FM4

32-BIT MICROCONTROLLER
FM4 Family
Communication Macro Part

PERIPHERAL MANUAL

SPANSION®

For the information for microcontroller supports, see the following web site.

http://www.spansion.com/jp/support/microcontrollers/

ARM®

Publication Number FM4_MN709-00004 Revision 2.0 Issue Date September 30, 2014

Preface

Thank you for your continued use of Spansion products.
Read this manual and "Data Sheet" thoroughly before using products in this family.

Purpose of this manual and intended readers

This manual explains the functions and operations of this family and describes how it is used. The manual is intended for engineers engaged in the actual development of products using this family.
For the descriptions on Analog macro, Timer, and Communication Macro, see the respective separate peripheral manual.

Note:

- *This manual explains the configuration and operation of the peripheral functions, but does not cover the specifics of each device in the series.*
  Users should refer to the respective data sheets of devices for device-specific details.

Trademark

ARM and Cortex are the registered trademarks of ARM Limited in the EU and other countries.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Sample programs and development environment

Spansion offers sample programs free of charge for using the peripheral functions of the FM4 family. Spansion also makes available descriptions of the development environment required for this family. Feel free to use them to verify the operational specifications and usage of this Spansion microcontroller.

Microcontroller support information:

http://www.spansion.com/support/microcontrollers/

Note:

- *Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.*
  Spansion assumes no responsibility for any damage that may occur as a result of using a sample program.

Overall Organization of This Manual

Peripheral Manual Communication Macro part has 8 chapters and APPENDIXES as shown below.

CHAPTER 1-1 : Multi-function Serial Interface
CHAPTER 1-2 : UART (Asynchronous Serial Interface)
CHAPTER 1-3 : CSIO (Clock Synchronous Serial Interface)
CHAPTER 1-4 : LIN Interface (Ver.2.1) (LIN Communication Control Interface Ver.2.1)
CHAPTER 1-5 : I2C Interface (I2C Communications Control Interface)
CHAPTER 2-1 : USB/Ethernet Clock Generation Block
CHAPTER 2-2 : USB Clock Generation
CHAPTER 2-3 : USB/Ethernet Clock Generation
CHAPTER 3-1 : USB Function
CHAPTER 3-2 : USB Host
CHAPTER 4 : Ethernet
CHAPTER 5-1 : CAN Prescaler
CHAPTER 5-2 : CAN Controller
CHAPTER 5-3 : CAN FD CONTROLLER
CHAPTER 5-4 : CAN FD Message RAM ECC Function
CHAPTER 5-5 : EXTERNAL TIME STAMP COUNTER FOR CAN FD
CHAPTER 6-1 : HDMI-CEC/Remote Control Reception
CHAPTER 6-2 : CEC Reception/Remote Reception
CHAPTER 6-3 : CEC Transmission
CHAPTER 7-1 : I2S Clock Generation
CHAPTER 7-2 : I2S (Inter-IC Sound Bus) Interface
CHAPTER 8-1 : High-Speed Quad Serial Peripheral Interface Configuration
CHAPTER 8-2 : High-Speed Quad Serial Peripheral Interface Prescaler
CHAPTER 8-3 : High-Speed Quad Serial Peripheral Interface Controller
APPENDIXES

Related Manuals

The manuals related to this family are listed below. See the manual appropriate to the applicable conditions.

The contents of these manuals are subject to change without notice. Contact us to check the latest versions available.

Peripheral Manual

- FM4 Family PERIPHERAL MANUAL
  (Called "PERIPHERAL MANUAL" hereafter)
- FM4 Family PERIPHERAL MANUAL Timer Part
  (Called "Timer Part" hereafter)
- FM4 Family PERIPHERAL MANUAL Analog Macro Part
  (Called "Analog Macro Part" hereafter)
- FM4 Family PERIPHERAL MANUAL Communication Macro Part (this manual)
  (Called "Communication Macro Part" hereafter)

Data sheet

For details about device-specific, electrical characteristics, package dimensions, ordering information etc., see the following document.

- 32-bit Microcontroller FM4 Family DATA SHEET

Note:

- *The data sheets for each series are provided.*
  See the appropriate data sheet for the series that you are using.

CPU Programming manual

For details about ARM Cortex-M4F core, see the following documents that can be obtained from http://www.arm.com/.

- Cortex-M4 Technical Reference Manual
- ARMv7-M Architecture Application Level Reference Manual

Flash Programming manual

For details about the functions and operations of the built-in flash memory, see the following document.

- FM4 Family FLASH PROGRAMMING MANUAL

Note:

- *Flash programming manuals for each series are provided.*
  See the appropriate flash programming manual for the series that you are using.

How to Use This Manual

Finding a function

The following methods can be used to search for the explanation of a desired function in this manual:

- Search from the table of the contents

  The table of the contents lists the manual contents in the order of description.

- Search from the register

  The address where each register is located is not described in the text. To verify the address of a register, see "A. Register Map" in "APPENDIXES".

About the chapters

Basically, this manual explains 1 peripheral function per chapter.

Terminology

This manual uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

Notations

- The notations in bit configuration of the register explanation of this manual are written as follows.
  - bit : bit number
  - Field : bit field name
  - Attribute : Attributes for read and write of each bit
    - R : Read only
    - W : Write only
    - R/W : Readable/Writable
    - - : Undefined
  - Initial value : Initial value of the register after reset
    - 0 : Initial value is "0"
    - 1 : Initial value is "1"
    - X : Initial value is undefined

- The multiple bits are written as follows in this manual.
  Example : bit7:0 indicates the bits from bit7 to bit0

- The values such as for addresses are written as follows in this manual.
  - Hexadecimal number : "0x" is attached in the beginning of a value as a prefix (example : 0xFFFF)
  - Binary number : "0b" is attached in the beginning of a value as a prefix (example: 0b1111)
  - Decimal number : Written using numbers only (example : 1000)

The target products in this manual

■ In this manual, the products are classified into the following groups and are described follows.
For the descriptions such as "TYPE1-M4", see the relevant items of the target product in the list below.

Table 1 TYPE1-FM4 Product list

Description in this manual	Flash memory size		
	1024 Kbytes	768 Kbytes	512 Kbytes
TYPE1-M4	MB9BF568M MB9BF568N MB9BF568R	MB9BF567M MB9BF567N MB9BF567R	MB9BF566M MB9BF566N MB9BF566R
	MB9BF468M MB9BF468N MB9BF468R	MB9BF467M MB9BF467N MB9BF467R	MB9BF466M MB9BF466N MB9BF466R
	MB9BF368M MB9BF368N MB9BF368R	MB9BF367M MB9BF367N MB9BF367R	MB9BF366M MB9BF366N MB9BF366R
	MB9BF168M MB9BF168N MB9BF168R	MB9BF167M MB9BF167N MB9BF167R	MB9BF166M MB9BF166N MB9BF166R

Table 2 TYPE2-FM4 Product list

Description in this manual	Flash memory size		
	512 Kbytes	384 Kbytes	256 Kbytes
TYPE2-M4	MB9BF566K MB9BF566L	MB9BF565K MB9BF565L	MB9BF564K MB9BF564L
	MB9BF466K MB9BF466L	MB9BF465K MB9BF465L	MB9BF464K MB9BF464L
	MB9BF366K MB9BF366L	MB9BF365K MB9BF365L	MB9BF364K MB9BF364L
	MB9BF166K MB9BF166L	MB9BF165K MB9BF165L	MB9BF164K MB9BF164L

Table 3 TYPE3-FM4 Product list

Description in this manual	Flash memory size		
	2 Mbytes	1.5 Mbytes	1 Mbytes
TYPE3-M4	S6E2CCA L0AGL20	S6E2CC9 L0AGL20	S6E2CC8 L0AGL20
	S6E2CCA LHAGL20	S6E2CC9 LHAGL20	S6E2CC8 LHAGL20
	S6E2CCAJ0AGV20	S6E2CC9J0AGV20	S6E2CC8J0AGV20
	S6E2CCAJHAGV20	S6E2CC9JHAGV20	S6E2CC8JHAGV20
	S6E2CCAJ0AGB10	S6E2CC9J0AGB10	S6E2CC8J0AGB10
	S6E2CCAJHAGB10	S6E2CC9JHAGB10	S6E2CC8JHAGB10
	S6E2CCAH0AGV20	S6E2CC9H0AGV20	S6E2CC8H0AGV20
	S6E2CCAHHAGV20	S6E2CC9HHAGV20	S6E2CC8HHAGV20
	S6E2C5A L0AGL20	S6E2C59 L0AGL20	S6E2C58 L0AGL20
	S6E2C5AJ0AGV20	S6E2C59J0AGV20	S6E2C58J0AGV20
	S6E2C5AJ0AGB10	S6E2C59J0AGB10	S6E2C58J0AGB10
	S6E2C5AH0AGV20	S6E2C59H0AGV20	S6E2C58H0AGV20
	S6E2C4A L0AGL20	S6E2C49 L0AGL20	S6E2C48 L0AGL20
	S6E2C4AJ0AGV20	S6E2C49J0AGV20	S6E2C48J0AGV20
	S6E2C4AJ0AGB10	S6E2C49J0AGB10	S6E2C48J0AGB10
	S6E2C4AH0AGV20	S6E2C49H0AGV20	S6E2C48H0AGV20
	S6E2C3A L0AGL20	S6E2C39 L0AGL20	S6E2C38 L0AGL20
	S6E2C3AJ0AGV20	S6E2C39J0AGV20	S6E2C38J0AGV20
	S6E2C3AJ0AGB10	S6E2C39J0AGB10	S6E2C38J0AGB10
	S6E2C3AH0AGV20	S6E2C39H0AGV20	S6E2C38H0AGV20
	S6E2C2A L0AGL20	S6E2C29 L0AGL20	S6E2C28L0AGL20
	S6E2C2A LHAGL20	S6E2C29 LHAGL20	S6E2C28LHAGL20
	S6E2C2AJ0AGV20	S6E2C29J0AGV20	S6E2C28J0AGV20
	S6E2C2AJHAGV20	S6E2C29JHAGV20	S6E2C28JHAGV20
	S6E2C2AJ0AGB10	S6E2C29J0AGB10	S6E2C28J0AGB10
	S6E2C2AJHAGB10	S6E2C29JHAGB10	S6E2C28JHAGB10
	S6E2C2AH0AGV20	S6E2C29H0AGV20	S6E2C28H0AGV20
	S6E2C2AHHAGV20	S6E2C29HHAGV20	S6E2C28HHAGV20
	S6E2C1AL0AGL20	S6E2C19L0AGL20	S6E2C18L0AGL20
	S6E2C1AJ0AGV20	S6E2C19J0AGV20	S6E2C18J0AGV20
	S6E2C1AJ0AGB10	S6E2C19J0AGB10	S6E2C18J0AGB10
	S6E2C1AH0AGV20	S6E2C19H0AGV20	S6E2C18H0AGV20

Table of Contents

CHAPTER1-1: Multi-function Serial Interface 21
1. Overview of the Multi-function Serial Interface 22
CHAPTER1-2: UART (Asynchronous Serial Interface) 23
1. Overview of UART (Asynchronous Serial Interface) 24
2. UART Interrupt 25
2.1. Received interrupt and flag set timing 26
2.2. Interrupt and flag set timing when received FIFO is used 28
2.3. Transmit interrupt and flag set timing 30
2.4. Interrupt and flag set timing when transmit FIFO is used 31
3. UART Operation 32
4. Dedicated Baud Rate Generator 40
4.1. Baud rate settings 41
5. Setting Procedure and Program Flow in Operation Mode 0 (Asynchronous Normal Mode) 46
6. Setting Procedure and Program Flow in Operation Mode 1 (Asynchronous Multiprocessor Mode) .. 49
7. UART (Asynchronous Serial Interface) Registers 53
7.1. Serial Control Register (SCR) 54
7.2. Serial Mode Register (SMR) 56
7.3. Serial Status Register (SSR) 58
7.4. Extended Communication Control Register (ESCR) 61
7.5. Received Data Register/Transmit Data Register (RDR/TDR) 63
7.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 65
7.7. FIFO Control Register 1 (FCR1) 66
7.8. FIFO Control Register 0 (FCR0) 68
7.9. FIFO Byte Register (FBYTE) 71
CHAPTER1-3: CSIO (Clock Synchronous Serial Interface) 73
1. Overview of CSIO (Clock Synchronous Serial Interface) 74
2. CSIO (Clock Synchronous Serial Interface) interrupts 75
2.1. Received interrupt and flag set timing 77
2.2. Interrupt and flag set timing when received FIFO is used 79
2.3. Transmit interrupt and flag set timing 81
2.4. Interrupt and flag set timing when transmit FIFO is used 82
2.5. Timer Interrupt Occurrence and Flag Setting Timing 83
2.6. Chip Select Error Occurrence and Flag Setting Timing 84
3. CSIO (Clock Synchronous Serial Interface) operations 86
3.1. Normal transfer (I) 86
3.2. Normal transfer (II) 96
3.3. SPI transfer (I) 106
3.4. SPI transfer (II) 116
4. Serial Timer Operation 126
5. Serial Chip Select Operation 131
6. Dedicated baud rate generator 141
6.1. Baud rate settings 142
6.2. CSIO (Clock Synchronous Serial Interface) setup procedure and program flow 145
7. CSIO (Clock Synchronous Serial Interface) registers 147
7.1. Serial Control Register (SCR) 150
7.2. Serial Mode Register (SMR) 153
7.3. Serial Status Register (SSR) 156
7.4. Extended Communication Control Register (ESCR) 158
7.5. Received Data Register/Transmit Data Register (RDR/TDR) 160
7.6. Serial Support Control Register (SACSR) 162
7.7. Serial Timer Register (STMR) 166

7.8. Serial Timer Comparison Register (STMCR) 167
7.9. Serial Chip Select Control Status Register (SCSCR) 168
7.10. Serial Chip Select Timing Register (SCSTR3:0) 172
7.11. Serial Chip Select Format Register (SCSFR2-0) 175
7.12. Transfer Byte Register (TBYTE0) 179
7.13. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 180
7.14. FIFO Control Register 1 (FCR1) 181
7.15. FIFO Control Register 0 (FCR0) 183
7.16. FIFO Byte Register (FBYTE) 186
8. Restrictions on CSIO (Clock Synchronous Serial Interface) 188
CHAPTER1-4: LIN Interface (Ver. 2.1)
(LIN Communication Control Interface Ver. 2.1) 189
1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1) 190
2. LIN Interface (Ver. 2.1) Interrupts 191
2.1. Received interrupt and flag set timing 192
2.2. Interrupt and flag set timing when received FIFO is used 194
2.3. Transmit interrupt and flag set timing 196
2.4. Interrupt and flag set timing when transmit FIFO is used 197
3. Dedicated Baud Rate Generator 198
3.1. Baud rate settings 199
4. LIN Interface (Ver. 2.1) Operations 204
5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow 216
6. LIN Interface (ver. 2.1) Registers 221
6.1. Serial Control Register (SCR) 222
6.2. Serial Mode Register (SMR) 225
6.3. Serial Status Register (SSR) 227
6.4. Extended Communication Control Register (ESCR) 230
6.5. Received Data Register/Transmit Data Register (RDR/TDR) 232
6.6. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 234
6.7. FIFO Control Register 1 (FCR1) 235
6.8. FIFO Control Register 0 (FCR0) 237
6.9. FIFO Byte Register (FBYTE) 240
CHAPTER1-5: I2C Interface (I2C Communications Control Interface) 243
1. Overview of I2C Interface (I2C Communications Control Interface) 244
2. I2C Interface interrupt 245
2.1. I2C interface operation 247
2.2. Master mode 248
2.3. Slave mode 280
2.4. Bus error 288
3. Dedicated Baud Rate Generator 289
4. I2C communication operation flowchart examples 291
5. I2C Interface Registers 298
5.1. I2C Bus Control Register (IBCR) 299
5.2. Serial Mode Register (SMR) 306
5.3. I2C Bus Status Register (IBSR) 308
5.4. Serial Status Register (SSR) 313
5.5. Received Data Register/Transmit Data Register (RDR/TDR) 317
5.6. Noise Filter Control Register (NFCR) 319
5.7. Received Data Register/Transmit Data Register (EIBCR) 320
5.8. 7-bit Slave Address Mask Register (ISMK) 322
5.9. 7-bit Slave Address Register (ISBA) 323
5.10. Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 324
5.11. FIFO Control Register 1 (FCR1) 325

5.12. FIFO Control Register 0 (FCR0) 327
5.13. FIFO Byte Register (FBYTE) 331
CHAPTER 2-1: USB/Ethernet Clock Generation Block 333
1. Overview and Configuration 334
CHAPTER 2-2: USB Clock Generation 335
1. Overview 336
2. Configuration and Block Diagram 337
3. Explanation of Operation 338
4. Setup Procedure Example 340
5. Register List 341
5.1. USB Clock Control Register (UCCR) 342
5.2. USB-PLL Control Register1 (UPCR1) 343
5.3. USB-PLL Control Register2 (UPCR2) 344
5.4. USB-PLL Control Register 3 (UPCR3) 345
5.5. USB-PLL Control Register 4 (UPCR4) 346
5.6. USB-PLL Control Register 5 (UPCR5) 347
5.7. USB-PLL Status Register (UP_STR) 348
5.8. USB-PLL Interrupt Source Enable Register (UPINT_ENR) 349
5.9. USB-PLL Interrupt Source Status Register (UPINT_STR) 350
5.10. USB-PLL Interrupt Source Clear Register (UPINT_CLR) 351
5.11. USB (ch.0) Enable Register (USBEN0) 352
5.12. USB (ch.1) Enable Register (USBEN1) 353
6. Usage Precautions 354
CHAPTER 2-3: USB/Ethernet Clock Generation 355
1. Overview 356
2. Configuration and Block Diagram 357
3. Description of operation 358
4. Example of setting procedure 360
5. List of Registers 361
5.1. USB/Ethernet Clock Setting Register (UCCR) 362
5.2. USB/Ethernet-PLL Setting Register1 (UPCR1) 363
5.3. USB/Ethernet-PLL Setting Register2 (UPCR2) 364
5.4. USB/Ethernet-PLL Setting Register3 (UPCR3) 365
5.5. USB/Ethernet-PLL Setting Register4 (UPCR4) 366
5.6. USB/Ethernet-PLL Setting Register5 (UPCR5) 367
5.7. USB/Ethernet-PLL Setting Register6 (UPCR6) 368
5.8. USB/Ethernet-PLL Setting Register7 (UPCR7) 369
5.9. USB/Ethernet-PLL State Register (UP_STR) 370
5.10. USB/Ethernet-PLL Interrupt Factor Enable Register (UPINT_ENR) 371
5.11. USB/Ethernet-PLL Interrupt Factor State Register (UPINT_STR) 372
5.12. USB/Ethernet-PLL Interrupt Factor Clear Register (UPINT_CLR) 373
5.13. USB (ch.0) Enable Register (USBEN0) 374
5.14. USB (ch.1) Enable Register (USBEN1) 375
6. Usage Precautions 376
CHAPTER 3-1: USB Function 377
1. Overview of USB Function 378
1.1. Features of USB function 378
2. Configuration of USB Function 379
3. Operations of USB Function 381
3.1. USB function operation 382
3.2. Detection of connection and disconnection 385
3.3. Operation of each register in response to a command 388
3.4. Suspend function 390

3.5. Wake-up function ... 391
3.6. DMA transfer function ... 393
3.7. NULL transfer function ... 397
3.8. STALL response/release of endpoint 0 ... 398
3.9. STALL response/release of endpoints 1 to 5 ... 400
4. Examples of USB Function Setting Procedures ... 404
5. USB Function Registers ... 411
5.1. UDC Control Register (UDCC) ... 413
5.2. EP0 Control Register (EP0C) ... 416
5.3. EP1 to EP5 Control Registers (EP1C to EP5C) ... 418
5.4. Time Stamp Register (TMSP) ... 421
5.5. UDC Status Register (UDCS) ... 422
5.6. UDC Interrupt Enable Register (UDCIE) ... 424
5.7. EP0I Status Register (EP0IS) ... 426
5.8. EP0O Status Register (EP0OS) ... 428
5.9. EP1 to EP5 Status Registers (EP1S to EP5S) ... 430
5.10. EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) ... 433
CHAPTER3-2: USB Host ... 435
1. Overview of USB host ... 436
2. USB host configuration ... 437
3. USB host operations ... 438
3.1. Device connection ... 439
3.2. USB bus resetting ... 441
3.3. Token packet ... 442
3.4. Data packet ... 444
3.5. Handshake packet ... 445
3.6. Retry function ... 446
3.7. SOF interrupt ... 447
3.8. Error status ... 449
3.9. End of packet ... 450
3.10. Suspend and resume operations ... 451
3.11. Device disconnection ... 453
4. USB host setting procedure examples ... 455
5. USB host registers ... 460
5.1. Host Control Registers 0 and 1 (HCNT0 and HCNT1) ... 462
5.2. Host Interrupt Register (HIRQ) ... 467
5.3. Host Error Status Register (HERR) ... 471
5.4. Host Status Register (HSTATE) ... 474
5.5. SOF Interrupt Frame Compare Register (HFCOMP) ... 477
5.6. Retry Timer Setup Register (HRTIMER) ... 478
5.7. Host Address Register (HADR) ... 479
5.8. EOF Setup Register (HEOF) ... 480
5.9. Frame Setup Register (HFRAME) ... 481
5.10. Host Token Endpoint Register (HTOKEN) ... 482
CHAPTER 4: Ethernet ... 485
CHAPTER 5-1: CAN Prescaler ... 487
1. Overview and configuration ... 488
2. CAN Prescaler Register ... 489
2.1. CAN Prescaler Register (CANPRE) ... 490
CHAPTER 5-2: CAN Controller ... 491
1. Overview ... 492
2. Configuration ... 493
3. CAN Controller Operations ... 494

3.1. Message objects 495
3.2. Message transmission 497
3.3. Message reception 500
3.4. FIFO buffer function 503
3.5. Interrupt function 505
3.6. Bit timing 506
3.7. Test mode 508
3.8. Software initialization 512
4. CAN Registers 513
4.1. CAN register functions 516
4.2. Total control registers 517
4.2.1. CAN Control Register (CTRLR) 518
4.2.2. CAN Status Register (STATR) 521
4.2.3. CAN Error Counter (ERRCNT) 523
4.2.4. CAN Bit Timing Register (BTR) 524
4.2.5. CAN Interrupt Register (INTR) 525
4.2.6. CAN Test Register (TESTR) 526
4.2.7. CAN Prescaler Extension Register (BRPER) 528
4.3. Message interface registers 529
4.3.1. IFx Command Request Register (IFxCREQ) 530
4.3.2. IFx Command Mask Register (IFxCMSK) 532
4.3.3. IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2) 537
4.3.4. IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2) 538
4.3.5. IFx Message Control Register (IFxMCTR) 539
4.3.6. IFx Data Registers A1, A2, B1, and B2 (IFxDTA1, IFxDTA2, IFxDTB1, and IFxDTB2) 540
4.4. Message objects 541
4.5. Message handler registers 547
4.5.1. CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2) 548
4.5.2. CAN New Data Registers 1, 2 (NEWDT1, NEWDT2) 550
4.5.3. CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2) 552
4.5.4. CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2) 554
5. Notes 555
CHAPTER 5-3: CAN FD CONTOROLLER 557
1. Overview 558
2. Configuration 559
2.1. Blockdiagram 559
2.2. Dual Clock Sources 560
2.3. Dual Interrupt Lines 560
3. Explanation of Operations 561
3.1. Operating Modes 561
3.1.1. Software Initialization 561
3.1.2. Normal Operation 563
3.1.3. CAN FD Operation 564
3.1.4. Transceiver Delay Compensation 565
3.1.5. Restricted Operation Mode 567
3.1.6. Bus Monitoring Mode 568
3.1.7. Disabled Automatic Retransmission 568
3.1.8. Power Down (Sleep Mode) 569
3.1.9. Test Modes 569
3.2. Timestamp Generation 571
3.3. Timeout Counter 571
3.4. Rx Handling 571

3.4.1. Acceptance Filtering 571
3.4.2. Rx FIFOs 575
3.4.3. Dedicated Rx Buffers 578
3.4.4. Debug on CAN Support 578
3.4.5. Protocol Exception Event 580
3.5. Tx Handling 581
3.5.1. Transmit Pause 581
3.5.2. Dedicated Tx Buffers 581
3.5.3. Tx FIFO 582
3.5.4. Tx Queue 583
3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO 583
3.5.6. Mixed Dedicated Tx Buffers/Tx Queue 584
3.5.7. Transmit Cancellation 584
3.5.8. Tx Event Handling 585
3.6. FIFO Acknowledge Handling 586
3.7. Configuring the CAN Bit Timing 587
3.7.1. CAN Bit Timing 587
3.7.2. CAN Bit Rates 589
4. Setup procedure examples 591
4.1. Configuration of CAN bus 593
4.2. Configuration of Message RAM 593
4.2.1. Configuration of ID Filter List 594
4.2.2. Configuration of Rx Buffer and Rx FIFO 595
4.2.3. Configuration of Tx Buffer and Tx FIFO/Queue 596
4.2.4. Configuration of ID Filter 597
4.3. Configuration of Error Monitor 598
4.4. Configuration of Interrupt 599
4.5. Control of Communication 600
4.5.1. CAN Transmit Mode Change 601
4.5.2. Configuration of Transmission Frame 602
4.6. Interrupt Handling Operation 603
4.6.1. Bus_Off status Handling Operation 604
4.6.2. Message RAM access failure Handling Operation 605
4.6.3. Bit Error Handling Operation 607
4.6.4. Tx Event FIFO Handling Operation 608
4.6.5. Dedicated Rx Buffer Handling Operation 609
4.6.6. High priority Message Handling Operation 610
4.6.7. Rx FIFO Handling Operation 611
5. Registers 612
5.1. Core Release Register (CREL) 614
5.2. Endian Register (ENDN) 615
5.3. Fast Bit Timing & Prescaler Register (FBTP) 616
5.4. Test Register (TEST) 618
5.5. RAM Watchdog (RWD) 620
5.6. CC Control Register (CCCR) 621
5.7. Bit Timing & Prescaler Register (BTP) 625
5.8. Timestamp Counter Configuration (TSCC) 628
5.9. Timestamp Counter Value (TSCV) 630
5.10. Timeout Counter Configuration (TOCC) 631
5.11. Timeout Counter Value (TOCV) 633
5.12. Error Counter Register (ECR) 634
5.13. Protocol Status Register (PSR) 636
5.14. Interrupt Register (IR) 639

5.15. Interrupt Enable (IE) ... 645
5.16. Interrupt Line Select (ILS) ... 647
5.17. Interrupt Line Enable (ILE) ... 649
5.18. Global Filter Configration (GFC) ... 650
5.19. Standard ID Filter Configuration (SIDFC) ... 652
5.20. Extended ID Filter Configuration (XIDFC) ... 653
5.21. Extended ID AND Mask (XIDAM) ... 654
5.22. High Priority Message Status (HPMS) ... 655
5.23. New Data 1 (NDAT1) ... 657
5.24. New Data 2 (NDAT2) ... 658
5.25. Rx FIFO Configuration (RXF0C) ... 659
5.26. Rx FIFO 0 Status (RXF0S) ... 661
5.27. Rx FIFO 0 Acknowledge (RXF0A) ... 663
5.28. Rx Buffer Configuration (RXBC) ... 664
5.29. Rx FIFO 1 Configuration (RXF1C) ... 665
5.30. Rx FIFO 1 Status (RXF1S) ... 667
5.31. Rx FIFO 1 Acknowledge (RXF1A) ... 669
5.32. Rx Buffer/FIFO Element Size Configuration (RXESC) ... 670
5.33. Tx Buffer Configuration (TXBC) ... 672
5.34. Tx FIFO/Queue Status (TXFQS) ... 674
5.35. Tx Buffer Element Size Configuration (TXESC) ... 676
5.36. Tx Buffer Request Pending (TXBRP) ... 677
5.37. Tx Buffer Add Request (TXBAR) ... 678
5.38. Tx Buffer Cancellation Request (TXBCR) ... 679
5.39. Tx Buffer Transmission Occurred (TXBTO) ... 680
5.40. Tx Buffer Cancellation Finished (TXBCF) ... 681
5.41. Tx Buffer Transmission Interrupt Enable (TXBTIE) ... 682
5.42. Tx Buffer Cancellation Finished Interrupt Enable (TXBCIE) ... 683
5.43. Tx Event FIFO Configuration (TXEFC) ... 684
5.44. Tx Event FIFO Status (TXEFS) ... 686
5.45. Tx Event FIFO Acknowledge (TXEFA) ... 688
6. Message RAM ... 689
6.1. Message RAM Configuration ... 689
6.2. Rx Buffer and FIFO Element ... 691
6.3. Tx Buffer Element ... 694
6.4. Tx Event FIFO Element ... 696
6.5. Standard Message ID Filter Element ... 698
6.6. Extended Message ID Filter Element ... 700
CHAPTER 5-4: CAN FD Message RAM ECC Function ... 703
1. Overview ... 704
2. Configuration ... 705
3. Interrupts ... 707
4. Explanation of Operation ... 708
4.1. Message RAM ECC Generation ... 709
4.2. Message RAM ECC Check ... 710
5. Setup precedure examples ... 711
6. Registers ... 712
6.1. CAN FD ECC Error Control Register (FDECR) ... 713
6.2. CAN FD ECC Error Status Register (FDESR) ... 715
6.3. CAN FD ECC Error Status Clear Register (FDESCR) ... 716
6.4. CAN FD ECC Double-bit Error Address Register (FDDEAR) ... 717
6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR) ... 718
7. Precations ... 719

CHAPTER 5-5: EXTERNAL TIME STAMP COUNTER FOR CAN FD 721
1. Overview 722
2. Configuration 723
3. Operations 724
3.1. Operation of Time Stamp Counter for CAN FD 724
3.1.1. Count up cycle for Time Stamp Counter 724
3.1.2. Time Stamp Counter 724
3.1.3. Clear Request 726
4. Example of the Operation 728
5. Register 729
5.1. Time Stamp Control Register (TSCNTR) 730
5.2. Time Stamp Mode Register (TSMDR) 731
5.3. Time Stamp Divider Register (TSDIVR) 732
5.4. Time Stamp Counter Data Register (TSCDTR) 733
5.5. Time Stamp Compare Clear Register (TSCPCLR) 734
CHAPTER 6-1: HDMI-CEC/Remote Control Reception 735
1. Configuration 736
CHAPTER 6-2: CEC Reception/Remote Reception 737
1. Overview 738
2. Configuration 739
3. Operations 740
3.1. SIRCS mode 740
3.1.1. Operational flow chart and waves of SIRCS mode 740
3.1.2. Basic operations of SIRCS mode 741
3.1.3. Start bit detection and interrupt output 741
3.1.4. Minimum pulse width violation 742
3.1.5. Device address comparison 742
3.1.6. Counter overflow detection and interrupt output 742
3.2. Operation of NEC/Association for Electric Home Appliance mode 743
3.2.1. Operational flow chart and waves of NEC/Association for Electric Home Appliance mode 743
3.2.2. Start bit detection 744
3.2.3. Repeat code detection 745
3.2.4. Minimum pulse width violation 745
3.2.5. Counter overflow detection and interrupt output 746
3.3. HDMI-CEC mode 747
3.3.1. Operational flow chart and waves in HDMI-CEC mode 747
3.3.2. Start bit detection and interrupt output 748
3.3.3. Minimum pulse width violation 749
3.3.4. Counter overflow detection and interrupt output 749
3.3.5. Device address comparison 750
3.3.6. Data bit width violation error pulse automatic output 750
3.3.7. EOM detection 751
3.3.8. ACK detection and interrupt output 751
3.4. Noise filter 751
4. Example of Setting 752
5. Registers 753
5.1. Reception Control Register (RCCR) 754
5.2. Reception Interrupt Control Register (RCST) 756
5.3. Device Address Setting Register 1, 2 (RCADR1, RCADR2) 758
5.4. Start Bit Detection Setting Register (RCSHW) 759
5.5. Minimum Pulse Width Setting Register (RCDAHW) 760
5.6. Threshold Value Setting Register (RCDBHW) 761

