflow occurrences (2 or more free-running cycles).

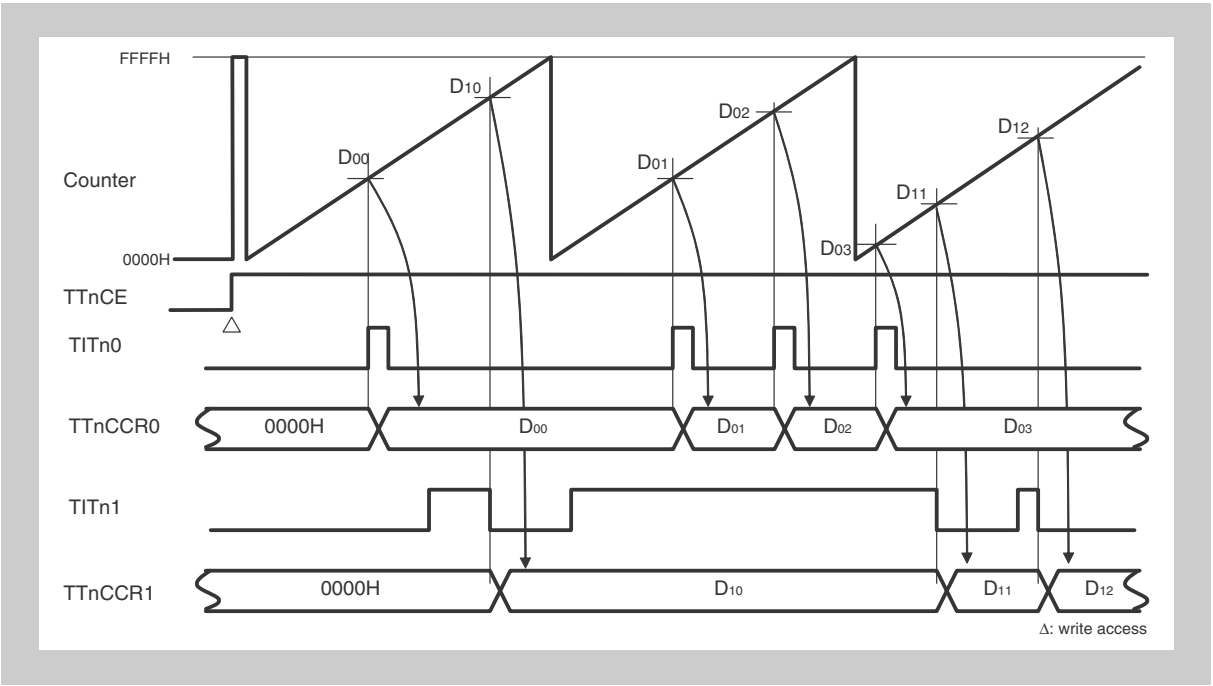


Figure 18-23 Basic operation timing in free-running mode (capture function) When TOTn0, TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0, 1 = 0)

Note
1. D_{00}, D_{01}: Values captured to TTnCCR0 register (0000H to FFFFH)
   D_{10}, D_{11}: Values captured to TTnCCR1 register (0000H to FFFFH)
2. TITn0: Setting to rising edge detection (TTnIOC1 register bits TTnIS1, TTnIS0 = 01)
   TITn1: Setting to falling edge detection (TTnIOC1 register bits TTnIS3, TTnIS2 = 10)

(3) Compare/capture function (TTnCCS1 = 0, TTnCCS0 = 1)

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH, an overflow interrupt (INTTTnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. The TTnCCR1 register is used as a compare register, and as the interval function upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output. Since the TTnCCR0 register is set to the capture function, the TOTn0 pin cannot be controlled even when TTnIOC0 register bit TTnOE0 is set to 1.

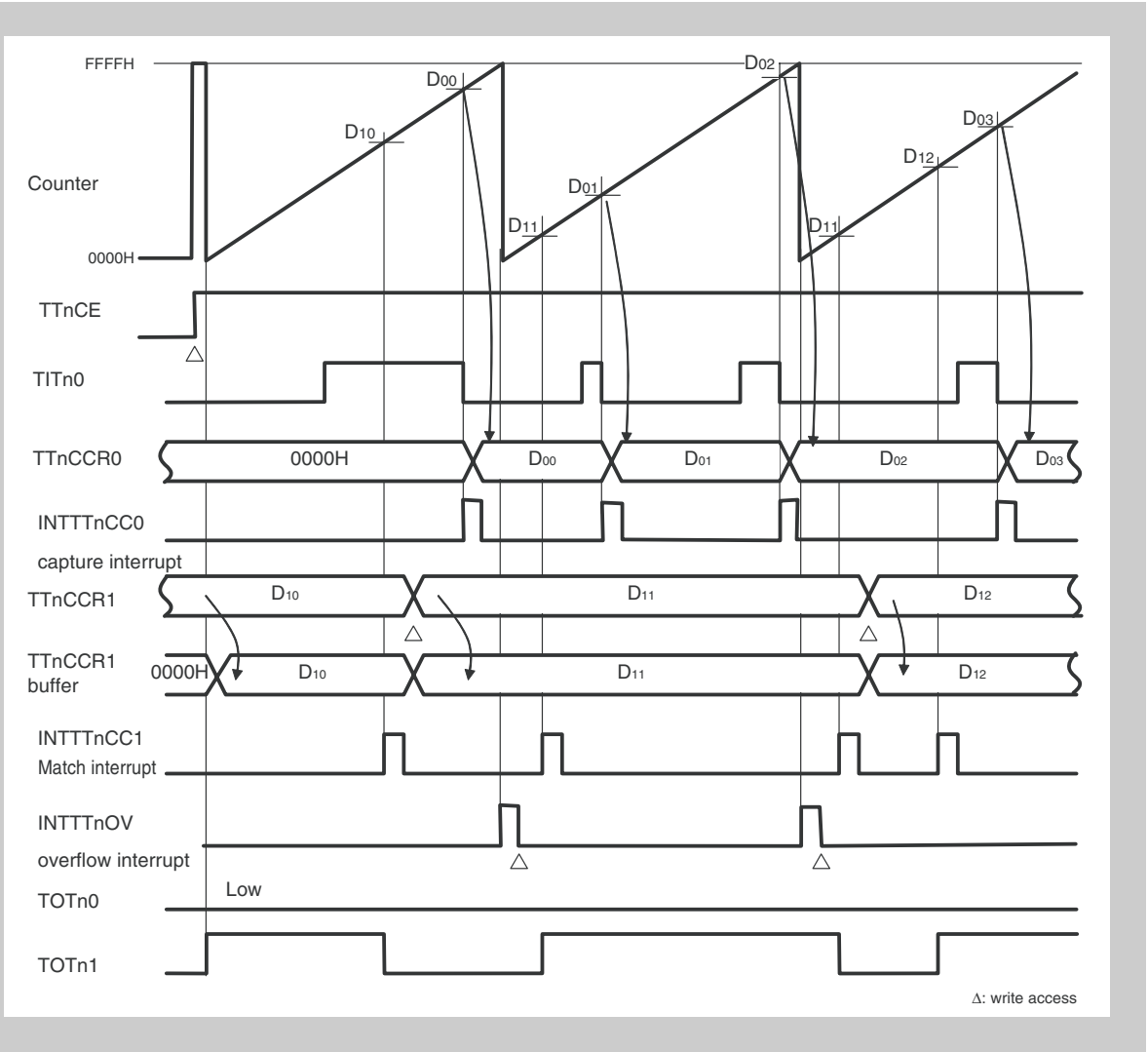


Figure 18-24 Basic operation timing in free-running mode (compare/capture function) When value of TTnCCR1 is rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note 1. D_{00}, D_{01}: Setting values of TTnCCR1 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}, D_{14}, D_{15}: Values captured to TTnCCR0 register (0000H to FFFFH)

2. TITn0: Setting to rising edge detection (TTnIOC1 register bits TtnIS1, TtnIS0 = 11)

(4) Overflow flag

When, in the free-running mode, the counter overflows from FFFFH to 0000H, the overflow flag (TTnOVF) is set to "1", and an overflow interrupt (INTTTnOV) is output.

The overflow flag is cleared through 0 write from the CPU.

(The overflow flag is not cleared by just being read.)

18.6.7 Pulse width measurement mode

In the pulse width measurement mode, counting is performed in the free-running mode. The counter value is saved to the TTnCCR0 register, and the counter is cleared to 0000H. As a result, the external input pulse width can be measured. However, when measuring a long pulse width that exceeds counter overflow, perform judgment with the overflow flag. Measurement of pulses during which overflow occurs twice or more is not possible, so adjust the counter's operating frequency. Even in the case of TITn1 pin edge detection, pulse width measurement can be similarly performed by using the TTnCCR1 register.

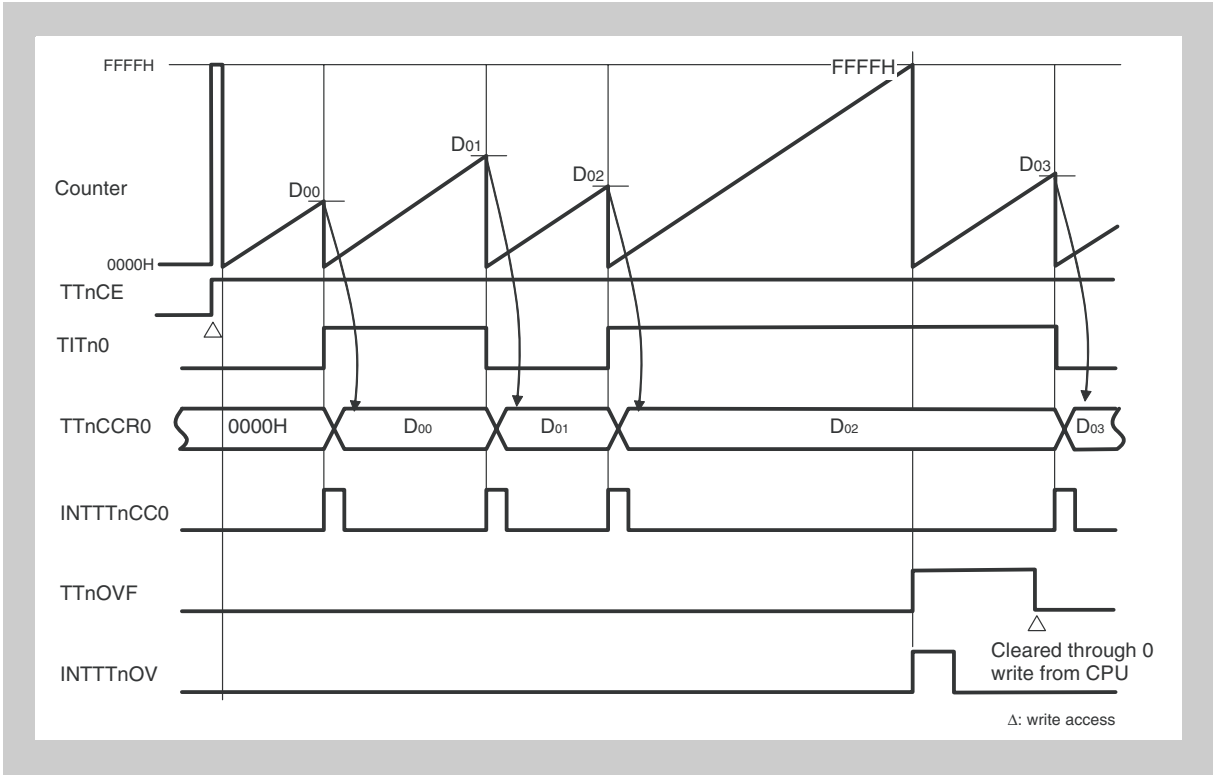


Figure 18-25 Basic operation timing in pulse width measurement mode (TTnOE0, 1 = 0, TTnOL0, 1 = 0)

Note 1. D_{00}, D_{01}, D_{02}, D_{03}: Values captured to TTnCCR0 register (0000H to FFFFH)

2. TITn0: Setting to rising edge/falling edge (both edges) detection (TTnIOC1 register bits TtnIS1, TtnIS0 = 11)

18.6.8 Triangular wave PWM mode

In the triangular wave PWM mode, similarly to in the PWM mode, when the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, variable duty and cycle type triangular wave PWM output is performed from pin TOTn1. The TOTn0 pin is toggle output upon a match with the TTnCCR0 buffer register and upon counter underflow. Upon a match between the counter and TTnCCR0 register during count operation, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output. Moreover, upon counter underflow, an overflow interrupt (INTTTnOV) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TTnCCR0 buffer register. However, since the next reload timing becomes valid when the TTnCCR1 register is written to, write the same value to the TTnCCR1 register even when wishing to rewrite only the value of the TTnCCR0 register. Reloading is not performed if only the TTnCCR0 register is rewritten. The reload timing is the underflow timing.

In the triangular wave PWM mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

Note In the triangular wave PWM mode, set the TTnCCR0 register to a value of 0 ≤ TTnCCR0 ≤ FFFEH.

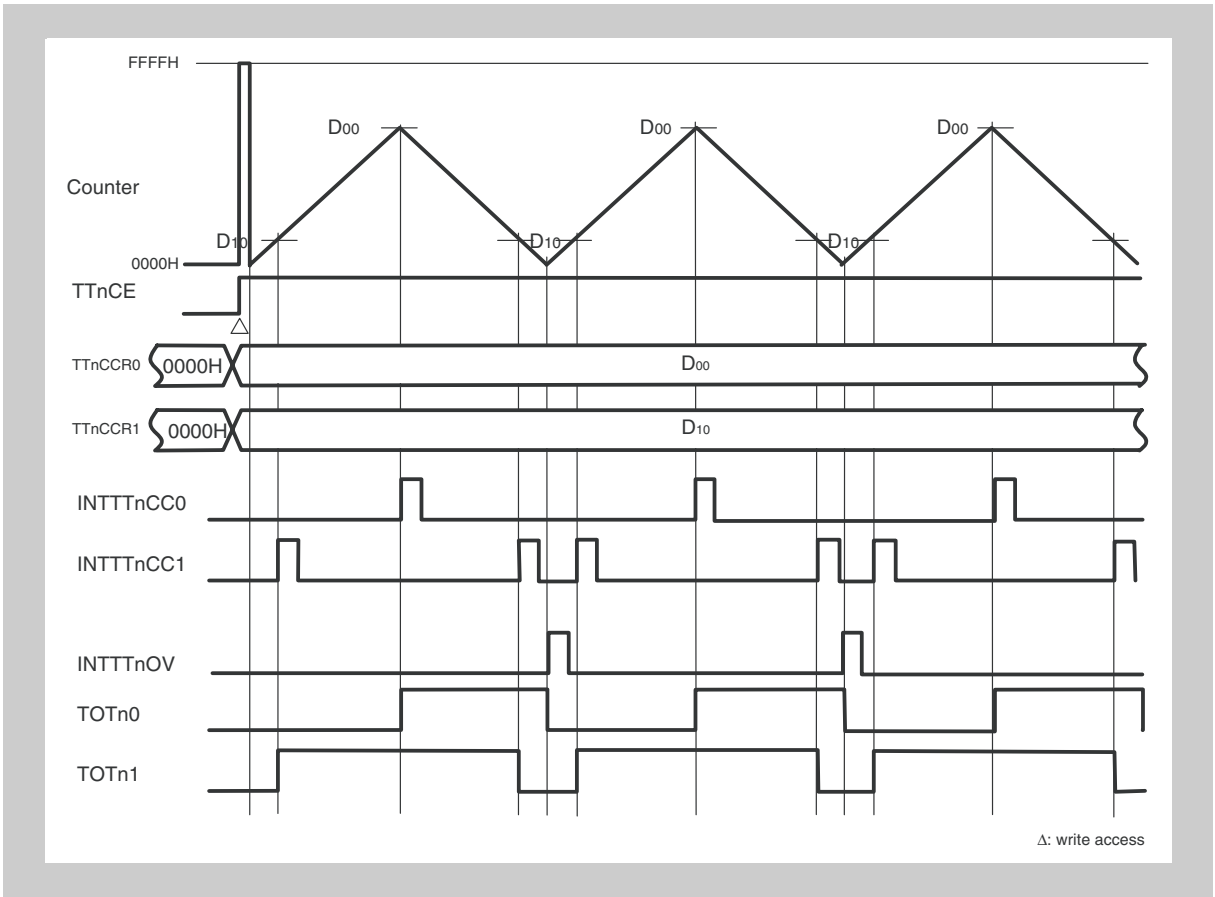


Figure 18-26 Basic operation timing in triangular wave pwm mode When TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

18.6.9 Encoder count function

Three encoder count function modes are provided, one for each capture compare function.

Mode	TTnCCR0 register	TTnCCR1 register
Encoder compare mode	Compare only	Compare only
Encoder capture mode	Capture only	Capture only
Encoder capture compare mode	Compare only	Capture only

(1) Counter up/down control

Counter up/down control is performed and the counter is operated according to the phase of signals TENCTn0 and TENCTn1 from the encoder and the set conditions of bits TTnUDS1 and TTnUDS0 of the TTnCTL2 register.

(2) Basic operation

To use the TTnCCR0 and TTnCCR1 registers are compare-only registers, enable rewrite during timer operation.

The rewrite method is anytime write.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 register. A compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 register.

To use the TTnCCR0 and TTnCCR1 registers are capture-only registers, save the counter value to the capture registers (TTnCCR0, TTnCCR1) through detection of the edges of pins TITn0 and TITn1. Specification of the detection of the edges of pins TITn0 and TITn1 is done with bits TTnIS3 to TTnIS0 of the TTnIOC1 register. Pin TOTn0 outputs the count status.

If the TTnCCR1 register is a compare-only register, pin TOTn1 is toggled upon a match between the counter and the TTnCCR1 register.

If the TTnCCR1 register is a capture-only register, pin TOTn1 becomes the level set with bit TTnOL1 of the TTnIOC0 register.

If TTnOL1 = 0, pin TOTn1 outputs a low level, and if TTnOL1 = 1, pin TOTn1 outputs a high level.

(3) Counter clear operation

Clearing of the counter to 0000H is performed under the following conditions in each mode.

Clear condition	Encoder compare	Encoder capture	Encoder capture compare
Method whereby counter is cleared to 0000H upon match with compare register (setting of TTnCTL2 register bits TTnECM1, TTnECM0)	√	-	√ (TTnECM0[Note])
Method whereby counter is cleared to 0000H upon detection of edge of pin TECR0 (setting of bits TTnECS1, TTnECS0 when TTnIOC3 register bit TTnSCE = 0)	√	√	√
Method whereby counter is cleared to 0000H by special clear function of encoder (setting of bits TTnZCL, TTnBCL, TTnACL when TTnIOC3 register bit TTnSCE = 1)	√	√	√

Note Since the TTnCCR1 register is a capture-only register, the setting of TTnCTL2 register bit TTnECM1 are invalid.

(4) Control through TTnCTL2 register

The settings of the TTnCTL2 register in the encoder compare mode, encoder capture mode, and encoder capture/compare mode (TTnMD3 to TTnMD0 = 1000B, 1001B, 1010B) are as follows.

TTnMD3 to 0	TTnUDS1 to 0	TTnECM1	TTnECM0	TTnLDE	Clear	Load
1000B	All settings possible 00B 01B 10B 11B	0	0	0	-	–
				1		√
			1	0	TTnCCR0	–
				1		•
		1	0	Invalid	TTnCCR1	–
			1	Invalid	TTnCCR0 TTnCCR1	–
1001B		Invalid			–	–
1010B		0	0	0	–	–
				1		√
			1	0	TTnCCR0	-
				1		•
		1	0	Invalid	–	–
			1	Invalid	TTnCCR0	–

- In the case of bits TTnUDS1 and TTnUDS0, up/down judgment control is performed for the phase input from pins TENCTn0 and TENCTn1.
- In the case of bits TTnECM1 and TTnECM0, counter clear control is performed upon a match between the counter value and the compare setting value.

  Bits TTnECM1 and TTnECM0 are valid in modes where the TTnCCR0 or TTnCCR1 register is used as a compare-only register.
  These bits are invalid in modes where the TTnCCR0 or TTnCCR1 register is used as a capture-only register.
- The TTnLDE bit controls the function to load to the counter the setting value of the TTnCCR0 register upon occurrence of counter underflow.
  Bit TTnLDE is valid only when the TTnECm bit setting is 00B, 01B, in a mode where the TTnCCR0 or TTnCCR1 register is used as a compare-only register.

  In the case of all other settings, bit TTnLDE is invalid even if manipulated.

  As an example of the use of the encoder count function, counter operation becomes possible between the setting values of registers 0000H to TTnCCR0 by using the counter load functions (TTnLDE = 1) indicated with "•" in the table, and the function for clearing the counter to 0000H in case the count operation following a match with the TTnCCR0 buffer register is up count (TTnECM0 = 1). (Refer to *"Counter load function for TTnCCR0 register setting value upon underflow (bit TTnLDE of register TTnCTL2))" on page 1086*).

(a) Up/down count selection specification (TTnCTL2 register bits TTnUDS1, TTnUDS0)

Counter up/down is judged according to the settings of bits TTnUDS1 and TTnUDS0, and the phases input from pins TENCTn0 and TENCTn1.

Bits TTnUDS1 and TTnUDS0 are valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.

1. TTnCTL2: TTnUDS1, 0 = 00B (count judgment mode 1)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Rising edge	High level	Down
Falling edge		
Both edges		
Rising edge	Low level	Up
Falling edge		
Both edges		

Operation example:

TTnIOC3: TTnEIS3 to 2	TENCTn1 pin input	Edge detection specification invalid
TTnIOC3: TTnEIS1 to 0 = 10B	TENCTn0 pin input	Rising edge detection

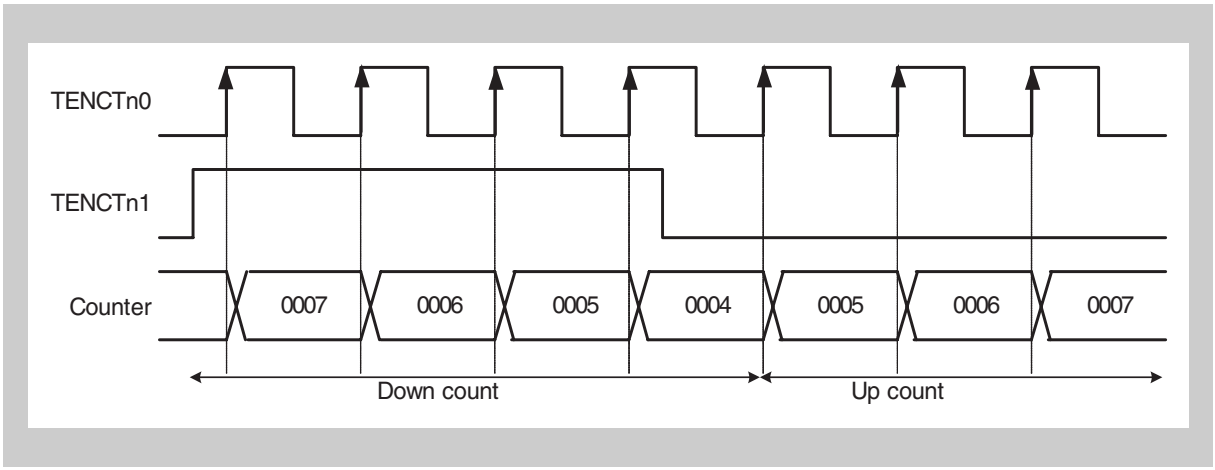


Figure 18-27 Encoder count function up/down count selection specification timings (1/6) - timing 1

Note Counting is performed when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

2. TTnCTL2: TTnUDS1, 0 = 01B (count judgment mode 2)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Low level	Rising edge	Down
	Falling edge	
	Both edges	
High level	Rising edge	
	Falling edge	
	Both edges	
Rising edge	Low level	Up
Falling edge		
Both edges		
Rising edge	High level	
Falling edge		
Both edges		
Simultaneous pin TENCTn0/TENCTn1 inputs		Hold

Operation example:

TTnIOC3: TTnEIS3, 2 = 10B TENCTn1 pin input Rising edge detection

TTnIOC3: TTnEIS1, 0 = 10B TENCTn0 pin input Rising edge detection

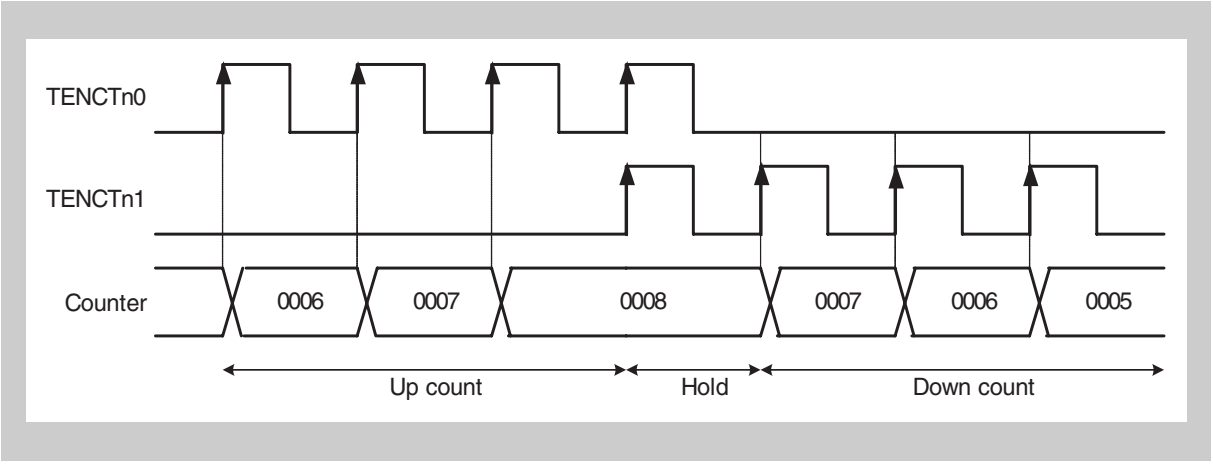


Figure 18-28 Encoder count function up/down count selection specification timings (2/6) - timing 2

Note The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

3. TTnCTL2: TTnUDS1, 0 = 10B (count judgment mode 3)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Low level	Falling edge	Hold
Rising edge	Low level	Down
High level	Rising edge	Hold
Falling edge	High level	
Rising edge	High level	
High level	Falling edge	
Falling edge	Low level	Up
Low level	Rising edge	Hold
Rising edge	Rising edge	Hold
Falling edge	Rising edge	
Rising edge	Falling edge	Down
Falling edge	Falling edge	Up

Operation example:

TTnIOC3: TTnEIS3 to 0	Pins TENCTn1, TENCTn0	Edge detection specification invalid

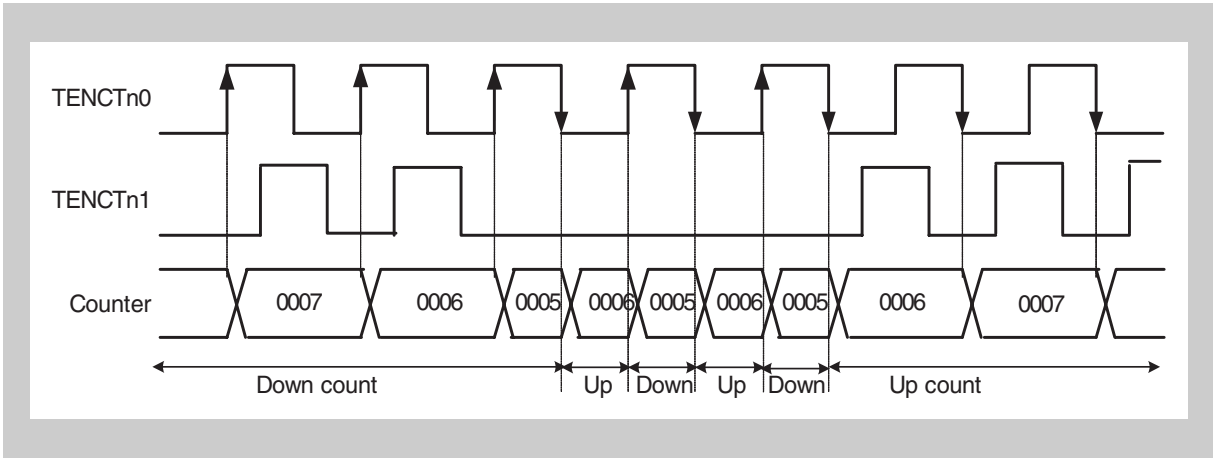


Figure 18-29 Encoder count function up/down count selection specification timings (3/6) - timing 3

4. TTnCTL2: TTnUDS1, 0 = 11B (count judgment mode 4)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Low level	Falling edge	Down
Rising edge	Low level	
High level	Rising edge	
Falling edge	High level	
Rising edge	High level	Up
High level	Falling edge	
Falling edge	Low level	
Low level	Rising edge	
Simultaneous pin TENCTn0/TENCTn1 inputs		Hold

Operation example 1:

TTnIOC2: TTnEIS3 to 0	Pins TENCTn1, TENCTn0	Edge detection specification invalid

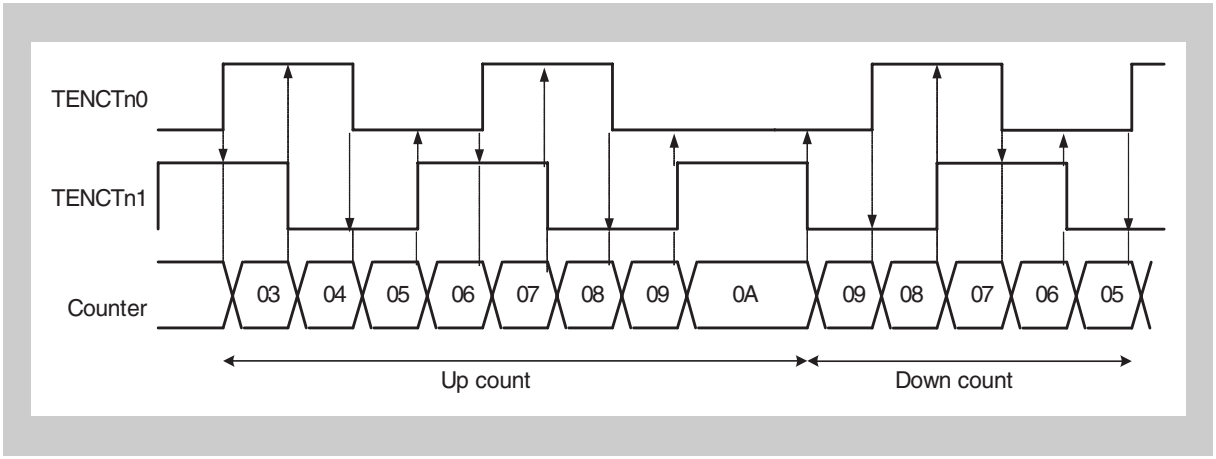


Figure 18-30 Encoder count function up/down count selection specification timings (4/6) - timing 4

Operation example 2:

TTnIOC2: TTnEIS3 to 0	Pins TENCTn1, TENCTn0	Edge detection specification invalid

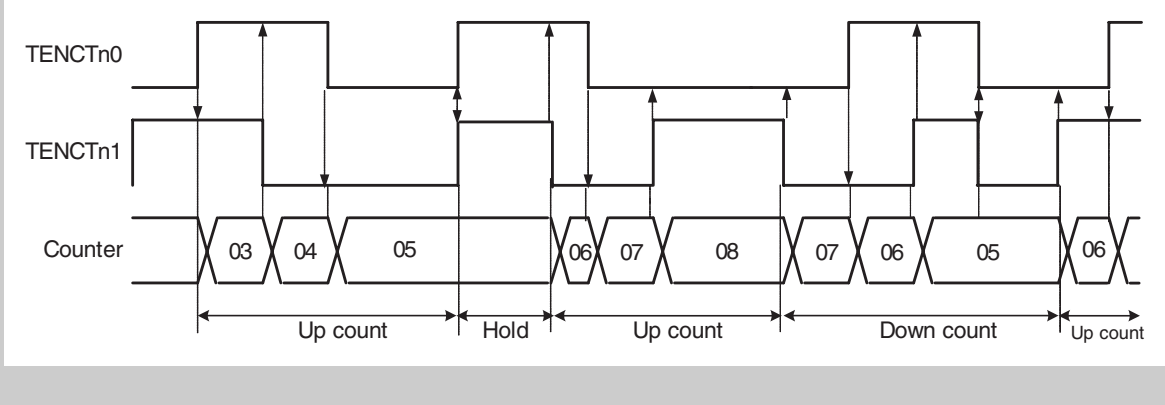


Figure 18-31 Encoder count function up/down count selection specification timings (5/6) - timing 5

Note The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

(b) Counter clear condition setting upon match between counter value and compare setting value (TTnCTL2 register bits TTnECM1, TTnECM0)

Counter operation is performed according to the setting values of these bits upon a match between the counter value and the compare setting value.

1. TTnECM1, 0 = 00B
   Counter clear is not performed upon a match between the counter and compare values.

2. TTnECM1, 0 = 01B
   Counter clear is performed upon a match between the counter and the TTnCCR0 register.

Next count operation	Description
Up count	Clear counter to 0000H.
Down count	Down count the counter value.

3. TTnECM1, 0 = 10B
   Operation is performed under the following conditions upon a match between the counter and TTnCCR1 register.

Next Count operation	Description
Up count	Up count the counter value.
Down count	Clear counter to 0000H.

4. TTnECM1, 0 = 11B
   – Operation is performed under the following conditions upon a match between the counter and TTnCCR0 register.

Next count operation	Description
Up count	Clear counter to 0000H.
Down count	Down count the counter value.

   – Operation is performed under the following conditions upon a match between the counter and TTnCCR1 register.

Next count operation	Description
Up count	Up count the counter value.
Down count	Clear counter to 0000H.

Caution In encoder compare mode (TTnMD3 to TTnMD0 bits = 1000B), or encoder capture-compare mode (TTnMD3 to TTnMD0 bits = 1010B), if the compare registers (TTnCCR0, TTnCCR1) are set to the same value of TTnTCW register when TTnECC bit = 0, the timer cannot perform the comparison with the compare registers (TTnCCR0, TTnCCR1) and TTnTCW register (which is the start value of TTnCNT). In this case the “encoder clear mode on match of counter and compare register” does not work at the start timing (TTnECM0 = 1, and/or TTnECM1 = 1).

(c) Counter load function for TTnCCR0 register setting value upon underflow (bit TTnLDE of register TTnCTL2))

The setting value of the TTnCCR0 register can be loaded to the counter upon counter underflow, by setting TTnLDE = 1.

Bit TTnLDE is valid in the encoder compare mode and encoder capture compare mode.

Count operation between 0000H and setting value of TTnCCR0 register setting:

Set TTnLDE = 1, TTnECM1, 0 = 01B and perform count operation. When TTnECM0 = 1, the counter is cleared to 0000H if the next count following a match between the counter and TTnCCR0 register is up count.

When TTnLDE = 1, the setting value of the TTnCCR0 register is loaded to the counter upon underflow.

Therefore, the setting value of the TTnCCR0 register is used as the maximum count value and count operation can be realized within "0000H-TTnCCR0 register setting values.

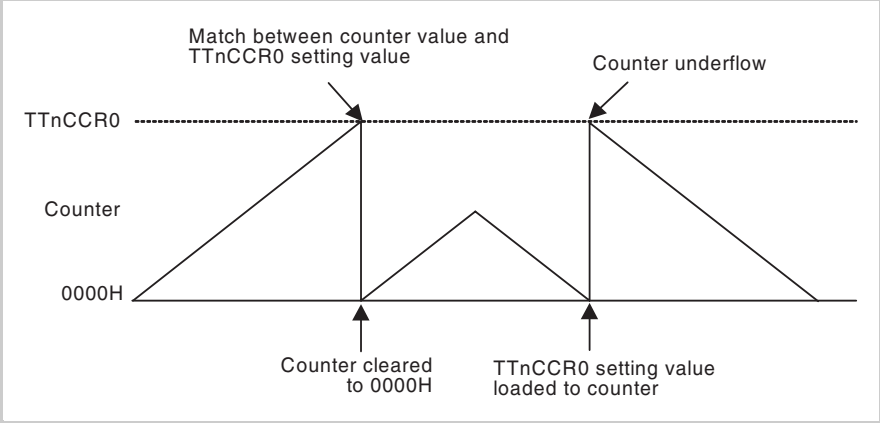


Figure 18-32 Encoder count function up/down count selection specification timings (6/6) - timing 6

(5) Counter clearing to 0000H through encoder clear input (pin TECRTn) (TTnIOC3 register bits TTnSCE, TTnECS1, TTnECS0)

There are two methods to clear the counter to 0000H through TECRTn pin input, and encoder clear input is controlled by bit TTnSCE. Bits TTnZCL, TTnBCL, TTnACL, TTnECS1, and TTnECS0 are controlled by the setting of bit TTnSCE.

These clear methods are valid in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.

TTnSCE	TTnZCL	TTnBCL	TTnACL	TTnECS1, 0	Method
0	Invalid	Invalid	Invalid	√	<1>
1	√	√	√	Invalid	<2>

1. Method to clear counter to 0000H through detection of valid edge of TECRTn pin input (TTnSCE = 0)

   When TTnSCE = 0, the counter is cleared to 0000H in synchronization with the internal operation clock upon detection of the valid edge set through TECRTn pin input edge detection specification. At this time, an encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 0, the setting of bits TTnZCL, TTnBCL, and TTnACL are invalid.

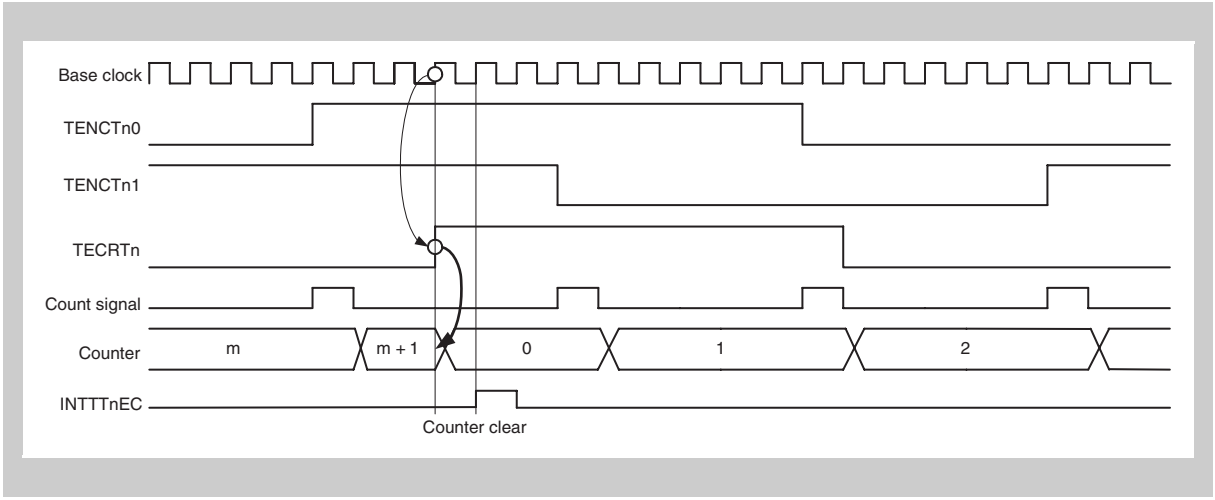


Figure 18-33 Counter clearing to 0000h through encoder clear input (pin tecrtn) timings (1/4)
When TTnSCE = 0, TTnECS1, 0 = 01B, TTnUDS = 11B are Set

2. Method to clear counter to 0000H through detection of level clear condition (TTnSCE = 1)

When TTnSCE = 1, the counter is cleared to 0000H according to the clear condition level of pins TECRTn, TENCTn1, and TENCTn0 set with bits TTnZCL, TTnBCL, and TTnACL. At this time, no encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 1, the settings of bits TTnECS1 and TTnECS0 are invalid.

Operation example:

When TTnSCE = 1, TTnCLA = 1, TTnCLB = 0, TTnCLZ = 1, TTnUDS = 11B are set

<Clear condition level>
TECRTn pin: High level, TENCTn1 pin: Low level,
TENCTn0 pin: High level

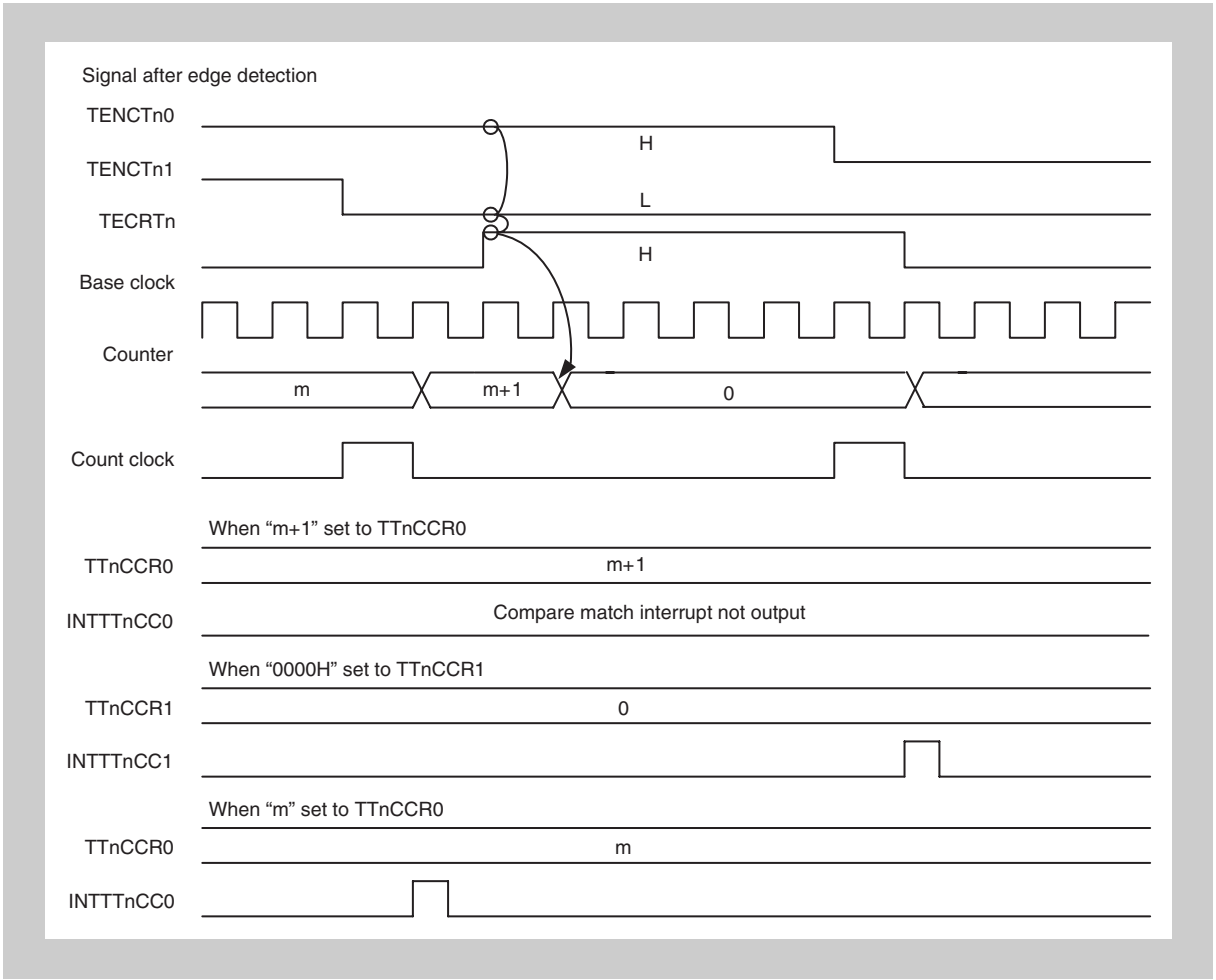


Figure 18-34 Counter clearing to 0000H through encoder clear input (pin TECRTn) timings (2/4)
When TECRTn pin input is delayed from TENCTn1 pin input during up count

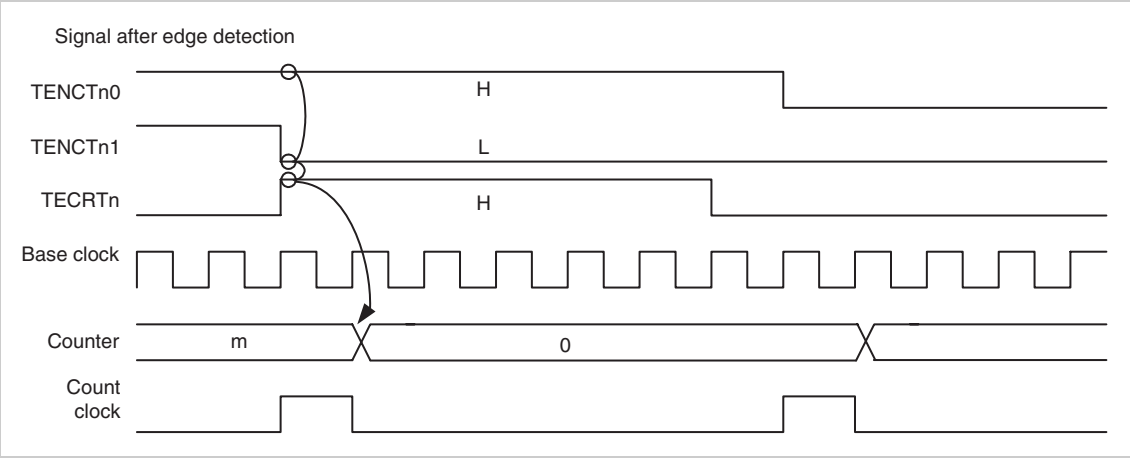


Figure 18-35 Counter clearing to 0000H through encoder clear input (pin TECRTn) timings (3/4)
When TECRTn pin input and TENCTn1 pin input occur simultaneously during up count

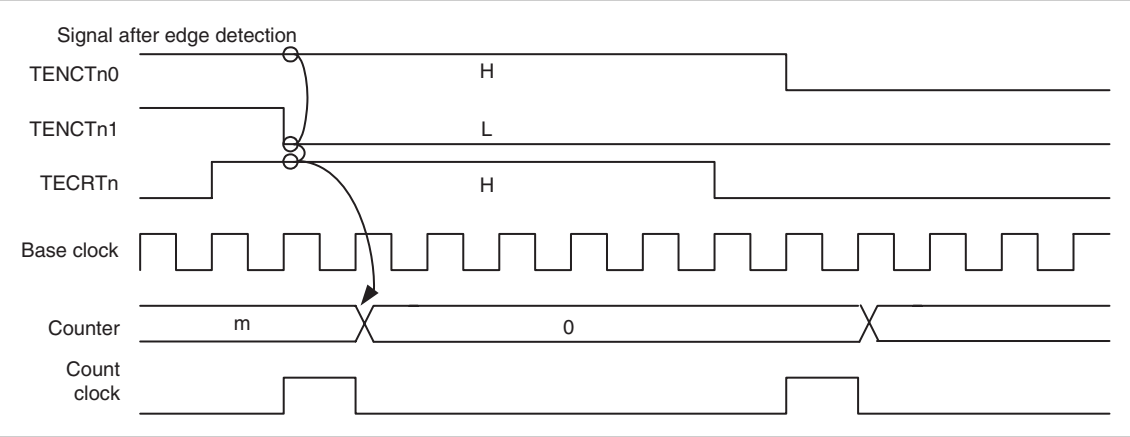


Figure 18-36 Counter clearing to 0000H through Encoder clear input (pin TECRTn) timings (3/4)
When TECRTn pin input occurs earlier than TENCTn1 pin input during up count

No miscount occurs due to TECRTn pin input delay because the clear condition is set according to the levels of pins TENCTn0, TENCTn1 and TECRTn, and the counter is cleared to 0000H upon clear condition detection.

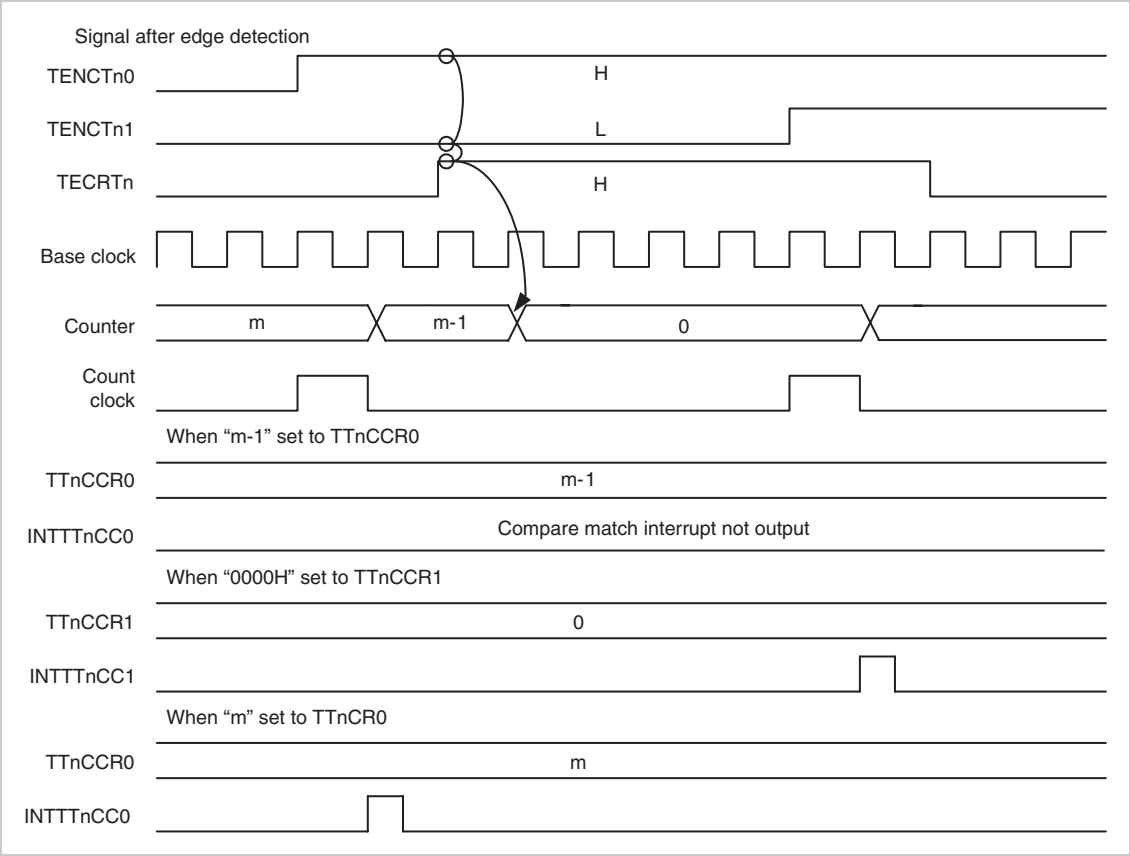


Figure 18-37 Counter clearing to 0000H through encoder clear input (pin TECRTn) timings (4/4)
When TECRTn pin input occurs later than TENCTn1 pin input during down count

No miscount occurs due to the TECRTn pin input delay during down count, similarly to during up count.

(6) Counter hold through bit TTnECC

By setting TTnCTL2 register bit TTnECC to 1, it is possible to switch to an encoder mode other than the currently operating mode during timer operation in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. The capture function/compare function of the TTnCCR0 and TTnCCR1 registers can be switched by switching the operation mode.

(a) Mode switching using bit TTnECC

To switch the encoder mode while holding the counter value, be sure to set TTnECC = 1 before setting TTnCE = 0.

Since the counter gets reset if TTnCE = 0 is set with TTnECC = 0 left unchanged, the counter value cannot be held in this case.

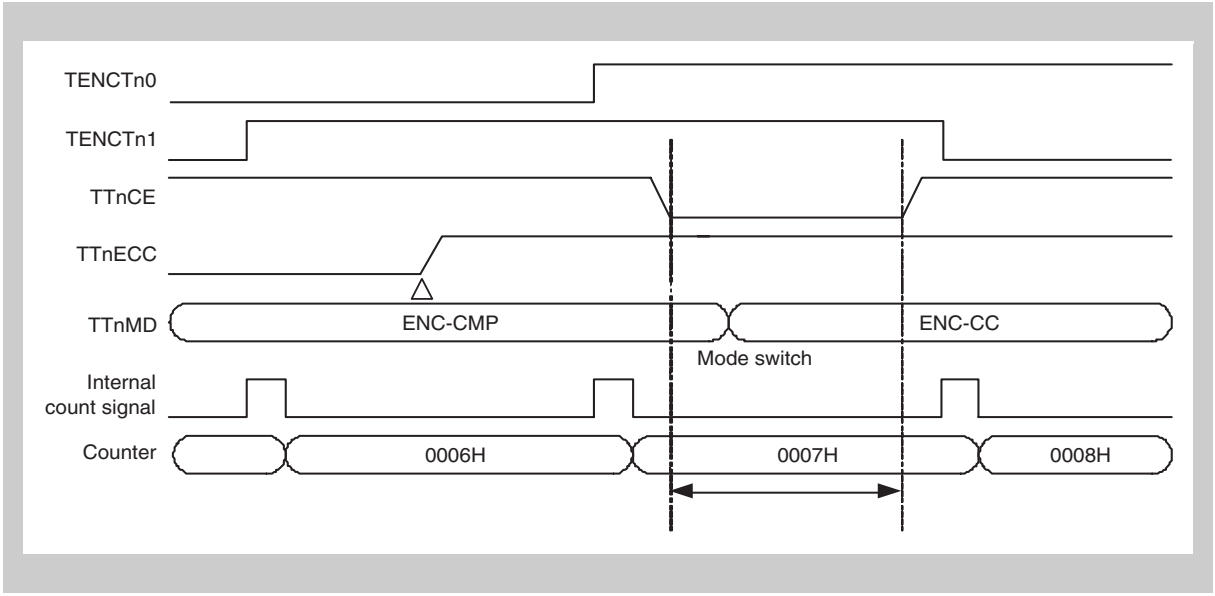


Figure 18-38 Counter hold through bit TTnECC timings (1/8) Mode switching using bit TTnECC

Caution To change the mode setting TTnCE = 0 once while the encoder itself is operating, perform initial setting processing within 1 cycle of the count clock. If switching is performed through bit TTnECC at the count signal output timing, miscount will occur.

(b) Initial counter operation through bit TTnECC setting

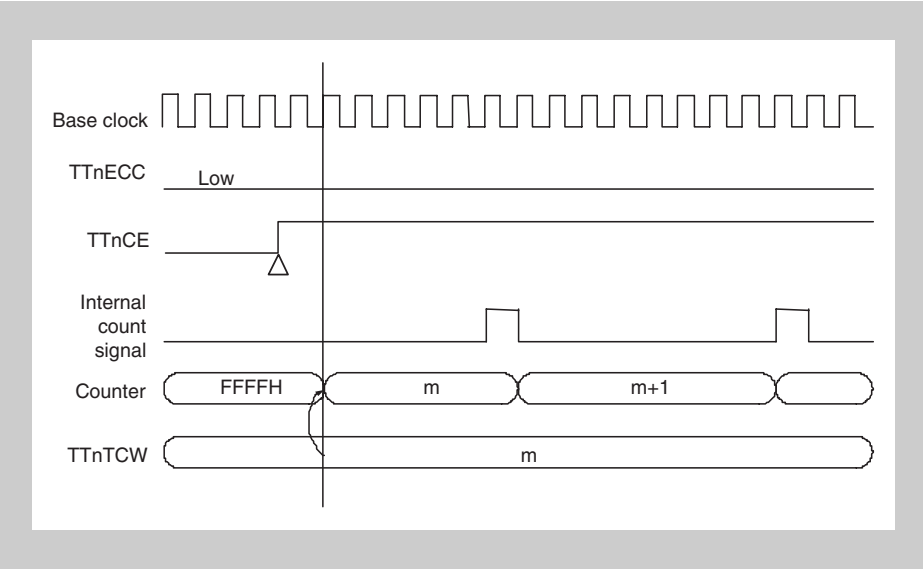


Figure 18-39 Counter hold through bit TTnECC timings (2/8)
Count operation when TTnECC = 0 is set

The setting value of the TTnTCW register is loaded to the counter and count operation is performed from the setting value of the TTnTCW register.

(Initial value 0000H of TTnTCW register)

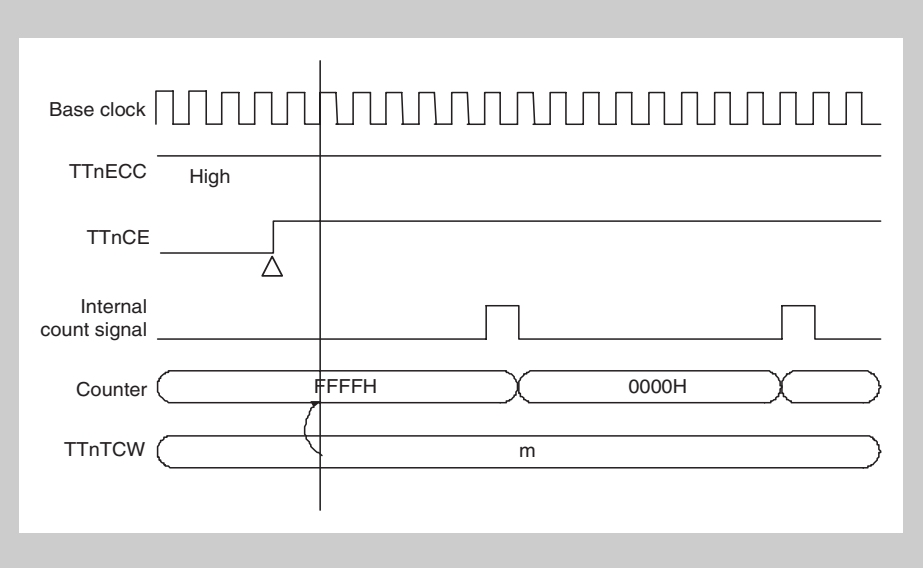


Figure 18-40 Counter hold through bit TTnECC timings (2/8)
Count operation when TTnECC = 1 is set

Since the setting value of the TTnTCW register is not loaded to the counter, the count operation is performed from initial value FFFFH.

As the initial operation, it is recommended to set TTnECC = 0 and load to the counter the value set to the TTnTCW register, then start the count operation.

(c) Bit TTnECC rewrite timing and its influence on counter

1. When setting value of bit TTnECC is rewritten 0 → 1 → 0 when TTnCE = 1

   Even if bit TTnECC rewrite is performed while TTnCE = 1, this has no influence on the counter operation.

   Judgment as whether to hold or reset the counter value is performed while TTnCE = 0.

   Moreover, judgment as to whether to load the setting value of the TTnTCW register to the counter is performed at the timing when the value of bit TTnCE changes from 0 to 1.

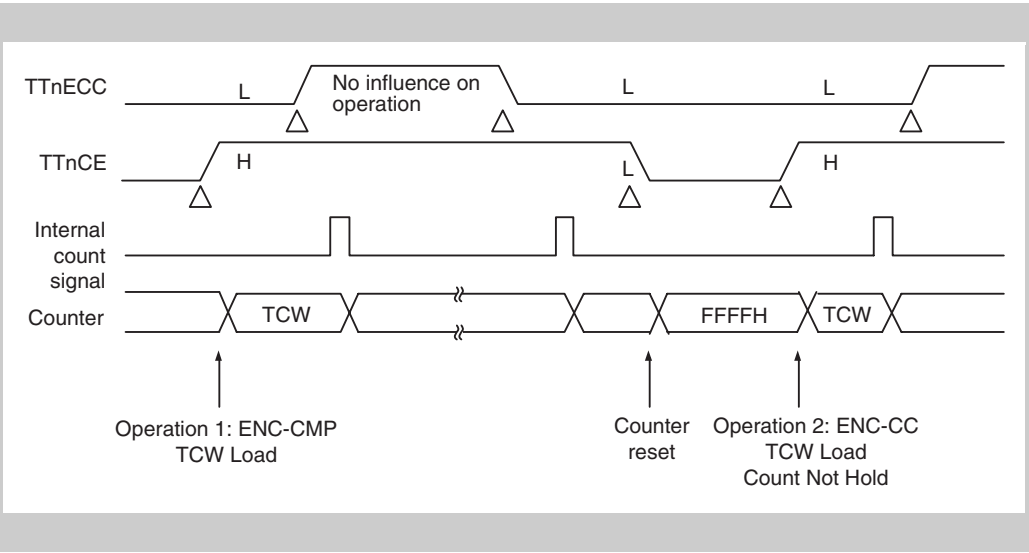


Figure 18-41 Counter hold through bit TTnECC timings (3/8)
When setting value of bit TTnECC is rewritten 0 → 1 → 0 when TTnCE = 1

2. When setting value of bit TTnECC is rewritten 1 → 0 → 1 while TTnCE = 0

   The counter is reset when the setting value of bit TTnECC is changed from 1 to 0 while TTnCE = 0

   Then, when TTnECC = 1 is set again and the value of bit TTnCE is changed from 0 to 1, counting restarts from the counter's initial value FFFFH, without the setting value of the TTnTCW being loaded to the counter.

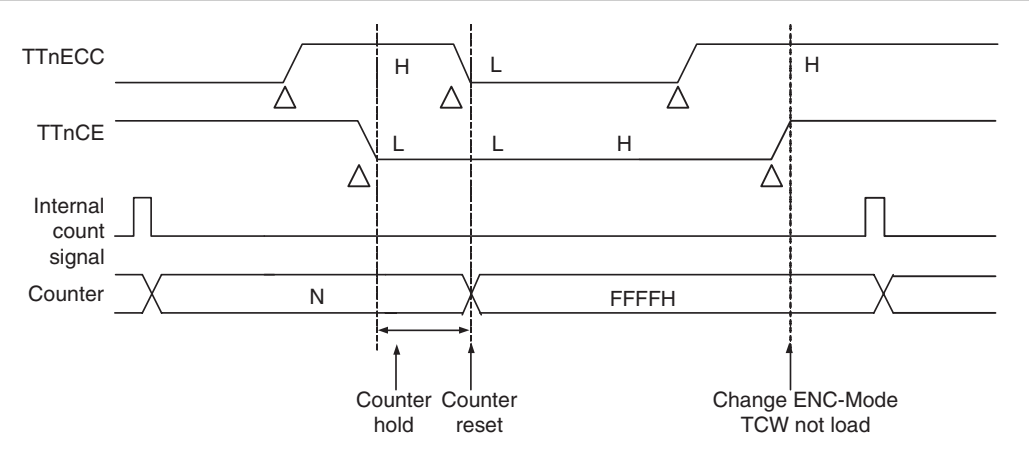


Figure 18-42 Counter hold through bit TTnECC timings (3/8) When setting value of bit TTnECC is rewritten 1 → 0 → 1 while TTnCE = 0

(d) Rewrite timing of bit TTnECC

When TTnCE = 0 and TTnECC = 0, setting TTnCE = 1 causes the setting value of the TTnTCW register to be loaded to the counter.

Perform rewrite of the TTnECC bit after the operation clock has become valid (after several clocks: TBD), following setting of TTnCE = 1.

If bit TTnECC is rewritten before the operation clock becomes valid, counting starts from FFFFH without loading the setting value of the TTnTCW register to the counter.

< Register setting conditions>

TTnCTL0: TTnMD3 to 0 = 1000B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 00B	Judgment of up/down count with count judgment mode 1
TTnCTL1: TTnECM1, 0 = 01B	Counter clear upon match between counter value and TTnCCR0 buffer register
TTnCTL1: TTnLDE = 1	Loading of setting value of TTnCCR0 register (p) upon underflow occurrence
TTnIOC3: TTnEIS1, 0 = 01B	Detection of rising edge of TENCTn0 and TENCTn1 pin
TTnIOC3: TTnSCE = 0, TTnECS1-0 = 00B	Valid edge detection clear (no edge specified)

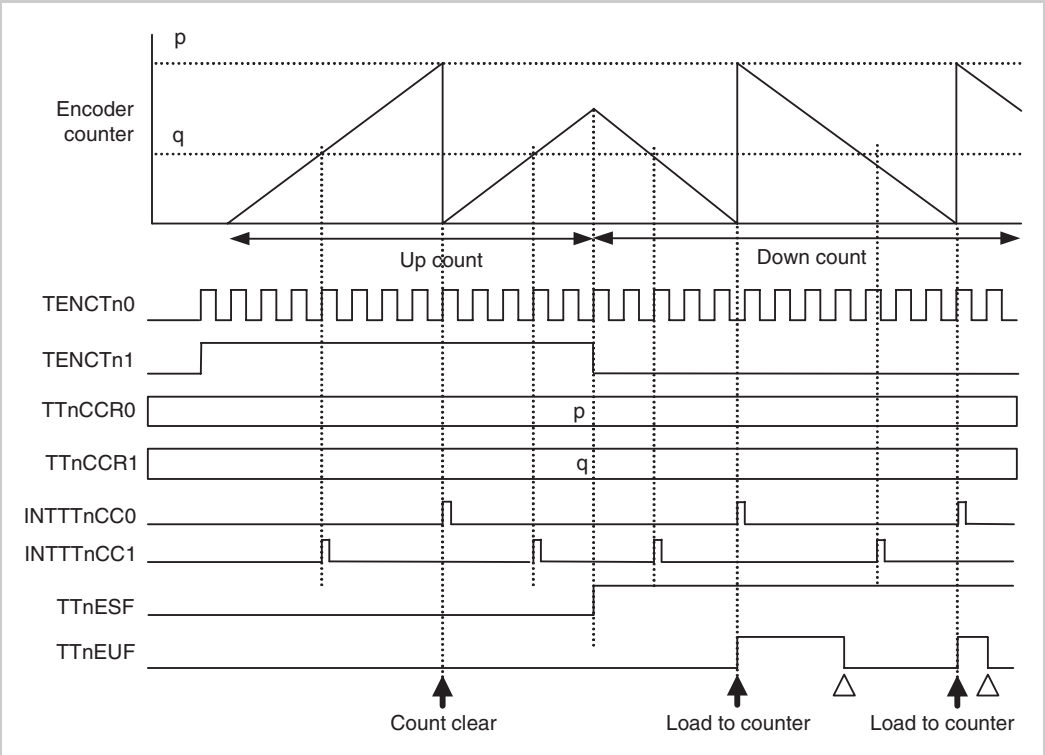


Figure 18-43 Counter hold through bit TTnECC timings (4/8)
Basic timing in encoder compare mode (1)

Since TTnUDS1, 0 and TTnEIS1, 0 that control the count operation are set to 00B and 01B (rising edge detection), respectively, the counter is operated through detection of the phase of pin TENCTn1 upon detection of the rising edge of TENCTn0 pin input.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 compare register (p).

At this time, the counter is cleared to 0000H if the next count operation is up count.

A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer register (q).

The counter is not cleared upon a match between the counter value and the TTnCCR1 register.

If underflow occurs when TTnLDE = 1 is set, the setting value of the TTnCCR0 buffer register (m) is loaded to the counter. A count operation is possible between 0000H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

<Setting conditions>

TTnCTL0: TTnMD3 to 0 = 1000B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 11B	Judgment of up/down count with count judgment mode 4
TTnCTL1: TTnECM1, 0 = 00B	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, TTnECS1-0 = 01B	Valid edge detection clear (rising edge specified)

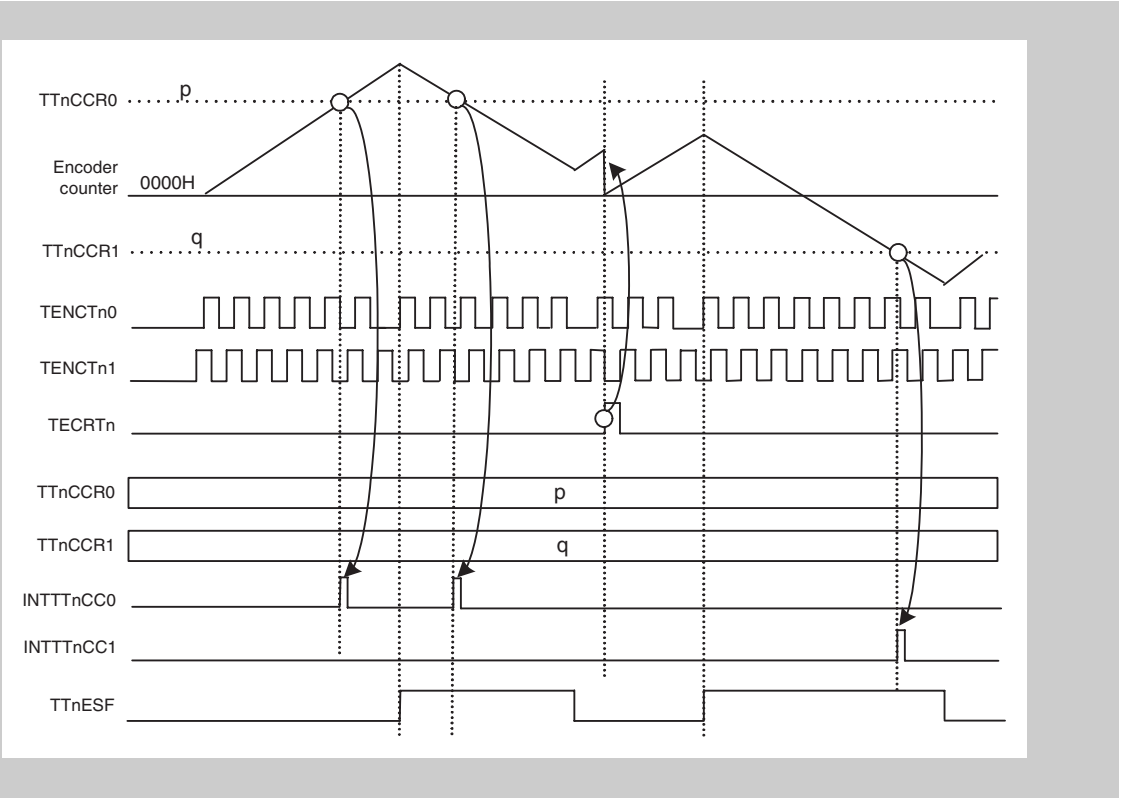


Figure 18-44 Counter hold through bitTTnECC timings (5/8)
Basic timing in encoder compare mode (2)

Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer register (p).

A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer register (q).

The counter is not cleared upon a match with the TTnCCR0 register or the TTnCCR1 register.

Clearing of the counter to 0000H is done upon detection of the valid edge of the encoder clear input (pin TECRTn) when TTnSCE = 0. When TTnECS = 01B is set, the counter is cleared to 0000H in synchronization with the operation clock, following detection of the rising edge of the TECRTn pin input.

<Setting conditions>

TTnCTL0: TTnMD3 to 0 = 1000B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 11B	Judgment of up/down count with count judgment mode 4
TTnCTL1: TTnECM1, 0 = 11B	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: TTnECM1 to 0 = 11B, the setting of bit TTnLDE is invalid.)
TTnIOC3: TTnSCE = 0, TTnECS1-0 = 00B	Valid edge detection clear (no edge specified)

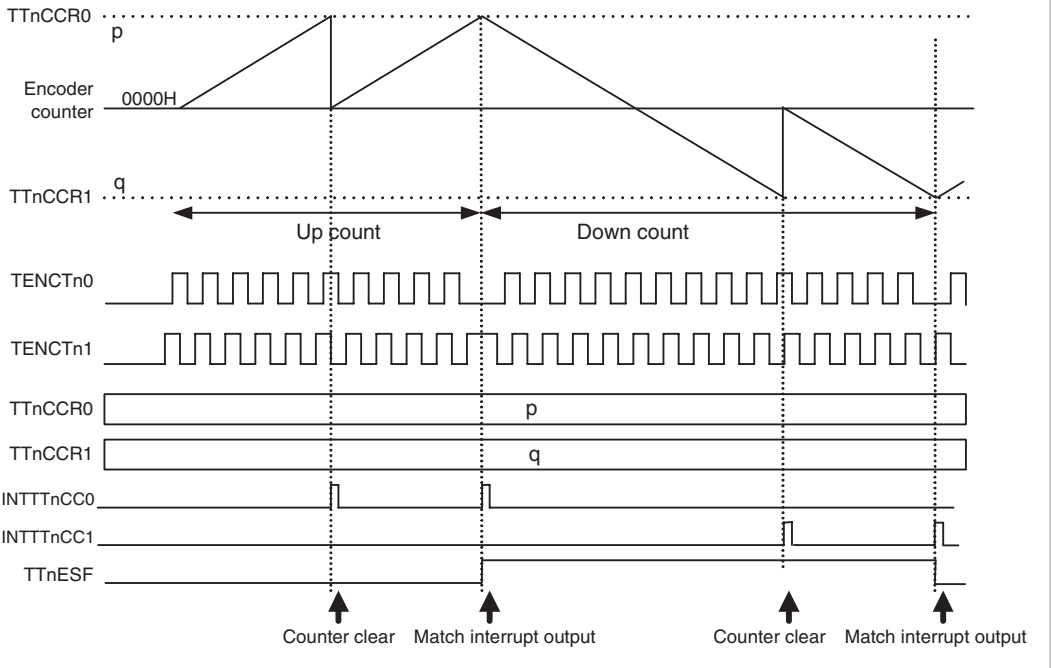


Figure 18-45 Counter hold through bit TTnECC timings (6/8)
Basic timing in encoder compare mode (3)

Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer (p).