5.7. Data Save Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL) 762
5.8. Clock Division Setting Register (RCCKD) 763
5.9. Repeat Code Interrupt Control Register (RCRC) 764
5.10. Repeat Code Detection Width Setting Register (RCRHW) 765
5.11. Data Bit Width Violation Interrupt Control Register (RCLE) 766
5.12. Maximum Data Bit Width Setting Register (RCLELW) 768
5.13. Minimum Data Bit Width Setting Register (RCLESW) 769
CHAPTER 6-3: CEC Transmission 771
1. Overview of CEC Transmission 772
2. Block Diagram of CEC Transmitting Circuit 773
3. CEC Transmission Interrupts 774
4. CEC Transmission Registers 775
5. CEC Transmission Operations 776
5.1. CEC Transmission Operations 777
5.2. Interrupt Factors and Timing Chart 778
5.3. Arbitration Lost Detection 779
5.4. Signal Free Detection 780
5.5. Filtering 781
5.6. CEC Transmission Operations Flow 782
6. CEC Transmission Register Set 783
6.1. Transmission Control Register (TXCTRL) 784
6.2. Transmission Data Register (TXDATA) 786
6.3. Transmission Status Register (TXSTS) 787
6.4. Signal Free Time Setting Register (SFREE) 789
CHAPTER 7-1: I^{2}S Clock Generation 791
1. Overview 792
2. Configuration and Block Diagram 793
3. Explanation of Operation 794
4. Setup Procedure Example 795
5. Register List 796
5.1. I^{2}S Clock Control Register (ICCR) 797
5.2. I^{2}S-PLL Control Register1 (IPCR1) 798
5.3. I^{2}S-PLL Control Register2 (IPCR2) 799
5.4. I^{2}S-PLL Control Register3 (IPCR3) 800
5.5. I^{2}S-PLL Control Register4 (IPCR4) 801
5.6. I^{2}S-PLL Control Register5 (IPCR5) 802
5.7. I^{2}S-PLL Status Register (IP_STR) 803
5.8. I^{2}S-PLL Interrupt Facter Enable Register (IPINT_ENR) 804
5.9. I^{2}S-PLL Interrupt Facter Status Register (IPINT_STR) 805
5.10. I^{2}S-PLL Interrupt Facter Clear Register (IPINT_CLR) 806
6. Usage Precautions 807
CHAPTER 7-2: I^{2}S(Inter-IC Sound bus) Interface 809
1. Overview of I^{2}S Interface 810
2. Features of I^{2}S Interface 810
3. Block Diagram of I^{2}S Interface 811
4. I^{2}S Interface Operation Description 812
4.1. Clock and Frame Sync Signal 813
4.2. Transfer Start, Stop, and Error Operations 814
4.3. Frame Configuration 820
4.4. FIFO Configuration and Description 824
5. List of I^{2}S Interface Registers 826
6. Details of I^{2}S Interface Registers 827
6.1. Receive FIFO Register (RXFDAT) 828

6.2. Transmit FIFO Register (TXFDAT) 829
6.3. Control Register (CNTREG) 830
6.4. Channel Control Register 0 (MCR0REG) 834
6.5. Channel Control Register 1 (MCR1REG) 836
6.6. Channel Control Register 2 (MCR2REG) 837
6.7. Operation Control Register (OPRREG) 838
6.8. Soft Reset Register (SRST) 840
6.9. Interrupt Control Register (INTCNT) 841
6.10. Status Register (STATUS) 845
6.11. DMA Startup Register (DMAACT) 848
6.12. Test Register (TSTREG) 850
7. Application Notes 851
7.1. I^2S and MSB-Justified Protocol 851
CHAPTER 8-1: High-Speed Quad Serial Peripheral Interface Configuration 857
1. Configuration of HS_SPI 858
CHAPTER 8-2: High-Speed Quad Serial Peripheral Interface Prescaler 859
1. Overview 860
2. Explanation of Operations 861
2.1. Clock setting 861
2.2. DMA BRIDGE control 862
3. Register set of the HS-SPI prescaler 863
3.1. Registers 863
3.2. Division Clock Register (QDCLKR) 864
3.3. DMA BRIDGE control Register (DBCNT) 865
CHAPTER 8-3: High-Speed Quad Serial Peripheral Interface controller 867
1. Overview of the HS_SPI controller 868
1.1. Features of the HS_SPI controller 868
2. HS_SPI controller Operations 869
2.1. DMA Interface 870
2.1.1. DMA transfer toward TX-FIFO 870
2.1.2. DMA transfer toward RX-FIFO 872
2.2. Interrupts 874
2.2.1. Transmission interrupt requests 874
2.2.2. Receive interrupt requests 875
2.2.3. Violation detection interrupt requests 876
2.3. Serial Interface 877
2.3.1. Clocking Mode 877
2.3.2. Serial Interface Bit Width 879
2.3.3. Shift direction 879
2.3.4. Endian of serial interface 881
2.3.5. Deselect time 883
2.3.6. Timing-corrected clock 883
2.4. Direct mode 885
2.4.1. FIFO 885
2.4.2. AHB interface clock (QHCLK) and serial clock 885
2.4.3. Transfer control mode 886
2.4.4. Operation mode 887
2.4.5. Tri-state control of data output 888
2.5. Command Sequencer Mode 889
2.5.1. Allocation on the memory 889
2.5.2. 32-bit memory address 890
2.5.3. AHB interface clock (QHCLK) and serial clock 893
2.5.4. Command sequence 893

3. HS_SPI controller register 900
3.1. Register 900
3.1.1. Total control register list 902
3.1.2. Serial interface control register list 902
3.1.3. Fault register list 902
3.1.4. DMA control register list 902
3.1.5. FIFO control register list (common to TX/RX) 902
3.1.6. TX-FIFO control register list 903
3.1.7. RX-FIFO control register list 904
3.1.8. Direct mode control register list 904
3.1.9. Command sequencer control register list (common to write/read) 905
3.1.10. Command sequence set register list (write command) 905
3.1.11. Command sequence set register list (read command) 905
3.1.12. Module identification register list 905
3.2. HS-SPI control register (HSSPIn_MCTRL) 906
3.3. HS-SPI peripheral communication set register (HSSPIn_PCC0 to 3) 908
3.4. HS-SPI transmission interrupt factor register (HSSPIn_TXF) 913
3.5. HS-SPI transmission interrupt enable register (HSSPIn_TXE) 916
3.6. HS-SPI transmission interrupt clear register (HSSPIn_TXC) 918
3.7. HS-SPI reception interrupt factor register (HSSPIn_RXF) 920
3.8. HS-SPI reception interrupt enable register (HSSPIn_RXE) 923
3.9. HS-SPI reception interrupt clear register (HSSPIn_RXC) 925
3.10. HS-SPI fault interrupt factor register (HSSPIn_FAULTF) 927
3.11. HS-SPI fault interrupt clear register (HSSPIn_FAULTC) 930
3.12. HS-SPI direct mode set register (HSSPIn_DMCFG) 932
3.13. HS-SPI direct mode DMA enable register (HSSPIn_DMDMAEN) 933
3.14. HS-SPI direct mode transfer start control register (HSSPIn_DMSTART) 934
3.15. HS-SPI direct mode transfer stop control register (HSSPIn_DMSTOP) 935
3.16. HS-SPI direct mode slave select register (HSSPIn_DMPSEL) 936
3.17. HS-SPI direct mode transfer protocol set register (HSSPIn_DMTRP) 937
3.18. HS-SPI direct mode transfer byte count set register (HSSPIn_DMBCC) 938
3.19. HS-SPI direct mode transfer count remaining register (HSSPIn_DMBCS) 939
3.20. HS-SPI direct mode status register (HSSPIn_DMSTATUS) 940
3.21. HS-SPI TX-FIFO register (HSSPIn_TXFIFO0 to 15) 942
3.22. HS-SPI RX-FIFO register (HSSPIn_RXFIFO0 to 15) 943
3.23. HS-SPI FIFO set register (HSSPIn_FIFOCFG) 944
3.24. HS-SPI command sequencer set register (HSSPIn_CSCFG) 947
3.25. HS-SPI command sequencer idle timer set register (HSSPIn_CSITIME) 950
3.26. HS-SPI command sequencer address extend register (HSSPIn_CSAEXT) 951
3.27. HS-SPI read command sequence data/control register (HSSPIn_RDCSDC0 to 7) 952
3.28. HS-SPI write command sequence data/control register (HSSPIn_WRCSDC0～7) 954
3.29. HS-SPI module identification register (HSSPIn_MID) 956
4. Precautions for using the HS_SPI controller 957
4.1. General precautions for usage 957
5. Setting procedure example of the HS_SPI controller 959
5.1. Direct mode, command sequencer mode 959
5.1.1. Initial settings 959
5.1.2. Data read for serial memory (direct mode: data transfer by interrupt) 960
5.1.3. Data read for serial memory (direct mode: DMA transfer) 961
5.1.4. Data write to serial memory (direct mode: data transfer by interrupt) 962
5.1.5. Data write to serial memory (direct mode: DMA transfer) 963
5.1.6. Data read for serial memory (command sequencer mode) 964
5.1.7. Data write to serial memory (command sequencer mode) 964

5.2. Setting during DMA transfer by DSTC 965
5.2.1. Setting during reception DMA transfer 965
5.2.2. Setting during transmission DMA transfer 966
APPENDIXES 967
A. Register Map 968
1. Register Map 970
1.1 FLASH_IF 971
1.1.1 TYPE1-M4, TYPE2-M4 products 971
1.1.2 TYPE3-M4 products 972
1.2 Unique ID 973
1.3 ECC Capture Address 973
1.4 Clock/Reset 974
1.5 HW WDT 976
1.6 SW WDT 976
1.7 Dual Timer 977
1.8 MFT 978
1.8.1 TYPE1-M4, TYPE2-M4 products 978
1.8.2 TYPE3-M4 products 981
1.9 PPG 984
1.10 Base Timer 987
1.11 IO Selector for Base Timer 988
1.12 QPRC 989
1.12.1 TYPE1-M4, TYPE2-M4 products 989
1.12.2 TYPE3-M4 products 990
1.13 QPRC NF 991
1.14 A/DC 992
1.15 CR Trim 993
1.16 EXTI 993
1.17 INT-Req. READ 994
1.17.1 TYPE1-M4, TYPE2-M4 products 994
1.17.2 TYPE3-M4 products 1000
1.18 D/AC 1007
1.19 HDMI-CEC 1008
1.20 GPIO 1009
1.20.1 TYPE1-M4, TYPE2-M4 products 1009
1.20.2 TYPE3-M4 products 1016
1.21 LVD 1025
1.22 DS_Mode 1025
1.23 USB Clock 1026
1.24 CAN_Prescaler 1027
1.25 MFS 1027
1.26 CRC 1029
1.27 Watch Counter 1029
1.28 RTC 1030
1.29 Low-speed CR Prescaler 1033
1.30 Peripheral Clock Gating 1034
1.30.1 TYPE1-M4, TYPE2-M4 products 1034
1.30.2 TYPE3-M4 products 1034
1.31 I2S_Prescaler 1035
1.32 EXT-Bus I/F 1036
1.32.1 TYPE1-M4 products 1036
1.32.2 TYPE3-M4 products 1039
1.33 USB 1042

1.34 DMAC 1044
1.35 DSTC 1046
1.36 CAN 1048
1.37 Ethernet-MAC 1050
1.38 Ethernet-Control 1050
1.39 I2S 1051
1.40 SD-Card 1051
1.41 CAN FD 1052
1.42 Programmable-CRC 1055
1.43 WorkFlash_IF 1055
1.44 Hi-Speed Quad SPI controller 1056
B List of Notes 1059
1. Notes when high-speed CR is used for the master clock 1060
MAJOR CHANGES 1061

CHAPTER 1-1: Multi-function Serial Interface

This chapter describes the overview of the multi-function serial interface.

1. Overview of the Multi-function Serial Interface

CODE: 9BFMFS_FM4-E01.0

1. Overview of the Multi-function Serial Interface

This multi-function serial interface has the following characteristics.

Interface Mode

The following interface modes are selectable for the multi-function serial interface depending on the operation mode settings.

- UART0 (Asynchronous normal serial interface)
- UART1 (Asynchronous multi-processor serial interface)
- CSIO (Clock synchronous serial interface) (SPI can be supported)
- LIN(LIN bus interface)
- I^2C (I^2C bus interface)

Note:

- *See Chapters "UART(Asynchronous normal serial interface)", "CSIO (Clock synchronous serial interface) (SPI can be supported)", "LIN(LIN bus interface)" and "I^2C (I^2C bus interface)" for details about each interface.*

Switching the Interface Mode

To communicate through each serial interface, the serial mode register (SMR) shown in Table 1-1 should be used to set the operation mode before starting the communication.

Table 1-1 Switching Interface Mode

MD2	MD1	MD0	Interface mode
0	0	0	UART0 (Asynchronous normal serial interface)
0	0	1	UART1 (Asynchronous multi-processor serial interface)
0	1	0	CSIO (Clock synchronization serial interface) (SPI can be supported)
0	1	1	LIN(LIN bus interface)
1	0	0	I^2C (I^2C bus interface)
Values other than the above			Setting is prohibited.

Notes:

- *Transmission and reception cannot be guaranteed when the operation mode is switched while one of the serial interfaces is still in use for transmission or reception operation.*
- *To switch the current operation mode, issue a programmable clear (SCR:UPCL=1) or disable the I^2C (ISMK:EN=0) , and switch the operation mode continuously. After the operation mode is set, set each register.*
- *The settings not listed in Table 1-1 are prohibited.*

Transmission/Reception FIFO

This function has a 64-BYTE transmission FIFO and 64-BYTE reception FIFO. The FIFO capacity should be converted to 64 bytes when reading through this text.

LIN Sync field Detection: LSYN

To use an ICU in the LIN bus interface mode, use the ICU of the multifunction timer.
For switching an input to an ICU, see the section for Extended Function Pin Setting Register in the chapter "I/O PORT" in "PERIPHERAL MANUAL".

CHAPTER 1-2: UART (Asynchronous Serial Interface)

This chapter explains the UART (asynchronous serial interface) function supported in operation mode 0 and 1 of the multifunction serial interface.

1. Overview of UART (Asynchronous Serial Interface)
2. UART Interrupt
3. UART Operation
4. Dedicated Baud Rate Generator
5. Setting Procedure and Program Flow in Operation Mode 0 (Asynchronous Normal Mode)
6. Setting Procedure and Program Flow in Operation Mode 1 (Asynchronous Multiprocessor Mode)
7. UART (Asynchronous Serial Interface) Registers

CODE: 9BFUART_FM4-E01.0_FM15U-J05.4

1. Overview of UART (Asynchronous Serial Interface)

UART (asynchronous serial interface) is a general-purpose serial data communications interface for asynchronous communications (start/stop synchronization) with external devices. It supports a bi-directional communications function (normal mode) and a master/slave type communications function (multi-processor mode: both master and slave modes supported). It also has transmit /received FIFO installed.

Functions of UART (Asynchronous Serial Interface)

		Function
1	Data	− Full duplex double buffer (when FIFO is not used) − Transmit /received FIFO (size: max 128 bytes each) *1 (when FIFO is used)
2	Serial input	− Run oversampling three times with the bus clock and determine the value of received data based on the majority sampling value.
3	Transfer system	Asynchronous
4	Baud rate	− A dedicated baud rate generator (constructed with a 15-bit reload counter) − The external clock input can be adjusted with the reload counter.
5	Data length	− 5 to 9 bits (in normal mode)/7 bits or 8 bits (in multiprocessor mode)
6	Signaling system	NRZ (Non Return to Zero), inverted NRZ
7	Start bit detection	− In synch with the falling edge of the start bit (in the NRZ system) − In synch with the rising edge of the start bit (in the inverted NRZ system)
8	Received error detection	− Framing error − Overrun error − Parity error *2
9	Hardware flow control	CTS/RTS-based automatic transmit /received control *3
10	Interrupt request	− Received interrupt (upon reception completed, framing error, overrun error or parity error*2) − Transmit interrupts (transmit data empty, transmit bus idle) − Transmit FIFO interrupt (when transmit FIFO is empty) − DMA(Transmit /Received) transferring support function is available.
11	Master/slave communications functions (in multiprocessor mode)	One (master)-to-n (slaves) communication is enabled. (Both master and slave systems are supported.)
12	FIFO options	− Transmit /received FIFO installed (maximum capacity: 128 bytes for transmit FIFO, 128 bytes for received FIFO) *1 − Transmit FIFO or received FIFO can be selected. − Transmit data can be resent. − Received FIFO interrupt timing can be changed via software. − FIFO resetting is supported independently.

*1 : The FIFO capacity size varies depending on the product type.

*2 : Parity errors are only generated in normal mode.

*3 : The channel number, which the hardware flow control input/output(RTS/CTS) can be used, is dependent on the product type. See “Data Sheet” of the product used.

2. UART Interrupt

UART generates transmit or received interrupts. These interrupt requests can be generated if:

- Received data is set in the Received Data Register (RDR) or a data received error occurs.
- Transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register and the data transmission is started.
- The transmit bus is idle (No data transmission occurs).
- Transmit FIFO data is requested.

UART Interrupt

Table 2-1 shows the relationships between the UART interrupt control bits and the interrupt factors.

Table 2-1 UART interrupt control bits and interrupt factors

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt factor	Interrupt factor enable bit	Operation to clear interrupt request flag
			0	1			
Received	RDRF	SSR	○	○	A single-byte received	SCR:RIE	Reading from the received data register (RDR)
					Received of a data volume matching the value set for FBYTE.		Reading from the Received Data Register (RDR) until received FIFO is emptied
					While the FRIIE bit is "1" and the received FIFO contains valid data, a received idle state continues for 8 bits or longer period.		
	ORE	SSR	○	○	Overrun error		Setting the received error flag clear bit (SSR:REC) to "1"
	FRE	SSR	○	○	Framing error		
	PE	SSR	○	x	Parity error		
Transmit	TDRE	SSR	○	○	The Transmit Data Register is empty	SCR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *1
	TBI	SSR	○	○	No data transmission	SCR:TBIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *1
	FDRQ	FCR1	○	○	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" or transmit FIFO is full.

*1 : Set the TIE bit to "1" only after the TDRE bit has been set to "0".

2.1 Received interrupt and flag set timing

Data reception can be interrupted by a Received Completion (SSR:RDRF=1) or a Received Error Occurrence (SSR:PE,ORE,FRE=1).

Received interrupt and flag set timing

Upon detection of the first stop bit, received data are stored in the Received Data Register (RDR). When the data received is completed (SSR:RDRF=1) or when a data received error occurs (SSR:PE, ORE, FRE=1), each flag is set. If received interrupts are enabled (SSR:RIE=1) then, a received interrupt occurs.

Note:

- *If a received error occurs, data in the Received Data Register (RDR) becomes invalid.*

Figure 2-1 RDRF (Received Data Register Full) flag bit set timing

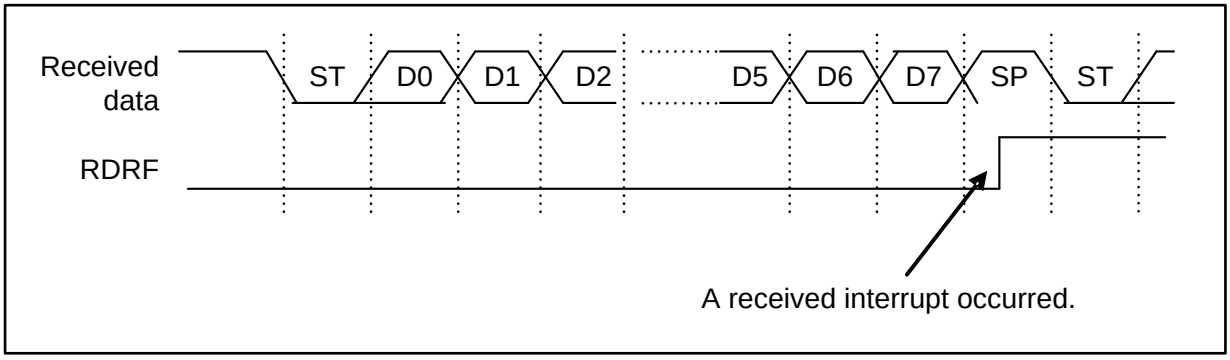
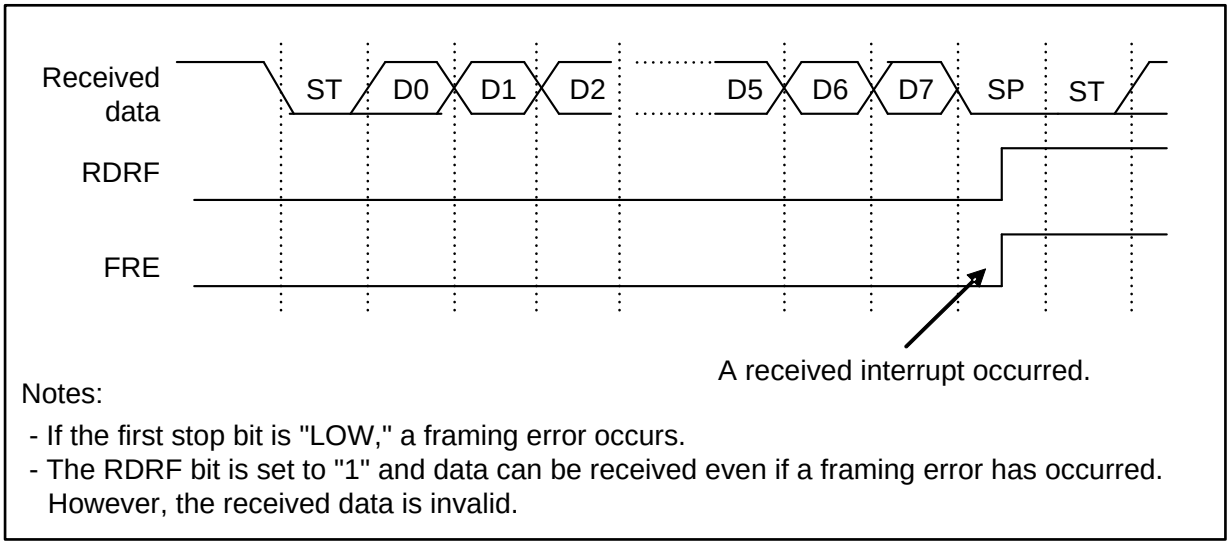


Figure 2-2 FRE (Framing Error) flag bit set timing

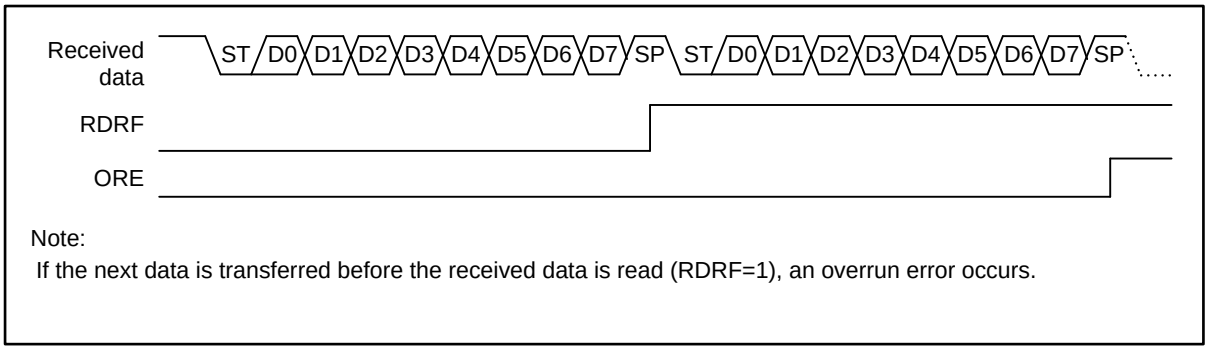


Note:

During reception, if the following is detected at the same time as the stop bit sampling point or before the 1 to 2 bus clocks, the relevant edge becomes invalid, which may disable normal received of the next data. To output frames continuously, adequate intervals are required between frames.

- *The falling edge of serial data (When ESCR:INV=0)*
- *The rising edge of serial data (When ESCR:INV=1)*

Figure 2-3 ORE (Overrun Error) flag bit set timing



Note:
If the next data is transferred before the received data is read (RDRF=1), an overrun error occurs.

2.2 Interrupt and flag set timing when received FIFO is used

If the received FIFO is used, an interrupt occurs when the FBYTE data (preset for the FBYTE register) is received.

Interrupt and flag set timing when received FIFO is used

- If the received FIFO is used, an interrupt occurs depending on the value set for the FBYTE register.
- When full FBYTE data is received, the received data full flag (SSR:RDRF) of the Serial Status register is set to "1". If received interrupts are enabled (SCR:RIE) during this time, a received interrupt occurs.
- If the following two conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the receive data full flag (SSR:RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FCR:FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.
- If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to zero (0).If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.
- When data is read from the Received Data Register (RDR) until received FIFO is emptied, the received data full flag (SSR:RDRF) is cleared.
- If the valid received data amount is the same as the FIFO capacity and if the next data is received, an overrun error (SSR:ORE=1) occurs.

Figure 2-4 Received interrupt timing when Received FIFO is used

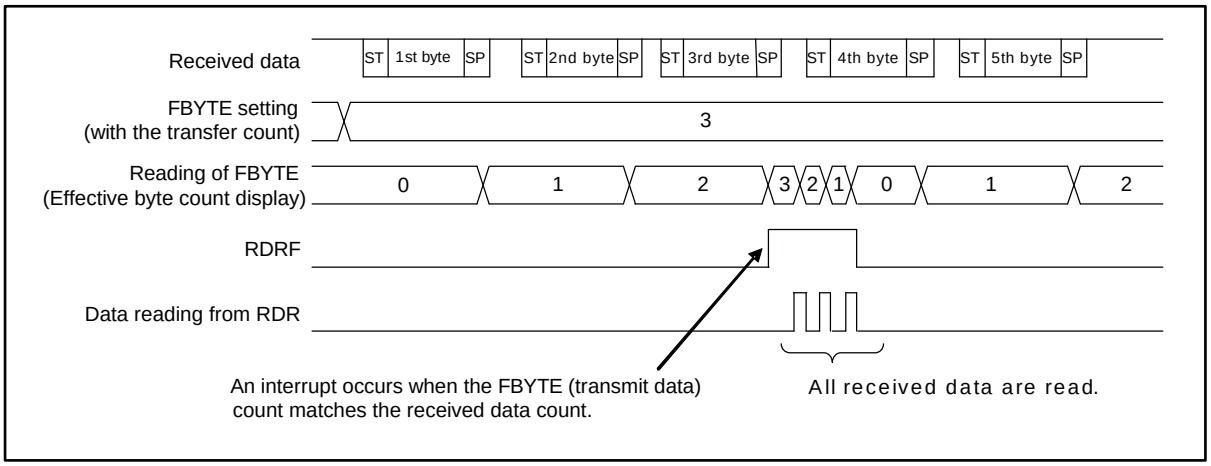
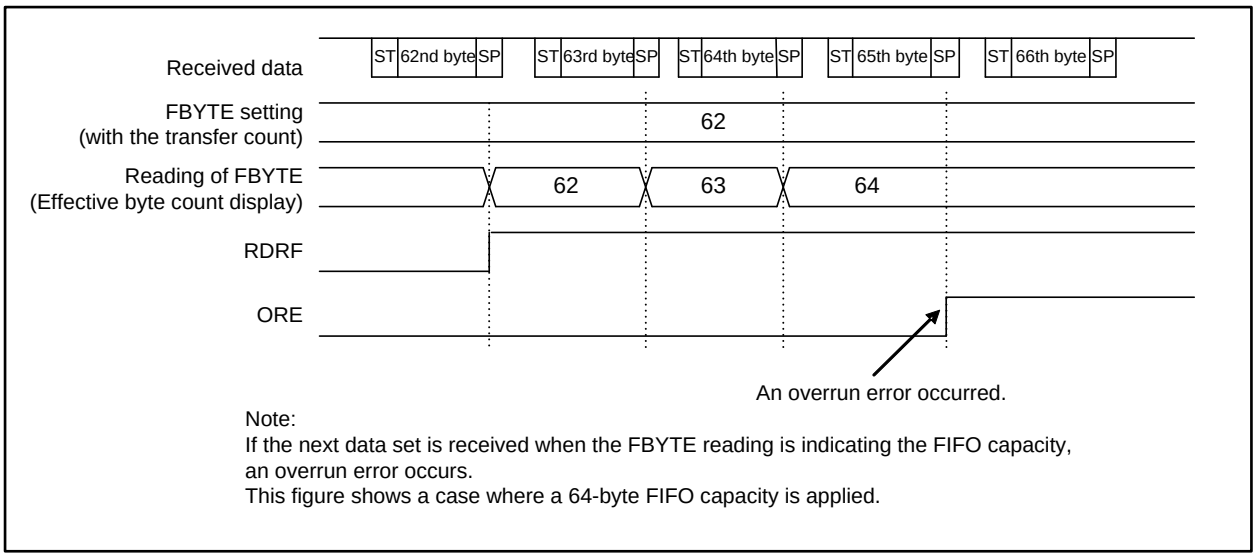


Figure 2-5 ORE (Overrun Error) flag bit set timing



Note:
If the next data set is received when the FBYTE reading is indicating the FIFO capacity, an overrun error occurs.
This figure shows a case where a 64-byte FIFO capacity is applied.

2.3 Transmit interrupt and flag set timing

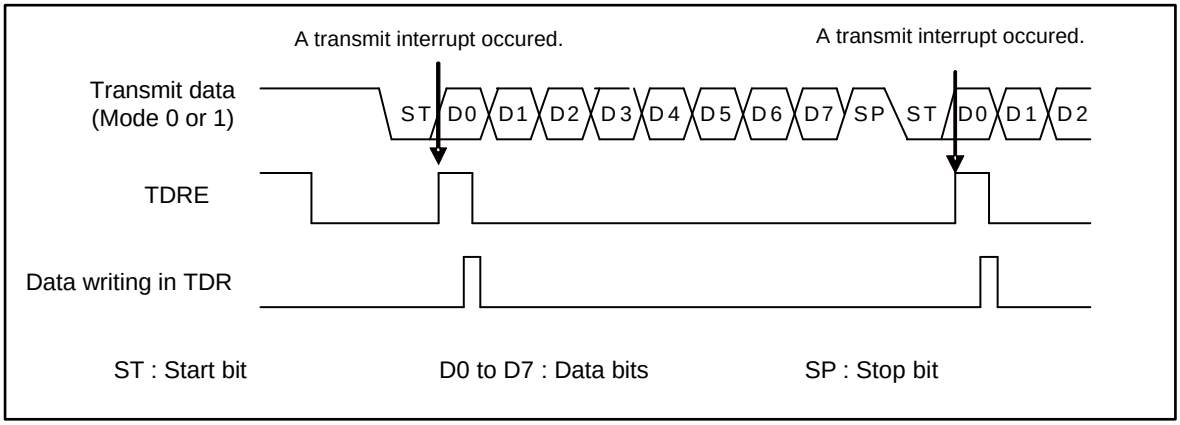
A transmit interrupt occurs when transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE = 1) and transmission starts and when no transmission is performed (SSR:TBI = 1).

Transmit interrupt and flag set timing

- Transmit data empty flag (SSR:TDRE) set timing

After data has been transferred from the Transmit Data Register (TDR) to the transmit shift register, the next data can be written in the TDR (SSR:TDRE = 1). If transmit interrupts are enabled (SCR:TIE = 1) during this time, a transmit interrupt occurs. As the SSR:TDRE bit is read only, the SSR:TDRE bit is cleared to "0" when data is written to the Transmit Data Register (TDR).

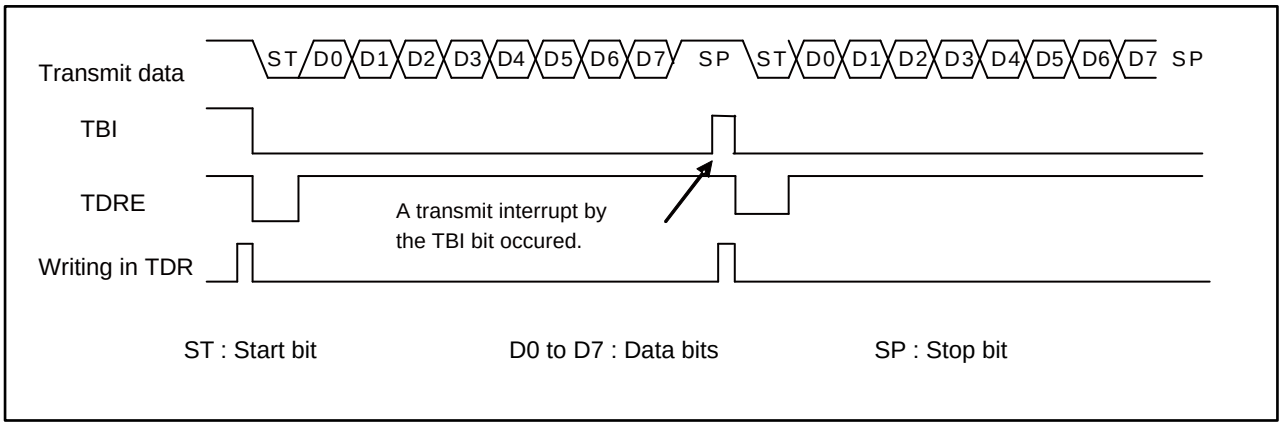
Figure 2-6 Transmit data empty flag (SSR:TDRE) set timing



- Transmit bus idle flag (SSR:TBI) set timing

If the Transmit Data Register is empty (SSR:TDRE=1) and no data is transmitted, the SSR:TBI bit is set to "1". If transmit bus idle interrupts are enabled (SCR:TBIE = 1) during this time, a transmit interrupt occurs. When transmit data is written to the Transmit Data Register (TDR), both the SSR:TBI bit and the transmit interrupt request are cleared.

Figure 2-7 Transmit bus idle flag (TBI) set timing



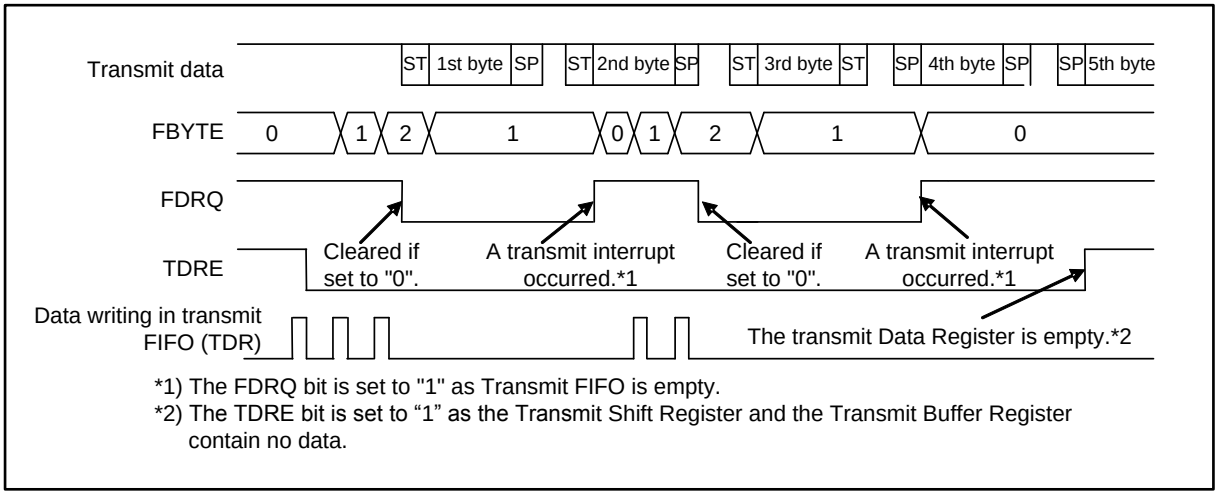
2.4 Interrupt and flag set timing when transmit FIFO is used

When the transmit FIFO is used, an interrupt occurs if the FIFO contains no data.

Transmit interrupt and flag set timing when transmit FIFO is used

- If the Transmit FIFO contains no data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "1". If FIFO transmit interrupts are enabled (FCR1:FTIE=1), a transmit interrupt occurs.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" when transmit FIFO becomes full.
- To check to see if transmit FIFO contains any data, read from the FIFO Byte Register (FBYTE). If FBYTE=0x00, no data exists in the transmit FIFO.

Figure 2-8 Transmit interrupt timing when transmit FIFO is used



*1) The FDRQ bit is set to "1" as Transmit FIFO is empty.
*2) The TDRE bit is set to “1” as the Transmit Shift Register and the Transmit Buffer Register contain no data.

3. UART Operation

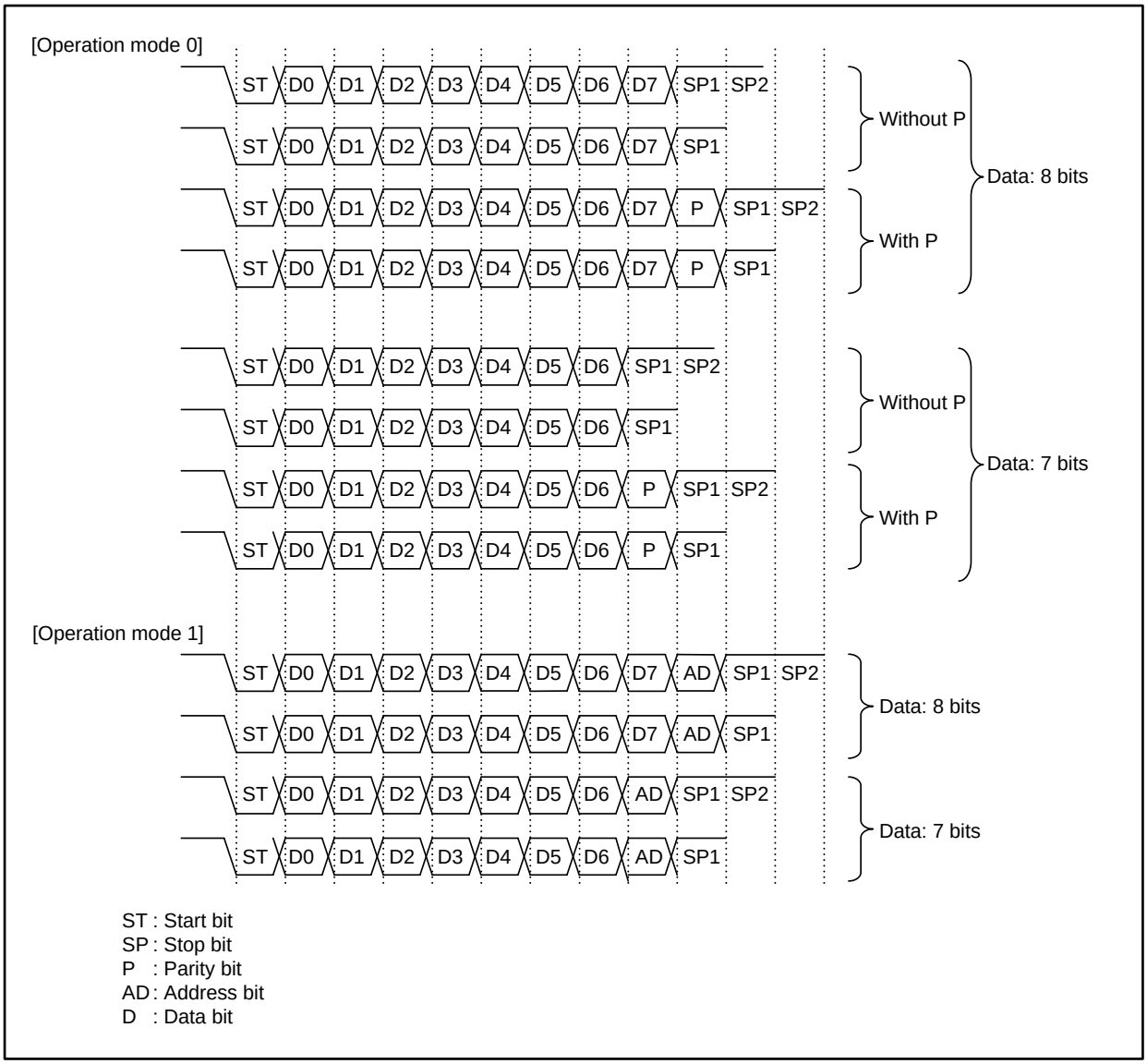
UART operates in bi-directional serial asynchronous communications in mode 0 and master/slave multiprocessor communications in mode 1.

UART operation

- Transmit/received data format
  - Transmit/received data always starts with a start bit, followed by transmit/received of data with the specified data bit length, and ends with at least one-bit long stop bit.
  - The BDS bit of the Serial Mode Register (SMR) determines the data transmission direction (LSB first or MSB first). If parity is used, the parity bit is always placed between the last data bit and the first stop bit.
  - In operation mode 0 (normal mode), selection is possible to use or not to use parity.
  - In operation mode 1 (multiprocessor mode), no parity is added, and instead, the AD bit is added.

Figure 3-1 shows the transmit/received data formats for operation mode 0 and 1.

Figure 3-1 Example transmit/received data format (operation mode 0/1)



Notes:

- *The above figure shows formats when the data length is set to 7 or 8 bits. (In operation mode 0, the data length can be set between 5 and 9 bits.)*
- *If the BDS bit of the Serial Mode Register (SMR) is set to "1" (MSB first), the bits are processed from D7, and then D6, D5, ... D1, and D0 (P), in that order.*
- *If the data length is set to X bits, the lower X bit of the Transmit/Received Data Register (TDR/RDR) is enabled.*

■ Data transmission

If the transmit data empty flag bit (TDRE) of the Serial Status Register (SSR) is "1", the transmit data can be written in the Transmit Data Register (TDR). (When transmit FIFO is enabled, transmit data can be written even if TDRE=0.)

If transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag bit (SSR:TDRE) is set to "0".

Setting the transmission enable bit of the serial control register (SCR:TXE) to "1" causes transmit data to be loaded to the transmit shift register, followed by sequential transmission starting with the start bit.

When transmission starts, the transmit data empty flag bit (SSR:TDRE) is set to "1" again. If transmit interrupts are then enabled (SCR:TIE=1), a transmit interrupt is generated. In the interrupt processing, the next transmit data set can be written in the Transmit Data Register,

Notes:

- *As the transmit data empty flag bit (SSR:TDRE) is initially set to "1", a transmit interrupt occurs as soon as transmit interrupts are enabled (SCR:TIE).*
- *As the FIFO transmit data request bit (FCR1:FDRQ) is initially set to "1", a transmit interrupt occurs as soon as FIFO transmit interrupts are enabled (FCR1:FTIE=1).*

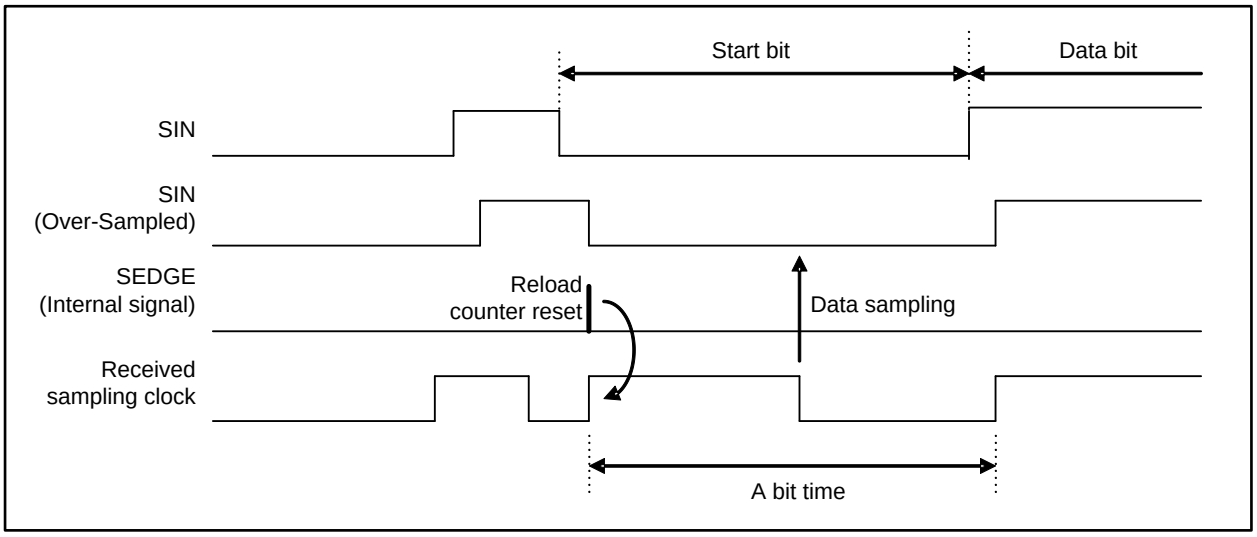
■ Data reception

- When reception is enabled (SCR:RXE=1), the interface performs reception.
- Upon detection of the start bit, one-frame data reception takes place according to the data format set in the extended communications control register (ESCR:PEN, P, L2, L1, L0) and serial mode register (SMR:BDS). A start bit is detected when falling (ESCR:INV=0) is detected after passing the noise filter (with the majority value applied after sampling serial data input three times with the bus clock) or if rising (ESCR:INV=1) is detected and "LOW" is detected for the data passing the sampling point.
- When one-frame reception is completed, the received data full flag bit (SSR:RDRF) is set to "1". If received interrupts are then enabled (SCR:RIE=1), a received interrupt is generated.
- To read received data, perform reading of the received data after one-frame data received is completed and check the state of the error flag of the Serial Status Register (SSR). Handle the received error if it is occurring.
- Reading of the received data causes the received data full flag bit (SSR:RDRF) to be cleared to "0".
- If received FIFO is enabled, the received data full flag bit (SSR:RDRF) is set to "1" when the number of received frames has reached the value set for received FBYTE.
- If the following two conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the interrupt flag (RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.
    If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to zero (0). If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.
- If received FIFO is enabled, received FIFO does not store data in which an error has occurred when the error flag of the Serial Status Register (SSR) is set to "1". Also note that the received data full flag bit (SSR:RDRF) is not set to "1". (However, the RDRF flag is set to "1" in an overrun error.) What the received FBYTE indicates is the number of data sets received normally before the error occurred. Unless the error flag of the Serial Status Register (SSR) is cleared to "0", received FIFO is not enabled.
- If received FIFO is enabled, the received data full flag bit (SSR:RDRF) is cleared to "0" when all data in received FIFO is out.

Notes:

- *Data in the Received Data Register (RDR) becomes valid when the received data register full flag bit (SSR:RDRF) is set to "1" and no received error occurs (SSR:PE, ORE, FRE=0).*
- *Although a noise filter is built in (with the majority value applied after sampling serial data input three times with the bus clock), wrong data may be received if any noise passes through the filter. As a countermeasure, you can design the board so as not to allow noise to pass through this filter or perform communications so that noise that has passed may not cause any problem (by adding check sum of data at the end and resending the data if any error occurs, for example).*
- *During reception, if the following is detected at the same time as the stop bit sampling point or before the 1 to 2 bus clocks, the relevant edge becomes invalid, which may disable normal reception of the next data. To output frames continuously, adequate intervals are required between frames.*
- *The falling edge of serial data (When ESCR:INV=0)*
- *The rising edge of serial data (When ESCR:INV=1)*

- Clock selection
  - You can use either an internal or external clock.
  - To use the external clock, set SMR:EXT to "1". IN this case, the external clock is subject to frequency division by the baud rate generator.
- Start bit detection
  - In asynchronous mode, the start bit is recognized based on detection of the falling edge of the SIN signal.
    For that reason, reception is not started unless the falling edge of the SIN signal is input even if reception is enabled (SCR:RXE=1).
  - Upon detection of the start bit's falling edge, the received reload counter of the baud rate generator is reset and reloaded to start countdown. Thus, sampling always takes place in the middle of data.

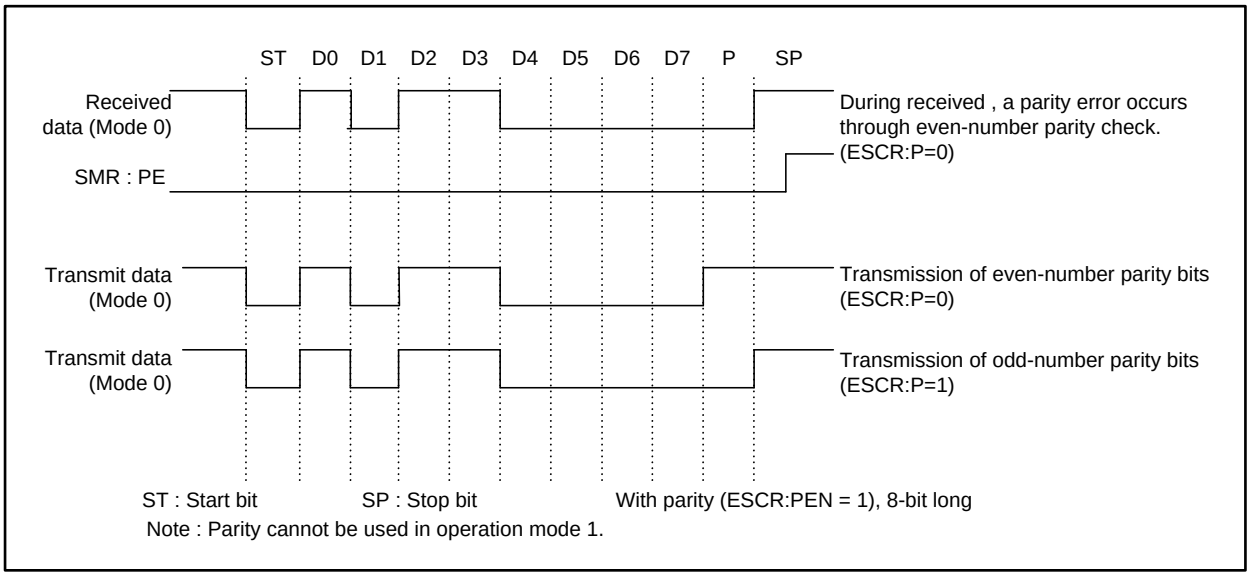


- Stop bit
  - You can select the bit length to be between one and four.
  - The received data full flag bit (SSR:RDRF) is set to "1" upon detection of the first stop bit.
- Error detection
  - In operation mode 0, parity, overrun and framing errors can be detected.
  - In operation mode 1, overrun and framing errors can be detected but parity errors cannot be detected.

- Parity bit
  - The parity bit can only be added in operation mode 0. The parity enable bit (ESCR:PEN) can be used to specify use or non-use of parity and the parity selection bit (ESCR:P) to set even-number parity or odd-number parity.
  - Parity cannot be used in operation mode 1.

Figure 3-2 shows transmit/received data when parity is enabled.

Figure 3-2 Operation when parity is enabled

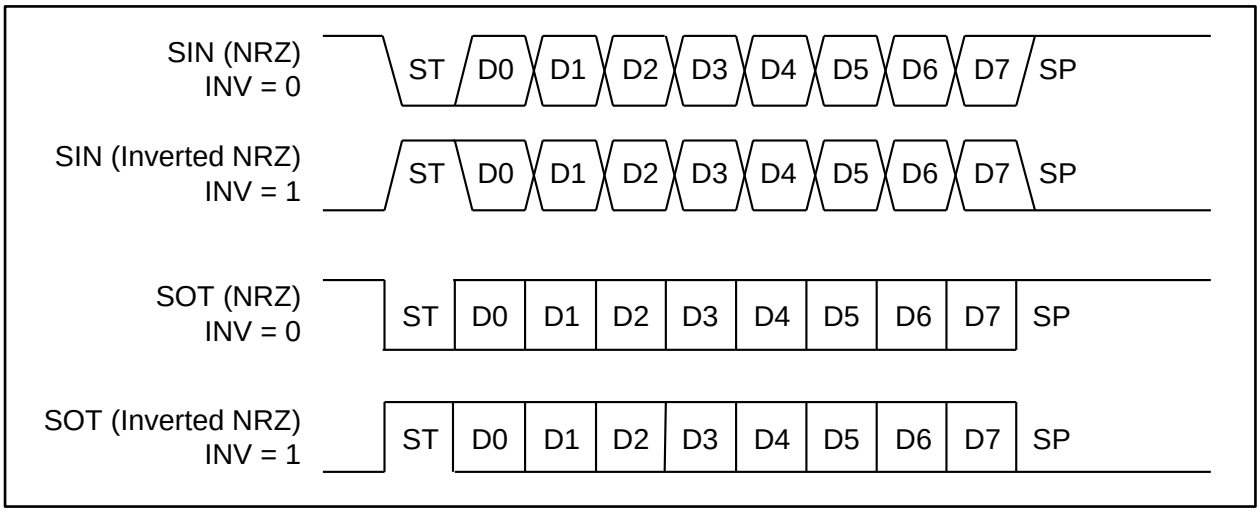


- Data signaling system

By setting up the INV bit of the extended communications control register, you can select either the NRZ (Non Return to Zero) signaling system (ESCR:INV=0) or inverted NRZ signaling system (ESCR:INV=1).

Figure 3-3 shows the NRZ and inverted NRZ signaling systems.

Figure 3-3 NRZ (Non Return to Zero) signaling system and inverted NRZ signaling system



- Data transfer system

As for the data bit transfer method, either LSB first or MSB first can be selected.

■ Hardware flow control

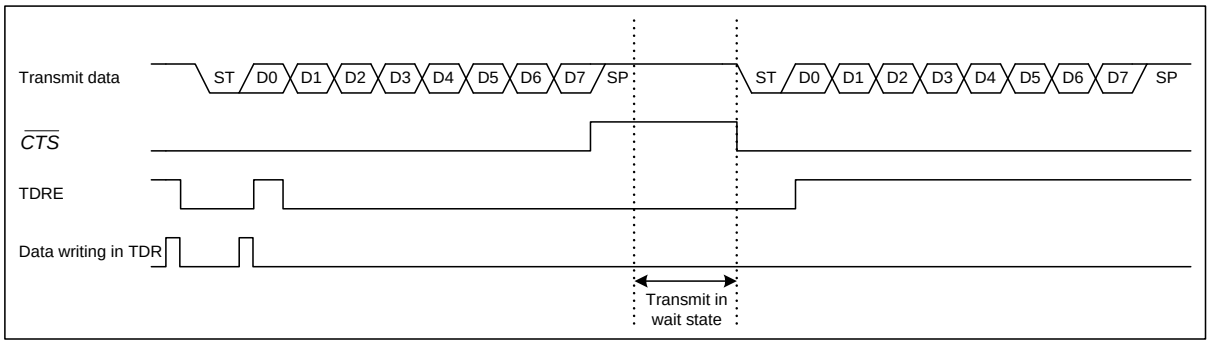
When flow control is enabled (ESCR:FLWEN=1), UART performs hardware flow control.

- During data transmission

  If $\overline{CTS}$ is "HIGH" after data is transmitted, the next data is not transmitted even if the transmit buffer contains data (TDRE=0) and the process waits until $\overline{CTS}$ is set to "LOW". To have transmission wait, input "HIGH" in $\overline{CTS}$ before the stop bit transmission is completed.

  Transmission continues up to the stop bit even if "HIGH" is input in $\overline{CTS}$ during transmission.

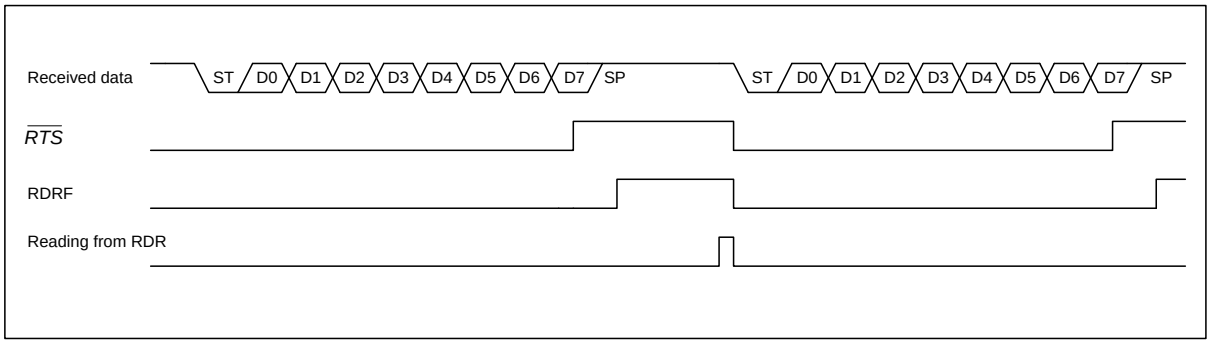
Figure 3-4 Hardware flow control during data transmission (SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)



- During data reception
- If FIFO is not used

  Upon reception of data one bit before the stop bit, "HIGH" is output to $\overline{RTS}$. After received data is read, "LOW" is output to $\overline{RTS}$.

Figure 3-5 Hardware flow control during data reception (with FIFO is unused.) (SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)

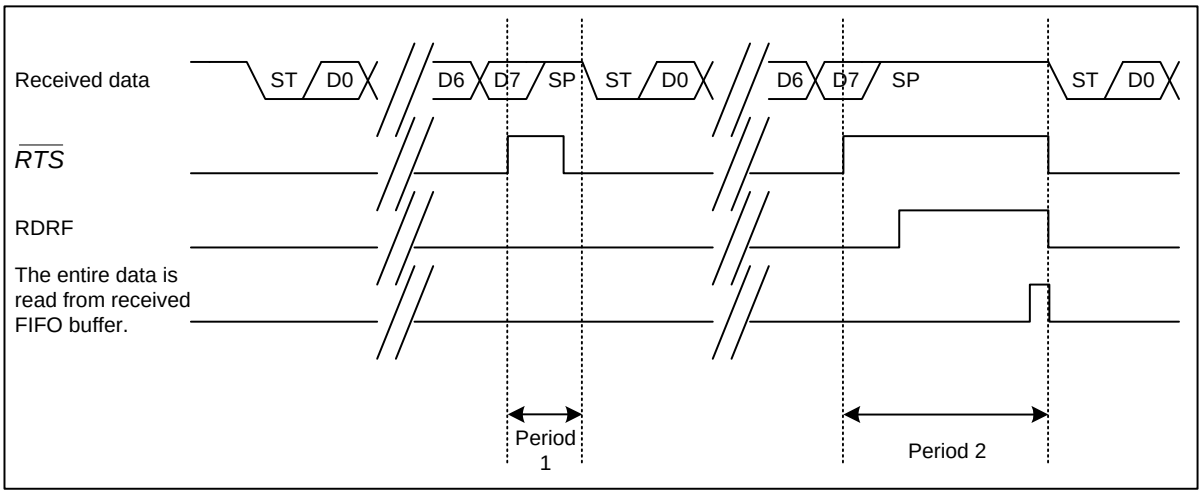


- If FIFO is used

  If SSR:RDRF is not set (the specified number of data sets are not received in received FIFO), $\overline{RTS}$ outputs "HIGH" upon reception of data one bit before the stop bit, but $\overline{RTS}$ outputs "LOW" upon detection of the stop bit. (For period 1)

  If SSR:RDRF is set (the specified number of data sets are received in received FIFO), $\overline{RTS}$ outputs "HIGH" upon reception of data one bit before the stop bit. $\overline{RTS}$ outputs "LOW" after all data is read from received FIFO. (For period 2)

Figure 3-6 Hardware flow control during data reception (with FIFO used) (SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)



Notes:

- *When reception operation is disabled (RXE=0), the $\overline{RTS}$ signal is fixed to "LOW".*
- *If the following two conditions are satisfied when received FIFO is used and if the received idle state continues for more than 8 baud rate clocks, RDRF is set to "1" but "LOW" is maintained for the $\overline{RTS}$ signal.*
  - *The received FIFO idle detection enable bit (FCR1:FRIIE) is "1".*
  - *The preset data amount is not received and some data remains in received FIFO.*
- *Performing programmable resetting (SCR:UPCL=1) clears the $\overline{RTS}$ signal to "LOW".*

4. Dedicated Baud Rate Generator

As for the UART transmit/received clock source, either of the following can be selected.

- Dedicated baud rate generator (reload counter)
- An external clock input to the baud rate generator (reload counter)

Selecting the UART baud rate

Select one of the following two baud rates.

- Baud rate obtained by dividing an internal clock using the dedicated baud rate generator (reload counter)

  This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).
  Each reload counter divides an internal clock by the set value.
  To set the clock source, select an internal clock (BGR1:EXT=0).

- Baud rate obtained by dividing an external clock using the dedicated baud rate generator (reload counter)

  Use an external clock for the clock source of the reload counter.
  To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).
  Each reload counter divides an external clock by the set value.
  To set the clock source, select use of an external clock and the baud rate generator clock (BGR1:EXT=1).
  This mode is designed for cases where an oscillator with a divided non-standard frequency is used.

Notes:

- *Set the external clock (BGR1:EXT=1) while the reload counter is suspended (BGR1/0=15' h00).*
- *If an external clock is selected (BGR1:EXT=1), its HIGH and LOW signals must have a width at least of two bus clocks.*

4.1 Baud rate settings

The following explains how to set the baud rate, and also a result of serial clock frequency calculation.

Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$V = \phi / b - 1$

V : Reload value b : Baud rate ϕ: Bus clock frequency or external clock frequency

(2) Calculation example

To set the 16 MHz bus clock, use the internal clock, and set the 19200 bps baud rate, set the reload value as follows:

Reload value:

$V = (16 \times 1000000) / 19200 - 1 = 832$

Therefore, the baud rate is:

$b = (16 \times 1000000) / (832 + 1) = 19208$ bps

(3) Baud rate error

The baud rate error can be calculated by the following equation.

Error (%) = (Calculated value – Target value) / Target value $\times 100$

Example: To set the 20 MHz bus clock and 153600 bps target baud rate:

Reload value = $(20 \times 1000000) / (129 + 1)$

Buad rate (Calculated value) = $(20 \times 1000000) / (129 + 1) = 153846$ (bps)

Error (%) = $(153846 - 153600) / 153600 \times 100 = 0.16$ (%):

Notes:

- *If the reload value is set to "0", the reload counter is stopped.*
- *If the reload value is an even number, in the received serial clock, the width of a "LOW" signal is longer than that of a "HIGH" signal by one bus clock cycle. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.*
- *Set the reload value to 4 or more. Note that data may not be received normally due to the baud rate error and reload value setting.*

Reload value and baud rate for each bus clock frequency

Table 4-1 Reload values and baud rates

Baud rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	-	-	-	7	0	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	138	-0.08
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.06	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	<0.01	554	-0.01	693	0.06	832	0.03	1110	0.01
19200	416	-0.08	520	-0.03	832	0.03	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	-0.03	1666	-0.02	2083	0.03	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

Value : BGR1/0 register set value (decimal)

ERR : Baud rate error (%)

Table 4-2 Reload values and baud rates (continued)

Baud rate (bps)	40 MHz		48 MHz		72 MHz		80 MHz		100 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	9	0	11	0	17	0	19	0	24	0
2.5M	15	0	-	-	-	-	31	0	39	0
2M	19	2	23	0	35	0	39	0	49	0
1M	39	0	47	0	71	0	79	0	99	0
500000	79	0	95	0	143	0	159	0	199	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22	216	<0.01
250000	159	0	191	0	287	0	319	0	399	0
230400	173	-0.22	207	0.16	312	-0.16	346	0.06	433	<0.01
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03	650	<0.01
125000	319	0	383	0	575	0	639	0	799	0
115200	346	0.06	416	-0.08	624	0	693	0.06	867	<0.01
76800	520	-0.03	624	0	937	-0.05	1041	-0.03	1301	<0.01
57600	693	0.06	832	0.04	1249	0	1388	<0.01	1735	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01	2603	<0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01	3471	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01	5207	<0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	<0.01	9599	<0.01
9600	4166	<0.08	4999	0	7499	0	8332	0	10416	0
7200	5555	<0.01	6666	<0.01	9999	0	11110	0	13888	0
4800	8332	<0.01	9999	0.02	14999	0	16666	0	20832	0
2400	16666	<0.01	19999	0	29999	0	-	-	-	-
1200	-	-	-	-	-	-	-	-	-	-
600	-	-	-	-	-	-	-	-	-	-
300	-	-	-	-	-	-	-	-	-	-

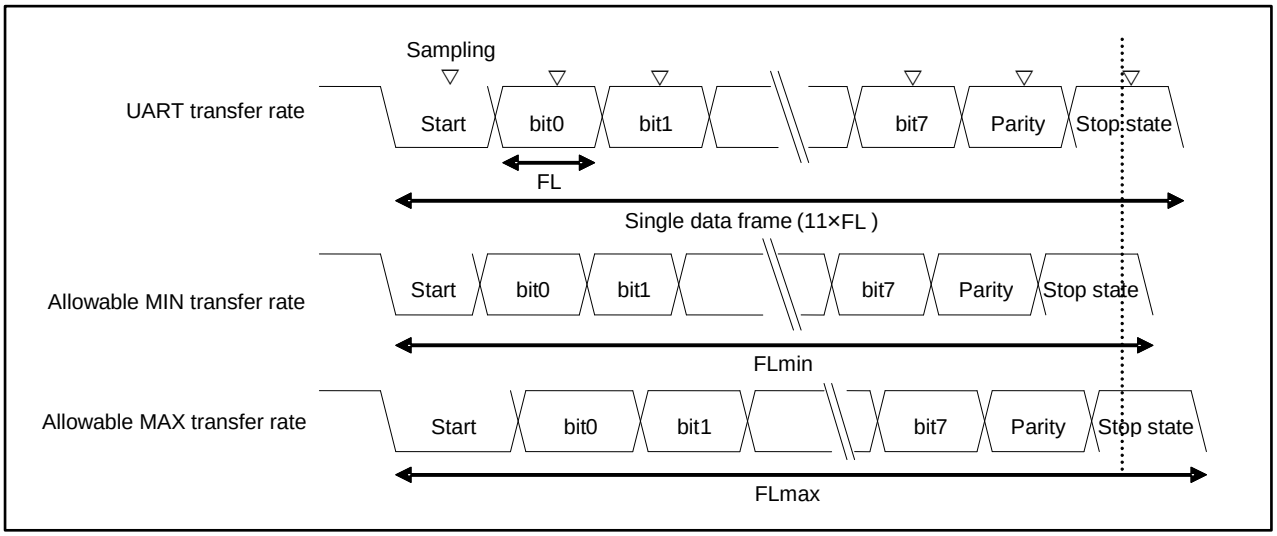
For frequencies not described in Table 4-1 and Table 4-2, calculate them conforming to “4.1 Baud rate settings”. (However, for the maximum frequencies which differ by products, see “Data Sheet” of the product used.

Allowable baud rate range for data reception

The following shows the range of baud rate error allowed for the destination to receive data.

Set the received baud rate error by using the following formulas to ensure that the value falls within the allowable range.

Figure 4-1 Allowable baud rate range for data reception



As shown in Figure 4-1, after detection of the start bit, the sampling timing of received data is determined by the counter set in the BGR1/0 register. Data can be received successfully if the bit sequence including the stop bit matches the sampling timing.

If this applies to a reception of 11 bits, a theoretical explanation can be given in the following.

Assuming that the sampling timing margin is one bus clock (ϕ), the minimum allowable transfer rate (FLmin) is determined as follows:

FLmin = (11bits × (V+1) - (V+1)/2 + 2)/ϕ = (21V + 25)/2 ϕ (s) V: Reload value, ϕ: Bus clock

Thus, the maximum baud rate that allows the destination to receive data (BGmax) is determined as follows.

BGmax = 11/FLmin = 22ϕ/(21V+25) (bps) V: Reload value, ϕ: Bus clock

When data is received at the maximum allowable transfer rate (FLmax), the starting point of the received 11th bit is sampled.