At this time, the counter is cleared to 0000H if the next count operation is up count.

A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer (q).

At this time, the counter is cleared to 0000H if the next count operation is down count.

<Setting conditions>

TTnCTL0: TTnMD3 to 0 = 1001B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 11B	Judgment of up/down count using count judgment method 4 (Since TTnCTL0: TTnMD3 to 0 = 1001B, the setting values of bits TTnECM1, TTnECM0, and TTnLDE are invalid.)
TTnIOC3: TTnSCE = 0, TTnECS1, 0 = 00B	Valid edge detection clear (no edge specified)
TTnOPT1: TTnIS1, 0 = 01B	Detection of rising edge of TITn0 pin input
TTnOPT1: TTnIS3, 2 = 01B	Detection of rising edge of TITn1 pin input

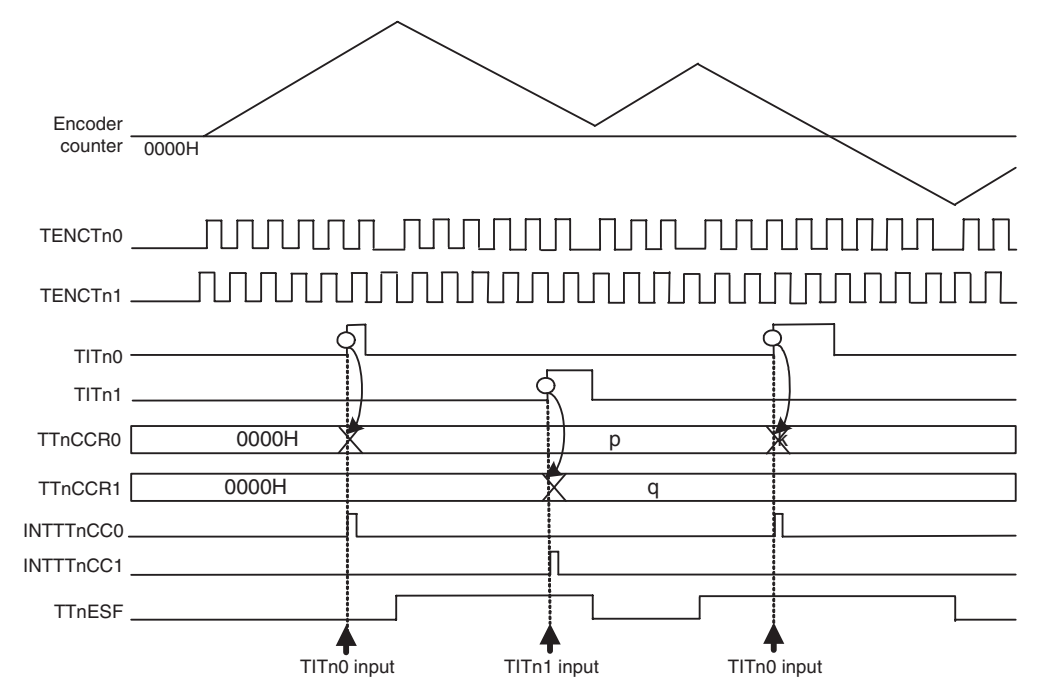


Figure 18-46 Counter hold through bit TTnECC timings (7/8)
Basic timing in encoder capture mode (1)

Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins 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.

<Setting conditions>

TTnCTL0: TTnMD3 to 0 = 1010B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 00B	Judgment of up/down count using count judgment mode 1
TTnCTL1: TTnECM1, 0 = 01B	Counter clear upon match between counter value and TTnCCR0 buffer register
TTnCTL1: TTnLDE = 1	Loading of setting value of TTnCCR0 register to counter upon occurrence of underflow
TTnIOC3: TTnEIS1, 0 = 01B	Detection of rising edge of TENCTn0 and TENCTn1 pin inputs
TTnIOC3: TTnSCE = 0, TTnECS1, 0 = 00B	Valid edge detection clear (no edge specified)
TTnOPT1: TTnIS3, 2 = 01B	Detection of rising edge of TITn1 pin input

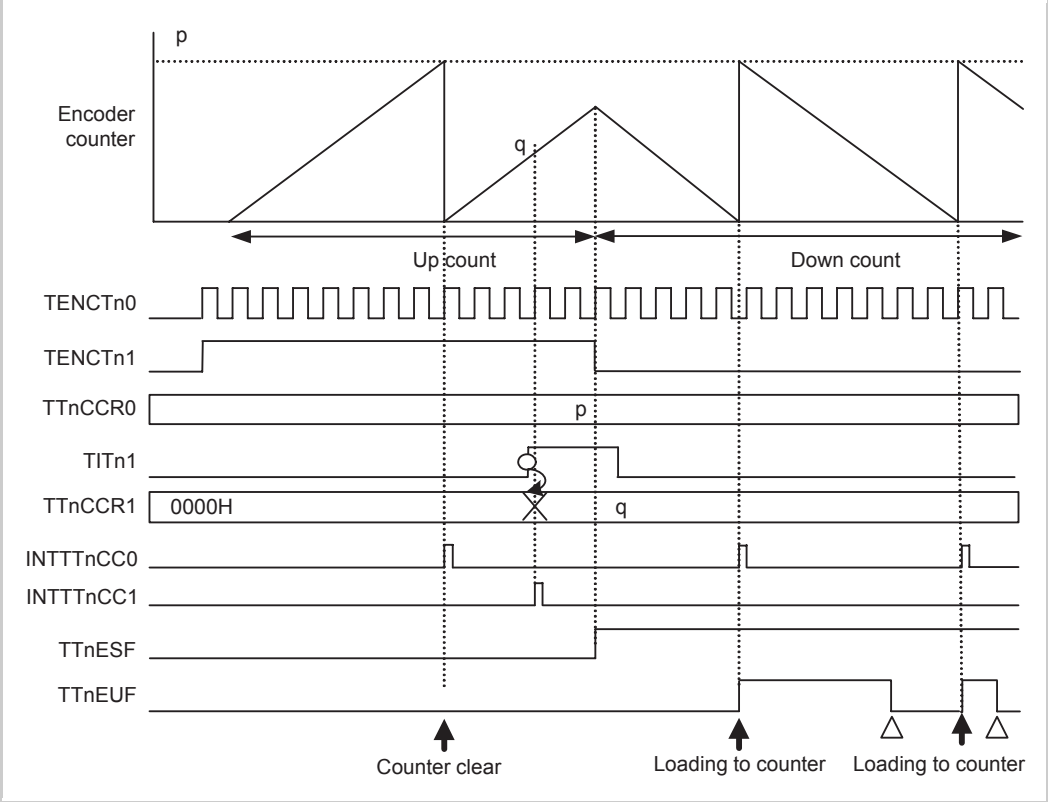


Figure 18-47 Counter hold through bit TTnECC timings (8/8)
Basic timing in encoder capture compare mode (1)

Since TTnUDS1, 0 and TTnEIS1, 0 that controls the count operation are set to 00B and 01B (rising edge detection), respectively, the counter operates through detection of the rising edge of the TENCTn0 pin input and detection of the phase of the TENCTn1 pin.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer register (m). At this time, the counter is cleared to 000H if the next count operation is up count.

Upon detection of the edge of the TITn1 pin input, the counter value is saved to the capture
register (TTnCCR1), and a capture interrupt (INTTTnCC1) is output.

If an underflow occurs due to the TTnLDE = 1 setting, the setting value of the TTnCCR0 buffer register is loaded to the counter. The count operation is possible between 0000H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

18.6.10 Offset trigger generation mode

In the offset trigger generation mode, the count value is saved to the capture register (TTnCCR0) upon detection of the valid edge of the TITn0 pin, and a capture interrupt (INTTTnCC0) is output. The counter is cleared to 0000H by capture input. (Counter clear operation is not performed using the TTnCCR1 register.)

The TTnCCR0 register and the TTnCCR1 register have their functions fixed as a capture register and a compare register, respectively. The TTnCCR1 register can be rewritten during count operation. Regarding compare register reload, the capture & clear timing upon detection of TITn0 pin input serves as the reload timing.

During count operation, a capture interrupt (INTTTnCC0) is output upon capture to the TTnCCR0 register through TITn0 pin input, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and the TTnCCR1 register.

The TOTn0 pin becomes the level set with bit TTnOL0. If TTnOL0 = 0, a low level is output 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 0000H.

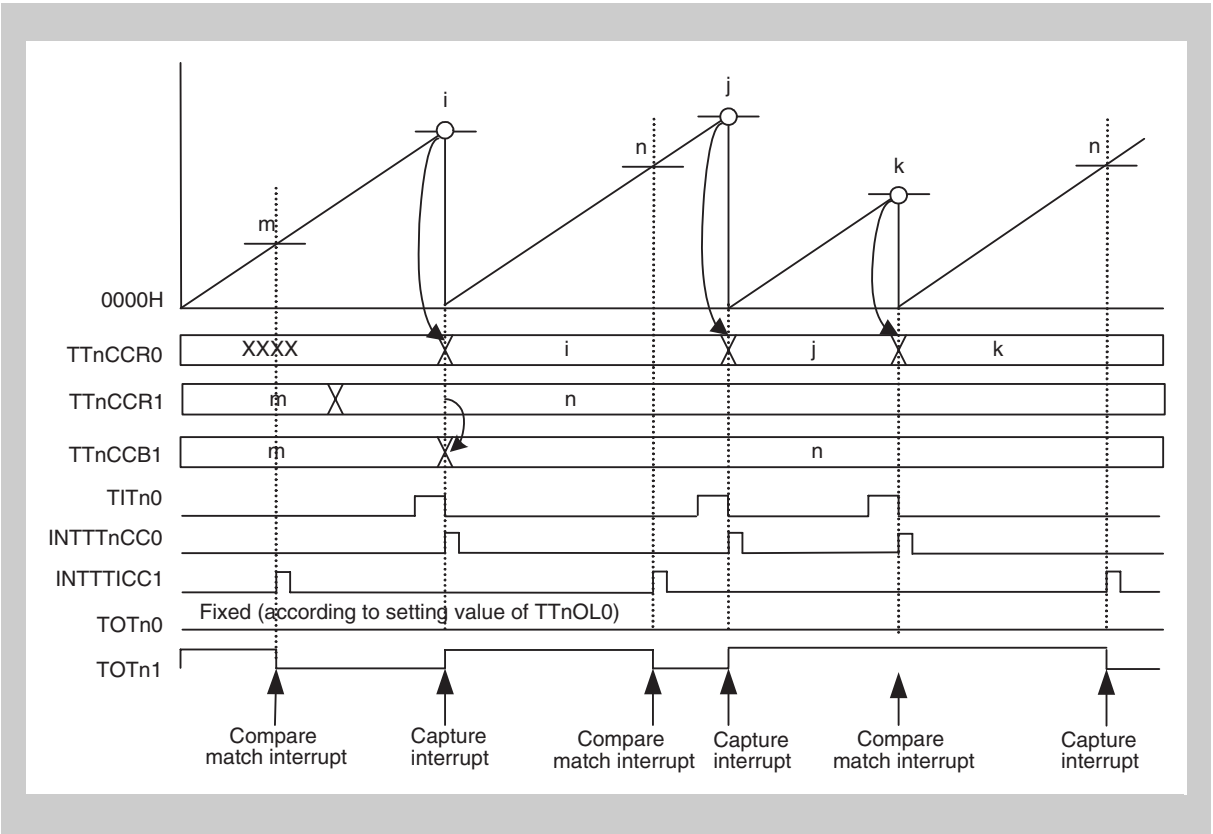


Figure 18-48 Basic timing in offset trigger generation mode

In the offset trigger generation mode, the setting value of the TTnCCR1 register is reloaded to the TTnCCR1 buffer register upon detection of the valid edge of pin TITn0. Until the edge of the TITn0 pin input is detected, the value of the TTnCCR1 register is not reloaded to the TTnCCR1 buffer register, even if this value is changed.

Pin TOTn1 is set when the counter is cleared to 0000H upon detection of the valid edge of pin TITn0, and it is reset upon a match between the counter value and the TTnCCR1 register.

Therefore, pin TOTn1 remains high level if the valid edge of the TITn0 pin input is detected before a match with the TTnCCR1 register occurs.

18.7 Cautions

18.7.1 TMT 'Encoder load enable'-mode limitation

Description If TMT operates

- in encoder compare mode (TTnCTL1.TTnMD[3:0] = 1000_B)
- or in encoder capture/compare mode (TTnCTL1.TTnMD[3:0] = 1010_B),
- and the 'Encoder load enable'-mode is set (TTnCTL2.TTnLDE = 1)

while the TMT operation is enabled (TTnCTL0.TTnCE = 1), an undefined value might be loaded into the counter (TTnCNT) when the TTnCCR0 register is rewritten during counter underflow.

Workaround If the TMT 'Encoder load enable'-mode is set (TTnCTL2.TTnLDE = 1) during TMT operation in encoder compare mode (TTnCTL1.TTnMD[3:0] = 1000_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).

Chapter 19 Timer Interconnection

Various functions can be implemented by connecting the timer I/O and the internal signal between timers.:

- 2-phase encoder function
- 3-phase pulse input control function
- 3-phase encoder function
- TMT as Filter for TMAA

The timer I/O control register (TAAIC3) is used for signal connection between timers. The following shows the timer I/O control registers according to their functions.

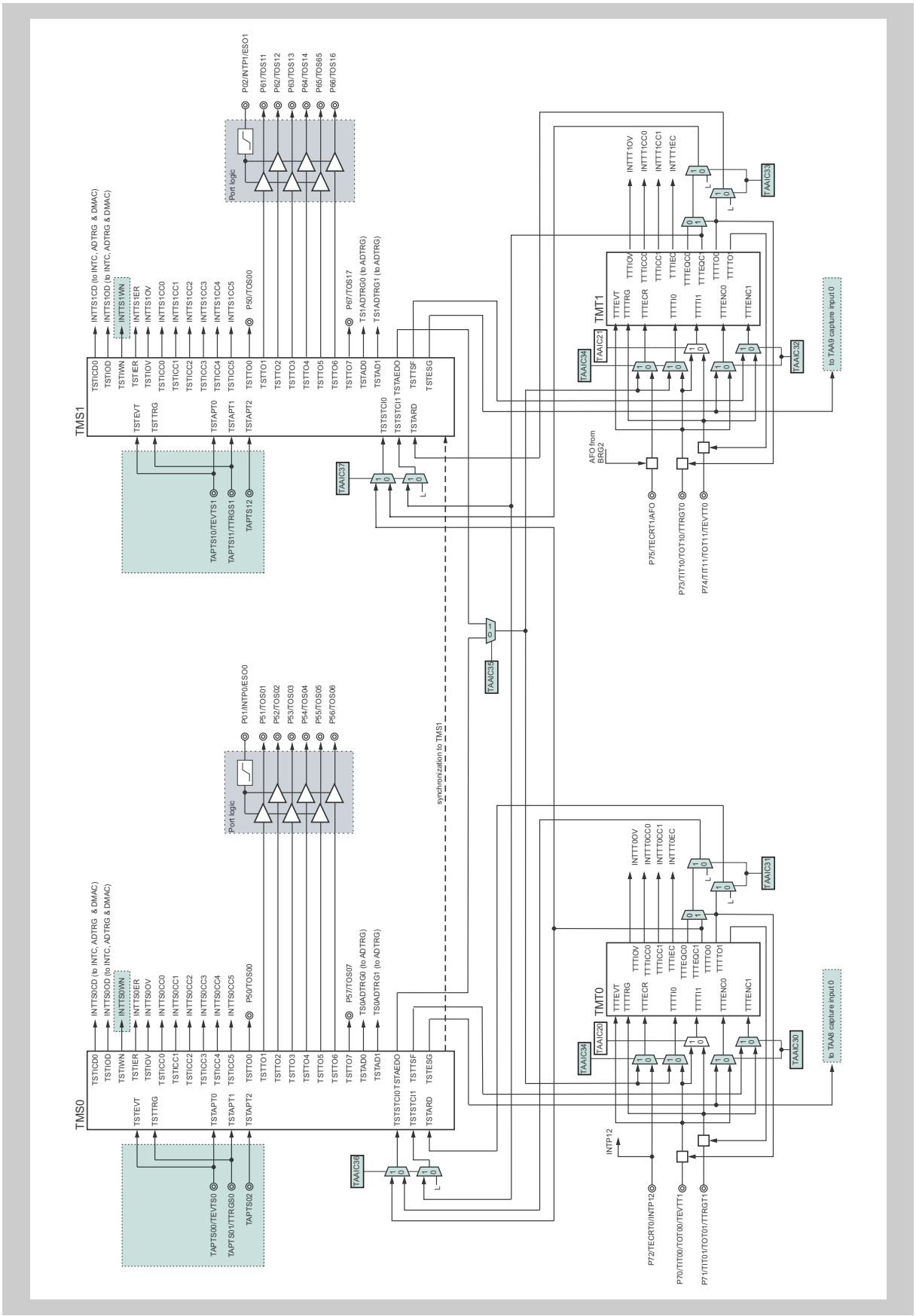


Figure 19-1 Time interconnection block diagram

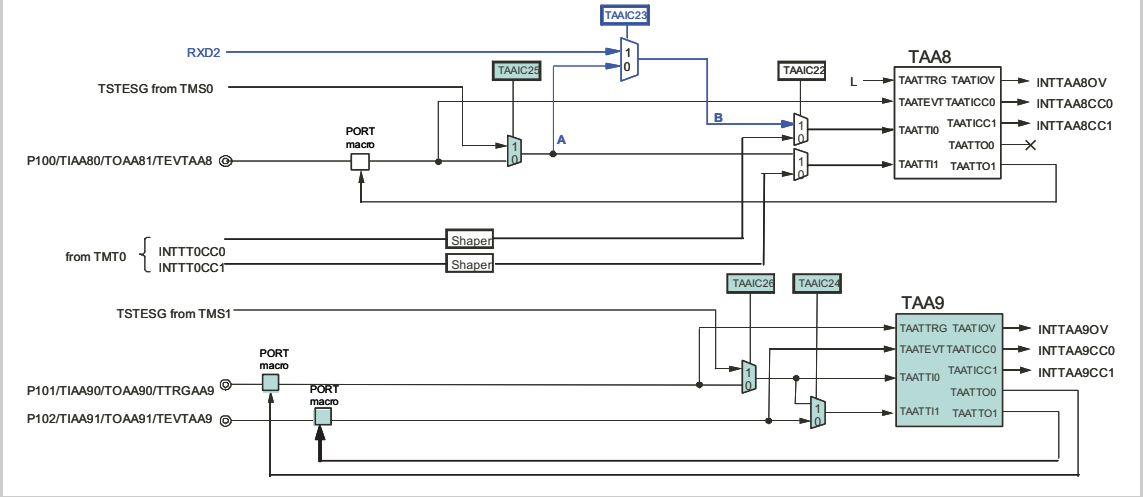


Figure 19-2 Timer AA configuration

19.1 Control register

(1) TAAIC3 - Timer I/O control register 3

The TAAIC3 register selects the timer input signals for enhanced timer control.

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

Address FFFF F6F3$_H$

Initial Value 00$_H$

7	6	5	4	3	2	1	0
TAAIC37	TAAIC36	TAAIC35	TAAIC34	TAAIC33	TAAIC32	TAAIC31	TAAIC30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19-1 TAAIC3 register contents (1/2)

Bit position	Bit name	Function
Bit position	Bit name	Function
7	TAAIC37	Selection of TMS1 TSTSTIC0/TSTSTIC1 input signal 0: "L" signal level TMS1 TSTSTIC1 TMS1 TSTSTIC0: input selected by TAAIC33 bit 1: TMS1 TSTSTIC0: TMT0 TTTEQC1 TMS1 TSTSTIC1: TMT1 TTTEQC1
6	TAAIC36	Selection of TMS0 TSTSTIC0/TSTSTIC1 input signal 0: "L" signal level TMS0 TSTSTIC1 TMS0 TSTSTIC0: input selected by TAAIC31 bit 1: TMS0 TSTSTIC0: TMT0 TTTEQC1 TMS0 TSTSTIC1: TMT1 TTTEQC1
5	TAAIC35	Signal selection input to selector controlled by TAAIC34 bit 0: TMS0 TSTAEDO 1: TMS1 TSTAEDO
4	TAAIC34	Selection of TMTn (n = 0 - 1) TTTECR/TTTTI0 input signal0: "L" level of "L" level of TMS1 0: TMT0 TTTECR:TECRT0 TMT1 TTTECR:TECRT1 TMT0 TTTTI0:TIT00 TMT1 TTTTI0:TIT10 1: TMT1 TTTTI0: selected by TAAIC35 bitt TMT0 TTTTI0: selected by TAAIC35 bitt TMT1 TTTECR: selected by TAAIC35 bitt TMT0 TTTECR: selected by TAAIC35 bitt
3	TAAIC33	Selection of TMS1 TSTARD 120 degree energizing pattern switch sequence control signal and Signal selection input to selector controlled by TAAIC37 bit. 0: "L" level of "L" level at TMS1 TSTARD input TAAIC37 selector input (for TMS1 TSTSTCI1) 1: TMS1 TSTARD:TTTTO0 (from TMT0) TAAIC37 selector (for TMS1 TSTSTCI0): TTTEQC0 or TTTEQC1 from TMT1, selected it with TTTTO0

Table 19-1 TAAIC3 register contents (2/2)

Bit position	Bit name	Function
2	TAAIC32	Selection of TMT1 TTTENC0/TTTENC1 encoder input signal 0: TMT1 TTTENC0:TIT11/TEVTT0 TMT1 TTTENC1:TIT10/TTRGT0 1: TMT1 TTTENC0:TSTESG (from TMS1) TMT1 TTTENC1:TSTTSF(from TMS1)
1	TAAIC31	Selection of TMS0 TSTARD 120 degree energizing pattern switch sequence control signal and Signal selection input to selector controlled by TAAIC36 bit 0: "L" level of "L" level at TMS0 TSTARD input TAAIC36 selector input (for TMS0 TSTSTCI1) 1: TMS0 TSTARD:TTTTO0 (from TMT0) TAAIC36 selector (for TMS0 TSTSTCI0): TTTEQC0 or TTTEQC1 from TMT0 selected with TTTTO0
0	TAAIC30	Selection of TMT0 TTTENC0/TTTENC1 encoder input signal 0: TMT0 TTTENC0:TIT00/TEVTT1 TMT0 TTTENC1:TIT01/TTRGT1 1: TMT0 TTTENC0:TSTESG(from TMS0) TMT0 TTTENC1:TSTTSF(from TMS0)

19.2 2-Phase Encoder Control Function

19.2.1 Function overview

The function enables shifting of the timer S output pattern, such as 120° conducting control by use of the 2-phase encoder.

Table 19-2 Control methods for 2-phase input control

Method	Encoder counter count cycle	Encoder counter clear method
Control method 1	1 motor rotation	Clear by Z-phase input
Control method 2	Arbitrary cycle	Arbitrary value
Control method 3	Count value to shift timer S pattern	Clear by compare value (count value to shift pattern)

The internal signals used in the 2-phase control function are shown below.

Table 19-3 Internal signals used in 2-phase encoder control function

Timer	Signal name	I/O	Function
Timer S	TSnARD	I	Controls the TOSn1 to TOSn6 pin output pattern order. Low level:Shifts the output pattern in normal rotation order. High level:Shifts the output pattern in reverse rotation order.
	TSnSTCI0	I	Output pattern shift trigger
	TSnSTCI1	I	Low level should be input in the 2-phase encoder control function.
	TSnADTRG0	O	Conversion start trigger signal for A/D converter 0. When the A/D converter is used in timer trigger mode, the A/D conversion start trigger is set regardless of the signal connection setting.
	TSnADTRG1	O	Conversion start trigger signal for A/D converter 0. When the A/D converter is used in timer trigger mode, the A/D conversion start trigger is set regardless of the signal connection setting.
Timer T	TTmENC0	I	Encoder A-phase input signal
	TTmENC1	I	Encoder B-phase input signal
	TTmECR	I	Encoder Z-phase input signal
	TTmTI0	I	Cycle measurement capture input signal
	TTmTI1	I	Position detection capture input signal
	INTTTmCC0	O	Timer Tm TTmCCR0 compare match interrupt request signal
	INTTTmCC1	O	Timer Tm TTmCCR1 compare match interrupt request signal

Note n = 0, 1, m = 0 to 5

19.2.2 Configuration

2-phase input control/position detection

(1) Control method 1

The following three types of timer configurations enable 2-phase input control/position detection control.

Table 19-4 Control Method 1 Timer Configuration (1)

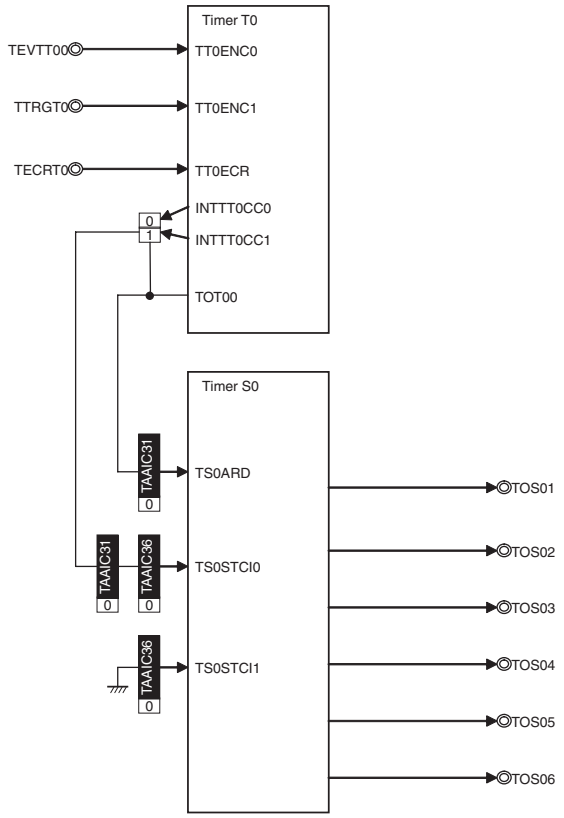
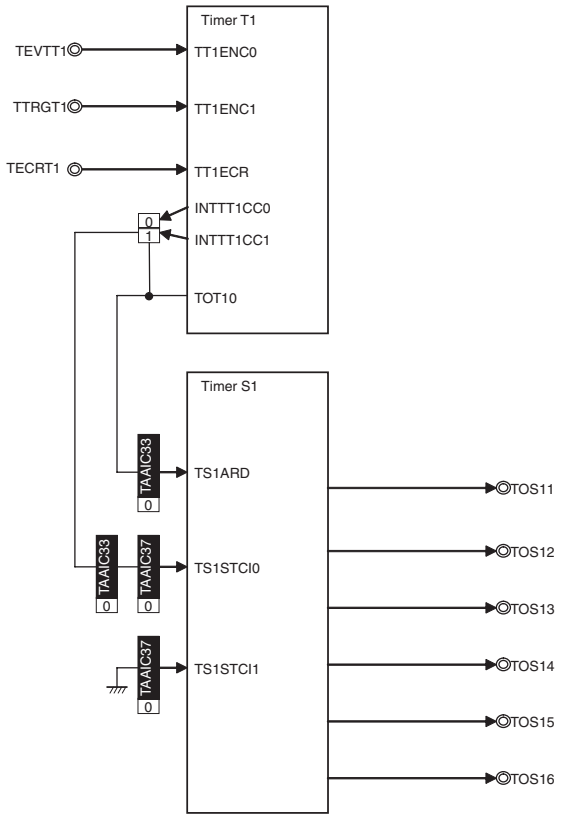
Timer	Function	Block diagram	Mode
Timer T0	Encoder position detection		•Encoder capture compare mode
Timer S0	Motor control (position detection)		Any of the following modes •High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function •120° conduction mode Special 120° conduction mode

Table 19-5 Control Method 1 Timer Configuration (2)

Timer	Function	Block diagram	Mode
Timer T4	Encoder position detection		•Encoder capture compare mode
Timer S0	Motor control (position detection)		Any of the following modes •High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function •120° conduction mode Special 120° conduction mode

19.3 3-Phase Pulse Input Control Function

19.3.1 Function overview

Phase variable control for timer S pattern output is enabled by combining the 3-phase pulse input (TAPTSn2 to TAPTSn0 pin input) of the timer S0/S1 and timer S connection mode of timer T (n = 0, 1).

- Phase fix control: Outputs the pattern fixed at a certain rotation angle.
- Phase variable control: Outputs the pattern where the phase of angle (time) discretionary to the rotation angle is varied.

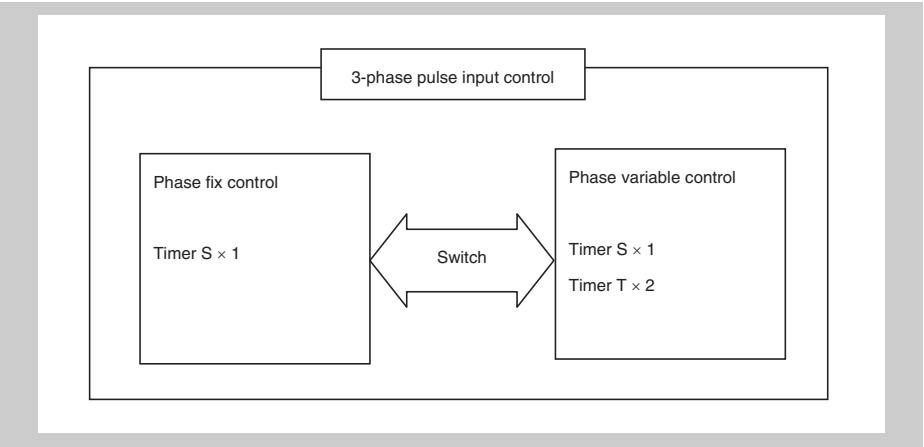


Figure 19-3 Phase fix control and phase variable control

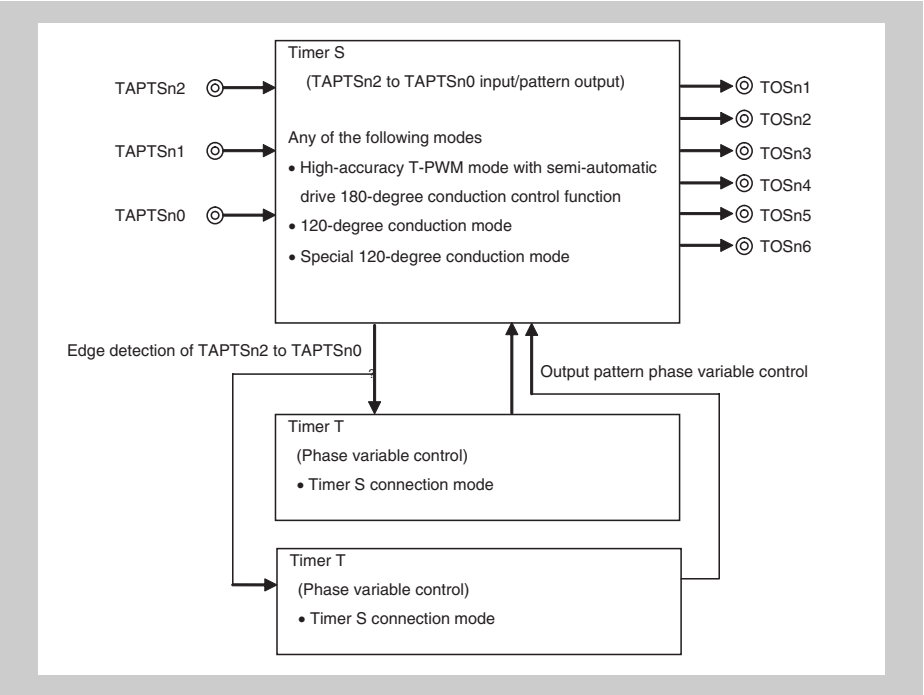
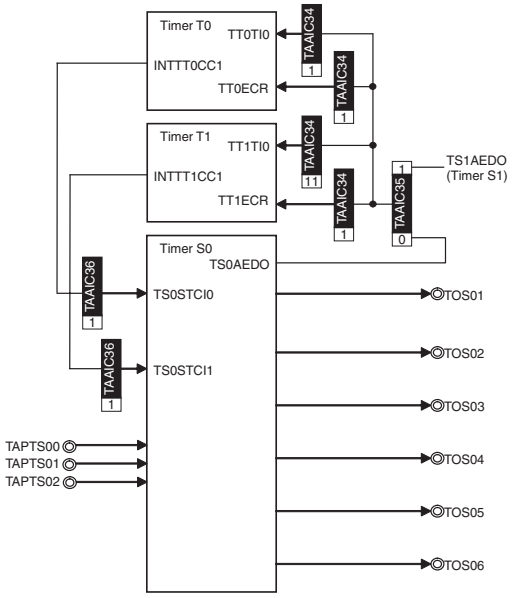


Figure 19-4 Block diagram of 3-phase pulse input control

19.3.2 Configuration

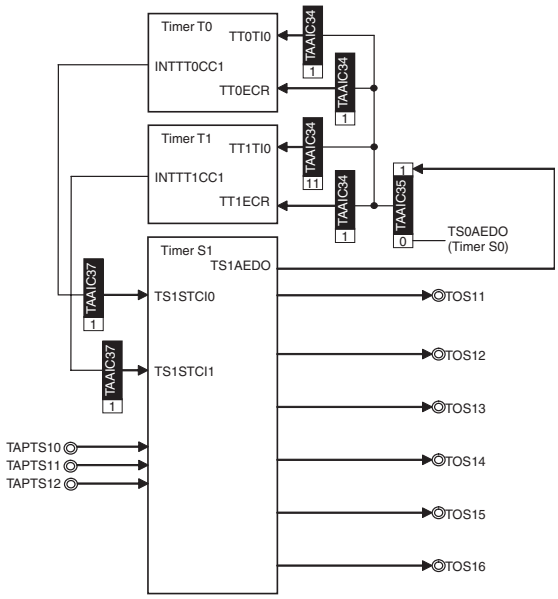
(1) 3-phase pulse input control 1

Table 19-6 3-phase pulse input control 1 timer configuration

Timer	Function	Block diagram	Setting	
			Synchroneous function	Operation mode
Timer T0	Offset control		Simultaneous start (master)	Timer S connetion mode
Timer T1			Simultaneous start (slave)	
Timer S0	Pattern output		Simultaneous start (slave)	Any of the following modes • High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function • 120° conduction mode • Special 120° conduction mode

(2) 3-phase pulse input control 2

Table 19-7 3-phase pulse input control 2 timer configuration

Timer	Function	Block diagram	Setting	
			Synchroneous function	Operation mode
Timer T0	Offset control		Simultaneous start (master)	Timer S connetion mode
Timer T1			Simultaneous start (slave)	
Timer S1	Pattern output		Simultaneous start (slave)	Any of the following modes • High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function • 120° conduction mode • Special 120° conduction mode

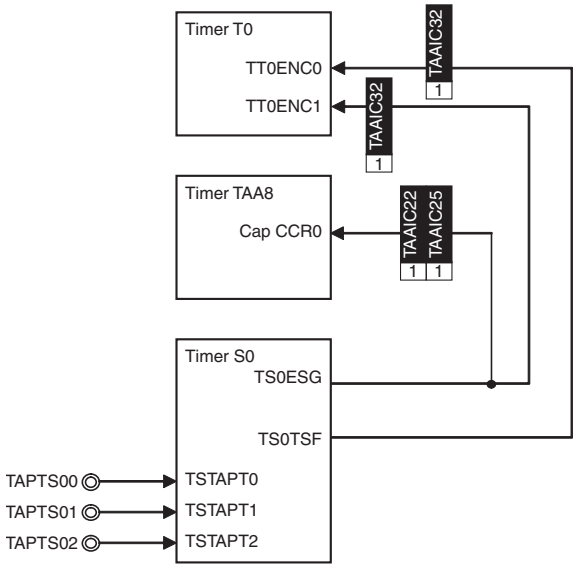
19.4 3-Phase Encoder Function

19.4.1 Function overview

The edges of the signals input to the TAPTS02 to TAPTS00 pins can be connected to any one of timers T0 to T3 to be counted.

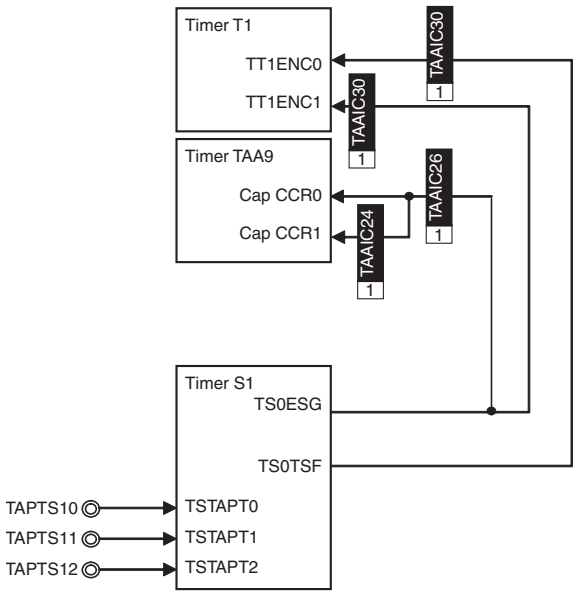
19.4.2 Configuration

Table 19-8 Block diagram of 3-phase encoder (1)

Timer	Function	Block diagram	Mode
Timer T0 Timer AA8	Encoder		Any of the following modes • Encoder compare mode • Encoder capture mode • Encoder capture compare mode
Timer S0	TAPTS02 to TAPTS00 input		Any of the following modes • High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function • 120° conduction mode • Special 120° conduction mode

Note n = 0 to 3 for F240, M192, n = 0, 1 for M128

Table 19-9 Block diagram of 3-phase encoder (2)

Timer	Function	Block diagram	Mode
Timer T0 Timer AA9	Encoder		Any of the following modes • Encoder compare mode • Encoder capture mode • Encoder capture compare mode
Timer S1	TAPTS02 to TAPTS00 input		Any of the following modes • High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function • 120° conduction mode • Special 120° conduction mode

19.5 TMT as Filter for TMAA

To support an extended filtering function (i.e. jittering input signals) for the input of TMAA8, there is a connectivity possible between TMT0 and TMAA8 .
Via this connectivity the INTTnCC0/1 interrupt signals can be chosen as capture-inputs for the CCR0/1 register of TMAA8 (refer also to the description of TAAIC2.TAAIC22 in *“Input Selection Registers” on page 613*).

Chapter 20 On-Chip Debug Unit

The microcontroller includes an on-chip debug unit. By connecting an N-Wire emulator, on-chip debugging can be executed.

20.1 Functional Outline

20.1.1 Debug functions

(1) Debug interface

Communication with the host machine is established by using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO signals via an N-Wire emulator. The communication specifications of N-Wire are used for the interface.

(2) On-chip debug

On-chip debugging can be executed by preparing wiring and a connector for on-chip debugging on the target system. An N-Wire emulator is used to connect the host PC to the on-chip debug unit.

(3) Forced reset function

The microcontroller can be forcibly reset.

(4) Break reset function

The CPU can be started in the debug mode immediately after reset of the CPU is released.

(5) Forced break function

Execution of the user program can be forcibly aborted.

(6) Hardware break function

Two breakpoints for instruction and data access can be used. The instruction breakpoint can abort program execution at any address. The access breakpoint can abort program execution by data access to any address.

(7) Software break function

Up to eight software breakpoints can be set in the internal flash memory area. The number of software breakpoints that can be set in the RAM area differs depending on the debugger to be used.

(8) Debug monitor function

A memory space for debugging that is different from the user memory space is used during debugging (background monitor mode). The user program can be executed starting from any address.

While execution of the user program is aborted, the user resources (such as memory and I/O) can be read and written, and the user program can be downloaded.

(9) Mask function

Each of the following signals can be masked. That means these signals will not be effective during debugging.

The correspondence with the mask functions of the debugger (ID850NWC) for the N-Wire emulator (IE-V850E1-CD-NW) of NEC Electronics is shown below.

- NMI0 mask function: NMI pin
- $\overline{\text{RESET}}$ mask function: external $\overline{\text{RESET}}$ pin
- $\overline{\text{WAIT}}$ mask function: external $\overline{\text{WAIT}}$ pin

(10) Timer function

The execution time of the user program can be measured.

20.1.2 Security function

This microcontroller has a N-Wire security function, that demands the user to input an ID code upon start of the debugger. The ID code is compared to a predefined ID code, written in advance to the internal flash memory by an external flash programmer. This function prevents unauthorized persons to operate the microcontroller in N-Wire debug mode and to read the internal flash memory area.

The ID code in the internal flash memory can only be written by an external flash programmer. It can't be changed in self-programming mode and therefore also not in N-Wire debugging mode.

ID code Be sure to write an ID code when writing a program to the internal flash memory.

The area of the ID code is 10 bytes wide and in the range of addresses $0000\ 0070_H$ to $0000\ 0079_H$.

The ID code when the memory is erased is shown below.

Address	ID code
$0000\ 0079_H$	FF_H
$0000\ 0078_H$	FF_H
$0000\ 0077_H$	FF_H
$0000\ 0076_H$	FF_H
$0000\ 0075_H$	FF_H
$0000\ 0074_H$	FF_H
$0000\ 0073_H$	FF_H
$0000\ 0072_H$	FF_H
$0000\ 0071_H$	FF_H
$0000\ 0070_H$	FF_H

Security bit Bit 7 of address $0000\ 0079_H$ enables or disables use of the N-Wire emulator.

- Bit 7 of address $0000\ 0079_H$

  0: disabled N-Wire emulator cannot connect to the on-chip debug unit.

  1: enabled N-Wire emulator can connect to the on-chip debug unit if the 10-byte ID code input matches the ID code stored in the flash memory

This security bit can only be modified by programming the flash memory via an external flash programmer. It is not possible to modify the security bit in self-programming mode, and therefore also not in N-Wire debugging mode.

After reset, the entire ID code area is set to FF_H. This means that

- N-Wire debugging is generally enabled
- the ID code is FF_H for all ID code bytes

Consequently, controller access is possible without any restriction.

Caution If access via the N-Wire interface should be disabled "block erase disabled" should be configured as well. Otherwise the flash memory blocks containing the ID code could be erased and N-Wire access could be enabled.

20.2 Connection to N-Wire Emulator

To connect the N-Wire emulator, a connector for emulator connection and a connection circuit must be mounted on the target system.

As a connector example the KEL connector is described in more detail. Other connectors, like for instance MICTOR connector (product name: 2-767004-2, Tyco Electronics AMP K.K.), are available as well. For the mechanical and electrical specification of these connectors refer to user's manual of the emulator to be used.

20.2.1 KEL connector

KEL connector product names:

- 8830E-026-170S (KEL): straight type
- 8830E-026-170L (KEL): right-angle type

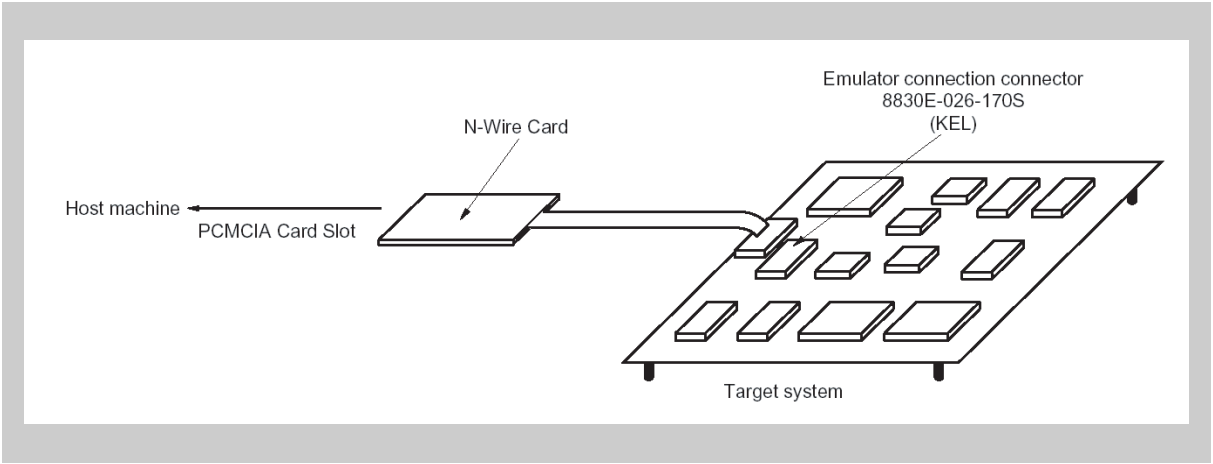


Figure 20-1 Connection to N-Wire emulator (NEC Electronics IE-V850E1-CD-NW: N-Wire Card)

(1) Pin configuration

Figure 20-2 shows the pin configuration of the connector for emulator connection (target system side), and *Table 20-1 on page 1124* shows the pin functions.

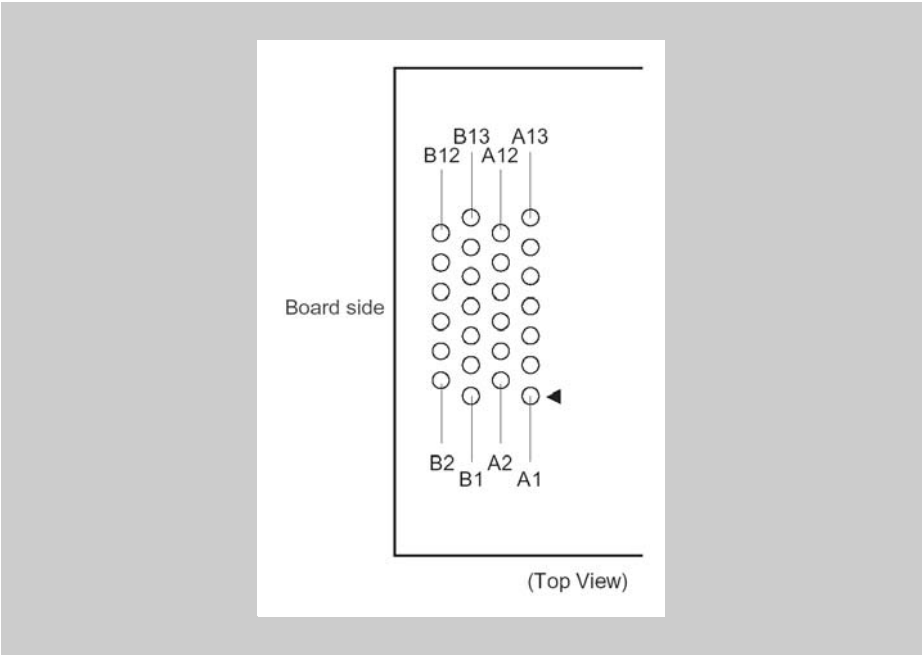


Figure 20-2 Pin configuration of connector for emulator connection (target system side)

Caution Evaluate the dimensions of the connector when actually mounting the connector on the target board.

(2) Pin functions

The following table shows the pin functions of the connector for emulator connection (target system side). “I/O” indicates the direction viewed from the device.

Table 20-1 Pin functions of connector for emulator connection (target system side)

Pin no.	Pin name	I/O	Pin function
A1	(Reserved 1)	–	(Connect to GND)
A2	(Reserved 2)	–	(Connect to GND)
A3	(Reserved 3)	–	(Connect to GND)
A4	(Reserved 4)	–	(Connect to GND)
A5	(Reserved 5)	–	(Connect to GND)
A6	(Reserved 6)	–	(Connect to GND)
A7	DDI	Input	Data input for N-Wire interface
A8	DCK	Input	Clock input for N-Wire interface
A9	DMS	Input	Transfer mode select input for N-Wire interface
A10	DDO	Output	Data output for N-Wire interface
A11	$\overline{\text{DRST}}$	Input	On-chip debug unit reset input
A12	$\overline{\text{RESET}}$	Input	Reset input.
A13	FLMD0	Input	Control signal for flash download (flash memory versions only)
B1	GND	–	–
B2	GND	–	–
B3	GND	–	–
B4	GND	–	–
B5	GND	–	–
B6	GND	–	–
B7	GND	–	–
B8	GND	–	–
B9	GND	–	–
B10	GND	–	–
B11	(Reserved 8)	–	(Connect to GND)
B12	(Reserved 9)	–	(Connect to GND)
B13	V_{DD}	–	3.3 V input (for monitoring power supply to target)

Caution

1. The connection of the pins not supported by the microcontroller is dependent upon the emulator to be used.
2. The pattern of the target board must satisfy the following conditions.
   - The pattern length must be 100 mm or less.
   - The clock signal must be shielded by GND.

(3) Example of recommended circuit

An example of the recommended circuit of the connector for emulator connection (target system side) is shown below.

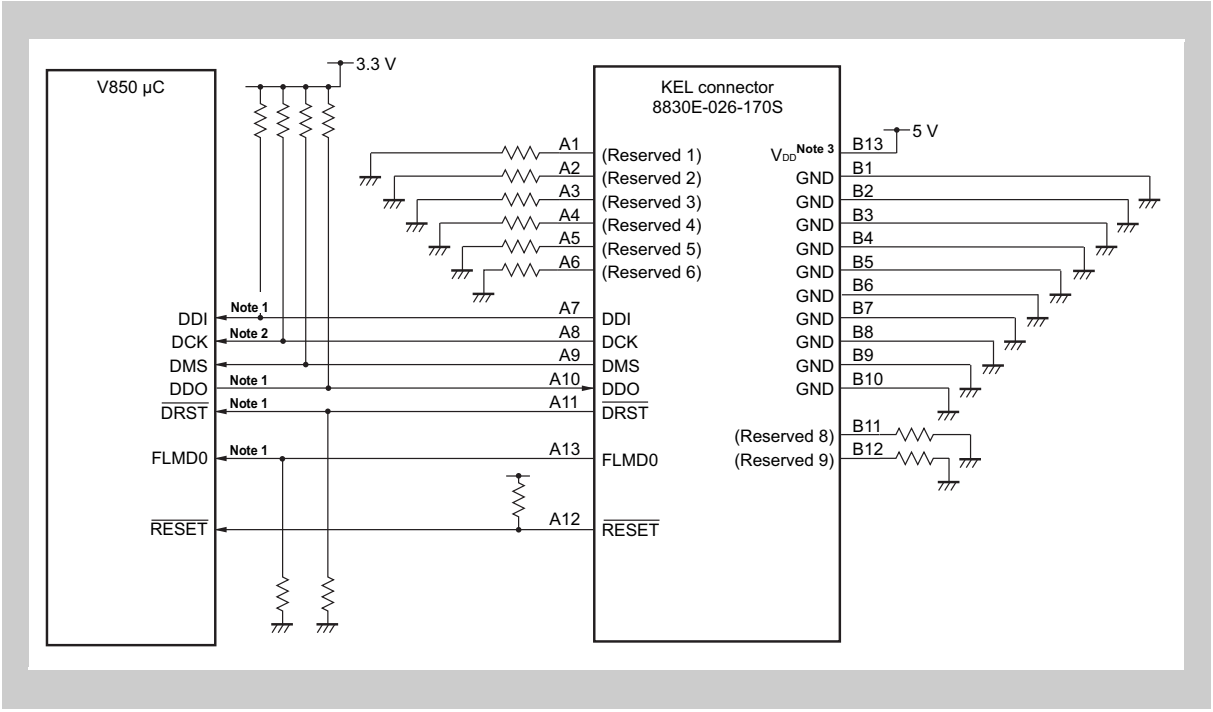


Figure 20-3 Example of recommended emulator connection circuit

Note
1. The pattern length must be 100 mm or less.
2. Shield the DCK signal by enclosing it with GND.
3. This pin is used to detect power to the target board. Connect the voltage of the N-Wire interface to this pin.

Caution The N-Wire emulator may not support a 5 V interface and may require a level shifter. Refer to the user's manual of the emulator to be used.

20.3 Restrictions and Cautions on On-Chip Debug Function

- Do not mount a device that was used for debugging on a mass-produced product (this is because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed).
- The $\overline{\text{RESET}}$ signal input is masked during a break.
- The on-chip debugging unit uses the exception vector address 60_H for software breakpoint (DBTRAP, refer to *"Interrupt Controller (INTC)" on page 125*). Thus the debugger takes over control when one of the following exceptions occur:
  - debug trap (DBTRAP)
  - illegal op-code detection (ILGOP)
  - ROM Correction (if supported by the microcontroller)

  The debugger executes its own exception handler. Therefore, the user's exception handler at address 60_H will not be executed.
- The access to FlexRay macro becomes impossible in case the $\overline{\text{WAIT}}$ mask function is used.

Chapter 21 Random Number Generator (RNG)

This microcontroller incorporates a hardware Random Number Generator (RNG).

21.1 Features

- Random number sequence passing FIPS test
- Random number format: 16 bits
- Seed generated by hardware

21.2 Configuration

(1) RNG - Random number register

The RNG register is a 16-bit register that holds the random number.

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.

Access The register is read-only, in 16-bit units.

Address FFFF $F700_H$

Initial Value undefined

Reset input causes an undefined register content.

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

21.3 Operation

21.3.1 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 time of the RNG register depends on

- the setting of the peripheral function wait control register VSWC
- the system clock frequency f_{xx}

(1) Single read access to RNG

- system clock f_{xx} = 128 MHz:
  $T_{single} = 124.5 \cdot 1/f_{xx} = 0.97$ µs
- system clock f_{xx} = 80 MHz:
  $T_{single} = 114.5 \cdot 1/f_{xx} = 1.43$ µs

(2) Consecutive read access to RNG

- system clock f_{xx} = 128 MHz:
  $T_{single} \leq T_{consecutive} \leq 1148.5 \cdot 1/f_{xx} = 0.9$ µs
- system clock f_{xx} = 80 MHz:
  $T_{single} \leq T_{consecutive} \leq 1138.5 \cdot 1/f_{xx} = 14.23$ µs

Chapter 22 Auxiliary Frequency Output Function (AFO)

This microcontroller has an Auxiliary Frequency Output Function (AFO).

22.1 Features

- Frequency up to 8 Mbps
- Programmable frequency output
- Interval timer function
- Interrupt request signal (INTBRG2)

22.2 Configuration

The AFO function includes the following hardware.

Table 22-1 AFO Configuration

Item	Configuration
Control registers	Prescaler mode register 2 (PRSM2)
	Prescaler compare register 2 (PRSCM2)

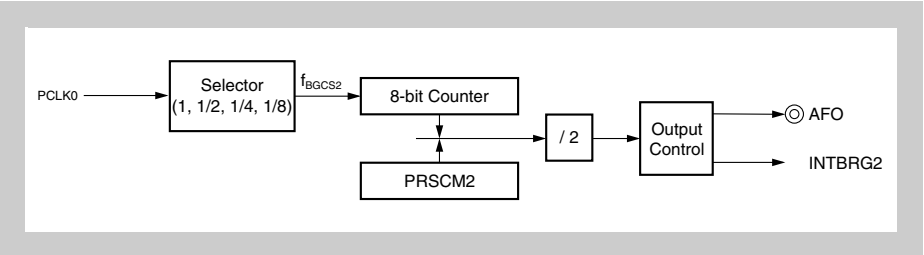


Figure 22-1 Block diagram of Auxiliary Frequency Output Function

Note PCLK0: 32 MHz peripheral clock (refer to *"Clock Generator" on page 179*)

22.3 Control Registers

(1) PRSM2 - Prescaler mode register 2

The PRSM2 register controls generation of a baud rate signal for the AFO function.

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

Address FFFF FDE0$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

Reset input clears this register to 00$_H$.

7	6	5	4	3	2	1	0
0	0	0	BGCE2	0	TODIS2	BGCS21	BGCS20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-2 PRSM2 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>4, 2</td><td>BGCE2,
TODIS2</td><td>Sets the baud rate generator output
<table>
<tr><th>BGCE</th><th>TODIS2</th><th>AFO</th><th>INTBRG2</th><th>Counter</th></tr>
<tr><td>0</td><td>0</td><td>fixed on 0</td><td>fixed on 0</td><td>fixed on 01$_H$</td></tr>
<tr><td>0</td><td>1</td><td>fixed on 0</td><td>fixed on 0</td><td>fixed on 01$_H$</td></tr>
<tr><td>1</td><td>0</td><td>active</td><td>active</td><td>active</td></tr>
<tr><td>1</td><td>1</td><td>fixed on 0</td><td>active</td><td>active</td></tr>
</table></td></tr>
<tr><td>1 to 0</td><td>BGCS2[1:0]</td><td>Sets the baud rate generator clock
<table>
<tr><th>BGCS21</th><th>BGCS20</th><th>Baud Rate Generator Clock Selection (f_{BGCS2})</th><th>Setting Value (k)</th></tr>
<tr><td>0</td><td>0</td><td>f_{PCLK0} / 1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>f_{PCLK0} / 2</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>f_{PCLK0} / 4</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>f_{PCLK0} / 8</td><td>3</td></tr>
</table></td></tr>
</table>

Caution
1. Do not rewrite the PRSM2 register during operation.
2. Set bits BGCS21, BGCS20 and TODIS2 before setting the BGCE2 bit to 1.

(2) PRSCM2 - Prescaler compare registers 2

The PRSCM2 register is an 8-bit compare register. Together with the clock selected in the PRSM2 register, this register sets the clock of the AFO.

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

Address FFFF FDE1$_H$

Initial Value 00$_H$. This register is initialized by $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
PRSCM27	PRSCM26	PRSCM25	PRSCM24	PRSCM23	PRSCM22	PRSCM21	PRSCM20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-3 PRSM2 register contents

Bit position	Bit name	Function
7 to 0	PRSCM2 [7:0]	Sets the AFO clock

PRSCM27	PRSCM26	…	PRSCM22	PRSCM21	PRSCM20	AFO clock	N
0	0	...	0	0	0	f_{BGCS2}/256	256
0	0	...	0	0	1	f_{BGCS2}	1
0	0	...	0	1	0	f_{BGCS2}/2	2
0	0	...	0	1	1	f_{BGCS2}/3	3
…	…	...	…	…	…	…	
1	1	...	1	0	0	f_{BGCS2}/252	252
1	1	...	1	0	1	f_{BGCS2}/253	253
1	1	...	1	1	0	f_{BGCS2}/254	254
1	1	...	1	1	1	f_{BGCS2}/255	255

Caution
1. Do not rewrite the PRSCM2 register during operation.
2. Set the PRSCM2 register before setting the PRSM2.BGCE2 bit to 1.
3. Do not set the AFO clock to a frequency higher than 8 MHz.

Note f_{BGCS2}: Clock frequency selected by bits PRSM2.BGCS2[1:0].

22.4 Operation

22.4.1 Auxiliary frequency generation

The auxiliary frequency output clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{AFO} = \frac{f_{BGCS2}}{N \times 2} = \frac{f_{XX}}{2^{(k+2)} \times N \times 2}$$

f_{AFO}	AFO clock
f_{BGCS2}	Clock frequency selected by bits PRSM2.BGCS21and PRSM2.BGCS20
f_{XX}	Main clock oscillation frequency
k	PRSM2 register setting value ($0 \leq k \leq 3$)
N	• if PRSCM2 = 01_H to FF_H: N = PRSCM2 register value (1 to 255) • if PRSCM2 = 00_H: N = 256

22.4.2 Interval timer function

The AFO function can be used as interval timer regardless whether the auxiliary frequency output is used or not. For this purpose an interrupt request signal (INTBRG2) is assigned, which can be handled like any maskable interrupt.

Chapter 23 Non Break Debug Unit (NBD)

This microcontroller includes a Non Break Debug (NBD) unit which allows to access code flash, RAM, Tuning RAM and peripherals in the background while the CPU is executing program code. The NBD interface is supported by different 3rd party tools like RTE-NBD2 (Midas Lab Inc.) or DCI-GSI1 (dSPACE). By connecting the 3rd party tools to the device, the on-chip debugging can be executed.

23.1 Functional Outline

23.1.1 Debug functions

(1) Debug interface

Communication with the host machine is established by using the signals CLK_DBG, SYNC, TRIG_DBG, MODE_DBG, AD0_DBG, AD1_DBG, AD2_DBG and AD3_DBG. The communication specifications of the NBD are used for this interface.

Table 23-1 NBD pin functions summary

Pin name	I/O	Pin function summary
CLK_DBG	Input	Serial clock input for debugging interface
SYNC	Input	Synchronization signal for debugging
AD0_DBG to AD3_DBG	Input/Output	Command data or RAM data input/output (4 bits)
TRIG_DBG	Output	Outputs trigger (falling edge) synchronized to timing of write to arbitrary specified RAM address or to timing of execution of instruction at specified address.
MODE_DBG	Input	NBD enable signal

(2) NBD debug

The NBD debug functions can be executed by connecting the 3rd-Party product to the NBD interface of this microcontroller. Additionally the 3rd-Party product must be connected to the host PC.

(3) RAM monitor

This function provides the reading and writing of an arbitrary RAM area from outside the microcontroller without halting the execution of a running program. The data transfer is done by a dedicated DMA channel for NBD.

The internal RAM area from $03FF0000_H$ to $03FFEFFF_H$ (60 KB) can be monitored.

(4) Flash read function

This function provides the reading of an arbitrary flash area from outside the microcontroller without halting the execution of a running program. The data transfer is done by a dedicated DMA channel for NBD.

The corresponding flash area is: 00000000_H to $000BFFFF_H$/$000F7FFF_H$ (992 KB/1 MB)

(5) Event detection function

The NBD unit provides an event detection function which is able to detect

- the execution of program code at a specified address
  corresponding address range: 00000000_H to $000BFFFF_H$/$000F7FFF_H$ (992 KB/1 MB)
- the writing of data to a specified internal RAM address
  corresponding address range: $03FF0000_H$ to $03FFEFFF_H$ (60 KB)

On event detection the signal TRIG_DBG is generated (falling edge).

Note Setting the address outside the above-mentioned detection range, will also cause a trigger event if data is written to specified address.

(6) Data tuning function

The code flash memory area is divided in to 248 blocks of 4 KB size. Each of these blocks can be replaced by a 4 KB block of tuning RAM enabling the CPU to fetch code or data from this data tuning RAM instead. The tuning RAM can be read and written by the NBD tool.

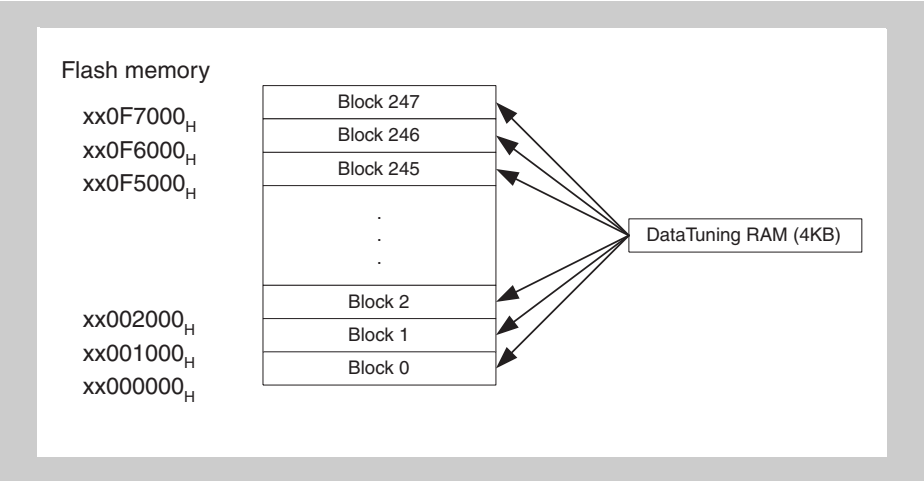


Figure 23-1 Data tuning RAM function

(7) Peripheral I/O register read function

This feature provides a peripheral I/O area read function from outside the chip by dedicated DMA for NBD. Reading registers of peripheral I/O area without halting the execution of a running program is possible by this functionality.

Corresponding peripheral I/O area: $0FFFF000_H$ to $0FFF\ FFFF_H$ (4 KB)

23.2 Connection to RTE-NBD2 Evaluator

To connect the RTE-NBD2 evaluator, a connector for the evaluator must be mounted on the target system.

As example a 16 Pin connector XG4C-1631/1634 manufactured by Omron Corporation is described.

Omron connector

The Omron product names are:

- XG4C-1631 (straight dip terminal)
- XG4C-1634 (right-angle dip terminal)

(1) Pin configuration

The table below shows the pin configuration of the connector mounted on the target system.

Table 23-2 Pin functions of connector for evaluator connection (target system side)

Pin no.	Pin name	I/O	Pin connection
1	TRIG_DBG	Output	Connected to NBD_TRG signal of NBD
2	VCC3.3	Output	Connected to NBD_VCC
3	Reserved	–	Not used, usually connected to NBD_OUT.
4	GND	–	(Connect to GND)
5	CLK_DBG	Input	Connected to NBD_CLK
6	GND	–	(Connect to GND)
7	SYNC	Input	Connected to NBD_SYNC
8	GND	–	(Connect to GND)
9	AD0_DBG	Input/Output	Connected to NBD_DATA0
10	GND	–	(Connect to GND)
11	AD1_DBG	Input/Output	Connected to NBD_DATA1
12	GND	–	(Connect to GND)
13	AD2_DBG	Input/Output	Connected to NBD_DATA2
14	AD3_DBG	Input/Output	Connected to NBD_DATA3
15	MODE_DBG	Input	Connected to NBD_DBG
16	Reserved	–	Not used, usually connected to NBD_RESETO

(2) Example of recommended circuit

An example of the recommended circuit of the connector for the evaluator connection (target system side) is shown below.

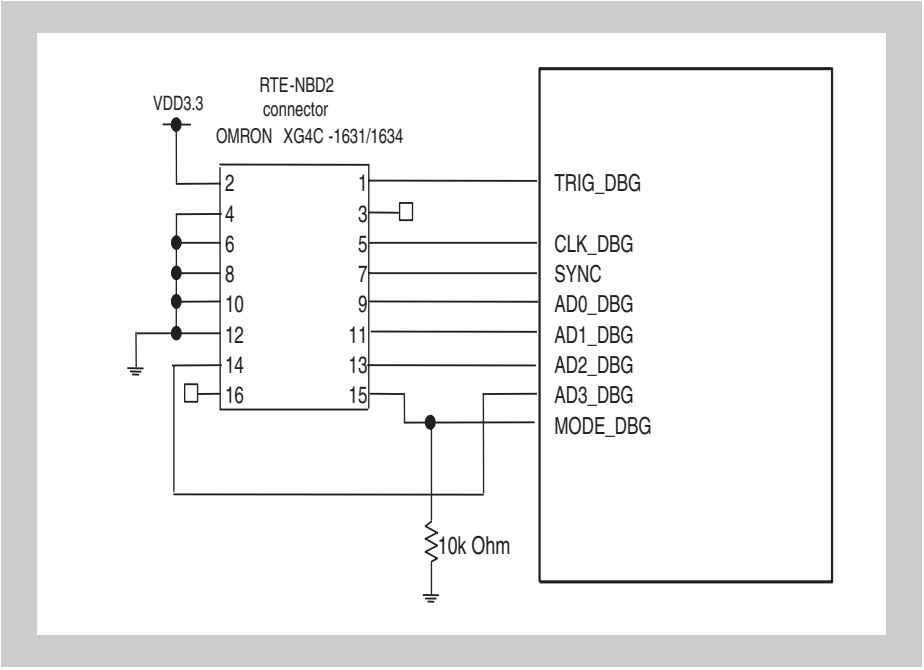


Figure 23-2 Example of recommended evaluator connection circuit

23.3 Restrictions and Cautions on NBD Functions

- Only the RAM monitor function can be used in conjunction with the NWIRE on-chip debugger. All other features like event detection, data tuning, peripheral I/O read and flash read are not available if the NWIRE debugger is connected.
- Do not reset the microcontroller during the NBD protocol execution. To reset the microcontroller, assert the reset pin after completion of the NBD protocol. When the microcontroller is put into reset state during NBD protocol execution the READY signal is not output. Therefore, SYNC must be inactive when the microcontroller is put into reset state.
- NBD can't be accessed from the reset beginning to the trimming end (included PLL lock-up time).
- The ICE space and target space can't be accessed during reset. Therefore changing of boot territory (4 KB) for tuning RAM is not possible.
- NBD must not be used during code flash programming by external programmer.
- NBD must not be used during code flash programming by Self Programming Library.
- NBD must not be used during data flash programming by Self Programming Library.
- If RAM monitor function or peripheral I/O read function is used during CRC program debugging, the CRC function doesn't work correctly with DMA support.
- It is prohibited to write to RAM by the RAM monitor function when the bit RAMECC.RAEENB = 1 (RAM ECC disables operation).
- The following registers can not be read by NBD peripheral I/O register read function
  - CSC0, CSC1, BPC, BSC, BEC, VSWC
  - programmable peripheral I/O area (all AFCAN register)
  - external memory area (FlexRay, CRC, Data Flash)

  The behavior of self clearing registers is the same for read access by NBD and read access by CPU. In both cases the registers or flags are cleared and the original status gets lost after NBD access.

  Accessing the following listed registers by peripheral I/O read function will cause a state change of the related receive state machine i.e. overrun error will not be generated because data has already been read by NBD:
  - UCnRX UART data receive register (n = 0 - 2)
  - UCnRXL UART data receive register low (n = 0 - 2)
  - CBnRX CSIB data receive register (n = 0 - 1)
  - CBnRXL CSIB data receive register low (n = 0 - 1)
  - CEnRX CSIE data receive register (n = 0 - 1)
  - CEnRXL CSIE data receive register low (n = 0 - 1)
  - CEnRXH CSIE data receive register high (n = 0 - 1)
- If event detection for specific internal RAM address is enabled and data is written to this specified internal RAM address by DMA the event is not detected.
- Reading and writing into tuning RAM is prohibited during fetch access to the tuning RAM.

- Writing to the tuning RAM by CPU is prohibited.
- There is a possibility that a double error interrupt is generated when NBD accesses the internal RAM while executing RAM monitor function.
- Do not read non-initialized internal RAM by RAM monitor function. There is a possibility that double errors occur.
- Specific protected sequences e.g. protected by PRCMD, may be disturbed by RAM monitor function (RAM read) and a protection error (PHS) may occur. Therefore it is recommended that the user always checks the status of PHS register.
- RAM monitor function or peripheral I/O read and simultaneous program execution within internal RAM is prohibited. When the RAM monitor function or peripheral I/O read are executed and one of the following instructions are fetched from the internal RAM, the CPU may deadlock:
  - bit manipulation instructions (SET1, CLR1, or NOT1) on any target data (RAM, SFR, etc.)
  - instructions with misaligned access to data in the internal RAM. Misaligned access means:
    - access to 32-bit (word) data on addresses with lower 2 address bits unequal 0
    - access to 16-bit (half-word) data on addresses with lowest address bit unequal 0

  Once the CPU has entered this deadlock state only a reset can be acknowledged, neither maskable nor non-maskable interrupts will be serviced any more.
- The external bus cycle may be extended at any cycles unrelated CPU cycles by inserting the clock stop operation of NBD.
- Using the WAIT signal of the external memory interface is prohibited during operation of the NBD flash read or tuning RAM function.
- The blocks 248 to 255 comprise no code flash, therefore the substitution of these blocks is not possible.
- Do not write EVTU_C and EVTU_A register by NBD during standby (HALT) mode. There is the possibility that the terminal TRIG_DBG becomes undefined when the PC execution event is selected.

23.4 NBD Registers

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

Table 23-3 NBD registers overview

Register name	Shortcut	Address (NBD space)
Chip ID register 0	TID0	000_H
Chip ID register 1	TID1	001_H
Chip ID register 2	TID2	002_H
Event address setting register	EVTU_A<7:0>	800_H
Event address setting register	EVTU_A<15:8>	801_H
Event address setting register	EVTU_A<23:16>	802_H
Event address setting register	EVTU_A<31:24>	803_H
Event condition setting register	EVTU_C	820_H
Data tuning RAM arrangement register	TLR0	830_H
Data tuning permission register	TCR0	831_H

(1) TIDn - Chip ID register n (n = 0- 2)

The TIDn registers comprise a chip ID readable via the NBD interface. The Chip ID is a fixed value for each product. These registers contain information regarding e.g. semiconductor maker, CPU and individual attributes.

Access This register is read-only in 8-bit units.

NBD Address Space TID0: 000_H

Initial Value $4E_H$

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

Table 23-4 TID0 register content

Bit position	Bit name	Function
7 to 0	MC	Semiconductor maker sort code NEC:$4E_H$

Access This register is read-only in 8-bit units.

NBD Address Space TID1: 001_H

Initial Value 01_H

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

Table 23-5 TID1 register content

Bit position	Bit name	Function
7 to 0	FC	CPU family code V850 family: 00_H V850E family: 01_H

Access This register is read-only in 8-bit units.

NBD Address Space TID2: 002_H

Initial Value 28_H

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

Table 23-6 TID2 register content

Bit position	Bit name	Function
7 to 0	SC	V850E/PHO3: 28_H

(2) EVTU_A - Event address setting register

The EVTU_A registers is used to specify the event object address of NBD. If the event object adress coincides and the event object condition is met, the TRIG_DBG signal is generated.

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

NBD Address Space
EVTU_A<7..0>: 800_H
EVTU_A<15..8>: 801_H
EVTU_A<23..16>: 802_H
EVTU_A<31..24>: 803_H

Initial Value not specified

7	6	5	4	3	2	1	0
EVUA7	EVUA6	EVUA5	EVUA4	EVUA3	EVUA2	EVUA1	EVUA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
EVUA15	EVUA14	EVUA13	EVUA12	EVUA11	EVUA10	EVUA9	EVUA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
EVUA23	EVUA22	EVUA21	EVUA20	EVUA19	EVUA18	EVUA17	EVUA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
31	30	29	28	27	26	25	24
EVUA31	EVUA30	EVUA29	EVUA28	EVUA27	EVUA26	EVUA25	EVUA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-7 EVTU_A register content

Bit position	Bit name	Function
31...0	EVAUn	Event object address

(3) EVTU_C - Event condition setting register

The EVTU_C registers is used to specify the event detection condition of NBD. If the event object adress coincides and the event object condition is met, the TRIG_DBG signal is generated.

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

NBD Address Space 820_H

Initial Value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PCU/DTU
R	R	R	R	R	R	R	R/W

Table 23-8 EVTU_C register content

Bit position	Bit name	Function
0	PCU/DTU	Event detection condition. 0: RAM write access event selected (Note). 1: PC execution event selected.

Note When the address coincides, even if the adress is located outside the specified RAM area, the event is detected.

(4) TLR0 - Data tuning RAM arrangement register

The TLR0 registers is used to specify the flash block number which has to be substituted by the data tuning RAM. The flash block number can be set from 0 to 255. Only blocks of 4 KByte can be replaced.

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

NBD Address Space 830_H

Initial Value 00_H

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

Table 23-9 EVTU_C register content

Bit position	Bit name	Function
7..0	BK	Flash block number which has to be replaced by data tuning RAM. Selectable block number is 0 to 255 each block represents a size of 4KByte.

Caution The blocks 248 to 255 comprise no code flash, therefore the substitution of these blocks is not possible.

(5) TCR0 - Data tuning permission register

The TCR0 registers is used to control read / write objects regarding NBD and CPU accesses.

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

NBD Address Space 831_H

Initial Value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TEC	TED
R	R	R	R	R	R	R/W	R/W

Table 23-10 TCR0 register content

Bit position	Bit name	Function
1	TEC	Enable / Disable fetch from Data Tuning RAM 0: CPU fetches from internal flash 1: CPU fetches from Data Tuning RAM
0	TED	Specify the read / write object for NBD 0: Read data from flash block specified by TLR0 register. 1: Read / write data from / to Data Tuning RAM

Caution Do not enable TLR0.TEC bit before flash block is specified by TLR0 register.

Note The TLR0 register has to be set before read access to internal flash or read / write access to Data Tuning RAM.

Chapter 24 FlexRay™ (FR)

Note FlexRay™ is a trademark of DaimlerChrysler AG.

24.1 About this Document

24.1.1 Conventions

The following conventions are used within this document:

- **Helvetica bold** Names of bits and signals
- CAPITALS POC states and CHI commands

Note The grey coloured cells in the register description indicate that access to such bits is ignored.

24.1.2 Definition

- FlexRay Frame: Header Segment + Payload Segment
- Message Buffer: Header Section + Data Section
- Message RAM: Header Partition + Data Partition
- Data Frame: FlexRay frame that is not a null frame

24.1.3 Scope

This chapter describes the FlexRay IP-module and its features from the application programmer's point of view.

24.1.4 References

This document refers to the following documents:

Ref	Author(s)	Title
1	FlexRay Group	FlexRay Communication System Protocol Specification v2.1 (05/05/12)
2	FlexRay Group	FlexRay Communication System Protocol Specification v2.1 Revision A Errata Sheet Version 1 (06/03/29)
3	BOSCH AE/EIP	Addendum to E-Ray FlexRay IP-Module Specification Revision 1.2.3
4	BOSCH AE/EIP	E-Ray FlexRay IP-Module Specification Revision 1.2.3u

24.1.5 Terms and abbreviations

This document uses the following terms and abbreviations:

Term	Meaning
AP	Action Point
BD	Bus Driver
BSS	Byte Start Sequence
CAS	Collision Avoidance Symbol
CC	Communication Controller
CHI	Controller Host Interface
CIF	Customer Interface
CRC	Cyclic Redundancy Check
FES	Frame End Sequence
FIFO	First In First Out (message buffer structure)
FSM	Finite State Machine
FSP	Frame and Symbol Processing Block
FSS	Frame Start Sequence
FTM	Fault Tolerant Midpoint
GIF	Generic Interface Block
GTU	Global Time Unit Block
IBF	Input Buffer
INT	Interrupt Control Block
MHD	Message Handler
MT	Macrotick
MTS	Media Access Test Symbol
NCT	Network Communication Time
NEM	Network Management Block
NIT	Network Idle Time
NM	Network Management
OBF	Output Buffer
POC	Protocol Operation Control
PRT	Protocol Controller Block
SDL	Specification and Description Language
SUC	System Universal Control Block
TBF	Transient Buffer
TDMA	Time Division Multiple Access (media access method)
TSS	Transmission Start Sequence
TT-D	Time Triggered Distributed Synchronization
µT	Microtick
WUP	Wakeup Pattern
WUS	Wakeup Symbol

24.2 Overview

The V850E/PHO3 includes a FlexRay-IP called ‘E-Ray’ version R1.0, which is designed by BOSCH. The E-Ray IP-module performs communication according to the FlexRay protocol specification v2.1. With maximum specified sample clock the bitrate can be programmed to values up to 10 MBit/s. Additional bus driver (BD) hardware is required for connection to the physical layer.

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message storage consists of a single-ported Message RAM that holds up to 128 message buffers. All functions concerning the handling of messages are implemented in the Message Handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the E-Ray IP-module can be accessed directly by the V850E/PHO3. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Interrupt Control, and to access the Message RAM via Input / Output Buffer.

The HOST CPU I/F is attached to the Generic I/F and it works as synchronous external memory interface with 32 bit width. The HOST CPU I/F is limited only to 32 bit access in order to protect consistency of some registers.

The E-Ray IP-module R1.0 supports the following features:

- Conform to FlexRay protocol specification v2.1
- Conform to E-Ray specification v1.2.x
- Data rates of up to 10 Mbit/s on each channel
- Up to 128 message buffers configurable
- 8 Kbyte of Message RAM for storage of e.g.
  - 128 messages buffers with max. 48 byte data section or up to
  - 30 messages buffers with max. 254 byte data section
- Configuration of message buffers with different payload lengths possible
- One configurable receive FIFO
- Each message buffer can be configured as receive buffer, as transmit buffer or as part of the receive FIFO
- Host access to message buffers via Input and Output Buffer
  Input Buffer: Holds message to be transferred to the Message RAM
  Output Buffer: Holds message read from the Message RAM
- Filtering for slot counter, cycle counter, and channel
- Maskable module interrupts
- Network Management supported
- Register/RAM access in 32 bit width only
- Message Buffer RAM is protected by ECC and 1 bit error will be automatically corrected.

24.2.1 Block diagram

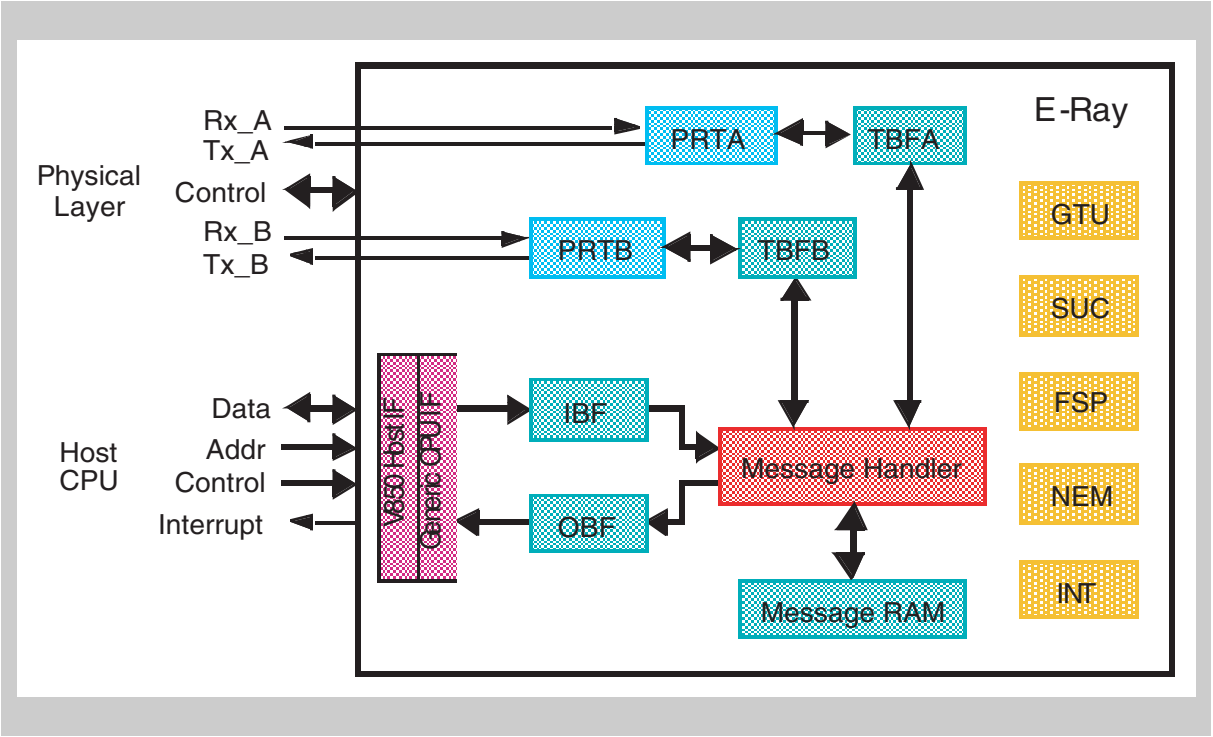


Figure 24-1 E-Ray block diagram

(1) V850 Host Interface (HIF)

The V850 host interface connects a V850 CPU to the E-Ray IP-module via the Generic CPU Interface. Configuration registers, status registers, and interrupt registers are attached to the respective blocks and can be accessed via the V850 Host Interface.

(2) Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host CPU can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

(3) Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host CPU can read the header and data section of the transferred message buffer from the Output Buffer.

(4) Message Handler (MHD)

The E-Ray Message Handler controls data transfers between the following components:

- Input / Output Buffer and Message RAM
- Transient Buffer RAMs of the two FlexRay Protocol Controllers and Message RAM

(5) Message RAM and ECC module

The Message RAM also called Message Buffer RAM (MBFRAM) consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

The MBFRAM is under protection of a ECC module. When E-Ray access the MBFRAM, or the CPU accesses the MBFRAM via the HIF, the ECC module automatically encodes/decodes the ECC code. In case of 1 bit error in reading data from the MBFRAM, the ECC module corrects it without indication. In case of a 2 bit error, the ECC module informs that to the E-Ray module and asserts the interrupt signal INTDEDFR.

- ECC (encoding/decoding) stand alone module for Message Buffer RAM
- 7 bit ECC parity code for 1 bit error correction (SEC) and 2 bits error detection (DED)
- Interrupt generation per DED
- 2 bit error detection is available when all-0 (or all-1) is read from the RAM data in case a WORD line is defected. (2 LSB of ECC parity bits are inverted when writing to the RAM and inverted again when reading from the RAM.)

Note In case of 3 or more bit errors at a time, this macro may not detect it. In addition, erroneous bit correction and/or interrupt generation may occur!

(6) TBF A/B (Transient Buffer RAM A/B)

The Transient Buffer stores the data section of two complete messages

(7) PRT A/B (FlexRay Channel Protocol Controllers)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the Transient Buffer RAMs for intermediate message storage and to the physical layer via bus driver BD.

They perform the following functionality:

- Control and check of bit timing
- Reception / transmission of FlexRay frames and symbols
- Check of header CRC
- Generation / check of frame CRC
- Interfacing to bus driver

The FlexRay Channel Protocol Controllers have interfaces to:

- Physical Layer (bus driver)
- Transient Buffer RAM
- Message Handler
- Global Time Unit
- System Universal Control
- Frame and Symbol Processing
- Network Management
- Interrupt Control

(8) Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick
- Generation of macrotick
- Fault tolerant clock synchronization by FTM algorithm
  - rate correction
  - offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

(9) System Universal Control (SUC)

The System Universal Control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation
- Monitor Mode

(10) Frame and Symbol Processing (FSP)

The Frame and Symbol Processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

(11) Network Management (NEM)

The Network Management performs the following functions:

- Handling of the network management vector

(12) Interrupt Control (INT)

The Interrupt Controller performs the following functions:

- Provides error and status interrupt flags
- Controls enable / disable of interrupt sources
- Controls assignment of interrupt sources to the two module interrupt lines
- Enable / disable of the two module interrupt lines
- Manages the two interrupt timers
- Stop watch time capturing

24.3 Host CPU Interface

- Synchronous 32bit bus interface with separated address input
- Access unit: 32bit only
- 2 state event interrupt and 2 timer interrupt signals synchronized to cluster clock
- Stop watch trigger input
- Software reset and E-Ray Macro disable control

24.3.1 Host CPU interface block diagram

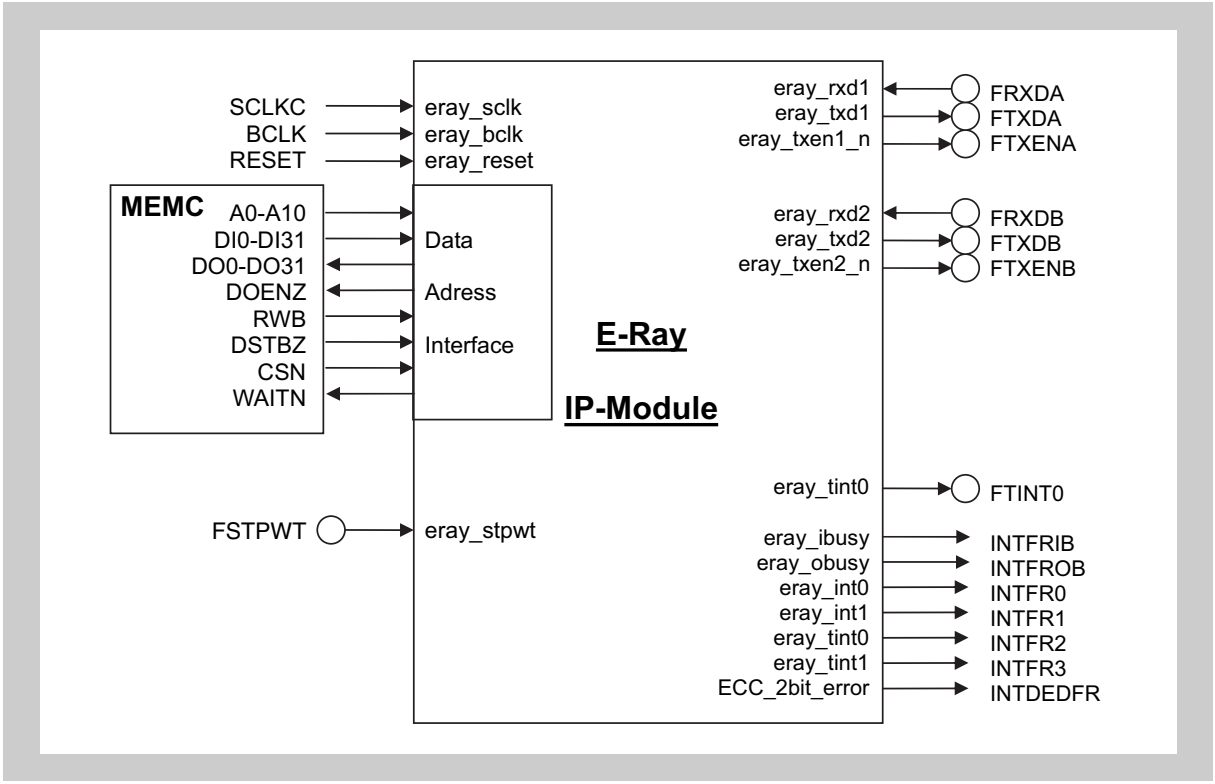


Figure 24-2 Host CPU interface block diagram

FRBCLK uses the same clock as the MEMC.

24.3.2 Signal description

(1) Host Interface

Table 24-1 Host CPU interface signals (1/2)

Signal	Type	Description	E-Ray name	Connect to
SCLKC	I	FlexRay sample clock (= 80MHz with quartz frequency 16MHz)	eray_sclk	Clock unit
BCLK	I	FlexRay macro clock (= MEMC clock, 40/32/20 MHz)	eray_bclk	Clock unit
Reset	I	E-Ray hardware reset. The hardware reset will be performed on MCU power up.	eray_reset	Reset unit
A0-10	I	Dedicated address input	-	MEMC
DI0-31	I	Data input	-	
DO0-31	O	Data output	-	
DOENZ	O	Data output enable signal	-	
RWB	I	Read / Write status	-	
DSTBZ	I	Data strobe	-	
CSN	I	MCU Chip select (CS5) E-Ray address area: 0x0F80 0000 – 0x0F80 07FF	-	
WAITN	O	Wait output signal	-	
FTINT0	O	FlexRay error/status interrupt signal 0 Interrupt handler address: 0x0550 This interrupt signal can be configured with the E-Ray registers FREILS and FRSILS.	eray_int0	Port
INTFR0	O	FlexRay error/status interrupt signal 0 Interrupt handler address: 0x0550 This interrupt signal can be configured with the E-Ray registers FREILS and FRSILS.	eray_int0	Interrupt Controller
INTFR1	O	FlexRay error/status interrupt signal 1 Interrupt handler address: 0x0560 This interrupt signal can be configured with the E-Ray registers FREILS and FRSILS.	eray_int1	Interrupt Controller
INTFR2	O	FlexRay timer 0 interrupt signal Interrupt handler address: 0x0570 This interrupt signal can be configured with the E-Ray register FRT0C.	eray_tint0	Interrupt Controller
INTFR3	O	FlexRay timer 1 interrupt signal Interrupt handler address: 0x0580 This interrupt signal can be configured with the E-Ray register FRT1C.	eray_tint1	Interrupt Controller
INTFRIB	O	FlexRay transfer IBF to MBF busy signal Interrupt handler address: 0x0590 This interrupt signal is connected to the E-Ray parameter FRIBCR.IBSYH.	eray_ibusy	Interrupt Controller

Table 24-1 Host CPU interface signals (2/2)

Signal	Type	Description	E-Ray name	Connect to
INTFROB	O	FlexRay transfer MBF to OBF busy signal Interrupt handler address: 0x0540 This interrupt signal is connected to the E-Ray parameter FROBCR.OBSYS.	eray_obusy	Interrupt Controller
FSTPWT	I	Stop watch trigger. A rising or falling edge of this signal will captured the actual time in the FRSTPW register.	eray_stpwt	Port
INTDEDFR	O	ECC error interrupt signal MCU interrupt name: INTDEDFR Interrupt handler address: 0x0010 This interrupt signal indicates a 2 bit ECC error in the E-Ray message RAM.	-	Interrupt Controller

(2) Physical Layer Interface

Table 24-2 Physical layer interface signal description

Signal	Type	Description	E-Ray name	Connect to
Channel A				
FRXDA	I	Data Receiver Input	eray_rxd1	Port
FTXDA	O	Data Transmitter Output	eray_txd1	Port
FTXENA	O	Transmit Enable Signal HIGH = transmission inactive LOW = transmission active	eray_txen1_n	Port
Channel B				
FRXDB	I	Data Receiver Input	eray_rxd2	Port
FTXDB	O	Data Transmitter Output	eray_txd2	Port
FTXENB	O	Transmit Enable Signal HIGH = transmission inactive LOW = transmission active	eray_txen2_n	Port

For each of the two channels a separate Bus Driver device is required.

24.3.3 Host CPU interface timing

The CPU read access to the RAM will take longer than to the registers. The additional wait state will be inserted by the FRWAITN signal. The write timing of the RAM or registers access is equal and consists of 3 clocks. The programmable data wait must be at least 1 bclk regardless of RAM/registers or read or write access.

Number of clock cycles in one access:

Table 24-3 Host access cycle

Access	Access cycle FRBCLK
Register/RAM write	3
Register read	3
RAM read	4

(1) Register and RAM write access

The write timing of the RAM or registers access is equal and consists of 3 clocks (is equal to 6 CPU clocks).

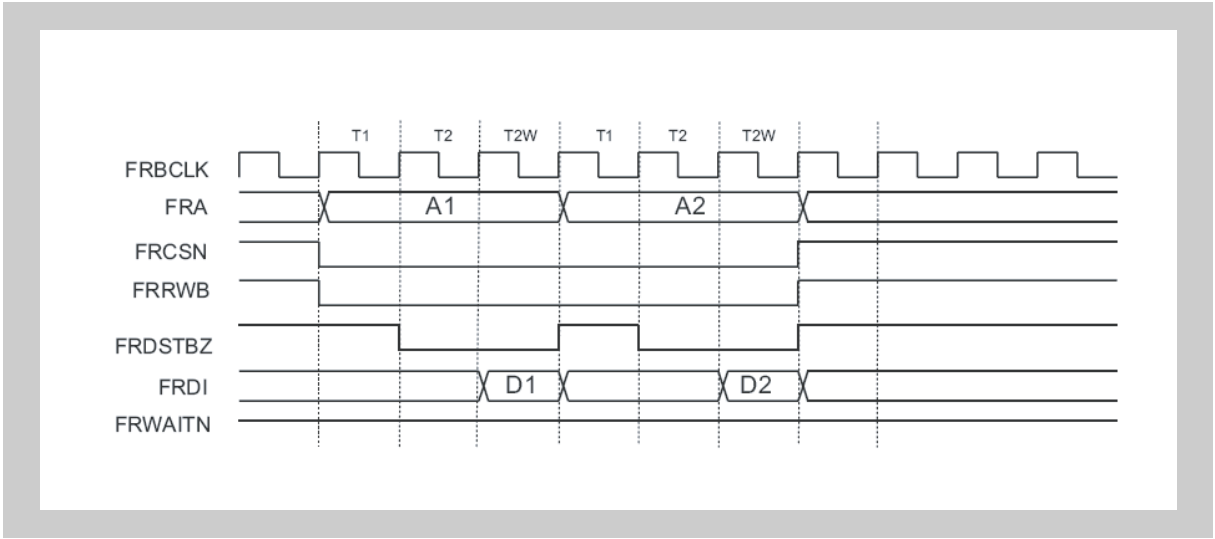


Figure 24-3 Register and RAM write access

(2) Register read access

The register read access takes 3 cycles (is equal to 6 CPU clocks).

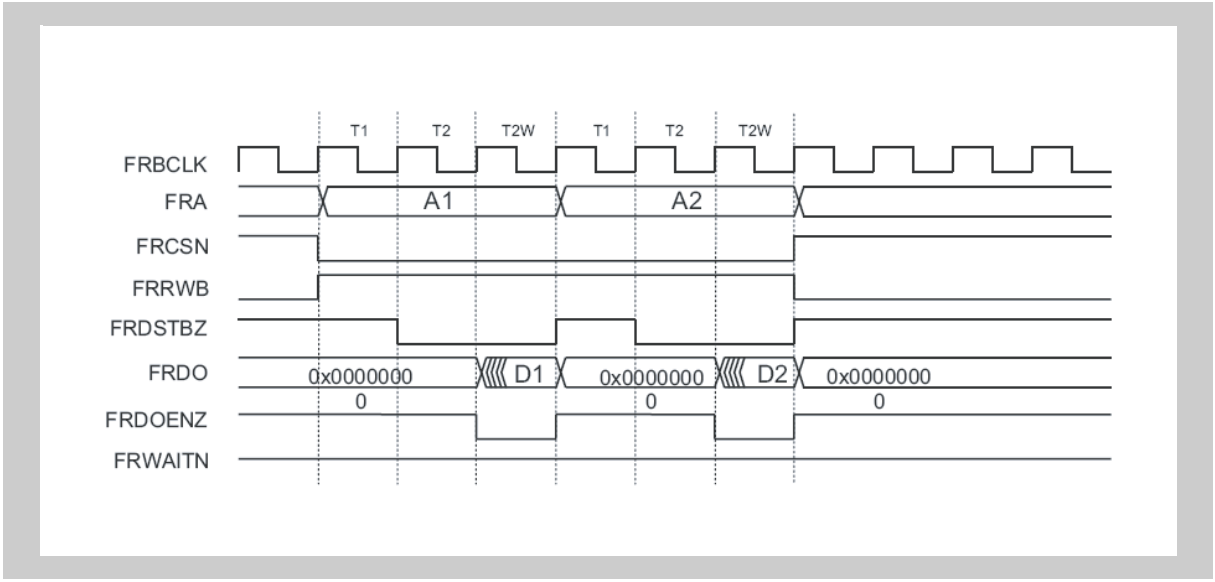


Figure 24-4 Register read access

(3) RAM read access

The RAM read access takes 4 cycles (is equal to 8 CPU clocks).

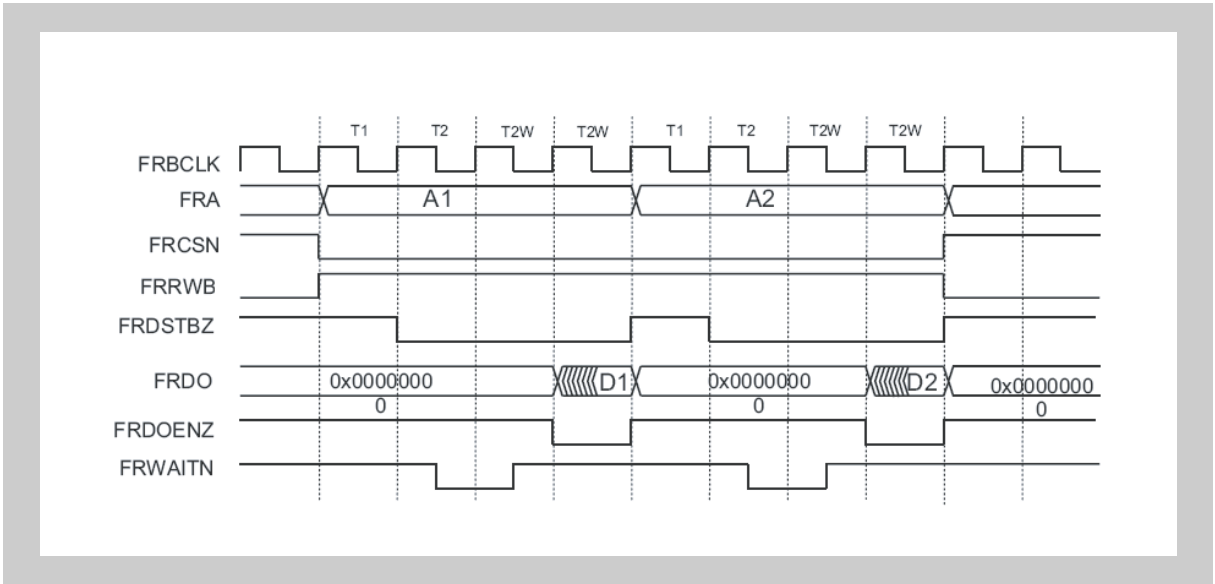


Figure 24-5 RAM read access

(4) Reset Timing

To perform a complete reset of the E-Ray module, signal **eray_reset** needs to be active for at least:

- Two **eray_bclk** cycles, clock period of **eray_bclk** . **eray_sclk**
- Two **eray_sclk** cycles, clock period of **eray_bclk** < **eray_sclk**

When leaving hard reset, an internal procedure is started that initializes the seven module-internal RAM blocks to zero. The module-internal RAMs can also be cleared by CHI command CLEAR_RAMS (**FRSUCC1.CMD[3:0]** = "1100") when the CC is in DEFAULT_CONFIG or CONFIG state. The initialization of the E-Ray internal RAM blocks requires 2048 **eray_bclk** cycles.

No Host access to IBF or OBF is possible during initialization of the internal RAM blocks after hard reset or after assertion of CHI command CLEAR_RAMS.

After hard reset all registers hold the reset values as summarized in *Table 24-5 on page 1160*. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS.

(5) Timing of IBF / OBF Transfer Busy Signals (FRIBUSY / FROBUSY)

A data transfer from Input Buffer (IBF) to the Message RAM or from Message RAM to Output Buffer (OBF) is initiated by a write access to the respective command request register (FRIBCR / FROBCR).

Signal **eray_ibusy** is activated when a write access to **FRIBCR.IBRH[6:0]** occurs while a transfer between Input Buffer Shadow and Message RAM is ongoing. This will also set bit **FRIBCR.IBSYH** to '1'. After completion of the ongoing transfer, Input Buffer Host and Input Buffer Shadow are swapped, **FRIBCR.IBSYH** is reset to '0', and **eray_ibusy** is deactivated.

Signal **eray_obusy** is active as long as a transfer between Message RAM and Output Buffer Shadow is ongoing. This is signalled by bit **FROBCR.OBSYS**.

In addition the status interrupt flags **FRSIR.TIBC** and **FRSIR.TOBC** are set whenever a data transfer has completed. If enabled, an interrupt is generated.

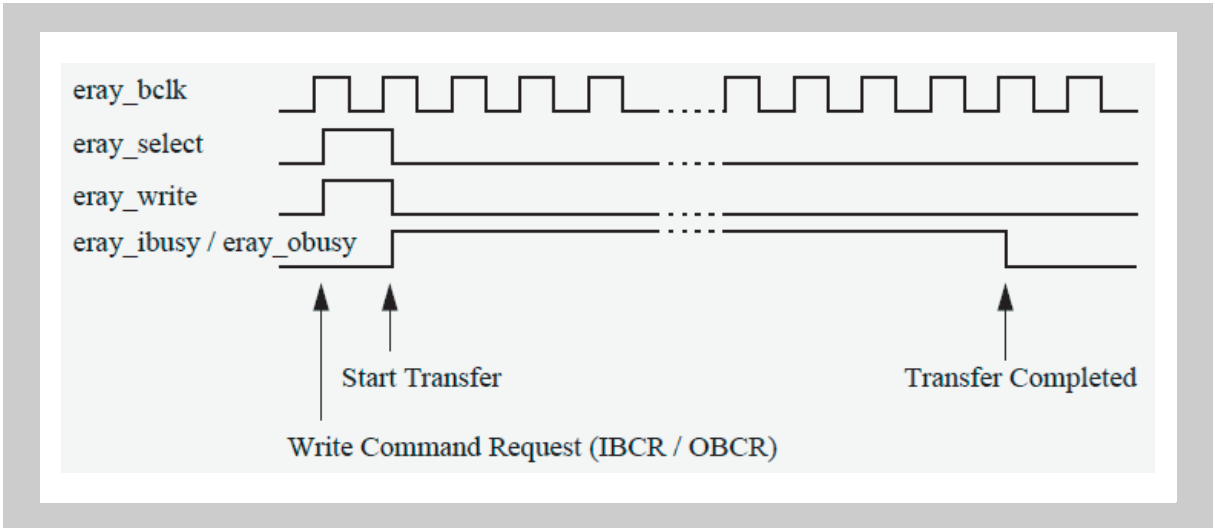


Figure 24-6 Timing of FRIBUSY and FROBUSY signal

The delay time until the respective busy signal (**eray_ibusy** or **eray_obusy**) is reset depends on the **eray_bclk** frequency, the length of the data section of the message buffer to be accessed, and the actual state of the Message Handler. A formula for calculation of this delay can be found in the Addendum to E-Ray FlexRay IP-Module Specification.

24.3.4 CPU, MEMC, Port and FlexRay IP register settings

Please notice that only FlexRay relevant parameters have been considers in this chapter. Parameters for other macros and features than FlexRay have be set additionally.

(1) Bus and memory control register

As for the required settings of the Bus and Memory Control refer to the chapter “Bus and Memory Control (BCU, MEMC)”.

(2) Port register

As for the required settings of the port control registers refer to the chapter “Pin Functions”.

(3) FRECCTL - ECC Control Register for E-Ray Message RAM

In case of 1 bit error in reading data from the MBFRAM, the ECC module corrects it without indication. In case of a 2 bit error, the ECC module informs that to the E-Ray module and asserts the interrupt signal INTDEDFR.

Bit		15	14	13	12	11	10	9	8
FRECCTL	R	0	0	0	0	0	ER2C	0	0
0x0F80 0800	W								
		0	0	0	0	0	0	0	0

Bit		7	6	5	4	3	2	1	0
	R	0	0	0	0	0	ECCER2	0	0
	W								
Reset		0	0	0	0	0	0	0	0

ECCER2

– 2 bits ECC error detection flag bit

This bit indicates the detection of a 2 bits error when message data is read.

When this flag is set to 1, interrupt signal (INTDEDFR) is generated. However, INTDEDFR is generated only once. This means subsequent INTDEDFR is suppressed even if further 2 bits error is detected as long as this bit remain set. Generation of INTDEDFR can be resumed by clearing this bit. This bit is read only and cleared by setting ER2C bit [bit10] to 1.

ECCER2	Description
0	No 2 bits error has been detected since this bit is cleared last time clear condition: 1) power up reset 2) by writing ER2C=1
1	2 bit error has been detected since this bit is cleared last time

Note 3 or more bits errors at a time may not be detected. In addition, erroneous bit correction and/or interrupt generation may occur!

ER2C

– ECC 2 bit error flag clear bit

This bit is used to clear the ECCER2 bit by setting this bit to 1. ER2C does not hold written data thus 0 is always seen in case of a read access.

Note The user must execute the following flow after detecting an ECC 2 bits error in the message buffer:

3. Reconfigure the erroneous message buffer header which is indicated by parameter FRMHDS.FMB
4. Clear bits FRMHDS.FMBD and FRMHDS.PMR
5. Clear bit FRECCTL.ECCER2 by writing "1" to bit FRECCTL.ER2C

24.4 Programmer's Model

24.4.1 Register map

The E-Ray module allocates an address space of 2 Kbytes (0x0000 to 0x07FF). The registers are organized as 32-bit registers. Host access to the Message RAM is done via the Input and Output Buffers. They buffer data to be transferred to and from the Message RAM under control of the Message Handler, avoiding conflicts between Host accesses and message reception / transmission.

The assignment of the message buffers is done according to the scheme shown in *Table 24-4* below. The number N of available message buffers depends on the payload length of the configured message buffers. The maximum number of message buffers is 128. The maximum payload length supported is 254 bytes.

The message buffers are separated into three consecutive groups:

Static Buffers	Transmit / receive buffers static segment
Static + Dynamic Buffers	Transmit / receive buffers static + dynamic segment
FIFO	Receive FIFO

The message buffer separation configuration can be changed in DEFAULT_CONFIG or CONFIG state only by programming the Message RAM Configuration register.

The first group starts with message buffer 0 and consists of static message buffers only. Message buffer 0 is dedicated to hold the startup / sync frame or the single slot frame, if the node transmits one, as configured by **TXST**, **TXSY**, and **TSM** in the SUC Configuration Register 1. In this case message buffer 0 has to be configured with the key slot ID and can only be (re)configured in DEFAULT_CONFIG or CONFIG state.

The second group consists of message buffers assigned to the static or dynamic segment.

The message buffers belonging to the third group are concatenated to a single receive FIFO.

Table 24-4 Assignment of message buffers

Message buffer 0	⇓ Static buffers
Message buffer 1	
…	⇓ Static + Dynamic Buffers
	⇓ FIFO
Message buffer N-1	
Message buffer N	

Address Area 0x0F80 0000 - 0x0F80 07FF

Table 24-5 E-Ray register map (1/3)

Offset address	Symbol	Name	Page	Reset	Acc	Block
V850 E-Ray registers						
0x0000	FRCI	Controller Information	page 1164	0x6572 6179	r	VIF
0x0004	FRVI	Vendor Information	page 1165	0x0104 0003	r	
0x0008	FRCS	Control Setting	page 1166	0x0000 0000	r/w	
0x000C		*reserved (1)*	-	0x0000 0000	r	
Special registers						
0x0010	FRTEST1	Test Register 1	page 1168	0x0000 0300	r/w	GIF
0x0014		*reserved (1)*		0x0000 0000	r	
0x0018		*reserved (1)*		0x0000 0000	r	
0x001C	FRLCK	Lock Register	page 1172	0x0000 0000	r/w	GIF
Interrupt registers						
0x0020	FREIR	Error Interrupt Register	page 1173	0x0000 0000	r/w	INT
0x0024	FRSIR	Status Interrupt Register	page 1177	0x0000 0000	r/w	
0x0028	FREILS	Error Interrupt Line Select	page 1180	0x0000 0000	r/w	
0x002C	FRSILS	Status Interrupt Line Select	page 1182	0x0303 FFFF	r/w	
0x0030	FREIES	Error Interrupt Enable Set	page 1184	0x0000 0000	r/w	
0x0034	FREIER	Error Interrupt Enable Reset	page 1184	0x0000 0000	r/w	
0x0038	FRSIES	Status Interrupt Enable Set	page 1186	0x0000 0000	r/w	
0x003C	FRSIER	Status Interrupt Enable Reset	page 1186	0x0000 0000	r/w	
0x0040	FRILE	Interrupt Line Enable	page 1188	0x0000 0000	r/w	
0x0044	FRT0C	Timer 0 Configuration	page 1189	0x0000 0000	r/w	
0x0048	FRT1C	Timer 1 Configuration	page 1190	0x0002 0000	r/w	
0x004C	FRSTPW1	Stop Watch Register 1	page 1191	0x0000 0000	r/w	
0x0050	FRSTPW2	Stop Watch Register 2		0x0000 0000	r/w	
0x0054 - 0x007C		*reserved (11)*		0x0000 0000	r	
CC control registers						
0x0080	FRSUCC1	SUC Configuration Register 1	page 1194	0x0C40 1080	r/w	SUC
0x0084	FRSUCC2	SUC Configuration Register 2	page 1199	0x0100 0504	r/w	
0x0088	FRSUCC3	SUC Configuration Register 3	page 1200	0x0000 0011	r/w	
0x008C	FRNEMC	NEM Configuration Register	page 1201	0x0000 0000	r/w	NEM
0x0090	FRPRTC1	PRT Configuration Register 1	page 1202	0x084C 0633	r/w	PRT
0x0094	FRPRTC2	PRT Configuration Register 2	page 1203	0x0F2D 0A0E	r/w	
0x0098	FRMHDC	MHD Configuration Register	page 1204	0x0000 0000	r/w	MHD
0x009C		*reserved (1)*		0x0000 0000	r	
0x00A0	FRGTUC1	GTU Configuration Register 1	page 1204	0x0000 0280	r/w	GTU
0x00A4	FRGTUC2	GTU Configuration Register 2	page 1205	0x0002 000A	r/w	
0x00A8	FRGTUC3	GTU Configuration Register 3	page 1206	0x0202 0000	r/w	
0x00AC	FRGTUC4	GTU Configuration Register 4	page 1207	0x0008 0007	r/w	
0x00B0	FRGTUC5	GTU Configuration Register 5	page 1208	0x0E00 0000	r/w	
0x00B4	FRGTUC6	GTU Configuration Register 6	page 1209	0x0002 0000	r/w	

Table 24-5 E-Ray register map (2/3)

Offset address	Symbol	Name	Page	Reset	Acc	Block
0x00B8	FRGTUC7	GTU Configuration Register 7	page 1210	0x0002 0004	r/w	GTU
0x00BC	FRGTUC8	GTU Configuration Register 8	page 1210	0x0000 0002	r/w	
0x00C0	FRGTUC9	GTU Configuration Register 9	page 1211	0x0000 0101	r/w	
0x00C4	FRGTUC10	GTU Configuration Register 10	page 1212	0x0002 0005	r/w	
0x00C8	FRGTUC11	GTU Configuration Register 11	page 1213	0x0000 0000	r/w	
0x00CC - 0x00FC		*reserved (13)*		0x0000 0000	r	
CC status registers						
0x0100	FRCCSV	CC Status Vector	page 1214	0x0010 4000	r	SUC
0x0104	FRCCEV	CC Error Vector	page 1217	0x0000 0000	r	
0x0108 - 0x010C		reserved (2)		0x0000 0000	r	
0x0110	FRSCV	Slot Counter Value	page 1218	0x0000 0000	r	GTU
0x0114	FRMTCCV	Macrotick and Cycle Counter Value	page 1218	0x0000 0000	r	
0x0118	FRRCV	Rate Correction Value	page 1219	0x0000 0000	r	
0x011C	FROCV	Offset Correction Value	page 1219	0x0000 0000	r	
0x0120	FRSFS	Sync Frame Status	page 1220	0x0000 0000	r	
0x0124	FRSWNIT	Symbol Window and NIT Status	page 1222	0x0000 0000	r	
0x0128	FRACS	Aggregated Channel Status	page 1224	0x0000 0000	r/w	
0x012C		*reserved (1)*		0x0000 0000	r	
0x0130 - 0x0168	FRESIDn	Even Sync ID [1...15]	page 1226	0x0000 0000	r	GTU
0x016C		*reserved (1)*		0x0000 0000	r	
0x0170 - 0x01A8	FROSIDn	Odd Sync ID [1...15]	page 1227	0x0000 0000	r	GTU
0x01AC		*reserved (1)*		0x0000 0000	r	
0x01B0 - 0x01B8	FRNMVn	Network Management Vector [1...3]	page 1228	0x0000 0000	r	NEM
0x01BC - 0x02FC		reserved (81)		0x0000 0000	r	
Message buffer control registers						
0x0300	FRMRC	Message RAM Configuration	page 1229	0x0180 0000	r/w	MHD
0x0304	FRFRF	FIFO Rejection Filter	page 1231	0x0180 0000	r/w	
0x0308	FRFRFM	FIFO Rejection Filter Mask	page 1232	0x0000 0000	r/w	
0x030C	FCL	*FIFO Critical Level*	page 1232	0x0000 0080	r/w	
Message buffer status registers						
0x0310	FRMHDS	Message Handler Status	page 1233	0x0000 0080	r/w	MHD
0x0314	FRLDTS	Last Dynamic Transmit Slot		0x0000 0000	r	
0x0318	FRFSR	FIFO Status Register		0x0000 0000	r	
0x031C	FRMHDF	Message Handler Constraints Flags		0x0000 0000	r/w	
0x0320	FRTXRQ1	Transmission Request 1	page 1238	0x0000 0000	r	
0x0324	FRTXRQ2	Transmission Request 2	page 1238	0x0000 0000	r	
0x0328	FRTXRQ3	Transmission Request 3	page 1238	0x0000 0000	r	

Table 24-5 E-Ray register map (3/3)

Offset address	Symbol	Name	Page	Reset	Acc	Block
0x032C	FRTXRQ4	Transmission Request 4	page 1238	0x0000 0000	r	MHD
0x0330	FRNDAT1	New Data 1	page 1240	0x0000 0000	r	
0x0334	FRNDAT2	New Data 2	page 1240	0x0000 0000	r	
0x0338	FRNDAT3	New Data 3	page 1240	0x0000 0000	r	
0x033C	FRNDAT4	New Data 4	page 1240	0x0000 0000	r	
0x0340	FRMBSC1	Message Buffer Status Changed 1	page 1242	0x0000 0000	r	
0x0344	FRMBSC2	Message Buffer Status Changed 2	page 1242	0x0000 0000	r	
0x0348	FRMBSC3	Message Buffer Status Changed 3	page 1242	0x0000 0000	r	
0x034C	FRMBSC4	Message Buffer Status Changed 4	page 1242	0x0000 0000	r	
0x0350 - 0x03EC		*reserved (40)*		0x0000 0000	r	
Identification Registers						
0x03F0	FRCREL	Core Release Register	page 1244	0x1006 0519	r	GIF
0x03F4	FRENDN	Endian Register	page 1245	0x8765 4321	r	
0x03F8 - 0x03FC		reserved (2)		0x0000 0000	r	
Input buffer						
0x0400 - 0x04FC	FRWRDSn	Write Data Section [1...64]	page 1246	0x0000 0000	r/w	IBF
0x0500	FRWRHS1	Write Header Section 1	page 1247	0x0000 0000	r/w	
0x0504	FRWRHS2	Write Header Section 2	page 1249	0x0000 0000	r/w	
0x0508	FRWRHS3	Write Header Section 3	page 1249	0x0000 0000	r/w	
0x050C		*reserved (1)*		0x0000 0000	r/w	
0x0510	FRIBCM	Input Buffer Command Mask	page 1250	0x0000 0000	r/w	
0x0514	FRIBCR	Input Buffer Command Request	page 1251	0x0000 0000	r/w	
0x0518 - 0x05FC		*reserved (58)*		0x0000 0000	r	
Output buffer						
0x0600 - 0x06FC	FRRDDSn	Read Data Section [1...64]	page 1253	0x0000 0000	r	OBF
0x0700	FRRDHS1	Read Header Section 1	page 1254	0x0000 0000	r	
0x0704	FRRDHS2	Read Header Section 2	page 1255	0x0000 0000	r	
0x0708	FRRDHS3	Read Header Section 3	page 1256	0x0000 0000	r	
0x070C	FRMBS	Message Buffer Status	page 1257	0x0000 0000	r	
0x0710	FROBCM	Output Buffer Command Mask	page 1260	0x0000 0000	r/w	
0x0714	FROBCR	Output Buffer Command Request	page 1261	0x0000 0000	r/w	
0x0718 - 0x07FC		*reserved (58)*		0x0000 0000	r	

The MCU base address of the FlexRay macro is 0x0F80 0000 (MCU E-Ray address area: 0x0F80 0000 – 0x0F80 07FF).

Note The FlexRay address space is mirrored after each 4kB in the address area 0x0F80 0000 - 0x0FBF FFFF. It is strongly requested only to use the first address space 0x0F80 0000 – 0x0F80 07FF to access the FlexRay register and RAM!

24.4.2 V850 E-Ray registers

(1) Controller information (FRCI)

This register contains FlexRay controller information.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRCI	R	MN31	MN30	MN29	MN28	MN27	MN26	MN25	MN24	MN23	MN22	MN21	MN20	MN19	MN18	MN17	MN16
0x0000	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Reset		0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MN15	MN14	MN13	MN12	MN11	MN10	MN9	MN8	MN7	MN6	MN5	MN4	MN3	MN2	MN1	MN0
	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Reset		0	1	1	0	0	0	0	1	0	1	1	1	1	0	0	1

Bit name	Function
MN[31:0]	Magic Number This field contains a code which shows the name of the Bosch IP. The value is "0x65726179" which shows "eray" as ASCII-codes.

(2) Vendor information (FRVI)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRVI	R	VI7	VI6	VI5	VI4	VI3	VI2	VI1	VI0	FMR7	FMR6	FMR5	FMR4	FMR3	FMR2	FMR1	FMR0
0x0004	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	PCN7	PCN6	PCN5	PCN4	PCN3	PCN2	PCN1	PCN0
	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit name	Function
VI[7:0]	Vendor ID This field contains a code which shows the producer of the macro. NEC is "0x01".
FMR[7:0]	FlexRay Macro Release Number This field contains a code which shows the current release number of the IP.
PCN[7:0]	Product Code Number This field contains a code which shows the chip-product. Actual value will be added in a later version of this document.

Vendor information bit vector value

Bit vector	V850E/CAG4-M
VI	0x01
FMR	0x04
PCN[7:0]	0x03

(3) Control settings (FRCS)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRCS	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	W	CSLK7	CSLK6	CSLK5	CSLK4	CSLK3	CSLK2	CSLK1	CSLK0	-	-	-	-	-	-	-	-
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MD	SR
	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
SR	Software Reset Software Reset bit. This bit is connected to the E-Ray reset line. If this bit is set, a reset will be executed immediately. The user must clear SR bit in order to release a reset. Unlock sequence is required for setting and clearing SR bit. Software reset does not affect to host IF logic itself.
MD	Macro Disable Macro disable bit. This bit is connected to the clock module. If this bit is set, internal clock distribution (eray_sclk, eray_bclk) is halted. Therefore software can NOT access registers/RAM of FlexRay Macro when MD=1. In case of that, Write access will be simply ignored and Read access will get incorrect value. Exceptionally V850 E-Ray registers (i.e. FRCS / FRVI / FRCS / FRTESTC) can be accessed even when MD=1.Unlock sequence is required for setting and clearing MD bit. Macro disable bit does not affect to host IF logic itself. The user must set MD only in CONFIG state and must wait a transfer between RAMs to complete. Otherwise the last received message can be destroy due to sudden clock stopping. Note that the read access to FRTEST1 during MD=1 doesn't cause clearing FRTEST1.CERA and CERB.
CSLK[7:0]	Control Setting Lock Key To set and clear SR bit and MD bit, the write operation has to be directly preceded by two consecutive write accesses to the Control Setting Lock Key. If the host access to other E-Ray's register in between this consecutive sequence, writing FRCS register is simply ignored.

	set SR bit	clear SR bit	set MD bit	clear MD bit
First write to FRCS Register	0x67xx_xxxx	0x67xx_xxxx	0x67xx_xxxx	0x67xx_xxxx
Second write to FRCS Register	0x98xx_xxx1	0x98xx_xxx0	0x98xx_xxx2	0x98xx_xxx0

Accessing to E-RAY registers and RAM is restricted by the combination of SR and MD bits.

SR	MD	CIF registers (FRCI, FRVI, FRCS, FRTESTC)	E-Ray registers	E-Ray RAM
0	0	R/W possible	R/W possible	R/W possible
0	1	R/W possible	Read only**	NOT possible*
1	0	R/W possible	NOT possible*	NOT possible*
1	1	R/W possible	NOT possible*	NOT possible*

Note * Read access: capture invalid data, Write access: simply ignored
** Write access: simply ignored

The table below explains the possible bit transition of SR and MD bits.

SR	MD	Bit transition
$0 \rightarrow 1$	0	allowed
0	$0 \rightarrow 1$	allowed
$0 \rightarrow 1$	$0 \rightarrow 1$	ignored
1	$0 \rightarrow 1$	allowed
$0 \rightarrow 1$	1	ignored
$0 \rightarrow 1$	$1 \rightarrow 0$	allowed

There is no restriction to clear SR and MD bits.

24.4.3 Special registers

(1) Test register 1 (FRTEST1)

The Test Register 1 holds the control bits to configure the test modes of the E-Ray module. Write access to these bits is only possible if bit **WRTEN** is set to ’1’.

When the E-Ray IP is operated in one of its test modes that requires **WRTEN** to be set (RAM Test Mode, I/O Test Mode, Asynchronous Transmit Mode, and Loop Back Mode) only the selected test mode functionality is available.

The test functions are not available in addition to the normal operational mode functions, they change the functions of parts of the E-Ray module. Therefore normal operation as specified outside this chapter and as required by the FlexRay protocol specification and the FlexRay conformance test is not possible. Test mode functions may not be combined with each other or with FlexRay protocol functions.

The test mode features are intended for hardware testing or for FlexRay bus analyzer tools. They are not intended to be used in FlexRay applications.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRTEST1	R	CERB3	CERB2	CERB1	CERB0	CERA3	CERA2	CERA1	CERA0	0	0	TXENB	TXENA	TXB	TXA	RXB	RXA
0x0010	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	AOB	AOA	0	0	0	0	0	0	ELBE	WRTEN
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
WRTEN	Write Test Register Enable Enables write access to the test registers. To set the bit from ’0’ to ’1’ the test mode key has to be written as defined in section Lock Register (FRLCK). The unlock sequence is not required when WRTEN is kept at ’1’ while other bits of the register are changed. The bit can be reset to ’0’ at any time. 1 =Write access to the Test Register is enabled 0 =Write access to the Test Register is disabled
ELBE	External Loop Back Enable There are two possibilities to perform a loop back test. External loop back via physical layer or internal loop back for in-system self-test (default). In case of an internal loop back pins **eray_txen1,2_n** are in their inactive state, pins **eray_txd1,2** are set to HIGH, pins **eray_rxd1,2** are not evaluated. Bit **ELBE** is evaluated only when POC is in loop back mode and test multiplexer control is in non-multiplexing mode **TMC[1:0]** = "00". 1 = External loop back 0 = Internal loop back (default)

Bit name	Function
AOA	Activity on A **AOA** is set when there is activity on channel A. It is reset when 11 consecutive '1' bits are detected or if the POC state is DEFAULT_CONFIG or CONFIG. During STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE the function of **AOA** is the inverse of zChannelIdle as specified in the FlexRay protocol spec v2.1, chapter 3, BITSTRB process. **AOA** should be ignored in all other POC states. 1 = Activity detected, channel A not idle 0 = No activity detected, channel A idle
AOB	Activity on B **AOB** is set when there is activity on channel B. It is reset when 11 consecutive '1' bits are detected or if the POC state is DEFAULT_CONFIG or CONFIG. During STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE the function of **AOB** is the inverse of zChannel-Idle as specified in the FlexRay protocol spec v2.1, chapter 3, BITSTRB process. **AOB** shouldbe ignored in all other POC states.1 = Activity detected, channel B not idle 0 = No activity detected, channel B idle
CERA[3:0]	Coding Error Report Channel A Set when a coding error is detected on channel A. Reset to zero when register **FRTEST1** is read or written. Once the **CERA[3:0]** is set it will remain unchanged until **FRTEST1** register is accessed. 0000 = No coding error detected 0001 = Header CRC error detected 0010 = Frame CRC error detected 0011 = Frame Start Sequence FSS too long 0100 = First bit of Byte Start Sequence BSS seen LOW 0101 = Second bit of Byte Start Sequence BSS seen HIGH 0110 = First bit of Frame End Sequence FES seen HIGH 0111 = Second bit of Frame End Sequence FES seen LOW 1000 = CAS / MTS symbol seen too short 1001 = CAS / MTS symbol seen too long 1010...1111 = reserved
CERB[3:0]	Coding Error Report Channel B Set when a coding error is detected on channel B. Reset to zero when register **FRTEST1** is read or written. Once the **CERB[3:0]** is set it will remain unchanged until **FRTEST1** register is accessed. 0000 = No coding error detected 0001 = Header CRC error detected 0010 = Frame CRC error detected 0011 = Frame Start Sequence FSS too long 0100 = First bit of Byte Start Sequence BSS seen LOW 0101 = Second bit of Byte Start Sequence BSS seen HIGH 0110 = First bit of Frame End Sequence FES seen HIGH 0111 = Second bit of Frame End Sequence FES seen LOW 1000 = CAS / MTS symbol seen too short 1001 = CAS / MTS symbol seen too long 1010...1111 = reserved

Note Coding errors are signalled in all states where frame decoding is possible. **CERA[3:0]** and **CERB[3:0]** should be ignored in all other states. The error codes regarding CAS / MTS symbols concern only the monitored bit pattern, irrelevant whether those bit patterns are seen in the symbol window or elsewhere.

The following Test Register 1 bits are used to test the interface to the physical layer (connectivity test) by driving / reading the respective pins.

Bit name	Function
RXA	Monitor Channel A Receive Pin 0 = **eray_rxd1** = ’0’ 1 = **eray_rxd1** = ’1’ The RXA pin has to be fixed to high or low, otherwise the value is undefined.
RXB	Mon itor Channel B Receive Pin 0 = **eray_rxd2** = ’0’ 1 = **eray_rxd2** = ’1’ The RXB pin has to be fixed to high or low, otherwise the value is undefined.
TXA	Control of Channel A Transmit Pin 0 = **eray_txd1** pin drives a ’0’ 1 = **eray_txd1** pin drives a ’1’
TXB	Control of Channel B Transmit Pin 0 = **eray_txd2** pin drives a ’0’ 1 = **eray_txd2** pin drives a ’1’
TXENA	Control of Channel A Transmit Enable Pin 0 = **ray_txen1_n** pin drives a ’0’ 1 = **eray_txen1_n** pin drives a ’1’
TXENB	Control of Channel B Transmit Enable Pin 0 = **eray_txen2_n** pin drives a ’0’ 1 = **eray_txen2_n** pin drives a ’1’

Asynchronous Transmit Mode (ATM)

The asynchronous transmit mode is entered by writing **FRSUCC1.CMD[3:0]** = "1110" while the CC is in CONFIG state and bit **FRTEST1.WRTEN** is set to ’1’. This write operation has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (unlock sequence). When called in any other state or when bit **FRTEST1.WRTEN** is not set, **FRSUCC1.CMD[3:0]** will be reset to "0000" = command_not_accepted. Reading **FRCCSV.POCS[5:0]** will return "00 1110" while the E-Ray module is in ATMmode. Asynchronous transmit mode can be left by writing **FRSUCC1.CMD[3:0]** = "0001" (CHI command: CONFIG).

In ATM mode transmission of a FlexRay frame is triggered by writing the number of the respective message buffer to **FRIBCR.IBRH[6:0]** while **FRIBCM.STXR** is set to ’1’. In this mode wakeup, startup, and clock synchronization are bypassed. The CHI command SEND_MTS results in the immediate transmission of an MTS symbol.

The cycle counter value of frames send in ATM mode can be programmed via **FRMTCCV.CCV[5.0]** (writeable in ATM and loop back mode only).

Loop Back Mode

The loop back mode is entered by writing **FRSUCC1.CMD[3:0]** = "1111" while the CC is in CONFIG state and bit **FRTEST1.WRTEN** is set to ’1’. This write operation has to be directly preceded by two consecutive write accesses to the Configuration Lock Key (unlock sequence). When called in any other state or when **FRTEST1.WRTEN** is not set, **FRSUCC1.CMD[3:0]** will be reset to "0000" = command_not_accepted. Reading **FRCCSV.POCS[5:0]** will return "00 1101" while the E-Ray module is in loop back mode.

Loop back mode can be left by writing **FRSUCC1.CMD[3:0]** = "0001" (CHI command: CONFIG).

The loop back mode is intended to check the module’s internal data paths. Normal, time triggered operation is not possible in loop back mode.

There are two possibilities to perform a loop back test. External loop back via physical layer (**FRTEST1.ELBE** = ’1’) or internal loop back for in-system self-test (**FRTEST1.ELBE** = ’0’). In case of an internal loop back pins **eray_txen1,2_n** are in their inactive state, pins **eray_txd1,2** are set to HIGH, pins **eray_rxd1,2** are not evaluated.

When the CC is in loop back mode, a loop back test is started by the Host writing a message to the Input Buffer and requesting the transmission by writing to register FRIBCR. The Message Handler will transfer the message into the Message RAM and then into the Transient Buffer of the selected channel. The Channel Protocol Controller (PRT) will read (in 32-bit words) the message from the transmit part of the Transient Buffer and load it into its Rx / Tx shift register. The serial transmission is looped back into the shift register; its content is written into the receive part of the channels’s Transient Buffer before the next word is loaded.

The PRT and the Message Handler will then treat this transmitted message like a received message, perform an acceptance filtering on frame ID and receive channel, and store the message into the Message RAM if it passed acceptance filtering. The loop back test ends with the Host requesting this received message from the Message RAM and then checking the contents of the Output Buffer.

Each FlexRay channel is tested separately. The E-Ray cannot receive messages from the FlexRay bus while it is in the loop back mode.

The cycle counter value of frames used in loop back mode can be programmed via **FRMTCCV.CCV[5.0]** (writeable in ATM and loop back mode only).

Note that in case of an odd payload the last two bytes of the looped-back payload will be shifted by 16 bits to the right inside the last 32-bit data word.

(2) Lock register (FRLCK)

The Lock Register is write-only. Reading the register will return 0x0000.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRLCK	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W	TMK7	TMK6	TMK5	TMK4	TMK3	TMK2	TMK1	TMK0	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
CLK[7:0]	Configuration Lock Key To leave CONFIG state by writing **FRSUCC1.CMD[3:0]** (commands READY, MONITOR_MODE, ATM, LOOP_BACK), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the SUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated. First write: **FRLCK.CLK[7:0]** = "1100 1110" (0xCE) Second write: **FRLCK.CLK[7:0]** = "0011 0001" (0x31) Third write: **FRSUCC1.CMD[3:0]**
TMK[7:0]	Test Mode Key To write bit **FRTEST1.WRTEN**, the write operation has to be directly preceded by two consecutive write accesses to the Test Mode Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Test Mode Key and the write access to the TEST1 register, **FRTEST1.WRTEN** is not set to '1' and the sequence has to be repeated. First write: **FRLCK.TMK[7:0]** = "0111 0101" (0x75) Second write: **FRLCK.TMK[7:0]** = "1000 1010" (0x8A) Third write: **FRTEST1.WRTEN = 1**

24.4.4 Interrupt registers

(1) Error interrupt register (FREIR)

The flags are set when the CC detects one of the listed error conditions. They remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hard reset will also clear the register.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FREIR	R	0	0	0	0	0	TABB	LTVB	EDB	0	0	0	0	0	TABA	LTVA	EDA
0x0020	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHF	IOBA	IIBA	EFA	RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
PEMC	POC Error Mode Changed This flag is set whenever the error mode signalled by **FRCCEV.ERRM[1:0]** has changed. 1 = Error mode has changed 0 = Error mode has not changed
CNA	Command Not Accepted The flag signals that the write access to the CHI command vector **FRSUCC1.CMD[3:0]** was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (**CCL** = '1'). 1 =CHI command not accepted 0 =CHI command accepted
SFBM	Sync Frames Below Minimum This flag signals that the number of sync frames received during the last communication cycle was below the limit required by the FlexRay protocol. May be set during startup and therefore should be cleared by the Host after the CC entered NORMAL_ACTIVE state. 1 = Less than the required minimum of sync frames received 0 = Sync node: 1 or more sync frames received Non-sync node:2 or more sync frames received
SFO	Sync Frame Overflow Set when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by **FRGTUC2.SNM[3:0]**. 1 = More sync frames received than configured by **FRGTUC2.SNM[3:0]** 0 = Number of received sync frames **FRGTUC2.SNM[3:0]**

Bit name	Function
CCF	Clock Correction Failure This flag is set at the end of the cycle whenever one of the following errors occurred: • Missing offset and / or rate correction • Clock correction limit reached The clock correction status is monitored in registers **FRCCEV** and **FRSFS**. A failure may occur during startup, therefore bit **CCF** should be cleared by the Host after the CC entered NORMAL_ACTIVE state. 1 = Clock correction failed 0 = No clock correction error
CCL	CHI Command Locked The flag signals that the write access to the CHI command vector **FRSUCC1.CMD[3:0]** was not successful because the execution of the previous CHI command has not yet completed. In this case bit **CNA** is also set to '1'. 1 = CHI command not accepted 0 = CHI command accepted
PERR	Parity Error The flag signals a parity error to the Host. It is set whenever one of the flags **FRMHDS.PIBF**, **FRMHDS.POBF**, **FRMHDS.PTBF1**, **FRMHDS.PTBF2** changes from '0' to '1'. 1 = Parity error detected 0 = No parity error detected
RFO	Receive FIFO Overrun The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register **FRFSR**. 1 = A receive FIFO overrun has been detected 0 = No receive FIFO overrun detected
EFA	Empty FIFO Access This flag is set by the CC when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty. 1 = Host access to empty FIFO occurred 0 = No Host access to empty FIFO occurred

Bit name	Function
IIBA	Illegal Input Buffer Access This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies: 1) The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the • Header section of message buffer 0, 1 if configured for transmission in key slot • Header section of static message buffers with buffer number < **FRMRC.FDB[7:0]** while **FRMRC.SEC[1:0]** = "01" • Header section of any static or dynamic message buffer while **FRMRC.SEC[1:0]** = "1x" • Header and / or data section of any message buffer belonging to the receive FIFO 2) The Host writes to any register of the Input Buffer while **FRIBCR.IBSYH** is set to '1'. 1 = Illegal Host access to Input Buffer occurred 0 = No illegal Host access to Input Buffer occurred
IOBA	Illegal Output buffer Access This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while **FROBCR.OBSYS** is set to '1'. 1 = Illegal Host access to Output Buffer occurred 0 = No illegal Host access to Output Buffer occurred
MHF	Message Handler Constraints Flag The flag signals a Message Handler constraints violation condition. It is set whenever one of the flags **FRMHDF.SNUA**, **FRMHDF.SNUB**, **FRMHDF.FNFA**, **MHDF.FNFB**, **FRMHDF.TBFA**, **FRMHDF.TBFB**, **FRMHDF.WAHP** changes from '0' to '1'. 1 = Message Handler failure detected 0 = No Message Handler failure detected

Channel-specific error flags channel A:

Bit name	Function
EDA	Error Detected on Channel A This bit is set whenever one of the flags **FRACS.SEDA**, **FRACS.CEDA**, **FRACS.CIA**, **FRACS.SBVA** changes from '0' to '1'. 1 =Error detected on channel A 0 =No error detected on channel A
LTVA	Latest Transmit Violation Channel A The flag signals a latest transmit violation on channel A to the Host. 1 =Latest transmit violation detected on channel A 0 =No latest transmit violation detected on channel A
TABA	The flag signals to the Host that a transmission across a slot boundary occurred for channel A. 1 = Transmission across slot boundary detected on channel A 0 = No transmission across slot boundary detected on channel A

Channel-specific error flags channel B:

Bit name	Function
EDB	Error Detected on Channel B This bit is set whenever one of the flags **FRACS.SEDB**, **FRACS.CEDB**, **FRACS.CIB**, **FRACS.SBVB** changes from ’0’ to ’1’. 1 =Error detected on channel B 0 =No error detected on channel B
LTVB	Latest Transmit Violation Channel B The flag signals a latest transmit violation on channel B to the Host. 1 =Latest transmit violation detected on channel B 0 =No latest transmit violation detected on channel B
TABB	The flag signals to the Host that a transmission across a slot boundary occurred for channel B. 1 = Transmission across slot boundary detected on channel B 0 = No transmission across slot boundary detected on channel B

(2) Status interrupt register (FRSIR)

The flags are set when the CC detects one of the listed events. The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hard reset will also clear the register.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSIR	R	0	0	0	0	0	0	MTSB	WUPB	0	0	0	0	0	0	MTSA	WUPA
0x0024	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
WST	Wakeup Status This flag is set when the wakeup status vector **FRCCSV.WSV[2:0]** is changed by a protocol event. 1 =Wakeup status changed 0 =Wakeup status unchanged
CAS	Collision Avoidance Symbol This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received. 1 = Bit pattern matching the CAS symbol received 0 = No bit pattern matching the CAS symbol received
CYCS	Cycle Start Interrupt This flag is set by the CC when a communication cycle starts 1 =Communication cycle started 0 =No communication cycle started
TXI	Transmit Interrupt This flag is set by the CC after successful frame transmission if bit **MBI** in the respective message buffer is set to '1'. 1 =At least one frame was transmitted successfully 0 =No frame transmitted
RXI	Receive Interrupt This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see New Data 1/2/3/4 (**FRNDAT1/2/3/4**)), and if bit **MBI** of that message buffer is set to '1'. 1 =At least one data section has been updated 0 =No data section has been updated
RFNE	Receive FIFO Not Empty This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The actual state of the receive FIFO is monitored in register **FRFSR**. 1 = Receive FIFO is not empty 0 = Receive FIFO is empty

Bit name	Function
RFCL	Receive FIFO Critical Level This flag is set when the receive FIFO fill level **FRFSR.RFFL[7:0]** is equal or greater than the critical level as configured by **FRFCL.CL[7:0]**. 1 = Receive FIFO critical level reached 0 = Receive FIFO below critical level
NMVC	Network Management Vector Changed This interrupt flag signals a change in the Network Management Vector visible to the Host. 1 =Network management vector changed 0 =No change in the network management vector
TI0	Timer Interrupt 0 This flag is set whenever timer 0 matches the conditions configured in register FRT0C. A Timer Interrupt 0 is also signalled on pin **eray_tint0**. 1 =Timer Interrupt 0 occurred 0 =No Timer Interrupt 0
TI1	Timer Interrupt 1 This flag is set whenever timer 1 matches the conditions configured in register FRT1C. A Timer Interrupt 1 is also signalled on pin **eray_tint1**. 1 =Timer Interrupt 1 occurred 0 =No Timer Interrupt 1
TIBC	Transfer Input Buffer Completed This flag is set whenever a transfer from Input Buffer to the Message RAM has completed and bit **FRIBCR.IBSYS** in the Input Buffer Command Request register has been reset by the Message Handler. 1 =Transfer between Input Buffer and Message RAM completed 0 =No transfer completed since bit was reset
TOBC	Transfer Output Buffer Completed This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and bit **FROBCR.OBSYS** in the Output Buffer Command Request register has been reset by the Message Handler. 1 =Transfer between Message RAM and Output Buffer completed 0 =No transfer completed since bit was reset
SWE	Stop Watch Event This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch register (FRSTPW1)). 1 = Stop Watch Event occurred 0 = No Stop Watch Event

Bit name	Function
SUCS	Startup Completed Successfully This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state. 1 =Startup completed successfully 0 =No startup completed successfully
MBSI	Message Buffer Status Interrupt This flag is set by the CC when the message buffer status **MBS** has changed and if bit **MBI** of that message buffer is set. 1 = Message buffer status of at least one message buffer with MBI = '1' has changed 0 = No message buffer status change of message buffer with MBI = '1'
SDS	Start of Dynamic Segment This flag is set by the CC when the dynamic segment starts. 1 = Dynamic segment started 0 = Dynamic segment not yet started

Channel-specific status flags:

Bit name	Function
WUPA	Wakeup Pattern Channel A This flag is set by the CC when a wakeup pattern was received on channel A. Only set when the CC is in WAKEUP, READY, or STARTUP state, or when in Monitor mode. 1 = Wakeup pattern received on channel A 0 = No wakeup pattern received on channel A
MTSA	MTS Received on Channel A (*vSS!ValidMTSA*) Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. 1 = MTS symbol received on channel A 0 = No MTS symbol received on channel A
WUPB	Wakeup Pattern Channel B This flag is set by the CC when a wakeup pattern was received on channel B. Only set when the CC is in WAKEUP, READY, or STARTUP state, or when in Monitor mode. 1 = Wakeup pattern received on channel B 0 = No wakeup pattern received on channel B
MTSB	MTS Received on Channel B (*vSS!ValidMTSB*) Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. 1 = MTS symbol received on channel B 0 = No MTS symbol received on channel B

(3) Error interrupt line select (FREILS)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FREILS	R	0	0	0	0	0	TABBL	LTVBL	EDBL	0	0	0	0	0	TABAL	LTVAL	EDAL
0x0028	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHFL	IOBAL	IIBAL	EFAL	RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Error Interrupt Line Select register assigns an interrupt generated by a specific error interrupt flag from register EIR to one of the two module interrupt lines:

1 = Interrupt assigned to interrupt line **eray_int1**

0 = Interrupt assigned to interrupt line **eray_int0**

Bit name	Function
PEMCL	POC Error Mode Changed Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
CNAL	Command Not Accepted Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
SFBML	Sync Frames Below Minimum Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
SFOL	Sync Frame Overflow Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
CCFL	Clock Correction Failure Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
CCLL	CHI Command Locked Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
PERRL	Parity Error Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
RFOL	Receive FIFO Overrun Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
EFAL	Empty FIFO Access Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
IIBAL	Illegal Output Buffer Access Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
IOBAL	Illegal Output Buffer Access Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**

Bit name	Function
MHFL	Message Handler Constraints Flag Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
EDAL	Error Detected on Channel A Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
LTVAL	Latest Transmit Violation Channel A Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TABAL	Transmission Across Boundary Channel A Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
EDBL	Error Detected on Channel B Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
LTVBL	Latest Transmit Violation Channel B Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TABBL	Transmission Across Boundary Channel B Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**

(4) Status Interrupt Line Select (FRSILS)

The settings in the Status Interrupt Line Select register assign an interrupt generated by a specific status interrupt flag to one of the two module interrupt lines (**eray_int0** or **eray_int1**).