Thus, the maximum allowable transfer rate (FLmax) is determined as follows:

10/11 × FLmax = (11bits × (V+1) – (V+1)/2)/ϕ V: Reload value, ϕ: Bus clock

FLmax = (21/20 × 11 × (V+1)/ϕ

Assuming that the sampling timing margin (ϕ) is two clocks, the maximum allowable transfer rate (FLmax) is determined as follows:

10/11× FLmax = (11bits × (V+1) – (V+1)/2 – 2)/ϕ V: Reload value, ϕ: Bus clock

FLmax = (21/20 × 11 × (V+1) – 44/20)/ϕ = (231V + 187)/20 ϕ (s) V: Reload value, ϕ: Bus clock

Accordingly, the minimum baud rate that allows the destination to receive data (BGmin) is determined as follows:

BGmin=11/FLmax=220ϕ/(231V+187) (bps) V: Reload value, ϕ: Bus clock

From the above formulas for obtaining the minimum/maximum baud rate, the allowable error between UART and the destination is obtained as follows.

Reload value (V)	Maximum allowable baud rate error	Minimum allowable baud rate error
3	0%	0
10	+2.98%	-3.08%
50	+4.37%	-4.40%
100	+4.56%	-4.58%
200	+4.66%	-4.67%
32767	+4.76%	-4.76%

Note:

- *Reception accuracy depends on the number of bits per frame, bus clock, and reload value. The higher the bus clock and frequency division ratio are, the higher the accuracy becomes.*

External clock

Writing "1" to the EXT bit of the Baud Rate Generator Register (BGR) causes the baud rate generator to divide the external clock's frequency.

Note:

- *The external clock signal synchronizes with the internal clock on UART. Therefore, an external clock that does not allow synchronization causes unstable operation.*

Functions of reload counter

There are two types of reload counters: The transmission reload counter and the received reload counter, both functioning as a dedicated baud rate generator. Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from the external or internal clock.

Starting counting

When the reload value is written to the Baud Rate Generator Register1, 0 (BGR1 or BGR0), the reload counter starts counting.

Restarting

The reload counter restarts counting in the following conditions.

- Common to transmit and received reload counters
  A programmable reset (SCR:UPCL bit)
- Received reload counter
  Detection of the start bit's falling edge in asynchronous mode

5. Setting Procedure and Program Flow in Operation Mode 0 (Asynchronous Normal Mode)

Operation mode 0 enables asynchronous bi-directional serial communications.

CPU-to-CPU connection

Select the bi-directional communication in operation mode 0 (normal mode). Connect two CPUs to each other as shown in Figure 5-1 and Figure 5-2.

Figure 5-1 A connection example of bi-directional communications in UART operation mode 0 (with flow control disabled)

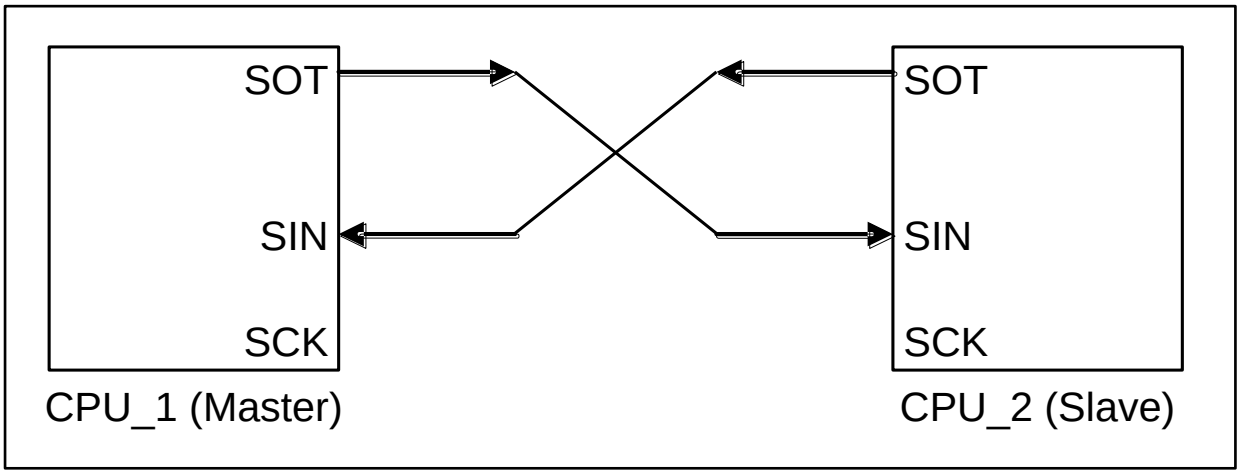
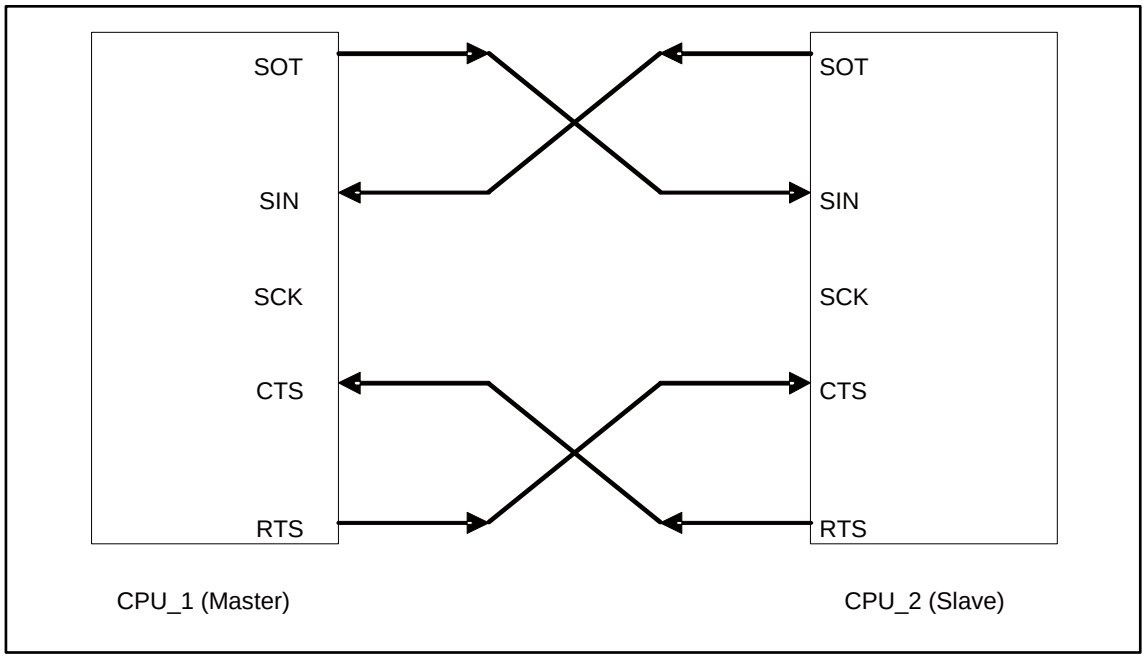


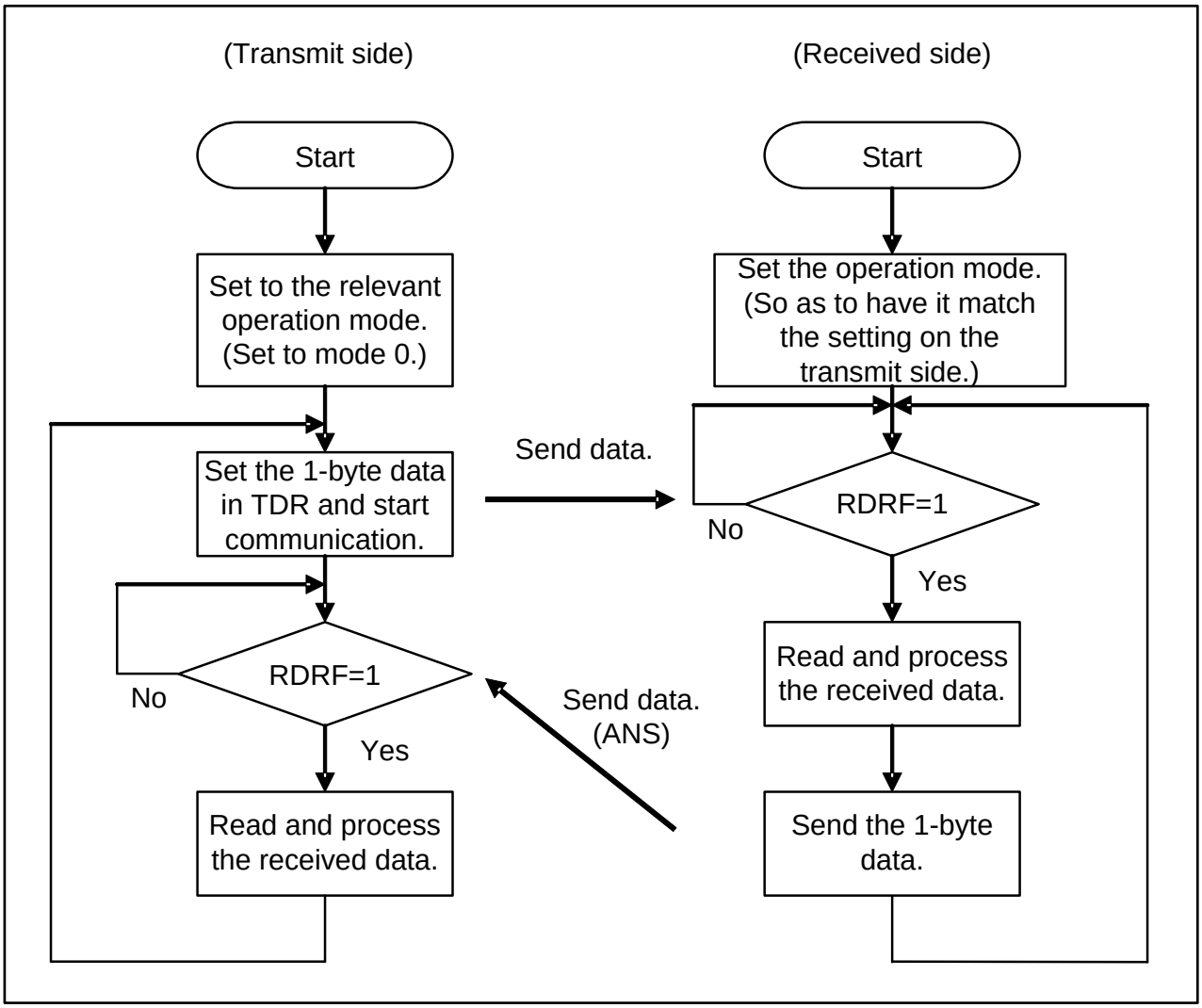
Figure 5-2 A connection example of bi-directional communications in UART operation mode 0 (with flow control)



Flowcharts

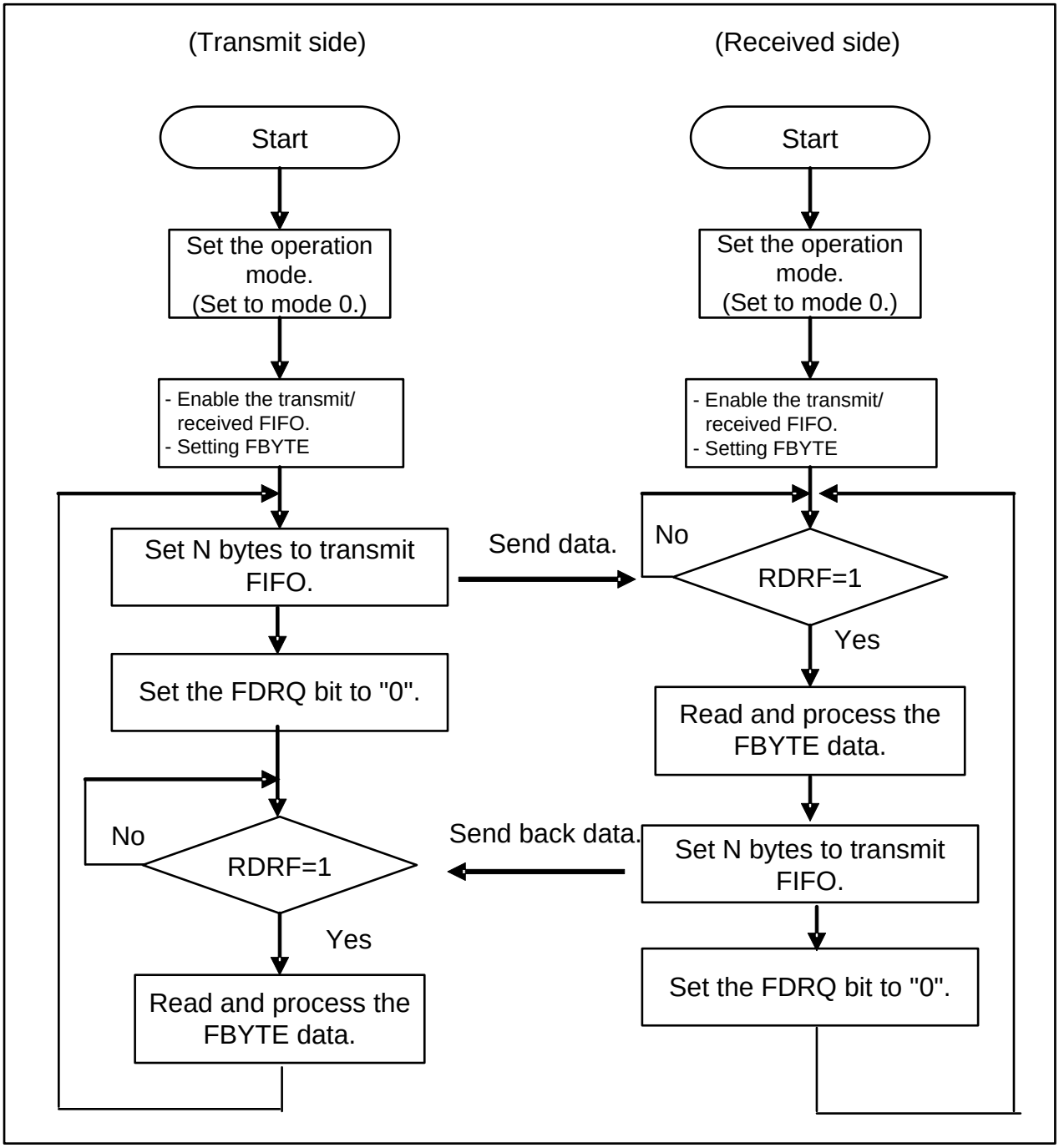
■ If FIFO is not used

Figure 5-3 An example of bidirectional communication flowchart (if FIFO is not used)



■ If FIFO is used

Figure 5-4 An example of bidirectional communication flowchart (if FIFO is used)



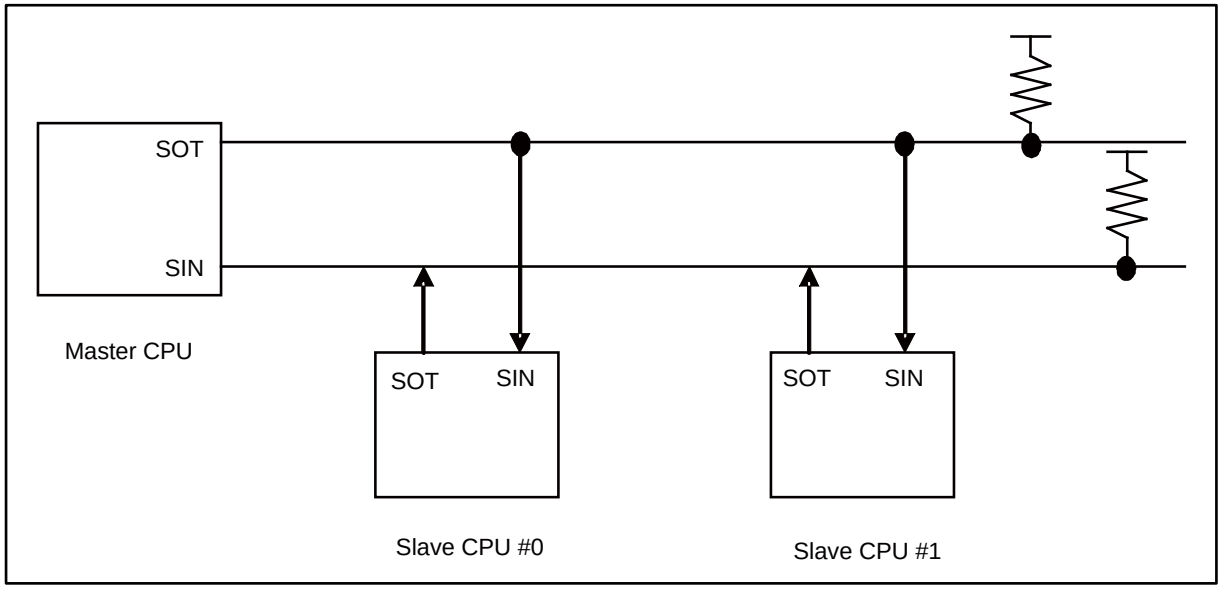
6. Setting Procedure and Program Flow in Operation Mode 1 (Asynchronous Multiprocessor Mode)

In operation mode 1 (multiprocessor mode), communications by master/slave connections with multiple CPUs are enabled. Either the master or slave function is available.

CPU-to-CPU connection

In a master/slave type communications, as shown in Figure 6-1, the communications system is configured with two common communication lines connected to the master CPU and multiple slave CPUs. UART can be used either as a master or a slave.

Figure 6-1 A connection example for master/slave type communications on UART



Function selection

In master/slave type communications, select the operation mode and data transfer system, as shown in Table 6-1.

Table 6-1 Selection of master/slave type communications functions

	Operation mode		Data	Parity	Stop state bit	bit direction
	Master mode CPU	Slave mode CPU				
Address transmit and reception	Mode 1 (A/D bit transmit)	Mode 1 (A/D bit reception)	AD=1 + 7 or 8 bits Address	OFF	One bit or 2 bits	LSB or MSB first
Data transmit and reception			AD=0 + 7 or 8 bits Data			

Note:

- *In operation mode 1, operate in word access mode for transmit/received data (TDR/RDR).*

■ Communications procedure

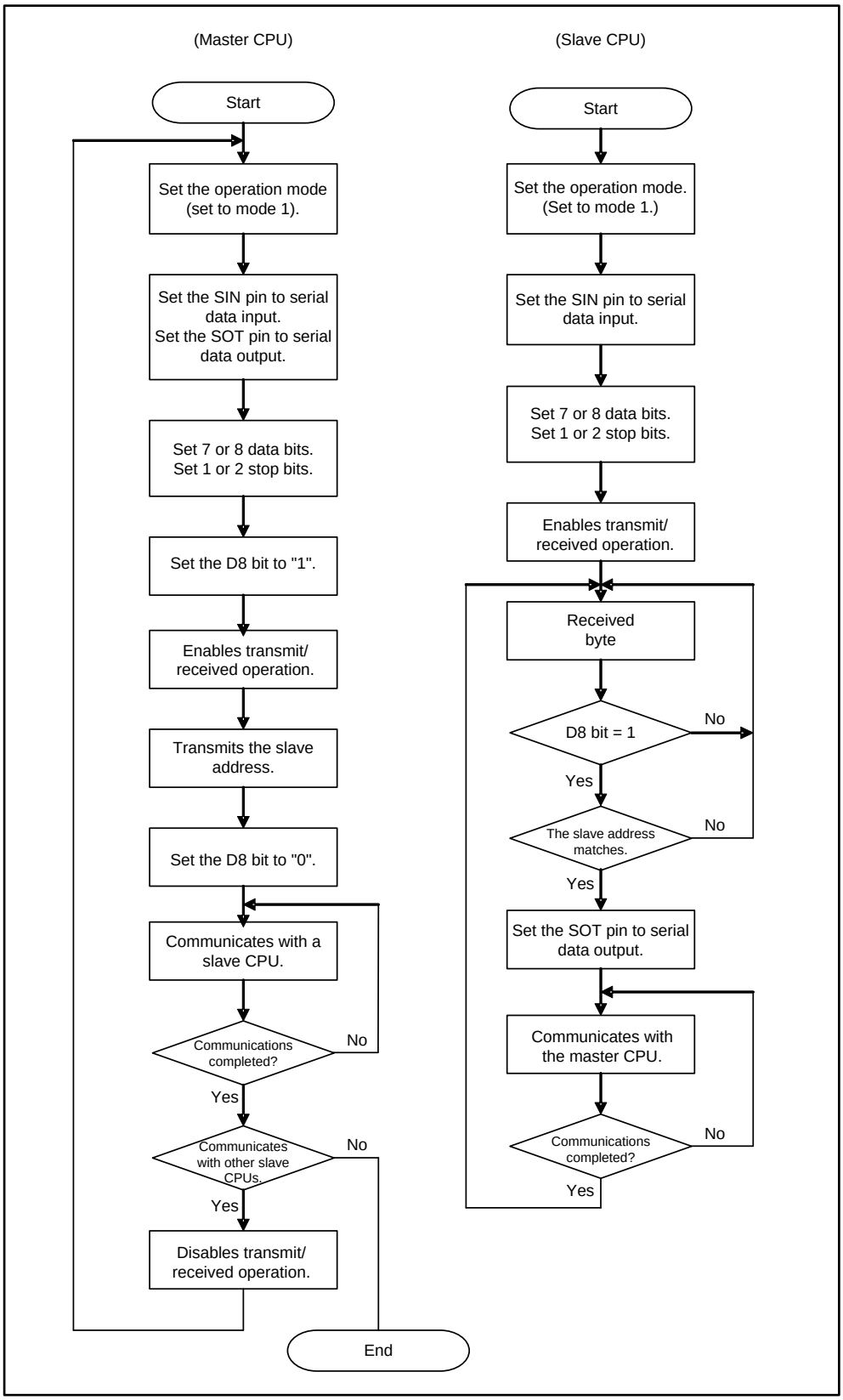
Communications start when the master CPU transmits address data. Address data is a data set whose D8 bit is "1", and used for selecting a slave CPU to communicate with. Each slave CPU judges the address as programmed, and communicates with the master CPU if that address matches the assigned address.

Figure 6-2 and Figure 6-3 show flowcharts of master/slave type communications (in multiprocessor mode).

Flowcharts

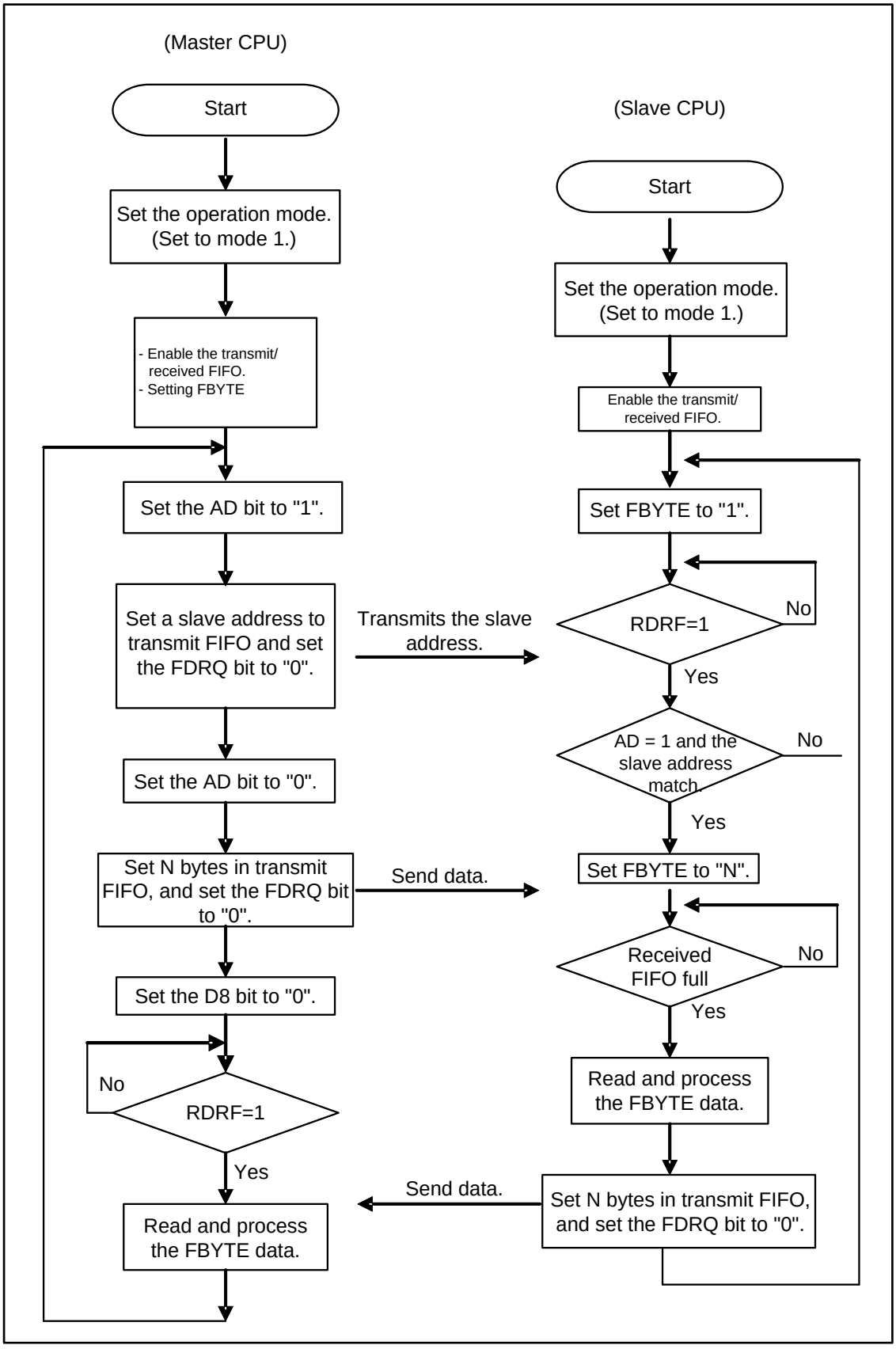
■ If FIFO is not used

Figure 6-2 An example flowchart for master/slave type communications (if FIFO buffer is not used)



■ If FIFO is used

Figure 6-3 An example flowchart for master/slave type communications (if FIFO buffer is used)



7. UART (Asynchronous Serial Interface) Registers

This section provides a list of UART (Asynchronous Serial Interface) registers.

UART (Asynchronous Serial Interface) registers list

Table 7-1 UART (Asynchronous Serial Interface) register list

	bit15 bit8	bit7 bit0
UART	SCR (Serial Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	ESCR (Extended Communication Control Register)
	RDR1/TDR1 (Transmit/Received Data Register 1)	RDR0/TDR0 (Transmit/Received Data Register 0)
	BGR1 (Baud Rate Generator Register 1)	BGR0 (Baud Rate Generator Register 0)
FIFO	FCR1 (FIFO Control Register 1)	FCR0 (FIFO Control Register 0)
	FBYTE2 (FIFO2 Byte Register)	FBYTE1 (FIFO1 Byte Register)

Table 7-2 UART (Asynchronous Serial Interface) bit assignment

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SBL	BDS	-	SOE
SSR/ ESCR	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	FLWEN	ESBL	INV	PEN	P	L2	L1	L0
TDR/ (RDR)	-							D8 (AD)	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

Operation mode

UART (Asynchronous Serial Interface) operates in two different modes. The Serial Mode Register (SMR) determines the mode to be enabled, depending on its setting, MD2, MD1 or MD0.

Table 7-3 UART (Asynchronous Serial Interface) operation modes

Operation mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (asynchronous normal mode)
1	0	0	1	UART1 (asynchronous multiprocessor mode)

7.1 Serial Control Register (SCR)

The Serial Control Register (SCR) can perform transmit/received enable/disable, transmit/received interrupt enable/disable, transmit bus idle interrupt enable/disable and UART reset operations.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	(SMR)
Attribute	R/W	-	-	R/W	R/W	R/W	R/W	R/W	
Initial value	0	-	-	0	0	0	0	0	

[bit15] UPCL: Programmable Clear bit

Initializes the UART internal state.

bit	Description	
	At writing	At reading
0	No effect on opreration.	"0" is always read.
1	Programmable clear	

If set to "1",

- UART is reset directly (software reset). However, the current register settings are maintained. The transmit or received state is disconnected immediately.
- The baud rate generator reloads the BGR1/0 register value and restarts operation.
- All of transmit/received interrupt factors (SSR:PE, FRE, ORE, RDRF, TDRE and TBI) are initialized (to 0b000011).
- $\overline{\text{RTS}}$ signal is cleared to "LOW".

If set to "0",

It has no effect on operation.

"0" is always read during reading.

Notes:

- *Disable an interrupt first, and then execute the programmable clear instruction.*
- *If the FIFO operation is used, disable it (FCR0:FE[2:1]=00) first and then execute Programmable Clear.*

[bit14:13] - : Unused bits

- *These bits' values are undefined when read.*
  These bits have no effect when written.

[bit12] RIE: Received interrupt enable bit

- This bit enables or disables an output of received interrupt request to the CPU.
- If the RIE bit and the received data flag bit (SSR:RDRF) are "1", or if any of the error flag bits (SSR:PE, ORE or FRE) is "1", a received interrupt request is output.

bit	Description
0	Disables the received interrupt.
1	Enables the received interrupt.

[bit11] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of Transmit Interrupt Request to the CPU.
- If the TIE bit and SSR:TDRE bit are "1", a Transmit Interrupt Request is output.

bit	Description
0	Disables a transmit interrupt.
1	Enables a transmit interrupt.

[bit10] TBIE: Transmit bus idle interrupt enable bit

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If the TBIE bit and TBI bit are "1", a transmit bus idle interrupt request is output.

bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit9] RXE: Received operation enable bit

Enables or disables UART received operation.

bit	Description
0	Disables data received.
1	Enables data received.

Notes:

- *Reception is not started unless the falling edge of the start bit (in NRZ format, when ESCR:INV=0) is input even if reception is enabled (RXE=1). (In the inverted NRZ format (ESCR:INV=1), reception is not started unless the rising edge is input).*
- *If data reception is disabled (RXE=0) during the received operation, the current data reception is stopped immediately.*
- *When the received operation is disabled (RXE=0), the* $\overline{\text{RTS}}$ *signal is fixed to "LOW".*

[bit8] TXE: Transmission operation enable bit

Enables or disables the UART transmission operation.

bit	Description
0	Disables the transmission.
1	Enables the transmission.

Note:

- *If data transmission is disabled (TXE=0) during the transmission operation, the current data transmission is stopped immediately.*

7.2 Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to set the operation mode, transfer direction, data length and to select the stop bit length as well as to enable/disable output of serial data to their pins.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCR)	MD2	MD1	MD0	-	SBL	BDS	Reserved	SOE
Attribute		R/W	R/W	R/W	-	R/W	R/W	-	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7:5] MD2, MD1, MD0: Operation mode set bit

Set operation mode of the Asynchronous Serial Interface..

* This chapter explains the registers and their operation in operation mode 0 (asynchronous normal mode) and in operation mode 1 (asynchronous multiprocessor mode).

bit7	bit6	bit5	Description
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multiprocessor mode)
0	1	0	Operation mode 2 (clock sync mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I^2C mode)
Other than the above			Setting is prohibited.

Notes:

- *Any bit setting other than above is prohibited.*
- *To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) and switch the operation mode continuously.*
- *After the operation mode has been switched, set each register correctly.*

[bit4] Reserved: Reserved bit

The read value is "0". Be sure to write "0".

[bit3] SBL: Stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

bit	Description	
0	ESCR:ESBL=0	1 bit
	ESCR:ESBL=1	3 bits
1	ESCR:ESBL=0	2 bits
	ESCR:ESBL=1	4 bits

Notes:

- *In the reception operation, only the first bit of the stop bit data is detected.*
- *Always set this bit when transmission is disabled (SCR:TXE=0).*

[bit2] BDS: Transfer direction select bit

Specifies to transmit the least significant bit of the transmit serial data first (LSB first; BDS=0) or the most significant bit first (MSB first; BDS=1).

bit	Description
0	LSB first (The least significant bit is first transferred.)
1	MSB first (The most significant bit is first transferred.)

Note:

- *Set this bit when transmission and reception are disabled (SCR:TXE=SCR:RXE=0).*

[bit1] Reserved bit

The read value is "0". Be sure to write "0".

[bit0] SOE: Serial data output enable bit

This bit enables or disables a serial data output.

bit	Description
0	Disables a serial data output.
1	Enables a serial data output.

Note:

- *If this bit is used as the SOT pin, the GPIO must also be set.*

7.3 Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the current transmit/received state, check the received error flag, and clears the received error flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	(ESCR)
Attribute	R/W	-	R	R	R	R	R	R	
Initial value	0	-	0	0	0	0	1	1	

[bit15] REC: Received error flag clear bit

This bit clears the PE, FRE and ORE flags of the Serial Status Register (SSR).

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	Clears the received error flag (PE, FRE, ORE).	

[bit14] - : Unused bit

This bit value is undefined when read.
This bit has no effect when written.

[bit13] PE: Parity error flag bit (only functions in operation mode 0)

- If a parity error occurs during data received with ESCR:PEN=1, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the PE bit and SCR:RIE bit are "1", a received interrupt request is output.
- If this flag is set, data in the Received Data Register (RDR) is invalid.
- If this flag is set when received FIFO is used, the received FIFO enable bit is cleared and the received data is not stored in received FIFO.

bit	Description
0	No parity error occurred.
1	A parity error occurred.