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSILS	R	0	0	0	0	0	0	MTSBL	WUPBL	0	0	0	0	0	0	MTSAL	WUPAL
0x002C	W																
Reset		0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
	W																
Reset		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit name	Function
WSTL	Wakeup Status Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
CASL	Collision Avoidance Symbol Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
CYCSL	Cycle Start Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TXIL	Transmit Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
RXIL	Receive Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
RFNEL	Receive FIFO Not Empty Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
RFCLL	Receive FIFO Critical Level Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
NMVCL	Network Management Vector Changed Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TI0L	Timer Interrupt 0 Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TI1L	Timer Interrupt 1 Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TIBCL	Transfer Input Buffer Completed Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
TOBCL	Transfer Output Buffer Completed Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**

Bit name	Function
SWEL	Stop Watch Event Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
SUCSL	Startup Completed Successfully Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
MBSIL	Message Buffer Status Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
SDSL	Start of Dynamic Segment Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
WUPAL	Wakeup Pattern Channel A Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
MTSAL	Media Access Test Symbol Channel A Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
WUPBL	Wakeup Pattern Channel B Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**
MTSBL	Media Access Test Symbol Channel B Interrupt Line 1 =Interrupt assigned to interrupt line **eray_int1** 0 =Interrupt assigned to interrupt line **eray_int0**

(5) Error interrupt enable set / reset (FREIES, FREIER)

The settings in the Error Interrupt Enable register determine which status changes in the Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to address 0x0030 and reset by writing to address 0x0034. Writing a '1' sets / resets the specific enable bit, writing a '0' has no effect. Reading from both addresses will result in the same value.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FREIES,R	R	0	0	0	0	0	TABBE	LTVBE	EDBE	0	0	0	0	0	TABAE	LTVAE	EDAE
S:0x0030 R:0x0034	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHFE	IOBAE	IIBAE	EFAE	RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
PEMCE	POC Error Mode Changed Interrupt Enable 1 =Protocol Error Mode Changed Interrupt enabled 0 =Interrupt disabled
CNAE	Command Not Accepted Interrupt Enable 1 =Command Not Valid Interrupt enabled 0 =Interrupt disabled
SFBME	Sync Frames Below Minimum Interrupt Enable 1 =Sync Frames Below Minimum Interrupt enabled 0 =Interrupt disabled
SFOE	Sync Frame Overflow Interrupt Enable 1 =Sync Frame Overflow Interrupt enabled 0 =Interrupt disabled
CCFE	Clock Correction Failure Interrupt Enable 1 =Clock Correction Failure Interrupt enabled 0 =Interrupt disabled
CCLE	CHI Command Locked Interrupt Enable 1 =CHI Command Locked Interrupt enabled 0 =Interrupt disabled
PERRE	Parity Error Interrupt Enable 1 = Parity Error Interrupt enabled 0 =Interrupt disabled
RFOE	Receive FIFO Overrun Interrupt Enable 1 =Receive FIFO Overrun Interrupt enabled 0 =Interrupt disabled
EFAE	Empty FIFO Access Interrupt Enable 1 = Empty FIFO Access Interrupt enabled 0 =Interrupt disabled
IIBAE	Illegal Input Buffer Access Interrupt Enable 1 =Illegal Input Buffer Access Interrupt enabled 0 =Interrupt disabled
IOBAE	Illegal Output Buffer Access Interrupt Enable 1 = Illegal Output Buffer Access Interrupt enabled 0 =Interrupt disabled

Bit name	Function
MHFE	Message Handler Constraints Flag Interrupt Enable 1 = Message Handler Constraints Flag Interrupt enabled 0 =Interrupt disabled
EDAE	Error Detected on Channel A Interrupt Enable 1 =Error Detected on Channel A Interrupt enabled 0 =Interrupt disabled
LTVAE	Latest Transmit Violation Channel A Interrupt Enable 1 =Latest Transmit Violation Channel A Interrupt enabled 0 =Interrupt disabled
TABAE	Transmission Across Boundary Channel A Interrupt Enable 1 = Transmission Across Boundary Channel A Interrupt enabled 0 =Interrupt disabled
EDBE	Error Detected on Channel B Interrupt Enable 1 =Error Detected on Channel B Interrupt enabled 0 =Interrupt disabled
LTVBE	Latest Transmit Violation Channel B Interrupt Enable 1 =Latest Transmit Violation Channel B Interrupt enabled 0 =Interrupt disabled
TABBE	Transmission Across Boundary Channel B Interrupt Enable 1 = Transmission Across Boundary Channel B Interrupt enabled 0 =Interrupt disabled

(6) Status interrupt enable set / reset (FRSIES, FRSIER)

The settings in the Status Interrupt Enable register determine which status changes in the Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to address 0x0038 and reset by writing to address 0x003C. Writing a '1' sets / resets the specific enable bit, writing a '0' has no effect. Reading from both addresses will result in the same value.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSIES,R	R	0	0	0	0	0	0	MTSBE	WUPBE	0	0	0	0	0	0	MTSAE	WUPAE
S:0x0038 R:0x003C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
WSTE	Wakeup Status Interrupt Enable 1 =Wakeup Status Interrupt enabled 0 =Interrupt disabled
CASE	Collision Avoidance Symbol Interrupt Enable 1 =Collision Avoidance Symbol Interrupt enabled 0 =Interrupt disabled
CYCSE	Cycle Start Interrupt Enable 1 =Cycle Start Interrupt enabled 0 =Interrupt disabled
TXIE	Transmit Interrupt Enable 1 =Transmit Interrupt enabled 0 =Interrupt disabled
RXIE	Receive Interrupt Enable 1 =Receive Interrupt enabled 0 =Interrupt disabled
RFNEE	Receive FIFO Not Empty Interrupt Enable 1 =Receive FIFO Not Empty Interrupt enabled 0 =Interrupt disabled
RFCLE	Receive FIFO Critical Level Interrupt Enable 1 =Receive FIFO Critical Level Interrupt enabled 0 =Interrupt disabled
NMVCE	Network Management Vector Changed Interrupt Enable 1 =Network Management Vector Changed Interrupt enabled 0 =Interrupt disabled
TI0E	Timer Interrupt 0 Enable 1 =Timer Interrupt 0 enabled 0 =Interrupt disabled
TI1E	Timer Interrupt 1 Enable 1 =Timer Interrupt 1 enabled 0 =Interrupt disabled
TIBCE	Transfer Input Buffer Completed Interrupt Enable 1 =Transfer Input Buffer Completed Interrupt enabled 0 =Interrupt disabled

Bit name	Function
TOBCE	Transfer Output Buffer Completed Interrupt Enable 1 =Transfer Output Buffer Completed Interrupt enabled 0 =Interrupt disabled
SWEE	Stop Watch Event Interrupt Enable 1 =Stop Watch Event Interrupt enabled 0 =Interrupt disabled
SUCSE	Startup Completed Successfully Interrupt Enable 1 =Startup Completed Successfully Interrupt enabled 0 =Interrupt disabled
MBSIE	Message Buffer Status Interrupt Enable 1 =Message Buffer Status Interrupt enabled 0 =Interrupt disabled
SDSE	Start of Dynamic Segment Interrupt Enable 1 = Start of Dynamic Segment Interrupt enabled 0 =Interrupt disabled
WUPAE	Wakeup Pattern Channel A Interrupt Enable 1 =Wakeup Pattern Channel A Interrupt enabled 0 =Interrupt disabled
MTSAE	MTS Received on Channel A Interrupt Enable 1 =MTS Received on Channel A Interrupt enabled 0 =Interrupt disabled
WUPBE	Wakeup Pattern Channel B Interrupt Enable 1 =Wakeup Pattern Channel B Interrupt enabled 0 =Interrupt disabled
MTSBE	MTS Received on Channel B Interrupt Enable 1 =MTS Received on Channel B Interrupt enabled 0 =Interrupt disabled

(7) Interrupt line enable (FRILE)

Each of the two interrupt lines to the Host CPU (**eray_int0**, **eray_int1**) can be enabled / disabled separately by programming bit **EINT0** and **EINT1**.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRILE	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EINT1	EINT0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
EINT0	Enable Interrupt Line 0 1 =Interrupt line **eray_int0** enabled 0 =Interrupt line **eray_int0** disabled
EINT1	Enable Interrupt Line 1 1 =Interrupt line **eray_int1** enabled 0 =Interrupt line **eray_int1** disabled

(8) Timer 0 configuration (FRT0C)

Absolute timer. Specifies in terms of cycle count and macrotick the point in time when the timer 0 interrupt occurs. When the timer 0 interrupt is asserted, output signal **eray_tint0** is set to '1' for the duration of one macrotick and **FRSIR.TI0** is set to '1'.

Timer 0 can be activated as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state. Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit **T0RC** to '0'.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRT0C	R	0	0	T0MO 13	T0MO 12	T0MO 11	T0MO 10	T0MO9	T0MO8	T0MO7	T0MO6	T0MO5	T0MO4	T0MO3	T0MO2	T0MO1	T0MO0
0x0044	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	T0CC6	T0CC5	T0CC4	T0CC3	T0CC2	T0CC1	T0CC0	0	0	0	0	0	0	T0MS	T0RC
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
T0RC	Timer 0 Run Control 1 = Timer 0 running 0 = Timer 0 halted
T0MS	Timer 0 Mode Select 1 = Continuous mode 0 = Single-shot mode
T0CC[6:0]	Timer 0 Cycle Code The 7-bit timer 0 cycle code determines the cycle set used for generation of the timer 0 interrupt. For details about the configuration of the cycle code see 5.7.3 Cycle Counter Filtering.
T0MO[13:0]	Timer 0 Macrotick Offset Configures the macrotick offset from the beginning of the cycle where the interrupt is to occur. The Timer 0 Interrupt occurs at this offset for each cycle in the cycle set.

Note The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0. In case the CC leaves NORMAL_ACTIVE or NORMAL_PASSIVE state, or if timer 0 is halted by Host command, output signal eray_tint0 is reset to '0' immediately.

(9) Timer 1 configuration (FRT1C)

Relative timer. After the specified number of macroticks has expired, the timer 1 interrupt is asserted, output signal **eray_tint1** is set to '1' for the duration of one macrotick and **FRSIR.TI1** is set to '1'.

Timer 1 can be activated as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state. Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit **T1RC** to '0'.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRT1C	R	0	0	T1MC 13	T1MC 12	T1MC 11	T1MC 10	T1MC9	T1MC8	T1MC7	T1MC6	T1MC5	T1MC4	T1MC3	T1MC2	T1MC1	T1MC0
0x0048	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T1MS	T1RC
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
T1RC	Timer 1 Run Control 1 = Timer 1 running 0 = Timer 1 halted
T1MS	Timer 1 Mode Select 1 = Continuous mode 0 = Single-shot mode
T1MC[13:0]	Timer 1 Macrotick Count When the configured macrotick count is reached the timer 1 interrupt is generated. Valid values are: 2 to 16383 MT in continuous mode 1 to 16383 MT in single-shot mode

Note In case the CC leaves NORMAL_ACTIVE or NORMAL_PASSIVE state, or if timer 1 is halted by Host command, output signal **eray_tint1** is reset to '0' immediately.

(10) Stop watch register (FRSTPW1)

The stop watch is activated by a rising or falling edge on pin **eray_stpwt**, by an interrupt 0,1 event (rising edge on pin **eray_int0** or **eray_int1**) or by the Host by writing bit **SSWT** to '1'. With the macrotick counter increment following next to the stop watch activation the actual cycle counter and macrotick values are captured in register STPW1 while the slot counter values for channel A and B are captured in register **FRSTPW2**.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSTPW1	R	0	0	SMTV13	SMTV12	SMTV11	SMTV10	SMTV9	SMTV8	SMTV7	SMTV6	SMTV5	SMTV4	SMTV3	SMTV2	SMTV1	SMTV0
0x004C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	SCCV5	SCCV4	SCCV3	SCCV2	SCCV1	SCCV0	0	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
ESWT	External Stop Watch Trigger If enabled an edge on input **eray_stpwt** or an interrupt 0,1 event (rising edge on pin **eray_int0** or **eray_int1**) activates the stop watch. In single-shot mode this bit is reset to '0' after the actual cycle counter and macrotick value are stored in the Stop Watch register. 1 = Stop watch trigger enabled 0 = Stop watch trigger disabled
SWMS	Stop Watch Mode Select 1 =Continuous mode 0 =Single-shot mode
EDGE	Stop Watch Trigger Edge Select 1 =Rising edge 0 =Falling edge
SSWT	Software Stop Watch Trigger When the Host writes this bit to '1' the stop watch is activated. After the actual cycle counter and macrotick value are stored in the Stop Watch register this bit is reset to '0'. The bit is only writeable while ESWT = '0'. 1 =Stop watch activated by software trigger 0 =Software trigger reset
EETP	Enable External Trigger Pin Enables stop watch trigger event via pin **eray_stpwt** if **ESWT** = '1'. 1 = Edge on pin **eray_stpwt** triggers stop watch 0 = Stop watch trigger via pin **eray_stpwt** disabled
EINT0	Enable Interrupt 0 Trigger Enables stop watch trigger by interrupt 0 event if **ESWT** = '1'. 1 = Interrupt 0 event triggers stop watch 0 = Stop watch trigger by interrupt 0 disabled

Bit name	Function
ESWT	External Stop Watch Trigger If enabled an edge on input **eray_stpwt** or an interrupt 0,1 event (rising edge on pin **eray_int0** or **eray_int1**) activates the stop watch. In single-shot mode this bit is reset to '0' after the actual cycle counter and macrotick value are stored in the Stop Watch register. 1 = Stop watch trigger enabled 0 = Stop watch trigger disabled
EINT1	Enable Interrupt 1 Trigger Enables stop watch trigger by interrupt 1event if **ESWT** = '1'. 1 = Interrupt 1 event triggers stop watch 0 = Stop watch trigger by interrupt 1 disabled
SCCV[5:0]	Stopped Cycle Counter Value State of the cycle counter when the stop watch event occurred. Valid values are 0 to 63.
SMTV[13:0]	Stopped Macrotick Value State of the macrotick counter when the stop watch event occurred. Valid values are 0 to 15,999.

Note Bits **ESWT** and **SSWT** cannot be set to '1' simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

(11) Stop watch register (FRSTPW2)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSTPW2	R	0	0	0	0	0	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB	SSCVB
0x0050	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA	SSCVA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
SSCVA[10:0]	Stop Watch Captured Slot Counter Value Channel A State of the slot counter for channel A when the stop watch event occurred. Valid values are 0 to 2047.
SSCVB[10:0]	Stop Watch Captured Slot Counter Value Channel B State of the slot counter for channel B when the stop watch event occurred. Valid values are 0 to 2047.

24.4.5 CC control registers

This section describes the registers provided by the CC to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Please consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered from hard reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to proceed as described in Section Lock Register (LCK).

All bits marked with an asterisk * can be updated in DEFAULT_CONFIG or CONFIG state only!

(1) SUC configuration register 1 (FRSUCC1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSUCC1	R	0	0	0	0	CCHB*	CCHA*	MTSB*	MTSA*	HCSE*	TSM*	WUCS*	PTA4*	PTA3*	PTA2*	PTA1*	PTA0*
0x0080	W																
Reset		0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	CSA4*	CSA3*	CSA2*	CSA1*	CSA0*	0	TXSY*	TXST*	PBSY	0	0	0	CMD3	CMD2	CMD1	CMD0
	W																
Reset		0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

Bit name	Function
CMD[3:0]	CHI Command Vector The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector **CMD[3:0]** will be reset to "0000" = command_not_accepted, and flag **FREIR.CNA** will be set to '1'. In case the previous CHI command has not yet completed, **FREIR.CCL** is set to '1' together with **FREIR.CNA**; the CHI command needs to be repeated. Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will **FREIR.CNA** be set. 0000 =command_not_accepted 0001 =CONFIG 0010 =READY 0011 =WAKEUP 0100 =RUN 0101 =ALL_SLOTS 0110 =HALT 0111 =FREEZE 1000 =SEND_MTS 1001 =ALLOW_COLDSTART 1010 =RESET_STATUS_INDICATORS 1011 =MONITOR_MODE 1100 =CLEAR_RAMS 1101 =reserved 1110 =reserved 1111 =reserved

Reading **CMD[3:0]** shows whether the last CHI command was accepted. The actual POC state is monitored by **FRCCSV.POCS[5:0]**. The reserved CHI commands belong to the hardware test functions. In general the Host must check **FRSUCC1.PBSY** before writing a new CHI command.

CMD[3:0]	Function
0000	command_not_accepted **CMD[3:0]** is reset to "0000" due to one of the following conditions: • Illegal command applied by the Host • Host applied command to leave CONFIG state without preceding config lock key • Host applied new command while execution of the previous Host command has not completed • Host writes **command_not_accepted** When **CMD[3:0]** is reset to "0000", **FREIR.CNA** is set, and - if enabled - an interrupt is generated. Commands which are not accepted are not executed.
0001	CONFIG Go to POC state CONFIG when called in POC states DEFAULT_CONFIG, READY, or in MONITOR_MODE. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
0010	READY Go to POC state READY when called in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
0011	WAKEUP Go to POC state WAKEUP when called in POC state READY. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
0100	RUN Go to POC state STARTUP when called in POC state READY. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
0101	ALL_SLOTS Leave single slot mode after successful startup / integration at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
0110	HALT Set halt request **FRCCSV.HRQ** and go to POC state HALT at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
0111	FREEZE Set the freeze status indicator **CCSV.FSI** and go to POC state HALT immediately. Can be called from any state.
1000	SEND_MTS Send single MTS symbol during the next following symbol window on the channel configured by **MTSA**, **MTSB**, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (**FRCCSV.SLM[1:0]** = "11"). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

CMD[3:0]	Function
1001	ALLOW_COLDSTART The command resets **FRCCSV.CSI** to enable the node to become leading coldstarter. When called in states DEFAULT_CONFIG, CONFIG, HALT, or MONITOR_MODE, **CMD[3:0]** will be reset to "0000" = command_not_accepted. To become leading coldstarter it is also required that both **TXST** and **TXSY** are set.
1010	RESET_STATUS_INDICATORS Resets status flags **FRCCSV.CSNI**, **FRCCSV.CSAI**, and **FRCCSV.WSV[2:0]** to their default values. May be called in POC states READY and STARTUP.When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.
1011	MONITOR_MODE Enter MONITOR_MODE when called in POC state CONFIG. In this mode the CC is able to receive FlexRay frames and wakeup pattern. It is also able to detect coding errors. The temporal integrity of received frames is not checked. This mode can be used for debugging purposes, e.g. in case that the startup of a FlexRay network fails. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted. For details see MONITOR_MODE
1100	CLEAR_RAMS Sets **FRMHDS.CRAM** when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted. **FRMHDS.CRAM** is also set when the CC leaves hard reset. By setting **FRMHDS.CRAM** all internal RAM blocks are initialized to zero. During the initialization of the RAMs, **PBSY** will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS. The initialization of the E-Ray internal RAM blocks requires 2048 **eray_bclk** cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after hard reset or after assertion of CHI command CLEAR_RAMS. Before asserting CHI command CLEAR_RAMS the Host should make sure that no transfer between Message RAM and IBF / OBF or the Transient Buffer RAMs is ongoing. This command also resets the Message Buffer Status registers FRMHDS, **FRLDTS**, **FRFSR**, **FRMHDF**, **FRTXRQ1/2/3/4**, **FRNDAT1/2/3/4**, and **FRMBSC1/2/3/4**.

Note All accepted commands with exception of CLEAR_RAMS and SEND_MTS will cause a change of the POC state in the **eray_sclk** domain after at most 8 cycles of the slower of the two clocks **eray_bclk** and **eray_sclk**, counted from the falling edge of the CHI input signal **eray_select**, assumed that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register CCSV will show data that is additionally delayed by synchronization from **eray_sclk** to **eray_bclk** domain and by the Host-specific CPU interface. The maximum additional delay is 12 cycles of the slower of the two clocks **eray_bclk** and **eray_sclk**.

Bit name	Function
PBSY	POC Busy Signals that the POC is busy and cannot accept a command from the Host. **CMD[3:0]** is locked against write accesses. Set to '1' after hard reset during initialization of internal RAM blocks. 1 = POC is busy, **CMD[3:0]** locked 0 = POC not busy, **CMD[3:0]** writeable
TXST	Transmit Startup Frame in Key Slot *(pKeySlotUsedForStartup)* Defines whether the key slot is used to transmit startup frames. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 1 = Key slot used to transmit startup frame, node is leading or following coldstarter 0 = No startup frame transmission in key slot, node is non-coldstarter
TXSY	Transmit Sync Frame in Key Slot *(pKeySlotUsedForSync)* Defines whether the key slot is used to transmit sync frames. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 1 = Key slot used to transmit sync frame, node is sync node 0 = No sync frame transmission in key slot, node is neither sync nor coldstart node
CSA[4:0]	Cold Start Attempts (*gColdStartAttempts*) Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node. It can be modified in DEFAULT_CONFIG or CONFIG state only. Must be identical in all nodes of a cluster. Valid values are 2 to 31.
PTA[4:0]	Passive to Active (*pAllowPassiveToActive*) Defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If set to "00000" the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. It can be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 31 even/odd cycle pairs.
WUCS	Wakeup Channel Select (*pWakeupChannel*) With this bit the Host selects the channel on which the CC sends the Wakeup pattern. The CC ignores any attempt to change the status of this bit when not in DEFAULT_CONFIG or CONFIG state. 1 =Send wakeup pattern on channel B 0 =Send wakeup pattern on channel A
TSM	Transmission Slot Mode (*pSingleSlotEnabled*) Selects the initial transmission slot mode. In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit **FRMRC.SPLM**. In case **TSM** = '1', message buffer 0 respectively message buffers 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots. **TSM** is a configuration bit which can only be set / reset by the Host. The bit can be written in DEFAULT_CONFIG or CONFIG state only. The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing **CMD[3:0]** = "0101" in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by **FRCCSV.SLM[1:0]**. 1 =SINGLE Slot Mode (default by hardware reset) 0 =ALL Slot Mode

Bit name	Function
HCSE	Halt due to Clock Sync Error (*pAllowHaltDueToClock*) Controls reaction of the CC to a clock synchronization error. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 1 =CC will enter HALT state 0 =CC will enter/remain in NORMAL_PASSIVE
MTSA	Select Channel A for MTS Transmission The bit selects channel A for MTS symbol transmission. The flag is reset by default and may be modified only in DEFAULT_CONFIG or CONFIG state. 1 = Channel A selected for MTS transmission 0 = Channel A disabled for MTS transmission
MTSB	Select Channel B for MTS Transmission The bit selects channel B for MTS symbol transmission. The flag is reset by default and may be modified only in DEFAULT_CONFIG or CONFIG state. 1 = Channel B selected for MTS transmission 0 = Channel B disabled for MTS transmission Note: **MTSA,B** may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to SUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in 21.4.3.2 Lock Register (LCK). This may be combined with CHI command SEND_MTS. If both bits **MTSA** and **MTSB** are set to '1' an MTS symbol will be transmitted on both channels when requested by writing **CMD[3:0]** = "1000".
CCHA	Connected to Channel A (*pChannels*) Configures whether the node is connected to channel A. 1 = Node connected to channel A (default by hardware reset) 0 = Not connected to channel A
CCHB	Connected to Channel B (*pChannels*) Configures whether the node is connected to channel B. 1 = Node connected to channel B (default by hardware reset) 0 = Not connected to channel B

Table 24-6 below references the CHI commands from the FlexRay Protocol Specification V2.1 (section 2.2.1.1, Table 2-2) to the E-Ray CHI command vector CMD[3:0].

Table 24-6 Reference to CHI Host command summary from FlexRay protocol specification (1/2)

CHI command	Where processed (POC States)	CHI Command Vector CMD[3:0]
ALL_SLOTS	POC: normal active, POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config, POC: config, POC: halt	ALLOW_COLDSTART
CONFIG	POC default config, POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE

Table 24-6 Reference to CHI Host command summary from FlexRay protocol specification (2/2)

CHI command	Where processed (POC States)	CHI Command Vector CMD[3:0]
HALT	POC: normal active, POC: normal passive	HALT
READY	All except POC:default config, POC:config, POC:ready, POC:halt	READY
RUN	POC:ready	RUN
WAKEUP	POC:ready	WAKEUP

(2) SUC configuration register 2 (FRSUCC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSUCC2	R	0	0	0	0	LTN3*	LTN2*	LTN1*	LTN0*	0	0	0	LT20*	LT19*	LT18*	LT17*	LT16*
0x0084	W																
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	LT15*	LT14*	LT13*	LT12*	LT11*	LT10*	LT9*	LT8*	LT7*	LT6*	LT5*	LT4*	LT3*	LT2*	LT1*	LT0*
	W																
Reset		0	0	0	0	0	1	0	1	1	0	1	0	0	1	0	0

Bit name	Function
LT[20:0]	Listen Timeout (*pdListenTimeout*) Configures wakeup / startup listen timeout in T. The range for pdListenTimeout is 1284 to 1283846 uT.
LTN[3:0]	Listen Timeout Noise (*gListenNoise* - 1) Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout. The range for gListenNoise is 2 to 16. **LTN[3:0]** must be configured identical in all nodes of a cluster.

Note The wakeup / startup noise timeout is calculated as follows:
pdListenTimeout • gListenNoise **= LT[20:0] •** (**LTN[3:0]** + 1)

(3) SUC configuration register 3 (FRSUCC3)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSUCC3	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0088	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	WCF3*	WCF2*	WCF1*	WCF0*	WCP3*	WCP2*	WCP1*	WCP0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bit name	Function
WCP[3:0]	Maximum Without Clock Correction Passive (*gMaxWithoutClockCorrectionPassive*) Defines the number of consecutive even/odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.
WCF[3:0]	Maximum Without Clock Correction Fatal (*gMaxWithoutClockCorrectionFatal*) Defines the number of consecutive even/odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.

Note The transition to HALT state is prevented if **FRSUCC1.HCSE** is not set.

(4) NEM configuration register (FRNEMC)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRNEMC	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	NML3*	NML2*	NML1*	NML0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
NML[3:0]	Network Management Vector Length (*gNetworkManagementVectorLength*) These bits configure the length of the NM vector. The configured length must be identical in all nodes of a cluster. Valid values are 0 to 12 bytes.

(5) PRT configuration register 1 (FRPRTC1)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRPRTC1	R	RWP5*	RWP4*	RWP3*	RWP2*	RWP1*	RWP0*	0	RXW8*	RXW7*	RXW6*	RXW5*	RXW4*	RXW3*	RXW2*	RXW1*	RXW0*
0x0090	W																
Reset		0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	BRP1*	BRP0*	SPP1*	SPP0*	0	CASM6	CASM5	CASM4	CASM3	CASM2	CASM1	CASM0	TSST3*	TSST2*	TSST1*	TSST0*
	W																
Reset		0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1

Bit name	Function
TSST[3:0]	Transmission Start Sequence Transmitter (*gdTSSTransmitter*) Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 T = 100ns@10Mbps). Must be identical in all nodes of a cluster. Valid values are 3 to 15 bit times.
CASM[6:0]	Collision Avoidance Symbol Max (*gdCASRxLowMax*) Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS). **CASM6** is fixed to '1'. Valid values are 67 to 99 bit times.
SPP[1:0]	Strobe Point Position Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by **SPP[1:0]**. 00, 11= Sample 5 (default) 01 = Sample 4 10 = Sample 6 **Note:** The current revision 2.1 of the FlexRay protocol requires that SPP[1:0] = "00". The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.
BRP[1:0]	Baud Rate Prescaler (*gdSampleClockPeriod*, *pSamplePerMicrotick*) The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock **eray_sclk** = 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate. 00 = 10 MBit/s (default) gdSampleClockPeriod = 12.5 ns = 1 • **eray_sclk** pSamplesPerMicrotick = 2 (1 T = 25 ns) 01 = 5 MBit/s gdSampleClockPeriod = 25 ns = 2 **eray_sclk** pSamplesPerMicrotick = 1 (1 T = 25 ns) 10, 11 = 2.5 MBit/s gdSampleClockPeriod = 50 ns = 4 **eray_sclk** pSamplesPerMicrotick = 1 (1 T = 50 ns)
RXW[8:0]	Wakeup Symbol Receive Window Length (*gdWakeupSymbolRxWindow*) Configures the number of bit times used by the node to test the duration of the received wakeup pattern. Must be identical in all nodes of a cluster. Valid values are 76 to 301 bit times.

Bit name	Function
RWP[5:0]	Repetitions of Tx Wakeup Pattern (*pWakeupPattern*) Configures the number of repetitions (sequences) of the Tx wakeup symbol. Valid values are 2 to 63.

(6) PRT configuration register 2 (FRPRTC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRPRTC2	R	0	0	TXL5*	TXL4*	TXL3*	TXL2*	TXL1*	TXL0*	TXI7*	TXI6*	TXI5*	TXI4*	TXI3*	TXI2*	TXI1*	TXI0*
0x0094	W																
Reset		0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	RXL5*	RXL4*	RXL3*	RXL2*	RXL1*	RXL0*	0	0	RXI5*	RXI4*	RXI3*	RXI2*	RXI1*	RXI0*
	W																
Reset		0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0

Bit name	Function
RXI[5:0]	Wakeup Symbol Receive Idle (*gdWakeupSymbolRxIdle*) Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 14 to 59 bit times.
RXL[5:0]	Wakeup Symbol Receive Low (*gdWakeupSymbolRxLow*) Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 10 to 55 bit times.
TXI[7:0]	Wakeup Symbol Transmit Idle (*gdWakeupSymbolTxIdle*) Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 45 to 180 bit times.
TXL[5:0]	Wakeup Symbol Transmit Low (*gdWakeupSymbolTxLow*) Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 15 to 60 bit times.

(7) MHD configuration register (FRMHDC)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMHDC 0x0098 R	0	0	0	SLT12*	SLT11*	SLT10*	SLT9*	SLT8*	SLT7*	SLT6*	SLT5*	SLT4*	SLT3*	SLT2*	SLT1*	SLT0*
FRMHDC 0x0098 W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	SFDL6*	SFDL5*	SFDL4*	SFDL3*	SFDL2*	SFDL1*	SFDL0*
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
SFDL[6:0]	Static Frame Data Length (*gPayloadLengthStatic*) Configures the cluster-wide frame length for all frames sent in the static segment in double bytes. The frame length must be identical in all nodes of a cluster. Valid values are 0 to 127.
SLT[12:0]	Start of Latest Transmit (*pLatestTx*) Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if **SLT[12:0]** is set to zero. Valid values are 0 to 7981 minislots.

(8) GTU configuration register 1 (FRGTUC1)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC1 0x00A0 R	0	0	0	0	0	0	0	0	0	0	0	0	UT19*	UT18*	UT17*	UT16*
FRGTUC1 0x00A0 W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UT15*	UT14*	UT13*	UT12*	UT11*	UT10*	UT9*	UT8*	UT7*	UT6*	UT5*	UT4*	UT3*	UT2*	UT1*	UT0*
W																
Reset	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0

Bit name	Function
UT[19:0]	Microtick per Cycle (*pMicroPerCycle*) Configures the duration of the communication cycle in microticks. Valid values are 640 to 640000 µT.

(9) GTU configuration register 2 (FRGTUC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC2	R	0	0	0	0	0	0	0	0	0	0	0	0	SNM3*	SNM2*	SNM1*	SNM0*
0x00A4	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MPC13*	MPC12*	MPC11*	MPC10*	MPC9*	MPC8*	MPC7*	MPC6*	MPC5*	MPC4*	MPC3*	MPC2*	MPC1*	MPC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit name	Function
MPC[13:0]	Macrotick Per Cycle (*gMacroPerCycle*) Configures the duration of one communication cycle in macroticks. The cycle length must be identical in all nodes of a cluster. Valid values are 10 to 16000 MT.
SNM[3:0]	Sync Node Max (*gSyncNodeMax*) Maximum number of frames within a cluster with sync frame indicator bit **SYN** set to ’1’. Must be identical in all nodes of a cluster. Valid values are 2 to 15.

(10) GTU configuration register 3 (FRGTUC3)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC3	R	0	MIOB6*	MIOB5*	MIOB4*	MIOB3*	MIOB2*	MIOB1*	MIOB0*	0	MIOA6*	MIOA5*	MIOA4*	MIOA3*	MIOA2*	MIOA1*	MIOA0*
0x00A8	W																
Reset		0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	UIOB7*	UIOB6*	UIOB5*	UIOB4*	UIOB3*	UIOB2*	UIOB1*	UIOB0*	UIOA7*	UIOA6*	UIOA5*	UIOA4*	UIOA3*	UIOA2*	UIOA1*	UIOA0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
UIOA[7:0]	Microtick Initial Offset Channel A (*pMicroInitialOffset[A]*) Configures the number of microticks which describe the distance between the macrotick boundary described by *gMacroInitialOffset* and the exact secondary time reference point. The parameter depends on *pDelayCompensation[A]* and therefore has to be set for each channel independently. Valid values are 0 to 240 µT.
UIOB[7:0]	Microtick Initial Offset Channel B (*pMicroInitialOffset[B]*) Configures the number of microticks which describe the distance between the macrotick boundary described by *gMacroInitialOffset* and the exact secondary time reference point. The parameter depends on *pDelayCompensation[B]* and therefore has to be set for each channel independently. Valid values are 0 to 240 µT.
MIOA[6:0]	Macrotick Initial Offset Channel A *(pMacroInitialOffset[A])* Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.
MIOB[6:0]	Macrotick Initial Offset Channel B *(pMacroInitialOffset[B])* Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.

(11) GTU configuration register 4 (FRGTUC4)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only. For details about configuration of **NIT[13:0]** and **OCS[13:0]** see Section Configuration of NIT Start and Offset Correction Start.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC4	R	0	0	OCS13*	OCS12*	OCS11*	OCS10*	OCS9*	OCS8*	OCS7*	OCS6*	OCS5*	OCS4*	OCS3*	OCS2*	OCS1*	OCS0*
0x00AC	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	NIT13*	NIT12*	NIT11*	NIT10*	NIT9*	NIT8*	NIT7*	NIT6*	NIT5*	NIT4*	NIT3*	NIT2*	NIT1*	NIT0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit name	Function
NIT[13:0]	Network Idle Time Start (*gMacroPerCycle - gdNIT - 1*) Configures the starting point of the Network Idle Time NIT at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if Macrotick = gMacroPerCycle - gdNIT -1 and the increment pulse of Macrotick is set. Must be identical in all nodes of a cluster. Valid values are 7 to 15997 MT.
OCS[13:0]	Offset Correction Start (*gOffsetCorrectionStart - 1*) Determines the start of the offset correction within the NIT phase, calculated from start of cycle. Must be identical in all nodes of a cluster. For cluster consisting of E-Ray implementations only, it is sufficient to program **OCS** = **NIT** + 1. Valid values are 8 to 15998 MT.

(12) GTU configuration register 5 (FRGTUC5)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC5	R	DEC7*	DEC6*	DEC5*	DEC4*	DEC3*	DEC2*	DEC1*	DEC0*	0	0	0	CDD4*	CDD3*	CDD2*	CDD1*	CDD0*
0x00B0	W																
Reset		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	DCB7*	DCB6*	DCB5*	DCB4*	DCB3*	DCB2*	DCB1*	DCB0*	DCA7*	DCA6*	DCA5*	DCA4*	DCA3*	DCA2*	DCA1*	DCA0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
DCA[7:0]	Delay Compensation Channel A (*pDelayCompensation[A]*) Used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05s. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 T.
DCB[7:0]	Delay Compensation Channel B (*pDelayCompensation[B]*) Used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05s. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 T.
CDD[4:0]	Cluster Drift Damping (*pClusterDriftDamping*) Configures the cluster drift damping value used in clock synchronization to minimize accumulation of rounding errors. Valid values are 0 to 20 T.
DEC[7:0]	Decoding Correction (*pDecodingCorrection*) Configures the decoding correction value used to determine the primary time reference point. Valid values are 14 to 143 T.

(13) GTU configuration register 6 (FRGTUC6)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC6	R	0	0	0	0	0	MOD10*	MOD9*	MOD8*	MOD7*	MOD6*	MOD5*	MOD4*	MOD3*	MOD2*	MOD1*	MOD0*
0x00B4	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	ASR10*	ASR9*	ASR8*	ASR7*	ASR6*	ASR5*	ASR4*	ASR3*	ASR2*	ASR1*	ASR0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
ASR[10:0]	Accepted Startup Range (*pdAcceptedStartupRange*) Number of microticks constituting the expanded range of measured deviation for startup frames during integration. Valid values are 0 to 1875 T.
MOD[10:0]	Maximum Oscillator Drift (*pdMaxDrift*) Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in µT. Valid values are 2 to 1923 µT.

(14) GTU configuration register 7 (FRGTUC7)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC7	R	0	0	0	0	0	0	NSS9*	NSS8*	NSS7*	NSS6*	NSS5*	NSS4*	NSS3*	NSS2*	NSS1*	NSS0*
0x00B8	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	SSL9*	SSL8*	SSL7*	SSL6*	SSL5*	SSL4*	SSL3*	SSL2*	SSL1*	SSL0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit name	Function
SSL[9:0]	Static Slot Length (*gdStaticSlot*) Configures the duration of a static slot in macroticks. The static slot length must be identical in all nodes of a cluster. Valid values are 4 to 659 MT.
NSS[9:0]	Number of Static Slots (*gNumberOfStaticSlots*) Configures the number of static slots in a cycle. At least 2 coldstart nodes must be configured to startup a FlexRay network. The number of static slots must be identical in all nodes of a cluster. Valid values are 2 to 1023.

(15) GTU configuration register 8 (FRGTUC8)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC8	R	0	0	0	NMS12*	NMS11*	NMS10*	NMS9*	NMS8*	NMS7*	NMS6*	NMS5*	NMS4*	NMS3*	NMS2*	NMS1*	NMS0*
0x00BC	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	MSL5*	MSL4*	MSL3*	MSL2*	MSL1*	MSL0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit name	Function
MSL[5:0]	Minislot Length (*gdMinislot*) Configures the duration of a minislot in macroticks. The minislot length must be identical in all nodes of a cluster. Valid values are 2 to 63 MT.
NMS[12:0]	Number of Minislots (*gNumberOfMinislots*) Configures the number of minislots within the dynamic segment of a cycle. The number of minislots must be identical in all nodes of a cluster. Valid values are 0 to 7986.

(16) GTU configuration register 9 (FRGTUC9)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC9	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSI1*	DSI0*
0x00C0	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MAPO4*	MAPO3*	MAPO2*	MAPO1*	MAPO0*	0	0	APO5*	APO4*	APO3*	APO2*	APO1*	APO0*
	W																
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit name	Function
APO[5:0]	Action Point Offset (*gdActionPointOffset*) Configures the action point offset in macroticks within static slots and symbol window. Must be identical in all nodes of a cluster. Valid values are 1 to 63 MT.
MAPO[4:0]	Minislot Action Point Offset (*gdMinislotActionPointOffset*) Configures the action point offset in macroticks. Must be identical in all nodes of a cluster. Valid values are 1 to 31 MT.
DSI[1:0]	Dynamic Slot Idle Phase (*gdDynamicSlotIdlePhase*) The duration of the dynamic slot idle phase has to be greater or equal than the idle detection time. Must be identical in all nodes of a cluster. Valid values are 0 to 2 Minislot.

(17) GTU configuration register 10 (FRGTUC10)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC10	R	0	0	0	0	0	MRC10*	MRC9*	MRC8*	MRC7*	MRC6*	MRC5*	MRC4*	MRC3*	MRC2*	MRC1*	MRC0*
0x00C4	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MOC13*	MOC12*	MOC11*	MOC10*	MOC9*	MOC8*	MOC7*	MOC6*	MOC5*	MOC4*	MOC3*	MOC2*	MOC1*	MOC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit name	Function
MOC[13:0]	Maximum Offset Correction (*pOffsetCorrectionOut*) Holds the maximum permitted offset correction value to be applied by the internal clock synchronization algorithm (absolute value). The CC checks only the internal offset correction value against the maximum offset correction value. Valid values are 5 to 15266 µT.
MRC[10:0]	Maximum Rate Correction (*pRateCorrectionOut*) Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks only the internal rate correction value against the maximum rate correction value (absolute value). Valid values are 2 to 1923 T.

(18) GTU configuration register 11 (FRGTUC11)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRGTUC11	R	0	0	0	0	0	ERC2*	ERC1*	ERC0*	0	0	0	0	0	EOC2*	EOC1*	EOC0*
0x00C8	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	ERCC1	ERCC0	0	0	0	0	0	0	EOCC1	EOCC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
EOCC[1:0]	External Offset Correction Control *(vExternOffsetControl)* By writing to **EOCC[1:0]** the external offset correction is enabled as specified below. Should be modified only outside NIT. 00, 01 = No external offset correction 10 = External offset correction value subtracted from calculated offset correction value 11 = External offset correction value added to calculated offset correction value
ERCC[1:0]	External Rate Correction Control *(vExternRateControl)* By writing to **ERCC[1:0]** the external rate correction is enabled as specified below. Should be modified only outside NIT. 00, 01 = No external rate correction 10 = External rate correction value subtracted from calculated rate correction value 11 = External rate correction value added to calculated rate correction value
EOC[2:0]	External Offset Correction (*pExternOffsetCorrection*) Holds the external clock offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated offset correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 µT.
ERC[2:0]	External Rate Correction (*pExternRateCorrection*) Holds the external clock rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated rate correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 µT.

24.4.6 CC status registers

The status vector may change faster than the Host can poll the status vector,depending on eray_bclk frequency.

(1) CC status vector (FRCCSV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRCCSV	R	0	0	PSL5	PSL4	PSL3	PSL2	PSL1	PSLO	RCA4	RCA3	RCA2	RCA1	RCA0	WSV2	WSV1	WSV0
0x0100	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	CSI	CSAI	CSNI	0	0	SLM1	SLM0	HRQ	FSI	POCS5	POCS4	POCS3	POCS2	POCS1	POCS0
	W																
Reset		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
POCS[5:0]	Protocol Operation Control Status Indicates the actual state of operation of the CC Protocol Operation Control 00 0000 = DEFAULT_CONFIG state 00 0001 = READY state 00 0010 = NORMAL_ACTIVE state 00 0011 = NORMAL_PASSIVE state 00 0100 = HALT state 00 0101 = MONITOR_MODE state 00 0110…00 1110 = reserved 00 1111 =CONFIG state Indicates the actual state of operation of the POC in the wakeup path 01 0000 = WAKEUP_STANDBY state 01 0001 = WAKEUP_LISTEN state 01 0010 = WAKEUP_SEND state 01 0011 = WAKEUP_DETECT state 01 0100…01 1111 = reserved Indicates the actual state of operation of the POC in the startup path 10 0000 = STARTUP_PREPARE state 10 0001 = COLDSTART_LISTEN state 10 0010 = COLDSTART_COLLISION_RESOLUTION state 10 0011 = COLDSTART_CONSISTENCY_CHECK state 10 0100 = COLDSTART_GAP state 10 0101 = COLDSTART_JOIN State 10 0110 = INTEGRATION_COLDSTART_CHECK state 10 0111 = INTEGRATION_LISTEN state 10 1000 = INTEGRATION_CONSISTENCY_CHECK state 10 1001 = INITIALIZE_SCHEDULE state 10 1010 = ABORT_STARTUP state 10 1011 = STARTUP_SUCCESS state 10 1100…11 1111 = reserved

Bit name	Function
FSI	Freeze Status Indicator (*vPOC!Freeze*) Indicates that the POC has entered the HALT state due to CHI command FREEZE or due to an error condition requiring an immediate POC halt. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state.
HRQ	Halt Request (*vPOC!CHIHaltRequest*) Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or when entering READY state.
SLM[1:0]	Slot Mode (*vPOC!SlotMode*) Indicates the actual slot mode of the POC in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE. Default is SINGLE. Changes to ALL, depending on **FRSUCC1.TSM**. In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL. Set to SINGLE in all other states. 00 = SINGLE 01 = reserved 10 = ALL_PENDING 11 = ALL
CSNI	Coldstart Noise Indicator (*vColdStartNoise*) Indicates that the cold start procedure occurred under noisy conditions. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.
CSAI	Coldstart Abort Indicator Coldstart aborted. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.
CSI	Cold Start Inhibit (*vColdStartInhibit*) Indicates that the node is disabled from cold starting. The flag is set whenever the POC enters READY state due to CHI command READY. The flag has to be reset under control of the Host by CHI command ALLOW_COLDSTART (CMD[3:0] = "1001"). 1 =Cold starting of node disabled 0 =Cold starting of node enabled

Bit name	Function
WSV[2:0]	Wakeup Status (*vPOC!WakeupStatus*) Indicates the status of the current wakeup attempt. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state. 000 = UNDEFINED. Wakeup not yet executed by the CC. 001 = RECEIVED_HEADER. Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state. 010 = RECEIVED_WUP. Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state. 011 = COLLISION_HEADER. Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel. 100 = COLLISION_WUP. Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel. 101 = COLLISION_UNKNOWN. Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header. 110 = TRANSMITTED. Set when the CC has successfully completed the transmission of the wakeup pattern. 111 = reserved
RCA[4:0]	Indicates the number of remaining coldstart attempts. The RUN command resets this counter to the maximum number of coldstart attempts as configured by **FRSUCC1.CSA[4:0]**. The initial value of **RCA[4:0]** during CONFIG and DEFAULT_CONFIG state is also **FRSUCC1.CSA[4:0]**.
PSL[5:0]	POC Status Log Status of **POCS[5:0]** immediately before entering HALT state. Set to HALT when FREEZE command is applied during HALT state and **FSI** is not already set i.e. the HALT state was not reached by FREEZE command. Reset to "00 0000" when leaving HALT state.

(2) CC error vector (FRCCEV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRCCEV	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0104	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	PTAC4	PTAC3	PTAC2	PTAC1	PTAC0	ERRM1	ERRM0	0	0	CCFC3	CCFC2	CCFC1	CCFC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

Bit name	Function
CCFC[3:0]	Clock Correction Failed Counter (*vClockCorrectionFailed*) The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the Missing Offset Correction error or Missing Rate Correction error are active. The Clock Correction Failed Counter is reset to '0' at the end of an odd communication cycle if neither the Offset Correction Failed nor the Rate Correction Failed errors are active. The Clock Correction Failed Counter stops at 15.
ERRM[1:0]	Error Mode (*vPOC!ErrorMode*) Indicates the actual error mode of the POC. 00 =ACTIVE (green) 01 =PASSIVE (yellow) 10 =COMM_HALT (red) 11 =reserved
PTAC[4:0]	Passive to Active Count (*vAllowPassiveToActive*) Indicates the number of consecutive even / odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when **PTAC[4:0]** equals **FRSUCC1.PTA[4:0] -1**.

(3) Slot counter value (FRSCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSCV	R	0	0	0	0	0	SCCB10	SCCB9	SCCB8	SCCB7	SCCB6	SCCB5	SCCB4	SCCB3	SCCB2	SCCB1	SCCB0
0x0110	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SCCA10	SCCA9	SCCA8	SCCA7	SCCA6	SCCA5	SCCA4	SCCA3	SCCA2	SCCA1	SCCA0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
SCCA[10:0]	Slot Counter Channel A (*vSlotCounter[A]*) Current slot counter value on channel A. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 2047.
SCCB[10:0]	Slot Counter Channel B (*vSlotCounter[B]*) Current slot counter value on channel B. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 2047.

(4) Macrotick and cycle counter value (FRMTCCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMTCCV	R	0	0	0	0	0	0	0	0	0	0	CCV5	CCV4	CCV3	CCV2	CCV1	CCV0
0x0114	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MTV13	MTV12	MTV11	MTV10	MTV9	MTV8	MTV7	MTV6	MTV5	MTV4	MTV3	MTV2	MTV1	MTV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
MTV[13:0]	Macrotick Value (*vMacrotick*) Current Macrotick value. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 15999.
CCV[5:0]	Cycle Counter Value (*vCycleCounter*) Current cycle counter value. The value is incremented by the CC at the start of a communication cycle. Valid values are 0 to 63.

(5) Rate correction value (FRRCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRRCV	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0118	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	RCV11	RCV10	RCV9	RCV8	RCV7	RCV6	RCV5	RCV4	RCV3	RCV2	RCV1	RCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
RCV[11:0]	Rate Correction Value (*vRateCorrection*) Rate correction value (two's complement). Calculated internal rate correction value before limitation. If the RCV value exceeds the limits defined by **FRGTUC10.MRC[10:0]**, flag **FRSFS.RCLR** is set to '1'.

(6) Offset correction value (FROCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FROCV	R	0	0	0	0	0	0	0	0	0	0	0	0	0	OCV18	OCV17	OCV16
0x011C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OCV15	OCV14	OCV13	OCV12	OCV11	OCV10	OCV9	OCV8	OCV7	OCV6	OCV5	OCV4	OCV3	OCV2	OCV1	OCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
OCV[18:0]	Offset Correction Value *(vOffsetCorrection)* Offset correction value (two's complement). Calculated internal offset correction value before limitation. If the OCV value exceeds the limits defined by **FRGTUC10.MOC[13:0]**, flag **FRSFS.OCLR** is set to '1'.

Note The external rate / offset correction value is added to the limited rate / offset correction value.

(7) Sync frame status (FRSFS)

The maximum number of valid sync frames in a communication cycle is 15.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSFS	R	0	0	0	0	0	0	0	0	0	0	0	0	RCLR	MRCS	OCLR	MOCS
0x0120	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	VSBO3	VSBO2	VSBO1	VSBO0	VSBE3	VSBE2	VSBE1	VSBE0	VSAO3	VSAO2	VSAO1	VSAO0	VSAE3	VSAE2	VSAE1	VSAE0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
VSAE[3:0]	Valid Sync Frames Channel A, **even** communication cycle Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by **FRSUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each even communication cycle.
VSAO[3:0]	Valid Sync Frames Channel A, **odd** communication cycle Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by **FRSUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each odd communication cycle.
VSBE[3:0]	Valid Sync Frames Channel B, **even** communication cycle Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by **FRSUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each even communication cycle.
VSBO[3:0]	Valid Sync Frames Channel B, **odd** communication cycle Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by **FRSUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Note The bit fields above are only valid if the respective channel is assigned to the CC by **FRSUCC1.CCHA** or **FRSUCC1.CCHB**.

Bit name	Function
MOCS	Missing Offset Correction Signal The Missing Offset Correction flag signals to the Host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase. 1 = Missing offset correction signal 0 = Offset correction signal valid
OCLR	Offset Correction Limit Reached The Offset Correction Limit Reached flag signals to the Host, that the offset correction value has exceeded its limit as defined by **FRGTUC10.MOC[13:0]**. The flag is updated by the CC at start of offset correction phase. 1 = Offset correction limit reached 0 = Offset correction below limit
MRCS	Missing Rate Correction Signal The Missing Rate Correction flag signals to the Host, that no rate correction calculation can be performed because no pairs of even / odd sync frames were received. The flag is updated by the CC at start of offset correction phase. 1 = Missing rate correction signal 0 = Rate correction signal valid
RCLR	Rate Correction Limit Reached The Rate Correction Limit Reached flag signals to the Host, that the rate correction value has exceeded its limit as defined by **FRGTUC10.MRC[10:0]**. The flag is updated by the CC at start of offset correction phase. 1 = Rate correction limit reached 0 = Rate correction below limit

(8) Symbol window and NIT status (FRSWNIT)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRSWNIT	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0124	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol window related status information. Updated by the CC at the end of the symbol window for each channel. During startup the status data is not updated. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
SESA	Syntax Error in Symbol Window Channel A (*vSS!SyntaxErrorA*) 1 =Syntax error during symbol window detected on channel A 0 =No syntax error detected
SBSA	Slot Boundary Violation in Symbol Window Channel A (*vSS!BViolationA*) 1 =Slot boundary violation during symbol window detected on channel A 0 =No slot boundary violation detected
TCSA	Transmission Conflict in Symbol Window Channel A (*vSS!TxConflictA*) 1 =Transmission conflict in symbol window detected on channel A 0 =No transmission conflict detected
SESB	Syntax Error in Symbol Window Channel B (*vSS!SyntaxErrorB*) 1 =Syntax error during symbol window detected on channel B 0 =No syntax error detected
SBSB	Slot Boundary Violation in Symbol Window Channel B (*vSS!BViolationB*) 1 =Slot boundary violation during symbol window detected on channel B 0 =No slot boundary violation detected
TCSB	Transmission Conflict in Symbol Window Channel B (*vSS!TxConflictB*) 1 =Transmission conflict in symbol window detected on channel B 0 =No transmission conflict detected
MTSA	MTS Received on Channel A *(vSS!ValidMTSA)* Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. When this bit is set to '1', also interrupt flag **FRSIR.MTSA** is set to '1'. 1 = MTS symbol received on channel A 0 = No MTS symbol received on channel A
MTSB	MTS Received on Channel B *(vSS!ValidMTSB)* Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. When this bit is set to '1', also interrupt flag **FRSIR.MTSB** is set to '1'. 1 = MTS symbol received on channel B 0 = No MTS symbol received on channel B

NIT related status information. Updated by the CC at the end of the NIT for each channel:

Bit name	Function
SENA	Syntax Error during NIT Channel A (*vSS!SyntaxErrorA*) 1 =Syntax error during NIT detected on channel A 0 =No syntax error detected
SBNA	Slot Boundary Violation during NIT Channel A (*vSS!BViolationA*) 1 =Slot boundary violation during NIT detected on channel A 0 =No slot boundary violation detected
SENB	Syntax Error during NIT Channel B (*vSS!SyntaxErrorB*) 1 =Syntax error during NIT detected on channel B 0 =No syntax error detected
SBNB	Slot Boundary Violation during NIT Channel B (*vSS!BViolationB*) 1 =Slot boundary violation during NIT detected on channel B 0 =No slot boundary violation detected

(9) Aggregated channel status (FRACS)

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception. The aggregated channel status also includes status data from the symbol window and the network idle time. The status data is updated (set) after each slot and aggregated until it is reset by the Host. During startup the status data is not updated. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRACS	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0128	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	SBVB	CIB	CEDB	SEDB	VFRB	0	0	0	SBVA	CIA	CEDA	SEDA	VFRA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
VFRA	Valid Frame Received on Channel A (*vSS!ValidFrameA*) One or more valid frames were received on channel A in any static or dynamic slot during the observation period. 1 =Valid frame(s) received on channel A 0 =No valid frame received
SEDA	Syntax Error Detected on Channel A (*vSS!SyntaxErrorA*) One or more syntax errors in static or dynamic slots including symbol window and NIT were observed on channel A. 1 =Syntax error(s) observed on channel A 0 =No syntax error observed
CEDA	Content Error Detected on Channel A (*vSS!ContentErrorA*) One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period. 1 =Frame(s) with content error received on channel A 0 =No frame with content error received
CIA	Communication Indicator Channel A One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation. 1 =Valid frame(s) received on channel A in slots containing any additional communication 0 =No valid frame(s) received in slots containing any additional communication
SBVA	Slot Boundary Violation on Channel A (*vSS!BViolationA*) One or more slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots including symbol window and NIT). 1 =Slot boundary violation(s) observed on channel A 0 =No slot boundary violation observed

Bit name	Function
VFRB	Valid Frame Received on Channel B (*vSS!ValidFrameB*) One or more valid frames were received on channel B in any static or dynamic slot during the observation period. Reset under control of the Host. 1 =Valid frame(s) received on channel B 0 =No valid frame received
SEDB	Syntax Error Detected on Channel B (*vSS!SyntaxErrorB*) One or more syntax errors in static or dynamic slots including symbol window and NIT were observed on channel B. 1 =Syntax error(s) observed on channel B 0 =No syntax error observed
CEDB	Content Error Detected on Channel B (*vSS!ContentErrorB*) One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period. 1 =Frame(s) with content error received on channel B 0 =No frame with content error received
CIB	Communication Indicator Channel B One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation. 1 =Valid frame(s) received on channel B in slots containing any additional communication 0 =No valid frame(s) received in slots containing any additional communication
SBVB	Slot Boundary Violation on Channel B (*vSS!BViolationB*) One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots including symbol window and NIT). 1 =Slot boundary violation(s) observed on channel B 0 =No slot boundary violation observed

Note The set condition of flags **CIA** and **CIB** is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

When one of the flags **SEDB**, **CEDB**, **CIB**, **SBVB** changes from ‘0‘ to ‘1‘, interrupt flag **FREIR.EDB** is set to ’1’.When one of the flags **SEDA**, **CEDA**, **CIA**, **SBVA** changes from ‘0‘ to ‘1‘, interrupt flag **FREIR.EDA** is set to ’1’.

(10) Even sync ID [1…15] (FRESIDn)

Registers FRESID1 to FRESID15 hold the frame IDs of the sync frames received in **even** communication cycles used for clock synchronisation up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags **RXEA**, **RXEB** are set. The value is updated during the NIT of each even communication cycle. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRESIDn	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0130 - 0x0168	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RXEB	RXEA	0	0	0	0	EID9	EID8	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
EID[9:0]	Even Sync ID (*vsSyncIDListA,B even*) Sync frame ID even communication cycle.
RXEA	Received / Configured Even Sync ID on Channel A Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = **EID[9:0]** (ESID1 only). 1 = Sync frame received on channel A / node configured to transmit sync frames 0 = No sync frame received on channel A / node not configured to transmit sync frames
RXEB	Received / Configured Even Sync ID on Channel B Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = **EID[9:0]** (ESID1 only). 1 = Sync frame received on channel B / node configured to transmit sync frames 0 = No sync frame received on channel B / node not configured to transmit sync frames

(11) Odd sync ID [1…15] (FROSIDn)

Registers FROSID1 to FROSID15 hold the frame IDs of the sync frames received in **odd** communication cycles used for clock synchronisation up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register OSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags **RXOA**, **RXOB** are set. The value is updated during the NIT of each odd communication cycle. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FROSIDn	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0170 -0x01A8	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RXOB	RXOA	0	0	0	0	OID9	OID8	OID7	OID6	OID5	OID4	OID3	OID2	OID1	OID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
OID[9:0]	Odd Sync ID (*vsSyncIDListA,B odd*) Sync frame ID odd communication cycle.
RXOA	Received / Configured Odd Sync ID on Channel A Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = **OID[9:0]** (OSID1 only). 1 = Sync frame received on channel A / node configured to transmit sync frames 0 = No sync frame received on channel A / node not configured to transmit sync frames
RXOB	Received / Configured Odd Sync ID on Channel B Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = **OID[9:0]** (OSID1 only). 1 = Sync frame received on channel B / node configured to transmit sync frames 0 = No sync frame received on channel B / node not configured to transmit sync frames

(12) Network management vector [1…3] (FRNMVn)

The three network management registers hold the accrued NM vector (configurable 0 to 12 bytes). The accrued NM vector is generated by the CC by bit-wise ORing each NM vector received (valid static frames with PPI = ’1’) on each channel (see chapter Network Management).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

NMVn-bytes exceeding the configured NM vector length are not valid.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRNMVn	R	NM31	NM30	NM29	NM28	NM27	NM26	NM25	NM24	NM23	NM22	NM21	NM20	NM19	NM18	NM17	NM16
0x01B0-0x01B8	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	NM15	NM14	NM13	NM12	NM11	NM10	NM9	NM8	NM7	NM6	NM5	NM4	NM3	NM2	NM1	NM0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The table below shows the assignment of the data bytes to the network management vector.

Table 24-7 Assignment of data bytes to network management vector

Word \ Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NMV1	Data3								Data2								Data1								Data0							
NMV2	Data7								Data6								Data5								Data4							
NMV3	Data11								Data10								Data9								Data8							

24.4.7 Message buffer control registers

(1) Message RAM configuration (FRMRC)

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO. The register can be written during DEFAULT_CONFIG or CONFIG state only.

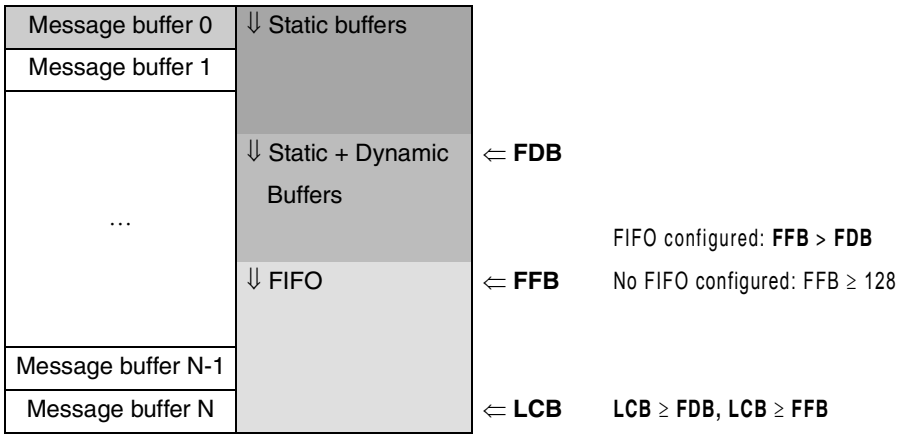
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMRC	R	0	0	0	0	0	SPLM*	SEC1*	SEC0*	LCB7*	LCB6*	LCB5*	LCB4*	LCB3*	LCB2*	LCB1*	LCB0*
0x0300	W																
Reset		0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FFB7*	FFB6*	FFB5*	FFB4*	FFB3*	FFB2*	FFB1*	FFB0*	FDB7*	FDB6*	FDB5*	FDB4*	FDB3*	FDB2*	FDB1*	FDB0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
FDB[7:0]	First Dynamic Buffer 0 = No group of message buffers exclusively for the static segment configured 1...127 = Message buffers 0 to **FDB** - 1 reserved for static segment ≥128 = No dynamic message buffers configured
FFB[7:0]	First Buffer of FIFO 0 = All message buffers assigned to the FIFO 1...127 = Message buffers from FFB to LCB assigned to the FIFO ≥128 = No message buffer assigned to the FIFO
LCB[7:0]	Last Configured Buffer 0…127= Number of message buffers is **LCB** + 1 ≥128= No message buffer configured
SEC[1:0]	Secure Buffers Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state. 00 = Reconfiguration of message buffers enabled with numbers < **FFB** enabled Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if **SPLM** = '1', also message buffer 1) is always locked 01 = Reconfiguration of message buffers with numbers < **FDB** and with numbers **FFB** locked and transmission of message buffers for static segment with numbers **FDB** disabled 10 = Reconfiguration of all message buffers locked 11 = Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers **FDB** disabled

Bit name	Function
SPLM	Sync Frame Payload Multiplex This bit is only evaluated if the node is configured as sync node (**FRSUCC1.TXSY** = '1') or for single slot mode operation (**SUCC1.TSM** = '1'). When this bit is set to '1' message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B. When this bit is set to '0', sync frames are transmitted from message buffer 0 with the same payload data on both channels. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen accordingly. 1 = Both message buffers 0 and 1 are locked against reconfiguration 0 = Only message buffer 0 locked against reconfiguration

Note In case the node is configured as sync node (**FRSUCC1.TXSY** = '1') or for single slot mode operation (**FRSUCC1.TSM** = '1'), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.



The programmer has to ensure that the configuration defined by **FDB[7:0]**, **FFB[7:0]**, and **LCB[7:0]** is valid. **The CC does not check for erroneous configurations!**

Note The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer.
In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the "Static Buffers" or at the beginning of the "Static + Dynamic Buffers" section.
The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conform configuration without a transmission slot in the static segment would still be operational.
The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via **FRWRHS2.PLC[6:0]** and **FRWRHS3.DP[10:0]**.When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.

(2) FIFO Rejection Filter (FRFRF)

The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO. The FRF register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRFRF	R	0	0	0	0	0	0	0	RNF*	RSS*	CYF6*	CYF5*	CYF4*	CYF3*	CYF2*	CYF1*	CYF0*
0x0304	W																
Reset		0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	FID10*	FID9*	FID8*	FID7*	FID6*	FID5*	FID4*	FID3*	FID2*	FID1*	FID0*	CH1*	CH0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
CH[1:0]	Channel Filter 11 =no reception 10 =receive only on channel A 01 =receive only on channel B 00 =receive on both channels Note: If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.
FID[10:0]	Frame ID Filter Determines the frame ID to be rejected by the FIFO. With the additional configuration of register **FRFRFM**, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When **FRFRFM.MFID[10:0]** is zero, a frame ID filter value of zero means that **no** frame ID is rejected. 0...2047 = Frame ID filter values
CYF[6:0]	Cycle Counter Filter The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles **not** belonging to the cycle set specified by **CYF[6:0]**, all frames are rejected. For details about the configuration of the cycle counter filter see Section Cycle Counter Filtering.
RSS	Reject in Static Segment If this bit is set, the FIFO is used only for the dynamic segment. 1 =Reject messages in static segment 0 =FIFO also used in static segment
RNF	Reject Null Frames If this bit is set, received null frames are not stored in the FIFO. 1 =Reject all null frames 0 =Null frames are stored in the FIFO

(3) FIFO rejection filter mask (FRFRFM)

The FIFO Rejection Filter Mask specifies which of the corresponding FRFRF bits are relevant for rejection filtering. If a bit is set, it indicates that the state of the corresponding bit in the FRFRF register will not affect whether or not the message is rejected by the FIFO. The **FRFRFM** register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRFRFM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0308	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MFID10*	MFID9*	MFID8*	MFID7*	MFID6*	MFID5*	MFID4*	MFID3*	MFID2*	MFID1*	MFID0*	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
MFID[10:0]	FIFO Rejection Filter Mask 1 =Ignore corresponding FIFO Rejection Filter bit. 0 =Corresponding FIFO Rejection Filter bit not ignored.

(4) FIFO Critical Level (FCL)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRRCL	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	CL7*	CL6*	CL5*	CL4*	CL3*	CL2*	CL1*	CL0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
CL[7:0]	Critical Level When the receive FIFO fill level **FRFSR.RFFL[7:0]** is equal or greater than the critical level configured by **CL[7:0]**, the receive FIFO critical level flag **FRFSR.RFCL** is set. If **CL[7:0]** is programmed to values > 128, bit **FRFSR.RFCL** is never set. When **FRFSR.RFCL** changes from '0' to '1' bit **FRSIR.RFCL** is set to '1', and if enabled, an interrupt is generated.

24.4.8 Message buffer status registers

(1) Message handler status (FRMHDS)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMHDS	R	0	MBU6	MBU5	MBU4	MBU3	MBU2	MBU1	MBU0	0	MBT6	MBT5	MBT4	MBT3	MBT2	MBT1	MBT0
0x0310	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	FMB6	FMB5	FMB4	FMB3	FMB2	FMB1	FMB0	CRAM	MFMB	FMBD	PTBF2	PTBF1	DMR	POBF	PIBF
	W																
Reset		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

A flag is cleared by writing a ’1’ to the corresponding bit position. Writing a ’0’ has no effect on the flag. The register will also be cleared by hard reset or by CHI command CLEAR_RAMS.

Bit name	Function
PIBF	Parity Error Input Buffer RAM 1,2 1 =Parity error occurred when Message Handler read Input Buffer RAM 1,2 0 =No parity error
POBF	Parity Error Output Buffer RAM 1,2 1 =Parity error occurred when CPU read Output Buffer RAM 1,2 0 =No parity error
DMR	Double bit error Message RAM 1 =Double bit occurred when reading the Message RAM 0 =No parity error
PTBF1	Parity Error Transient Buffer RAM A 1 =Parity error occurred when reading Transient Buffer RAM A 0 =No parity error
PTBF2	Parity Error Transient Buffer RAM B 1 =Parity error occurred when reading Transient Buffer RAM B 0 =No parity error

Note When one of the flags **PIBF**, **POBF**, **PMR**, **PTBF1**, **PTBF2** changes from ’0’ to ’1’ **FREIR.PERR** is set to ’1’.

Bit name	Function
FMBD	Faulty Message Buffer Detected 1 =Message buffer referenced by **FMB[6:0]** holds faulty data due to a parity error 0 =No faulty message buffer
MFMB	Multiple Faulty Message Buffers detected 1 =Another faulty message buffer was detected while flag **FMBD** is set 0 =No additional faulty message buffer
CRAM	Clear all internal RAM’s Signals that execution of the CHI command CLEAR_RAMS is ongoing (all bits of all internal RAM blocks are written to ’0’). The bit is set by hardware reset or by CHI command CLEAR_RAMS. 1 =Execution of the CHI command CLEAR_RAMS ongoing 0 =No execution of the CHI command CLEAR_RAMS

Bit name	Function
FMB[6:0]	Faulty Message Buffer Double bit error occurred when reading from or writing to the message buffer referenced by **FMB[6:0]**. Value only valid when one of the flags **PIBF, PMR, PTBF1, PTBF2,** and flag **FMBD** is set. Is not updated while flag FMBD is set.
MBT[6:0]	Message Buffer Transmitted Number of last successfully transmitted message buffer. If the message buffer is configured for single-shot mode, the respective TXR flag in the Transmission Request Register 1,2,3,4 was reset.
MBU[6:0]	Message Buffer Updated Number of message buffer that was updated last by the CC. For this message buffer the respective **ND** and / or **MBC** flag in the **FRNDAT1/2/3/4** registers and the **FRMBSC1/2/3/4** registers are also set.

Note **MBT[6:0]** and **MBU[6:0]** are reset when the CC leaves CONFIG state or enters STARTUP state.

(2) Last Dynamic Transmit Slot (FRLDTS)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRLDTS	R	0	0	0	0	0	LDTB10	LDTB9	LDTB8	LDTB7	LDTB6	LDTB5	LDTB4	LDTB3	LDTB2	LDTB1	LDTB0
0x0314	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	LDTA10	LDTA9	LDTA8	LDTA7	LDTA6	LDTA5	LDTA4	LDTA3	LDTA2	LDTA1	LDTA0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
LDTA[10:0]	Last Dynamic Transmission Channel A Value of vSlotCounter[A] at the time of the last frame transmission on channel A in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.
LDTB[10:0]	Last Dynamic Transmission Channel B Value of vSlotCounter[B] at the time of the last frame transmission on channel B in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

(3) FIFO Status Register (FRFSR)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRFSR	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0318	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RFFL7	RFFL6	RFFL5	RFFL4	RFFL3	RFFL2	RFFL1	RFFL0	0	0	0	0	0	RFO	RFCL	RFNE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state..

Bit name	Function
RFNE	Receive FIFO Not Empty This flag is set by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, interrupt flag **FRSIR.RFNE** is set. The bit is reset after the Host has read all message from the FIFO. 1 = Receive FIFO is not empty 0 = Receive FIFO is empty
RFCL	Receive FIFO Critical Level This flag is set when the receive FIFO fill level **FRRFFL[7:0]** is equal or greater than the critical level as configured by **FRFCL.CL[7:0]**. The flag is cleared by the CC as soon as **FRRFFL[7:0]** drops below **FRFCL.CL[7:0]**. When **RFCL** changes from '0' to '1' bit **FRSIR.RFCL** is set to '1', and if enabled, an interrupt is generated. 1 = Receive FIFO critical level reached 0 = Receive FIFO below critical level
RFO	Receive FIFO Overrun The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag **FREIR.RFO** is set.The flag is cleared by the next FIFO read access issued by the Host. 1 = A receive FIFO overrun has been detected 0 = No receive FIFO overrun detected
RFFL[7:0]	Receive FIFO Fill Level Number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.

(4) Message Handler Constraints Flags (FRMHDF)

Some constraints exist for the Message Handler regarding eray_bclk frequency, Message RAM configuration, and FlexRay bus traffic (see Addendum to E-Ray FlexRay IP-Module Specification). To simplify software development, constraints violations are reported by setting flags in the **FRMHDF**.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMHDF	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x031C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	WAHP	0	0	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A flag is cleared by writing a ’1’ to the corresponding bit position. Writing a ’0’ has no effect on the flag. A hard reset will also clear the register. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit name	Function
SNUA	Status Not Updated Channel A This flag is set by the CC when the Message Handler, due to overload condition, was not able to update a message buffer’s status MBS with respect to channel A. 1 = MBS for channel A not updated 0 = No overload condition occurred when updating MBS for channel A
SNUB	Status Not Updated Channel B This flag is set by the CC when the Message Handler, due to overload condition, was not able to update a message buffer’s status MBS with respect to channel B. 1 = MBS for channel B not updated 0 = No overload condition occurred when updating MBS for channel B
FNFA	Find Sequence Not Finished Channel A This flag is set by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel A. 1 = Find sequence not finished for channel A 0 = No find sequence not finished for channel A
FNFB	Find Sequence Not Finished Channel B This flag is set by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel B. 1 = Find sequence not finished for channel B 0 = No find sequence not finished for channel B

Bit name	Function
TBFA	Transient Buffer Access Failure A This flag is set by the CC when a read or write access to TBF A requested by PRT A could not complete within the available time. 1 = TBF A access failure 0 = No TBF A access failure
TBFB	Transient Buffer Access Failure B This flag is set by the CC when a read or write access to TBF B requested by PRT B could not complete within the available time. 1 = TBF B access failure 0 = No TBF B access failure
WAHP	Write Attempt to Header Partition Outside DEFAULT_CONFIG and CONFIG state this flag is set by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses. 1 = Write attempt to header partition 0 = No write attempt to header partition

Note When one of the flags **SNUA**, **SNUB**, **FNFA**, **FNFB**, **TBFA**, **TBFB**, **WAHP** changes from ’0’ to ’1’, interrupt flag **FREIR.MHF** is set to ’1’.

(5) Transmission request 1/2/3/4 (FRTXRQ1/2/3/4)

The four registers reflect the state of the **TXR** flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining **TXR** flags have no meaning.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRTXRQ4	R	TXR127	TXR126	TXR125	TXR124	TXR123	TXR122	TXR121	TXR120	TXR119	TXR118	TXR117	TXR116	TXR115	TXR114	TXR113	TXR112
0x032C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR111	TXR110	TXR109	TXR108	TXR107	TXR106	TXR105	TXR104	TXR103	TXR102	TXR101	TXR100	TXR99	TXR98	TXR97	TXR96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRTXRQ3	R	TXR95	TXR94	TXR93	TXR92	TXR91	TXR90	TXR89	TXR88	TXR87	TXR86	TXR85	TXR84	TXR83	TXR82	TXR81	TXR80
0x0328	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR79	TXR78	TXR77	TXR76	TXR75	TXR74	TXR73	TXR72	TXR71	TXR70	TXR69	TXR68	TXR67	TXR66	TXR65	TXR64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRTXRQ2	R	TXR63	TXR62	TXR61	TXR60	TXR59	TXR58	TXR57	TXR56	TXR55	TXR54	TXR53	TXR52	TXR51	TXR50	TXR49	TXR48
0x0324	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR47	TXR46	TXR45	TXR44	TXR43	TXR42	TXR41	TXR40	TXR39	TXR38	TXR37	TXR36	TXR35	TXR34	TXR33	TXR32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRTXRQ1	R	TXR31	TXR30	TXR29	TXR28	TXR27	TXR26	TXR25	TXR24	TXR23	TXR22	TXR21	TXR20	TXR19	TXR18	TXR17	TXR16
0x0320	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR15	TXR14	TXR13	TXR12	TXR11	TXR10	TXR9	TXR8	TXR7	TXR6	TXR5	TXR4	TXR3	TXR2	TXR1	TXR0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
TXR[127:0]	Transmission Request If the flag is set, the respective message buffer is ready for transmission respectively transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

(6) New data 1/2/3/4 (FRNDAT1/2/3/4)

The four registers reflect the state of the **ND** flags of all configured message buffers. **ND** flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining **ND** flags have no meaning. The registers are reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRNDAT4	R	ND127	ND126	ND125	ND124	ND123	ND122	ND121	ND120	ND119	ND118	ND117	ND116	ND115	ND114	ND113	ND112
0x033C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND111	ND110	ND109	ND108	ND107	ND106	ND105	ND104	ND103	ND102	ND101	ND100	ND99	ND98	ND97	ND96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRNDAT3	R	ND95	ND94	ND93	ND92	ND91	ND90	ND89	ND88	ND87	ND86	ND85	ND84	ND83	ND82	ND81	ND80
0x0338	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND79	ND78	ND77	ND76	ND75	ND74	ND73	ND72	ND71	ND70	ND69	ND68	ND67	ND66	ND65	ND64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRNDAT2	R	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
0x0334	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRNDAT1	R	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
0x0330	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
ND[127:0]	New Data The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An **ND** flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

(7) Message buffer status changed 1/2/3/4 (FRMBSC1/2/3/4)

The four registers reflect the state of the **MBC** flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining **MBC** flags have no meaning. The registers are reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMBSC4	R	MBC127	MBC126	MBC125	MBC124	MBC123	MBC122	MBC121	MBC120	MBC119	MBC118	MBC117	MBC116	MBC115	MBC114	MBC113	MBC112
0x034C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC111	MBC110	MBC109	MBC108	MBC107	MBC106	MBC105	MBC104	MBC103	MBC102	MBC101	MBC100	MBC99	MBC98	MBC97	MBC96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMBSC3	R	MBC95	MBC94	MBC93	MBC92	MBC91	MBC90	MBC89	MBC88	MBC87	MBC86	MBC85	MBC84	MBC83	MBC82	MBC81	MBC80
0x0348	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC79	MBC78	MBC77	MBC76	MBC75	MBC74	MBC73	MBC72	MBC71	MBC70	MBC69	MBC68	MBC67	MBC66	MBC65	MBC64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMBSC2	R	MBC63	MBC62	MBC61	MBC60	MBC59	MBC58	MBC57	MBC56	MBC55	MBC54	MBC53	MBC52	MBC51	MBC50	MBC49	MBC48
0x0344	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC47	MBC46	MBC45	MBC44	MBC43	MBC42	MBC41	MBC40	MBC39	MBC38	MBC37	MBC36	MBC35	MBC34	MBC33	MBC32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMBSC1	R	MBC31	MBC30	MBC29	MBC28	MBC27	MBC26	MBC25	MBC24	MBC23	MBC22	MBC21	MBC20	MBC19	MBC18	MBC17	MBC16
0x0340	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBC15	MBC14	MBC13	MBC12	MBC11	MBC10	MBC9	MBC8	MBC7	MBC6	MBC5	MBC4	MBC3	MBC2	MBC1	MBC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
MBC[127:0]	Message Buffer Status Changed An **MBC** flag is set whenever the Message Handler changes one of the status flags **VFRA**, **VFRB**, **SEOA**, **SEOB**, **CEOA**, **CEOB**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB** in the header section (see 4.11.5 Message Buffer Status (MBS) and 5.12.1 Header Partition, header 4) of the respective message buffer. An **MBC** flag is reset when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

24.4.9 Identification Registers

(1) Core Release Register (FRCREL)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRCREL	R	REL3	REL2	REL1	REL0	STEP7	STEP6	STEP5	STEP4	STEP3	STEP2	STEP1	STEP0	YEAR3	YEAR2	YEAR1	YEAR0
0x03F0	W																
Reset		0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MON7	MON6	MON5	MON4	MON3	MON2	MON1	MON0	DAY7	DAY6	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
	W																
Reset		0	0	0	0	0	1	0	1	1	0	0	1	1	0	0	1

Bit name	Function
DAY[7:0]	Design Time Stamp, Day Two digits, BCD-coded. 0x19: The IP was desinged on the 19th of May.
MON[7:0]	Design Time Stamp, Month Two digits, BCD-coded. 0x05: The IP was designed in May.
YEAR[3:0]	Design Time Stamp, Year One digit, BCD-coded. 0x6: The IP was designed in the year 2006.
STEP[7:0]	Step of Core Release Two digits, BCD-coded. 0x00: The IP release is the Revision 1.0.0
REL[3:0]	Core Release One digit, BCD-coded. 0x1: The IP release is the Revision 1.0.0

Table below shows how releases are coded in register CREL.

Table 24-8 Coding for releases

Release	Step	Sub-Step	Name
0	7	0	Beta2
0	7	1	Beta2ct
0	7	2	Revision 1.0 RC1
1	0	0	Revision 1.0.0

(2) Endian Register (FRENDN)

The Message Handler Status register gives the Host CPU access to the actual state of the Message Handler.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRENDN	R	ETV31	ETV30	ETV29	ETV28	ETV27	ETV26	ETV25	ETV24	ETV23	ETV22	ETV21	ETV20	ETV19	ETV18	ETV17	ETV16
0x03F4	W																
Reset		0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ETV15	ETV14	ETV13	ETV12	ETV11	ETV10	ETV9	ETV8	ETV7	ETV6	ETV5	ETV4	ETV3	ETV2	ETV1	ETV0
	W																
Reset		0	0	0	0	0	1	0	1	1	0	0	1	1	0	0	1

Bit name	Function
ETV[31:0]	Endianness Test Value The endianness test value is 0x87654321.

24.4.10 Input buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in Section 4.11.5 Message Buffer Status (MBS) is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via **FRWRHS2.PLC[6.0]** and **FRWRHS3.DP[10:0]**. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in Section Data Transfer from Input Buffer to Message RAM.

(1) Write data section [1…64] (FRWRDSn)

Holds the data words to be transferred to the data section of the addressed message buffer. The data words (DWn) are written to the Message RAM in transmission order from DW1 (byte0, byte1) to DWPL (PL = number of data words as defined by the payload length configured **FRWRHS2.PLC[6:0]**).

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRWRDSn	R	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
0x0400 - 0x04FC	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
MD[31:0]	Message Data **MD[7:0]** = DW2n-1, byte4n-4 **MD[15:8]** = DW2n-1, byte4n-3 **MD[23:16]** = DW2n, byte4n-2 **MD[31:24]** = DW2n, byte4n-1

Note DW127 is located on WRDS64.MD[15:0]. In this case WRDS64.MD[31:16] is unused (no valid data). The Input Buffer RAMs are initialized to zero when leaving hard reset or by CHI command CLEAR_RAMS.
Transmission order on the FlexRay bus is FRWRDSn[7:0], FRWRDSn[15:8], FRWRDSn[23:16], FRWRDSn[31:24] with the most significant bit transmitted first. To check how the E-Ray's endianness matches with the Host CPU's endianness, read register ENDN.

(2) Write header section 1 (FRWRHS1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRWRHS1	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
0x0500	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
FID[10:0]	Frame ID Frame ID of the selected message buffer. The frame ID defines the slot number for transmission / reception of the respective message. **Message buffers with frame ID = '0' are considered as not valid.**
CYC[6:0]	Cycle Code The 7-bit cycle code determines the cycle set used for cycle counter filtering. For details about the configuration of the cycle code see chapter Cycle Counter Filtering.
CHA, CHB	Channel Filter Control The 2-bit channel filtering field associated with each buffer serves as a filter for Receive Buffers, and as a control field for transmit buffers.

CHA	CHB	**Transmit Buffer** transmit frame on	**Receive Buffer** store frame received from
1	1	both channels (static segment only)	channel A or B (store first semantically valid frame, static segment only)
1	0	channel A	channel A
0	1	channel B	channel B
0	0	no transmission	ignore frame

Note If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to '1', no frames are transmitted resp. received frames are ignored (same function as **CHA = CHB** = '0')

Bit name	Function
CFG	Message Buffer Configuration Bit This bit is used to configure the corresponding buffer as Transmit Buffer or as Receive Buffer. For message buffers belonging to the receive FIFO the bit is not evaluated. 1 =The corresponding buffer is configured as **Transmit Buffer** 0 =The corresponding buffer is configured as **Receive Buffer**
PPIT	Payload Preamble Indicator Transmit This bit is used to control the state of the Payload Preamble Indicator in transmit frames. If the bit is set in a static message buffer, the respective message buffer holds network management information. If the bit is set in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the E-Ray module, but can be done by the Host CPU. 1 =Payload Preamble Indicator set 0 =Payload Preamble Indicator not set
TXM	Transmission Mode This bit is used to select the transmission mode. 1 =Single-shot mode 0 =Continuous mode
MBI	Message Buffer Interrupt This bit enables the receive / transmit interrupt for the corresponding message buffer. After a dedicated receive buffer has been updated by the Message Handler, flag **RXI** and /or **MBSI** in the Status Interrupt register are set. After successful transmission the **TXI** flag in the Status Interrupt register is set. 1 =The corresponding message buffer interrupt is enabled 0 =The corresponding message buffer interrupt is disabled

(3) Write header section 2 (FRWRHS2)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRWRHS2	R	0	0	0	0	0	0	0	0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
0x0504	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
CRC[10:0]	Header CRC (*vRF!Header!HeaderCRC*) Receive Buffer: Configuration not required Transmit Buffer: Header CRC calculated and configured by the Host For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by **FRMHDC.SFDL[6:0]**.
PLC[6:0]	Payload Length Configured Length of data section (number of 2-byte words) as configured by the Host. During static segment the static frame payload length as configured by **FRMHDC.SFDL[6:0]** defines the payload length for all static frames. If the payload length configured by **PLC[6:0]** is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is logical zero.

(4) Write header section 3 (FRWRHS3)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRWRHS3	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0508	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
DP[10:0]	Data Pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

(5) Input buffer command mask (FRIBCM)

Configures how the message buffer in the Message RAM selected by the Input Buffer Command Request register is updated. When IBF Host and IBF Shadow are swapped, also mask bits **LHSH**, **LDSH**, and **STXRH** are swapped with bits **LHSS**, **LDSS**, and **STXRS** to keep them attached to the respective Input Buffer transfer.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRIBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRS	LDSS	LHSS
0x0510	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRH	LDSH	LHSH
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
LHSH	Load Header Section Host 1 =Header section selected for transfer from Input Buffer to the Message RAM 0 =Header section is not updated
LDSH	Load Data Section Host 1 =Data section selected for transfer from Input Buffer to the Message RAM 0 =Data section is not updated
STXRH	Set Transmission Request Host If this bit is set to '1', the **TXR** flag for the selected message buffer is set in the **FRTXRQ1/2/3/4** registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed. **TXR** is evaluated for transmit buffers only. 1 = Set **TXR** flag, transmit buffer released for transmission 0 = Reset **TXR** flag
LHSS	Load Header Section Shadow 1 = Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished) 0 = Header section is not updated
LDSS	Load Data Section Shadow 1 = Data section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished) 0 = Data section is not updated
STXRS	Set Transmission Request Shadow 1 = Set **TXR** flag, transmit buffer released for transmission (operation ongoing or finished) 0 = Reset **TXR** flag

(6) Input buffer command request (FRIBCR)

When the Host writes the number of a target message buffer in the Message RAM to **FRIBRH[6:0]** in the Input Buffer Command Request register, IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under **FRIBRH[6:0]** and **FRIBRS[6:0]** are also swapped (see also *"Host access to message RAM" on page 1295*).

With this write operation the **FRIBSYS** bit in the Input Buffer Command Request register is set to '1'. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by **FRIBRS[6:0]**.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may configure the next message in the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, the **FRIBSYS** bit is set back to '0' and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to **FRIBRH[6:0]**.

If a write access to **FRIBRH[6:0]** occurs while **FRIBSYS** is '1', **FRIBSYH** is set to '1'. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, **FRIBSYH** is reset to '0'. **FRIBSYS** remains set to '1', and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under **FRIBRH[6:0]** and **FRIBRS[6:0]** are also swapped.

Any write access to an Input Buffer register while both **FRIBSYS** and **FRIBSYH** are set will cause the error flag **FREIR.IIBA** to be set. In this case the Input Buffer will not be changed.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRIBCR	R	IBSYS	0	0	0	0	0	0	0	0	IBRS6	IBRS5	IBRS4	IBRS3	IBRS2	IBRS1	IBRS0
0x0514	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	IBSYH	0	0	0	0	0	0	0	0	IBRH6	IBRH5	IBRH4	IBRH3	IBRH2	IBRH1	IBRH0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
IBRH[6:0]	Input Buffer Request Host Selects the target message buffer in the Message RAM for data transfer from Input Buffer. Valid values are 0x0000 to 0x007F (0...127).
IBSYH	Input Buffer Busy Host Set to '1' by writing **IBRH[6:0]** while **IBSYS** is still '1'. After the ongoing transfer between IBF Shadow and the Message RAM has completed, the **IBSYH** is set back to '0'. 1 = Request while transfer between IBF Shadow and Message RAM in progress 0 = No request pending
IBRS[6:0]	Input Buffer Request Shadow Number of the target message buffer actually updated / lately updated. Valid values are 0x0000 to 0x007F (0...127).
IBSYS	Input Buffer Busy Shadow Set to '1' after writing **IBRH[6:0]**. When the transfer between IBF Shadow and the Message RAM has completed, **IBSYS** is set back to '0'. 1 =Transfer between IBF Shadow and Message RAM in progress 0 =Transfer between IBF Shadow and Message RAM completed

24.4.11 Output buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in Section Data Transfer from Message RAM to Output Buffer.

(1) Read data section [1…64] (FRRDDSn)

Holds the data words read from the data section of the addressed message buffer. The data words (DWn) are read from the Message RAM in reception order from DW1 (byte0, byte1) to DWPL (PL = number of data words as defined by the payload length configured **FRRDHS2.PLC[6:0]**).

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRRDDSn	R	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
0x0600 - 0x06FC	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
MD[31:0]	Message Data **MD[7:0]** = DW2n-1, byte4n-4 **MD[15:8]** = DW2n-1, byte4n-3 **MD[23:16]** = DW2n, byte4n-2 **MD[31:24]** = DW2n, byte4n-1

Note DW127 is located on RDDS64.MD[15:0]. In this case RDDS64.MD[31:16] is unused (no valid data). The Output Buffer RAMs are initialized to zero when leaving hard reset or by CHI command CLEAR_RAMS.

(2) Read header section 1 (FRRDHS1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRRDHS1	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
0x0700	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Values as configured by the Host via WRHS1:

Bit name	Function
ID[10:0]	Frame ID
CYC[6:0]	Cycle Code
CHA, CHB	Channel Filter Control
CFG	Message Buffer Direction Configuration Bit
PPIT	Payload Preamble Indicator Transmit
TXM	Transmission Mode
MBI	Message Buffer Interrupt

In case that the message buffer read from the Message RAM belongs to the receive FIFO, **FID[10:0]** holds the received frame ID, while **CYC[6:0]**, **CHA**, **CHB**, **CFG**, **PPIT**, **TXM**, and **MBI** are reset to ’0’.

(3) Read header section 2 (FRRDHS2)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRRDHS2	R	0	PLR6	PLR5	PLR4	PLR3	PLR2	PLR1	PLR0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
0x0704	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
CRC[10:0]	Header CRC (*vRF!Header!HeaderCRC*) Receive Buffer: Header CRC updated from received frame Transmit Buffer: Header CRC calculated and configured by the Host
PLC[6:0]	Payload Length Configured Length of data section (number of 2-byte words) as configured by the Host.
PLR[6:0]	Payload Length Received (*vRF!Header!Length*) Payload length value updated from received data frames (exception: if message buffer belongs to the receive FIFO **PLR[6:0]** is also updated from received null frames)

When a message is stored into a message buffer the following behaviour with respect to payload

length received and payload length configured is implemented:

PLR[6:0] > **PLC[6:0]**:
The payload data stored in the message buffer is truncated to the payload length configured if **PLC[6:0]** even or else truncated to **PLC[6:0]** + 1.

PLR[6:0] PLC[6:0]:
The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by **PLC[6:0]** are filled with undefined data

PLR[6:0] = zero:
The message buffers data section is filled with undefined data

PLC[6:0] = zero:
Message buffer has no data section configured. No data is stored into the message buffers data section.

Note The Message RAM is organized in 4-byte words. When received data is stored into a message buffers data section, the number of 2-byte data words written into the message buffer is **PLC[6:0]** rounded to the next even value. **PLC[6:0]** should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.

(4) Read header section 3 (FRRDHS3)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRRDHS3	R	0	0	RES	PPI	NFI	SYN	SFI	RCI	0	0	RCC5	RCC4	RCC3	RCC2	RCC1	RCC0
0x0708	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
DP[10:0]	Data Pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.
RCC[5:0]	Receive Cycle Count (*vRF!Header!CycleCount*) Cycle counter value updated from received frame.
RCI	Received on Channel Indicator (vSS!Channel) Indicates the channel from which the received data frame was taken to update the respective receive buffer. 1 =Frame received on channel A 0 =Frame received on channel B
SFI	Startup Frame Indicator (*vRF!Header!SuFIndicator*) A startup frame is marked by the startup frame indicator. 1 =The received frame is a startup frame 0 =No startup frame received
SYN	Sync Frame Indicator (*vRF!Header!SyFIndicator*) A sync frame is marked by the sync frame indicator. 1 =The received frame is a sync frame 0 =No sync frame received
NFI	Null Frame Indicator (*vRF!Header!NFIndicator*) Is set to '1' after storage of the first received data frame. 1 =Received frame is **not** a Null frame 0 =Received frame is a Null frame
PPI	Payload Preamble Indicator (*vRF!Header!PPIndicator*) The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 1 =Static segment: Network management vector at the beginning of the payload Dynamic segment: Message ID at the beginning of the payload 0 =The payload segment of the received frame does not contain a network management vector or a message ID
RES	Reserved Bit (*vRF!Header!Reserved*) Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.

Note Header 3 is updated from data frames only.

(5) Message buffer status (FRMBS)

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer. The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state. If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated. The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all MBS flags are reset to zero independent of which FRIBCM bits are set or not. For details about receive / transmit filtering see Sections Filtering and Masking, Transmit Process, and Receive Process. Whenever the Message Handler changes one of the flags **VFRA**, **VFRB**, **SEOA**, **SEOB**, **CEOA**, **CEOB**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB** the respective message buffer's **MBC** flag in registers **FRMBSC1/2/3/4** is set.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRMBS	R	0	0	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	0	0	CCS5	CCS4	CCS3	CCS2	CCS1	CCS0
0x070C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FTB	FTA	0	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **Transmit & receive buffer**

Bit name	Function
VFRA	Valid Frame Received on Channel A A valid frame indication is set if a valid frame was received on channel A. 1 =Valid frame received on channel A 0 =No valid frame received on channel A
VFRB	Valid Frame Received on Channel B A valid frame indication is set if a valid frame was received on channel B. 1 =Valid frame received on Channel B 0 =No valid frame received on Channel B
SEOA	Syntax Error Observed on Channel A (*vSS!SyntaxErrorA*) A syntax error was observed in the configured slot on channel A. 1 =Syntax error observed on channel A 0 =No syntax error observed on channel A
SEOB	Syntax Error Observed on Channel B (*vSS!SyntaxErrorB*) A syntax error was observed in the configured slot on channel B. 1 =Syntax error observed on channel B 0 =No syntax error observed on channel B
CEOA	Content Error Observed on Channel A (*vSS!ContentErrorA*) A content error was observed in the configured slot on channel A. 1 =Content error observed on channel A 0 =No content error observed on channel A

Bit name	Function
CEOB	Content Error Observed on Channel B (*vSS!ContentErrorB*) A content error was observed in the configured slot on channel B. 1 =Content error observed on channel B 0 =No content error observed on channel B
SVOA	Slot Boundary Violation Observed on Channel A (*vSS!BViolationA*) A slot boundary violation was observed on channel A i.e. channel active at the start or at the end of the configured slot. 1 =Slot boundary violation observed on channel A 0 =No slot boundary violation observed on channel A
SVOB	Slot Boundary Violation Observed on Channel B (*vSS!BViolationB*) A slot boundary violation was observed on channel B i.e. channel active at the start or at the end of the configured slot. 1 =Slot boundary violation observed on channel B 0 =No slot boundary violation observed on channel B
TCIA	Transmission Conflict Indication Channel A (*vSS!TxConflictA*) A transmission conflict indication is set if a transmission conflict has occurred on channel A. 1 =Transmission conflict occurred on channel A 0 =No transmission conflict occurred on channel A
TCIB	Transmission Conflict Indication Channel B (*vSS!TxConflictB*) A transmission conflict indication is set if a transmission conflict has occurred on channel B. 1 =Transmission conflict occurred on Channel B 0 =No transmission conflict occurred on Channel B
ESA	Empty Slot Channel A In an empty slot there is no activity on the bus i.e. no frame transmission detected. 1 =No bus activity detected in the configured slot on channel A 0 =Bus activity detected in the configured slot on channel A
ESB	Empty Slot Channel B In an empty slot there is no activity on the bus i.e. no frame transmission detected. 1 =No bus activity detected in the configured slot on channel B 0 =Bus activity detected in the configured slot on channel B
MLST	Message Lost The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame. Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset by a Host write to the message buffer via IBF or when a new message is stored into the message buffer **after** the message buffers **ND** flag was reset by reading out the message buffer via OBF. 1 = Unprocessed message was overwritten 0 = No message lost
FTA	Frame Transmitted on Channel A Indicates that this node has transmitted a data frame in the configured slot on channel A. 1 = Data frame transmitted on channel A 0 = No data frame transmitted on channel A
FTB	Frame Transmitted on Channel B Indicates that this node has transmitted a data frame in the configured slot on channel B. 1 = Data frame transmitted on channel B 0 = No data frame transmitted on channel B

Note The FlexRay protocol specification requires that **FTA**, and **FTB** can only be reset by the Host. Therefore the Cycle Count Status **CCS[5:0]** for these bits is only valid for the cycle where the bits are set to '1'.r

Bit name	Function
CCS[5:0]	Cycle Count Status Actual cycle count when status was updated.

For receive buffers (CFG = '0') the following status bits are updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.

Bit name	Function
RCIS	Received on Channel Indicator Status (vSS!Channel) Indicates the channel on which the frame was received. 1 = Frame received on channel A 0 = Frame received on channel B
SFIS	Startup Frame Indicator Status (vRF!Header!SuFIndicator) A startup frame is marked by the startup frame indicator. 1 = The received frame is a startup frame 0 = No startup frame received
SYNS	Sync Frame Indicator Status (vRF!Header!SyFIndicator) A sync frame is marked by the sync frame indicator. 1 = The received frame is a sync frame 0 = No sync frame received
NFIS	Null Frame Indicator Status (vRF!Header!NFIndicator) If set to '0' the payload segment of the received frame contains no usable data. 1 = Received frame is **not** a null frame 0 = Received frame is a null frame
PPIS	Payload Preamble Indicator Status (vRF!Header!PPIndicator) The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 1 = Static segment: Network management vector at the beginning of the payload Dynamic segment: Message ID at the beginning of the payload 0 = The payload segment of the received frame does not contain a network management vector or a message ID
RESS	Reserved Bit Status (vRF!Header!Reserved) Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.

(6) Output buffer command mask (FROBCM)

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by FROBCR.OBRS[6:0]. Mask bits **RDSS** and **RHSS** are copied to the register internal storage when a Message RAM transfer is requested by **FROBCR.REQ**. When OBF Host and OBF Shadow are swapped, mask bits **RDSH** and **RHSH** are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer. The data transfer between Output Buffer and Message RAM is described in detail in Section 21.5.11.2 Data Transfer from Message RAM to Output Buffer.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FROBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSH	RHSH
0x0710	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSS	RHSS
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
RHSS	Read Header Section Shadow 1 =Header section selected for transfer from Message RAM to Output Buffer 0 =Header section is not read
RDSS	Read Data Section Shadow 1 =Data section selected for transfer from Message RAM to Output Buffer 0 =Data section is not read
RHSH	Read Header Section Host 1 =Header section selected for transfer from Message RAM to Output Buffer 0 =Header section is not read
RDSH	Read Data Section Host 1 =Data section selected for transfer from Message RAM to Output Buffer 0 =Data section is not read

Note After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag **MBC** of the selected message buffer in the **FRMBSC1/2/3/4** registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag **ND** of the selected message buffer in the **FRNDAT1/2/3/4** registers is cleared.

(7) Output buffer command request (FROBCR)

After setting bit **REQ** to ’1’ while **OBSYS** is ’0’, **OBSYS** is automatically set to ’1’, **OBRS[6:0]** is copied to the register internal storage, mask bits **FROBCM.RDSS** and **FROBCM.RHSS** are copied to register FROBCM internal storage, and the transfer of the message buffer selected by **OBRS[6:0]** from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signalled by setting **OBSYS** back to ’0’.

By setting bit **VIEW** to ’1’ while **OBSYS** is ’0’, OBF Host and OBF Shadow are swapped. Additionally mask bits **FROBCM.RDSH** and **FROBCM.RHSH** are swapped with the register FROBCM internal storage to keep them attached to the respective Output Buffer transfer. **OBRH[6:0]** signals the number of the message buffer currently accessible by the Host.

If bits **REQ** and **VIEW** are set to ’1’ with the same write access while **OBSYS** is ’0’, **OBSYS** is automatically set to ’1’ and OBF Shadow and OBF Host are swapped. Additionally mask bits **FROBCM.RDSH** and **FROBCM.RHSH** are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards **OBRS[6:0]** is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signalled by setting **OBSYS** back to ’0’.

Any write access to **FROBCR[15:8]** while **OBSYS** is set will cause the error flag **FREIR.IOBA** to be set. In this case the Output Buffer will not be changed.

The data transfer between Output Buffer and Message RAM is described in detail in Section 21.5.11.2 Data Transfer from Message RAM to Output Buffer.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FROBCR	R	0	0	0	0	0	0	0	0	0	OBRH6	OBRH5	OBRH4	OBRH3	OBRH2	OBRH1	OBRH0
0x0714	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OBSYS	0	0	0	0	0	REQ	VIEW	0	OBRS6	OBRS5	OBRS4	OBRS3	OBRS2	OBRS1	OBRS0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit name	Function
OBRS[6:0]	Output Buffer Request Shadow Number of source message buffer to be transferred from the Message RAM to OBF Shadow. Valid values are 0x0000 to 0x007F (0…127). If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index Register (GIDX) to OBF Shadow.
VIEW	View Shadow Buffer Toggles between OBF Shadow and OBF Host. Only writeable while OBSYS = ’0’. 1 =Swap OBF Shadow and OBF 0 =No action

Bit name	Function
REQ	Request Message RAM Transfer Requests transfer of message buffer addressed by **OBRS[6:0]** from Message RAM to OBF Shadow. Only writeable while **OBSYS** = '0'. 1 =Transfer to OBF Shadow requested 0 =No request
OBSYS	Output Buffer Shadow Busy Set to '1' after setting bit **REQ**. When the transfer between the Message RAM and OBF Shadow has completed, **OBSYS** is set back to '0'. 1 =Transfer between Message RAM and OBF Shadow in progress 0 =No transfer in progress
OBRH[6:0]	Output Buffer Request Host Number of message buffer currently accessible by the Host via **FRRDHS[1...3]**, MBS, and **FRRDDS[1...64]**. By writing **VIEW** to 1 OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host. Valid values are 0x00 to 0x7F (0...127).

24.5 Functional Description

This chapter describes the E-Ray implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification v2.1.

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

24.5.1 Communication cycle

A communication cycle in FlexRay consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

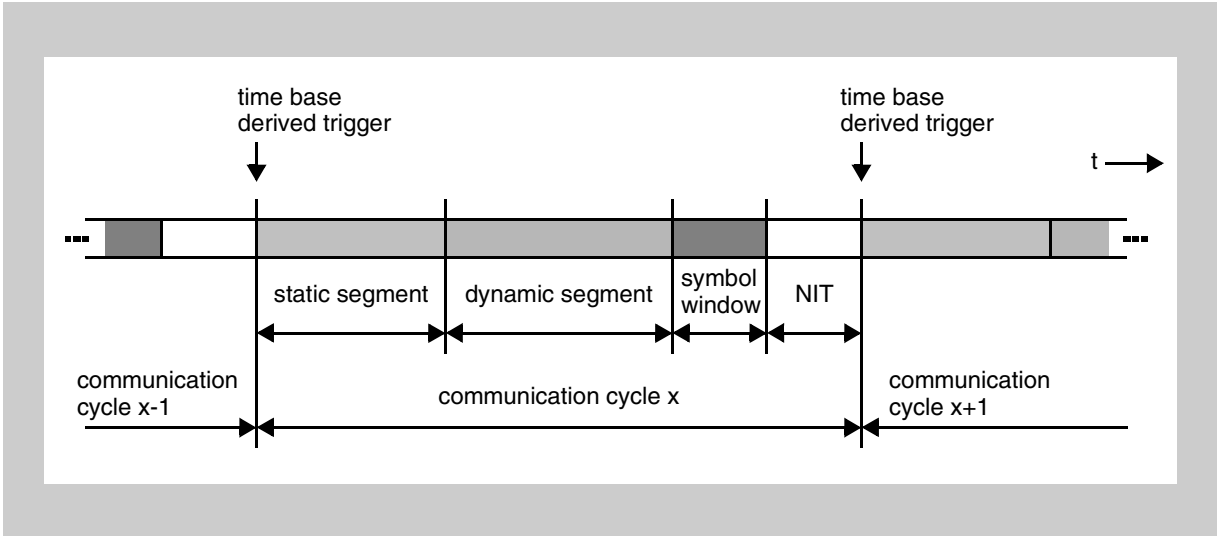


Figure 24-7 Structure of communication cycle

(1) Static Segment

The Static Segment is characterized by the following features:

• Time slots of fixed length (optionally protected by bus guardian)

• Start of frame transmission at action point of the respective static slot

• Payload length same for all frames on both channels

Parameters: Number of Static Slots **FRGTUC7.NSS[9:0]**, Static Slot Length **FRGTUC7.SSL[9:0]**, Payload Length Static **FRMHDC.SFDL[6:0]**, Action Point Offset **FRGTUC9.APO[5:0]**

(2) Dynamic segment

The Dynamic Segment is characterized by the following features:

• All controllers have bus access (no bus guardian protection possible)

• Variable payload length and duration of slots, different for both channels

• Start of transmission at minislot action point

Parameters: Number of Minislots **FRGTUC8.NMS[12:0]**, Minislot Length **FRGTUC8.MSL[5:0]**, Minislot Action Point Offset **FRGTUC9.MAPO[4:0]**, Start of Latest Transmit (last minislot) **FRMHDC.SLT[12:0]**

(3) Symbol window

During the symbol window only **one** media access test symbol (MTS) may be transmitted per channel. MTS symbols are send in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

• Send single symbol

• Transmission of the MTS symbol starts at the symbol windows action point

Parameters: Symbol Window Action Point Offset **FRGTUC9.APO[4:0]** (same as for static slots), Network Idle Time Start **FRGTUC4.NIT[13:0]**

(4) Network idle time (NIT)

During network idle time the CC has to perform the following tasks:

• Calculate clock correction terms (offset and rate)

• Distribute offset correction over multiple macroticks after offset correction start

• Perform cluster cycle related tasks

Parameters: Network Idle Time Start **FRGTUC4.NIT[13:0]**, Offset Correction Start **FRGTUC4.OCS[13:0]**

(5) Configuration of NIT Start and Offset Correction Start

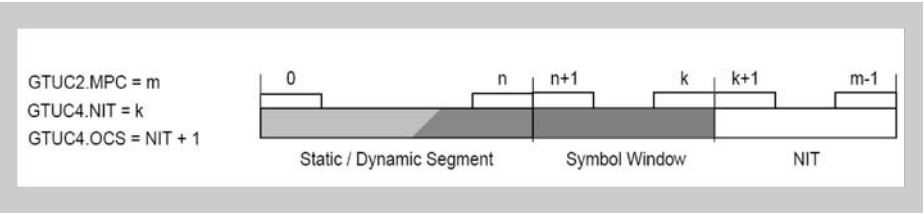


Figure 24-8 Configuration of NIT start and offset correction start

The number of macroticks per cycle gMacroPerCycle is assumed to be m. It is configured by programming **FRGTUC2.MPC** = m.

The static / dynamic segment starts with macrotick 0 and ends with macrotick n:

n = static segment length + dynamic segment offset
+ dynamic segment length - 1MT

n = gNumberOfStaticSlots • gdStaticSlot + dynamic segment offset
+ gNumberOfMinislots • gdMinislot - 1 MT

The static segment length is configured by **FRGTUC7.SSL** and **FRGTUC7.NSS**.
The dynamic segment length is configured by **FRGTUC8.MSL** and **FRGTUC8.NMS**.

The dynamic segment offset is:
If gdActionPointOffset gdMinislotActionPointOffset:
    dynamic segment offset = 0 MT

Else if gdActionPointOffset > gdMinislotActionPointOffset:
    dynamic segment offset = gdActionPointOffset
                - gdMinislotActionPointOffset

The NIT starts with macrotick k+1 and ends with the last macrotick of cycle m-1. It has to be configured by setting **FRGTUC4.NIT** = k.

For the E-Ray the offset correction start is required to be **FRGTUC4.OCS >= FRGTUC4.NIT** + 1 = k+1.

The length of symbol window results from the number of macroticks between the end of the static / dynamic segment and the beginning of the NIT. It can be calculated by k - n.

24.5.2 Communication modes

The FlexRay Protocol Specification v2.1 defines the Time-Triggered Distributed (TT-D) mode.

(1) Time-triggered distributed (TT-D)

In TT-D mode the following configurations are possible:

- **Pure static**: minimum 2 static slots + symbol window (optional)
- **Mixed static/dynamic**: minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes need to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

24.5.3 Clock synchronization

In TT-D mode a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

(1) Global time

Activities in a FlexRay node, including communication, are based on the concept of a global time, even though each individual node maintains its own view of it. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotick (MT) = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (μT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

(2) Local time

Internally, nodes time their behaviour with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (μT).

Node specific:

- Oscillator clock -> prescaler -> microtick (μT)
- μT = basic unit of time measurement in a CC, clock correction is done in units of μTs
- Cycle counter + macrotick counter = nodes local view of the global time

(3) Synchronization process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (**GTUC2.SNM[3:0]**) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment, valid on both channels (two-channel cluster), is measured. The calculation of correction terms is done during NIT (offset: every cycle, rate: odd cycle) by using a FTA / FTM algorithm.

Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of **every** communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of Ts
- Correction done in **odd** numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened / shortened)

Rate (frequency) Correction

- Pairs of deviation values measured and stored in even / odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of **odd** numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of Ts
- Distributed over macroticks comprising the next **even / odd** cycle pair (MTs lengthened / shortened)

Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit **FRMRC.SPLM** has to be programmed to ’1’.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames **FRSUCC1.TXSY** must be set to ’1’.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset / rate correction value is a signed integer
- External offset / rate correction value is added to calculated offset / rate correction value
- Aggregated offset / rate correction term (external + internal) is **not** checked against configured limits

24.5.4 Error handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set **FREIR.PEMC** and may trigger an interrupt to the Host if enabled. The actual error mode is signalled by **FRCCEV.ERRM[1:0]**.

Table 24-9 Error modes of the POC (degradation model)

Error Mode	Activity
ACTIVE (green)	**Full operation**, State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers **FREIR** and **FRSIR**.
PASSIVE (yellow)	**Reduced operation**, State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers **FREIR** and **FRSIR**.
COMM_HALT (red)	**Operation halted**, State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers EIR and SIR. The bus drivers are disabled.

(1) Clock correction failed counter

When the Clock Correction Failed Counter reaches the "maximum without clock correction passive" limit defined by **FRSUCC3.WCP[3:0]**, the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the "maximum without clock correction fatal" limit defined by **FRSUCC3.WCF[3:0]**, it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter **FRCCEV.CCFC[3:0]** allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any **odd** communication cycle during which either the missing offset correction **FRSFS.MOCS** or the missing rate correction **FRSFS.MRCS** flag is set.

The Clock Correction Failed Counter is reset to zero at the end of an **odd** communication cycle if neither the missing offset correction **FRSFS.MOCS** nor the missing rate correction **FRSFS.MRCS** flag is set.

The Clock Correction Failed Counter stops incrementing when the "maximum without clock correction fatal" value **FRSUCC3.WCF[3:0]** is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

Note The transition to HALT state is prevented if **FRSUCC1.HCSE** is not set.

(2) Passive to active counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. **FRSUCC1.PTA[4:0]** defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If **FRSUCC1.PTA[4:0]** is set to zero the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

(3) HALT command

In case the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing **FRSUCC1.CMD[3:0]** = "0110". In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from **FRCCSV.PSL[5:0]**.

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state **FRSUCC1.CMD[3:0]** will be reset to "0000" = command_not_accepted and bit **FREIR.CNA** is set to ’1’. If enabled an interrupt to the Host is generated.

(4) FREEZE command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing **FRSUCC1.CMD[3:0]** = "0111". The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from **FRCCSV.PSL[5:0]**.

24.5.5 Communication controller states

(1) Communication controller state diagram

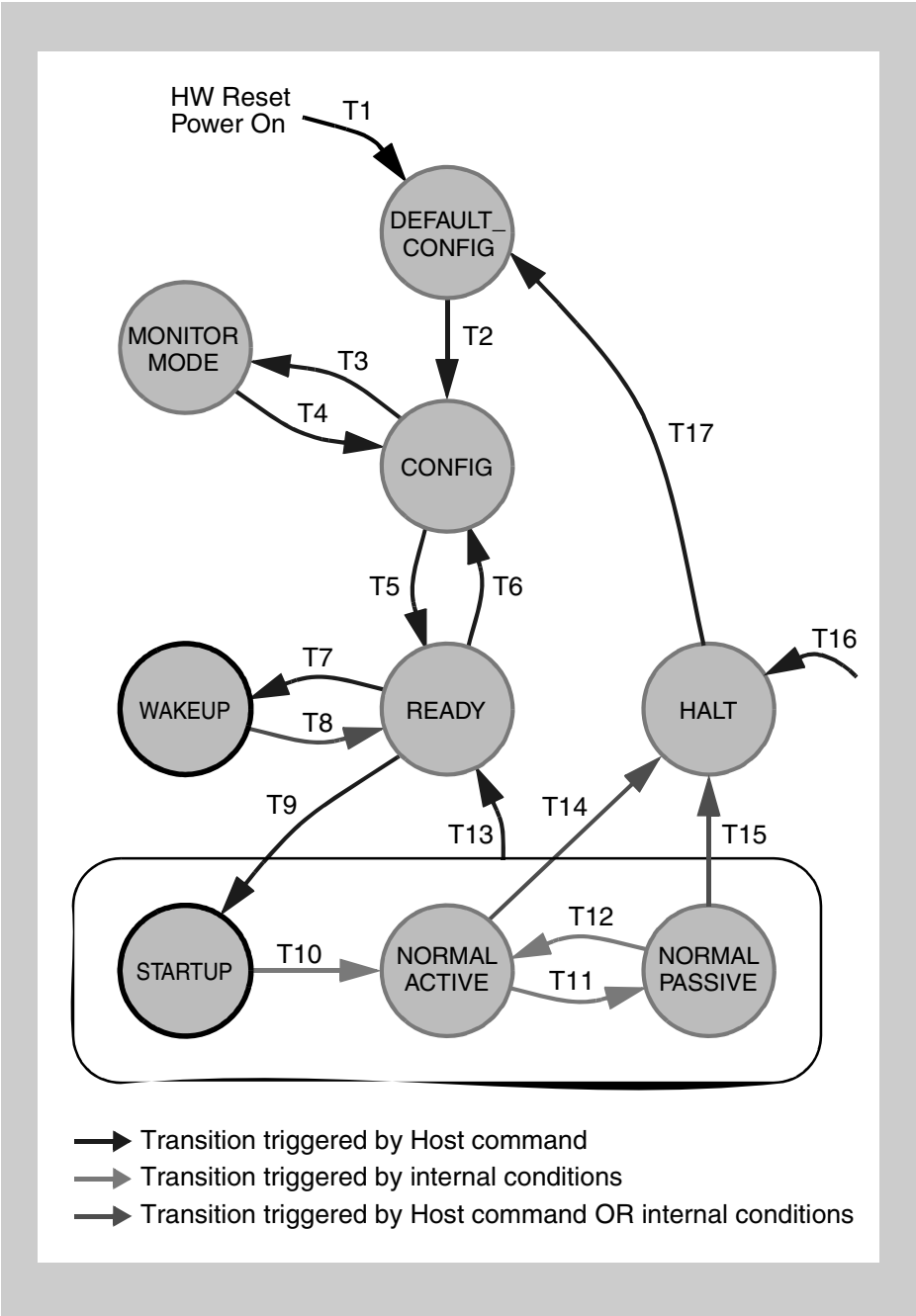


Figure 24-9 Overall state diagram of E-Ray communication controller

State transitions are controlled by externals pins **eray_reset** and **eray_rxd1,2**, by the POC state machine, and by the CHI Command Vector **FRSUCC1.CMD[3:0]**.

The CC exits from **all** states to HALT state after application of the FREEZE command (**FRSUCC1.CMD[3:0]** = "0111").

Table 24-10 State transitions of E-Ray overall state machine

T#	Condition	From	To
1	Hard reset	All States	DEFAULT_CONFIG
2	Command CONFIG, **FRSUCC1.CMD[3:0]** = "0001"	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command MONITOR_MODE, **FRSUCC1.CMD[3:0]** = "1011"	CONFIG	MONITOR_MODE
4	Command CONFIG, **FRSUCC1.CMD[3:0]** = "0001"	MONITOR_MODE	CONFIG
5	Unlock sequence followed by command READY, **FRSUCC1.CMD[3:0]** = "0010"	CONFIG	READY
6	Command CONFIG, **FRSUCC1.CMD[3:0]** = "0001"	READY	CONFIG
7	Command WAKEUP, **FRSUCC1.CMD[3:0]** = "0011"	READY	WAKEUP
8	Complete, non-aborted transmission of wakeup pattern OR received WUP OR received frame header OR command READY, **FRSUCC1.CMD[3:0]** = "0010"	WAKEUP	READY
9	Command RUN, **FRSUCC1.CMD[3:0]** = "0100"	READY	STARTUP
10	Successful startup	STARTUP	NORMAL_ACTIVE
11	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by **FRSUCC3.WCP[3:0]**	NORMAL_ACTIVE	NORMAL_PASSIVE
12	Number of valid correction terms reached the Passive to Active limit configured by **FRSUCC1.PTA[4:0]**	NORMAL_PASSIVE	NORMAL_ACTIVE
13	Command READY, **FRSUCC1.CMD[3:0]** = "0010"	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
14	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by **FRSUCC3.WCF[3:0]** AND bit **FRSUCC1.HCSE** set to '1' OR command HALT, **FRSUCC1.CMD[3:0]** = "0110"	NORMAL_ACTIVE	HALT
15	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by **FRSUCC3.WCF[3:0]** AND bit **FRSUCC1.HCSE** set to '1' OR command HALT, **FRSUCC1.CMD[3:0]** = "0110"	NORMAL_PASSIVE	HALT
16	Command FREEZE, **FRSUCC1.CMD[3:0]** = "0111"	All States	HALT
17	Command CONFIG, **FRSUCC1.CMD[3:0]** = "0001"	HALT	DEFAULT_CONFIG

(2) DEFAULT_CONFIG state

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When leaving hardware reset (external reset signal **eray_reset** is disasserted)
- When exiting from HALT state

To leave DEFAULT_CONFIG state the Host has to write **CMD[3:0]** = "0001" in the SUC Configuration Register 1. The CC then transits to CONFIG state.

(3) CONFIG state

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from MONITOR_MODE or READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyse status information and configuration. Before leaving CONFIG state the Host has to assure that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described in chapter Lock Register (**FRLCK**). Directly after unlocking the CONFIG state the Host has to write **FRSUCC1.CMD[3:0]** to enter the next state.

Note IStatus bits MHDS[14:0], registers **FRTXRQ1/2/3/4**, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (**eray_sclk**, **eray_bclk**). To do this the Host has to assure that all Message RAM transfers have finished before turning off the clocks.

(4) MONITOR_MODE

After unlocking CONFIG state and writing **FRSUCC1.CMD[3:0]** = "1011" the CC enters MONITOR_MODE. In this mode the CC is able to receive FlexRay frames and to detect wakeup pattern. The temporal integrity of received frames is not checked, and therefore cycle counter filtering is not supported. This mode can be used for debugging purposes in case e.g. that startup of a FlexRay network fails. After writing **FRSUCC1.CMD[3:0]** = "0001" the CC transits back to CONFIG state.

In MONITOR_MODE the pick first valid mechanism is disabled. This means that a receive message buffer may only be configured to receive on one channel. Received frames are stored into message buffers according to frame ID and receive channel. Null frames are handled like data frames. After frame reception only status bits **FRMBS.VFRA**, **FRMBS.VFRB**, **FRMBS.MLST**, **FRMBS.RCIS**, **FRMBS.SFIS**, **FRMBS.SYNS**, **FRMBS.NFIS**, **FRMBS.PPIS**, **FRMBS.RESS** have valid values. In MONITOR_MODE the receive FIFO is not available.

(5) READY state

After unlocking CONFIG state and writing **FRSUCC1.CMD[3:0]** = "0010" the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

• When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing **FRSUCC1.CMD[3:0]** = "0010" (READY command).

The CC exits from this state

• To CONFIG state by writing **FRSUCC1.CMD[3:0]** = "0001" (CONFIG command)

• To WAKEUP state by writing **FRSUCC1.CMD[3:0]** = "0011" (WAKEUP command)

• To STARTUP state by writing **FRSUCC1.CMD[3:0]** = "0100" (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

Note Status bits **FRMHDS[14:0]**, registers **FRTXRQ1/2/3/4**, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

(6) WAKEUP state

The description below is intended to help configuring wakeup for the E-Ray IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The CC enters this state

• When exiting from READY state by writing **FRSUCC1.CMD[3:0]** = "0011" (WAKEUP command).

The CC exits from this state to READY state

• After complete non-aborted transmission of wakeup pattern

• After WUP reception

• After detecting a WUP collision

• After reception of a frame header

• By writing **FRSUCC1.CMD[3:0]** = "0010" (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an **external** wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing **FRSUCC1.WUCS**. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag **FRSIR.WST**. The wakeup status vector can be read from **FRCCSV.WSV[2:0]**. If a valid wakeup pattern was received also either flag **FRSIR.WUPA** or flag **FRSIR.WUPB** is set.

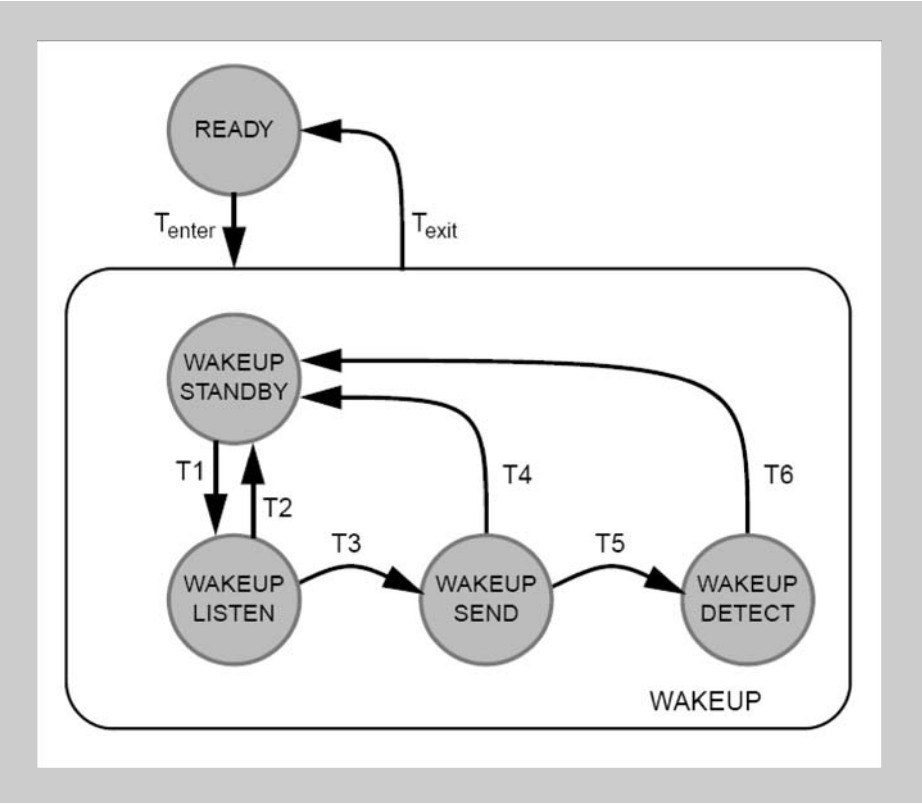


Figure 24-10 Structure of POC state WAKEUP

Table 24-11 State transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to POC WAKEUP state by writing **FRSUCC1.CMD[3:0]** = "0011" (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit **WUCS** in the SUC Configuration Register 1 OR frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired OR WUP detected on wakeup channel selected by bit **FRSUCC1.WUCS** OR frame header received on either available channel	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) OR Host commands change to READY state by writing **FRSUCC1.CMD[3:0]** = "0010" (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters listen timeout **FRSUCC2.LT[20:0]** and listen timeout noise **FRSUCC2.LTN[3:0]**. Listen timeout enables a fast cluster wakeup in case of a noise free environment,

while listen timeout noise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by **FRSUCC2.LT[20:0]**. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification v2.1 recommends that two different CCs shall awake the two channels.

Host activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signalled to their local Host.

Wakeup procedure controlled by Host (single-channel wakeup):

• Configure the CC in CONFIG state
- Select wakeup channel by programming bit **FRSUCC1.WUCS**

• Check local BDs whether a WUP was received

• Activate BD of selected wakeup channel

• Command CC to enter READY state

• Command CC to start wakeup on the configured channel by writing **FRSUCC1.CMD[3:0]** = "0011"
- CC enters WAKEUP
- CC returns to READY state and signals status of wakeup attempt to the Host

• Wait predefined time to allow the other nodes to wakeup and configure themselves

• Coldstart node:
- In a dual channel cluster wait for WUP on the other channel
- Reset coldstart inhibit flag **FRCCSV.CSI** by writing **FRSUCC1.CMD[3:0]** = "1001" (ALLOW_COLDSTART command)

• Command CC to enter startup by writing **FRSUCC1.CMD[3:0]** = "0100" (RUN command)

Wakeup procedure triggered by BD:

• Wakeup recognized by BD

• BD triggers power-up of Host (if required)

• BD signals wakeup event to Host

• Host configures its local CC

• If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves

• Host commands CC to enter STARTUP state by writing **FRSUCC1.CMD[3:0]** = "0100" (RUN command)

Wakeup pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers **FRPRTC1** and **FRPRTC2**.

• Single channel wakeup, wakeup symbol may not be sent on both channels at the same time

• Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)

• Wakeup symbol must be configured identical in all nodes of a cluster

• Wakeup symbol transmit low time configured by **FRPRTC2.TXL[5:0]**

• Wakeup symbol idle time used to listen for activity on the bus, configured by **FRPRTC2.TXI[7:0]**

• A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup

• Number of repetitions configurable by **FRPRTC1.RWPFR[5:0]** (2 to 63 repetitions)

• Wakeup symbol receive window length configured by **PRTC1.RXW[8:0]**

• Wakeup symbol receive low time configured by **FRPRTC2.RXL[5:0]**

• Wakeup symbol receive idle time configured by **FRPRTC2.RXI[5:0]**

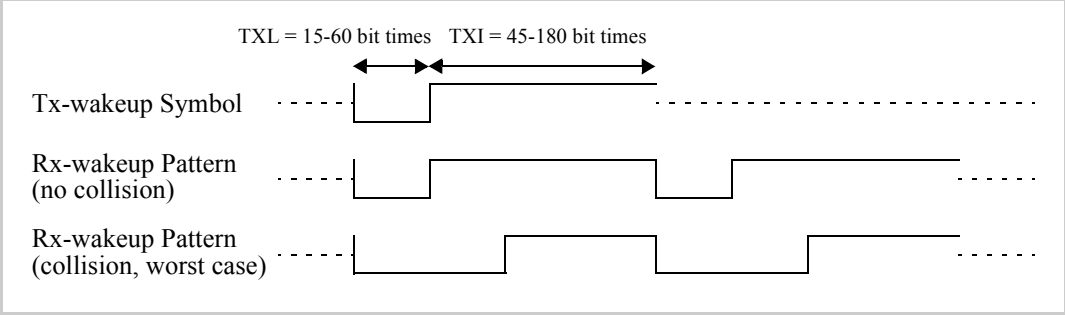


Figure 24-11 Timing of wakeup pattern

(7) STARTUP state

The description below is intended to help configuring startup for the E-Ray IP-module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.2.

Any node entering STARTUP state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter NORMAL_ACTIVE state via:

• Coldstart path initiating the schedule synchronization (leading coldstart node)

• Coldstart path joining other coldstart nodes (following coldstart node)

• Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits **FRSUCC1.TXST** and **FRSUCC1.TXSY** set to ’1’. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by **FRSUCC1.CSA[4:0]**.

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes.

The node can only leave the integration phase and actively participate in communication when these checks are passed.

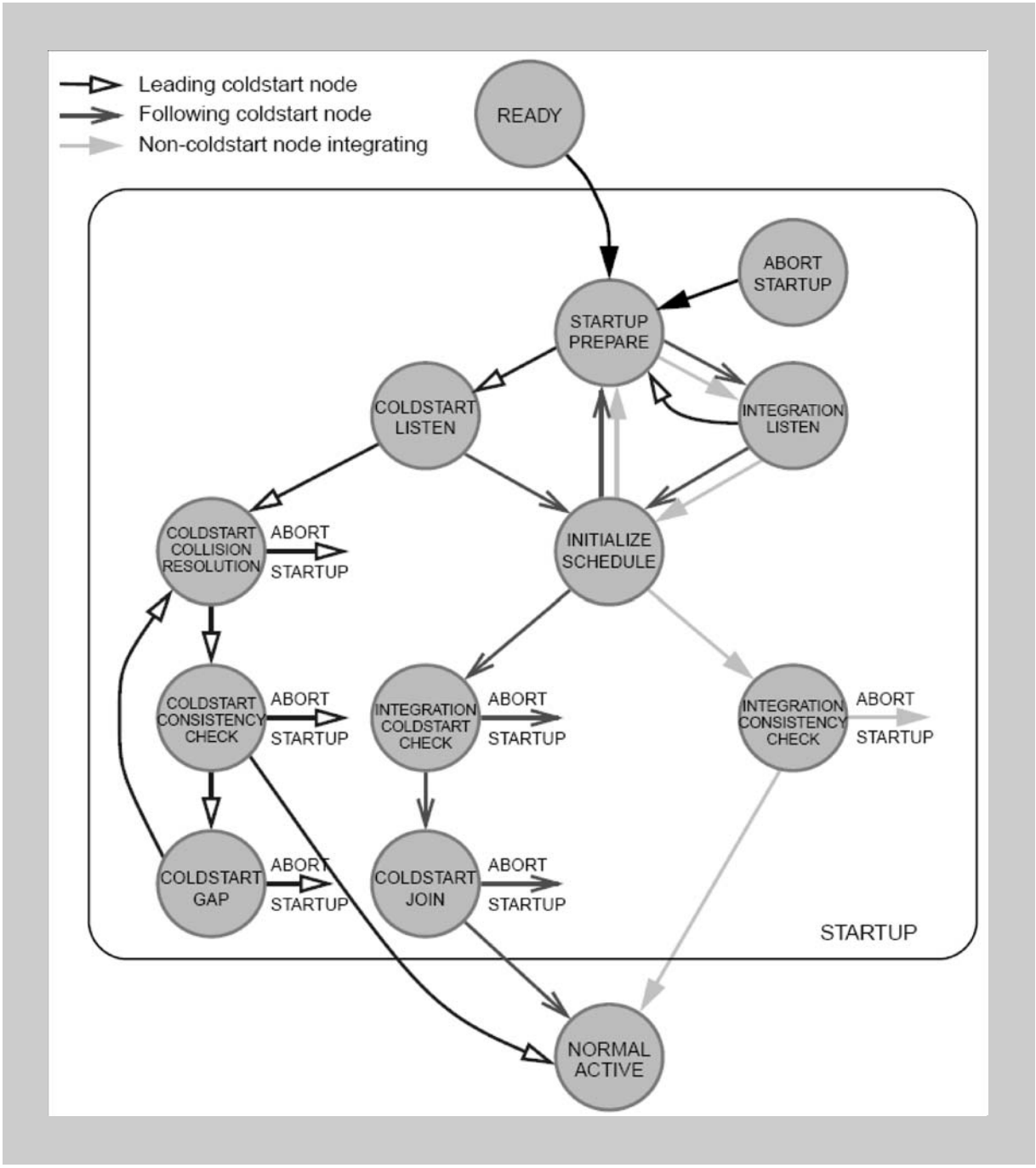


Figure 24-12 State diagram time-triggered startup

Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit **FRCCSV.CSI** is set, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit **FRCCSV.CSI** is set whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (**FRSUCC1.CMD[3:0]** = "1001")

Startup Timeouts The CC supplies two different μT timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are reset when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

Note The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values **FRSUCC2.LT[20:0]** and **FRSUCC2.LTN[3:0]**.

Startup timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming **FRSUCC2.LT[20:0]** (see chapter SUC Configuration Register 2 (**FRSUCC2**)).

The startup timeout is: pdListenTimeout = **FRSUCC2.LT[20:0]**

The startup timer is restarted upon:
• Entering the COLDSTART_LISTEN state
• Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:
• If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
• When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

Startup noise timeout

At the same time the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming **FRSUCC2.LTN[3:0]** (see chapter SUC Configuration Register 2 (**FRSUCC2**)).

The startup noise timeout is:
pdListenTimeout • gListenNoise **= SUCC2.LT[20:0] •** (**SUCC2.LTN[3:0]** + 1)

The startup noise timer is restarted upon:
• Entering the COLDSTART_LISTEN state
• Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

Path of leading Coldstart Node (initiating coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by **FRSUCC1.CSA[4:0]**. The number of remaining coldstarts attempts can be read from **FRCCSV.RCA[4:0]**. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is larger than one and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

Path of following Coldstart Node (responding to leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state integrating coldstart nodes begin to transmit their own startup frames. Thereby the node that initiated the coldstart and the nodes joining it can check if their schedules agree to each other. If for the following three cycles the clock correction does not signal errors and at least one other coldstart node is visible, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

Path of Non-coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signalled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

(8) NORMAL_ACTIVE state

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even / odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions
- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to
- HALT state by writing **FRSUCC1.CMD[3:0]** = "0110" (HALT command, at the end of the current cycle)
- HALT state by writing **FRSUCC1.CMD[3:0]** = "0111" (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing **FRSUCC1.CMD[3:0]** = "0010" (READY command)

(9) NORMAL_PASSIVE state

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state
- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to
- HALT state by writing **FRSUCC1.CMD[3:0]** = "0110" (HALT command, at the end of the current cycle)
- HALT state by writing **FRSUCC1.CMD[3:0]** = "0111" (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when **FRCCEV.PTAC[4:0]** equals **FRSUCC1.PTA[4:0] - 1**
- To READY state by writing **FRSUCC1.CMD[3:0]** = "0010" (READY command)

(10) HALT state

In this state all communication (reception and transmission) is stopped.

The CC enters this state
- By writing **FRSUCC1.CMD[3:0]** = "0110" (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing **FRSUCC1.CMD[3:0]** = "0111" (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and **FRSUCC1.HCSE** is set
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and **FRSUCC1.HCSE** is set

The CC exits from this state to DEFAULT_CONFIG state
- By writing **FRSUCC1.CMD[3:0]** = "0001" (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analysing purposes.

When the Host writes **FRSUCC1.CMD[3:0]** = "0110" (HALT command), the CC sets bit **FRCCSV.HRQ** and enters HALT state at the next end of cycle.

When the Host writes **FRSUCC1.CMD[3:0]** = "0111" (FREEZE command), the CC enters HALT state immediately and sets bit **FRCCSV.FSI**.

The POC state from which the transition to HALT state took place can be read from **FRCCSV.PSL[5:0]**.

24.5.6 Network management

The accrued Network Management (NM) vector can be read from registers NMV13. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (**PPI**) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by **FRNEMC.NML[3:0]**. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the **PPI** bit set, bit **PPIT** in the header section of the respective transmit buffer has to be set via **FRWRHS1.PPIT**. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

Note In case a message buffer is configured for transmission / reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by **FRNEMC.NML[3:0]**.
When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case NMV1…3 holds the value from the cycle before.

24.5.7 Filtering and masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated / transmitted if the required matches occur.

Filtering is done on the following fields:

- Channel ID
- Frame ID
- Cycle Counter

The following filter combinations for acceptance / transmit filtering are allowed:

- Frame ID + Channel ID
- Frame ID + Channel ID + Cycle Counter

All configured filters must match in order to store a received message in a message buffer.

Note For the FIFO the acceptance filter is configured by the FIFO Rejection Filter and the FIFO Rejection Filter Mask.

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

(1) Slot counter filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

(2) Cycle counter filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 resp. 1 is configured to hold the startup / sync frame or the single slot frame by bits **FRSUCC1.TXST**, **FRSUCC1.TXSY**, and **FRSUCC1.TSM**, cycle counter filtering for message buffer 0 resp. 1 must be disabled.

Note Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is **not** allowed.

The set of cycle numbers belonging to a cycle set is determined as described in *Table 24-12*.

Table 24-12 Definition of cycle set

Cycle Code	Matching Cycle Counter Values		
0b000000x	all Cycles		
0b000001c	every second Cycle	at (Cycle Count)mod2	= c
0b00001cc	every fourth Cycle	at (Cycle Count)mod4	= cc
0b0001ccc	every eighth Cycle	at (Cycle Count)mod8	= ccc
0b001cccc	every sixteenth Cycle	at (Cycle Count)mod16	= cccc
0b01ccccc	every thirty-second Cycle	at (Cycle Count)mod32	= ccccc
0b1cccccc	every sixty-fourth Cycle	at (Cycle Count)mod64	= cccccc

Table 24-13 below gives some examples for valid cycle sets to be used for cycle counter filtering:

Table 24-13 Examples for valid cycle sets

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7- .... -63 ↵
0b0000100	0-4-8-12- .... -60 ↵
0b0001110	6-14-22-30- .... -62 ↵
0b0011000	8-24-40-56 ↵
0b0100011	3-35 ↵
0b1001001	9 ↵

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Other filter criteria must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Other filter criteria must also be met.

(3) Channel ID filtering

There is a 2-bit channel filtering field (**CHA**, **CHB**) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see *Table 24-14*).

Table 24-14 Channel filtering configuration

CHA	CHB	Transmit Buffer transmit frame	Receive Buffer store valid receive frame
1	1	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)
1	0	on channel A	received on channel A
0	1	on channel B	received on channel B
0	0	no transmission	ignore frame

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (**CHA** and **CHB** set).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (**CHA** and **CHB** set).

Note If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to ’1’, no frames are transmitted resp. received frames are ignored (same function as **CHA = CHB** = ’0’).

(4) FIFO filtering

For FIFO filtering there is one rejection filter and one rejection filter mask available. The FIFO filter consists of channel filter **FRFRF.CH[1:0]**, frame ID filter **FRFRF.FID[10:0]**, and cycle counter filter **FRFRF.CYF[6:0]**. Registers FRF and FRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles **not** belonging to the cycle set specified by **FRFRF.CYF[6:0]**, **all** frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

24.5.8 Transmit process

(1) Static segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

(2) Dynamic segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by **FRMHDC.SLT[12:0]** defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

(3) Transmit buffers

E-Ray message buffers can be configured as transmit buffers by programming bit **CFG** in the header section of the respective message buffer to ’1’ via **FRWRHS1**.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment: channel A **or** channel B, channel A **and** channel B
- Dynamic segment: channel A **or** channel B

Message buffer 0 resp. 1 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by **FRSUCC1.TXST**, **FRSUCC1.TXSY**, and **FRSUCC1.TSM**. In this case, it can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup / sync frame per communication cycle. Transmission of startup / sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of **FRMRC.SEC[1:0]** (see 5.11.1 Reconfiguration of Message Buffers). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the **FRPPIT** bit in the header section of the respective message buffer to '1' and write the network management information to the data section of the message buffer.

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by **FRMHDC.SFDL[6:0]**, the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is logical zero.

Note In case of an odd payload length (PLC = 1,3,5,...) the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is all zero.

Each transmit buffer provides a transmission mode flag **TXM** that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In **single-shot mode** the CC resets the respective **TXR** flag after transmission has completed. Now the Host may update the transmit buffer.

In **continuous mode**, the CC does not reset the respective transmission request flag **TXR** after successful transmission. In this case a frame is sent out each time the filter criteria match. The **TXR** flag can be reset by the Host by writing the respective message buffer number to the FRIBCR register while bit **FRIBCM.STXRH** is set to '0'.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

(4) Frame transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via WRHS1, WRHS2, and WRHS3
- Write the data section of the transmit buffer via WRDSn
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register FRIBCR
- If configured in register FRIBCM, the transmission request flag **TXR** for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective **TXR** bit (**TXR** = '0') in the TRXQ1/2/3/4 registers (single-shot mode only).

After transmission has completed, the respective **TXR** flag in the TXRQ1/2/3/4 register is reset (single-shot mode), and, if bit **MBI** in the header section of the message buffer is set, flag **FRSIR.TXI** is set to '1'. If enabled, an interrupt is generated.

(5) Null frame transmission

If in static segment the Host does not set the transmission request flag before transmit time, and if there is no other transmit buffer with matching filter criteria, the CC transmits a null frame with the null frame indication bit **set to '0'** and the payload data **set to zero**.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (**TXR** = '0').
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status MBS is updated.

Null frames are not transmitted in the dynamic segment.

24.5.9 Receive process

(1) Dedicated Receive buffers

A portion of the E-Ray message buffers can be configured as dedicated receive buffers by programming bit **CFG** in the header section of the respective message buffer to '0' via **FRWRHS1**.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A **or** channel B, channel A **and** channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A **or** channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of **FRMRC.SEC[1:0]** Buffers). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

(2) Frame reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via **FRWRHS1**, **FRWRHS2**, and **FRWRHS3**
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register **FRIBCR**

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective **ND** flag in the **FRNDAT1/2/3/4** registers is set, and, if bit **MBI** in the header section of that message buffer is set, flag **FRSIR.RXI** is set to '1'. If enabled, an interrupt is generated.

In case that bit **ND** was already set when the Message Handler updates the message buffer, bit **FRMBS.MLST** of the respective message buffer is set and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status MBS is updated.

When the Message Handler changed the message buffer status MBS in the header section of a message buffer, the respective **MBC** flag in the MBSC1/2/3/4 registers is set, and if bit **MBI** in the header section of that message buffer is set, flag **SIR.MBSI** is set to '1'. If enabled an interrupt is generated.

If the payload length of a received frame **PLR[6:0]** is longer than the value programmed by **PLC[6:0]** in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in chapter Data Transfer from Message RAM to Output Buffer.

Note The **ND** and **MBC** flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

(3) Null frame reception

The payload segment of a received null frame is **not** copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status MBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the message buffer status MBS in the header section of a message buffer, the respective **MBC** flag in the **FRMBSC1/2/3/4** register is set, and if bit **MBI** in the header section of that message buffer is set, flag **FRSIR.MBSI** is set to ’1’. If enabled, an interrupt is generated.

24.5.10 FIFO function

(1) Description

A group of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by **FRMRC.FFB[7:0]** and ending with the message buffer referenced by **FRMRC.LCB[7:0]**. Up to 127 message buffers can be assigned to the FIFO.

Every **valid** incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the message buffer status MBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit **FRSIR.RFNE** shows that the FIFO is not empty, bit **FRSIR.RFCL** is set when the receive FIFO fill level **FRFSR.RFFL[7:0]** is equal or greater than the critical level as configured by **FRFCL.CL[7:0]**, bit **FREIR.RFO** shows that a FIFO overrun has been detected. If enabled, interrupts are generated.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag **FREIR.RFO**.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag **FRSIR.RFNE** is set. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in *Figure 24-13* for a three message buffer FIFO.

The programmable FIFO Rejection Filter (FRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit **FRFRF.RSS** is set to ’1’ (default), all messages received in the static segment are rejected by the FIFO. If bit **FRFRF.RNF** is set to ’1’ (default), received null frames are not stored in the FIFO.

The FIFO Rejection Filter Mask (FRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked ’don’t care’ for rejection filtering.