[bit12] FRE: Framing error flag bit

- If a framing error occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the FRE bit and SCR:RIE bit are "1", a received interrupt request is output.
- If this flag is set, data in the Received Data Register (RDR) is invalid.
- If this flag is set when received FIFO is used, the received FIFO enable bit is cleared and the received data is not stored in received FIFO.

bit	Description
0	No framing error occurred.
1	A framing error occurred.

[bit11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the ORE and SCR:RIE bits are "1", a received interrupt request is output.
- If this flag is set, data in the Received Data Register (RDR) is invalid.
- If this flag is set when received FIFO is used, the received FIFO enable bit is cleared and the received data is not stored in received FIFO.

bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit10] RDRF: Received data full flag bit

- This flag shows the state of the Received Data Register (RDR).
- When the received data is loaded in the RDR, this bit is set to "1". When data is read from the Received Data Register (RDR), this bit is cleared to "0".
- If the RDRF bit and SCR:RIE bit are "1", a received interrupt request is output.
- If the received FIFO is used and if a certain count of data is received by the received FIFO, the RDRF bit is set to "1".
- If received FIFO is used, if both of the following conditions are satisfied, and if the Received Idle state continues more than 8 baud rate clocks, the RDRF bit is set to "1".
  - The received FIFO idle detection enable bit (FCR1:FRIIE) is "1".
  - The preset data amount is not received and some data remains in received FIFO.

  If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted.
- If the received FIFO is used and if this buffer is emptied, this bit is cleared to "0".

bit	Description
0	The Received Data Register (RDR) is empty.
1	The Received Data Register (RDR) contains data.

[bit9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When data is loaded to the transmit shift register and when the transmission is started, this bit is set to "1" to indicate that the TDR does not have the valid data.
- If the TDRE bit and SCR:TIE bit are "1", a transmit interrupt request is output.
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- For the TDRE bit set/reset timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register is empty.

[bit8] TBI: Transmit bus idle flag

- This bit indicates that UART is not transmitting data.
- When transmit data is written in the Transmit Data Register (TDR), this bit is set to "0".
- If the Transmit Data Register is empty (TDRE=1) and not transmitting data, this bit is set to "1".
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", this bit is set to "1".
- If this bit is "1" and if the transmit bus idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

bit	Description
0	During data transmission
1	No data transmission

7.4 Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to set a transmit/received data length, enable/disable a parity bit, select a parity bit, invert the serial data format and set stop bit length selection.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSR)	FLWEN	ESBL	INV	PEN	P	L2	L1	L0
Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7] FLWEN: Flow control enable bit

Selects to enable or disable the hardware flow control operation.

bit	Description
0	Disables hardware flow control.
1	Enables hardware flow control.

Notes:

- *Set this bit when data transmission and reception is disabled (SCR:TXE=0, RXE=0).*
- *Set this bit to "1" only when the hardware flow control is desired.*

[bit6] ESBL: Extension stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

bit	Description	
0	SMR:SBL=0	1 bit
	SMR:SBL=1	2 bits
1	SMR:SBL=0	3 bits
	SMR:SBL=1	4 bits

Notes:

- *In the reception operation, only the first bit of the stop bit data is detected.*
- *Always set this bit when transmission is disabled (SCR:TXE=0).*

[bit5] INV: Inverted serial data format bit

Selects NRZ or inverted NRZ for the serial data format.

bit	Description
0	NRZ format
1	Inverted NRZ format

[bit4] PEN: Parity enable bit (only functions in operation mode 0)

Sets to add (for transmit) and detect (for reception) a parity bit or not to.

bit	Description
0	Disables parity.
1	Enables parity.

Note:

- *In operation mode 1, this bit is internally fixed at "0".*

[bit3] P: Parity select bit (only functions in operation mode 0)

When set to enable parity (ESCR:PEN=1, this bit is set to either odd-number parity "1" or even-number parity "0".

bit	Description
0	Even-number parity
1	Odd-number parity

[bit2:0] L2, L1, L0: Data length select bit

These bits set a length of transmit/received data.

bit2	bit1	bit0	Description
0	0	0	8-bit length
0	0	1	5-bit length
0	1	0	6-bit length
0	1	1	7-bit length
1	0	0	9-bit length

Notes:

- *Any setting other than the above is prohibited.*
- *In operation mode 1, set the data length to seven or eight bits. Any other setting is prohibited.*

7.5 Received Data Register/Transmit Data Register (RDR/TDR)

The Received and Transmit Data Registers are allocated at the same address. This register functions as the Received Data Register when data is read from it. This register operates as the Transmit Data Register when data is written in it.

When the FIFO operation is enabled, the RDR/TDR address functions as the FIFO read/write address.

Received Data Register (RDR)

bit	15 ... 9	8	7	6	5	4	3	2	1	0
Field		D8	D7	D6	D5	D4	D3	D2	D1	D0
Attribute		R	R	R	R	R	R	R	R	R
Initial value		0	0	0	0	0	0	0	0	0

The Received Data Register (RDR) is a 9-bit data buffer register for serial data reception.

- When serial data signals are sent to the Serial Input pin (SIN), they are converted by a shift register and stored in the Received Data Register (RDR).
- The upper bits are set to "0" according to the data length, as follows.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

(X represents the received data bit.)

- When the received data is stored in the Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is set to "1". If a received interrupt is enabled (SSR:RIE=1), a received interrupt request is generated.
- The Received Data Register (RDR) must be read only when the received data full flag bit (SSR:RDRF) is "1". When data is read from the Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is cleared to "0" automatically.
- If a received error occurs (when SSR:PE, ORE or FRE is "1"), data in the Received Data Register (RDR) becomes invalid.
- In operation mode 1 (multiprocessor mode), 7-bit or 8-bit long operation takes place and the received AD bit is stored in the D8 bit.
- For 9-bit long data transfer and in operation mode 1, data must be read from RDR by 16-bit data accessing.

Notes:

- *If the Received FIFO is used and if the preset amount of data is received in the Received FIFO buffer, SSR:RDRF is set to "1".*
- *If the received FIFO is used and if this buffer is emptied, the SSR:RDRF bit is cleared to "0".*
- *If a received error occurs when received FIFO is used (SSR:PE, ORE, or FRE is "1"), the received FIFO enable bit is cleared and the received data is not stored in the received FIFO buffer.*

Transmit Data Register (TDR)

bit	15 ... 9	8	7	6	5	4	3	2	1	0
Field		D8	D7	D6	D5	D4	D3	D2	D1	D0
Attribute		W	W	W	W	W	W	W	W	W
Initial value		1	1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a 9-bit data buffer register for serial data transmission.

- If data transmission is enabled (SCR:TXE=1) and if the transmit data is written in the Transmit Data Register (TDR), the transmit data is transferred to the Transmit Shift Register. The transmit data is then converted into serial data and sent out from the serial data output pin (SOT).
- The upper bits are sequentially made invalid according to the data length as follows.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

(X means a transmit data bit.)

- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When the transmit data is transferred to the transmit shift register and data transmission is started, and if transmit FIFO is disabled or if transmit FIFO is empty, the transmit data empty flag (SSR:TDRE) is set to "1".
- If the transmit data empty flag (SSR:TDRE) is "1", transmit data can be written. If a transmit interrupt is enabled, a transmit interrupt occurs. Perform transmit data write after a transmit interrupt is generated or when the transmit data empty flag (SSR:TDRE) is "1".
- If the transmit data empty flag (SSR:TDRE) is "0" and transmit FIFO is disabled or the transmit FIFO buffer is full, no transmit data can be written.
- In operation mode 1 (multiprocessor mode), 7-bit or 8-bit long operation takes place and the AD bit is sent by writing to the D8 bit.
- For 9-bit long data transfer and in operation mode 1, data must be written in TDR by 16-bit data accessing.

Notes:

- *The Transmit Data Register is a write-only register. While the Received Data Register is a read-only register. As the transmission and received registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) instructions cannot be used.*
- *For the transmit data empty flag (SSR:TDRE) set timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".*

7.6 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks. Also, an external clock can be selected as the clock source of the reload counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXT	(BGR1)							(BGR0)							
Attribute	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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- The Baud Rate Generator Registers are used to set a frequency division ratio of serial clocks.
- The BGR1 register corresponds to the upper bits, and the BGR0 register corresponds to the lower bits. The reload value to be counted can be written, and the BGR1/BGR0 set value can be read.
- When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the reload counter starts its counting.
- The EXT bit (bit15) specifies to use the clock source of reload counter as the internal clock or the external clock. If EXT=0 is set, an internal clock is used. If EXT=1 is set, an external clock is used.

[bit15] EXT: External clock select bit

bit	Description
0	Uses the internal clock.
1	Uses an external clock.

[bit14:8] BGR1: Baud Rate Generator Register 1

bit14:8	Description
Write	Writes data in bit8 to bit14 of reload counter.
Read	Read the BGR1 set value.

[bit7:0] BGR0: Baud Rate Generator Register 0

bit7:0	Description
Write	Write data in bit0 to bit7 of reload counter.
Read	Read the BGR0 set value.

Notes:

- *Data must be written in the Baud Rate Generator Registers (BGR1 and BGR0) by 16-bit data accessing.*
- *If the current values of Baud Rate Generator Registers (BGR1, BGR0) are changed, the new values are reloaded only after the counter value has reached "15h00". In order to validate the new set values immediately, change the BGR1/BGR0 set values and execute the programmable clear (UPCL).*
- *If the reload value is an even number, in the received serial clock, the width of a "LOW" signal is longer than that of a "HIGH" signal by one bus clock cycle. If the value is an odd number, the width of a LOW signal is the same as that of a HIGH signal.*
- *Set a value "4" or higher to BGR1/BGR0. Note that data may not be received successfully depending on the baud rate error and reload value settings.*
- *To change the setting to an external clock (EXT=1) while the Baud Rate Generator is running, write "0" to the Baud Rate Generators 1 and 0 (BGR1, BGR0), execute Programmable Clear (UPCL) and then set for an external clock (EXT=1).*

7.7 FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select the transmit or received FIFO, enable the transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
Attribute	-			R/W	R/W	R/W	R/W	R/W	
Initial value	-			0	0	1	0	0	

[bit15:13] Reserved bits

The read value is "0". Be sure to write "0".

[bit12] FLSTE: Re-transmission data lost detect enable bit

This bit enables the FIFO re-transmission data lost flag (FLST) detection.

bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

Note:

- *To set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".*

[bit11] FRIIE: Received FIFO idle detection enable bit

This bit sets to detect the received idle state if the received FIFO contains valid data and if it continues more than 8-bit hours. If the received interrupt is enabled (SCR:RIE=1), a received interrupt is generated when the received idle state is detected.

bit	Description
0	Disables the received FIFO idle detection.
1	Enables the received FIFO idle detection.

Note:

- *In case of using Received FIFO, set this bit to "1".*

[bit10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. At this time, if a transmit FIFO interrupt is enabled (FTIE=1), a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).

The FDRQ bit is reset when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

Notes:

- *"0" written when transmit FIFO is enabled is valid.*
- *If the FBYTE (for transmission) is "0", this bit cannot be set to "0".*
- *If this bit is set to "1", it has no effect on the operation.*
- *If a read-modify-write instruction is issued, "1" is read.*

[bit9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit8] FSEL: FIFO select bit

This bit selects the transmit or received FIFO.

bit	Description
0	Transmit FIFO:FIFO1; Received FIFO:FIFO2
1	Transmit FIFO:FIFO2; Received FIFO:FIFO1

Notes:

- *This bit is not cleared by the FIFO Reset (FCR0:FCL[2:1]=11).*
- *To change this bit state, first disable the FIFO operation (FCR0:FE[2:1]=00).*

7.8 FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute		-	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7] - : Unused bit

When read, always "0" is read.
When written, always set this bit to "0".

[bit6] FLST: FIFO re- transmit data lost flag bit

This bit shows that the re- transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- Data is written (overwritten) in the FIFO buffer when the FLSTE bit of FIFO Control Register 1 (FCR1) is "1" and the write pointer for transmit FIFO matches the read pointer which has been saved by the FSET bit.

The FLST bit is reset when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data identified by the read pointer (saved by the FSET bit) is overwritten. Therefore, the FLD bit cannot set the data re-transmission even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in the FIFO buffer again.

bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.
When the re-transmission setting has finished, this bit is set to "0".

bit	Description
0	Not reloaded
1	Reloaded

Notes:

- *If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.*
- *When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".*
- *After you have set the TIE bit and TBIE bit to "0", set this bit to "1". After you have enabled transmit FIFO, set the SCR:TIE bit and SCR:TBIE bit to "1".*

[bit4] FSET: FIFO pointer save bit

This bit saves the transmit FIFO read pointer.

If the read pointer value is saved before being transmitted and if the FLST bit is "0", the data can be re-transmitted even if a communication error or others have occurred.

If set to "1", the current read pointer value is saved.
If set to "0", the read pointer is not saved..

bit	Description	
	At writing	At reading
0	Not saved	"0" is always read.
1	The read pointer value is saved.	

Note:

- *This bit can be set to "1" only when the transmission byte count (FBYTE) is "0".*

[bit3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

If this bit is set to "1", the FIFO2 internal state is initialized.

Only the FCR1:FLST bit is initialized, and the other bits of FCR1/FCR0 registers are kept.

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	FIFO2 is reset.	

Notes:

- *Disable the transmit and reception first, and then reset FIFO2.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The valid data count of the FBYTE2 register is set to "0".*

[bit2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 state.

If this bit is set to "1", the FIFO1 internal state is initialized.

Only the FCR1:FLST bit is initialized, and the other bits of FCR1/FCR0 registers are kept.

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	FIFO1 is reset.	

Notes:

- *Disable the transmit and reception first, and then reset FIFO1.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The valid data count of the FBYTE1 register is set to "0".*

[bit1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If FIFO2 is set as transmit FIFO (FCR1:FSEL=1) and if data exists in FIFO2 when this bit is set to "1", the data transmission starts immediately when the UART is enabled to transmit data (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both SCR:TIE bit and SCR:TBIE bit to "1".
- If received FIFO is selected by the FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- If FIFO2 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO2 is used as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and no valid data exists in received FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO2 is used as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO2 state is held even if the FIFO2 operation is disabled.

bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

[bit0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- When the FIFO1 is set as transmit FIFO (FCR1:FSEL=0) and if data exists in FIFO1 when this bit is set to "1", the data transmission starts immediately when the UART is set to enable data transmission (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both TIE bit and SCR:TBIE bit to "1".
- If received FIFO is selected by the FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- If FIFO1 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO1 is used as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and no valid data exists in received FIFO (FBYTE2=0) after reception is disabled (SCR:RXE=0).
- If FIFO1 is used as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO1 state is held even if the FIFO1 operation is disabled.

bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

7.9 FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer. Also, this register can be used to generate a received interrupt when certain number of data sets are received in the received FIFO.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count of data written from or received in FIFO. The following shows the settings of the FCR1:FSEL bit.

Table 7-4 Display of data count

FSEL	FIFO selection	Data count display
0	FIFO2: Received FIFO, FIFO1:Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmit FIFO, FIFO1:Received FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to flag a received interrupt for the FBYTE register of received FIFO. If this specified transfer count matches the FBYTE register display, the receive data full flag bit (RDRF) is set to "1".
- If the following two conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the receive data full flag bit (RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.

If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to zero (0). If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.

FBYTE1, FBYTE2: FIFO2 data count display bits, FIFO1 data count display bits

At writing	Sets the transfer data count.
At reading	Reads the effective count of data.

Read (Effective data count)

During transmit : The number of data sets already written in the FIFO buffer but not transmitted yet
During reception : The number of data sets reception in FIFO

Write (Transfer data count)

During transmit: Set "0x00".
During reception: Set the data count to generate a received interrupt.

Table 7-5 Data Count to be Saved in FIFO

FIFO Capacity	Operation Mode	Data Length	Max. FBYTE Count	Count of Data to be Stored
16 BYTES	Mode 0	5 bits to 8 bits	16	16
	Mode 0	9 bits	8	8
	Mode 1	Enitre bits		
32 BYTES	Mode 0	5 bits to 8 bits	32	32
	Mode 0	9 bits	16	16
	Mode 1	Entire bits		
64 BYTES	Mode 0	5 bits to 8 bits	64	64
	Mode 0	9 bits	32	32
	Mode 1	Entire bits		
128 BYTES	Mode 0	5 bits to 8 bits	128	128
	Mode 0	9 bits	64	64
	Mode 1	Entire bits		

Notes:

- *Set "0x00" in the FBYTE register of transmit FIFO.*
- *Set a data value equal to or greater than "1" in the FBYTE register of received FIFO.*
- *This state can be changed only after the data reception has been disabled.*
- *A read-modify-write instruction cannot be used for this register.*
- *Any setting exceeding the FIFO capacity is prohibited.*

CHAPTER 1-3: CSIO (Clock Synchronous Serial Interface)

This chapter explains the Clock Synchronous Serial Interface (CSIO) function that is supported in Operation mode 2. This CSIO is a part of the multifunction serial interface functions.

---

1. Overview of CSIO (Clock Synchronous Serial Interface)
2. CSIO (Clock Synchronous Serial Interface) interrupts
3. CSIO (Clock Synchronous Serial Interface) operations
4. Serial Timer Operation
5. Serial Chip Select Operation
6. Dedicated baud rate generator
7. CSIO (Clock Synchronous Serial Interface) registers
8. Restrictions on CSIO (Clock Synchronous Serial Interface)

CODE: 9BFCSIO_FM4-E01.0_FM15C-J05.4

1. Overview of CSIO (Clock Synchronous Serial Interface)

The CSIO is a general-purpose serial data communication interface (supporting the SPI) to allow synchronous communication with an external device. It also has transmit/received FIFO (up to 128 bytes each) [*1] installed.

CSIO (Clock Synchronous Serial Interface) functions

		Function
1	Data buffer	− Full duplex double buffer (when FIFO is not used) − Transmit/Received FIFO (up to 128 bytes each) [*1] (if FIFO is used)
2	Transfer system	− Clock synchronization (without start/stop bit) − Master/slave function − SPI supported (for both master and slave modes)
3	Baud rate	− Dedicate baud rate generator provided (configured with a 15-bit reload counter; in master mode operation) − An external clock can be entered (in the slave mode operation).
4	Data length	Variable from 5 bits to 16 bits.
5	Received error detection	Overrun error
6	Interrupt request	− Received interrupt (a received completion, an overrun error) − Transmit interrupt (a transmit data empty, a transmit bus idle) − Transmit FIFO interrupt (when transmit FIFO is empty) − DMA (Transmit/Received) transferring support functions are available.
7	Serial chip select	− One channel control (single control) − Setup/hold/deselect time can be set to be changeable. − Active level can be set for each channel.
8	Synchronous transmit function	− Data can be sent at a specific period automatically in synchronization with serial timer.
9	Timer function	− 16-bit serial timer is mounted. − Operation clock division ratio can be selected from 1/1 to 1/256.
10	Synchronous mode	− Master or slave function
11	Pin access	− The serial data output pin can be set to "1".
12	FIFO options	− FIFO for transmit/received installed (maximum capacity: 128 bytes for transmit FIFO, 128 bytes for received FIFO) * − Transmit FIFO or received FIFO can be selected. − Transmit data can be resent. − Received FIFO interrupt timing can be changed via software. − FIFO resetting is supported independently.

* : The FIFO capacity size varies depending on the product.

2. CSIO (Clock Synchronous Serial Interface) interrupts

The CSIO interrupts contain the received interrupt, the transmit interrupt, and the status interrupt. These interrupt requests can be generated if

- A received data is set in the Received Data Register (RDR) or a data received error occurs.
- A transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register and the data transmission is started
- The transmit bus is idle (No data transmission occurs).
- A transmit FIFO data is requested.
- The serial timer comparison value (STMCR) and the serial timer value (STMR) match.
- The chip select error occurs.

CSIO interrupts

Table 2-1 shows the CSIO interrupt control bits and the interrupt factors.

Table 2-1 CSIO interrupt control bits and interrupt factors

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Operation to clear interrupt request flag
Reception	RDRF	SSR	A single-byte reception	SCR:RIE	Reading from the received data register (RDR)
			Reception of a data volume matching the value set for FBYTE.		Reading from the Received Data Register (RDR) until received FIFO is emptied
			The FRIIE bit is "1", received FIFO contains valid data, and the Received Idle state continues more than 8 bits time hours.		
	ORE	SSR	Overrun error		Setting the Received Error Flag Clear bit (SSR:REC) to "1"
Transmission	TDRE	SSR	The Transmit Data Register is empty.	SCR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *
	TBI	SSR	No data transmission	SCR:TBIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *
Transmission	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" or transmit FIFO is full.
	CSE	SACAR	In Slave mode (SCR:MS=1), or when the serial chip select pin is in inactive master mode (SCR:MS=0) at transferring, the transmit count is the set value of TBYTE or less and the next transmit data is not written in TDR (SSR:TDRE=1)	SACSR:CSEIE	Writing “0” to the Chip Select Flag Bit (SACSR:CSE).

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Operation to clear interrupt request flag
Status	TINT	SACSR	The values of the Serial Timer Register (STMR) and the Serial Timer Comparison Register (STMCR) match.	SACSR:TINTE	Writing “0” to the Timer Interrupt Flag bit (SACSR:TINT).

* : Set the TIE bit to "1" only after the TDRE bit has been set to "0".

2.1 Received interrupt and flag set timing

Data reception can be interrupted by a Received Completion (SSR:RDRF=1) or a Received Error Occurrence (SSR:ORE=1).

Received interrupt and flag set timing

When the last data bit is detected, the received data is stored in the Received Data Register (RDR). When the data reception is completed (SSR:RDRF=1) or when a data received error occurs (SSR:ORE=1), each flag is set. If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt occurs.

Note:

- *If a received error occurs, data in the Received Data Register (RDR) is invalidated.*

Figure 2-1 Data receiving and flag set timing

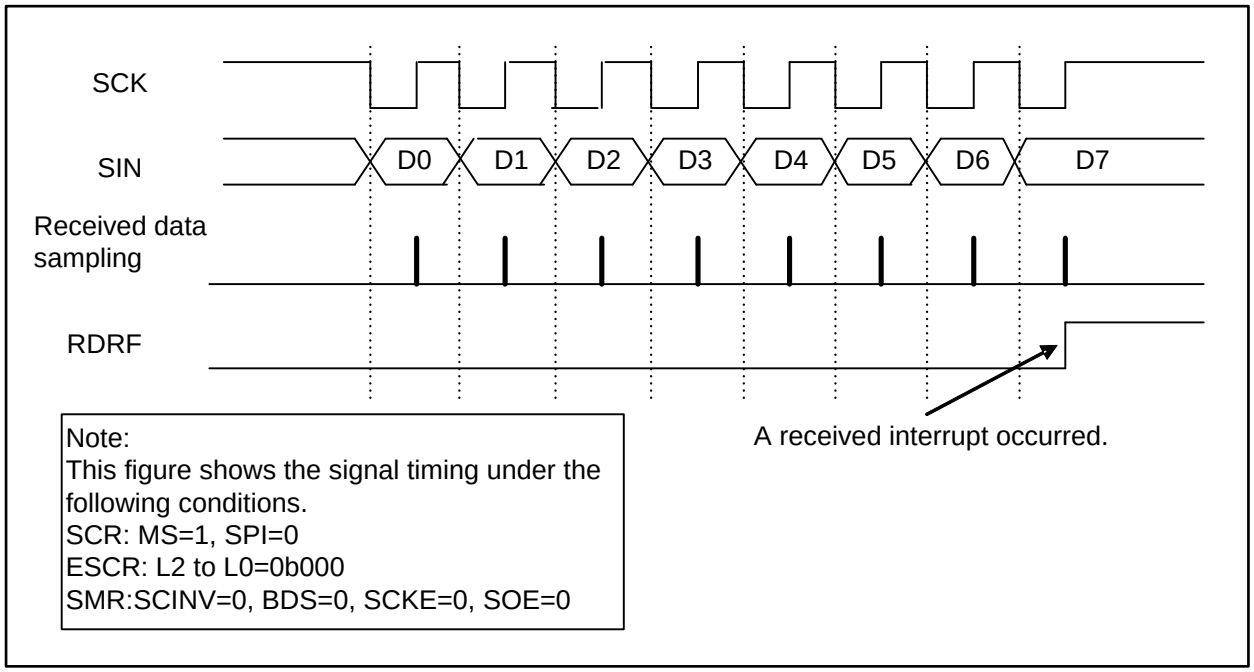
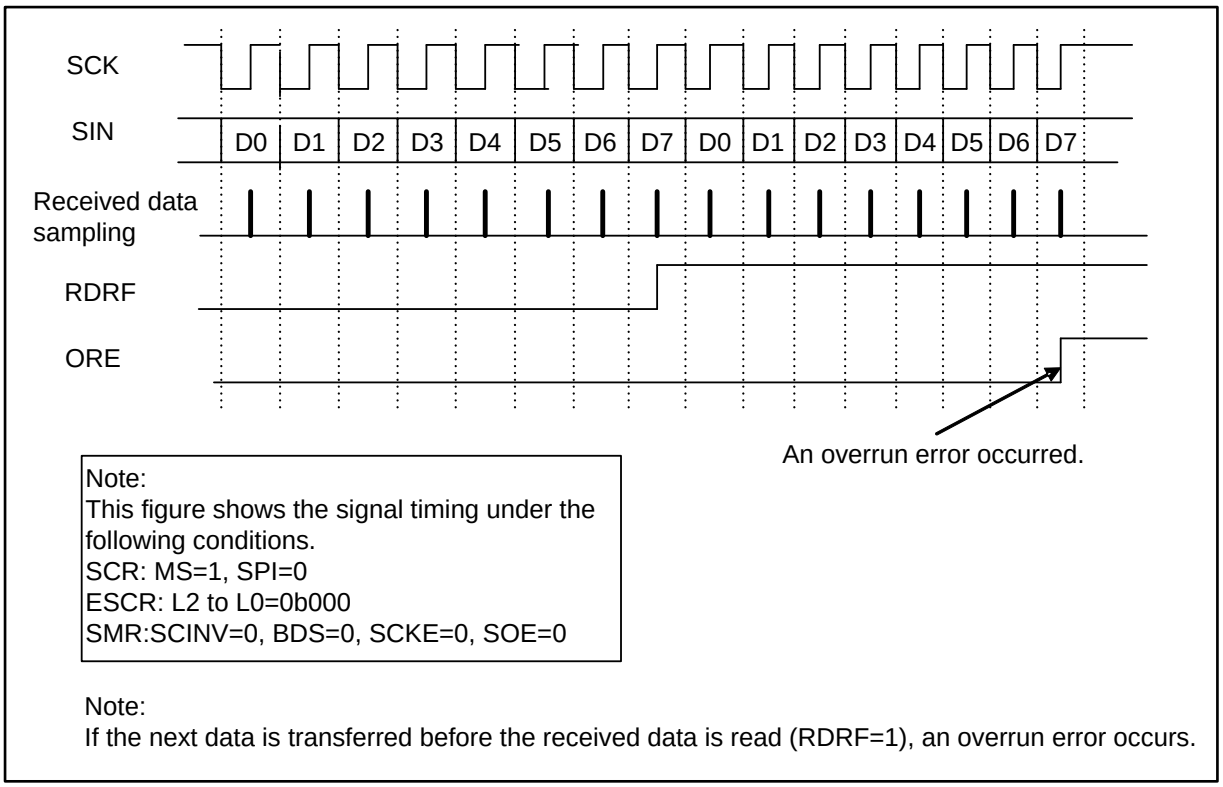


Figure 2-2 ORE (Overrun Error) flag set timing



Note:
This figure shows the signal timing under the following conditions.
SCR: MS=1, SPI=0
ESCR: L2 to L0=0b000
SMR:SCINV=0, BDS=0, SCKE=0, SOE=0

Note:
If the next data is transferred before the received data is read (RDRF=1), an overrun error occurs.

2.2 Interrupt and flag set timing when received FIFO is used

If received FIFO is used, an interrupt occurs when the FBYTE data (preset for the FBYTE register (FBYTE)) is received.

Received interrupt and flag set timing when received FIFO is used

- If received FIFO is used, an interrupt occurs depending on the value set for the FBYTE register.
- When the amount of data set for transfer count in the FBYTE register is received, the received data full flag bit (SSR:RDRF) of the Serial Status Register is set to "1". If a received interrupt (SCR:RIE) is enabled during this time, a received interrupt occurs.
- If the following two conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the received data full flag bit (RDRF) is set to "1".
  - The received FIFO idle detect enable bit (FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.

If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to "0". If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.

- When the received data (RDR) is all read and received FIFO is emptied, the received data full flag (SSR:RDRF) is cleared.
- If the display of the valid received data amount is the same as the FIFO capacity and if the next data is received, an overrun error (SSR:ORE=1) occurs.

Figure 2-3 Received interrupt occurrence timing when received FIFO is used

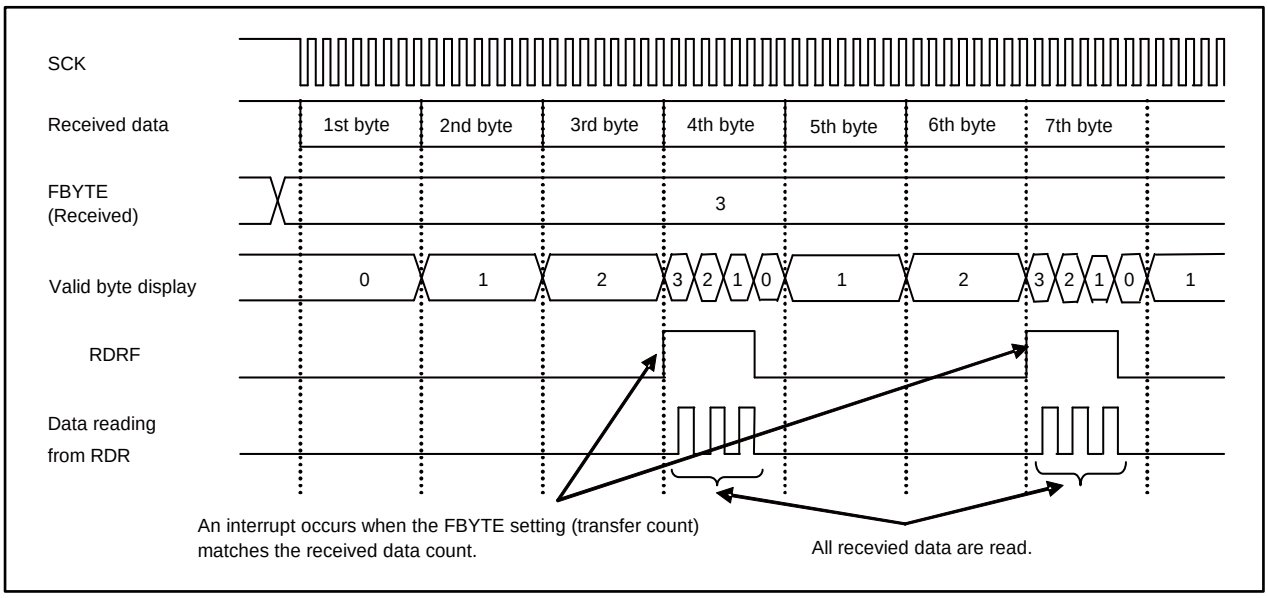
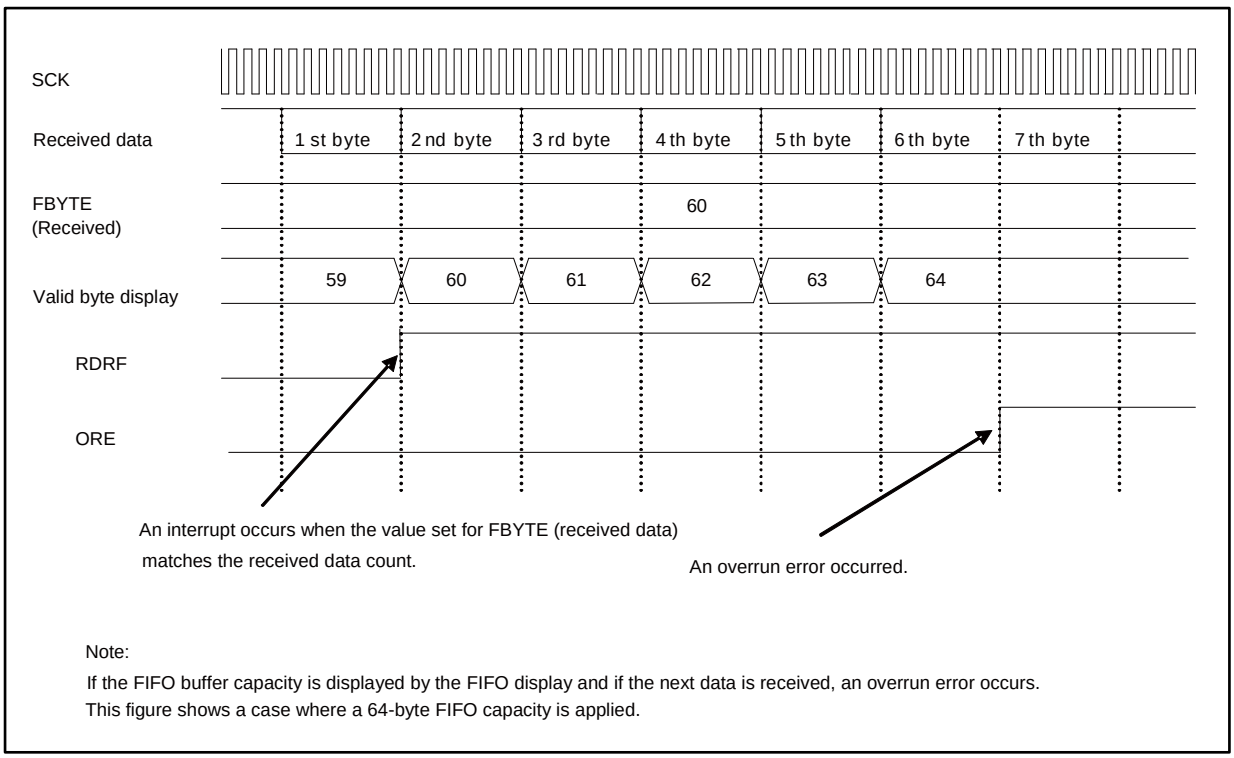


Figure 2-4 ORE (Overrun Error) flag bit set timing



Note:
If the FIFO buffer capacity is displayed by the FIFO display and if the next data is received, an overrun error occurs.
This figure shows a case where a 64-byte FIFO capacity is applied.

2.3 Transmit interrupt and flag set timing

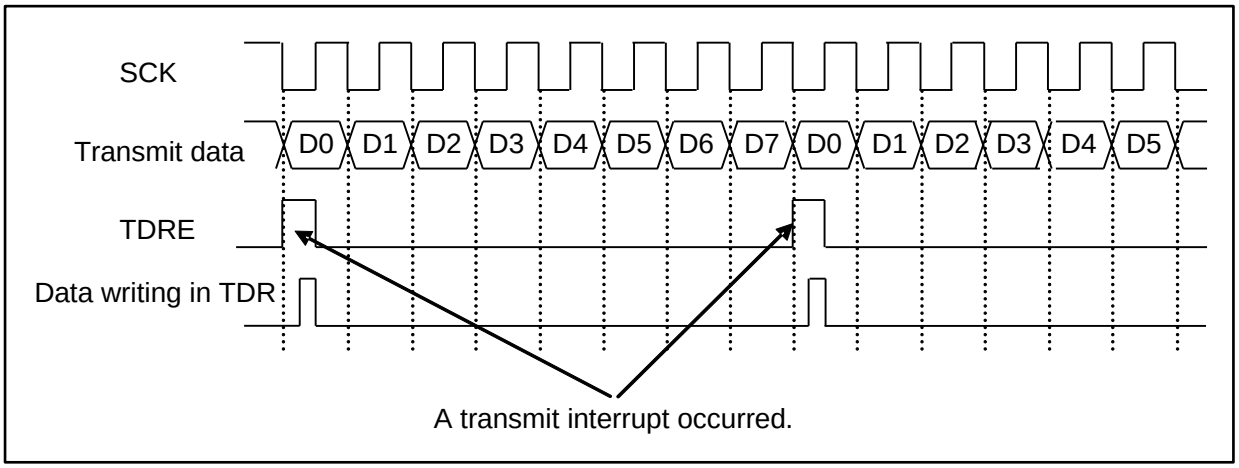
A transmit interrupt occurs if transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE=1) and the data transmission is started, or if no data is transmitted (SSR:TBI=1).

Transmit interrupt and flag set timing

- Transmit data empty flag (SSR:TDRE) set timing

After data has been transferred from the Transmit Data Register (TDR) to the transmit shift register, the next data can be written in the TDR (SSR:TDRE=1). If a transmit interrupt is enabled (SCR:TIE=1) during this time, a transmit interrupt occurs. As the SSR:TDRE bit is read only, the SSR:TDRE bit is cleared to "0" when data is written to the Transmit Data Register (TDR).

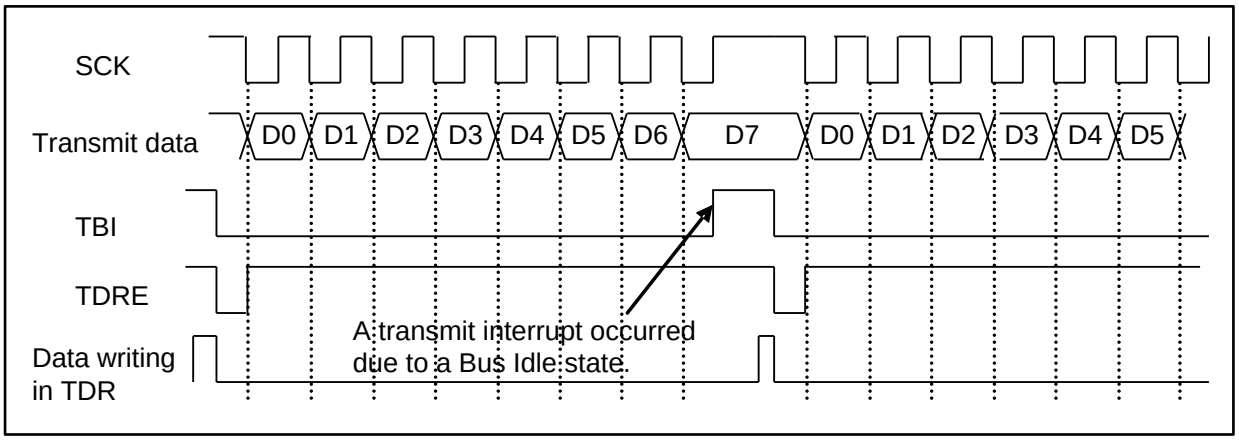
Figure 2-5 Transmit data empty flag (SSR:TDRE) set timing



- Transmit bus idle flag (SSR:TBI) set timing

If the Transmit Data Register is empty (SSR:TDRE=1) and no data is transmitted, the SSR:TBI bit is set to "1". If a transmit bus idle interrupt is enabled (SCR:TBIE=1) during this time, a transmit interrupt occurs. When transmit data is written to the Transmit Data Register (TDR), both the SSR:TBI bit and the transmit interrupt request are cleared.

Figure 2-6 Transmit bus idle flag (TBI) set timing (SCSCR:CSEN0=0, SACSR:TSYNE=0)



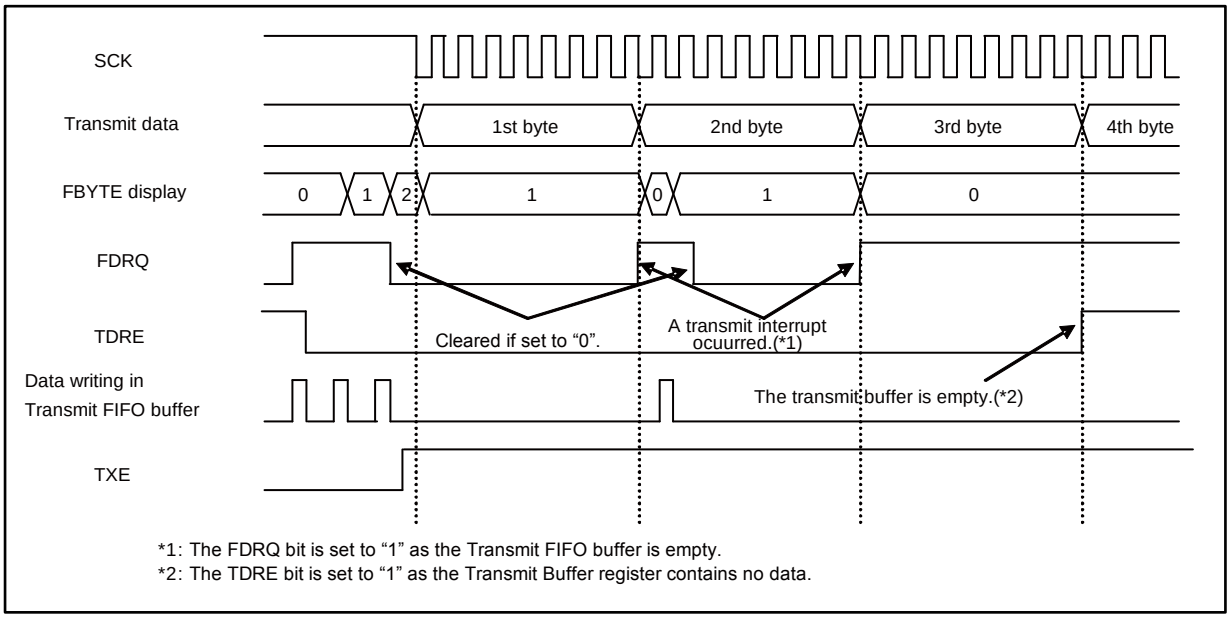
2.4 Interrupt and flag set timing when transmit FIFO is used

When transmit FIFO is used, an interrupt occurs if the buffer contains no data.

Transmit interrupt and flag set timing when transmit FIFO is used

- If transmit FIFO contains no data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "1".
  If a FIFO transmit interrupt is enabled (FCR1:FTIE=1) during this time, a transmit interrupt occurs.
- If you have written the required data in transmit FIFO after occurrence of a transmit interrupt, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- When transmit FIFO is filled with data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "0".
- You can check a presence of data in transmit FIFO by reading the FIFO Byte Register (FBYTE).
  If FBYTE=0x00, no data exists in transmit FIFO.

Figure 2-7 Transmit interrupt occurrence timing when transmit FIFO is used



*1: The FDRQ bit is set to “1” as the Transmit FIFO buffer is empty.
*2: The TDRE bit is set to “1” as the Transmit Buffer register contains no data.

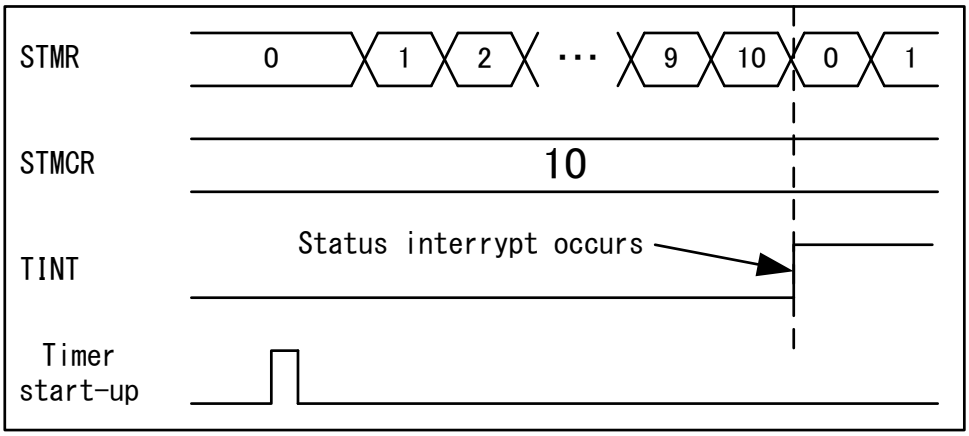
2.5 Timer Interrupt Occurrence and Flag Setting Timing

Timer interrupt occurs when the values of the Serial Timer Register (STMR) and The Serial Timer Comparison Register (STMCR) match.

Timer Interrupt Occurrence and Flag Setting Timing

- When the values of the Serial Timer Register (STMR) and the Serial Timer Comparison Register (STMCR) match, the Timer Interrupt Flag (SACSR:TINT) is set to “1”.
  At this time, when the Timer Interrupt is enabled (SACSR:TINTE=1), the Status Interrupt occurs.

Figure 2-8 Timer Interrupt Occurrence Timing



2.6 Chip Select Error Occurrence and Flag Setting Timing

In Master mode (SCR:MS=0), the Chip Select Error occurs when only the data of the frame count not greater than the TBYTE set value is transmitted and no valid data exists in the Transmit Data Register (TDR). Moreover, the Chip Select Error occurs at transmitting in Slave Mode Operation (SCR:MS=1) when Serial Chip Select pin becomes inactive.

Chip Select Error Occurrence and Flag Setting Timing

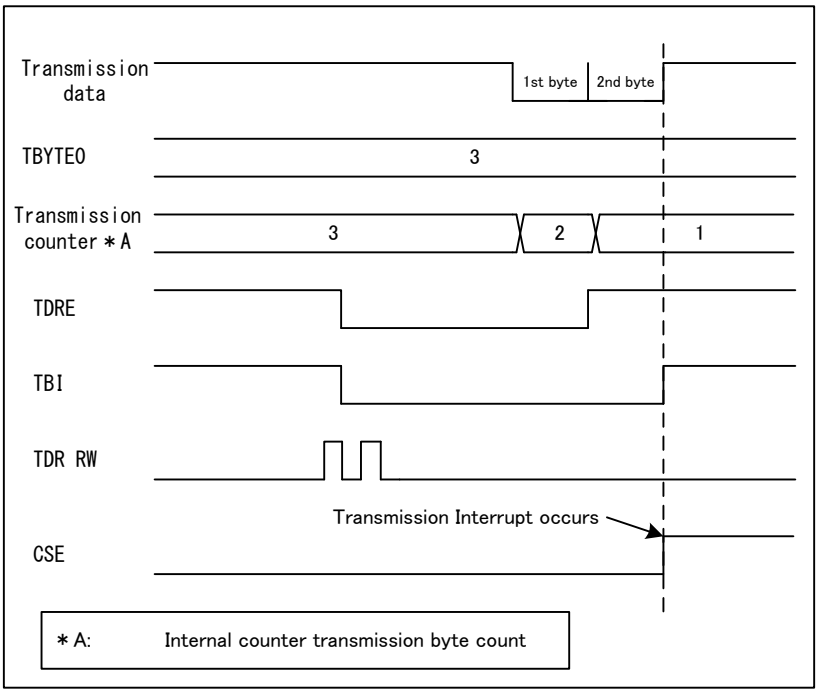
■ Master Mode (SCR:MS=0)

The Chip Select Error occurs when the Transmit Byte Error is enabled (TBEEN=1) and no valid data exists in the Transmit Data Register (TDR) before the data frame of TBYTE set value (SSR:TDRE=1) If the one of the following conditions meets:

- When the Chip Select is used
- When the synchronous transmission with the Serial Timer is used.

At this time, when the Chip Select Error Interrupt is enabled (SACSR:CSEIE=1), the transmit error occurs.

Figure 2-9 Chip Select Error Occurrence Timing (SCSCR:CSEN0=0, SACSR:TSYNE=1)



Notes:

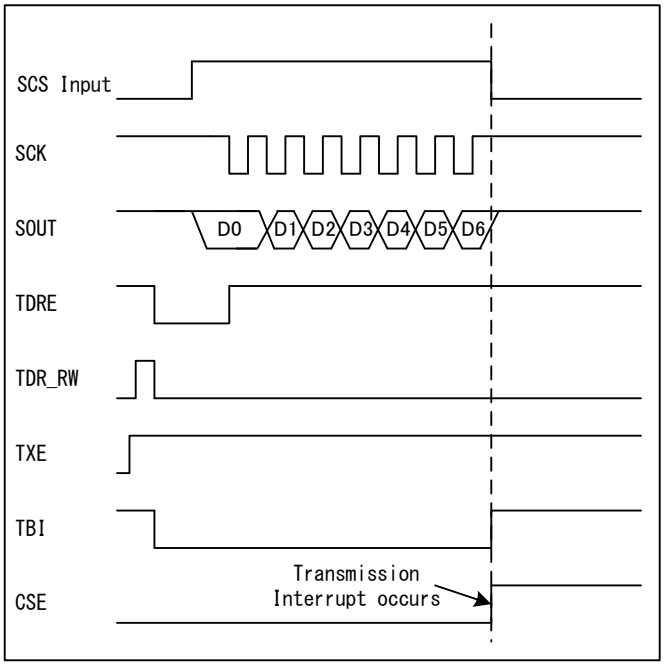
- *When the Serial Chip Select is used, the Chip Select Error Flag (SACSR:CSE) is set to "1" after the elapse of Deselect Time from the chip Select Error occurrence. Moreover, when the transmit data is written to the Transmit Data Register (TDR) in the Hold Delay Time, the transmission operation does not start and the Chip Select Error Flag (SACSR:CSE) is set to "1" after the elapse of the Deselect Time.*
- *When the Chip Select Error Flag (SACSR:CSE) is set to "1", the transmission operation does not start even if the transmit data is written to the Transmit Data Register (TDR).*
- *While using the Synchronous Transmission, when the Chip Select Error Flag(SACSR:CSE) is set to "1", the transmission operation does not start even if the following condition is met:*
  - *At the transmission in synchronization with serial timer, the Real Timer Register (STMR) and the Serial Timer Comparison Register match.*

■ Slave Mode (SCR:MS=1)

The Chip Select Error occurs when the Serial Chip Select pin becomes inactive while transmitting (SSR:TBI=1).

At this time, if the Chip Select Error Interrupt is enabled (SACSR:CSEIE=1), the Transmission Interrupt occurs.

Figure 2-10 Chip Select Error Occurrence Timing (CSLVL=0, SCR:SPI=0)



3. CSIO (Clock Synchronous Serial Interface) operations

The clock synchronous data transfer is used.

3.1 Normal transfer (I)

Features

	Item	Description
1	Serial clock (SCK) signal mark level	"HIGH"
2	Transmit data output timing	SCK signal falling edge
3	Received data sampling	SCK signal rising edge
4	Data length	5 bits to 16 bits

Register settings

The register values required for normal transfer (I) are listed on the table below.

Table 3-1 Normal transfer (I) register settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SMR	0	1/0	0	*	*	*	*	*	0	1	0	0	0	*	1/0	*
SSR/	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	L3	-	WT1	WT0	L2	L1	L0
ESCR	0	-	-	-	-	-	-	-	0	*	-	*	*	*	*	*
TDR1/0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RDR1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
TDR3/2	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RDR3/2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1 : Set to "1".

0 : Set to "0".

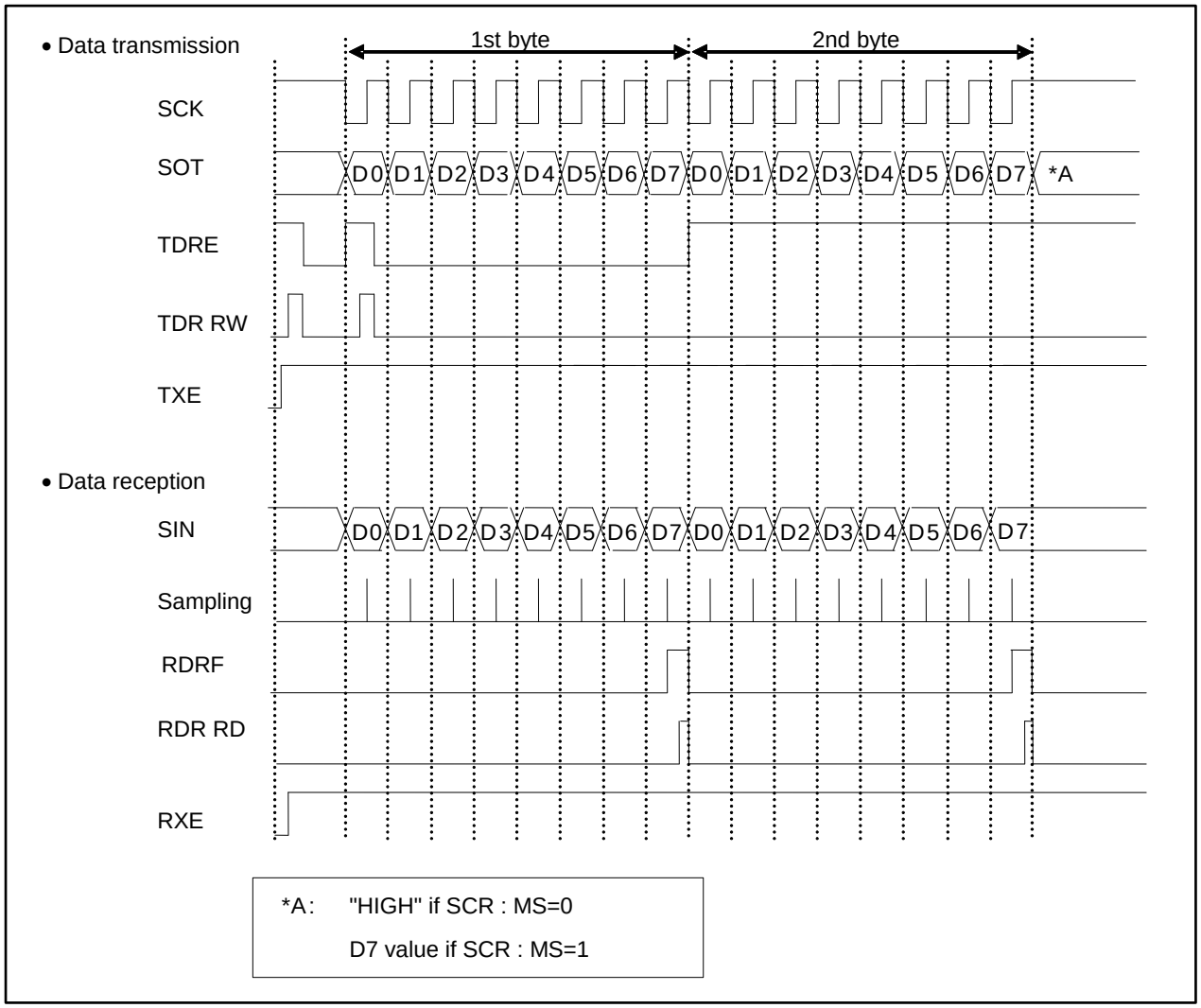
* : User-dependent values

Note:

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

- *During master mode operation: SCR:MS=0, SMR:SCKE=1*
- *During slave mode operation: SCR:MS=1, SMR:SCKE=0*

Normal transfer (I) timing chart (When serial chip select pin is not used.)



Master mode operation (SCR:MS=0, SMR:SCKE=1)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) output.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the received data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

Notes:

- *To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.*
- *If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.*

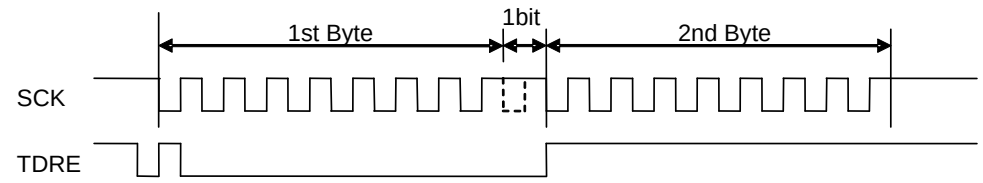
■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

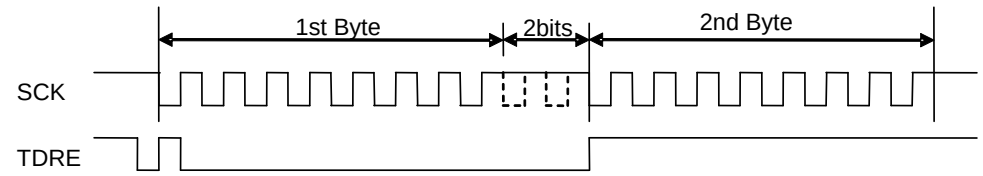
■ Continuous data transmit or reception waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

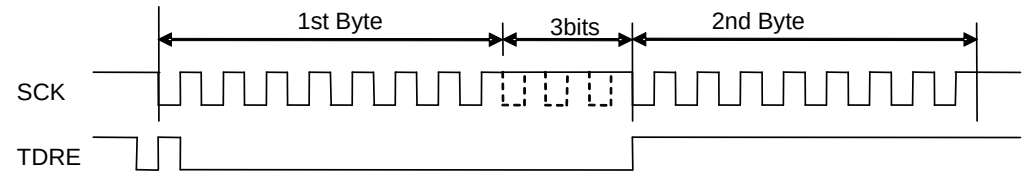
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- ESCR:WT1, ESCR:WT0=10 (in master mode operation)



- ESCR:WT1, ESCR:WT0=11 (in master mode operation)



Slave mode operation (SCR:MS=1, SMR:SCKE=0)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) input.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

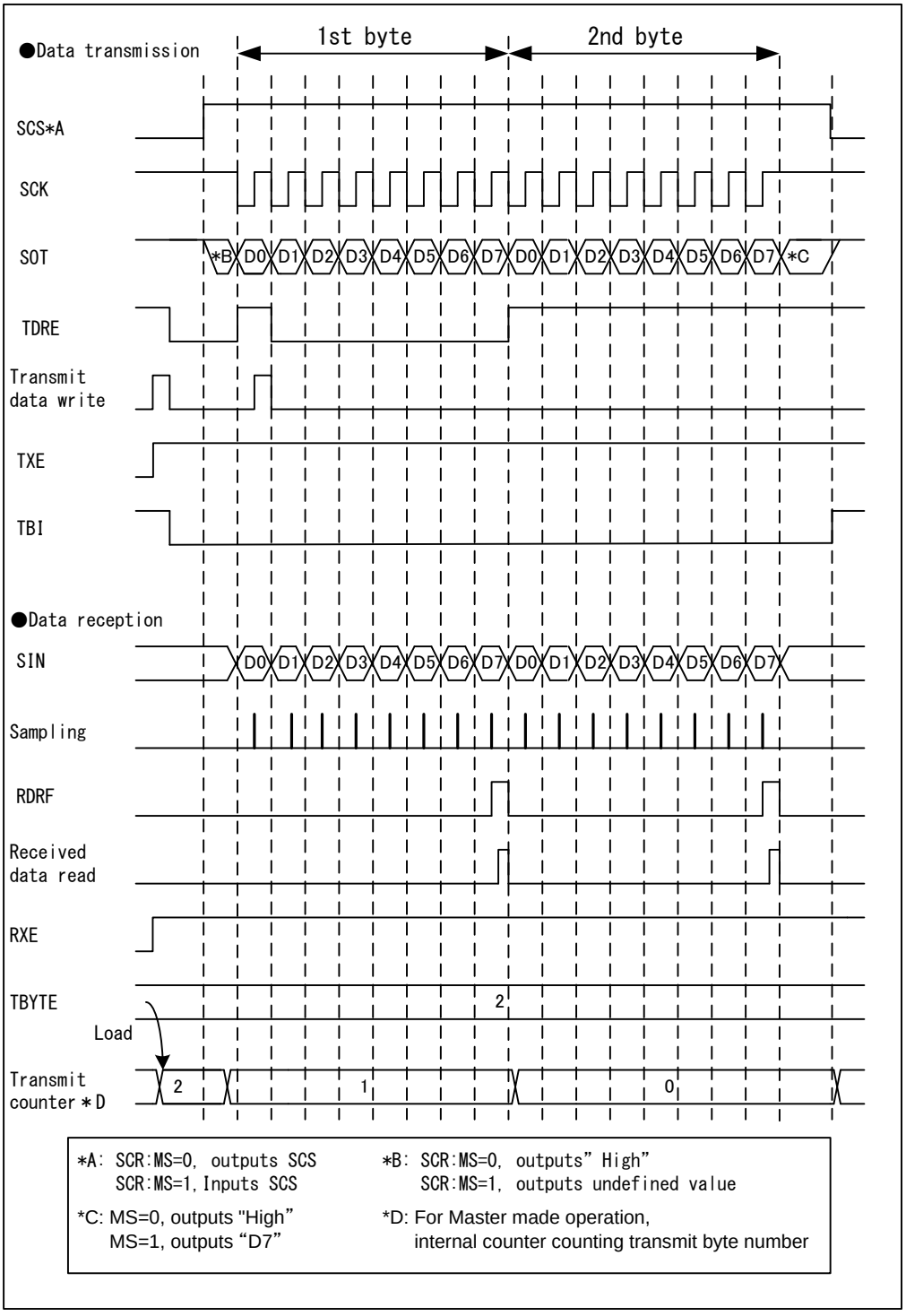
■ Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the received data is sampled at a rising edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a rising edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

Normal transmission (I) Timing Chart (When Serial Chip Select pin is used)



Master mode operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: “n” is the number of the serial chip select pin used

■ Data Transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1), and data reception is disabled (SCR:RXE=0) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". And then, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, this causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) output.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. After completing the times of the data transmission specified with TBYTE, the serial clock stops.
4. After the elapse of the hold time of the Serial Chip Select pin following the Serial Clock stop, the Serial Chip Select pin (SCS) becomes inactive. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

■ Data Reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1), data reception is enabled (SCR:RXE=1), and a dummy data is written to TDR, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, the received data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception is completed for the time specified with TBYTE, the serial clock is stopped.
5. After the serial clock is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

Notes:

- *To perform only the data reception, write a dummy data to TDR in order to output the Serial Clock (SCS).*

■ Data transmission and reception

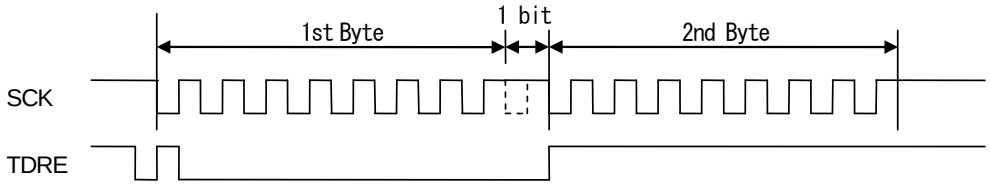
1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" . Then, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a rising edge of the serial clock (SCK) output during the data transmission and reception. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception and transmission are completed for the time specified with TBYTE, the serial clock is stopped.

5. After the serial clock is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

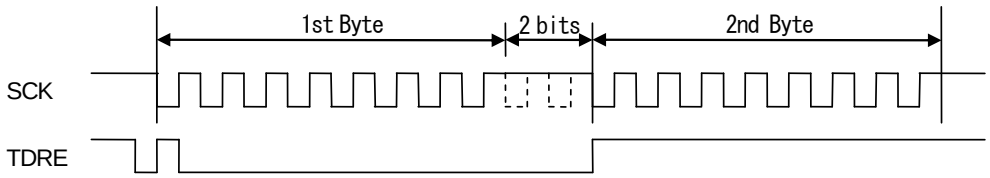
■ Continuous Data Transmit or Reception Waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

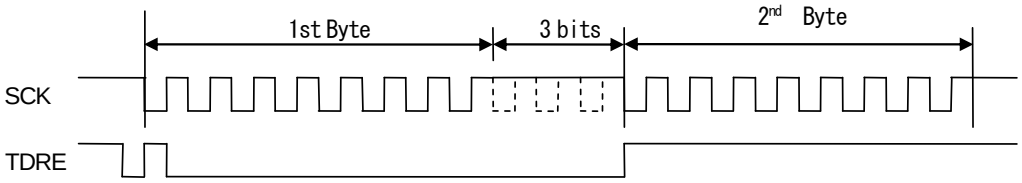
・ ESCR:WT1=0, ESCR:WT0=1(in master mode operation)



・ ESCR:WT1=1, ESCR:WT0=0(in master mode operation)



・ ESCR:WT1=1, ESCR:WT0=1(in master mode operation)



Slave mode operation (SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=1, SCSCR:CSOE=0, SCSCR:SCAM=0)

■ Data Transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0".
2. When the Serial Chip Select pin (SCS) becomes active, the transmission operation is started and the transmit data is output in synchronization with the falling edge of serial clock (SCK) input.
3. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
4. When the Serial Chip Select pin (SCS) becomes inactive, the transmission operation is stopped and the serial output pin (SOT) becomes “High”.

■ Data Reception

1. If the serial data output is disabled (SMR:SOE=0) , data reception is enabled (SCR:RXE=1), and the serial chip select pin (SCS) becomes active, the data reception is started and the received data is sampled at a rising edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
3. The received data (RDR) can be read during this time.
4. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
5. When the serial chip select pin (SCS) becomes inactive, the data reception is stopped.

■ Data Reception and Transmission

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. During the data reception and transmission, the received data is sampled at a rising edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. When the serial chip select pin (SCS) becomes inactive, the data reception and transmission is stopped and the serial output pin (SOT) becomes “High”.

3.2 Normal transfer (II)

Features

	Item	Description
1	Serial clock (SCK) signal mark level	"LOW"
2	Transmit data output timing	SCK signal rising edge
3	Received data sampling	SCK signal falling edge
4	Data length	5 bits to 16 bits

Register settings

The register values required for normal transfer (II) are listed on the table below.