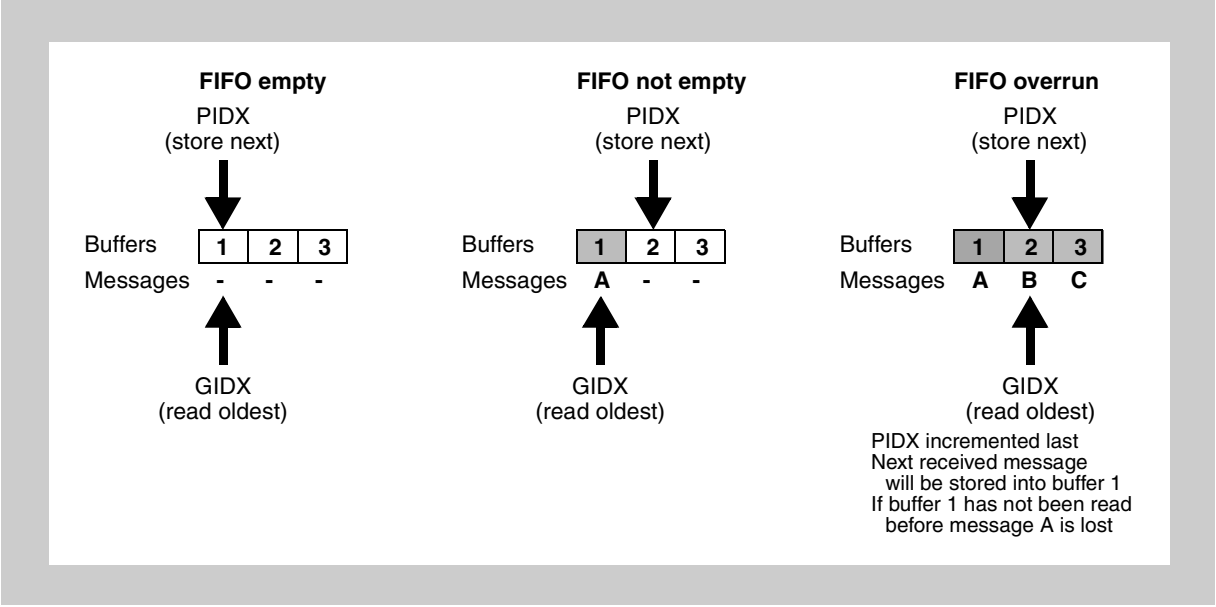


Figure 24-13 FIFO status: empty, not empty, overrun

(2) Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via **FRWRHS2.PLC[6:0]**. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via **FRWRHS3.DP[10:0]**.

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

Note It is recommended to program the MBI bits of the message buffers belonging to the FIFO to '0' via FRWRHS1.MBI to avoid generation of RX interrupts.
If the payload length of a received frame is longer than the value programmed by **FRWRHS2.PLC[6:0]** in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

(3) Access to the FIFO

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by **FRMRC.FFB[7:0]**) to the register FROBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

24.5.11 Message handling

The Message Handler controls data transfers between the Input / Output Buffer and the Message RAM and between the Message RAM and the two Transient Buffer RAMs. All accesses to the internal RAMs except the Message RAM are 32+1 bit accesses. The additional bit is used for parity checking. The accesses to the Message RAM are 32+7 bit accesses. The additional 7bits are the ECC code.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to **FRGTUC7.NSS[9:0]**. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from FRGTUC7.NSS[9:0] + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

(1) Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers WRHS1...3.

Reconfiguration has to be enabled via control bits **FRMRC.SEC[1:0]** in the Message RAM Configuration register.

If a message buffer has not been transmitted / updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission / reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted / updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to Table 16 below:

Start of Scan in Slot	Scan for Slots
1	2...15, 1 (next cycle)
8	6...23, 1 (next cycle)
16	24...31, 1 (next cycle)
24	32...39, 1 (next cycle)
...	...

Table 24-15 Scan of Message RAM

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by **FRMRC.FDB[7:0]**. In case aMessage RAM scan starts while the CC is in

dynamic segment, the scan starts with the message buffer number configured by **FRMRC.FDB[7:0]**.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the "Static Buffers", it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the "Static + Dynamic Buffers", it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

Note Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted / updated from a received frame.

(2) Host access to message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target / source message buffer to be accessed to FRIBCR or FROBCR register.

The FRIBCM and FROBCM registers can be used to write / read header and data section of the selected message buffer separately.

If bit **FRIBCM.STXR** is set to = ’1’, the transmission request flag **TXR** of the selected message buffer is automatically set after the message buffer has been updated. If bit **FRIBCM.STXR** is reset to ’0’, the transmission request flag **TXR** of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are build up as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host / OBF Host), while the other half (IBF Shadow / OBF Shadow) is accessed by the Message Handler for data transfers between IBF / OBF and Message RAM.

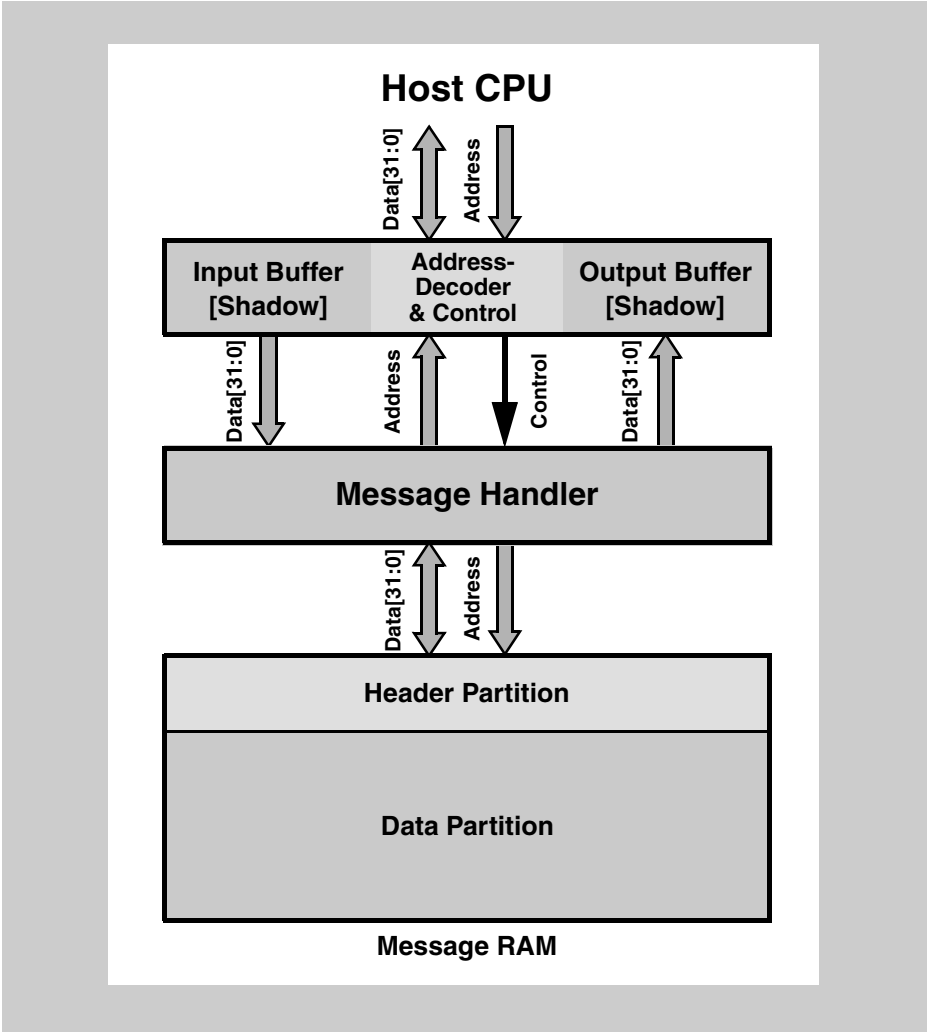


Figure 24-14 Host access to Message RAM

Data Transfer from Input Buffer to Message RAM

To configure / update a message buffer in the Message RAM, the Host has to write the data to FRWRDSn and the header to FRWRHS1…3. The specific action is selected by configuring the Input Buffer Command Mask **FRIBCM**.

When the Host writes the number of the target message buffer in the Message RAM to **FRIBCR.IBRH[6:0]**, IBF Host and IBF Shadow are swapped (see *Figure 24-15*).

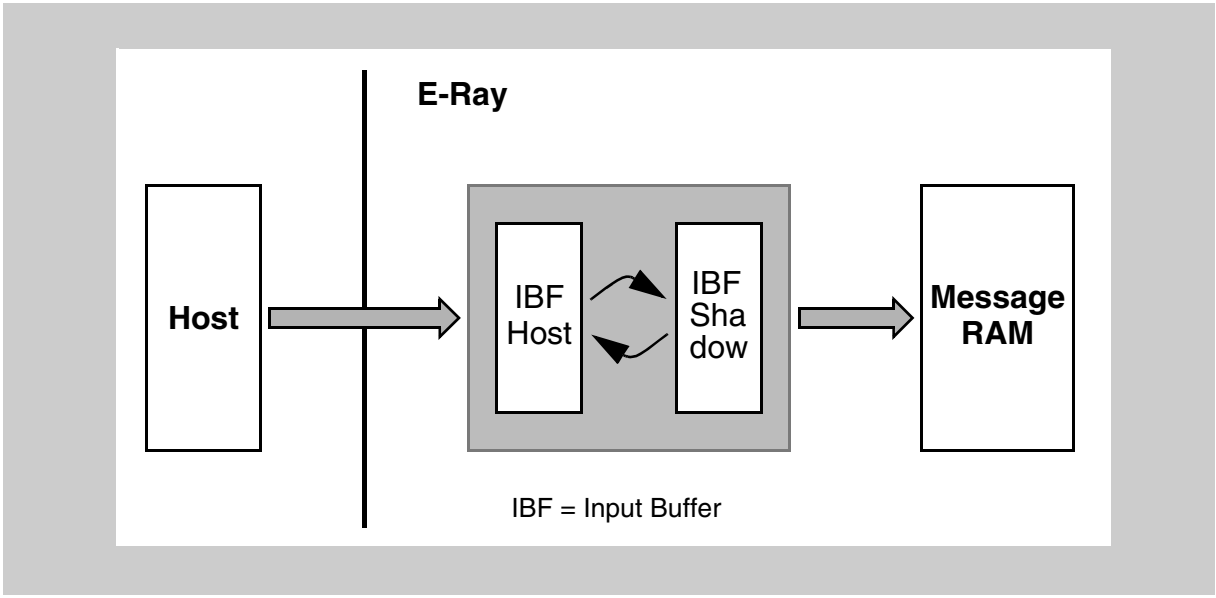


Figure 24-15 Double buffer structure Input Buffer

In addition the bits in the FRIBCM and FRIBCR registers are also swapped to keep them attached to the respective IBF section.

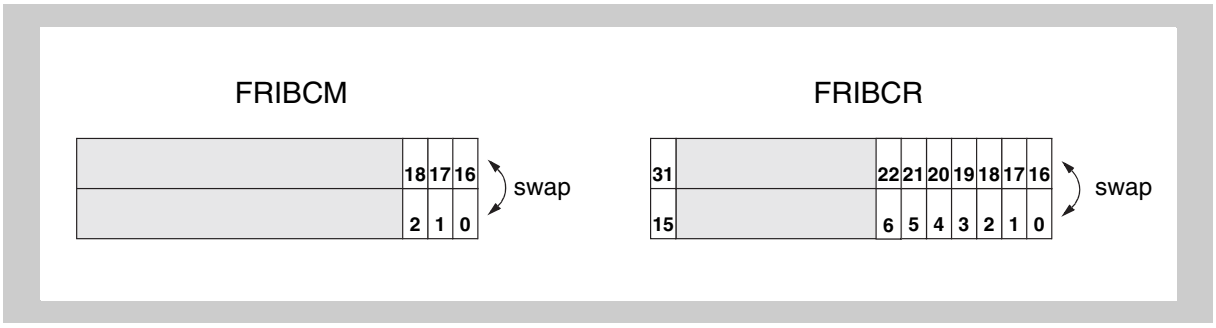


Figure 24-16 Swapping of FRIBCM and FRIBCR bits

With this write operation bit **FRIBCR.IBSYS** is set to ’1’. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the MessageRAMselected by **FRIBCR.IBRS[6:0]**.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit **FRIBCR.IBSYS** is set back to ’0’ and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to **FRIBCR.IBRH[6:0]**.

If a write access to **FRIBCR.IBRH[6:0]** occurs while **FRIBCR.IBSYS** is ’1’, **FRIBCR.IBSYH** is set to ’1’. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, **FRIBCR.IBSYH** is reset to ’0’, **FRIBCR.IBSYS** remains set to ’1’, and the next transfer to the Message RAM is started. In addition the message buffer

numbers stored under **FRIBCR.IBRH[6:0]** and **FRIBCR.IBRS[6:0]** and the command mask flags are also swapped.

Example of a 32-bit Host access sequence:

Configure / update n-th message buffer via IBF

- Wait until **FRIBCR.IBSYH** is reset
- Write data section to WRDSn
- Write header section to WRHS1...3
- Write Command Mask: write **FRIBCM.STXRH**, **FRIBCM.LDSH**, **FRIBCM.LHSH**
- Demand data transfer to target message buffer: write **FRIBCR.IBRH[6:0]**

Configure / update (n+1)th message buffer via IBF

- Wait until **FRIBCR.IBSYH** is reset
- Write data section to WRDSn
- Write header section to WRHS1...3
- Write Command Mask: write **FRIBCM.STXRH**, **FRIBCM.LDSH**, **FRIBCM.LHSH**
- Demand data transfer to target message buffer: write **FRIBCR.IBRH[6:0]**

Note Any write access to IBF while **FRIBCR.IBSYH** is ’1’ will set error flag **FREIR.IIBA** to ’1’. In this case the write access has no effect.

Table 24-16 Assignment of FRIBCM bits

Pos.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow ongoing or finished
17	r	LDSS	Load Data Section Shadow ongoing or finished
16	r	LHSS	Load Header Section Shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 24-17 Assignment of FRIBCR bits

Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22...16	r	IBRS[6:0]	IBF Request Shadow, number of message buffer currently / last updated
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH[6:0]
6...0	r/w	IBRH[6:0]	IBF Request Host, number of message buffer to be updated next

Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register FROBCR to trigger the data transfer as configured in FROBCM. After the transfer has completed, the Host can read the transferred data from FRRDDSn, FRRDHS1...3, and MBS.

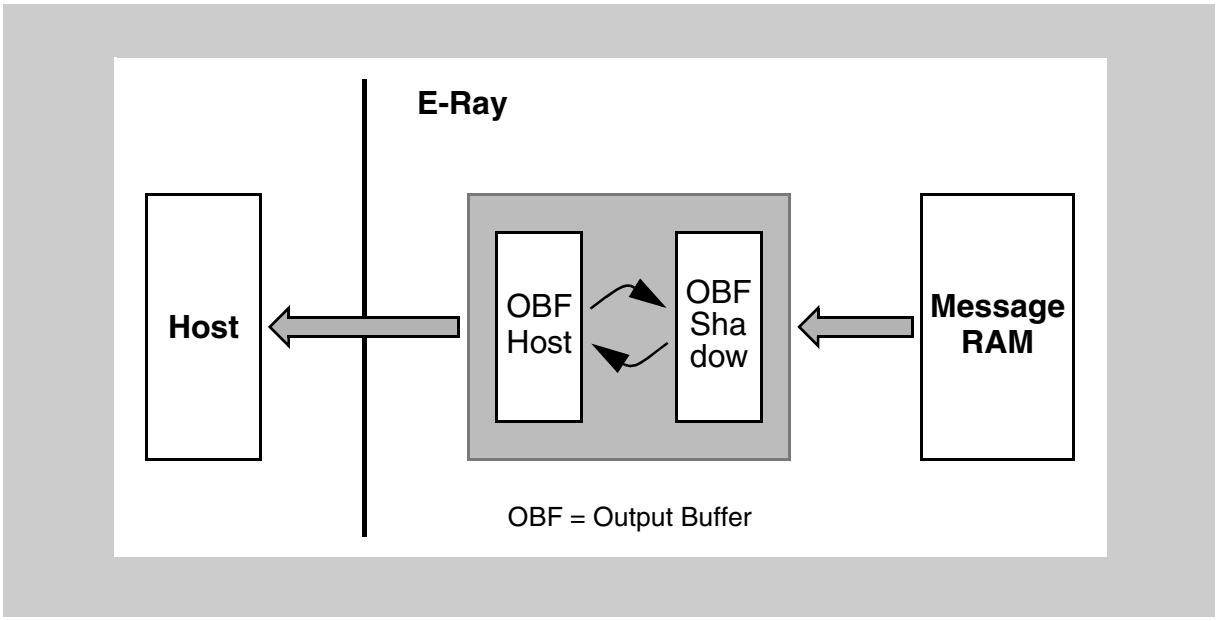


Figure 24-17 Double buffer structure Output Buffer

OBF Host and OBF Shadow as well as bits **FROBCM.RHSS**, **FROBCM.RDSS**, **FROBCM.RHSH**, **FROBCM.RDSH** and bits **FROBCR.OBRS[6:0]**, **FROBCR.OBRH[6:0]** are swapped under control of bits **FROBCR.VIEW** and **FROBCR.REQ**.

Writing bit **FROBCR.REQ** to ’1’ copies bits **FROBCM.RHSS**, **FROBCM.RDSS** and bits **FROBCR.OBRS[6:0]** to an internal storage.

After setting **FROBCR.REQ** to ’1’, **FROBCR.OBSYS** is set to ’1’, and the transfer of the message buffer selected by **FROBCR.OBRS[6:0]** from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, the **FROBCR.OBSYS** bit is set back to ’0’. Bits **FROBCR.REQ** and **FROBCR.VIEW** can only be set to ’1’ while **FROBCR.OBSYS** is ’0’.

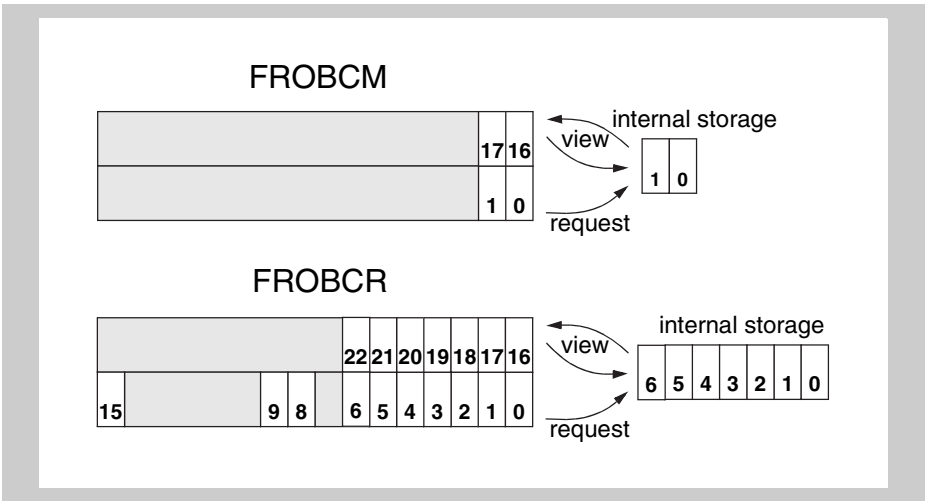


Figure 24-18 Swapping of FROBCM and FROBCR bits

OBF Host and OBF Shadow are swapped by setting bit **FROBCR.VIEW** to '1' while bit **FROBCR.OBSYS** is '0'.

In addition bits **FROBCR.OBRH[6:0]** and bits **FROBCM.RHSH**, **FROBCM.RDSH** are swapped with the registers internal storage thus assuring that the message buffer number stored in **FROBCR.OBRH[6:0]** and the mask configuration stored in **FROBCM.RHSH**, **FROBCM.RDSH** matches the transferred data stored in OBF Host.

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits **REQ** and **VIEW** are set to '1' with the same write access while **OBSYS** is '0', **OBSYS** is automatically set to '1' and OBF Shadow and OBF Host are swapped. Additionally mask bits **FROBCM. RDSH** and **FROBCM.RHSH** are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards **OBRS[6:0]** is copied to the register internal storage, mask bits **FROBCM.RDSS** and **FROBCM.RHSS** are copied to register FROBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signalled by setting **OBSYS** back to '0'.

Example of an 32-bit Host access to a single message buffer:

If a single message buffer has to be read out, two separate write accesses to **FROBCR.REQ** and **FROBCR.VIEW** are necessary:

- Wait until **FROBCR.OBSYS** is reset
- Write Output Buffer Command Mask **FROBCM.RHSS**, **FROBCM.RDSS**
- Request transfer of message buffer to OBF Shadow by writing **FROBCR.OBRS[6:0]** and **FROBCR.REQ** (in case of and 8-bit Host interface, **FROBCR.OBRS[6:0**] has to be written before **FROBCR.REQ**).
- Wait until FROBCR.OBSYS is reset
- Toggle OBF Shadow and OBF Host by writing **FROBCR.VIEW** = '1'
- Read out transferred message buffer by reading FRRDDSn, **FRRDHS1...3**, and MBS

Example of an 32-bit Host access sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until **FROBCR.OBSYS** is reset
- Write Output Buffer Command Mask **FROBCM.RHSS**, **FROBCM.RDSS** for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing **FROBCR.OBRS[6:0]** and **FROBCR.REQ** (in case of an 8-bit Host interface, **FROBCR.OBRS[6:0]** has to be written **before FROBCR.REQ**).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until **FROBCR.OBSYS** is reset
- Write Output Buffer Command Mask **FROBCM.RHSS**, **FROBCM.RDSS** for 2nd message buffer
- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer

to OBF Shadow simultaneously by writing **FROBCR.OBRS[6:0]** of 2nd message buffer, **FROBCR.REQ**, and **FROBCR.VIEW** (in case of and 8-bit Host interface, **FROBCR.OBRS[6:0]** has to be written **before FROBCR.REQ** and **FROBCR.VIEW**).

- Read out 1st transferred message buffer by reading RDDSn, **FRRDHS1...3**, and MBS

Demand access to last requested message buffer without request of another message buffer:

- Wait until **FROBCR.OBSYS** is reset
- Demand access to last transferred message buffer by writing **FROBCR.VIEW**
- Read out last transferred message buffer by reading **FRRDDSn**, **FRRDHS1...3**, and MBS

Table 24-18 Assignment of FROBCM bits

Pos.	Access	Bit	Function
17	r	RDSH	Data Section available for Host access
16	r	RHSH	Header Section available for Host access
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 24-19 Assignment of FROBCR bits

Pos.	Access	Bit	Function
22…16	r	OBRH[6:0]	OBF Request Host, number of message buffer available for Host access
15	r	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	r/w	REQ	Request Transfer from Message RAM to OBF Shadow
8	r/w	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6…0	r/w	OBRS[6:0]	OBF Request Shadow, number of message buffer for next request

(3) FlexRay protocol controller access to message RAM

The two Transient Buffer RAMs (TBF A,B) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each Transient Buffer RAM is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If e.g. the Message Handler writes the next message to be send to Transient Buffer Tx, the FlexRay Channel Protocol Controller can access Transient Buffer Rx to store the message it is actually receiving. During transmission of the message stored in Transient Buffer Tx, the Message Handler transfers the last received message stored in Transient Buffer Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the Transient Buffer RAMs and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

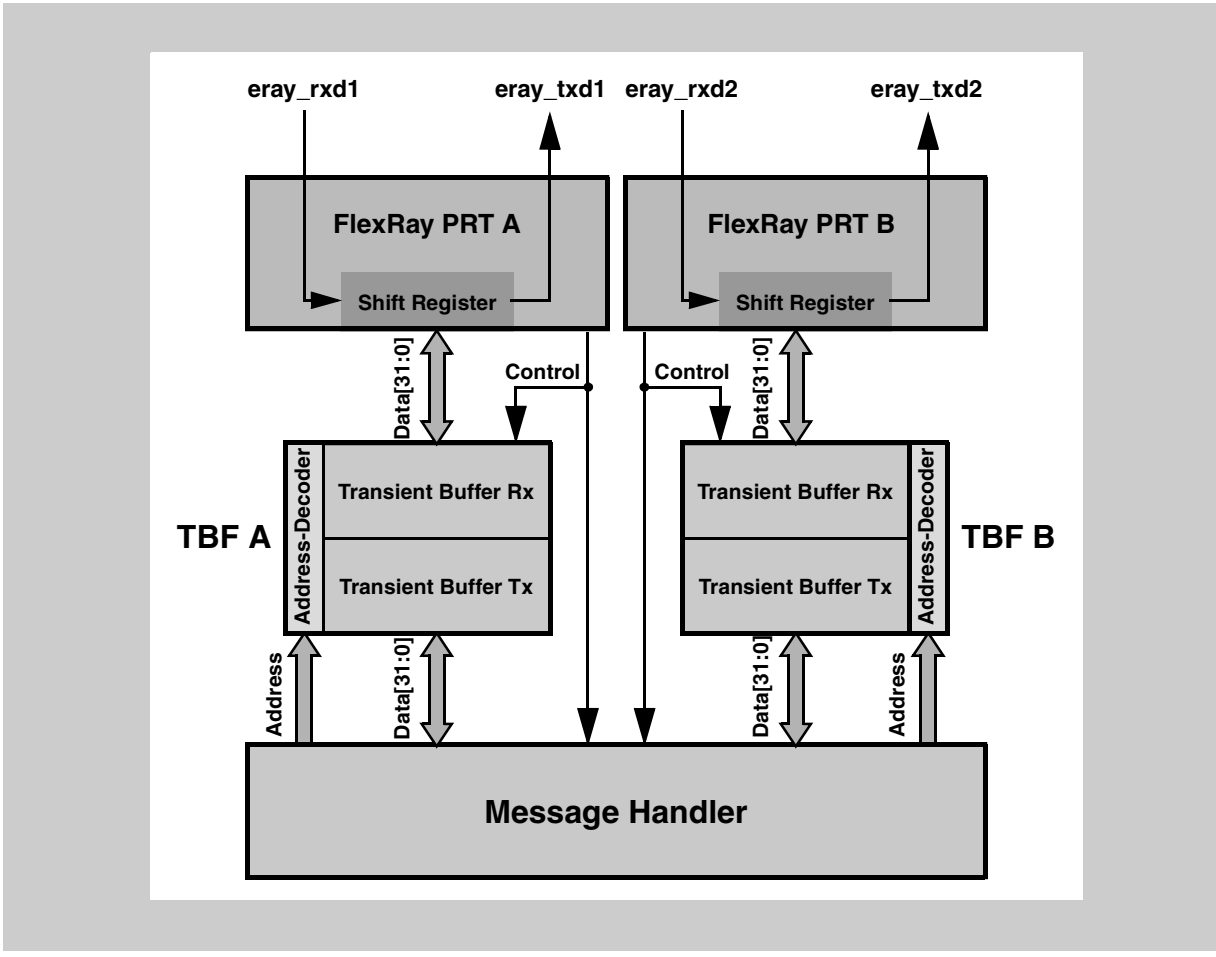


Figure 24-19 Access to transient buffer RAMs

24.5.12 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception / transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is organized 2408 x 39 = 79872 bit. Each 32-bit word is protected by a 7-bit ECC-code. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the Message RAM has a structure as shown in *Figure 24-20*.

The data partition is allowed to start at Message RAM word number: (MRC.LCB + 1) • 4

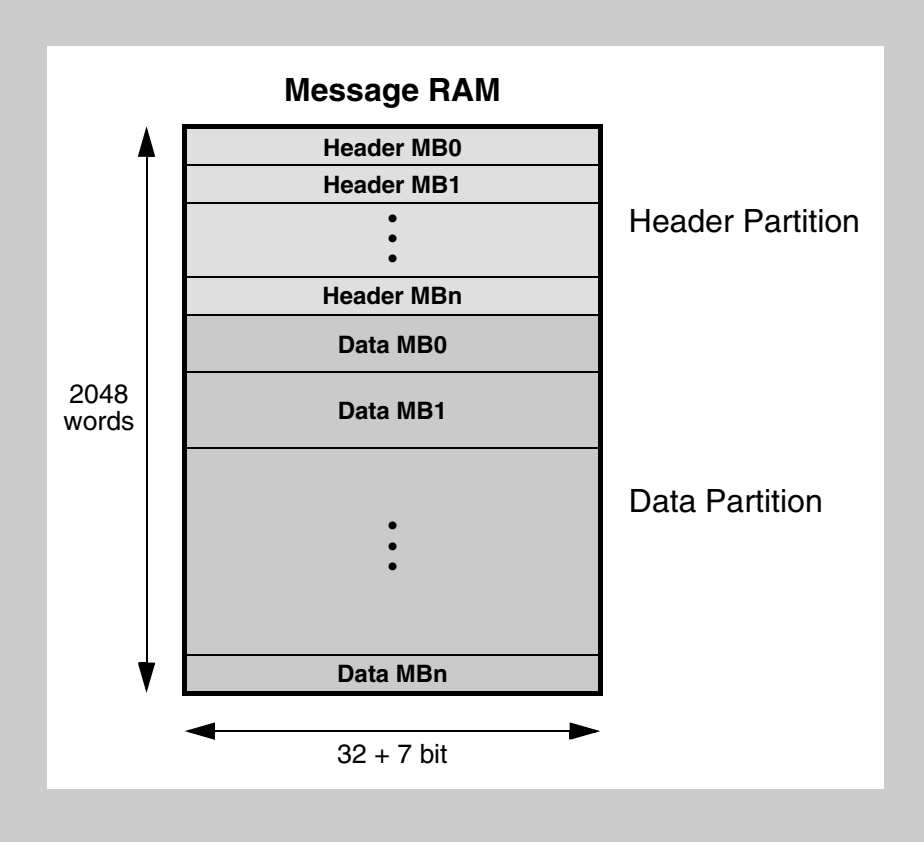


Figure 24-20 Structure of message RAM

Header partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32+1 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

Restriction: header partition + data partition may not occupy more than 2048 39-bit words.

(1) Header partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in Table below. Configuration of the header sections of the message buffers is done via IBF (WRHS13). Read access to the header sections is done via OBF (RDHS13 + MBS). The data pointer has to be calculated by the programmer to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 39-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host CPU.

Payload Length Received **PLR[6:0]**, Receive Cycle Count **RCC[5:0]**, Received on Channel Indicator **RCI**, Startup Frame Indicator **SFI**, Sync Frame Indicator **SYN**, Null Frame Indicator **NFI**, Payload Preamble Indicator **PPI**, and Reserved Bit **RES** are updated from received valid data frames only.

Header word 4 of each configured message buffer holds the respective Message Buffer Status information.

Table 24-20 Header section of a message buffer in the Message RAM

Bit Word	32 - 38	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E			MBI	TXM	PPIT	CFG	CHB	CHA		Cycle Code												Frame ID										
2	E		Payload Length Received								Payload Length Configured												Tx Buffer: Header CRC Configured Rx Buffer: Header CRC Received										
3	E			RES	PPI	NFI	SYN	SFI	RCI			Receive Cycle Count											Data Pointer										
4	E			RESS	PPIS	NFIS	SYNS	SFIS	RCIS			Cycle Count Status						FTB	FTA		MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
…	E	…																															
…	E	…																															

- Frame Configuration
- Filter Configuration
- Message Buffer Control
- Message RAM Configuration
- Updated from received Data Frame
- Message Buffer Status MBS
- ECC code
- unused

Header 1

Write access via WRHS1, read access via RDHS1:

- Frame ID - Slot counter filtering configuration
- Cycle Code - Cycle counter filtering configuration
- CHA, CHB - Channel filtering configuration
- CFG - Message buffer direction configuration: receive / transmit
- PPIT - Payload Preamble Indicator Transmit
- TXM - Transmit mode configuration: single-shot / continuous
- MBI - Message buffer receive / transmit interrupt enable

Header 2

Write access via WRHS2, read access via RDHS2:

- Header CRC
  - Transmit Buffer: Configured by the Host (calculated from frame header)
  - Receive Buffer: Updated from received frame
- Payload Length Configured
  - Length of data section (2-byte words) as configured by the Host
- Payload Length Received
  - Length of payload segment (2-byte words) stored from received frame
  stored from received frame

Header 3

Write access via WRHS3, read access via FRRDHS3:

- Data Pointer - Pointer to the beginning of the corresponding data section in the data partition

Read access via RDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count - Cycle count from received frame
- RCI - Received on Channel Indicator
- SFI - Startup Frame Indicator
- SYN - Sync Frame Indicator
- NFI - Null Frame Indicator
- PPI - Payload Preamble Indicator
- RES - Reserved bit

Header 4

Read access via MBS, updated by the CC at the end of the configured slot.

- VFRA - Valid Frame Received on channel A
- VFRB - Valid Frame Received on channel B
- SEOA - Syntax Error Observed on channel A
- SEOB - Syntax Error Observed on channel B
- CEOA - Content Error Observed on channel A
- CEOB - Content Error Observed on channel B
- SVOA - Slot boundary Violation Observed on channel A
- SVOB - Slot boundary Violation Observed on channel B
- TCIA - Transmission Conflict Indication channel A
- TCIB - Transmission Conflict Indication channel B
- ESA - Empty Slot Channel A
- ESB - Empty Slot Channel B
- MLST - Message LoST
- FTA - Frame Transmitted on Channel A
- FTB - Frame Transmitted on Channel B
- Cycle Count Status- Actual cycle count when status was updated

- RCIS - Received on Channel Indicator Status
- SFIS - Startup Frame Indicator Status
- SYNS - Sync Frame Indicator Status
- NFIS - Null Frame Indicator Status
- PPIS - Payload Preamble Indicator Status
- RESS - Reserved bit Status

(2) Data partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception / transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes plus 7-bits ECC code.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the programmer has to assure that the data pointers point to addresses within the data partition. Table 27 below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused.

Table 24-21 Example for structure of the data partition in the Message RAM

Bit Word	32 - 38	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	E	**MB0** Data3								**MB0** Data2								**MB0** Data1								**MB0** Data0							
...	E	unused								unused								**MB0** Data5								**MB0** Data4							
...	E	**MB1** Data3								**MB1** Data2								**MB1** Data1								**MB1** Data0							
...	E	...								...								...								...							
...	E	**MB1** Data(k)								**MB1** Data(k-1)								**MB1** Data(k-2)								**MB1** Data(k-3)							
...	E	...								...								...								...							
...	E	...								...								...								...							
...	E	...								...								...								...							
...	E	**MBn** Data3								**MBn** Data2								**MBn** Data1								**MBn** Data0							
...	E	...								...								...								...							
1535	E	...								...								...								...							
1536	E	**MBn** Data(m)								**MBn** Data(m-1)								**MBn** Data(m-2)								**MBn** Data(m-3)							

(3) Parity check & ECC

There is a parity checking mechanism & ECC implemented in the E-Ray core to assure the integrity of the data stored in all RAM blocks as shown in *Table 24-21*.

The Message RAM has the ECC decoding/encoding module. When data is written to the Message RAM, the local ECC encoder generates the ECC code. The ECC code is stored together with the respective data word. The ECC code is decoded each time a data word is read from the Message RAM and corrects single-bit error per word and detects double-bit error per word.

The RAM blocks except Message RAM have a parity generator / checker attached . When data is written to a RAM block, the local parity generator generates the parity bit. The E-Ray core uses an even parity (with an even number of ones in the 32-bit data word a zero parity bit is generated). The parity bit is stored together with the respective data word. The parity is checked each time a data word is read from any of the RAM blocks. The E-Ray core's internal data buses have a width of 32 bits.

If a parity error is detected in the RAMs except Message RAM, the respective error flag is set. The parity error flags **FRMHDS.PIBF**, **FRMHDS.POBF**, **FRMHDS.PTBF1**, **FRMHDS.PTBF2**, and the faulty message buffer indicators **FRMHDS.FMBD**, **FRMHDS.MFMB**, **FRMHDS.FMB[6:0]** are located in the Message Handler Status register. These single error flags control the error interrupt flag **FREIR.PERR**.

As for Message RAM, the single-bit error is silently corrected with ECC unit. In case of that, no error information is displayed in any interrupt registers or status registers. In case of double-bit error at reading Message RAM, **FRMHDS.DMR** is set and **FRMHDS.FMBD**, **FRMHDS.MFMB**, **FRMHDS.FMB[6:0]** indicates the faulty message buffer.

Note Note that the error interrupt flag **FREIR.PERR** is not set in case of the double-bit error at Message RAM. But the output pin **INTDEDFR** asserts a pulse for 1 FRBCLK period.

Figure 24-21 shows the data paths between the RAM blocks and the parity generators / checkers.

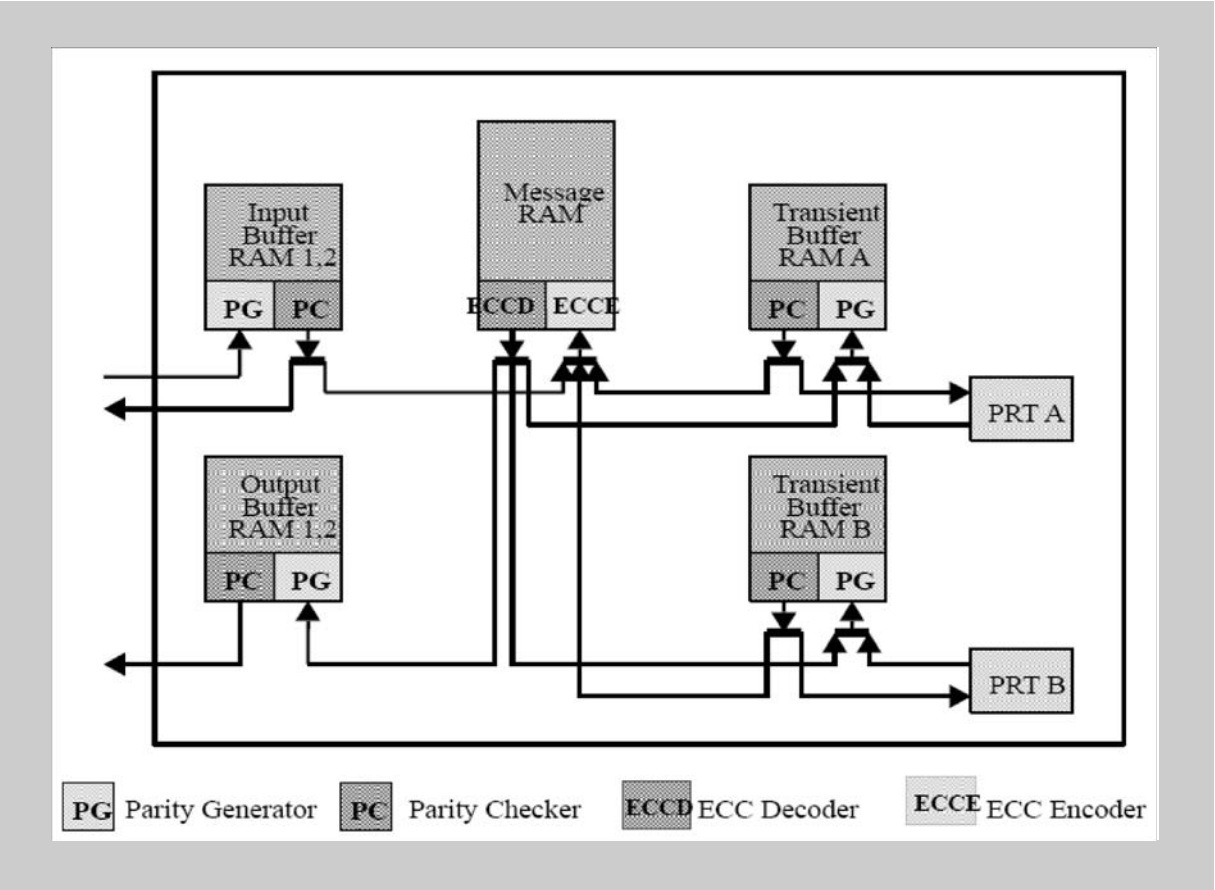


Figure 24-21 Parity/ECC generation and check

In case a parity error has been detected at Input Buffer RAM1/2, Output Buffer RAM1/2, or Transient Buffer RAM A/B, the following actions will be performed:

In all cases of parity error
• The respective parity error flag in the FRMHDS register is set
• The parity error flag **FREIR.PERR** is set, and if enabled, a module interrupt to the CPU will be generated.

In case a double-bit error has been detected at Message Buffer RAM, the following actions will be performed.

In case of double-bit error at Message RAM
• The respective parity error / double-bit error flag in the Message Handler Status register is set
• The output pin INTDEDFR asserts a pulse for 1FRBCLK period.

Additionally in specific cases
1) Memory error during data transfer from Input Buffer RAM 1,2 ⇒ Message RAM
a) Transfer of header and/or data section:
• **FRMHDS.PIBF** bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer
• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer
• Transmit buffer: Transmission request for the respective message buffer is not set
b) Transfer of data section only:
Double-bit error when reading header section of respective message buffer from Message RAM.
• **FRMHDS.DMR** bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer

• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer
• The data section of the respective message buffer is not updated
• Transmit buffer: Transmission request for the respective message buffer is not set

2) Parity error during Host reading Input Buffer RAM 1,2
• **FRMHDS.PIBF** bit is set

3) Double-bit error during scan of header sections in Message RAM
• **FRMHDS.DMR** bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer
• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer
• Ignore message buffer (message buffer is skipped)

4) Double-bit error during data transfer from Message RAM ⇒ Transient Buffer RAM 1,2
• **FRMHDS.DMR** bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer
• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer
• Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero

5) Parity error during data transfer from Transient Buffer RAM 1, 2 ⇒ Protocol Controller 1, 2
• **FRMHDS.PTBF1,2** bit is set
• Frames already in transmission are invalidated by setting the frame CRC to zero

6) Parity error during data transfer from Transient Buffer RAM 1, 2 ⇒ Message RAM
a) Double-bit error when reading header section of respective message buffer from Message RAM:
• **FRMHDS.DMR** bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer
• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer
• The data section of the respective message buffer is not updated
b) Parity error when reading Transient Buffer RAM 1, 2:
• **FRMHDS.PTBF**1,2 bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer
• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer

7) Double-bit error during data transfer from Message RAM ⇒ Output Buffer RAM
• **FRMHDS.DMR** bit is set
• **FRMHDS.FMBD** bit is set to indicate that **FRMHDS.FMB[6:0]** points to a faulty message buffer

• **FRMHDS.FMB[6:0]** indicates the number of the faulty message buffer
8) Parity error during Host reading Output Buffer RAM 1,2
• **FRMHDS.POBF** bit is set

9) Parity error during data read of Transient Buffer RAM 1, 2
When a parity error occurs when the Message Handler reads a frame with network management information (PPI = ’1’) from the Transient Buffer RAM 1, 2 the corresponding network management vector NMV1 3 is not updated from that frame.

24.5.13 Module interrupt

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the controller, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host CPU to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports disable / enable controls for each individual interrupt source separately.

An interrupt may be triggered when

- An error was detected
- A status flag is set
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers **FREIR** and **FRSIR**.

Table 24-22 Module interrupt flags and interrupt line enable (1/2)

Register	Bit	Function
FREIR	PEMC	Protocol Error Mode Changed
	CNA	Command Not Valid
	SFBM	Sync Frames Below Minimum
	SFO	Sync Frame Overflow
	CCF	Clock Correction Failure
	CCL	CHI Command Locked
	PERR	Parity Error
	RFO	Receive FIFO Overrun
	EFA	Empty FIFO Access
	IIBA	Illegal Input Buffer Access
	IOBA	Illegal Output Buffer Access
	MHF	Message Handler Constraints Flag
	EDA	Error Detected on Channel A
	LTVA	Latest Transmit Violation Channel A
	TABA	Transmission Across Boundary Channel A
	EDB	Error Detected on Channel B
	LTVB	Latest Transmit Violation Channel B
	TABB	Transmission Across Boundary Channel B

Table 24-22 Module interrupt flags and interrupt line enable (2/2)

Register	Bit	Function
FRSIR	WST	Wakeup Status
	CAS	Collision Avoidance Symbol
	CYCS	Cycle Start Interrupt
	TXI	Transmit Interrupt
	RXI	Receive Interrupt
	RFNE	Receive FIFO not Empty
	RFCL	Receive FIFO Critical Level
	NMVC	Network Management Vector Changed
	TI0	Timer Interrupt 0
	TI1	Timer Interrupt 1
	TIBC	Transfer Input Buffer Completed
	TOBC	Transfer Output Buffer Completed
	SWE	Stop Watch Event
	SUCS	Startup Completed Successfully
	MBSI	Message Buffer Status Interrupt
	SDS	Start of Dynamic Segment
	WUPA	Wakeup Pattern Channel A
	MTSA	MTS Received on Channel A
	WUPB	Wakeup Pattern Channel B
	MTSB	MTS Received on Channel B
FRILE	EINT0	Enable Interrupt Line 0
	EINT1	Enable Interrupt Line 1

The interrupt lines to the Host, **eray_int0** and **eray_int1**, are controlled by the enabled interrupts. In addition each of the two interrupt lines can be enabled / disabled separately by programming bit **FRILE.EINT0** and **FRILE.EINT1**.

The two timer interrupts generated by interrupt timer 0 and 1 are available on pins **eray_tint0** and **eray_tint1**. They can be configured via registers **FRT0C** and **FRT1C**.

A stop watch event may be triggered via input pin **eray_stpwt**.

The status of the data transfer between IBF / OBF and the Message RAM is signalled on pins **eray_ibusy** and **eray_obusy**. When a transfer has completed bit **FRSIR.TIBC** or **FRSIR.TOBC** is set.

24.6 Appendix

24.6.1 Register bit overview

FRCI	Controller Information															
0x0000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MN31	MN30	MN29	MN28	MN27	MN26	MN25	MN24	MN23	MN22	MN21	MN20	MN19	MN18	MN17	MN16
W																
page 1164	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MN15	MN14	MN13	MN12	MN11	MN10	MN9	MN8	MN7	MN6	MN5	MN4	MN3	MN2	MN1	MN0
W																
FRVI	Vendor Information															
0x0004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VI7	VI6	VI5	VI4	VI3	VI2	VI1	VI0	FMR7	FMR6	FMR5	FMR4	FMR3	FMR2	FMR1	FMR0
W																
page 1165	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	PCN7	PCN6	PCN5	PCN4	PCN3	PCN2	PCN1	PCN0
W																
FRCS	Control Settings															
0x0008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSLK7	CSLK6	CSLK5	CSLK4	CSLK3	CSLK2	CSLK1	CSLK0	0	0	0	0	0	0	0	0
W																
page 1166	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MD	SR
W																
FRTEST1	Test Register 1															
0x0010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CERB3	CERB2	CERB1	CERB0	CERA3	CERA2	CERA1	CERA0	0	0	TXENB	TXENA	TXB	TXA	RXB	RXA
W																
page 1168	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	AOB	AOA	0	0	0	0	0	0	ELBE	WRTEN
W																
FRLCK	Lock Register															
0x001C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
page 1172	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	TMK7	TMK6	TMK5	TMK4	TMK3	TMK2	TMK1	TMK0	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0
FREIR	Error Interrupt Register															
0x0020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	TABB	LTVB	EDB	0	0	0	0	0	TABA	LTVA	EDA
W																
page 1173	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	MHF	IOBA	IIBA	EFA	RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
W																
FRSIR	Status Interrupt Register															
0x0024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	0	0	0	0	MTSB	WUPB	0	0	0	0	0	0	MTSA	WUPA
	W																
page 1177		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
	W																
FREILS		Error Interrupt Line Select															
0x0028		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	TABBL	LTVBL	EDBL	0	0	0	0	0	TABAL	LTVAL	EDAL
	W																
page 1180		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHFL	IOBAL	IIBAL	EFAL	RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
	W																
FRSILS		Status Interrupt Line Select															
0x002C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	MTSBL	WUPBL	0	0	0	0	0	0	MTSAL	WUPAL
	W																
page 1182		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
	W																
FREIES FREIER		Error Interrupt Enable Set Error Interrupt Enable Reset															
0x0030 0x0034		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	TABBE	LTVBE	EDBE	0	0	0	0	0	TABAE	LTVAE	EDAE
	W																
page 1184		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHFE	IOBAE	IIBAE	EFAE	RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
	W																
FRSIES FRSIER		Status Interrupt Enable Set Status Interrupt Enable Reset															
0x0038 0x003C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	MTSBE	WUPBE	0	0	0	0	0	0	MTSAE	WUPAE
	W																
page 1186		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFFE	RFCLE	RXIE	TXIE	CYCSE	CASE	WSTE
	W																
FRILE		Interrupt Line Enable															
0x0040		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1188		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EINT1	EINT0
	W																
FRT0C		Timer 0 Configuration															
0x0044		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	T0MO 13	T0MO 12	T0MO 11	T0MO 10	T0MO9	T0MO8	T0MO7	T0MO6	T0MO5	T0MO4	T0MO3	T0MO2	T0MO1	T0MO0
	W																
page 1189		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	T0CC6	T0CC5	T0CC4	T0CC3	T0CC2	T0CC1	T0CC0	0	0	0	0	0	0	T0MS	T0RC
	W																

FRT1C	Timer 1 Configuration															
0x0048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	T1MC 13	T1MC 12	T1MC 11	T1MC 10	T1MC9	T1MC8	T1MC7	T1MC6	T1MC5	T1MC4	T1MC3	T1MC2	T1MC1	T1MC0
W																
page 1190	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T1MS	T1RC
W																
FRSTPW1	Stop Watch Register 1															
0x004C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	SMTV 13	SMTV 12	SMTV 11	SMTV 10	SMTV9	SMTV8	SMTV7	SMTV6	SMTV5	SMTV4	SMTV3	SMTV2	SMTV1	SMTV0
W																
page 1191	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SCCV5	SCCV4	SCCV3	SCCV2	SCCV1	SCCV0	0	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
W																
FRSTPW2	Stop Watch Register 2															
0x0050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	SSCVB10	SSCVB9	SSCVB8	SSCVB7	SSCVB6	SSCVB5	SSCVB4	SSCVB3	SSCVB2	SSCVB1	SSCVB0
W																
page 1193	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	SSCVA10	SSCVA9	SSCVA8	SSCVA7	SSCVA6	SSCVA5	SSCVA4	SSCVA3	SSCVA2	SSCVA1	SSCVA0
W																
FRSUCC1	SUC Configuration Register 1															
0x0080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	CCHB*	CCHA*	MTSB*	MTSA*	HCSE*	TSM*	WUCS*	PTA4*	PTA3*	PTA2*	PTA1*	PTA0*
W																
page 1194	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSA4*	CSA3*	CSA2*	CSA1*	CSA0*	0	TXSY*	TXST*	PBSY	0	0	0	CMD3	CMD2	CMD1	CMD0
W																
FRSUCC2	SUC Configuration Register 2															
0x0084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	LTN3*	LTN2*	LTN1*	LTN0*	0	0	0	LT20*	LT19*	LT18*	LT17*	LT16*
W																
page 1199	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LT15*	LT14*	LT13*	LT12*	LT11*	LT10*	LT9*	LT8*	LT7*	LT6*	LT5*	LT4*	LT3*	LT2*	LT1*	LT0*
W																
FRSUCC3	SUC Configuration Register 3															
0x0088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
page 1200	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	WCF3*	WCF2*	WCF1*	WCF0*	WCP3*	WCP2*	WCP1*	WCP0*
W																
FRNEMC	NEM Configuration Register															
0x008C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1201		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	NML3*	NML2*	NML1*	NML0*
	W																
FRPRTC1		PRT Configuration Register 1															
0x0090		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	RWP5*	RWP4*	RWP3*	RWP2*	RWP1*	RWP0*	0	RXW8*	RXW7*	RXW6*	RXW5*	RXW4*	RXW3*	RXW2*	RXW1*	RXW0*
	W																
page 1202		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	BRP1*	BRP0*	SPP1*	SPP0*	0	CASM6*	CASM5*	CASM4*	CASM3*	CASM2*	CASM1*	CASM0*	TSST3*	TSST2*	TSST1*	TSST0*
	W																
FRPRTC2		PRT Configuration Register 2															
0x0094		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	TXL5*	TXL4*	TXL3*	TXL2*	TXL1*	TXL0*	TXI7*	TXI6*	TXI5*	TXI4*	TXI3*	TXI2*	TXI1*	TXI0*
	W																
page 1203		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	RXL5*	RXL4*	RXL3*	RXL2*	RXL1*	RXL0*	0	0	RXI5*	RXI4*	RXI3*	RXI2*	RXI1*	RXI0*
	W																
FRMHDC		MHD Configuration Register															
0x0098		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	SLT12*	SLT11*	SLT10*	SLT9*	SLT8*	SLT7*	SLT6*	SLT5*	SLT4*	SLT3*	SLT2*	SLT1*	SLT0*
	W																
page 1204		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	SFDL6*	SFDL5*	SFDL4*	SFDL3*	SFDL2*	SFDL1*	SFDL0*
	W																
FRGTUC1		GTU Configuration Register 1															
0x00A0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	UT19*	UT18*	UT17*	UT16*
	W																
page 1204		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	UT15*	UT14*	UT13*	UT12*	UT11*	UT10*	UT9*	UT8*	UT7*	UT6*	UT5*	UT4*	UT3*	UT2*	UT1*	UT0*
	W																
FRGTUC2		GTU Configuration Register 2															
0x00A4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	SNM3*	SNM2*	SNM1*	SNM0*
	W																
page 1205		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MPC13*	MPC12*	MPC11*	MPC10*	MPC9*	MPC8*	MPC7*	MPC6*	MPC5*	MPC4*	MPC3*	MPC2*	MPC1*	MPC0*
	W																
FRGTUC3		GTU Configuration Register 3															
0x00A8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	MIOB6*	MIOB5*	MIOB4*	MIOB3*	MIOB2*	MIOB1*	MIOB0*	0	MTIO6*	MTIO5*	MTIO4*	MTIO3*	MTIO2*	MTIO1*	MTIO0*
	W																
page 1206		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	UIOB7*	UIOB6*	UIOB5*	UIOB4*	UIOB3*	UIOB2*	UIOB1*	UIOB0*	UIOA7*	UIOA6*	UIOA5*	UIOA4*	UIOA3*	UIOA2*	UIOA1*	UIOA0*
	W																
FRGTUC4		GTU Configuration Register 4															
0x00AC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	OCS13*	OCS12*	OCS11*	OCS10*	OCS9*	OCS8*	OCS7*	OCS6*	OCS5*	OCS4*	OCS3*	OCS2*	OCS1*	OCS0*
	W																
page 1207		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	NIT13*	NIT12*	NIT11*	NIT10*	NIT9*	NIT8*	NIT7*	NIT6*	NIT5*	NIT4*	NIT3*	NIT2*	NIT1*	NIT0*
	W																
FRGTUC5		GTU Configuration Register 5															
0x00B0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	DEC7*	DEC6*	DEC5*	DEC4*	DEC3*	DEC2*	DEC1*	DEC0*	0	0	0	CDD4*	CDD3*	CDD2*	CDD1*	CDD0*
	W																
page 1208		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	DCB7*	DCB6*	DCB5*	DCB4*	DCB3*	DCB2*	DCB1*	DCB0*	DCA7*	DCA6*	DCA5*	DCA4*	DCA3*	DCA2*	DCA1*	DCA0*
	W																
FRGTUC6		GTU Configuration Register 6															
0x00B4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	MOD 10*	MOD9*	MOD8*	MOD7*	MOD6*	MOD5*	MOD4*	MOD3*	MOD2*	MOD1*	MOD0*
	W																
page 1209		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	ASR10*	ASR9*	ASR8*	ASR7*	ASR6*	ASR5*	ASR4*	ASR3*	ASR2*	ASR1*	ASR0*
	W																
FRGTUC7		GTU Configuration Register 7															
0x00B8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	NSS9*	NSS8*	NSS7*	NSS6*	NSS5*	NSS4*	NSS3*	NSS2*	NSS1*	NSS0*
	W																
page 1210		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	SSL9*	SSL8*	SSL7*	SSL6*	SSL5*	SSL4*	SSL3*	SSL2*	SSL1*	SSL0*
	W																
FRGTUC8		GTU Configuration Register 8															
0x00BC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	NMS 12*	NMS 11*	NMS 10*	NMS9*	NMS8*	NMS7*	NMS6*	NMS5*	NMS4*	NMS3*	NMS2*	NMS1*	NMS0*
	W																
page 1210		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	MSL5*	MSL4*	MSL3*	MSL2*	MSL1*	MSL0*
	W																
FRGTUC9		GTU Configuration Register 9															
0x00C0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSI1*	DSI0*
	W																
page 1211		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MAPO4*	MAPO3*	MAPO2*	MAPO1*	MAPO0*	0	0	APO5*	APO4*	APO3*	APO2*	APO1*	APO0*
	W																
FRGTUC10		GTU Configuration Register 10															
0x00C4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	MRC10*	MRC9*	MRC8*	MRC7*	MRC6*	MRC5*	MRC4*	MRC3*	MRC2*	MRC1*	MRC0*
	W																
page 1212		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MOC13*	MOC12*	MOC11*	MOC10*	MOC9*	MOC8*	MOC7*	MOC6*	MOC5*	MOC4*	MOC3*	MOC2*	MOC1*	MOC0*
	W																
FRGTUC11		GTU Configuration Register 11															
0x00C8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	0	0	0	ERC2*	ERC1*	ERC0*	0	0	0	0	0	EOC2*	EOC1*	EOC0*
	W																
page 1213		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	ERCC1	ERCC0	0	0	0	0	0	0	ECC1	ECC0
	W																
FRCCSV		CC Status Vector															
0x0100		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	PSL5	PSL4	PSL3	PSL2	PSL1	PSL0	RCA4	RCA3	RCA2	RCA1	RCA0	WSV2	WSV1	WSV0
	W																
page 1214		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	CSI	CSAI	CSNI	0	0	SLM1	SLM0	HRQ	FSI	POCS5	POCS4	POCS3	POCS2	POCS1	POCS0
	W																
FRCCEV		CC Error Vector															
0x0104		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1217		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	PTAC4	PTAC3	PTAC2	PTAC1	PTAC0	ERRM1	ERRM0	0	0	CCFC3	CCFC2	CCFC1	CCFC0
	W																
FRSCV		Slot Counter Value															
0x0110		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	SCCB10	SCCB9	SCCB8	SCCB7	SCCB6	SCCB5	SCCB4	SCCB3	SCCB2	SCCB1	SCCB0
	W																
page 1218		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SCCA10	SCCA9	SCCA8	SCCA7	SCCA6	SCCA5	SCCA4	SCCA3	SCCA2	SCCA1	SCCA0
	W																
FRMTCCV		Macrotick and Cycle Counter Value															
0x0114		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	CCV5	CCV4	CCV3	CCV2	CCV1	CCV0
	W																
page 1218		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MTV13	MTV12	MTV11	MTV10	MTV9	MTV8	MTV7	MTV6	MTV5	MTV4	MTV3	MTV2	MTV1	MTV0
	W																
FRRCV		Rate Correction Value															
0x0118		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1219		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	RCV11	RCV10	RCV9	RCV8	RCV7	RCV6	RCV5	RCV4	RCV3	RCV2	RCV1	RCV0
	W																
FROCV		Offset Correction Value															
0x011C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	OCV18	OCV17	OCV16
	W																
page 1219		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OCV15	OCV14	OCV13	OCV12	OCV11	OCV10	OCV9	OCV8	OCV7	OCV6	OCV5	OCV4	OCV3	OCV2	OCV1	OCV0
	W																
FRSFS		Sync Frame Status															
0x0120		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	0	0	0	0	0	0	0	0	0	0	RCLR	MRCS	OCLR	MOCS
	W																
page 1220		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	VSBO3	VSBO2	VSBO1	VSBO0	VSBE3	VSBE2	VSBE1	VSBE0	VSAO3	VSAO2	VSAO1	VSAO0	VSAE3	VSAE2	VSAE1	VSAE0
	W																
FRSWNIT		Symbol Window and NIT Status															
0x0124		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1222		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
	W																
FRACS		Aggregated Channel Status															
0x0128		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1224		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	SBVB	CIB	CEDB	SEDB	VFRB	0	0	0	SBVA	CIA	CEDA	SEDA	VFRA
	W																
FRESIDn		Even Sync ID [1…15]															
0x0130 to 0x0168		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1226		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RXEB	RXEA	0	0	0	0	EID9	EID8	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
	W																
FROSIDn		Odd Sync ID [1…15]															
0x0170 to 0x01A8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1227		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RXOB	RXOA	0	0	0	0	OID9	OID8	OID7	OID6	OID5	OID4	OID3	OID2	OID1	OID0
	W																
FRNMVn		Network Management Vector [1…3]															
0x01B0 to 0x01B8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	NM31	NM30	NM29	NM28	NM27	NM26	NM25	NM24	NM23	NM22	NM21	NM20	NM19	NM18	NM17	NM16
	W																
page 1228		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	NM15	NM14	NM13	NM12	NM11	NM10	NM9	NM8	NM7	NM6	NM5	NM4	NM3	NM2	NM1	NM0
	W																
FRMRC		Message RAM Configuration															
0x0300		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	SPLM	SEC1	SEC0	LCB7*	LCB6*	LCB5*	LCB4*	LCB3*	LCB2*	LCB1*	LCB0*
	W																
page 1229		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FFB7*	FFB6*	FFB5*	FFB4*	FFB3*	FFB2*	FFB1*	FFB0*	FDB7*	FDB6*	FDB5*	FDB4*	FDB3*	FDB2*	FDB1*	FDB0*
	W																

FRFRF	FIFO Rejection Filter															
0x0304	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	RNF*	RSS*	CYF6*	CYF5*	CYF4*	CYF3*	CYF2*	CYF1*	CYF0*
W																
page 1231	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	FID10*	FID9*	FID8*	FID7*	FID6*	FID5*	FID4*	FID3*	FID2*	FID1*	FID0*	CH1*	CH0*
W																
FRFRFM	FIFO Rejection Filter Mask															
0x0308	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
page 1232	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	MFID 10*	MFID 9*	MFID 8*	MFID 7*	MFID 6*	MFID 5*	MFID 4*	MFID 3*	MFID 2*	MFID 1*	MFID 0*	0	0
W																
FRFCL	FIFO Critical Level															
0x030C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
page 1232	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	CL7	CL6	CL5	CL4	CL3	CL2	CL1	CL0
W																
FRMHDS	Message Handler Status															
0x0310	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MBU6	MBU5	MBU4	MBU3	MBU2	MBU1	MBU0	0	MBT6	MBT5	MBT4	MBT3	MBT2	MBT1	MBT0
W																
page 1233	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	FMB6	FMB5	FMB4	FMB3	FMB2	FMB1	FMB0	CRAM	MFMB	FMBD	PTBF2	PTBF1	DMR	POBF	PIBF
W																
FRLDTS	Last Dynamic Transmit Slot															
0x0314	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	LDTB10	LDTB9	LDTB8	LDTB7	LDTB6	LDTB5	LDTB4	LDTB3	LDTB2	LDTB1	LDTB0
W																
page 1234	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LDTA10	LDTA9	LDTA8	LDTA7	LDTA6	LDTA5	LDTA4	LDTA3	LDTA2	LDTA1	LDTA0
W																
FRFSR	FIFO Status Register															
0x0318	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
page 1235	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFFL	RFFL	RFFL	RFFL	RFFL	RFFL	RFFL	RFFL	0	0	0	0	0	RFO	RFCL	RFNE
W																
FRMHDF	Message Handler Constraints Flags															
0x031C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
page 1236	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	WAHP	0	0	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
W																
FRTXRQ1	Transmission Request 1															

0x0320	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXR31	TXR30	TXR29	TXR28	TXR27	TXR26	TXR25	TXR24	TXR23	TXR22	TXR21	TXR20	TXR19	TXR18	TXR17	TXR16
W																
page 1238	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR15	TXR14	TXR13	TXR12	TXR11	TXR10	TXR9	TXR8	TXR7	TXR6	TXR5	TXR4	TXR3	TXR2	TXR1	TXR0
W																
FRTXRQ2	Transmission Request 2															
0x0324	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXR63	TXR62	TXR61	TXR60	TXR59	TXR58	TXR57	TXR56	TXR55	TXR54	TXR53	TXR52	TXR51	TXR50	TXR49	TXR48
W																
page 1238	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR47	TXR46	TXR45	TXR44	TXR43	TXR42	TXR41	TXR40	TXR39	TXR38	TXR37	TXR36	TXR35	TXR34	TXR33	TXR32
W																
FRTXRQ3	Transmission Request 3															
0x0328	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXR95	TXR94	TXR93	TXR92	TXR91	TXR90	TXR89	TXR88	TXR87	TXR86	TXR85	TXR84	TXR83	TXR82	TXR81	TXR80
W																
page 1238	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR79	TXR78	TXR77	TXR76	TXR75	TXR74	TXR73	TXR72	TXR71	TXR70	TXR69	TXR68	TXR67	TXR66	TXR65	TXR64
W																
FRTXRQ4	Transmission Request 4															
0x032C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXR127	TXR126	TXR125	TXR124	TXR123	TXR122	TXR121	TXR120	TXR119	TXR118	TXR117	TXR116	TXR115	TXR114	TXR113	TXR112
W																
page 1238	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR111	TXR110	TXR109	TXR108	TXR107	TXR106	TXR105	TXR104	TXR103	TXR102	TXR101	TXR100	TXR99	TXR98	TXR97	TXR96
W																
FRNDAT1	New Data 1															
0x0330	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
W																
page 1240	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
W																
FRNDAT2	New Data 2															
0x0334	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
W																
page 1240	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
W																
FRNDAT3	New Data 3															
0x0338	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ND95	ND94	ND93	ND92	ND91	ND90	ND89	ND88	ND87	ND86	ND85	ND84	ND83	ND82	ND81	ND80
W																
page 1240	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ND79	ND78	ND77	ND76	ND75	ND74	ND73	ND72	ND71	ND70	ND69	ND68	ND67	ND66	ND65	ND64
W																
FRNDAT4	New Data 4															
0x033C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	ND127	ND126	ND125	ND124	ND123	ND122	ND121	ND120	ND119	ND118	ND117	ND116	ND115	ND114	ND113	ND112
	W																
page 1240		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND111	ND110	ND109	ND108	ND107	ND106	ND105	ND104	ND103	ND102	ND101	ND100	ND99	ND98	ND97	ND96
	W																
FRMBSC1		Message Buffer Status Changed 1															
0x0340		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	MBC31	MBC30	MBC29	MBC28	MBC27	MBC26	MBC25	MBC24	MBC23	MBC22	MBC21	MBC20	MBC19	MBC18	MBC17	MBC16
	W																
page 1242		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC15	MBC14	MBC13	MBC12	MBC11	MBC10	MBC9	MBC8	MBC7	MBC6	MBC5	MBC4	MBC3	MBC2	MBC1	MBC0
	W																
FRMBSC2		Message Buffer Status Changed 2															
0x0344		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	MBC63	MBC62	MBC61	MBC60	MBC59	MBC58	MBC57	MBC56	MBC55	MBC54	MBC53	MBC52	MBC51	MBC50	MBC49	MBC48
	W																
page 1242		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC47	MBC46	MBC45	MBC44	MBC43	MBC42	MBC41	MBC40	MBC39	MBC38	MBC37	MBC36	MBC35	MBC34	MBC33	MBC32
	W																
FRMBSC3		Message Buffer Status Changed 3															
0x0348		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	MBC95	MBC94	MBC93	MBC92	MBC91	MBC90	MBC89	MBC88	MBC87	MBC86	MBC85	MBC84	MBC83	MBC82	MBC81	MBC80
	W																
page 1242		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC79	MBC78	MBC77	MBC76	MBC75	MBC74	MBC73	MBC72	MBC71	MBC70	MBC69	MBC68	MBC67	MBC66	MBC65	MBC64
	W																
FRMBSC4		Message Buffer Status Changed 4															
0x034C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	MBC127	MBC126	MBC125	MBC124	MBC123	MBC122	MBC121	MBC120	MBC119	MBC118	MBC117	MBC116	MBC115	MBC114	MBC113	MBC112
	W																
page 1242		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC111	MBC110	MBC109	MBC108	MBC107	MBC106	MBC105	MBC104	MBC103	MBC102	MBC101	MBC100	MBC99	MBC98	MBC97	MBC96
	W																
FRWRDSn		Write Data Section [1…64]															
0x0400 to 0x04FC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
	W																
page 1246		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
FRWRHS1		Write Header Section 1															
0x0500		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
	W																
page 1247		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
FRWRHS2		Write Header Section 2															
0x0504		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	0	0	0	0	0	0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
	W																
page 1249		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W																
FRWRHS3		Write Header Section 3															
0x0508		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
page 1249		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10*	DP9*	DP8*	DP7*	DP6*	DP5*	DP4*	DP3*	DP2*	DP1*	DP0*
	W																
FRIBCM		Input Buffer Command Mask															
0x0510		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRS	LDSS	LHSS
	W																
page 1250		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRH	LDSH	LHSH
	W																
FRIBCR		Input Buffer Command Request															
0x0514		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	IBSYS	0	0	0	0	0	0	0	0	IBRS6	IBRS5	IBRS4	IBRS3	IBRS2	IBRS1	IBRS0
	W																
page 1251		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	IBSYH	0	0	0	0	0	0	0	0	IBRH6	IBRH5	IBRH4	IBRH3	IBRH2	IBRH1	IBRH0
	W																
FRRDDSn		Read Data Section [1…64]															
0x0600 to 0x06FC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
	W																
page 1253		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
FRRDHS1		Read Header Section 1															
0x0700		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
	W																
page 1254		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
FRRDHS2		Read Header Section 2															
0x0704		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	PLR6	PLR5	PLR4	PLR3	PLR2	PLR1	PLR0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
	W																
page 1255		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W																
FRRDHS3		Read Header Section 3															
0x0708		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

	R	0	0	RES	PPI	NFI	SYN	SFI	RCI	0	0	RCC5	RCC4	RCC3	RCC2	RCC1	RCC0
	W																
page 1256		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
FRMBS		Message Buffer Status															
0x070C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	0	0	CCS5	CCS4	CCS3	CCS2	CCS1	CCS0
	W																
page 1257		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R			0	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
	W																
FROBCM		Output Buffer Command Mask															
0x0710		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSH	RHSH
	W																
page 1260		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R															RDSS	RHSS
	W																
FROBCR		Output Buffer Command Request															
0x0714		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R	0	0	0	0	0	0	0	0	0	OBRH6	OBRH5	OBRH4	OBRH3	OBRH2	OBRH1	OBRH0
	W																
page 1261		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OBSYS	0	0	0	0	0	REQ	VIEW		OBRS6	OBRS5	OBRS4	OBRS3	OBRS2	OBRS1	OBRS0
	W																

24.6.2 Assignment of FlexRay Configuration Parameters

Table 24-23 FlexRay configuration parameters (1/2)

Parameter	Bit(field)
pKeySlotusedForStartup	SUCC1.TXST
pKeySlotUsedForSync	SUCC1.TXSY
gColdStartAttempts	SUCC1.CSA[4:0]
pAllowPassiveToActive	SUCC1.PTA[4:0]
pWakeupChannel	SUCC1.WUCS
pSingleSlotEnabled	SUCC1.TSM
pAllowHaltDueToClock	SUCC1.HCSE
pChannels	SUCC1.CCHA SUCC1.CCHB
pdListenTimeOut	SUCC2.LT[20:0]
gListenNoise	SUCC2.LTN[3:0]
gMaxWithoutClockCorrectionPassive	SUCC3.WCP[3:0]
gMaxWithoutClockCorrectionFatal	SUCC3.WCF[3:0]
gNetworkManagementVectorLength	NEMC.NML[3:0]
gdTSSTransmitter	PRTC1.TSST[3:0]
gdCASRxLowMax	PRTC1.CASM[6:0]

Table 24-23 FlexRay configuration parameters (2/2)

Parameter	Bit(field)
gdSampleClockPeriod	PRTC1.BRP[1:0]
pSamplesPerMicrotick	PRTC1.BRP[1:0]
gdWakeupSymbolRxWindow	PRTC1.RXW[8:0]
pWakeupPattern	PRTC1.RWP[5:0]
gdWakeupSymbolRxIdle	PRTC2.RXI[5:0]
gdWakeupSymbolRxLow	PRTC2.RXL[5:0]
gdWakeupSymbolTxIdle	PRTC2.TXI[7:0]
gdWakeupSymbolTxLow	PRTC2.TXL[5:0]
gPayloadLengthStatic	MHDC.SFDL[6:0]
pLatestTx	MHDC.SLT[12:0]
pMicroPerCycle	GTUC1.UT[19:0]
gMacroPerCycle	GTUC2.MPC[13:0]
gSyncNodeMax	GTUC2.SNM[3:0]
pMicroInitialOffset[A]	GTUC3.UIOA[7:0]
pMicroInitialOffset[B]	GTUC3.UIOB[7:0]
pMacroInitialOffset[A]	GTUC3.MIOA[6:0]
pMacroInitialOffset[B]	GTUC3.MIOB[6:0]
gdNIT	GTUC4.NIT[13:0]
gOffsetCorrectionStart	GTUC4.OCS[13.0]
pDelayCompensation[A]	GTUC5.DCA[7:0]
pDelayCompensation[B]	GTUC5.DCB[7:0]
pClusterDriftDamping	GTUC5.CDD[4:0]
pDecodingCorrection	GTUC5.DEC[7:0]
pdAcceptedStartupRange	GTUC6.ASR[10:0]
pdMaxDrift	GTUC6.MOD[10:0]
gdStaticSlot	GTUC7.SSL[9:0]
gNumberOfStaticSlots	GTUC7.NSS[9:0]
gdMinislot	GTUC8.MSL[5:0]
gNumberOfMinislots	GTUC8.NMS[12:0]
gdActionPointOffset	GTUC9.APO[5:0]
gdMinislotActionPointOffset	GTUC9.MAPO[4:0]
gdDynamicSlotIdlePhase	GTUC9.DSI[1:0]
pOffsetCorrectionOut	GTUC10.MOC[13:0]
pRateCorrectionOut	GTUC10.MRC[10:0]
pExternOffsetCorrection	GTUC11.EOC[2:0]
pExternRateCorrection	GTUC11.ERC[2:0]

24.7 Cautions

24.7.1 Missing cycle start interrupt in startup phase

Description When communication is restarted by CHI command RUN after the CC left STARTUP,NORMAL_ACTIVE, or NORMAL_PASSIVE state by application of CHI command READY it may happen, that the cycle start interrupt flag SIR.CYCS is not set in

- cycle 0 in case of a leading coldstarter
- cycles 0-2 in case of an integrating node

of the following startup phase (see FlexRay Protocol Spec v2.1, fig. 7-10).