Table 3-2 Normal transfer (II) register settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SMR1	0	1/0	0	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SSR/	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	L3	-	WT1	WT0	L2	L1	L0
ESCR	0	-	-	-	-	-	-	-	0	*	-	*	*	*	*	*
TDR1/0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RDR1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
TDR3/2	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RDR3/2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

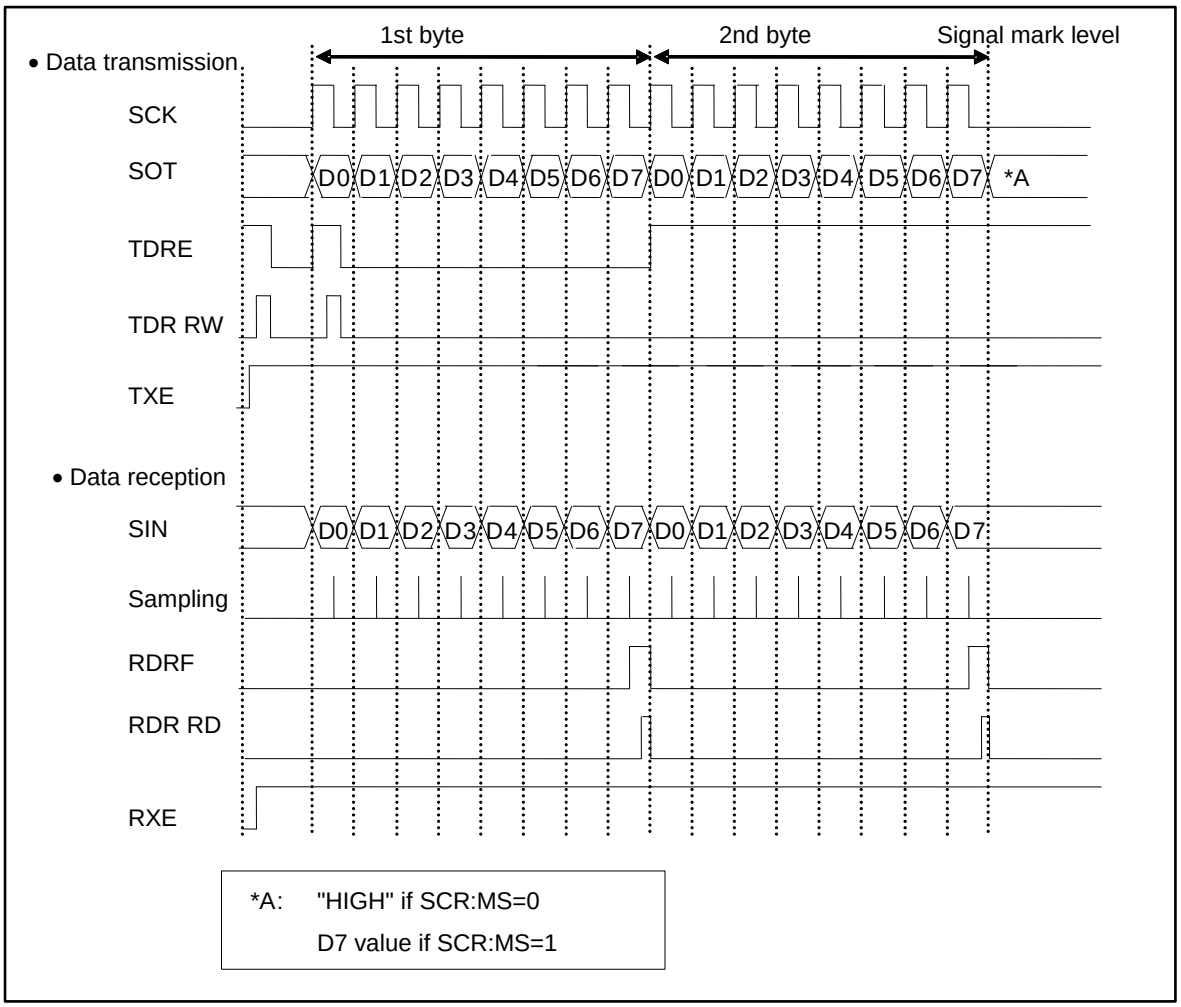
1 : Set to "1".
0 : Set to "0".
* : User-dependent values

Note:

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

- *During master mode operation: SCR:MS=0, SMR:SCKE=1*
- *During slave mode operation: SCR:MS=1, SMR:SCKE=0*

Normal transfer (II) timing chart (Serial chip select pin is not used)



Master mode operation (SCR:MS=0, SMR:SCKE=1)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the received data is sampled at a falling edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

Notes:

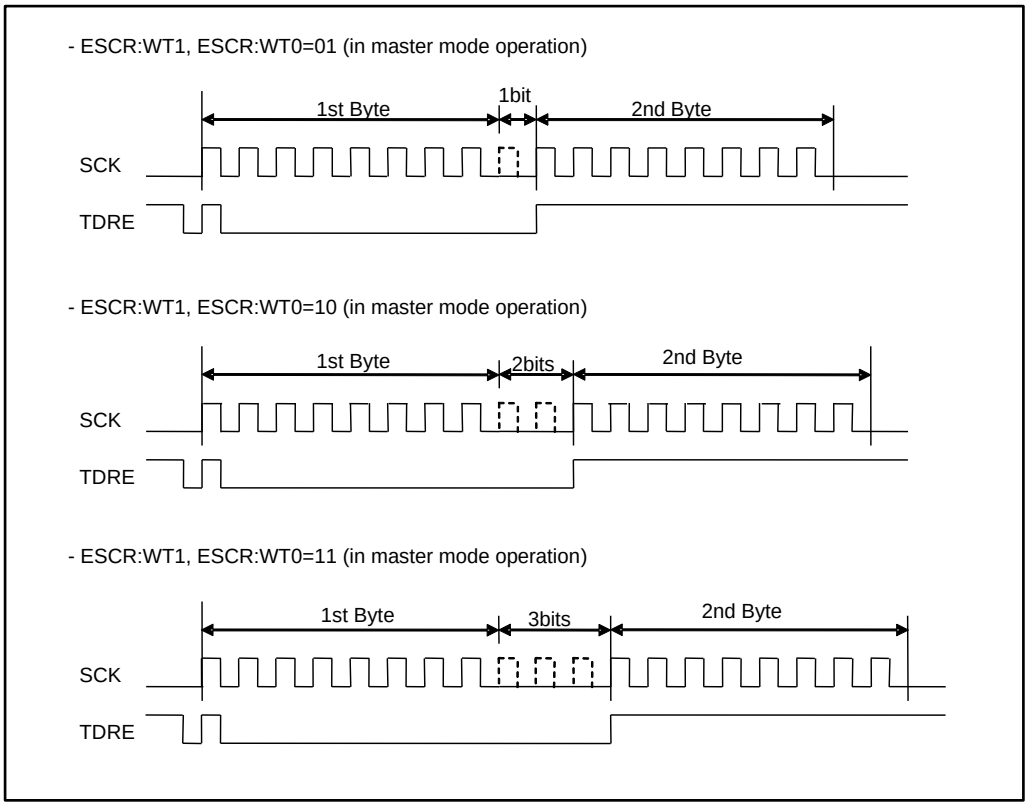
- *To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.*
- *If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.*

■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

■ Continuous data transmit or reception waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.



Slave mode operation (SCR:MS=1, SMR:SCKE=0)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a rising edge of the serial clock (SCK) input.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

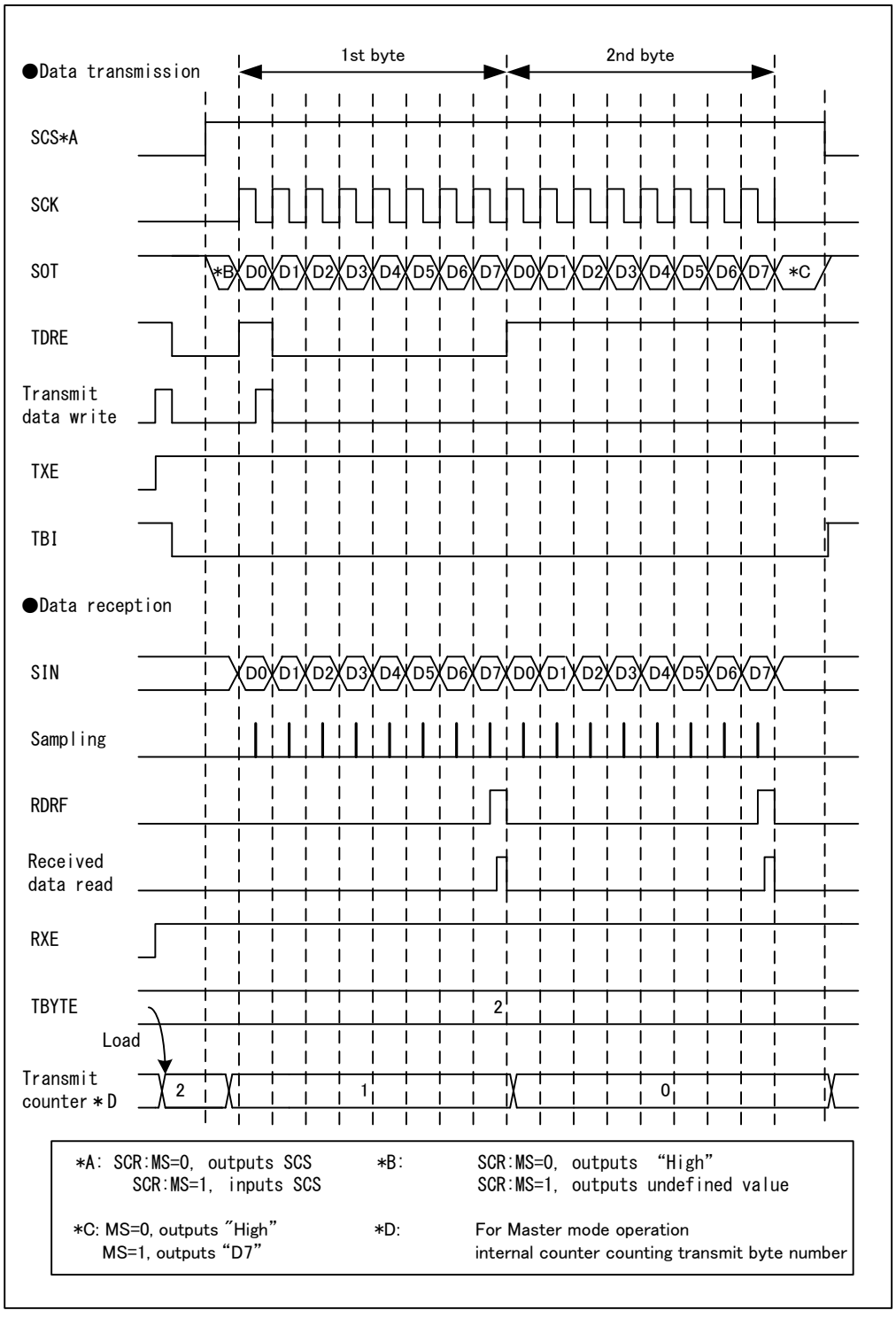
■ Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the received data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" and the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

Normal transfer (II) timing chart (Serial chip select pin is used)



Master mode operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: “n” is the number of the serial chip select pin used

■ Data Transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". Then, the serial chip select pin (SCS) becomes active and then, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, this causes the transmit data to be output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. After completing the times of the data transmission specified with TBYTE, the serial clock is stopped.
4. After the elapse of the hold time of the Serial Chip Select pin following the Serial Clock stop, the Serial Chip Select pin (SCS) becomes inactive. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

■ Data Reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1), data reception is enabled (SCR:RXE=1), and a dummy data is written to TDR, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, the received data is sampled at a falling edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception is completed for the time specified with TBYTE, the serial clock is stopped.
5. After the serial clock is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

Notes:

To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.

If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.

■ Data transmission and reception

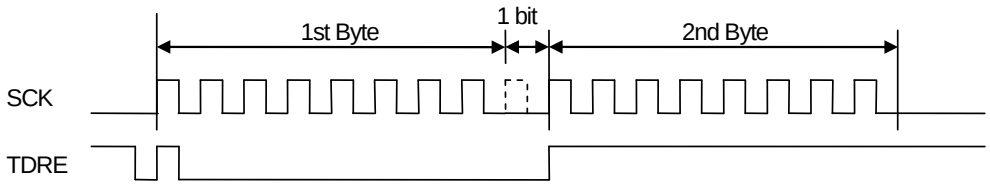
1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" . Then, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) output during the data transmission and reception. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

4. After the data reception and transmission are completed for the time specified with TBYTE, the serial clock is stopped.
5. After the serial clock output is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

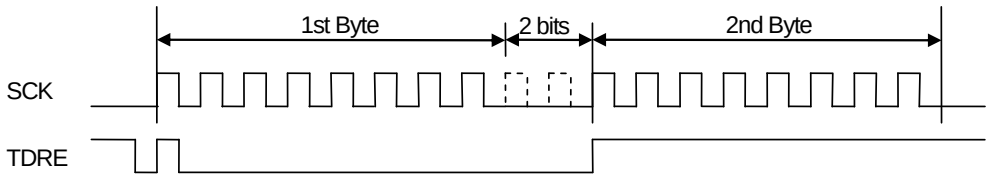
■ Continuous data transmit or reception waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

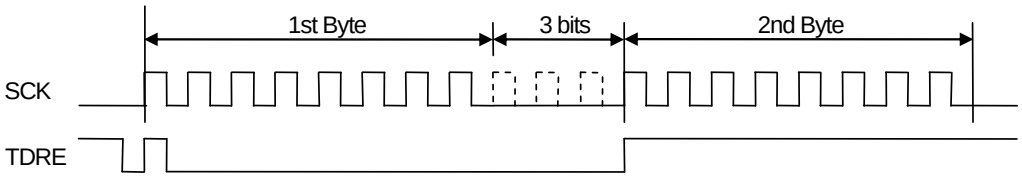
・ ESCR.WT1=0, ESCR.WT0=1(in master mode operation)



・ ESCR.WT1=1, ESCR.WT0=0(in master mode operation)



・ ESCR.WT1=1, ESCR.WT0=1(in master mode operation)



Slave mode operation(SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=1, SCSCR:CSOE=0, SCSCR:SCAM=0)

■ Data Transmission

1. If serial data output is enabled (SMR:SOE=1), and data transmission is enabled (SCR:TXE=1), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0".
2. When the Serial Chip Select pin (SCS) becomes active, the transmit data output is started. Then the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input.
3. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
4. If the Serial Chip Select pin (SCS) becomes inactive, the data transmission is stopped and the serial output pin (SOT) becomes “High”.

■ Data Reception

1. If the serial data output is disabled (SMR:SOE=0) , data reception is enabled (SCR:RXE=1), and the serial chip select pin (SCS) becomes active, the data reception is started and the received data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
3. The received data (RDR) can be read during this time.
4. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
5. When the serial chip select pin (SCS) becomes inactive, the data reception is stopped.

■ Data Transmission and Reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE bit is set to "0" . Then, the Serial Chip Select pin (SCS) becomes active and the serial clock output is started. After starting the Serial Clock output, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) input during the data transmission and reception. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. The serial clock output is stopped when the Serial Chip Select pin (SCS) becomes inactive and the serial output pin (SOT) becomes “High”.

3.3 SPI transfer (I)

Features

	Item	Description
1	Serial clock (SCK) signal mark level	"HIGH"
2	Transmit data output timing	SCK signal rising edge
3	Received data sampling	SCK signal falling edge
4	Data length	5 bits to 16 bits

Register settings

The register values required for SPI transfer (I) are listed on the table below.

Table 3-3 SPI transfer (I) register settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SMR	0	1/0	1	*	*	*	*	*	0	1	0	0	0	*	1/0	*
SSR/	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	L3	-	WT1	WT0	L2	L1	L0
ESCR	0	-	-	-	-	-	-	-	0	*	-	*	*	*	*	*
TDR1/0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RDR1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
TDR3/2	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RDR3/2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

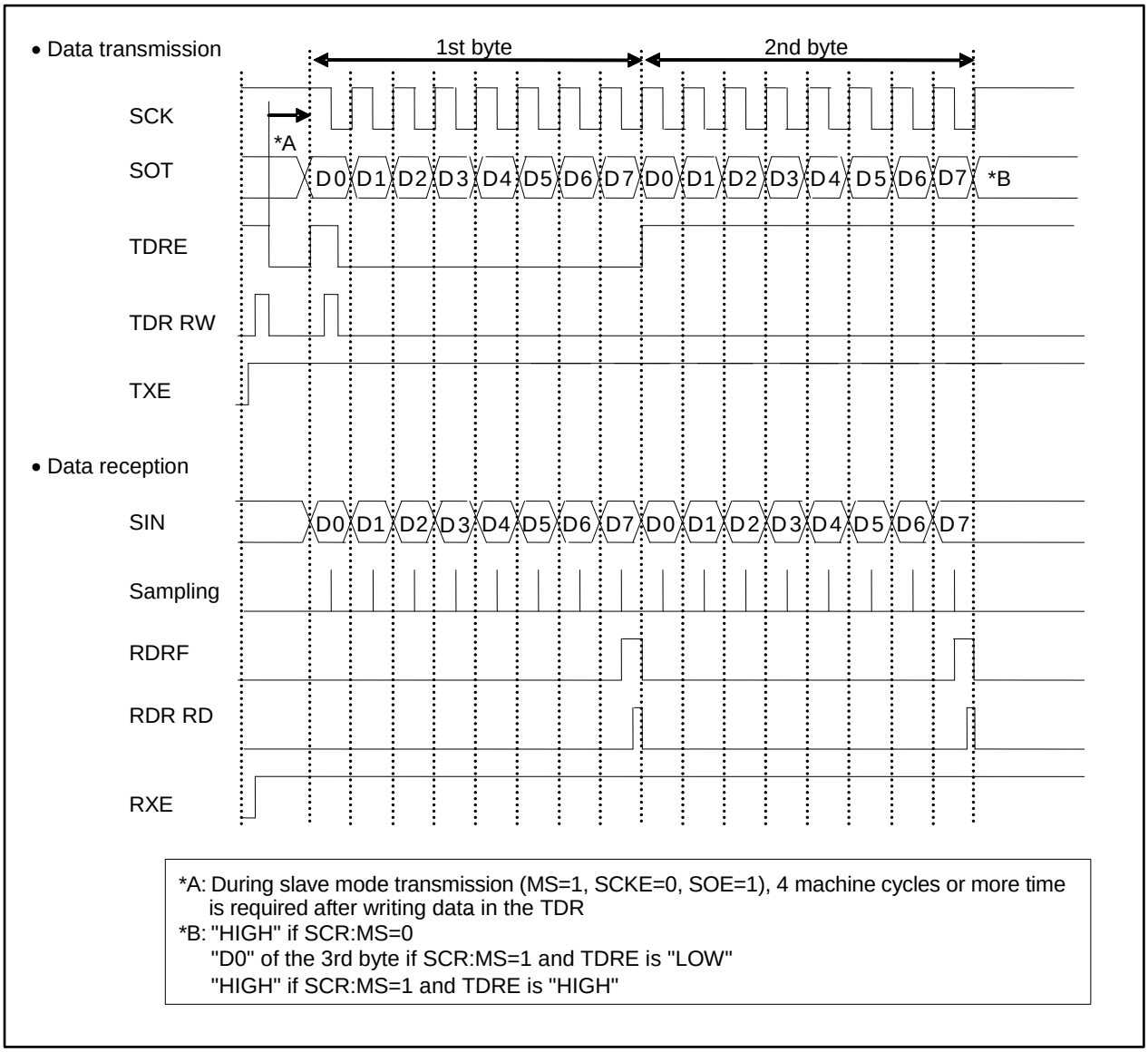
1 : Set to "1".
0 : Set to "0".
* : User-dependent values

Note:

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

- *During master mode operation: SCR:MS=0, SMR:SCKE=1*
- *During slave mode operation: SCR:MS=1, SMR:SCKE=0*

SPI transfer (I) timing chart (Serial chip select pin is not used)



*A: During slave mode transmission (MS=1, SCKE=0, SOE=1), 4 machine cycles or more time is required after writing data in the TDR
*B: "HIGH" if SCR:MS=0
"D0" of the 3rd byte if SCR:MS=1 and TDRE is "LOW"
"HIGH" if SCR:MS=1 and TDRE is "HIGH"

Master mode operation (SCR:MS=0, SMR:SCKE=1)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the first bit to output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the received data is sampled at a falling edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

Notes:

- *To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.*
- *If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.*

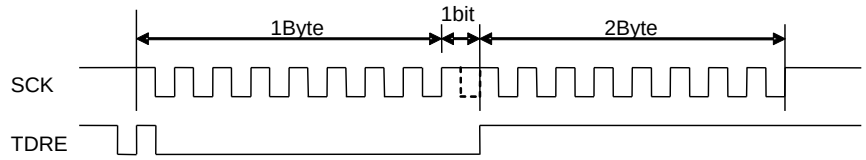
■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output. The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of the first serial clock. If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

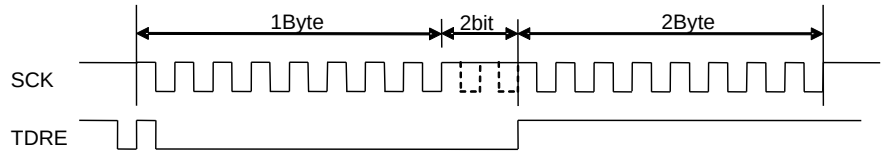
■ Continuous data transmit or reception waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

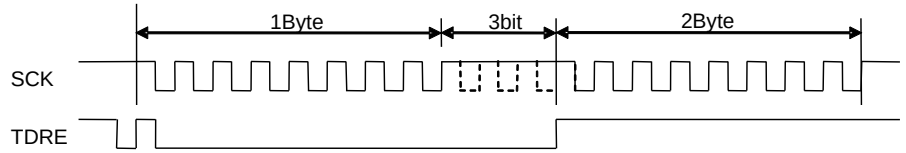
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- ESCR:WT1, ESCR:WT0=10 (in master mode operation)



- ESCR:WT1, ESCR:WT0=11 (in master mode operation)



Slave mode operation (SCR:MS=1, SMR:SCKE=0)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the first bit to output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output.
2. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

Note:

- *If data transmission is enabled (SCR:TXE=1) and if the first transmit data is written in the TDR at a time other than the serial clock (SCK) signal mark level, the first data bit is not output and the data transmission may fail. After the data transmission is enabled (SCR:TXE=1), the first transmit data must be written in the TDR at a signal mark level of the serial clock (SCK).*

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the received data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

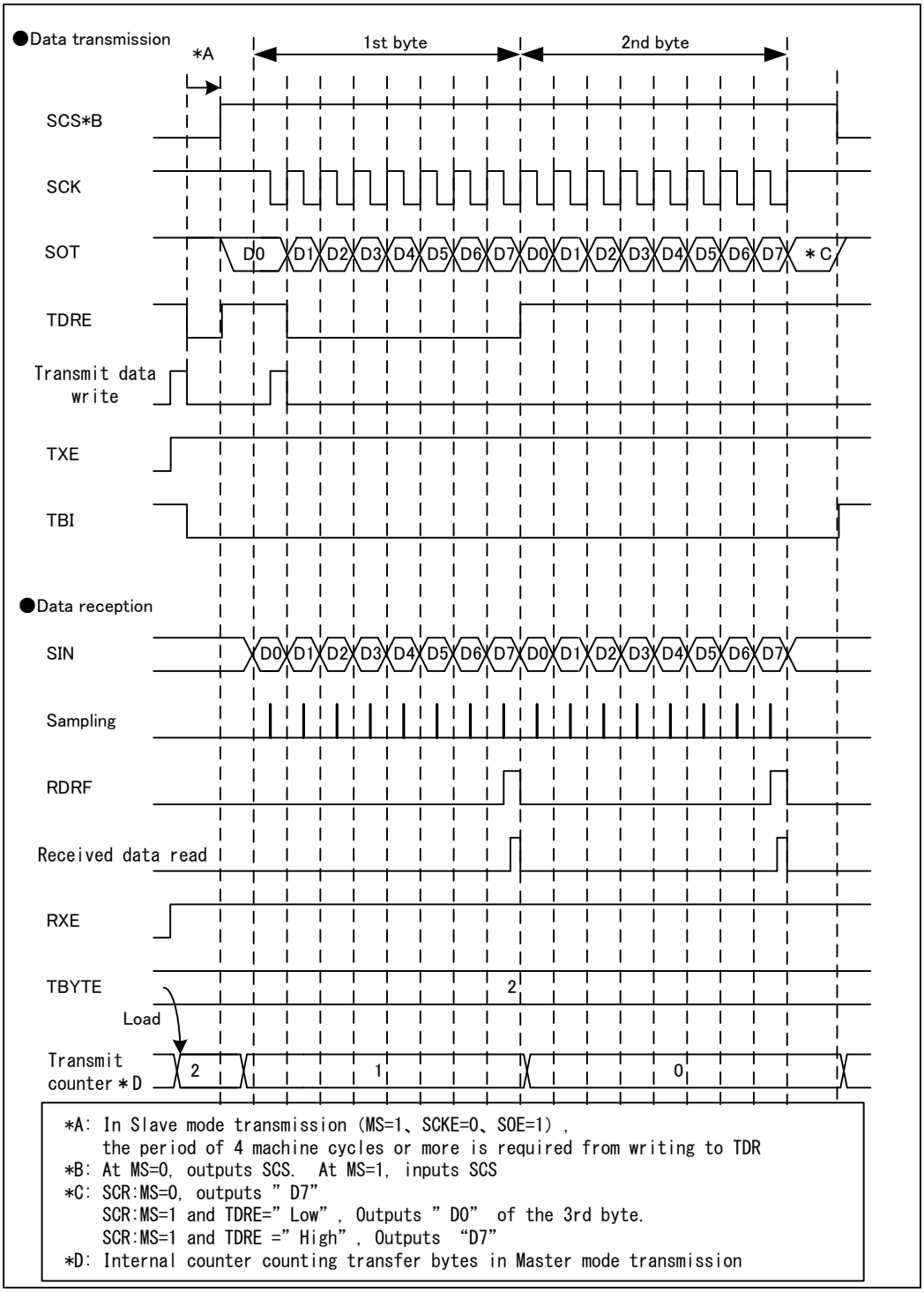
■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

■ Continuous switching from data reception to transmission

1. Disable the serial data output (SMR:SOE=0), enable a received interrupt (SCR:RIE=1), enable data reception (SCR:RXE=1), and enable data transmission (SCR:TXE=1). If dummy data is written in the TDR at a signal mark level of serial clock (SCK), the received data is sampled at a falling edge of serial clock (SCK) input.
2. To continue data reception, write a dummy data in the TDR between the time when a received interrupt is requested and when the next serial clock (SCK) rises.
3. To switch the data reception to the data transmission, enable the serial data output (SMR:SOE=1), disable a received interrupt (SCR:RIE=0), and disable data reception (SCR:RXE=0) between the time when a received interrupt is requested and when the next serial clock (SCK) rises. Also, output the transmit data in synchronization with a rising edge of serial clock after the transmit data has been written in the TDR and the data reception has completed.

SPI transfer (I) timing chart (Serial chip select pin is used)



Master mode operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: “n” is the number of the serial chip select pin used.

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". Then, the transmit data of the first bit is output and the Serial Chip Select pin (SCS) becomes active at the same time, and then, the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, this causes the transmit data to be output in synchronization with a rising edge of the serial clock (SCK) output.
2. The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of the first serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. After completing the times of the data transmission specified with TBYTE, the serial clock is stopped.
4. After the elapse of the hold time of the Serial Chip Select pin following the Serial Clock stop, the Serial Chip Select pin (SCS) becomes inactive. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the Serial Chip Select pin (SCS) becomes active and then, the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the serial clock output, the received data is sampled at a falling edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception is completed for the time specified with TBYTE, the serial clock output is stopped.
5. After the serial clock output is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

Notes:

- *To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.*
- *If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.*

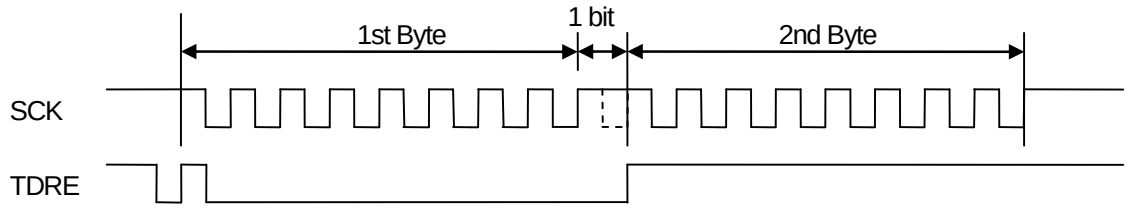
■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output and the the Serial Chip Select pin (SCS) becomes active at the same time. The serial clock output is started after the elapse of setup time of the Serial Chip Select pin. After the serial clock output, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output. The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of the first serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) output. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception is completed for the time specified with TBYTE, the serial clock output is stopped.
5. After the serial clock output is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

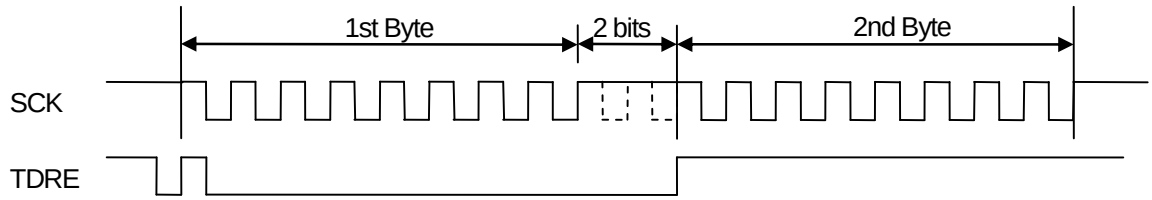
■ Continuous data transmit or reception waiting [∗]

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

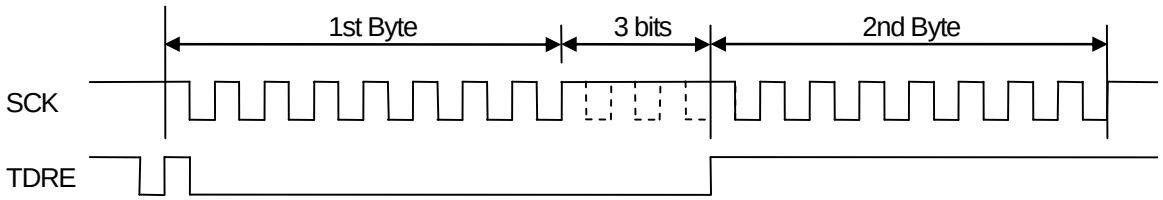
- ESCR.WT1=0, ESCR.WT0=1(in master mode



- ESCR.WT1=1, ESCR.WT0=0(in master mode operation)



- ESCR.WT1=1, ESCR.WT0=1(in master mode



Slave mode operation (SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN=1, SCSCR:SCAM=0) [*□]

■ Data Transmission

1. If serial data output is enabled (SMR:SOE=1), and data transmission is enabled (SCR:TXE=1), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0".
2. When the Serial Chip Select pin (SCS) becomes active, the transmit data output is started. Then the transmit data is output in synchronization with a rising edge of the serial clock (SCK) output.
3. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
4. If the Serial Chip Select pin (SCS) becomes inactive, the data transmission is stopped and the serial output pin (SOT) becomes “High”.

Note:

- *If the data transmission is enabled (SCR:TXE=1) and the first transmit data is written to TDR at a level other than the mark level, the data of the first bit is not output and the normal data transmission is not executed. After the data transmission is enabled (SCR:TXE=1), write the first transmit data to TDR when the serial clock (SCK) is at the Mark level*

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0) , data reception is enabled (SCR:RXE=1), and the serial chip select pin (SCS) becomes active, the data reception is started and the received data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
3. The received data (RDR) can be read during this time.
4. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
5. When the serial chip select pin (SCS) becomes inactive, the data reception is stopped.

■ Data reception and transmission

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0". Then, the Serial Chip Select pin (SCS) becomes active, so, the data transmission and reception is started and the first bit is output. The transmit data output is started after the elapse of setup time of the Serial Chip Select pin. After the data transmission and reception started, the transmit data is output in synchronization with a rising edge of the serial clock (SCK) input. When the first bit of the transmit data is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a falling edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. When the Serial Chip Select pin (SCS) becomes inactive, the serial clock output is stopped and the serial output pin (SOT) becomes “High”.

3.4 SPI transfer (II)

Features

	Item	Description
1	Serial clock (SCK) signal mark level	"LOW"
2	Transmit data output timing	SCK signal falling edge
3	Received data sampling	SCK signal rising edge
4	Data length	5 bits to 9 bits

Register settings

The register values required for SPI transfer (II) are listed on the table below.