Scope This behaviour is only of concern for applications where READY command is used to leave STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state.

Effects Leading coldstarter: In cycle 0 of the startup phase no cycle start interrupt is generated

Integrating node: In cycles 0-2 of the startup phase no cycle start interrupt is generated.

Workaround Don't leave STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by application of CHI command READY.

24.7.2 Update of status registers CCEV and CCSV delayed with respect to interrupt flags EIR.PEMC, SIR.WST, SIR.SUCS, SIR.WUPA, SIR.WUPB

Description A change of Error Mode, Protocol Operation Control Status, and Wakeup Status sets the interrupt flags EIR.PEMC, SIR.WST, SIR.SUCS, SIR.WUPA, and SIR.WUPB. The status registers are copied into the Host clock domain (CCEV, CCSV) to allow read access. This clock domain crossing delays the update of the respective status fields CCEV.ERRM[1:0], CCSV.POCS[5:0], and CCSV.WSV[2:0] by up to 5 cycles of the slower of the two clocks eray_bclk and eray_sclk in relation to the change of the interrupt flags.

Scope This behaviour is only of concern for applications where evaluation of the status fields CCEV.ERRM[1:0], CCSV.POCS[5:0], and CCSV.WSV[2:0] is triggered by a change of one of the interrupt flags EIR.PEMC, SIR.WST, SIR.SUCS, SIR.WUPA, or SIR.WUPB.

Effects In case the Host reads the Error Mode CCEV.ERRM[1:0], the Protocol Operation Control Status CCSV.POCS[5:0], or the Wakeup Status CCSV.WSV[2:0] directly after the respective interrupt flag was set, it may happen that registers CCEV respectively CCSV are not yet updated.

Workaround: Delay reading of status fields CCEV.ERRM[1:0], CCSV.POCS[5:0], or CCSV.WSV[2:0] for at least 6 cycles of the slower of the two clocks eray_bclk and eray_sclk after the respective interrupt flag was set.

24.7.3 Clearing of interrupt flags EIR.PEMC, SIR.WST, SIR.WUPA, SIR.WUPB, SIR.CAS, SIR.SUCS not accepted

Description A change of POC Error Mode, Wakeup Status, reception of a Collision Avoidance Symbol, or a successfully completed startup sets some of the interrupt flags EIR.PEMC, SIR.WST, SIR.WUPA, SIR.WUPB, SIR.CAS, and SIR.SUCS. Because the set condition for the respective interrupt flag may be active for up to 4 eray_bclk + 4 eray_sclk cycles, it can happen that the flag is not reset to ’0’ if the Host clears the flag directly after it has been set.

Scope: This behaviour is only of concern for applications where the interrupt flags EIR.PEMC, SIR.WST, SIR.WUPA, SIR.WUPB, SIR.CAS, SIR.SUCS are used for interrupt generation.

Effects: In case the Host clears one of the affected flags, it may happen that the interrupt stays active. A following read access will return ’1’ for the respective flag.

Workaround: After clearing of one of the affected interrupt flags, the Host has to re-read the flag. If the flag is still set, the Host has to repeat the sequence until it reads a ’0’.

24.7.4 Content error detected in sync frame at slot boundary

Description In case a slot is configured for sync frame reception and a valid sync frame was received on one channel while a content error is detected exactly at the last eray_sclk of that slot on the other channel, the valid sync frame is used for clock synchronization. For this case the FlexRay protocol specification requires that the valid frame is not used for clock synchronization. See FlexRay protocol specification v2.1, Fig. 6-8 on page 135.

If the content error on the other channel is detected at least one eray_sclk earlier, the valid sync frame is not used for clock synchronisation (correct behaviour). If the frame on the other channel is decoded one eray_sclk later, this results in a boundary violation, not a content error; the valid sync frame is used for clock synchronisation (also correct behaviour).

Scope This behaviour is only of concern for cases where a valid sync frame is received on one channel while a frame with content error is decoded at the last eray_sclk of the slot on the other channel.

Effects In the described case the valid sync frame is used for clock synchronization.

Workaround Not necessary.

24.7.5 Loop back mode operates only at 10 MBit/s

Description The looped back data is falsified at the two lower baud rates of 5 and 2.5 MBit/s.

Scope: This behaviour is only of concern for test cases where loop back is used with the baud rate prescaler (PRTC1.BRP[1:0]) configured to 5 or 2.5 MBit/s.

Effects: The loop back self test is only possible at the highest baud rate.

Workaround: Run loop back tests with 10 MBit/s (PRTC1.BRP[1:0] = "00").

24.7.6 False clock correction value possible in case action point coincides with end of received syncframe

Description In case the action point coincides with the end of a received sync frame, the relative deviation value (= time between secondary time reference point and action point) of this particular sync frame is seen as zero.

Critical Case When a valid sync frame is received in a specific time relation to the node's internal action point (internal signal prt_frame_decoded is high one eray_sclk period after high pulse of gtu_action_point). Not critical when the frame is decoded one eray_sclk before or one eray_sclk after this critical case.

Scope This behaviour is only of concern for cases where the frame length is configured to a smaller value than the action point offset, and the complete frame is received between start of the slot and the slot's action point.

Effects In the described case a relative deviation value of zero is calculated from the received valid sync frame. This may result in a lower absolute value for the next offset and/or rate correction.

Workaround Avoid configurations with frame length smaller than action point offset.

24.7.7 Noise following a dynamic frame that delays idle detection may fail to stop slot counting for the remainder of the dynamic segment

Description If (in case of noise) the time between 'potential idle start on X' and 'CHIRP on X' (see Protocol Spec. v2.1, Figure 5-21) is greater than gdDynamicSlotIdlePhase, the E-Ray will not remain for the remainder of the current dynamic segment in the state 'wait for the end of dynamic slot rx'. Instead, the E-Ray continues slot counting. This may enable the node to further transmissions in the current dynamic segment.

Scope This behaviour is only of concern for noise that is seen only locally and that is detected in the time window between the end of a dynamic frame's DTS and idle detection ('CHIRP on X').

Effects In the described case the faulty node may not stop slot counting and may continue to transmit dynamic frames. This may lead to a frame collision in the current dynamic segment.

Workaround None.

24.7.8 Mismatch between macrotick value and cycle counter value at cycle boundary

Description The macrotick value MTCCV.MTV[13:0] is updated one eray_bclk period before the cycle counter value MTCCV.CCV[5:0] is updated. If a stop watch event occurs in the last macrotick of a cycle (latest 4 eray_bclk periods before cycle counter increment), the captured macrotick value STPW1.SMTV[13:0] and the captured cycle counter value STPW1.SCCV[5:0] are also mismatching.

Scope This behaviour is only of concern for cases where register MTCCV is read one eray_bclk period before the cycle counter is incremented.

For the stop watch feature the erratum is limited to the case where the stop watch event occurs in the last macrotick of a cycle (latest 4 eray_bclk periods before cycle counter increment).

Effects In the described cases the values of macrotick and cycle counter in register MTCCV and the captured values of macrotick and cycle counter in register STPW1 may not be consistent.

Workaround Re-read register MTCCV if it was read at the end of a cycle.

Consider the reduced time resolution if the stop watch was triggered at the end of a cycle.

24.7.9 Wrong payload data transmitted due to disturbed channel-idle sequence

Description If the E-Ray does not detect channel-idle (11 consecutive bit HIGH) between the end of a transmission and the beginning of the next transmission in the following transmit slot due to disturbances on the FlexRay bus, a frame with wrong payload data may be transmitted in the following transmit slot.

Note Disturbances on the FlexRay bus in the extent needed to reproduce the described behaviour are beyond the scope of the FlexRay Conformance Test.

Scope This behaviour is only of concern for the case where no channel-idle sequence is detected between two frames transmitted by the same node in two consecutive static transmit slots.

Effects The frame transmitted in a slot following a transmit slot with completely destroyed channel-idle sequence holds wrong payload data.

Workaround None.

24.7.10 Status of Network Management Vector NMVn after the CC has left NORMAL_ACTIVE or NORMAL_PASSIVE state by application of commands READY, HALT, or FREEZE

Description When communication is restarted after the CC has left NORMAL_ACTIVE or NORMAL_PASSIVE state by application of commands READY, HALT, or FREEZE, it may happen that NMVn holds invalid data before re-entering NORMAL_ACTIVE state.

Scope This behaviour is only of concern for cases where NMVn is read by the Host when the CC is outside NORMAL_ACTIVE or NORMAL_PASSIVE state.

Effects In the described case the Host may read data from NMVn which originates from previous communication cycles.

Workaround Don’t evaluate the Network Management Vector NMVn outside NORMAL_ACTIVE or NORMAL_PASSIVE state.

24.7.11 Reception of wakeup pattern signalled for channel disconnected by SUCC1.CCHA/B = ’0’

Detail In case the CC is physically connected to a two channel network while one of the two channels is disabled by the Host by programming SUCC1.CCHA/B = ’0’, the CC will signal the reception of a wakeup pattern on that channel by SIR.WST = ’1’ and CCSV.WSV = "010" (RECEIVED_WUP).

Scope This behaviour is only of concern for cases where the CC is physically connected to both channels of a two channel network and one of the two channels is configured to be disconnected by the Host by programming SUCC1.CCHA/B = ’0’.

Effects Despite SUCC1.CCHA/B = ’0’ the CC signals the reception of wakeup pattern on the virtually disconnected channel.

Workaround None.

24.7.12 Clock correction value calculated with wrong deviation value

Description In case of receiving a valid frame after detecting an invalid frame before the action point, the clockcorrection value is calculated with a wrong deviation value.

Scope This behaviour is only of concern for the case where an invalid frame (at least valid TSS, FSS, and BSS) is detected before the action point and a valid frame, starting after the action point, is received in the same slot.

Effects The relative deviation (= time between secondary time reference point and action point) of the invalid frame is stored for the following rate and offset correction value calculation. This wrong deviation value may be used by the FTM algorithm to calculate the midpoint. Only in this case a wrong offset and/or rate correction value is calculated.

Workaround None.

24.7.13 Wrong payload data written to receive buffer

Description When a receive slot is directly followed by a transmit slot and when the eray_bclk frequency is below the minimum frequency required for the configured (minislot) action point offset as listed in the table below, it may happen, that a word of the received payload is stored twice into the respective receive buffer in the Message RAM while all following words are shifted by one address, and the last word is lost.

With the maximum payload of 254 byte and both channels used (**SUCC1.CCHA/B** = '1') the problem may appear, depending on the configured (M)APO and TSST, if the eray_bclk is below the minimum value listed in the table below.

(M)APO[a]	Minimum era_bclk	
	TSST = 10	TSST = 3
1	63 MHz	70 MHz
2	49 MHz	53 MHz
3	40 MHz	43 MHz
4	34 MHz	36 MHz
5	30 MHz	31 MHz

a) APO: Action Point Offset
MAPO: Minislot Action Point Offset

$$bclk_{min} = \frac{148 \cdot 3}{(M)APO \cdot 2\mu s + (TSST + 41) \cdot 0,1\mu s}$$

In case only one channel is used (**SUCC1.CCHA** or **B** = '0') the values above can be multiplied by 0.66. With lower payload values the minimum eray_bclk also decreases.

Scope This behaviour is only of concern for the case where eray_bclk is below a minimum frequency. This frequency depends on the configuration of **GTUC9.APO**, **GTUC9.MAPO**, and **PRTC1.TSST** as well as on the configured payload.

Effects The payload of the affected receive buffer is falsified.

Workaround Configure (M)APO and TSST according to table above depending on the used eray_bclk frequency.

24.7.14 In case of a faulty configuration of pLatestTx and transmission across dynamic segment boundary error flags EIR.LTVA/B may not be set when a latest transmit violation occurs

Description Prerequisites:

1. Faulty configuration of pLatestTx (**MHDC.SLT[12:0]**) does not prevent transmission of frame X in cycle n and frame Y in cycle n+1.
2. The last dynamic frame X transmitted in cycle n ends in minislot m. Fault Case:
   In case frame Y is the only frame transmitted in the dynamic segment of cycle n+1 and frame Y is transmitted across the end of dynamic segment and the last minislot of dynamic segment has the value m, error flags **EIR.LTVA/B** (Latest Transmit Violation Channel A,B) are not set while **EIR.TABA/B** (Transmission Across Boundary Channel A,B) are set as specified.

Scope This behaviour is only of concern for the case where a transmission across dynamic segment boundary is not prevented because of a faulty configuration of pLatestTx (**MHDC.SLT[12:0]**).

Effects In case of transmission across dynamic segment boundary the CC enters POC state HALT and the error flags **EIR.TABA** or **EIR.TABB** are set. In case of a faulty configuration of pLatestTx, the flags **EIR.LTVA** or **EIR.LTVB** are not set.

Workaround Configure pLatestTx as required by the FlexRay protocol specification v2.1.

24.7.15 In case eray_bclk is below eray_sclk/2, TEST1.CERA/B may fail to report a detected coding error

Description All detected coding errors should be reported in the Test Register 1, at **TEST1.CERA/B**. If eray_bclk is below eray_sclk/2, it may happen, depending on phase difference of eray_bclk and eray_sclk, that **TEST1.CERA/B** are not updated in case of a detected coding error and remain in the state "0000" = "No coding error detected".

Scope The erratum is limited to the case where eray_bclk is below eray_sclk/2.

Effects Coding errors not reported via **TEST1.CERA/B**. The frame decoding is not affected.

Workaround None.

24.7.16 Generating EIR.SFO considers number of sync frames in the last cycle only, double cycle not evaluated

Description In case that there are different sync IDs received for the two cycles of a double cycle, and the total number of sync frames with different IDs received in the double cycle exceeds the maximum number of sync frames as configured by **GTUC2.SNM[3:0]** while the number of sync frames within each cycle of the double cycle is below **GTUC2.SNM[3:0]**, the sync frame overflow indication **EIR.SFO** is not set.

Scope This behaviour is only of concern for the case where sync frames with different IDs are received in even and odd cycles and where the total number of sync frames is greater than **GTUC2.SNM[3:0]**.

Effects No sync frame overflow signalled if number of received sync frames in each of the two cycles of a double cycle is below the maximum number of sync frames as configured by **GTUC2.SNM[3:0]**.

Workaround None.

24.7.17 CAS collision in case of macrotick length > CAS

Description A leading coldstarter that has switched from state COLDSTART_LISTEN to COLDSTART_COLLISION_RESOLUTION and that receives a CAS symbol transmitted by another coldstarter in the time window of cCASActionPointOffset (1 MT) after the state change will transmit a CAS symbol at the CAS action point. This CAS symbol should have been suppressed.

Scope This behaviour is only of concern for the case where the macrotick is configured to be longer than the CAS symbol.

Effects A CAS collision will disturb the first cycle of the startup, delaying the startup success by one cycle.

Workaround None.

24.7.18 Reception of more than gSyncNodeMax different sync frames per double cycle may lead to incorrect offset correction value

Description In case of receiving gSyncNodeMax or more sync frames in an even cycle, only frames with the same sync frame IDs, as received in the even cycle, may be used for offset correction term calculation in the following odd cycle.

The E-Ray erroneously uses the first gSyncNodeMax sync frames for offset correction term calculation in the odd cycle, regardless whether they have been also received in the previous even cycle.

Scope This behaviour is only of concern for the case where more than gSyncNodeMax nodes are configured to transmit sync frames and where different sets of sync frames are transmitted in even and odd cycle.

Effects In the described case the offset correction term may base on a different set of sync frames than the rate correction term. In this case registers ESIDn / OSIDn hold the IDs of the first received sync frames up to gSyncNodeMax used for offset correction term calculation.

Workaround Avoid faulty configurations with more than gSyncNodeMax nodes configured to be transmitter of sync frames.

24.7.19 Time stamp of the wrong channel may be used for offset correction term calculation if reception time stamps of a sync frame are different for channel A and B

Description In case the temporal deviation (= time between primary time reference point and action point offset) is different for channel A and B and the values have the following combination • greater than or equal zero on one channel and negative on the other channel the channel with a relative deviation value greater than or equal zero is choosen for offset correction term calculation instead of the negative value.

Scope This behaviour is only of concern for the case where both channels are used and if there is a large difference in the propagation delays on channel A and B.

Effects In case of the described relation between measured deviation values on channel A and B, the calculated offset correction term may have an error of maximum the difference between the two deviation values. Thus, the error of the local time of the node is limited to the difference of the temporal deviation values of both channels.

Workaround For dual channel FlexRay systems sync frames have to be transmitted on both channels. In practice, the propagation delay between two nodes is expected to be nearly the same on both channels. If this is not the case, the channel depending parameter pDelayCompensation[A/B] (**GTUC5.DCA,B**) has to be used to compensate the different propagation delays. With a correct adjustment of the parameter the difference of the deviation values on both channels is expected to be very low. Therefore, the error of the local time caused by the implementation error is also minimized.

24.7.20 Sync frame reception after noise or aborted frame before action point

Description In case noise or an aborted frame leads to the detection of a secondary time reference point (STRP), and after this, a valid sync frame is detected in the same slot and the STRP of the valid sync frame occures simultanoeusly with the action point, the temporal deviation value of the first detected STRP is stored instead of the value of the correct STRP.

Scope This behaviour is only of concern for the case of noise before reception of a valid sync frame or a frame reception starting before action point is aborted and then a valid sync frame is received in the same slot.

Effects In the described case a wrong deviation value is used for correction term calculation. Depending on number of sync frames and other measured temporal deviation values it may lead to an incorrect rate or offset correction value.

Workaround None.

24.7.21 Additional SIR.WST events

Description The Status Interrupt flag **FRSIR.WST** is set not only at the conditions described in der E-Ray Specification at chapter 4.4.2, but also when the wakeup process is aborted by CHI-Commands FREEZE or READY.

Scope This behaviour is only of concern for cases where a wakeup process is intentionally aborted by the Host.

Effects **FRSIR.WST** is set even if the conditions described in the E-Ray Specification are not met.

Workaround Ignore **FRSIR.WST** after the Host intentionally aborted a wakeup process.

24.7.22 Update of Aggregated Channel Status ACS in dynamic segment in minislots following ID 2047

Description In case the slot counter has reached ID 2047 and the end of dynamic segment is not reached, the slot counter wraps around to 0 and stays there until the end of the dynamic segment. In this state (slot counter = 0) the E-Ray erroneously updates register ACS every minislot. The correct behaviour is that the slot status is only updated at the end of the dynamic segment.

Scope This behaviour is only of concern for the following case:
gNumberOfStaticSlots + gNumberOfMinislots > cSlotIDMax = 2047.

Effects In the described case register ACS is updated every minislot until the end of the dynamic segment. This update may also lead to an update of error interrupt flags **FREIR.EDA, B**.

Workaround Avoid configurations with
gNumberOfStaticSlots + gNumberOfMinislots > cSlotIDMax.

24.7.23 Faulty update of FRLDTS.LDTA,B[10:0] due to ECC error

Description Faulty update of **FRLDTS.LDTA,B[10:0]** due to ECC error.

Scope This behaviour is only of concern for cancelled transmissions of dynamic frames when a ECC error occurs during data transfer from Message RAM to the Transient Buffer.

Effects In case a ECC error occurs when the message handler transfers data from the Message RAM to the Transient Buffer it may happen, that the transmission is not started.

Workaround When the described condition occurs, the slot counter value is captured and **FRLDTS.LDTA,B[10:0]** is updated with this value at the end of the dynamic segment.

24.7.24 Improper resolution of startup collision

Description In case a CAS symbol is received during startup exactly at the beginning of cycle 0, the detection of following startup frames is not possible.

Scope This behaviour is only of concern for case where a CAS symbol is received during startup exactly at the beginning of cycle 0.

Effects In the described case the CC is not able to transit from STARTUP to NORMAL_ACTIVE state.

Workaround Leave and re-enter STARTUP state by Host commands READY and RUN.

24.7.25 'integration successful on X' and 'integration abort on Y' at the same point in time leads to inconsistent states of SUC and GTU

Description

In case of 'integration successful on X' and 'integration abort on Y' at the same point in time, the SUC prioritizes the input event of successful integration, leaves the state POC:initialize schedule and enters, depending of its startup role, either POC:integration coldstart check or POC:integration consistency check.

The reaction of the GTU depends on the related channels and the actual state of clock synchronisation startup process.

Assumed is that the GTU is in state CSP:A active. For the combination of 'integration successful on A' and 'integration abort on B' the GTU stops the macrotick generation process and terminates both clock synchronisation startup processes. In this case the CC is stuck in POC:integration coldstart check or POC:integration consistency check.

For the other combination of 'integration successful on B' and 'integration abort on A' the GTU stops the macrotick generation process but keeps the clock synchronisation startup process of channel A running. This leads to a delayed (best case two cycles) successful startup of the E-Ray.

If the GTU is in state CSP:B active, the same is true with the swapped channels. The combination of 'integration successful on B' and 'integration abort on A' leads to the stuck condition and the combination of 'integration successful on A' and 'integration abort on B' leads to a delayed startup.

Scope

This behaviour is only of concern for cases of simultaneous generation of internal signals 'integration successful on X' on one channel and 'integration abort Y' on the other channel.

Effects

In the described cases the E-Ray either is stuck in startup states or extends the startup by at least two cycles.

Workaround

Use a timer to measure how long the E-Ray stays in state POC:integration coldstart check or POC:integration consistency. If the timeout is reached, re-enter the startup state by using the CHI commands READY and RUN.

24.7.26 Detection of parity errors outside immediate scope

Description The protocol engine reads at least the first two words of its Transient Buffer and one word more than required by the payload count for each transmitted message, even if no data is needed (null frame or payload count is zero). If a parity error is detected when the protocol engine reads from the Transient Buffer, the transmitted message is invalidated by setting its CRC code to zero, as described in chapter 5.12.3 of the User's Manual.

Scope This behaviour is only of concern for the case where a parity error occurs in one of the words in the Transient Buffer that are read but not needed for a particular message.

Effects The transmitted message is invalidated.

Workaround The transmitted message is invalidated.

24.7.27 FRCCSV.SLM [1:0] delayed to FRCCSV.POCS[5:0] on transitions between states WAKEUP and READY

Description When the POC state changes between WAKEUP and READY the content of register **FRCCSV** may show a slight discontinuity, i.e. **FRCCSV.SLM [1:0]** may be updated late.

Scope This behaviour is only of concern for configurations with **FRSUCC1.TSM** = ’0’ (ALL Slot Mode).

Effects None, **FRCCSV.SLM [1:0]** only relevant in POC states NORMAL_ACTIVE and NORMAL_PASSIVE.

Workaround Ignore **FRCCSV.SLM [1:0]** in states READY and WAKEUP.

24.7.28 Wakeup listen counter started one bit time early

Description If the protocol engine is in the state WAKEUP_LISTEN and if the parameter gdWakeup-SymbolRxLow is programmed to a value 11-n, then the channel idle recognition CHIRPcomes n bit times early.

Scope This behaviour is only of concern for configurations with gdWakeupSymbolRxLow < 11. For bit rates of 10/5/2.5 MBit/s, Protocol Version 2.1 Rev A requires a minimum gdWakeupSymbolRx-Low of 46/23/11 gdBit.

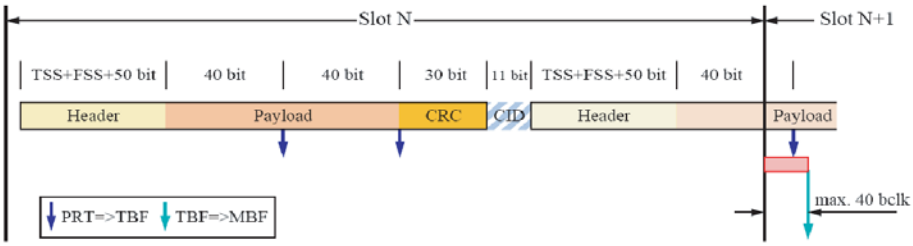
Effects The wakeup pattern is transmitted n bit times early.

Workaround Set gdWakeupSymbolRxLow to value ≥ 11.

24.7.29 Payload corruption after reception of valid frame followed by slot boundary crossing frame

Description If after reception of a valid frame in slot N the reception of a second frame (transmitted by a mis-synchronized node) starts in the same slot and the second frame's payload is stored into the TBF shortly after the slot boundary, the first 32 bit payload data of the first received frame in the TBF is overwritten by the payload data from the second frame.

Scope This behaviour is only of concern for cases where the complete header and a part of the payload of another frame is received in slot N.



The transfer of a received valid frame from TBF to MBF is initiated by the end of the actual slot. Execution is started not later than 40 bclk periods after the slot change. 40 bclk periods equate 10 bit times, when bclk=40MHz. Corruption of payload occurs when a transfer from PRT to TBF happens in the red marked time window between start of new slot and start of transfer from TBF to MBF.

Effects The first two words (4 byte) of the received valid frame's payload are corrupted.

Workaround

1. Ensure that the static slot length is configured suited to the static payload length.
2. Check Message Buffer Status for boundary violation.

24.7.30 FRCCSV.RCA[4:0] and FRCCSV.PSL[5:0] inconsistent after FREEZE command

Description When the FREEZE command is applied during STARTUP it may happen, that after the command **FRCCSV.RCA** and **FRCCSV.PSL** are inconsistent.

Scope This behaviour is only of concern for cases where the FREEZE command coincides with a protocoltriggered state change where **FRCCSV.RCA[4:0]** is decremented.

Effects **FRCCSV.RCA[4:0]** is decremented but **FRCCSV.PSL[5:0]** does not show state COLDSTART_COLLISION_RESOLUTION.

Workaround Ignore **FRCCSV.RCA[4:0]** after CHI command FREEZE.

24.7.31 CLEAR_RAMS command does not clear the 1st RAM word

Description After execution of the CLEAR_RAMS command, the 1st RAM word holds the last data written to IBF. Reason is that the registers to support byte access to the RAM are cleared one clock cycle after the CLEAR_RAMS command was started.

Scope This behaviour is only of concern for cases where the CLEAR_RAMS command is applied during ERay operation. The execution of the CLEAR_RAMS sequence after hard reset is not affected.

Effects Execution of the CLEAR_RAMS command does not reset the first word of an E-Ray internal RAM to zero.

Workaround Write 0x00000000 to any address of IBF directly before applying CLEAR_RAMS command.

24.7.32 Cycle filtering in slot 1

Description The message buffer for slot 1 is searched in parallel to every scan in the previous cycle. A message buffer scan is started every 8th slot. A running scan is aborted when the NIT is reached.The message buffer used for slot 1 depends on the cycle filter configuration and the bus activity in the dynamic segment. If the scan is aborted between the first and the last message buffer assigned to slot 1, it is unpredictable, if the correct message buffer is used.

Scope This behaviour is only of concern for the case where two (or more) message buffers are configured for slot 1 and cycle filtering is used.

Effects If a running message buffer scan is interrupted by the NIT it cannot be guaranteed that the correct message buffer is used for transmission in slot 1of the next cycle.

Workaround If cycle filtering is used, assign all message buffers configured for slot 1 to the static buffers section. I.e. message buffer number < **FRMRC.FDB[4:0]**.

24.7.33 Corruption of frame received in slot N by second frame reception before action point of transmit slot N+1

Description If a receive slot N is followed by a transmit slot N+1, and if between end of frame reception in slot N and start of frame transmission in slot N+1 the reception of a frame (transmitted by a mis-synchronized node) is started, it may happen, that header and/or payload of the valid frame received in slot N is corrupted.

Case 1:
Frame reception completed in slot N

Case 2:
Frame reception across slot boundary between slot N and slot N+1

Case 3:
Frame reception starts in slot N+1

Scope This behaviour is only of concern for the case where a receive slot is followed by a transmit slot and where at least the complete header of another frame is received before the frame received in slot N has been stored into the respective message buffer.

Effects Case 1:
Syntax error signalled for slot N, correct behaviour, no corruption

Case 2:
Boundary violation signalled for slot N and slot N+1. Header and/or payload of the message buffer assigned to slot N may be corrupted or frame received in slot N may be completely rejected.

Case 3:
No error signalled for slot N. Frame received in slot N may be not stored or header of the message buffer assigned to slot N may be corrupted. Corruption of the assigned message buffer's payload may only occur when eray_bclk is below 25 MHz.

Workaround None

24.7.34 Corruption of frame received in slot N by second frame reception before action point of transmit slot N+1

Description If the protocol engine is in the state Coldstart_Gap, it stops transmitting its own startup frame (according to the protocol specification), but the status data exported to the CHI (SFS, OSIDn register) lists its own startup-frame as transmitted.

Scope This behaviour is only of concern for leading coldstart nodes.

Effects Misleading status data.

Workaround Ignore misleading status data in state Coldstart_Gap.

24.7.35 Corruption of frame received in slot N by second frame reception before action point of transmit slot N+1

Description In case of a WUP that starts with a duration at the LOW level that is longer then n*gdWakeupSymbolRxWindow but shorter then (n*gdWakeupSymbolRxWindow + gdWakeupSymbolRxLow), followed by a duration of at least gdWakeupSymbolRxIdle at the HIGH level and followed by a duration of at least gdWakeupSymbolRxLow at the LOW level, the last part of this received within a window with a duration of at most gdWakeup-SymbolRxWindow, then this WUP is detected as valid while it should be considered invalid according to figure 3-39 of the FlexRay protocol specification.

Scope This behaviour is only of concern for cluster wakeup with disturbances on a channel.

Effects Detection of WUP is independent of WakeupSymbolRxWindow's phase.

Workaround None needed.

Chapter 25 Reset Function

25.1 Features

- Reset function by $\overline{\text{RESET}}$ input
  Analog filter eliminates noise from the $\overline{\text{RESET}}$ pin
- Forced reset function by DCU (refer to *"On-Chip Debug Unit" on page 1119*)
- Clock Monitor reset CLMRES
- Reset function by DDI = L while $\overline{\text{DRST}}$= L (N-Wire interface)

25.2 Configuration

During a system reset, most pins enter the high-impedance state.

Therefore, if an external device always requires a defined input level (e.g. external memory) a pull-up (or pull-down) resistor must be connected to each concerned output pin to prevent signal lines from floating. If no resistor is connected, the external device may be destroyed when these pins enter the high-impedance state.

For a list of all pin states after reset, see *"Pin Functions during and after Reset" on page 57*.

(1) Hardware status

With reset the hardware is initialized. When the reset status is released, program execution is started.

The following table describes the status of the clocks during reset and after reset release. Note that the clock status "operates" does not inevitably mean that any function using this clock source operates as well. The function may additionally require to be enabled by other means.

Table 25-1 Hardware status during and after reset

Item	During reset	After reset
Main oscillator	Operation continues	Operation continues
Internal oscillator	Operation stopped	Starts oscillation
PLL clocks	Operation stopped	Starts operation
CPU system clock	Operation stopped	Starts operation after PLL stabilization time
CPU	Operation stopped	Initialized and program execution starts after PLL stabilization time
Peripheral clocks	Operation stopped	Start after PLL stabilization time
On-chip peripheral functions	Operation stopped	Initialized and can be operated after PLL stabilization time
I/O pins (port/alternative function pins)	See chapter *"Pin Functions" on page 1* for a description.	

(2) Register status

With each reset the registers of the CPU, internal RAM, and on-chip peripherals are initialized. After a reset, make sure to set the registers to the values needed within your program.

Table 25-2 Initial values of CPU and internal RAM after reset

On-chip hardware		Register name	Initial value after Reset
CPU	Program registers	General-purpose register (r0)	0000 0000$_H$
		General-purpose registers (r1 to r31)	Undefined
	System registers	Program counter (PC)	0000 0000$_H$
		Program status word (PSW)	0000 0020$_H$
		Status saving registers during maskable interrupt (EIPC, EIPSW)	Undefined
		Status saving registers during non-maskable interrupt (NMI) (FEPC, FEPSW)	Undefined
		Status saving registers during CALLT execution (CTPC, CTPSW)	Undefined
		Status saving registers during exception/debug trap (DBPC, DBPSW)	Undefined
		Interrupt source register (ECR)	0000 0000$_H$
		CALLT base pointer (CTBP)	Undefined
		Floating-point arithmetic control register (ECT)	0000 0000$_H$
		Floating-point arithmetic status register (EFG)	0000 0000$_H$
Internal RAM			Undefined
Peripherals		Macro internal registers	The reset values of the various registers are given in the chapters of the peripheral functions

25.3 Operation

When a low-level signal is input to the $\overline{\text{RESET}}$ pin, a system reset is effected and each on-chip hardware is initialized.

When the oscillation stabilization time has elapsed, the external $\overline{\text{RESET}}$ can be released. The internal PLL stabilization time counter (PSTC) is started after some delay, induced by the analog noise rejection filter. The PSTC counts 2^{14} oscillator clocks (f_X) and determines the PLL lock state. The internal system reset is released, the CPU is supplied with the PLL output clock and the user's software is started up.

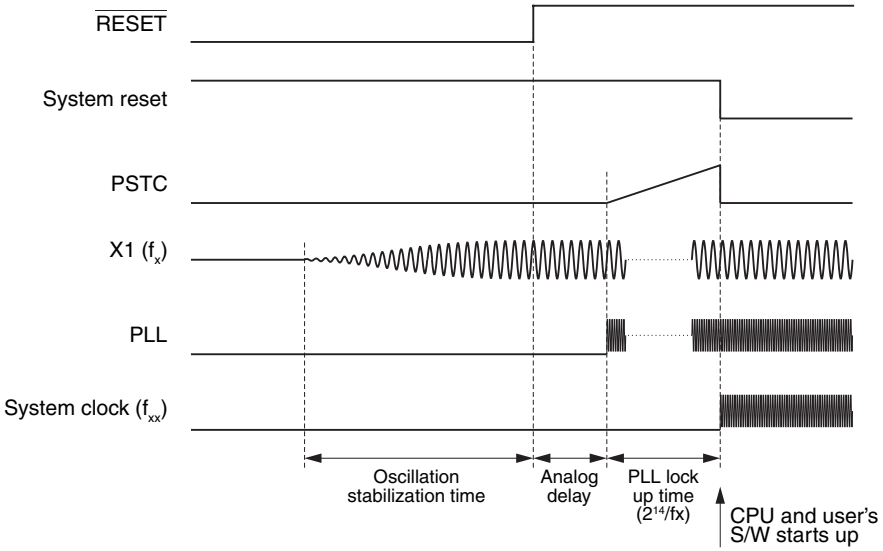


Figure 25-1 Reset timing

Note
1. If no clock is supplied (i.e. the oscillator does not work) the internal system reset will not be released independently from input level of the external $\overline{\text{RESET}}$ pin.
2. During $\overline{\text{RESET}}$ the PLL is stopped. Therefore the PLL lock-up time is always necessary after $\overline{\text{RESET}}$ release.
3. The on-chip debug function can force the activation of the system reset independently from input level of the $\overline{\text{RESET}}$ pin.

Chapter 26 Boundary Scan

The V850E/PHO3 microcontroller provides a JTAG interface and the boundary scan test methods in accordance with the IEEE1149.1 specification.

The boundary scan mode is selected via the operation mode selection pins MODE0 to MODE3, refer to *"CPU System Functions" on page 97*.

26.1 Boundary Scan Pins

The pins of boundary scan test bus are shared with the pins for the N-Wire on-chip debug unit, as described in *Table 26-1*.

Table 26-1 Boundary scan and N-Wire pins

Boundary scan (JTAG) pin name	N-Wire pin name	I/O	Function
TCK	DCK	I	Clock input
TDI	DDI	I	Command/data input
TDO	DDO	O	Command/data output
TMS	DMS	I	Mode selection
$\overline{\text{TRST}}$	$\overline{\text{DRST}}$	I	Reset

Following pins are not part of the boundary scan chain:

- JTAG interface ($\overline{\text{TRST}}$, TCK, TMS, TDI, TDO)
- Operation mode selection pins (MODE0 to MODE3)
- A/D Converter input pins (ANI00 to ANI09, ANI10 to ANI19)
- Main oscillator crystal pins (X1, X2)
- Power supply and ground pins (VDD15x, VSS15x, VDD30x, VSS30x, CVDD15, CVSS15, AVDD, AVSS, AVREF0, AVREF1, AVSS0, AVSS1)
- Miscellaneous pins (ICA, IC3)

TDO/DDO The TDO/DDO output pin of the JTAG/N-Wire interface may output different levels, depending on the mode:

Table 26-2 TDO/DDO pin levels

$\overline{\text{DRST}}$/$\overline{\text{TRST}}$	Operation mode	TDO/DDO output
0	X	Hi-Z
1	N-Wire	output level "0" or "1"
	Boundary scan	output level "0", "1" or Hi-Z

26.2 Boundary Scan Functions

Following IEEE1149.1 boundary scan functions are supported:

Table 26-3 Boundary scan functions

Instruction code	Instruction name
00000_B	EXTEST
00011_B	SAMPLE/PRELOAD
11111_B	BYPASS

Appendix A Special Function Registers

The following tables list all registers that are accessed via the NPB (NEC peripheral bus). The registers are called “special function registers” (SFR).

Table A-1 lists all CAN special function registers.

Table A-2 lists all FlexRay-specific registers.

A.1 CAN Registers

The CAN registers are accessible via the programmable peripheral area. The addresses are given as offsets to the programmable peripheral base address (refer to *“CAN module register and message buffer addresses” on page 454*.)

Table A-1 CAN special function registers (1/3)

Address offset	Register name	Shortcut	1	8	16	32
0x000	CAN0 Global Macro Control register	C0GMCTRL	-	-	R/W	-
0x000	CAN0 Global Macro Control register low byte	C0GMCTRLL	R/W	R/W	-	-
0x001	CAN0 Global Macro Control register high byte	C0GMCTRLH	R/W	R/W	-	-
0x002	CAN0 Global Macro Clock Selection register	C0GMCS	R/W	R/W	-	-
0x006	CAN0 Global Macro Automatic Block Transmission register	C0GMABT	-	-	R/W	-
0x006	CAN0 Global Macro Automatic Block Transmission register low byte	C0GMABTL	R/W	R/W	-	-
0x007	CAN0 Global Macro Automatic Block Transmission register high byte	C0GMABTH	R/W	R/W	-	-
0x008	CAN0 Global Macro Automatic Block Transmission Delay register	C0GMABTD	R/W	R/W	-	-
0x040	CAN0 Module Mask 1 register lower half word	C0MASK1L	-	-	R/W	-
0x042	CAN0 Module Mask 1 register upper half word	C0MASK1H	-	-	R/W	-
0x044	CAN0 Module Mask 2 register lower half word	C0MASK2L	-	-	R/W	-
0x046	CAN0 Module Mask 2 register upper half word	C0MASK2H	-	-	R/W	-
0x048	CAN0 Module Mask 3 register lower half word	C0MASK3L	-	-	R/W	-
0x04A	CAN0 Module Mask 3 register upper half word	C0MASK3H	-	-	R/W	-
0x04C	CAN0 Module Mask 4 register lower half word	C0MASK4L	-	-	R/W	-
0x04E	CAN0 Module Mask 4 register upper half word	C0MASK4H	-	-	R/W	-
0x050	CAN0 Module Control register	C0CTRL	-	-	R/W	-
0x052	CAN0 Module Last Error Code register	C0LEC	R/W	R/W	-	-
0x053	CAN0 Module Information register	C0INFO	R	R	-	-
0x054	CAN0 Module Error Counter	C0ERC	-	-	R/W	-
0x056	CAN0 Module Interrupt Enable register	C0IE	-	-	R/W	-
0x056	CAN0 Module Interrupt Enable register low byte	C0IEL	R/W	R/W	-	-
0x057	CAN0 Module Interrupt Enable register high byte	C0IEH	R/W	R/W	-	-

Table A-1 CAN special function registers (2/3)

Address offset	Register name	Shortcut	1	8	16	32
0x058	CAN0 Module Interrupt Status register	C0INTS	-	-	R/W	-
0x058	CAN0 Module Interrupt Status register low byte	C0INTSL	R/W	R/W	-	-
0x05A	CAN0 Module Bit-Rate Prescaler register	C0BRP	R/W	R/W	-	-
0x05C	CAN0 Bit Rate register	C0BTR	-	-	R/W	-
0x05E	CAN0 Module Last In-Pointer register	C0LIPT	-	R/W	-	-
0x060	CAN0 Module Receive History List Get Pointer register	C0RGPT	-	-	R/W	-
0x060	CAN0 Module Receive History List Get Pointer register low byte	C0RGPTL	R/W	R/W	-	-
0x062	CAN0 Module Last Out-Pointer register	C0LOPT	-	R	-	-
0x064	CAN0 Module Transmit History List Get Pointer register	C0TGPT	-	-	R/W	-
0x064	CAN0 Module Transmit History List Get Pointer register low byte	C0TGPTL	R/W	R/W	-	-
0x066	CAN0 Module Time Stamp register	C0TS	-	-	R/W	-
0x066	CAN0 Module Time Stamp register low byte	C0TSL	R/W	R/W	-	-
0x067	CAN0 Module Time Stamp register high byte	C0TSH	R/W	R/W	-	-
0x100 to 0x4EF	CAN0 Message Buffer registers, see *Table 14-20 on page 457*.					
0x600	CAN1 Global Macro Control register	C1GMCTRL	-	-	R/W	-
0x600	CAN1 Global Macro Control register low byte	C1GMCTRLL	R/W	R/W	-	-
0x601	CAN1 Global Macro Control register high byte	C1GMCTRLH	R/W	R/W	-	-
0x602	CAN1 Global Macro Clock Selection register	C1GMCS	R/W	R/W	-	-
0x606	CAN1 Global Macro Automatic Block Transmission register	C1GMABT	-	-	R/W	-
0x606	CAN1 Global Macro Automatic Block Transmission register low byte	C1GMABTL	R/W	R/W	-	-
0x607	CAN1 Global Macro Automatic Block Transmission register high byte	C1GMABTH	R/W	R/W	-	-
0x608	CAN1 Global Macro Automatic Block Transmission Delay register	C1GMABTD	R/W	R/W	-	-
0x640	CAN1 Module Mask 1 register lower half word	C1MASK1L	-	-	R/W	-
0x642	CAN1 Module Mask 1 register upper half word	C1MASK1H	-	-	R/W	-
0x644	CAN1 Module Mask 2 register lower half word	C1MASK2L	-	-	R/W	-
0x646	CAN1 Module Mask 2 register upper half word	C1MASK2H	-	-	R/W	-
0x648	CAN1 Module Mask 3 register lower half word	C1MASK3L	-	-	R/W	-
0x64A	CAN1 Module Mask 3 register upper half word	C1MASK3H	-	-	R/W	-
0x64C	CAN1 Module Mask 4 register lower half word	C1MASK4L	-	-	R/W	-
0x64E	CAN1 Module Mask 4 register upper half word	C1MASK4H	-	-	R/W	-
0x650	CAN1 Module Control register	C1CTRL	-	-	R/W	-
0x652	CAN1 Module Last Error Code register	C1LEC	R/W	R/W	-	-
0x653	CAN1 Module Information register	C1INFO	R	R	-	-
0x654	CAN1 Module Error Counter	C1ERC	-	-	R/W	-
0x656	CAN1 Module Interrupt Enable register	C1IE	-	-	R/W	-
0x656	CAN1 Module Interrupt Enable register low byte	C1IEL	R/W	R/W	-	-

Table A-1 CAN special function registers (3/3)

Address offset	Register name	Shortcut	1	8	16	32
0x657	CAN1 Module Interrupt Enable register high byte	C1IEH	R/W	R/W	-	-
0x658	CAN1 Module Interrupt Status register	C1INTS	-	-	R/W	-
0x658	CAN1 Module Interrupt Status register low byte	C1INTSL	R/W	R/W	-	-
0x65A	CAN1 Module Bit-Rate Prescaler register	C1BRP	R/W	R/W	-	-
0x65C	CAN1 Bit Rate register	C1BTR	-	-	R/W	-
0x65E	CAN1 Module Last In-Pointer register	C1LIPT	-	R/W	-	-
0x660	CAN1 Module Receive History List Get Pointer register	C1RGPT	-	-	R/W	-
0x660	CAN1 Module Receive History List Get Pointer register low byte	C1RGPTL	R/W	R/W	-	-
0x662	CAN1 Module Last Out-Pointer register	C1LOPT	-	R	-	-
0x664	CAN1 Module Transmit History List Get Pointer register	C1TGPT	-	-	R/W	-
0x664	CAN1 Module Transmit History List Get Pointer register low byte	C1TGPTL	R/W	R/W	-	-
0x666	CAN1 Module Time Stamp register	C1TS	-	-	R/W	-
0x666	CAN1 Module Time Stamp register low byte	C1TSL	R/W	R/W	-	-
0x667	CAN1 Module Time Stamp register high byte	C1TSH	R/W	R/W	-	-
0x700 to 0xAEF	CAN1 Message Buffer registers, see *Table 14-20 on page 457*.					

A.2 FlexRay Registers

The addresses are given as offsets to the FlexRay base address (refer to *"Bus and Memory Control (BCU, MEMC)" on page 225*).

Table A-2 FlexRay-specific registers (1/3)

Address offset	Register name	Shortcut	1	8	16	32
0x0000	Controller Information	FRCI	-	-	-	R
0x0004	Vendor Information	FRVI	-	-	-	R
0x0008	Control Setting	FRCS	-	-	-	R/W
0x0010	Test Register 1	FRTEST1	-	-	-	R/W
0x0014	Test Register 2	FRTEST2	-	-	-	R/W
0x001C	Lock Register	FRLCK	-	-	-	R/W
0x0020	Error Interrupt Register	FREIR	-	-	-	R/W
0x0024	Status Interrupt Register	FRSIR	-	-	-	R/W
0x0028	Error Interrupt Line Select	FREILS	-	-	-	R/W
0x002C	Status Interrupt Line Select	FRSILS	-	-	-	R/W
0x0030	Error Interrupt Enable Set	FREIES	-	-	-	R/W
0x0034	Error Interrupt Enable Reset	FREIER	-	-	-	R/W
0x0038	Status Interrupt Enable Set	FRSIES	-	-	-	R/W
0x003C	Status Interrupt Enable Reset	FRSIER	-	-	-	R/W
0x0040	Interrupt Line Enable	FRILE	-	-	-	R/W
0x0044	Timer 0 Configuration	FRT0C	-	-	-	R/W
0x0048	Timer 1 Configuration	FRT1C	-	-	-	R/W
0x004C	Stop Watch Register	FRSTPW	-	-	-	R/W
0x0080	SUC Configuration Register 1	FRSUCC1	-	-	-	R/W
0x0084	SUC Configuration Register 2	FRSUCC2	-	-	-	R/W
0x0088	SUC Configuration Register 3	FRSUCC3	-	-	-	R/W
0x008C	NEM Configuration Register	FRNEMC	-	-	-	R/W
0x0090	PRT Configuration Register 1	FRPRTC1	-	-	-	R/W
0x0094	PRT Configuration Register 2	FRPRTC2	-	-	-	R/W
0x0098	MHD Configuration Register	FRMHDC	-	-	-	R/W
0x00A0	GTU Configuration Register 1	FRGTUC1	-	-	-	R/W
0x00A4	GTU Configuration Register 2	FRGTUC2	-	-	-	R/W
0x00A8	GTU Configuration Register 3	FRGTUC3	-	-	-	R/W
0x00AC	GTU Configuration Register 4	FRGTUC4	-	-	-	R/W
0x00B0	GTU Configuration Register 5	FRGTUC5	-	-	-	R/W
0x00B4	GTU Configuration Register 6	FRGTUC6	-	-	-	R/W
0x00B8	GTU Configuration Register 7	FRGTUC7	-	-	-	R/W
0x00BC	GTU Configuration Register 8	FRGTUC8	-	-	-	R/W
0x00C0	GTU Configuration Register 9	FRGTUC9	-	-	-	R/W
0x00C4	GTU Configuration Register 10	FRGTUC10	-	-	-	R/W
0x00C8	GTU Configuration Register 11	FRGTUC11	-	-	-	R/W

Table A-2 FlexRay-specific registers (2/3)

Address offset	Register name	Shortcut	1	8	16	32
0x0100	CC Status Vector	FRCCSV	-	-	-	R
0x0104	CC Error Vector	FRCCEV	-	-	-	R
0x0110	Slot Counter Value	FRSCV	-	-	-	R
0x0114	Macrotick and Cycle Counter Value	FRMTCCV	-	-	-	R
0x0118	Rate Correction Value	FRRCV	-	-	-	R
0x011C	Offset Correction Value	FROCV	-	-	-	R
0x0120	Sync Frame Status	FRSFS	-	-	-	R
0x0124	Symbol Window and NIT Status	FRSWNIT	-	-	-	R
0x0128	Aggregated Channel Status	FRACS	-	-	-	R/W
0x0130 to 0x0168	Even Sync ID [1...15]	FRESIDn	-	-	-	R
0x0170 to 0x01A8	Odd Sync ID [1...15]	FROSIDn	-	-	-	R
0x01B0 to 0x01B8	Network Management Vector [1...3]	FRNMVn	-	-	-	R
0x0300	Message RAM Configuration	FRMRC	-	-	-	R/W
0x0304	FIFO Rejection Filter	FRFRF	-	-	-	R/W
0x0308	FIFO Rejection Filter Mask	FRFRFM	-	-	-	R/W
0x0310	Message Handler Status	FRMHDS	-	-	-	R/W
0x0320	Transmission Request 1	FRTXRQ1	-	-	-	R
0x0324	Transmission Request 2	FRTXRQ2	-	-	-	R
0x0328	Transmission Request 3	FRTXRQ3	-	-	-	R
0x032C	Transmission Request 4	FRTXRQ4	-	-	-	R
0x0330	New Data 1	FRNDAT1	-	-	-	R
0x0334	New Data 2	FRNDAT2	-	-	-	R
0x0338	New Data 3	FRNDAT3	-	-	-	R
0x033C	New Data 4	FRNDAT4	-	-	-	R
0x0340	Message Buffer Status Changed 1	FRMBSC1	-	-	-	R
0x0344	Message Buffer Status Changed 2	FRMBSC2	-	-	-	R
0x0348	Message Buffer Status Changed 3	FRMBSC3	-	-	-	R
0x034C	Message Buffer Status Changed 4	FRMBSC4	-	-	-	R
0x0400 to 0x04FC	Write Data Section [1...64]	FRWRDSn	-	-	-	R/W
0x0500	Write Header Section 1	FRWRHS1	-	-	-	R/W
0x0504	Write Header Section 2	FRWRHS2	-	-	-	R/W
0x0508	Write Header Section 3	FRWRHS3	-	-	-	R/W
0x0510	Input Buffer Command Mask	FRIBCM	-	-	-	R/W
0x0514	Input Buffer Command Request	FRIBCR	-	-	-	R/W
0x0600 to 0x06FC	Read Data Section [1...64]	FRRDDSn	-	-	-	R
0x0700	Read Header Section 1	FRRDHS1	-	-	-	R
0x0704	Read Header Section 2	FRRDHS2	-	-	-	R
0x0708	Read Header Section 3	FRRDHS3	-	-	-	R
0x070C	Message Buffer Status	FRMBS	-	-	-	R

Table A-2 FlexRay-specific registers (3/3)

Address offset	Register name	Shortcut	1	8	16	32
0x0710	Output Buffer Command Mask	FROBCM	-	-	-	R/W
0x0714	Output Buffer Command Request	FROBCR	-	-	-	R/W
0x0800	ECC Control Register for E-Ray Message RAM	FRECCTL	-	-	-	R/W

A.3 Other Special Function Registers

Table A-3 Other special function registers (1/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF000	PortAL	PAL	-	-	R/W	-
0xFFFFF000	PortAL low byte	PALL	R/W	R/W	-	-
0xFFFFF001	PortAL high byte	PALH	R/W	R/W	-	-
0xFFFFF002	PortAH	PAH	R/W	R/W	-	-
0xFFFFF004	PortDL	PDL	-	-	R/W	-
0xFFFFF004	PortDL low byte	PDLL	R/W	R/W	-	-
0xFFFFF005	PortDL high byte	PDLH	R/W	R/W	-	-
0xFFFFF006	PortDH	PDH	-	-	R/W	-
0xFFFFF006	PortDH low byte	PDHL	R/W	R/W	-	-
0xFFFFF007	PortDH high byte	PDHH	R/W	R/W	-	-
0xFFFFF008	PortCS	PCS	R/W	R/W	-	-
0xFFFFF00A	PortCT	PCT	R/W	R/W	-	-
0xFFFFF00C	PortCM	PCM	R/W	R/W	-	-
0xFFFFF00E	PortCD	PCD	R/W	R/W	-	-
0xFFFFF020	PortAL mode	PMAL	-	-	R/W	-
0xFFFFF020	PortAL mode low byte	PMALL	R/W	R/W	-	-
0xFFFFF021	PortAL mode high byte	PMALH	R/W	R/W	-	-
0xFFFFF022	PortAH mode	PMAH	R/W	R/W	-	-
0xFFFFF024	PortDL mode	PMDL	-	-	R/W	-
0xFFFFF024	PortDL mode low byte	PMDLL	R/W	R/W	-	-
0xFFFFF025	PortDL mode high byte	PMDLH	R/W	R/W	-	-
0xFFFFF026	PortDH mode	PMDH	-	-	R/W	-
0xFFFFF026	PortDH mode low byte	PMDHL	R/W	R/W	-	-
0xFFFFF027	PortDH mode high byte	PMDHH	R/W	R/W	-	-
0xFFFFF028	PortCS mode	PMCS	R/W	R/W	-	-
0xFFFFF02A	PortCT mode	PMCT	R/W	R/W	-	-
0xFFFFF02C	PortCM mode	PMCM	R/W	R/W	-	-
0xFFFFF02E	PortCD mode	PMCD	R/W	R/W	-	-
0xFFFFF040	PortAL mode control	PMCAL	-	-	R/W	-
0xFFFFF040	PortAL mode control low byte	PMCALL	R/W	R/W	-	-
0xFFFFF041	PortAL mode control high byte	PMCALH	R/W	R/W	-	-
0xFFFFF042	PortAH mode control	PMCAH	R/W	R/W	-	-
0xFFFFF044	PortDL mode control	PMCDL	-	-	R/W	-
0xFFFFF044	PortDL mode control low byte	PMCDLL	R/W	R/W	-	-
0xFFFFF045	PortDL mode control high byte	PMCDLH	R/W	R/W	-	-
0xFFFFF046	PortDH mode control	PMCDH	-	-	R/W	-
0xFFFFF046	PortDH mode control low byte	PMCDHL	R/W	R/W	-	-
0xFFFFF047	PortDH mode control high byte	PMCDHH	R/W	R/W	-	-
0xFFFFF048	PortCS mode control	PMCCS	R/W	R/W	-	-

Table A-3 Other special function registers (2/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF04A	PortCT mode control	PMCCT	R/W	R/W	-	-
0xFFFFF04C	PortCM mode control	PMCCM	R/W	R/W	-	-
0xFFFFF04E	PortCD mode control	PMCCD	R/W	R/W	-	-
0xFFFFF060	Chip select area control register 0	CSC0	-	-	R/W	-
0xFFFFF062	Chip select area control register 1	CSC1	-	-	R/W	-
0xFFFFF064	Peripheral area selection control register	BPC	-	-	R/W	-
0xFFFFF066	Bus size configuration register	BSC	-	-	R/W	-
0xFFFFF068	Endian configuration register	BEC	-	-	R/W	-
0xFFFFF06E	Internal peripheral function wait control register	VSWC	R/W	R/W	-	-
0xFFFFF100	Interrupt mask control register 0	IMR0	-	-	R/W	-
0xFFFFF100	Interrupt mask control register 0L	IMR0L	R/W	R/W	-	-
0xFFFFF101	Interrupt mask control register 0H	IMR0H	R/W	R/W	-	-
0xFFFFF102	Interrupt mask control register 1	IMR1	-	-	R/W	-
0xFFFFF102	Interrupt mask control register 1L	IMR1L	R/W	R/W	-	-
0xFFFFF103	Interrupt mask control register 1H	IMR1H	R/W	R/W	-	-
0xFFFFF104	Interrupt mask control register 2	IMR2	-	-	R/W	-
0xFFFFF104	Interrupt mask control register 2L	IMR2L	R/W	R/W	-	-
0xFFFFF105	Interrupt mask control register 2H	IMR2H	R/W	R/W	-	-
0xFFFFF106	Interrupt mask control register 3	IMR3	-	-	R/W	-
0xFFFFF106	Interrupt mask control register 3L	IMR3L	R/W	R/W	-	-
0xFFFFF107	Interrupt mask control register 3H	IMR3H	R/W	R/W	-	-
0xFFFFF108	Interrupt mask control register 4	IMR4	-	-	R/W	-
0xFFFFF108	Interrupt mask control register 4L	IMR4L	R/W	R/W	-	-
0xFFFFF109	Interrupt mask control register 4H	IMR4H	R/W	R/W	-	-
0xFFFFF10A	Interrupt mask control register 5	IMR5	-	-	R/W	-
0xFFFFF10A	Interrupt mask control register 5L	IMR5L	R/W	R/W	-	-
0xFFFFF10B	Interrupt mask control register 5H	IMR5H	R/W	R/W	-	-
0xFFFFF10C	Interrupt mask control register 6	IMR6	-	-	R/W	-
0xFFFFF10C	Interrupt mask control register 6L	IMR6L	R/W	R/W	-	-
0xFFFFF10D	Interrupt mask control register 6H	IMR6H	R/W	R/W	-	-
0xFFFFF10E	Interrupt mask control register 7	IMR7	-	-	R/W	-
0xFFFFF10E	Interrupt mask control register 7L	IMR7L	R/W	R/W	-	-
0xFFFFF10F	Interrupt mask control register 7H	IMR7H	R/W	R/W	-	-
0xFFFFF110	Interrupt control register	ERRIC	R/W	R/W	-	-
0xFFFFF112	Interrupt control register	OSDIC	R/W	R/W	-	-
0xFFFFF114	Interrupt control register	PIC0	R/W	R/W	-	-
0xFFFFF116	Interrupt control register	PIC1	R/W	R/W	-	-
0xFFFFF118	Interrupt control register	PIC2	R/W	R/W	-	-
0xFFFFF11A	Interrupt control register	PIC3	R/W	R/W	-	-
0xFFFFF11C	Interrupt control register	PIC4	R/W	R/W	-	-
0xFFFFF11E	Interrupt control register	PIC5	R/W	R/W	-	-

Table A-3 Other special function registers (3/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF120	Interrupt control register	PIC6	R/W	R/W	-	-
0xFFFFF122	Interrupt control register	PIC7	R/W	R/W	-	-
0xFFFFF124	Interrupt control register	PIC8	R/W	R/W	-	-
0xFFFFF126	Interrupt control register	PIC9	R/W	R/W	-	-
0xFFFFF128	Interrupt control register	PIC10	R/W	R/W	-	-
0xFFFFF12A	Interrupt control register	PIC11	R/W	R/W	-	-
0xFFFFF12C	Interrupt control register	PIC12	R/W	R/W	-	-
0xFFFFF12E	Interrupt control register	TS0OVIC	R/W	R/W	-	-
0xFFFFF130	Interrupt control register	TS0CCIC0	R/W	R/W	-	-
0xFFFFF132	Interrupt control register	TS0CCIC1	R/W	R/W	-	-
0xFFFFF134	Interrupt control register	TS0CCIC2	R/W	R/W	-	-
0xFFFFF136	Interrupt control register	TS0CCIC3	R/W	R/W	-	-
0xFFFFF138	Interrupt control register	TS0CCIC4	R/W	R/W	-	-
0xFFFFF13A	Interrupt control register	TS0CCIC5	R/W	R/W	-	-
0xFFFFF13C	Interrupt control register	TS0CDIC0	R/W	R/W	-	-
0xFFFFF13E	Interrupt control register	TS0ODIC	R/W	R/W	-	-
0xFFFFF140	Interrupt control register	TS0ERIC	R/W	R/W	-	-
0xFFFFF142	Interrupt control register	TS0WNIC	R/W	R/W	-	-
0xFFFFF144	Interrupt control register	TS1OVIC	R/W	R/W	-	-
0xFFFFF146	Interrupt control register	TS1CCIC0	R/W	R/W	-	-
0xFFFFF148	Interrupt control register	TS1CCIC1	R/W	R/W	-	-
0xFFFFF14A	Interrupt control register	TS1CCIC2	R/W	R/W	-	-
0xFFFFF14C	Interrupt control register	TS1CCIC3	R/W	R/W	-	-
0xFFFFF14E	Interrupt control register	TS1CCIC4	R/W	R/W	-	-
0xFFFFF150	Interrupt control register	TS1CCIC5	R/W	R/W	-	-
0xFFFFF152	Interrupt control register	TS1CDIC0	R/W	R/W	-	-
0xFFFFF154	Interrupt control register	TS1ODIC	R/W	R/W	-	-
0xFFFFF156	Interrupt control register	TS1ERIC	R/W	R/W	-	-
0xFFFFF158	Interrupt control register	TS1WNIC	R/W	R/W	-	-
0xFFFFF15A	Interrupt control register	TT0OVIC	R/W	R/W	-	-
0xFFFFF15C	Interrupt control register	TT0CCIC0	R/W	R/W	-	-
0xFFFFF15E	Interrupt control register	TT0CCIC1	R/W	R/W	-	-
0xFFFFF160	Interrupt control register	TT0ECIC	R/W	R/W	-	-
0xFFFFF162	Interrupt control register	TT1OVIC	R/W	R/W	-	-
0xFFFFF164	Interrupt control register	TT1CCIC0	R/W	R/W	-	-
0xFFFFF166	Interrupt control register	TT1CCIC1	R/W	R/W	-	-
0xFFFFF168	Interrupt control register	TT1ECIC	R/W	R/W	-	-
0xFFFFF16A	Interrupt control register	TAA0OVIC	R/W	R/W	-	-
0xFFFFF16C	Interrupt control register	TAA0CCIC0	R/W	R/W	-	-
0xFFFFF16E	Interrupt control register	TAA0CCIC1	R/W	R/W	-	-
0xFFFFF170	Interrupt control register	TAA1OVIC	R/W	R/W	-	-

Table A-3 Other special function registers (4/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF172	Interrupt control register	TAA1CCIC0	R/W	R/W	-	-
0xFFFFF174	Interrupt control register	TAA1CCIC1	R/W	R/W	-	-
0xFFFFF176	Interrupt control register	TAA2OVIC	R/W	R/W	-	-
0xFFFFF178	Interrupt control register	TAA2CCIC0	R/W	R/W	-	-
0xFFFFF17A	Interrupt control register	TAA2CCIC1	R/W	R/W	-	-
0xFFFFF17C	Interrupt control register	TAA3OVIC	R/W	R/W	-	-
0xFFFFF17E	Interrupt control register	TAA3CCIC0	R/W	R/W	-	-
0xFFFFF180	Interrupt control register	TAA3CCIC1	R/W	R/W	-	-
0xFFFFF182	Interrupt control register	TAA4OVIC	R/W	R/W	-	-
0xFFFFF184	Interrupt control register	TAA4CCIC0	R/W	R/W	-	-
0xFFFFF186	Interrupt control register	TAA4CCIC1	R/W	R/W	-	-
0xFFFFF188	Interrupt control register	TAA5OVIC	R/W	R/W	-	-
0xFFFFF18A	Interrupt control register	TAA5CCIC0	R/W	R/W	-	-
0xFFFFF18C	Interrupt control register	TAA5CCIC1	R/W	R/W	-	-
0xFFFFF18E	Interrupt control register	TAA6OVIC	R/W	R/W	-	-
0xFFFFF190	Interrupt control register	TAA6CCIC0	R/W	R/W	-	-
0xFFFFF192	Interrupt control register	TAA6CCIC1	R/W	R/W	-	-
0xFFFFF194	Interrupt control register	TAA7OVIC	R/W	R/W	-	-
0xFFFFF196	Interrupt control register	TAA7CCIC0	R/W	R/W	-	-
0xFFFFF198	Interrupt control register	TAA7CCIC1	R/W	R/W	-	-
0xFFFFF19A	Interrupt control register	TAA8OVIC	R/W	R/W	-	-
0xFFFFF19C	Interrupt control register	TAA8CCIC0	R/W	R/W	-	-
0xFFFFF19E	Interrupt control register	TAA8CCIC1	R/W	R/W	-	-
0xFFFFF1A0	Interrupt control register	TAA9OVIC	R/W	R/W	-	-
0xFFFFF1A2	Interrupt control register	TAA9CCIC0	R/W	R/W	-	-
0xFFFFF1A4	Interrupt control register	TAA9CCIC1	R/W	R/W	-	-
0xFFFFF1A6	Interrupt control register	BRGIC	R/W	R/W	-	-
0xFFFFF1A8	Interrupt control register	BRGIC2	R/W	R/W	-	-
0xFFFFF1AA	Interrupt control register	FRIC0	R/W	R/W	-	-
0xFFFFF1AC	Interrupt control register	FRIC1	R/W	R/W	-	-
0xFFFFF1AE	Interrupt control register	FRIC2	R/W	R/W	-	-
0xFFFFF1B0	Interrupt control register	FRIC3	R/W	R/W	-	-
0xFFFFF1B2	Interrupt control register	FRIBIC	R/W	R/W	-	-
0xFFFFF1B4	Interrupt control register	FROBIC	R/W	R/W	-	-
0xFFFFF1B6	Interrupt control register	C0ERRIC	R/W	R/W	-	-
0xFFFFF1B8	Interrupt control register	C0WUPIC	R/W	R/W	-	-
0xFFFFF1BA	Interrupt control register	C0RECIC	R/W	R/W	-	-
0xFFFFF1BC	Interrupt control register	C0TRXIC	R/W	R/W	-	-
0xFFFFF1BE	Interrupt control register	C1ERRIC	R/W	R/W	-	-
0xFFFFF1C0	Interrupt control register	C1WUPIC	R/W	R/W	-	-
0xFFFFF1C2	Interrupt control register	C1RECIC	R/W	R/W	-	-

Table A-3 Other special function registers (5/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF1C4	Interrupt control register	C1TRXIC	R/W	R/W	-	-
0xFFFFF1D6	Interrupt control register	CB0TIC	R/W	R/W	-	-
0xFFFFF1D8	Interrupt control register	CB0RIC	R/W	R/W	-	-
0xFFFFF1DA	Interrupt control register	CB0REIC	R/W	R/W	-	-
0xFFFFF1DC	Interrupt control register	CB1TIC	R/W	R/W	-	-
0xFFFFF1DE	Interrupt control register	CB1RIC	R/W	R/W	-	-
0xFFFFF1E0	Interrupt control register	CB1REIC	R/W	R/W	-	-
0xFFFFF1E2	Interrupt control register	CE0OFIC	R/W	R/W	-	-
0xFFFFF1E4	Interrupt control register	CE0CIC	R/W	R/W	-	-
0xFFFFF1E6	Interrupt control register	CE1OFIC	R/W	R/W	-	-
0xFFFFF1E8	Interrupt control register	CE1CIC	R/W	R/W	-	-
0xFFFFF1EA	Interrupt control register	UC0REIC	R/W	R/W	-	-
0xFFFFF1EC	Interrupt control register	UC0TIC	R/W	R/W	-	-
0xFFFFF1EE	Interrupt control register	UC1REIC	R/W	R/W	-	-
0xFFFFF1F0	Interrupt control register	UC1TIC	R/W	R/W	-	-
0xFFFFF1F2	Interrupt control register	ADIC0	R/W	R/W	-	-
0xFFFFF1F4	Interrupt control register	ADIC1	R/W	R/W	-	-
0xFFFFF1F6	Interrupt control register	DMAIC2	R/W	R/W	-	-
0xFFFFF1F8	Interrupt control register	DMAIC3	R/W	R/W	-	-
0xFFFFF1FA	In-service priority register	ISPR	R	R	-	-
0xFFFFF1FC	Command register	PRCMD	-	W	-	-
0xFFFFF200	A/D Converter 0 mode register 0	ADM00	R/W	R/W	-	-
0xFFFFF201	A/D Converter 0 mode register 1	ADM01	R/W	R/W	-	-
0xFFFFF202	A/D Converter 0 mode register 2	ADM02	R/W	R/W	-	-
0xFFFFF210	A/D conversion result register 00	ADCR00	-	-	R	-
0xFFFFF211	A/D conversion result register 00H	ADCR00H	-	R	-	-
0xFFFFF212	A/D conversion result register 01	ADCR01	-	-	R	-
0xFFFFF213	A/D conversion result register 01H	ADCR01H	-	R	-	-
0xFFFFF214	A/D conversion result register 02	ADCR02	-	-	R	-
0xFFFFF215	A/D conversion result register 02H	ADCR02H	-	R	-	-
0xFFFFF216	A/D conversion result register 03	ADCR03	-	-	R	-
0xFFFFF217	A/D conversion result register 03H	ADCR03H	-	R	-	-
0xFFFFF218	A/D conversion result register 04	ADCR04	-	-	R	-
0xFFFFF219	A/D conversion result register 04H	ADCR04H	-	R	-	-
0xFFFFF21A	A/D conversion result register 05	ADCR05	-	-	R	-
0xFFFFF21B	A/D conversion result register 05H	ADCR05H	-	R	-	-
0xFFFFF21C	A/D conversion result register 06	ADCR06	-	-	R	-
0xFFFFF21D	A/D conversion result register 06H	ADCR06H	-	R	-	-
0xFFFFF21E	A/D conversion result register 07	ADCR07	-	-	R	-
0xFFFFF21F	A/D conversion result register 07H	ADCR07H	-	R	-	-
0xFFFFF220	A/D conversion result register 08	ADCR08	-	-	R	-

Table A-3 Other special function registers (6/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF221	A/D conversion result register 08H	ADCR08H	-	R	-	-
0xFFFFF222	A/D conversion result register 09	ADCR09	-	-	R	-
0xFFFFF223	A/D conversion result register 09H	ADCR09H	-	R	-	-
0xFFFFF22E	A/D conversion result register 0 for DMA	ADDMA0	-	-	R	-
0xFFFFF240	A/D Converter 1 mode register 0	ADM10	R/W	R/W	-	-
0xFFFFF241	A/D Converter 1 mode register 1	ADM11	R/W	R/W	-	-
0xFFFFF242	A/D Converter 1 mode register 2	ADM12	R/W	R/W	-	-
0xFFFFF250	A/D conversion result register 10	ADCR10	-	-	R	-
0xFFFFF251	A/D conversion result register 10H	ADCR10H	-	R	-	-
0xFFFFF252	A/D conversion result register 11	ADCR11	-	-	R	-
0xFFFFF253	A/D conversion result register 11H	ADCR11H	-	R	-	-
0xFFFFF254	A/D conversion result register 12	ADCR12	-	-	R	-
0xFFFFF255	A/D conversion result register 12H	ADCR12H	-	R	-	-
0xFFFFF256	A/D conversion result register 13	ADCR13	-	-	R	-
0xFFFFF257	A/D conversion result register 13H	ADCR13H	-	R	-	-
0xFFFFF258	A/D conversion result register 14	ADCR14	-	-	R	-
0xFFFFF259	A/D conversion result register 14H	ADCR14H	-	R	-	-
0xFFFFF25A	A/D conversion result register 15	ADCR15	-	-	R	-
0xFFFFF25B	A/D conversion result register 15H	ADCR15H	-	R	-	-
0xFFFFF25C	A/D conversion result register 16	ADCR16	-	-	R	-
0xFFFFF25D	A/D conversion result register 16H	ADCR16H	-	R	-	-
0xFFFFF25E	A/D conversion result register 17	ADCR17	-	-	R	-
0xFFFFF25F	A/D conversion result register 17H	ADCR17H	-	R	-	-
0xFFFFF260	A/D conversion result register 18	ADCR18	-	-	R	-
0xFFFFF261	A/D conversion result register 18H	ADCR18H	-	R	-	-
0xFFFFF262	A/D conversion result register 19	ADCR19	-	-	R	-
0xFFFFF263	A/D conversion result register 19H	ADCR19H	-	R	-	-
0xFFFFF26E	A/D conversion result register 1 for DMA	ADDMA1	-	-	R	-
0xFFFFF270	A/D Converter 0 trigger source select register	ADTRSEL0	R/W	R/W	-	-
0xFFFFF272	A/D Converter 1 trigger source select register	ADTRSEL1	R/W	R/W	-	-
0xFFFFF300	DMA transfer memory start address register 0	MAR0	-	-	R/W	-
0xFFFFF302	DMA transfer memory start address register 1	MAR1	-	-	R/W	-
0xFFFFF304	DMA transfer memory start address register 2	MAR2	-	-	R/W	-
0xFFFFF306	DMA transfer memory start address register 3	MAR3	-	-	R/W	-
0xFFFFF308	DMA transfer memory start address register 4	MAR4	-	-	R/W	-
0xFFFFF30A	DMA transfer memory start address register 5	MAR5	-	-	R/W	-
0xFFFFF30C	DMA transfer memory start address register 6	MAR6	-	-	R/W	-
0xFFFFF30E	DMA transfer memory start address register 7	MAR7	-	-	R/W	-
0xFFFFF324	DMA transfer SFR start address register 2	SAR2	-	R/W	-	-
0xFFFFF326	DMA transfer SFR start address register 3	SAR3	-	R/W	-	-
0xFFFFF340	DMA transfer count register 0	DTCR0	-	R/W	-	-

Table A-3 Other special function registers (7/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF342	DMA transfer count register 1	DTCR1	-	R/W	-	-
0xFFFFF344	DMA transfer count register 2	DTCR2	-	R/W	-	-
0xFFFFF346	DMA transfer count register 3	DTCR3	-	R/W	-	-
0xFFFFF348	DMA transfer count register 4	DTCR4	-	R/W	-	-
0xFFFFF34A	DMA transfer count register 5	DTCR5	-	R/W	-	-
0xFFFFF34C	DMA transfer count register 6	DTCR6	-	R/W	-	-
0xFFFFF34E	DMA transfer count register 7	DTCR7	-	R/W	-	-
0xFFFFF360	DMA mode control register	DMAMCL	R/W	R/W	-	-
0xFFFFF362	DMA status register	DMASL	R/W	R/W	-	-
0xFFFFF364	DMA data size control register	DMADSCL	R/W	R/W	-	-
0xFFFFF388	DMA trigger factor register 4	DTFR4	-	R/W	-	-
0xFFFFF38A	DMA trigger factor register 5	DTFR5	-	R/W	-	-
0xFFFFF38C	DMA trigger factor register 6	DTFR6	-	R/W	-	-
0xFFFFF38E	DMA trigger factor register 7	DTFR7	-	R/W	-	-
0xFFFFF400	Port0	P0	R	R	-	-
0xFFFFF402	Port1	P1	R/W	R/W	-	-
0xFFFFF404	Port2	P2	R/W	R/W	-	-
0xFFFFF406	Port3	P3	R/W	R/W	-	-
0xFFFFF408	Port4	P4	R/W	R/W	-	-
0xFFFFF40A	Port5	P5	R/W	R/W	-	-
0xFFFFF40C	Port6	P6	R/W	R/W	-	-
0xFFFFF40E	Port7	P7	R/W	R/W	-	-
0xFFFFF410	Port8	P8	R/W	R/W	-	-
0xFFFFF412	Port9	P9	R/W	R/W	-	-
0xFFFFF414	Port10	P10	R/W	R/W	-	-
0xFFFFF416	Port11	P11	R/W	R/W	-	-
0xFFFFF422	Port1 mode	PM1	R/W	R/W	-	-
0xFFFFF424	Port2 mode	PM2	R/W	R/W	-	-
0xFFFFF426	Port3 mode	PM3	R/W	R/W	-	-
0xFFFFF428	Port4 mode	PM4	R/W	R/W	-	-
0xFFFFF42A	Port5 mode	PM5	R/W	R/W	-	-
0xFFFFF42C	Port6 mode	PM6	R/W	R/W	-	-
0xFFFFF42E	Port7 mode	PM7	R/W	R/W	-	-
0xFFFFF430	Port8 mode	PM8	R/W	R/W	-	-
0xFFFFF432	Port9 mode	PM9	R/W	R/W	-	-
0xFFFFF434	Port10 mode	PM10	R/W	R/W	-	-
0xFFFFF436	Port11 mode	PM11	R/W	R/W	-	-
0xFFFFF442	Port1 mode control	PMC1	R/W	R/W	-	-
0xFFFFF444	Port2 mode control	PMC2	R/W	R/W	-	-
0xFFFFF446	Port3 mode control	PMC3	R/W	R/W	-	-
0xFFFFF448	Port4 mode control	PMC4	R/W	R/W	-	-

Table A-3 Other special function registers (8/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF44A	Port5 mode control	PMC5	R/W	R/W	-	-
0xFFFFF44C	Port6 mode control	PMC6	R/W	R/W	-	-
0xFFFFF44E	Port7 mode control	PMC7	R/W	R/W	-	-
0xFFFFF450	Port8 mode control	PMC8	R/W	R/W	-	-
0xFFFFF452	Port9 mode control	PMC9	R/W	R/W	-	-
0xFFFFF454	Port10 mode control	PMC10	R/W	R/W	-	-
0xFFFFF456	Port11 mode control	PMC11	R/W	R/W	-	-
0xFFFFF462	Port1 function control register	PFC1	R/W	R/W	-	-
0xFFFFF464	Port2 function control register	PFC2	R/W	R/W	-	-
0xFFFFF468	Port4 function control register	PFC4	R/W	R/W	-	-
0xFFFFF46E	Port7 function control register	PFC7	R/W	R/W	-	-
0xFFFFF470	Port8 function control register	PFC8	R/W	R/W	-	-
0xFFFFF472	Port9 function control register	PFC9	R/W	R/W	-	-
0xFFFFF474	Port10 function control register	PFC10	R/W	R/W	-	-
0xFFFFF480	Bus cycle configuration register 0	BCT0	-	-	R/W	-
0xFFFFF482	Bus cycle configuration register 1	BCT1	-	-	R/W	-
0xFFFFF484	Data wait control register 0	DWC0	-	-	R/W	-
0xFFFFF486	Data wait control register 1	DWC1	-	-	R/W	-
0xFFFFF488	Address wait control register	AWC	-	-	R/W	-
0xFFFFF48A	Bus cycle control register	BCC	-	-	R/W	-
0xFFFFF48E	Bus clock dividing frequency control register	DVC	-	R/W	-	-
0xFFFFF580	TMS0 control register 0	TS0CTL0	R/W	R/W	-	-
0xFFFFF581	TMS0 control register 1	TS0CTL1	R/W	R/W	-	-
0xFFFFF582	TMS0 I/O control register 0	TS0IOC0	-	-	R/W	-
0xFFFFF582	TMS0 I/O control register 0L	TS0IOC0L	R/W	R/W	-	-
0xFFFFF583	TMS0 I/O control register 0H	TS0IOC0H	R/W	R/W	-	-
0xFFFFF584	TMS0 I/O control register 0	TS0IOC2	R/W	R/W	-	-
0xFFFFF585	TMS0 I/O control register 0	TS0IOC4	R/W	R/W	-	-
0xFFFFF586	TMS0 option register 0	TS0OPT0	R/W	R/W	-	-
0xFFFFF587	TMS0 option register 7	TS0OPT7	R/W	R/W	-	-
0xFFFFF588	TMS0 option register 4	TS0OPT4	R/W	R/W	-	-
0xFFFFF589	TMS0 option register 5	TS0OPT5	R/W	R/W	-	-
0xFFFFF58A	TMS0 option register 1	TS0OPT1	-	-	R/W	-
0xFFFFF58A	TMS0 option register 1L	TS0OPT1L	R/W	R/W	-	-
0xFFFFF58B	TMS0 option register 1H	TS0OPT1H	R/W	R/W	-	-
0xFFFFF58C	TMS0 dead time setting register 1	TS0DTC1	-	-	R/W	-
0xFFFFF58E	TMS0 dead time setting register 0	TS0DTC0	-	-	R/W	-
0xFFFFF590	TMS0 compare register 5	TS0CCR5	-	-	R/W	-
0xFFFFF592	TMS0 compare register 4	TS0CCR4	-	-	R/W	-
0xFFFFF594	TMS0 pattern register 1	TS0PAT1	-	-	R/W	-
0xFFFFF596	TMS0 pattern register 0	TS0PAT0	-	-	R/W	-

Table A-3 Other special function registers (9/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF598	TMS0 compare register 0	TS0CCR0	-	-	R/W	-
0xFFFFF59A	TMS0 compare register 3	TS0CCR3	-	-	R/W	-
0xFFFFF59C	TMS0 compare register 2	TS0CCR2	-	-	R/W	-
0xFFFFF59E	TMS0 compare register 1	TS0CCR1	-	-	R/W	-
0xFFFFF5A0	TMS0 option register 2	TS0OPT2	-	-	R/W	-
0xFFFFF5A0	TMS0 option register 2L	TS0OPT2L	R/W	R/W	-	-
0xFFFFF5A1	TMS0 option register 2H	TS0OPT2H	R/W	R/W	-	-
0xFFFFF5A2	TMS0 option register 3	TS0OPT3	-	-	R/W	-
0xFFFFF5A2	TMS0 option register 3L	TS0OPT3L	R/W	R/W	-	-
0xFFFFF5A3	TMS0 option register 3H	TS0OPT3H	R/W	R/W	-	-
0xFFFFF5A4	TMS0 option register 6	TS0OPT6	-	-	R/W	-
0xFFFFF5A4	TMS0 option register 6L	TS0OPT6L	R/W	R/W	-	-
0xFFFFF5A5	TMS0 option register 6H	TS0OPT6H	R/W	R/W	-	-
0xFFFFF5A6	TMS0 counter read register	TS0CNT	-	-	R	-
0xFFFFF5A8	TMS0 sub-counter read register	TS0SBC	-	-	R	-
0xFFFFF5C0	TMS1 control register 0	TS1CTL0	R/W	R/W	-	-
0xFFFFF5C1	TMS1 control register 1	TS1CTL1	R/W	R/W	-	-
0xFFFFF5C2	TMS1 I/O control register 0	TS1IOC0	-	-	R/W	-
0xFFFFF5C2	TMS1 I/O control register 0L	TS1IOC0L	R/W	R/W	-	-
0xFFFFF5C3	TMS1 I/O control register 0H	TS1IOC0H	R/W	R/W	-	-
0xFFFFF5C4	TMS1 I/O control register 0	TS1IOC2	R/W	R/W	-	-
0xFFFFF5C5	TMS1 I/O control register 0	TS1IOC4	R/W	R/W	-	-
0xFFFFF5C6	TMS1 option register 0	TS1OPT0	R/W	R/W	-	-
0xFFFFF5C7	TMS1 option register 7	TS1OPT7	R/W	R/W	-	-
0xFFFFF5C8	TMS1 option register 4	TS1OPT4	R/W	R/W	-	-
0xFFFFF5C9	TMS1 option register 5	TS1OPT5	R/W	R/W	-	-
0xFFFFF5CA	TMS1 option register 1	TS1OPT1	-	-	R/W	-
0xFFFFF5CA	TMS1 option register 1L	TS1OPT1L	R/W	R/W	-	-
0xFFFFF5CB	TMS1 option register 1H	TS1OPT1H	R/W	R/W	-	-
0xFFFFF5CC	TMS1 dead time setting register 1	TS1DTC1	-	-	R/W	-
0xFFFFF5CE	TMS1 dead time setting register 0	TS1DTC0	-	-	R/W	-
0xFFFFF5D0	TMS1 compare register 5	TS1CCR5	-	-	R/W	-
0xFFFFF5D2	TMS1 compare register 4	TS1CCR4	-	-	R/W	-
0xFFFFF5D4	TMS1 pattern register 1	TS1PAT1	-	-	R/W	-
0xFFFFF5D6	TMS1 pattern register 0	TS1PAT0	-	-	R/W	-
0xFFFFF5D8	TMS1 compare register 0	TS1CCR0	-	-	R/W	-
0xFFFFF5DA	TMS1 compare register 3	TS1CCR3	-	-	R/W	-
0xFFFFF5DC	TMS1 compare register 2	TS1CCR2	-	-	R/W	-
0xFFFFF5DE	TMS1 compare register 1	TS1CCR1	-	-	R/W	-
0xFFFFF5E0	TMS1 option register 2	TS1OPT2	-	-	R/W	-
0xFFFFF5E0	TMS1 option register 2L	TS1OPT2L	R/W	R/W	-	-

Table A-3 Other special function registers (10/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF5E1	TMS1 option register 2H	TS1OPT2H	R/W	R/W	-	-
0xFFFFF5E2	TMS1 option register 3	TS1OPT3	-	-	R/W	-
0xFFFFF5E2	TMS1 option register 3L	TS1OPT3L	R/W	R/W	-	-
0xFFFFF5E3	TMS1 option register 3H	TS1OPT3H	R/W	R/W	-	-
0xFFFFF5E4	TMS1 option register 6	TS1OPT6	-	-	R/W	-
0xFFFFF5E4	TMS1 option register 6L	TS1OPT6L	R/W	R/W	-	-
0xFFFFF5E5	TMS1 option register 6H	TS1OPT6H	R/W	R/W	-	-
0xFFFFF5E6	TMS1 counter read register	TS1CNT	-	-	R	-
0xFFFFF5E8	TMS1 sub-counter read register	TS1SBC	-	-	R	-
0xFFFFF600	TAA0 control register 0	TAA0CTL0	R/W	R/W	-	-
0xFFFFF601	TAA0 control register 1	TAA0CTL1	R/W	R/W	-	-
0xFFFFF602	TAA0 I/O control register 0	TAA0IOC0	R/W	R/W	-	-
0xFFFFF603	TAA0 I/O control register 1	TAA0IOC1	R/W	R/W	-	-
0xFFFFF604	TAA0 I/O control register 2	TAA0IOC2	R/W	R/W	-	-
0xFFFFF605	TAA0 option register 0	TAA0OPT0	R/W	R/W	-	-
0xFFFFF606	TAA0 capture/compare register 0	TAA0CCR0	-	-	R/W	-
0xFFFFF608	TAA0 capture/compare register 1	TAA0CCR1	-	-	R/W	-
0xFFFFF60A	TAA0 counter read buffer register	TAA0CNT	-	-	R	-
0xFFFFF60C	TAA0 I/O control register 4	TAA0IOC4	R/W	R/W	-	-
0xFFFFF60E	TAA0 specific overflow value control register	TAA0SVC	R/W	R/W	-	-
0xFFFFF610	TAA1 control register 0	TAA1CTL0	R/W	R/W	-	-
0xFFFFF611	TAA1 control register 1	TAA1CTL1	R/W	R/W	-	-
0xFFFFF612	TAA1 I/O control register 0	TAA1IOC0	R/W	R/W	-	-
0xFFFFF613	TAA1 I/O control register 1	TAA1IOC1	R/W	R/W	-	-
0xFFFFF614	TAA1 I/O control register 2	TAA1IOC2	R/W	R/W	-	-
0xFFFFF615	TAA1 option register 0	TAA1OPT0	R/W	R/W	-	-
0xFFFFF616	TAA1 capture/compare register 0	TAA1CCR0	-	-	R/W	-
0xFFFFF618	TAA1 capture/compare register 1	TAA1CCR1	-	-	R/W	-
0xFFFFF61A	TAA1 counter read buffer register	TAA1CNT	-	-	R	-
0xFFFFF61C	TAA1 I/O control register 4	TAA1IOC4	R/W	R/W	-	-
0xFFFFF61D	TAA1 option register 1	TAA1OPT1	R/W	R/W	-	-
0xFFFFF61E	TAA1 specific overflow value control register	TAA1SVC	R/W	R/W	-	-
0xFFFFF620	TAA2 control register 0	TAA2CTL0	R/W	R/W	-	-
0xFFFFF621	TAA2 control register 1	TAA2CTL1	R/W	R/W	-	-
0xFFFFF622	TAA2 I/O control register 0	TAA2IOC0	R/W	R/W	-	-
0xFFFFF623	TAA2 I/O control register 1	TAA2IOC1	R/W	R/W	-	-
0xFFFFF624	TAA2 I/O control register 2	TAA2IOC2	R/W	R/W	-	-
0xFFFFF625	TAA2 option register 0	TAA2OPT0	R/W	R/W	-	-
0xFFFFF626	TAA2 capture/compare register 0	TAA2CCR0	-	-	R/W	-
0xFFFFF628	TAA2 capture/compare register 1	TAA2CCR1	-	-	R/W	-
0xFFFFF62A	TAA2 counter read buffer register	TAA2CNT	-	-	R	-

Table A-3 Other special function registers (11/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF62C	TAA2 I/O control register 4	TAA2IOC4	R/W	R/W	-	-
0xFFFFF62D	TAA2 option register 1	TAA2OPT1	R/W	R/W	-	-
0xFFFFF62E	TAA2 specific overflow value control register	TAA2SVC	R/W	R/W	-	-
0xFFFFF630	TAA3 control register 0	TAA3CTL0	R/W	R/W	-	-
0xFFFFF631	TAA3 control register 1	TAA3CTL1	R/W	R/W	-	-
0xFFFFF632	TAA3 I/O control register 0	TAA3IOC0	R/W	R/W	-	-
0xFFFFF633	TAA3 I/O control register 1	TAA3IOC1	R/W	R/W	-	-
0xFFFFF634	TAA3 I/O control register 2	TAA3IOC2	R/W	R/W	-	-
0xFFFFF635	TAA3 option register 0	TAA3OPT0	R/W	R/W	-	-
0xFFFFF636	TAA3 capture/compare register 0	TAA3CCR0	-	-	R/W	-
0xFFFFF638	TAA3 capture/compare register 1	TAA3CCR1	-	-	R/W	-
0xFFFFF63A	TAA3 counter read buffer register	TAA3CNT	-	-	R	-
0xFFFFF63C	TAA3 I/O control register 4	TAA3IOC4	R/W	R/W	-	-
0xFFFFF63D	TAA3 option register 1	TAA3OPT1	R/W	R/W	-	-
0xFFFFF63E	TAA3 specific overflow value control register	TAA3SVC	R/W	R/W	-	-
0xFFFFF640	TAA4 control register 0	TAA4CTL0	R/W	R/W	-	-
0xFFFFF641	TAA4 control register 1	TAA4CTL1	R/W	R/W	-	-
0xFFFFF642	TAA4 I/O control register 0	TAA4IOC0	R/W	R/W	-	-
0xFFFFF643	TAA4 I/O control register 1	TAA4IOC1	R/W	R/W	-	-
0xFFFFF644	TAA4 I/O control register 2	TAA4IOC2	R/W	R/W	-	-
0xFFFFF645	TAA4 option register 0	TAA4OPT0	R/W	R/W	-	-
0xFFFFF646	TAA4 capture/compare register 0	TAA4CCR0	-	-	R/W	-
0xFFFFF648	TAA4 capture/compare register 1	TAA4CCR1	-	-	R/W	-
0xFFFFF64A	TAA4 counter read buffer register	TAA4CNT	-	-	R	-
0xFFFFF64C	TAA4 I/O control register 4	TAA4IOC4	R/W	R/W	-	-
0xFFFFF64E	TAA4 specific overflow value control register	TAA4SVC	R/W	R/W	-	-
0xFFFFF650	TAA5 control register 0	TAA5CTL0	R/W	R/W	-	-
0xFFFFF651	TAA5 ontrol register 1	TAA5CTL1	R/W	R/W	-	-
0xFFFFF652	TAA5 I/O control register 0	TAA5IOC0	R/W	R/W	-	-
0xFFFFF653	TAA5 I/O control register 1	TAA5IOC1	R/W	R/W	-	-
0xFFFFF654	TAA5 I/O control register 2	TAA5IOC2	R/W	R/W	-	-
0xFFFFF655	TAA5 option register 0	TAA5OPT0	R/W	R/W	-	-
0xFFFFF656	TAA5 capture/compare register 0	TAA5CCR0	-	-	R/W	-
0xFFFFF658	TAA5 capture/compare register 1	TAA5CCR1	-	-	R/W	-
0xFFFFF65A	TAA5 counter read buffer register	TAA5CNT	-	-	R	-
0xFFFFF65C	TAA5 I/O control register 4	TAA5IOC4	R/W	R/W	-	-
0xFFFFF65D	TAA5 option register 1	TAA5OPT1	R/W	R/W	-	-
0xFFFFF65E	TAA5 specific overflow value control register	TAA5SVC	R/W	R/W	-	-
0xFFFFF660	TAA6 control register 0	TAA6CTL0	R/W	R/W	-	-
0xFFFFF661	TAA6 control register 1	TAA6CTL1	R/W	R/W	-	-
0xFFFFF662	TAA6 I/O control register 0	TAA6IOC0	R/W	R/W	-	-

Table A-3 Other special function registers (12/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF663	TAA6 I/O control register 1	TAA6IOC1	R/W	R/W	-	-
0xFFFFF664	TAA6 I/O control register 2	TAA6IOC2	R/W	R/W	-	-
0xFFFFF665	TAA6 option register 0	TAA6OPT0	R/W	R/W	-	-
0xFFFFF666	TAA6 capture/compare register 0	TAA6CCR0	-	-	R/W	-
0xFFFFF668	TAA6 capture/compare register 1	TAA6CCR1	-	-	R/W	-
0xFFFFF66A	TAA6 counter read buffer register	TAA6CNT	-	-	R	-
0xFFFFF66C	TAA6 I/O control register 4	TAA6IOC4	R/W	R/W	-	-
0xFFFFF66D	TAA6 option register 1	TAA6OPT1	R/W	R/W	-	-
0xFFFFF66E	TAA6 specific overflow value control register	TAA6SVC	R/W	R/W	-	-
0xFFFFF670	TAA7 control register 0	TAA7CTL0	R/W	R/W	-	-
0xFFFFF671	TAA7 control register 1	TAA7CTL1	R/W	R/W	-	-
0xFFFFF672	TAA7 I/O control register 0	TAA7IOC0	R/W	R/W	-	-
0xFFFFF673	TAA7 I/O control register 1	TAA7IOC1	R/W	R/W	-	-
0xFFFFF674	TAA7 I/O control register 2	TAA7IOC2	R/W	R/W	-	-
0xFFFFF675	TAA7 option register 0	TAA7OPT0	R/W	R/W	-	-
0xFFFFF676	TAA7 capture/compare register 0	TAA7CCR0	-	-	R/W	-
0xFFFFF678	TAA7 capture/compare register 1	TAA7CCR1	-	-	R/W	-
0xFFFFF67A	TAA7 counter read buffer register	TAA7CNT	-	-	R	-
0xFFFFF67C	TAA7 I/O control register 4	TAA7IOC4	R/W	R/W	-	-
0xFFFFF67D	TAA7 option register 1	TAA7OPT1	R/W	R/W	-	-
0xFFFFF67E	TAA7 specific overflow value control register	TAA7SVC	R/W	R/W	-	-
0xFFFFF680	TAA8 control register 0	TAA8CTL0	R/W	R/W	-	-
0xFFFFF681	TAA8 control register 1	TAA8CTL1	R/W	R/W	-	-
0xFFFFF682	TAA8 I/O control register 0	TAA8IOC0	R/W	R/W	-	-
0xFFFFF683	TAA8 I/O control register 1	TAA8IOC1	R/W	R/W	-	-
0xFFFFF684	TAA8 I/O control register 2	TAA8IOC2	R/W	R/W	-	-
0xFFFFF685	TAA8 option register 0	TAA8OPT0	R/W	R/W	-	-
0xFFFFF686	TAA8 capture/compare register 0	TAA8CCR0	-	-	R/W	-
0xFFFFF688	TAA8 capture/compare register 1	TAA8CCR1	-	-	R/W	-
0xFFFFF68A	TAA8 counter read buffer register	TAA8CNT	-	-	R	-
0xFFFFF68C	TAA8 I/O control register 4	TAA8IOC4	R/W	R/W	-	-
0xFFFFF68E	TAA8 specific overflow value control register	TAA8SVC	R/W	R/W	-	-
0xFFFFF690	TMT0 control register 0	TT0CTL0	R/W	R/W	-	-
0xFFFFF691	TMT0 control register 1	TT0CTL1	R/W	R/W	-	-
0xFFFFF692	TMT0 control register 2	TT0CTL2	R/W	R/W	-	-
0xFFFFF693	TMT0 I/O control register 0	TT0IOC0	R/W	R/W	-	-
0xFFFFF694	TMT0 I/O control register 1	TT0IOC1	R/W	R/W	-	-
0xFFFFF695	TMT0 I/O control register 2	TT0IOC2	R/W	R/W	-	-
0xFFFFF696	TMT0 I/O control register 3	TT0IOC3	R/W	R/W	-	-
0xFFFFF697	TMT0 option register 0	TT0OPT0	R/W	R/W	-	-
0xFFFFF698	TMT0 option register 1	TT0OPT1	R/W	R/W	-	-

Table A-3 Other special function registers (13/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF699	TMT0 option register 2	TT0OPT2	R/W	R/W	-	-
0xFFFFF69A	TMT0 capture/compare register 0	TT0CCR0	-	-	R/W	-
0xFFFFF69C	TMT0 capture/compare register 1	TT0CCR1	-	-	R/W	-
0xFFFFF69E	TMT0 counter read buffer register	TT0CNT	-	-	R	-
0xFFFFF6A0	TMT1 control register 0	TT1CTL0	R/W	R/W	-	-
0xFFFFF6A1	TMT1 control register 1	TT1CTL1	R/W	R/W	-	-
0xFFFFF6A2	TMT1 control register 2	TT1CTL2	R/W	R/W	-	-
0xFFFFF6A3	TMT1 I/O control register 0	TT1IOC0	R/W	R/W	-	-
0xFFFFF6A4	TMT1 I/O control register 1	TT1IOC1	R/W	R/W	-	-
0xFFFFF6A5	TMT1 I/O control register 2	TT1IOC2	R/W	R/W	-	-
0xFFFFF6A6	TMT1 I/O control register 3	TT1IOC3	R/W	R/W	-	-
0xFFFFF6A7	TMT1 option register 0	TT1OPT0	R/W	R/W	-	-
0xFFFFF6A8	TMT1 option register 1	TT1OPT1	R/W	R/W	-	-
0xFFFFF6A9	TMT1 option register 2	TT1OPT2	R/W	R/W	-	-
0xFFFFF6AA	TMT1 capture/compare register 0	TT1CCR0	-	-	R/W	-
0xFFFFF6AC	TMT1 capture/compare register 1	TT1CCR1	-	-	R/W	-
0xFFFFF6AE	TMT1 counter read buffer register	TT1CNT	-	-	R	-
0xFFFFF6B0	TAA9 control register 0	TAA9CTL0	R/W	R/W	-	-
0xFFFFF6B1	TAA9 control register 1	TAA9CTL1	R/W	R/W	-	-
0xFFFFF6B2	TAA9 I/O control register 0	TAA9IOC0	R/W	R/W	-	-
0xFFFFF6B3	TAA9 I/O control register 1	TAA9IOC1	R/W	R/W	-	-
0xFFFFF6B4	TAA9 I/O control register 2	TAA9IOC2	R/W	R/W	-	-
0xFFFFF6B5	TAA9 option register 0	TAA9OPT0	R/W	R/W	-	-
0xFFFFF6B6	TAA9 capture/compare register 0	TAA9CCR0	-	-	R/W	-
0xFFFFF6B8	TAA9 capture/compare register 1	TAA9CCR1	-	-	R/W	-
0xFFFFF6BA	TAA9 counter read buffer register	TAA9CNT	-	-	R	-
0xFFFFF6BC	TAA9 I/O control register 3	TAA9IOC4	R/W	R/W	-	-
0xFFFFF6BD	TAA9 option register 1	TAA9OPT1	R/W	R/W	-	-
0xFFFFF6BE	TAA9 specific overflow value control register	TAA9SVC	R/W	R/W	-	-
0xFFFFF6C0	FlexRay IBUSY/OBUSY edge selection register	FRESEL	R/W	R/W	-	-
0xFFFFF6C1	FlexRay SCLK selection register	SCLKSEL	R/W	R/W	-	-
0xFFFFF6D0	Interrupt share enable register	INTSEL	R/W	R/W	-	-
0xFFFFF6D2	Interrupt source flag register	INTERRF	R/W	R/W	-	-
0xFFFFF6F0	Timer input control register 0	TAAIC0	R/W	R/W	-	-
0xFFFFF6F1	Timer input control register 1	TAAIC1	R/W	R/W	-	-
0xFFFFF6F2	Timer input control register 2	TAAIC2	R/W	R/W	-	-
0xFFFFF6F3	Timer input control register 3	TAAIC3	R/W	R/W	-	-
0xFFFFF700	Random number register	RNG	-	-	R	-
0xFFFFF7A0	Noise rejection time control register	NRC	R/W	R/W	-	-
0xFFFFF802	Peripheral status register	PHS	R/W	R/W	-	-
0xFFFFF860	CAN clock selection register	CANCKSEL	R/W	R/W	-	-

Table A-3 Other special function registers (14/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF880	External interrupt edge specification register	INTM0	R/W	R/W	-	-
0xFFFFF882	External interrupt edge specification register	INTM1	R/W	R/W	-	-
0xFFFFF884	External interrupt edge specification register	INTM2	R/W	R/W	-	-
0xFFFFF886	External interrupt edge specification register	INTM3	R/W	R/W	-	-
0xFFFFF888	Emergency shut-off control register 5	PESC5	R/W	R/W	-	-
0xFFFFF88A	Emergency shut-off status register 5	ESOST5	R/W	R/W	-	-
0xFFFFF88C	Emergency shut-off control register 6	PESC6	R/W	R/W	-	-
0xFFFFF88E	Emergency shut-off status register 6	ESOST6	R/W	R/W	-	-
0xFFFFF890	Emergency shut-off mask register 5	PESMK5	R/W	R/W	-	-
0xFFFFF892	Emergency shut-off mask register 6	PESMK6	R/W	R/W	-	-
0xFFFFF8A0	Flash ROM ECC error register	ROMEAD	-	-	-	R
0xFFFFF8B0	RAM ECC error register	RAMEAD	-	-	R	-
0xFFFFF8B0	RAM ECC error address register	RAMEAD	-	-	R	-
0xFFFFF8B2	RAM ECC data location register	RAMEDLR	-	R	-	-
0xFFFFF8B4	RAM ECC control register	RAMECC	R/W	R/W	-	-
0xFFFFF990	TMT0 counter write buffer register	TT0TCW	-	-	R/W	-
0xFFFFF9A0	TMT1 counter write buffer register	TT1TCW	-	-	R/W	-
0xFFFFFA00	UARTC0 control register 0	UC0CTL0	R/W	R/W	-	-
0xFFFFFA01	UARTC0 control register 1	UC0CTL1	-	R/W	-	-
0xFFFFFA02	UARTC0 control register 2	UC0CTL2	-	R/W	-	-
0xFFFFFA03	UARTC0 option control register 0	UC0OPT0	R/W	R/W	-	-
0xFFFFFA04	UARTC0 status register	UC0STR	R/W	R/W	-	-
0xFFFFFA06	UARTC0 receive data register	UC0RX	-	-	R	-
0xFFFFFA06	UARTC0 receive data register L	UC0RXL	-	R	-	-
0xFFFFFA08	UARTC0 transmit data register	UC0TX	-	-	R/W	-
0xFFFFFA08	UARTC0 transmit data register L	UC0TXL	-	R/W	-	-
0xFFFFFA0A	UARTC0 option control register 1	UC0OPT1	R/W	R/W	-	-
0xFFFFFA0B	UARTC0 option control register 2	UC0OPT2	R	R	-	-
0xFFFFFA20	UARTC1 control register 0	UC1CTL0	R/W	R/W	-	-
0xFFFFFA21	UARTC1 control register 1	UC1CTL1	-	R/W	-	-
0xFFFFFA22	UARTC1 control register 2	UC1CTL2	-	R/W	-	-
0xFFFFFA23	UARTC1 option control register 0	UC1OPT0	R/W	R/W	-	-
0xFFFFFA24	UARTC1 status register	UC1STR	R/W	R/W	-	-
0xFFFFFA26	UARTC1 receive data register	UC1RX	-	-	R	-
0xFFFFFA26	UARTC1 receive data register L	UC1RXL	-	R	-	-
0xFFFFFA28	UARTC1 transmit data register	UC1TX	-	-	R/W	-
0xFFFFFA28	UARTC1 transmit data register L	UC1TXL	-	R/W	-	-
0xFFFFFA2A	UARTC1 option control register 1	UC1OPT1	R/W	R/W	-	-
0xFFFFFA2B	UARTC1 option control register 2	UC1OPT2	R	R	-	-
0xFFFFFA40	UARTC2 control register 0	UC2CTL0	R/W	R/W	-	-
0xFFFFFA41	UARTC2 control register 1	UC2CTL1	-	R/W	-	-

Table A-3 Other special function registers (15/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFFA42	UARTC2 control register 2	UC2CTL2	-	R/W	-	-
0xFFFFFA43	UARTC2 option control register 0	UC2OPT0	R/W	R/W	-	-
0xFFFFFA44	UARTC2 status register	UC2STR	R/W	R/W	-	-
0xFFFFFA46	UARTC2 receive data register	UC2RX	-	-	R	-
0xFFFFFA46	UARTC2 receive data register L	UC2RXL	-	R	-	-
0xFFFFFA48	UARTC2 transmit data register	UC2TX	-	-	R/W	-
0xFFFFFA48	UARTC2 transmit data register L	UC2TXL	-	R/W	-	-
0xFFFFFA4A	UARTC2 option control register 1	UC2OPT1	R/W	R/W	-	-
0xFFFFFA4B	UARTC2 option control register 2	UC2OPT2	R	R	-	-
0xFFFFFB40	TMS0 option register 1A	TS0OPT1A	-	-	R/W	-
0xFFFFFB40	TMS0 option register 1AL	TS0OPT1AL	R/W	R/W	-	-
0xFFFFFB41	TMS0 option register 1AH	TS0OPT1AH	R/W	R/W	-	-
0xFFFFFB42	TMS0 dead time setting register 1A	TS0DTC1A	-	-	R/W	-
0xFFFFFB44	TMS0 dead time setting register 0A	TS0DTC0A	-	-	R/W	-
0xFFFFFB46	TMS0 compare register 5A	TS0CCR5A	-	-	R/W	-
0xFFFFFB48	TMS0 compare register 4A	TS0CCR4A	-	-	R/W	-
0xFFFFFB4A	TMS0 compare register 3B	TS0CCR3B	-	-	R/W	-
0xFFFFFB4C	TMS0 compare register 2B	TS0CCR2B	-	-	R/W	-
0xFFFFFB4E	TMS0 compare register 1B	TS0CCR1B	-	-	R/W	-
0xFFFFFB50	TMS0 compare register 0A	TS0CCR0A	-	-	R/W	-
0xFFFFFB52	TMS0 compare register 3A	TS0CCR3A	-	-	R/W	-
0xFFFFFB54	TMS0 compare register 2A	TS0CCR2A	-	-	R/W	-
0xFFFFFB56	TMS0 compare register 1A	TS0CCR1A	-	-	R/W	-
0xFFFFFB58	TMS0 extension control register	TS0XCTL	R/W	R/W	-	-
0xFFFFFBC0	TMS1 option register 1A	TS1OPT1A	-	-	R/W	-
0xFFFFFBC0	TMS1 option register 1AL	TS1OPT1AL	R/W	R/W	-	-
0xFFFFFBC1	TMS1 option register 1AH	TS1OPT1AH	R/W	R/W	-	-
0xFFFFFBC2	TMS1 dead time setting register 1A	TS1DTC1A	-	-	R/W	-
0xFFFFFBC4	TMS1 dead time setting register 0A	TS1DTC0A	-	-	R/W	-
0xFFFFFBC6	TMS1 compare register 5A	TS1CCR5A	-	-	R/W	-
0xFFFFFBC8	TMS1 compare register 4A	TS1CCR4A	-	-	R/W	-
0xFFFFFBCA	TMS1 compare register 3B	TS1CCR3B	-	-	R/W	-
0xFFFFFBCC	TMS1 compare register 2B	TS1CCR2B	-	-	R/W	-
0xFFFFFBCE	TMS1 compare register 1B	TS1CCR1B	-	-	R/W	-
0xFFFFFBD0	TMS1 compare register 0A	TS1CCR0A	-	-	R/W	-
0xFFFFFBD2	TMS1 compare register 3A	TS1CCR3A	-	-	R/W	-
0xFFFFFBD4	TMS1 compare register 2A	TS1CCR2A	-	-	R/W	-
0xFFFFFBD6	TMS1 compare register 1A	TS1CCR1A	-	-	R/W	-
0xFFFFFBD8	TMS1 extension control register	TS1XCTL	R/W	R/W	-	-
0xFFFFFC30	Port8 function control expansion register	PFCE8	R/W	R/W	-	-
0xFFFFFC32	Port9 function control expansion register	PFCE9	R/W	R/W	-	-

Table A-3 Other special function registers (16/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFFD00	CSIB0 control register 0	CB0CTL0	R/W	R/W	-	-
0xFFFFFD01	CSIB0 control register 1	CB0CTL1	R/W	R/W	-	-
0xFFFFFD02	CSIB0 control register 2	CB0CTL2	-	R/W	-	-
0xFFFFFD03	CSIB0 status register	CB0STR	R/W	R/W	-	-
0xFFFFFD04	CSIB0 receive data register 0	CB0RX0	-	-	R	-
0xFFFFFD04	CSIB0 receive data register 0L	CB0RX0L	-	R	-	-
0xFFFFFD06	CSIB0 transmit data register	CB0TX0	-	-	R/W	-
0xFFFFFD06	CSIB0 transmit data register	CB0TX0L	-	R/W	-	-
0xFFFFFD20	CSIB1 control register 0	CB1CTL0	R/W	R/W	-	-
0xFFFFFD21	CSIB1 control register 1	CB1CTL1	R/W	R/W	-	-
0xFFFFFD22	CSIB1 control register 2	CB1CTL2	-	R/W	-	-
0xFFFFFD23	CSIB1 status register	CB1STR	R/W	R/W	-	-
0xFFFFFD24	CSIB1 receive data register 0	CB1RX0	-	-	R	-
0xFFFFFD24	CSIB1 receive data register 0L	CB1RX0L	-	R	-	-
0xFFFFFD26	CSIB1 transmit data register	CB1TX0	-	-	R/W	-
0xFFFFFD26	CSIB1 transmit data register	CB1TX0L	-	R/W	-	-
0xFFFFFD40	Queued CSI0 control register 0	CE0CTL0	R/W	R/W	-	-
0xFFFFFD41	Queued CSI0 control register 1	CE0CTL1	R/W	R/W	-	-
0xFFFFFD42	CSIE0 receive data register	CE0RX0	-	-	R	-
0xFFFFFD42	CSIE0 receive data register L	CE0RXL0	-	R	-	-
0xFFFFFD43	CSIE0 receive data register H	CE0RXH0	-	R	-	-
0xFFFFFD44	CSIE0 chip select data buffer register	CE0CS	-	-	R/W	-
0xFFFFFD46	CSIE0 transmission data buffer register	CE0TX0	-	-	R/W	-
0xFFFFFD46	CSIE0 transmission data buffer register L	CE0TXL0	-	R/W	-	-
0xFFFFFD47	CSIE0 transmission data buffer register H	CE0TXH0	-	R/W	-	-
0xFFFFFD48	Queued CSI0 status register	CE0STR	R/W	R/W	-	-
0xFFFFFD49	Queued CSI0 control register 2	CE0CTL2	R/W	R/W	-	-
0xFFFFFD4C	Queued CSI0 control register 3	CE0CTL3	R/W	R/W	-	-
0xFFFFFD4D	Queued CSI0 control register 4	CE0CTL4	R/W	R/W	-	-
0xFFFFFD50	Queued CSI0 enhanced timing register 0	CE0OPT0	-	-	R/W	-
0xFFFFFD52	Queued CSI0 enhanced timing register 1	CE0OPT1	-	-	R/W	-
0xFFFFFD54	Queued CSI0 enhanced timing register 2	CE0OPT2	-	-	R/W	-
0xFFFFFD56	Queued CSI0 enhanced timing register 3	CE0OPT3	-	-	R/W	-
0xFFFFFD58	Queued CSI0 enhanced timing register 4	CE0OPT4	-	-	R/W	-
0xFFFFFD5A	Queued CSI0 enhanced timing register 5	CE0OPT5	-	-	R/W	-
0xFFFFFD5C	Queued CSI0 enhanced timing register 6	CE0OPT6	-	-	R/W	-
0xFFFFFD5E	Queued CSI0 enhanced timing register 7	CE0OPT7	-	-	R/W	-
0xFFFFFD80	Queued CSI1 control register 0	CE1CTL0	R/W	R/W	-	-
0xFFFFFD81	Queued CSI1 control register 1	CE1CTL1	R/W	R/W	-	-
0xFFFFFD82	CSIE1 receive data register	CE1RX0	-	-	R	-
0xFFFFFD82	CSIE1 receive data register L	CE1RXL0	-	R	-	-

Table A-3 Other special function registers (17/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFFD83	CSIE1 receive data register H	CE1RXH0	-	R	-	-
0xFFFFFD84	CSIE1 chip select data buffer register	CE1CS	-	-	R/W	-
0xFFFFFD86	CSIE1 transmission data buffer register	CE1TX0	-	-	R/W	-
0xFFFFFD86	CSIE1 transmission data buffer register L	CE1TXL0	-	R/W	-	-
0xFFFFFD87	CSIE1 transmission data buffer register H	CE1TXH0	-	R/W	-	-
0xFFFFFD88	Queued CSI1 status register	CE1STR	R/W	R/W	-	-
0xFFFFFD89	Queued CSI1 control register 2	CE1CTL2	R/W	R/W	-	-
0xFFFFFD8C	Queued CSI1 control register 3	CE1CTL3	R/W	R/W	-	-
0xFFFFFD8D	Queued CSI1 control register 4	CE1CTL4	R/W	R/W	-	-
0xFFFFFD90	Queued CSI1 enhanced timing register 0	CE1OPT0	-	-	R/W	-
0xFFFFFD92	Queued CSI1 enhanced timing register 1	CE1OPT1	-	-	R/W	-
0xFFFFFD94	Queued CSI1 enhanced timing register 2	CE1OPT2	-	-	R/W	-
0xFFFFFD96	Queued CSI1 enhanced timing register 3	CE1OPT3	-	-	R/W	-
0xFFFFFDC0	Prescaler mode register 0	PRSM0	R/W	R/W	-	-
0xFFFFFDC1	Prescaler compare registers 0	PRSCM0	R/W	R/W	-	-
0xFFFFFDD0	Prescaler mode register 1	PRSM1	R/W	R/W	-	-
0xFFFFFDD1	Prescaler compare registers 1	PRSCM1	R/W	R/W	-	-
0xFFFFFDE0	Prescaler mode register 2	PRSM2	R/W	R/W	-	-
0xFFFFFDE1	Prescaler compare registers 2	PRSCM2	R/W	R/W	-	-
0xFFFFFE00	DMA wait control register 0	DMAWC0	-	R/W	-	-
0xFFFFFE02	DMA wait control register 1	DMAWC1	-	R/W	-	-

Revision History

This revision list shows all functional changes compared to the previous manual version U17754EE3V0UM00 (published in October 2007).

Chapter 24 "FlexRay™ (FR)"

The *Chapter 24 "FlexRay™ (FR)"* has been completely revised. Thus the changes in this chapter compared to the previous manual version are not listed in the below table, but are marked with change bars.

Chapter	Page	Description
1	25	µPD70F3483 device added
2	93	recommended connection of unused pins corrected
3	110	program start address in normal operation mode corrected
7	197	size of code flash rewriting units corrected
11	336	caution concerning "UARTCn receive error interrupt" added
13	384	caution added: CEnCS.CEnCS[3:0] is prohibited in enhanced timing mode
14	553	CAN Rx via interrupt description enhanced

Index

Numerics

16-bit data busses
    Access to 273
MODE 110
8-bit data busses
    Access to 267

A

A/D conversion result register for DMA (ADDMAn) 579
A/D conversion result registers (ADCRnm, ADCRnmH) 576
A/D Converter
    Operation in A/D trigger mode 586
    Operation in external trigger mode 596
    Operation in timer trigger mode 590
A/D Converter mode register 0 (ADMn0) 571
A/D Converter mode register 1 (ADMn1) 572
A/D Converter mode register 2 (ADMn2) 574
A/D Converter trigger source select register (ADTRSELn) 575
Access to
    16-bit data busses 273
    8-bit data busses 267
    External devices (initialization) 238
ADCRnm 576
ADCRnmH 576
ADDMAn 579
Address setup wait control register (AWC) 249
Address space 111
    CPU 111
    Images 111
    Physical 111
ADIC0 143
ADIC1 143
ADMn0 571
ADMn1 572
ADMn2 574
ADTRSELn 575
AFO (Auxiliary Frequency Output) 1129
Aggregated Channel Status (FRACS) 1224
Analog filtered inputs 87
Anytime write
    TMT 1046
Asynchronous Serial Interface
    see UARTC
Auxiliary Frequency Output 1129
AWC 249

B

Batch rewrite
    TMT 1048
Baud rate generator
    CSIB 364
    UARTC 329
BCC 253
BCTn 247
BCU
    Registers 240
BCU (Bus Control Unit) 225
BEC 246
Boundary operation conditions 237
Boundary Scan 1349
BPC 240
BRGIC 143
BRGIC2 143
BSC 245
Bus and memory control 225
    Registers 239
Bus clock dividing frequency control register (DVC) 255
Bus cycle configuration register (BCTn) 247
Bus cycle control register (BCC) 253
Bus size configuration register (BSC) 245

C

C0ERRIC 143
C0RECIC 143
C0TRXIC 143
C0WUPIC 143
C1ERRIC 143
C1RECIC 143
C1TRXIC 143
C1WUPIC 143
C2ERRIC 143
C2RECIC 143
C2TRXIC 143
C2WUPIC 143
C3ERRIC 143
C3RECIC 143
C3TRXIC 143
C3WUPIC 143
CALLT base pointer (CTBP) 107
CAN (Controller area network) 429
CAN clock selection register (CANCKSEL) 182
CAN Controller 429
    Baud rate settings 530
    Bit set/clear function 461
    Configuration 431

Connection with target system 453
Control registers 463
Diagnosis functions 525
Functions 442
Initialization 499
Internal registers 454
Interrupt function 524
Message reception 503
Message transmission 511
Operation 538
Overview of functions 430
Power saving modes 519
Register access type 456
Register bit configuration 458
Special operational modes 525
Time stamp function 529
Transition from initialization mode to operation mode 501
CAN protocol 432
CANCKSEL 182
CANn global automatic block transmission control register (CnGMABT) 466
CANn global automatic block transmission delay register (CnGMABTD) 468
CANn global clock selection register (CnGMCS) 465
CANn global control register (CnGMCTRL) 463
CANn message configuration register m (CnMCONFm) 493
CANn message control register m (CnMCTRLm) 496
CANn message data byte register (CnMDATAxm) 490
CANn message data length register m (CnMDLCm) 492
CANn message ID register m (CnMIDLm, CnMIDHm) 495
CANn module bit rate prescaler register (CnBRP) 481
CANn module bit rate register (CnBTR) 482
CANn module control register (CnCTRL) 471
CANn module error counter register (CnERC) 477
CANn module information register (CnINFO) 476
CANn module interrupt enable register (CnIE) 478
CANn module interrupt status register (CnINTS) 480
CANn module last error information register (CnLEC) 475
CANn module last in-pointer register (CnLIPT) 483
CANn module last out-pointer register (CnLOPT) 485
CANn module mask control register (CnMASKaL, CnMASKaH) 469
CANn module receive history list register (CnRGPT) 484
CANn module time stamp register (CnTS) 488
CANn module transmit history list register (CnTGPT) 486
CB0REIC 143
CB0RIC 143
CB0TIC 143
CB1REIC 143
CB1RIC 143
CB1TIC 143
CBnCTL0 340
CBnCTL1 176, 342
CBnCTL2 343
CBnRX0 339
CBnSTR 345
CBnTX 174
CBnTX0 339
CC Error Vector (FRCCEV) 1217
CC Status Vector (FRCCSV) 1214
CCRCDn (CPU-CRC data register) 173
CCRCINn (CPU-CRC input register) 172, 1140
CE0CIC 143
CE0OFIC 143
CE1CIC 143
CE1OFIC 143
CEnCS 383
CEnCTL0 373
CEnCTL1 375
CEnCTL2 378
CEnCTL3 379
CEnCTL4 380
CEnOPTm 388
CEnRX0 382
CEnSTR 386
CEnTX0 385
Chip area select control register (CSCn) 242
Chip select data buffer registers (CEnCS) 383
Chip select signals 228
Clock Generator 179
Clock Monitor 183
Clocked Serial Interface
see CSIB
CnBRP 481
CnBTR 482
CnCTRL 471
CnERC 477

CnGMABT 466
CnGMABTD 468
CnGMCS 465
CnGMCTRL 463
CnIE 478
CnINFO 476
CnINTS 480
CnLEC 475
CnLIPT 483
CnLOPT 485
CnMASKaH 469
CnMASKaL 469
CnMCONFm 493
CnMCTRLm 496
CnMDATAxm 490
CnMDLCm 492
CnMIDHm 495
CnMIDLm 495
CnRGPT 484
CnTGPT 486
CnTS 488
Command register (PRCMD) 47
Control settings (FRCS) 1166
Controller Information (FRCI) 1164
CPU
    Address space 111
    Functions 97
    Register set 99
CPU-CRC 169
CRC
    base address 234
    CCRCDn (CPU-CRC data register) 173
    CCRCINn (CPU-CRC input register) 172, 1140
    CPU-CRC 169
    CRCC (CRC control register) 175
    DATA-CRC 169
    DCRCIN 174
    Registers 171
CRC (Cyclic Redundancy Check) 169
CRCC (CRC control register) 175
CS 228
CSCn 242
CSIB
    Baud rate generator 364
    Control registers 340
    Operation 346
    Operation flow 358
    Output pins 357
CSIB (Clocked Serial Interface) 337
CSIB transmit data register (CBnTX) 174
CSIB transmit data register (CBnTX0) 339
CSIBn control register 0 (CBnCTL0) 340
CSIBn control register 1 (CBnCTL1) 176, 342
CSIBn control register 2 (CBnCTL2) 343
CSIBn receive data register (CBnRX0) 339
CSIBn status register (CBnSTR) 345
CTBP 107
CTPC 102
CTPSW 105
Cyclic Redundancy Check (CRC) 169

D

Data access order 267
Data address space
    Recommended use 122
Data busses
    Access order 267
Data flash 232
    base address 232
Data flash control register (DFLCTL) 201
Data space 113
Data wait control register (DWCn) 251
DATA-CRC 169
DBPC 102
DBPSW 105
DCRCIN 174
Debug function (on-chip) 1119
    Code protection 219
    Restrictions and Cautions 1126
Debug trap 162
DFLCTL 201
Digitally filtered inputs 88
DMA (direct memory access) 279
DMA Controller 279
DMA data size control register (DMDSCL) 289
DMA mode control register (DMAMCL) 288
DMA status register (DMASL) 288
DMA transfer
    Forcible termination 280
    Serial data reception 296
    Serial data transmission 298
DMA transfer count registers (DTCRn) 287
DMA transfer memory start address registers (MARn) 285
DMA transfer SFR start address registers (SARm) 284
DMA trigger factor registers (DTFRn) 286
DMA wait control registers (DMAWCm) 283
DMAIC2 143
DMAIC3 143

DMAMCL 288
DMASL 288
DMAWCm 283
DMDSCL 289
DTCRn 287
DTFRn 286
DVC 255
DWCn 251

E

ECC Control Register for E-Ray Message RAM (FRECCTL) 1158
ECR 106
ECT 109
EFG 108
EIPC 102
EIPSW 105
Element pointer 100
Emergency shut off control register (PESCn) 43
Emergency shut-off mask register (PESMKn) 46
Emergency shut-off status register (ESOSTn) 45
Endian configuration register (BEC) 246
Endian format 256
ERRIC 143
Error Interrupt Enable Reset (FREIER) 1184
Error Interrupt Enable Set (FREIES) 1184
Error Interrupt Line Select (FREILS) 1180
Error Interrupt Register (FREIR) 1173
ESOSTn 45
Even Sync ID (FRESIDn) 1226
Exception status flag (EP) 159
Exception trap 160
External bus properties 236
    Bus access 236
    Bus priority order 236
    Bus width 236
External devices
    Initialization for access 238
    Interface timing 259
External interrupt edge specification register (INTM0) 150
External interrupt edge specification register (INTM1) 151
External interrupt edge specification register (INTM2) 152
External interrupt edge specification register (INTM3) 152
External memory area 116

F

FEPC 102
FEPSW 105
FIFO Rejection Filter (FRFRF) 1231
Fixed peripheral I/O area 230
Flash area 115
Flash memory 191
    Self-programming 213
Flash programmer
    Communication mode 203
    Pin connection 205
Flash programming
    Mode 110
Flash ROM ECC error register (ROMEAD) 198
FlexRay 1145
    Appendix 1313
    base address 233
    Functional Description 1263
    Host interface 1151
    Overview 1147
    Programmer's model 1159
Floating point arithmetic control register (ECT) 109
Floating point arithmetic flag register (EFG) 108
Floating point arithmetic register set 107
FRACS 1224
FRCCEV 1217
FRCCSV 1214
FRCI 1164
FRCS 1166
FRECCTL 1158
FREIER 1184
FREIES 1184
FREILS 1180
FREIR 1173
FRESIDn 1226
FRFRF 1231
FRGTUC1 1204
FRGTUC10 1212
FRGTUC11 1213
FRGTUC2 1205
FRGTUC3 1206
FRGTUC4 1207
FRGTUC5 1208
FRGTUC6 1209
FRGTUC7 1210
FRGTUC8 1210
FRGTUC9 1211
FRIBCM 1250
FRIBCR 1251
FRIBIC 143
FRIC0 143

FRIC1 143
FRIC2 143
FRIC3 143
FRILE 1188
FRLCK 1172
FRMBS 1257
FRMBSC1 1242
FRMBSC2 1242
FRMBSC3 1242
FRMBSC4 1242
FRMHDC 1204
FRMHDS 1233, 1244
FRMRC 1229
FRMTCCV 1218
FRNDAT1 1240
FRNDAT2 1240
FRNDAT3 1240
FRNDAT4 1240
FRNEMC 1201
FRNMVn 1228
FROBCM 1260
FROBCR 1261
FROBIC 143
FROCV 1219
FROSIDn 1227
FRPRTC1 1202
FRPRTC2 1203
FRRCV 1219
FRRDDSn 1253
FRRDHS1 1254
FRRDHS2 1255
FRRDHS3 1256
FRSCV 1218
FRSFS 1220
FRSIER 1186
FRSIES 1186
FRSILS 1182
FRSIR 1177
FRSTPW 1191, 1193
FRSUCC1 1194
FRSUCC2 1199
FRSUCC3 1200
FRSWNIT 1222
FRT0C 1189
FRT1C 1190
FRTEST1 1168
FRTEST2 1172
FRTXRQn 1238
FRVI 1165
FRWRDSn 1246
FRWRHS1 1247
FRWRHS2 1249
FRWRHS3 1249

G

General purpose registers (r0 to r31) 100
Global pointer 100
GTU Configuration Register 1 (FRGTUC1) 1204
GTU Configuration Register 10 (FRGTUC10) 1212
GTU Configuration Register 11 (FRGTUC11) 1213
GTU Configuration Register 2 (FRGTUC2) 1205
GTU Configuration Register 3 (FRGTUC3) 1206
GTU Configuration Register 4 (FRGTUC4) 1207
GTU Configuration Register 5 (FRGTUC5) 1208
GTU Configuration Register 6 (FRGTUC6) 1209
GTU Configuration Register 7 (FRGTUC7) 1210
GTU Configuration Register 8 (FRGTUC8) 1210
GTU Configuration Register 9 (FRGTUC9) 1211

H

HALT mode 187, 188
    release 189

I

ID code 1121
Idle pins
    Recommended connection 93
Images in address space 111
IMRn 147
inital value of Pn registers changed to "undefined" 41
Initialization for access to external devices 238
Input Buffer Command Mask (FRIBCM) 1250
Input Buffer Command Request (FRIBCR) 1251
In-service priority register (ISPR) 148
INTC (Interrupt Controller) 125
Internal flash area 115
Internal peripheral function wait control register (VSWC) 241
Internal RAM area 115
INTERRF 156
Interrupt
    Maskable 136
    Non-maskable 132
    Processing (multiple interrupts) 164
    Response time 166
Interrupt Controller 125
    Debug trap 162
    Edge and level detection configuration 150

Exception trap 160
Periods in which interrupts are not acknowledged 167
Software exception 157
Interrupt Line Enable (FRILE) 1188
Interrupt mask registers IMRn 147
Interrupt share enable register (INTSEL) 155
Interrupt source flag register (INTERRF) 156
Interrupt/exception source register (ECR) 106
INTM0 150
INTM1 151
INTM2 152
INTM3 152
INTSEL 155
ISPR 148

J

JTAG 1349

L

Link pointer 100
Lock Register (FRLCK) 1172

M

Macrotick and Cycle Counter Value (FRMTCCV) 1218
MARn 285
Maskable interrupt status flag (ID) 149
Maskable interrupts 136
Maskable Interrupts Control Register (xxICn) 143
MEMC (Memory Controller) 225
Memory 115
Access configuration 256
Areas 115
Blocks 228
Controller registers 247
Message Buffer Status (FRMBS) 1257
Message Buffer Status Changed 1 (FRMBSC1) 1242
Message Buffer Status Changed 2 (FRMBSC2) 1242
Message Buffer Status Changed 3 (FRMBSC3) 1242
Message Buffer Status Changed 4 (FRMBSC4) 1242
Message Handler Status (FRMHDS) 1233, 1244
Message RAM Configuration (FRMRC) 1229
MHD Configuration Register (FRMHDC) 1204

N

NBD (Non Break Debug Unit) 1133
NEM Configuration Register (FRNEMC) 1201
Network Management Vector (FRNMVn) 1228
New Data 1 (FRNDAT1) 1240
New Data 2 (FRNDAT2) 1240
New Data 3 (FRNDAT3) 1240
New Data 4 (FRNDAT4) 1240
Noise elimination
Pin input 87
Noise rejection time control register (NRC) 90
Non Break Debug Unit (NBD) 1133
Non-maskable interrupts 132
Normal operation mode 110
NRC 90
N-Wire
Code protection 219
Connection to emulator 1122
Emulator 1119
ID code 1121
Security function 1121

O

Odd Sync ID (FROSIDn) 1227
Offset Correction Value (FROCV) 1219
on-chip debug cautions revised 1126
Operation modes 110
Flash programming mode 110
Normal operation mode 110
OSDIC 143
Output Buffer Command Mask (FROBCM) 1260
Output Buffer Command Request (FROBCR) 1261

P

Package pins assignment 94
PC 102
PC saving registers 102
Peripheral area selection control register (BPC) 240
Peripheral I/O area
fixed 230
programmable 116, 230
Peripheral status register (PHS) 48
PESCn 43
PESMKn 46
PFCEn 40
PFCn 40
PHS 48
Physical address space 111
PIC0 143
PIC1 143
PIC10 143
PIC11 143

PIC12 143
PIC2 143
PIC3 143
PIC4 143
PIC5 143
PIC6 143
PIC7 143
PIC8 143
PIC9 143
Pin functions 31
    After reset/in stand-by modes 91
    List 56
    Unused pins 93
pins 110
PMCn 38
PMn 39
Pn 41
Port function control expansion register (PFCEn) 40
Port function control register (PFCn) 40
Port groups 32
    Configuration 49
    Configuration registers 36
    List 49
Port mode control register (PMCn) 38
Port mode register (PMn) 39
Port register (Pn) 41
power save function 187
PPA (programmable peripheral I/O area) 230
PRCMD 47
Prescaler compare registers (PRSCMn) 365
Prescaler compare registers 2 (PRSCM2) 1131
Prescaler mode register 2 (PRSM2) 1130
Prescaler mode registers (PRSMn) 365
Program counter (PC) 102
Program space 113
Program status word (PSW) 103
Programmable peripheral I/O area 116, 230
PRSCM2 1131
PRSCMn 365
PRSM2 1130
PRSMn 365
PRT Configuration Register 1 (FRPRTC1) 1202
PRT Configuration Register 2 (FRPRTC2) 1203
PSW 103
PSW saving registers 105

Q

Queued CSI control register 0 (CEnCTL0) 373
Queued CSI control register 1 (CEnCTL1) 375
Queued CSI control register 2 (CEnCTL2) 378
Queued CSI control register 3 (CEnCTL3) 379
Queued CSI control register 4 (CEnCTL4) 380
Queued CSI enhanced timing registers (CEnOPTm) 388
Queued-CSI status registers (CEnSTR) 386

R

RAM area 115
RAM ECC control register (RAMECC) 120
RAM ECC error address register (RAMEAD) 118
RAM ECC error data location register (RAMEDLR) 119
RAMEAD 118
RAMECC 120
RAMEDLR 119
Random number register (RNG) 1127
Rate Correction Value (FRRCV) 1219
Read Data Section (FRRDDSn) 1253
Read Header Section 1 (FRRDHS1) 1254
Read Header Section 2 (FRRDHS2) 1255
Read Header Section 3 (FRRDHS3) 1256
Receive data buffer registers (CEnRX0) 382
regID (system register number) 101
Reload
    TMT 1048
Reset
    Register status after reset 1346
RNG 1127
ROM
    see Internal flash area
ROMEAD 198

S

SARm 284
Saturated operation instructions 104
SFR (special function register) 1351
Slot Counter Value (FRSCV) 1218
Software exception 157
Special function registers (list) 1351
Stack pointer 100
Status Interrupt Enable Reset (FRSIER) 1186
Status Interrupt Enable Set (FRSIES) 1186
Status Interrupt Line Select (FRSILS) 1182
Status Interrupt Register (FRSIR) 1177
Stop Watch Register (FRSTPW) 1191, 1193
SUC Configuration Register 1 (FRSUCC1) 1194
SUC Configuration Register 2 (FRSUCC2) 1199
SUC Configuration Register 3 (FRSUCC3) 1200
Symbol Window and NIT Status (FRSWNIT) 1222

Sync Frame Status (FRSFS) 1220
System register set 101

T

TAA capture/compare register 0 (TAAnCCR0) 610
TAA capture/compare register 1 (TAAnCCR1) 611
TAA control register 0 (TAAnCTL0) 616
TAA counter read buffer register (TAAnCNT) 612
TAA dedicated I/O control register 0 (TAAnIOC0) 619
TAA dedicated I/O control register 1 (TAAnIOC1) 620
TAA I/O control register 2 (TAAnIOC2) 621
TAA I/O control register 3 (TAAnIOC3) 622
TAA option register 0 (TAAnOPT0) 623
TAA option register 1 (TAAnOPT1) 624
TAA specific overflow value control register 0 (TAAnSVC) 625
TAA timer control register 1 (TAAnCTL1) 617
TAA0CCIC0 143
TAA0CCIC1 143
TAA0OVIC 143
TAA1CCIC0 143
TAA1CCIC1 143
TAA1OVIC 143
TAA2CCIC0 143
TAA2CCIC1 143
TAA2OVIC 143
TAA3CCIC0 143
TAA3CCIC1 143
TAA3OVIC 143
TAA4CCIC0 143
TAA4CCIC1 143
TAA4OVIC 143
TAA5CCIC0 143
TAA5CCIC1 143
TAA5OVIC 143
TAA6CCIC0 143
TAA6CCIC1 143
TAA6OVIC 143
TAA7CCIC0 143
TAA7CCIC1 143
TAA7OVIC 143
TAA8CCIC0 143
TAA8CCIC1 143
TAA8OVIC 143
TAA9CCIC0 143
TAA9CCIC1 143
TAA9OVIC 143
TAAIC0 613
TAAIC1 614
TAAIC2 615
TAAIC3 1108
TAAnCCR0 610
TAAnCCR1 611
TAAnCNT 612
TAAnCTL0 616
TAAnCTL1 617
TAAnIOC0 619
TAAnIOC1 620
TAAnIOC2 621
TAAnIOC3 622
TAAnOPT0 623
TAAnOPT1 624
TAAnSVC 625
Test Register 1 (FRTEST1) 1168
Test Register 2 (FRTEST2) 1172
Text pointer 100
Timer 0 Configuration (FRT0C) 1189
Timer 1 Configuration (FRT1C) 1190
Timer I/O control register 3 (TAAIC3) 1108
Timer input control register 0 (TAAIC0) 613
Timer input control register 1 (TAAIC1) 614
Timer input control register 2 (TAAIC2) 615
Timer S 677
    A/D conversion trigger function 816
    Basic operation 759
    Configuration 679
    Control register 694
    Error/warning interrupt 823
    Features 677
    Flags 787
    Interrupt thinning out function 805
    Match interrupt 783
    Operation in each mode 827
Timer Sn compare register 0 (TSnCCR0) 682
Timer Sn compare register 0A (TSnCCR0A) 682
Timer Sn compare register 1 (TSnCCR1) 683
Timer Sn compare register 1A (TSnCCR1A) 683
Timer Sn compare register 1B (TSnCCR1B) 684
Timer Sn compare register 2 (TSnCCR2) 685
Timer Sn compare register 2A (TSnCCR2A) 685
Timer Sn compare register 2B (TSnCCR2B) 686
Timer Sn compare register 3 (TSnCCR3) 687
Timer Sn compare register 3A (TSnCCR3A) 687
Timer Sn compare register 3B (TSnCCR3B) 688
Timer Sn compare register 4 (TSnCCR4) 689
Timer Sn compare register 4A (TSnCCR4A) 689
Timer Sn compare register 5 (TSnCCR5) 690

Timer Sn compare register 5A (TSnCCR5A) 690
Timer Sn control register 0 (TSnCTL0) 694
Timer Sn control register 1 (TSnCTL1) 696
Timer Sn counter read register (TSnCNT) 691
Timer Sn dead time setting register 0 (TSnDTC0) 692
Timer Sn dead time setting register 0A (TSnDTC0A) 692
Timer Sn dead time setting register 1 (TSnDTC1) 693
Timer Sn dead time setting register 1A (TSnDTC1A) 693
Timer Sn extension control register (TSnXCTL) 758
Timer Sn I/O control register 0 (TSnIOC0) 698
Timer Sn I/O control register 2 (TSnIOC2) 700
Timer Sn I/O control register 4 (TSnIOC4) 701
Timer Sn option register (TSnOPT6) 748
Timer Sn option register 0 (TSnOPT0) 703
Timer Sn option register 4 (TSnOPT4) 718
Timer Sn option register 5 (TSnOPT5) 737
Timer Sn option register 7 (TSnOPT7) 752
Timer Sn option registers 1, 1H, 1L (TSnOPT1, TSnOPT1H, TSnOPT1L) 705
Timer Sn option registers 1A, 1AH, 1AL (TSnOPT1A, TSnOPT1AH, TSnOPT1AL) 705
Timer Sn option registers 2, 2H, 2L (TSnOPT2, TSnOPT2H, TSnOPT2L) 710
Timer Sn option registers 3, 3H, 3L (TSnOPT3, TSnOPT3H, TSnOPT3L) 714
Timer Sn pattern register 0 (TSnPAT0) 754
Timer Sn pattern register 1 (TSnPAT1) 756
Timer Sn sub-counter read register (TSnSBC) 691
TMT
    Anytime write 1046
    Batch rewrite 1048
    Reload 1048
TMTn capture/compare register 0 (TTnCCR0) 1023
TMTn capture/compare register 1 (TTnCCR1) 1024
TMTn control register 0 (TTnCTL0) 1026
TMTn control register 1 (TTnCTL1) 1028
TMTn control register 2 (TTnCTL2) 1030
TMTn counter read buffer register (TTnCNT) 1025
TMTn counter write buffer register (TTnTCW) 1025
TMTn I/O control register 0 (TTnIOC0) 1032
TMTn I/O control register 1 (TTnIOC1) 1033
TMTn I/O control register 2 (TTnIOC2) 1034
TMTn I/O control register 3 (TTnIOC3) 1035
TMTn option register 0 (TTnOPT0) 1037
TMTn option register 1 (TTnOPT1) 1039
TMTn option register 2 (TTnOPT2) 1041
TP3CC1IC 143
Transmission data buffer registers (CEnTX0) 385
Transmission Request n (FRTXRQn) 1238
TS0CCIC0 143
TS0CCIC1 143
TS0CCIC2 143
TS0CCIC3 143
TS0CCIC4 143
TS0CCIC5 143
TS0CDIC0 143
TS0ERIC 143
TS0ODIC 143
TS0OVIC 143
TS0WNIC 143
TS1CCIC0 143
TS1CCIC1 143
TS1CCIC2 143
TS1CCIC4 143
TS1CCIC5 143
TS1CDIC0 143
TS1ERIC 143
TS1ODIC 143
TS1OVIC 143
TS1WNIC 143
TSnCCR0 682
TSnCCR0A 682
TSnCCR1 683
TSnCCR1A 683
TSnCCR1B 684
TSnCCR2 685
TSnCCR2A 685
TSnCCR2B 686
TSnCCR3 687
TSnCCR3A 687
TSnCCR3B 688
TSnCCR4 689
TSnCCR4A 689
TSnCCR5 690
TSnCCR5A 690
TSnCNT 691
TSnCTL0 694
TSnCTL1 696
TSnDTC0 692
TSnDTC0A 692
TSnDTC1 693
TSnDTC1A 693

TSnIOC0 698
TSnIOC2 700
TSnIOC4 701
TSnOPT0 703
TSnOPT1, TSnOPT1H, TSnOPT1L 705
TSnOPT1A, TSnOPT1AH, TSnOPT1AL 705
TSnOPT2, TSnOPT2H, TSnOPT2L 710
TSnOPT3, TSnOPT3H, TSnOPT3L 714
TSnOPT4 718
TSnOPT5 737
TSnOPT6 748
TSnOPT7 752
TSnPAT0 754
TSnPAT1 756
TSnSBC 691
TSnXCTL 758
TT0CCIC0 143
TT0CCIC1 143
TT0ECIC 143
TT0OVIC 143
TT1CCIC0 143
TT1CCIC1 143
TT1ECIC 143
TT1OVIC 143
TTnCCR0 1023
TTnCCR1 1024
TTnCNT 1025
TTnCTL0 1026
TTnCTL1 1028
TTnCTL2 1030
TTnIOC0 1032
TTnIOC1 1033
TTnIOC2 1034
TTnIOC3 1035
TTnOPT0 1037
TTnOPT1 1039
TTnOPT2 1041
TTnTCW 1025

U

UARTC
- Cautions 335
- Dedicated baud rate generator 329
- Interrupt Request Signals 314
- Operation 315

UARTCn control register 0 (UCnCTL0) 304
UARTCn control register 1 (UCnCTL1) 330
UARTCn control register 2 (UCnCTL2) 331
UARTCn option control register 0 (UCnOPT0) 307
UARTCn option control register 1 (UCnOPT1) 309
UARTCn option control register 2 (UCnOPT2) 310
UARTCn receive data register (UCnRX) 312
UARTCn receive shift register 304
UARTCn status register (UCnSTR) 310
UARTCn transmit data register (UCnTX) 313
UARTCn transmit shift register 304
UC0REIC 143
UC0TIC 143
UC1REIC 143
UC1TIC 143
UCnCTL0 304
UCnCTL1 330
UCnCTL2 331
UCnOPT0 307
UCnOPT1 309
UCnOPT2 310
UCnRX 312
UCnSTR 310
UCnTX 313

V

Vendor Information (FRVI) 1165
VSWC 241

W

Wait functions (access to external devices) 257
Write Data Section (FRWRDSn) 1246
Write Header Section 1 (FRWRHS1) 1247
Write Header Section 2 (FRWRHS2) 1249
Write Header Section 3 (FRWRHS3) 1249
Write protected registers 123

Z

Zero register 100