Table 3-4 SPI transfer (II) register settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SMR	0	1/0	1	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SSR/	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	L3	-	WT1	WT0	L2	L1	L0
ESCR	0	-	-	-	-	-	-	-	0	*	-	*	*	*	*	*
TDR1/0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RDR1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
TDR3/2	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
RDR3/2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

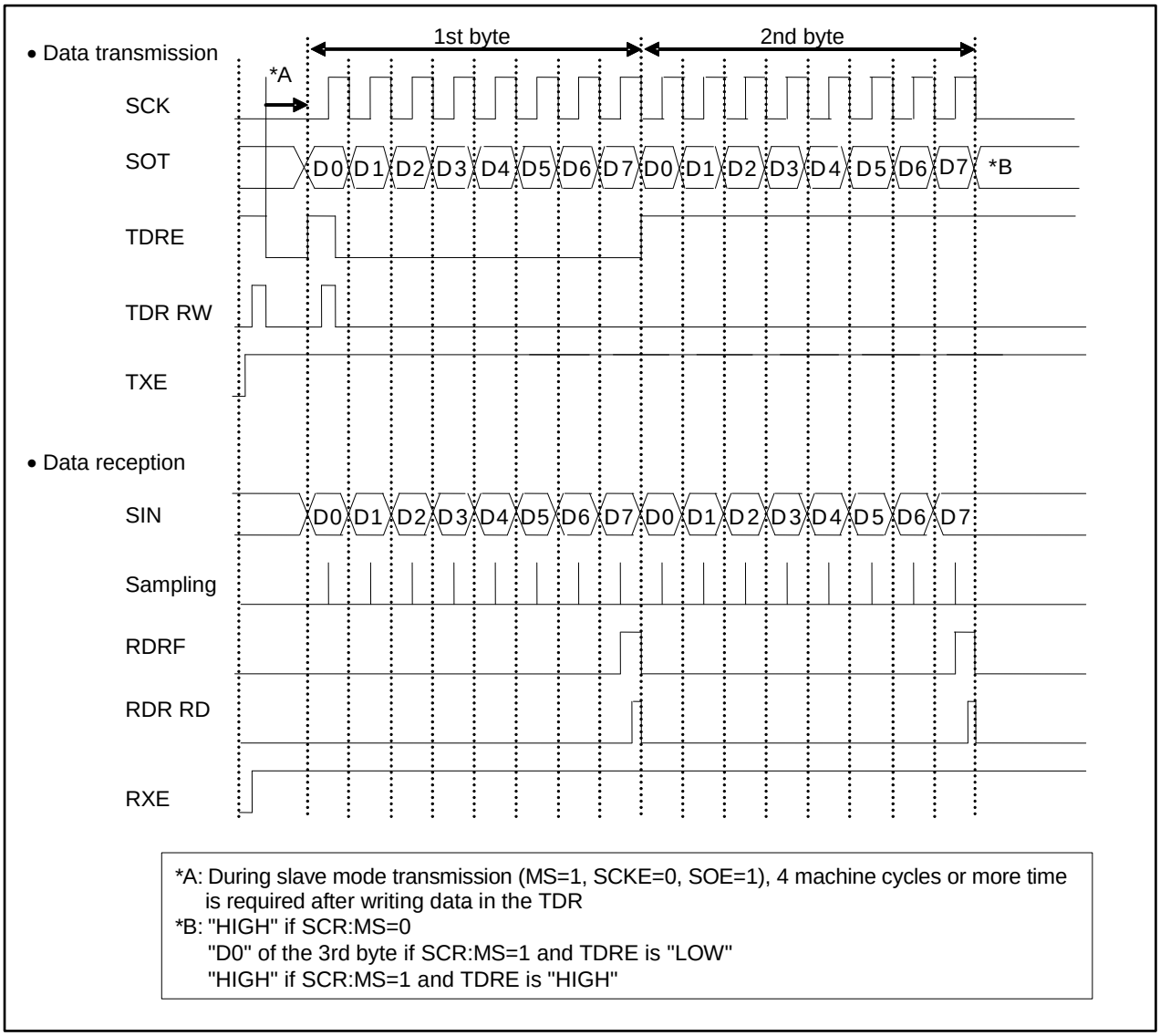
1 : Set to "1".
0 : Set to "0".
* : User-dependent values

Note:

The above bit setting (1/0) varies depending on the master or slave mode operation. Set as follows.

- *During master mode operation: SCR:MS=0, SMR:SCKE=1*
- *During slave mode operation: SCR:MS=1, SMR:SCKE=0*

SPI transfer (II) timing chart (Serial chip select pin is not used)



*A: During slave mode transmission (MS=1, SCKE=0, SOE=1), 4 machine cycles or more time is required after writing data in the TDR
*B: "HIGH" if SCR:MS=0
"D0" of the 3rd byte if SCR:MS=1 and TDRE is "LOW"
"HIGH" if SCR:MS=1 and TDRE is "HIGH"

Master mode operation (SCR:MS=0, SMR:SCKE=1)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) output.
2. The SSR:TDRE bit is set to "1" before a half cycle of a rising edge of the first serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the received data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

Notes:

- *To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.*
- *If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.*

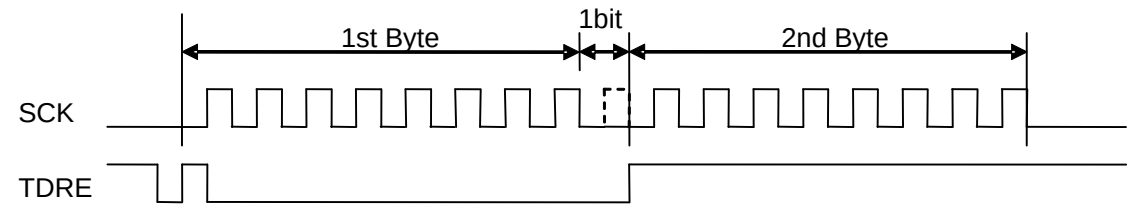
■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output. The SSR:TDRE bit is set to "1" before a half cycle of a rising edge of the first serial clock. If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

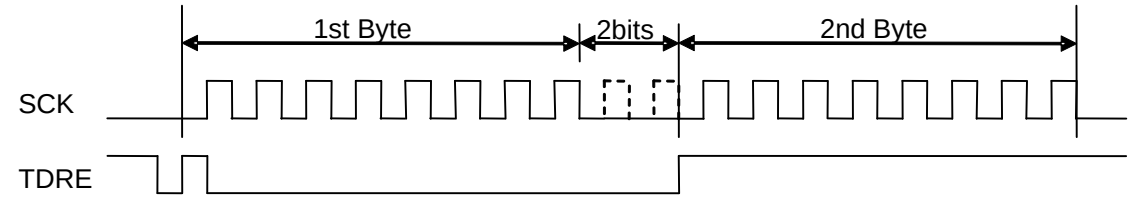
■ Continuous data transmit or reception waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

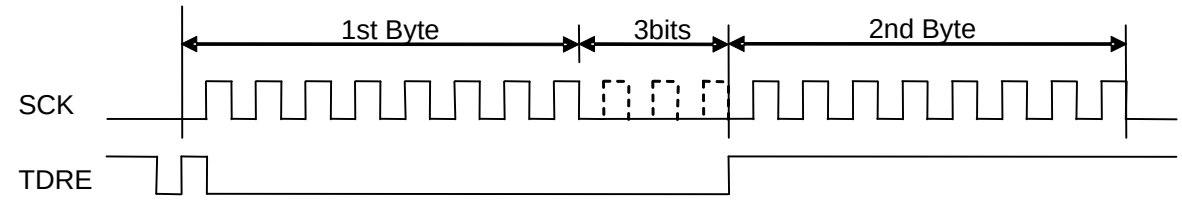
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- ESCR:WT1, ESCR:WT0=10 (in master mode operation)



- ESCR:WT1, ESCR:WT0=11 (in master mode operation)



Slave mode operation (SCR:MS=1, SMR:SCKE=0)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1) and data transmission is enabled (SCR:TXE=1) and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". This causes the first bit to output. Then, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input.
2. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

Note:

- *If data transmission is enabled (SCR:TXE=1) and if the first transmit data is written in the TDR at a time other than the serial clock (SCK) signal mark level, the first data bit is not output and the data transmission may fail. After the data transmission is enabled (SCR:TXE=1), the first transmit data must be written in the TDR at a signal mark level of the serial clock (SCK).*

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0) and data reception is enabled (SCR:RXE=1), the received data is sampled at a rising edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".

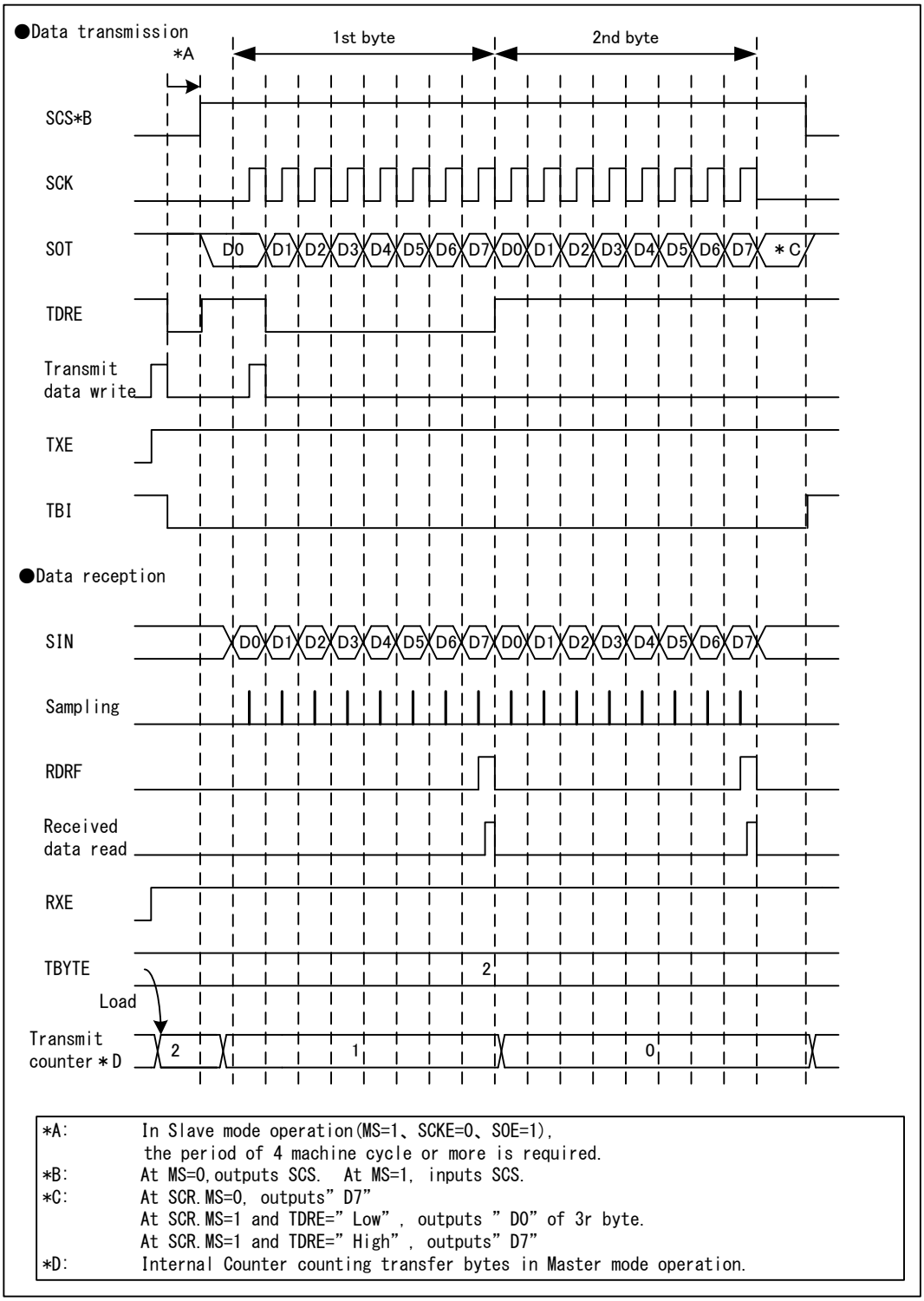
■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input. When the first bit of transmit data is output, the SSR:TDRE bit is set to "1". If a transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a rising edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".

■ Continuous switching from data reception to transmission

1. Disable the serial data output (SMR:SOE=0), enable a received interrupt (SCR:RIE=1), enable data reception (SCR:RXE=1), and enable data transmission (SCR:TXE=1). If dummy data is written in the TDR at a signal mark level of serial clock (SCK), the received data is sampled at a falling edge of serial clock (SCK) input.
2. To continue data reception, write a dummy data in the TDR between the time when a received interrupt is requested and when the next serial clock (SCK) rises.
3. To switch the data reception to the data transmission, enable the serial data output (SMR:SOE=1), disable a received interrupt (SCR:RIE=0), and disable data reception (SCR:RXE=0) between the time when a received interrupt is requested and when the next serial clock (SCK) rises. Also, output the transmit data in synchronization with a rising edge of serial clock after the transmit data has been written in the TDR and the data reception has completed.

SPI transfer (II) timing chart (Serial chip select pin is not used)



*A:	In Slave mode operation(MS=1、SCKE=0、SOE=1), the period of 4 machine cycle or more is required.
*B:	At MS=0,outputs SCS. At MS=1, inputs SCS.
*C:	At SCR.MS=0, outputs" D7"
	At SCR.MS=1 and TDRE=" Low", outputs " D0" of 3r byte.
	At SCR.MS=1 and TDRE=" High", outputs" D7"
*D:	Internal Counter counting transfer bytes in Master mode operation.

Master mode operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: “n” is the number of the serial chip select pin used.

Data transmission

1. If serial data output is enabled (SMR:SOE=1), data transmission is enabled (SCR:TXE=1) and data reception is disabled (SCR:RXE=0), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0". Then, the transmit data of the first bit is output and the Serial Chip Select pin (SCS) becomes active at the same time, and then, the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the Serial Clock output, this causes the transmit data to be output in synchronization with a falling edge of the serial clock (SCK) output.
   The SSR:TDRE bit is set to "1" before a half cycle of a falling edge of the first serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
2. After completing the times of the data transmission specified with TBYTE, the serial clock is stopped.
3. After the elapse of the hold time of the Serial Chip Select pin following the Serial Clock stop, the Serial Chip Select pin (SCS) becomes inactive. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

Data reception

1. If the serial data output is disabled (SMR:SOE=0), data transmission is enabled (SCR:TXE=1) and data reception is enabled (SCR:RXE=1), and when a dummy data is written in the TDR, the Serial Chip Select pin (SCS) becomes active and then, the serial clock output is started after the elapse of the setup time of the Serial Chip Select pin. After starting the serial clock output, the received data is sampled at a rising edge of serial clock (SCK) output.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1) during this time, a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception is completed for the time specified with TBYTE, the serial clock output is stopped.
5. After the serial clock output is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

Notes:

- *To perform data reception only, write a dummy data in the TDR so that the serial clock (SCK) is output.*
- *If the FIFO transmission and reception are enabled, the serial clocks (SCK) for the preset number of frames are output when the frames to be transferred are set in the FBYTE register.*

Data transmission and reception

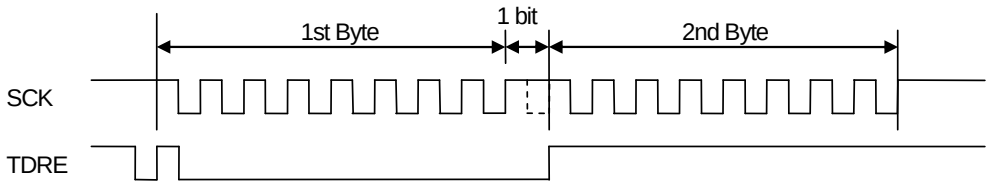
1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output and the Serial Chip Select pin (SCS) becomes active at the same time. The serial clock output is started after the elapse of setup time of the Serial Chip Select pin. After the serial clock output, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output. The SSR:TDRE bit is set to "1" before a half cycle of a rising edge of the first serial clock (SCK) output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.

3. The received data is sampled at a rising edge of the serial clock (SCK) output. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. After the data reception is completed for the time specified with TBYTE, the serial clock output is stopped.
5. After the serial clock output is stopped, the Serial Chip Select pin (SCS) becomes inactive after the elapse of the hold time of the Serial Chip Select pin. However, if the Serial Chip Select Active Level is held (SCSCR:SCAM=1), the Serial Chip Select pin (SCS) holds the active state.

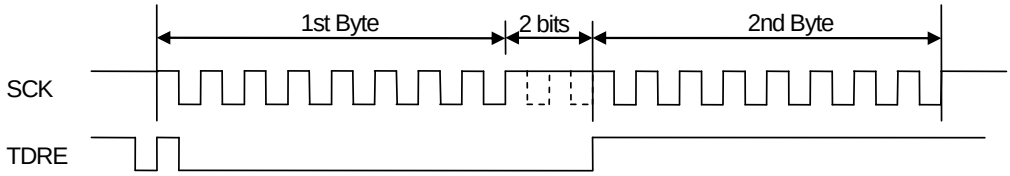
■ Continuous data transmit or reception waiting

If anything other than ESCR:WT1, ESCR:WT0=00 is set for the continuous data transmission or reception, a wait is inserted between frames.

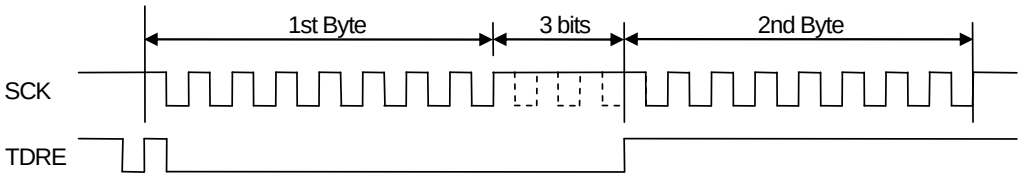
・ ESCR.WT1=0, ESCR.WT0=1(in master mode operation)



・ ESCR.WT1=1, ESCR.WT0=0(in master mode operation)



・ ESCR.WT1=1, ESCR.WT0=1(in master mode operation)



Slave mode operation (SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN=1, SCSCR:SCAM=0)

■ Data transmission

1. If serial data output is enabled (SMR:SOE=1), and data transmission is enabled (SCR:TXE=1), and when the transmit data is written in the TDR, the SSR:TDRE bit is set to "0".
2. When the Serial Chip Select pin (SCS) becomes active, the transmit data output is started and the first bit of the transmit data is output. After starting the data transmission, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) output.
3. When the transmit data of the first bit is output, the SSR:TDRE bit is set to "1". Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
4. If the Serial Chip Select pin (SCS) becomes inactive, the data transmission is stopped and the serial output pin (SOT) becomes “High”.

Note:

- *If data transmission is enabled (SCR:TXE=1) and if the first transmit data is written in the TDR at a time other than the serial clock (SCK) signal mark level, the first data bit is not output and the data transmission may fail. After the data transmission is enabled (SCR:TXE=1), the first transmit data must be written in the TDR at a signal mark level of the serial clock (SCK)*

■ Data reception

1. If the serial data output is disabled (SMR:SOE=0) , data reception is enabled (SCR:RXE=1), and the serial chip select pin (SCS) becomes active, the data reception is started and the received data is sampled at a falling edge of serial clock (SCK) input.
2. When the last bit is received, the SSR:RDRF bit is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is output.
   The received data (RDR) can be read during this time.
3. When the received data (RDR) is read, the SSR:RDRF bit is cleared to "0".
4. When the serial chip select pin (SCS) becomes inactive, the data reception is stopped.

■ Data transmission and reception

1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output and the Serial Chip Select pin (SCS) becomes active at the same time. After the starting data transmission and reception, the transmit data is output in synchronization with a falling edge of the serial clock (SCK) input. The SSR:TDRE bit is set to "1" after the first bit of transmit data is output. Therefore, if the transmit interrupt is enabled (SCR:TIE=1), a transmit interrupt request is output. During this time, the transmit data of the 2nd byte can be written in the register.
3. The received data is sampled at a rising edge of the serial clock (SCK) input. When the last bit of received data is received, the SSR:RDRF bit is set to "1". If the received interrupt is enabled (SCR:RIE=1), a received interrupt request is output. The received data (RDR) can be read during this time. When the received data is read, the SSR:RDRF bit is cleared to "0".
4. After the Serial Chip Select pin (SCS) becomes inactive, the data transmission and reception is stopped and the serial output pin (SOT) becomes “High”.

4. Serial Timer Operation

The serial timer is used for either timer function or synchronous transmission function.

Operations of serial timer

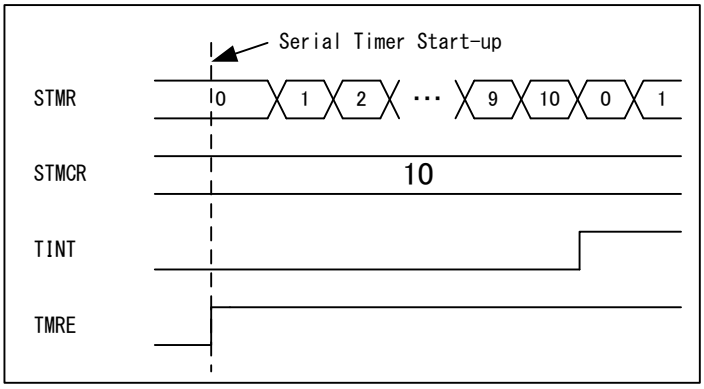
■ Starting method of serial timer

The serial timer is started by setting Serial Timer Enable bit (SACSR:TMRE) to “1”.

- Start-up with Serial Timer Enable bit (SACSR:TMRE)

  When Serial Timer Enable bit (SACSR:TMRE) is set to “1”, the serial timer is started and the serial timer register (STMR) counts from 0.

Figure 4-1 Start-up with Serial Timer Enable bit (STMCR=10, SACSR:TSYNE=0))



■ Stop method of serial timer

When the Serial Timer Enable bit (SACSR:TMRE) is set to “0”, the serial timer is stopped.
In this case, the value of the serial Timer Register (STMR) is held.

■ Timer operation

When the Synchronous Transmission Enable bit (SACSR:TSYNE) is 0, the serial timer functions as timer.

When the values of Serial Timer Register (STMR) and Serial Timer Comparison Register (STMCR) match, the Timer Interrupt Flag (SACSR:TINT) is set to “1” and the Serial Timer Register (STMR) is reset to “0”.

Figure 4-2 Timer operation (STMCR=10, SACSR:TSYNE=0)

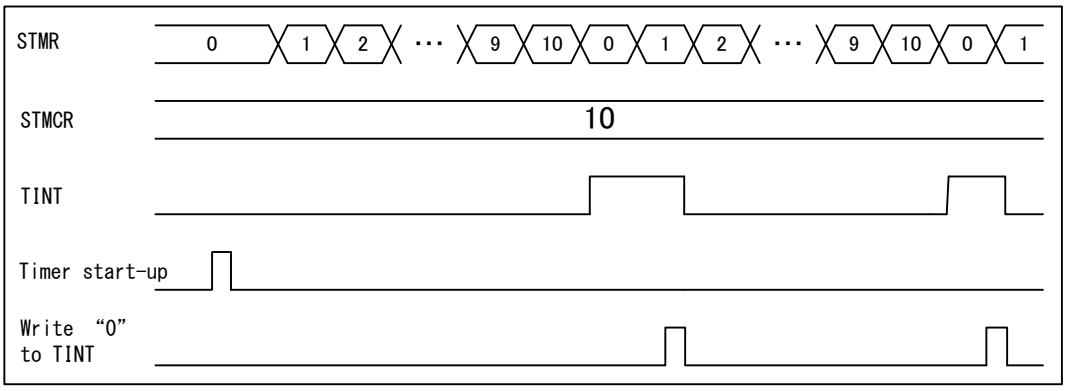


Figure 4-3 Serial Timer Initial Setting Flow Chart

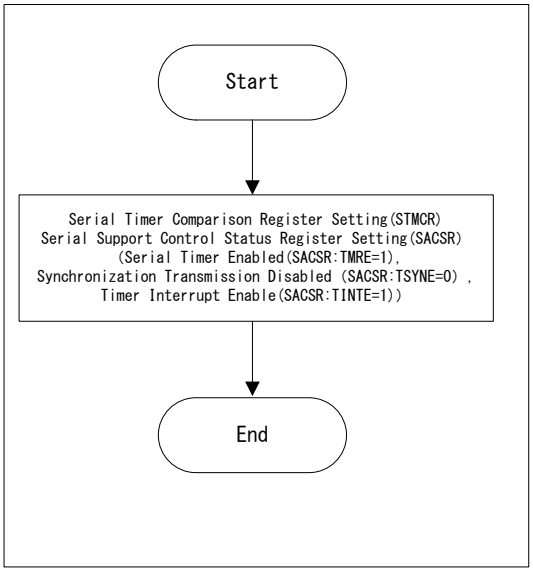
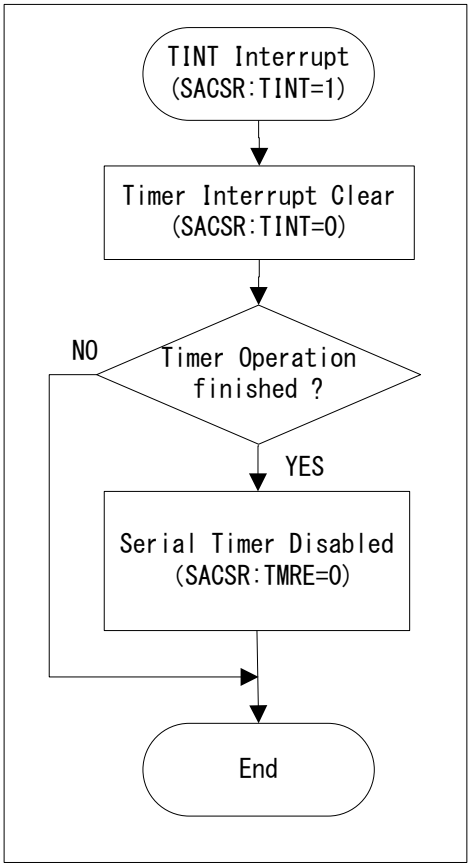


Figure 4-4 Serial Timer Interrupt Process Flow Chart



Note:

When the following conditions are met, the Timer Interrupt Flag (SACSR:TINT) is fixed to “1”.

- *The Timer Comparison Register (STMCR) is set to "0x0000" when Synchronous Transmission is disabled (SACSR:TSYNE=”0”)*
- *The division ratio of Timer Operation Clock (SACSR:TDIV) is set to "0000" during the timer operation.*

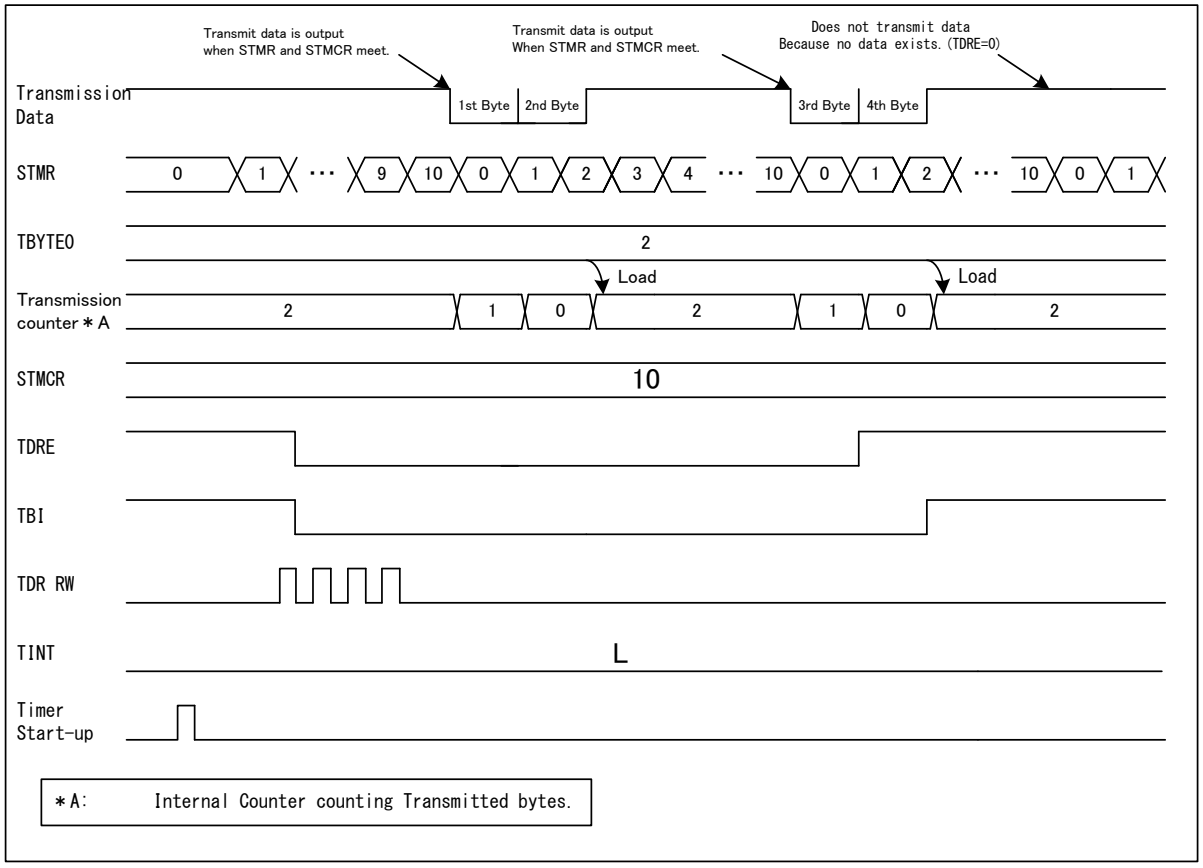
■ Transmission in synchronization with the timer

When the Synchronous Transmission Enable bit (SACSR:TSYNE) is “1”, the serial timer is used for synchronous transmission.

The transmission in synchronization with the timer is implemented as follows:

1. In the case where data exists in Transmission data register (SSR:TDRE=”0”), when the values of the Serial Timer Register (STMR) and Serial Timer Comparison Register (STMCR) match, the transmission is started and the Serial Timer Register is reset to”0”. The data of the count specified with TBYTE0 is transmitted.
2. After the data of the count specified with TBYTE0 has been transmitted, the transmission is stopped until the values of the Serial Timer Register (STMR) and Serial Timer Comparison Register (STMCR) match again.

Figure 4-5 Transmission in Synchronization with Timer (STMR=10, TBYTE0=2, SACSR:TSYNE=1)



In the case where the Synchronous Transmission is enabled (SACSR:TSYNE=1) and the Serial Timer Register (STMR) and the Serial Timer Comparison Register match, the transmission is not started in the following conditions:

- When transmission is disabled (SCR:TXE=0)
- In slave mode operation (SCR:MS=1)
- When no valid data exists in the transmission data register (SSR:TDRE=1)

However, when no valid data exists in the transmission data register (SSR:TDRE=1), if the synchronous transmission is enabled (SACSR:TSYNE="1") and the Serial Timer Register (STMR) and the Serial Timer Comparison Register match, the transmission is started immediately after writing transmission data to the transmission data register.

When a valid data exists in the Transmission Data Register (TDR) after the data of the count specified in TBYTE has been finished (SSR:TDRE=0), the transmission data is not transferred until the Serial Timer Register (STMR) and the Serial Timer Comparison Register match.

But, when the Serial Timer Register (STMR) and the Serial Timer Comparison Register match during transmitting (SSR:TBI=0) at Synchronous Transmission enabled (SACSR:TSYNE="1"), transmission is reserved. When the transmission is reserved, the transmission continues after the transmission of times specified in TBYTE0 has been finished.

The transmission reservation is released with one of the following conditions:

- Programmable reset (SCR:UPCL=1)
- Transmission is disabled (SCR:TXE=0)
- Data select error (SACSR:CSE=1)

To execute the synchronous reception, disable the Serial Data output (SMR:SOE=0), enable the Transmission (SCR:TXE=1) and reception (SCR:RXE=1), and write dummy data of the reception count to TDR.

Figure 4-6 Timer Synchronization Transmission Initial Setting Flowchart

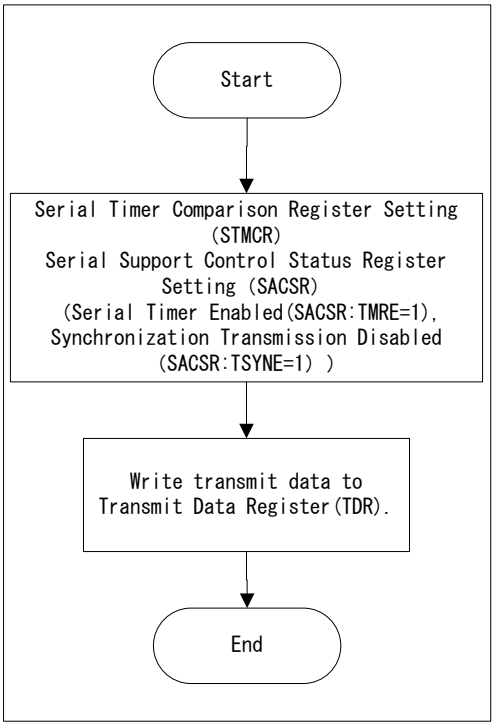
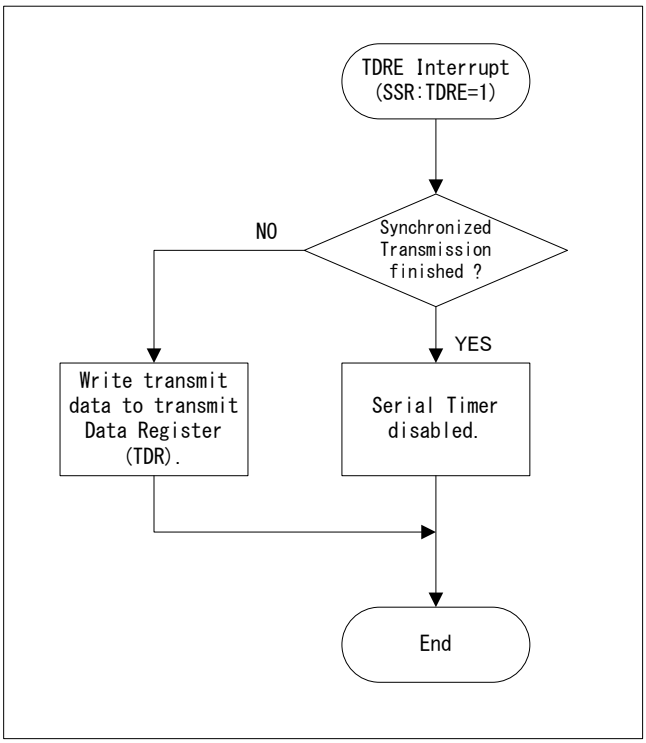


Figure 4-7 Timer Synchronization Transmission Interrupt Handling Flowchart



Note:

When no valid data exists in the Transmit Data Register (TDR) (SSR:TDRE=1) before transmitting the data frames of set value in TBYTE, execute the operations:

- *When the Transfer Byte Error is enabled (TBEEN=1), the Chip Select Error (SACSR:CSE=1) occurs. When the Chip Select Error Flag (SACSR:CSE) is set to "1", the transfer is not started even if the transmit data is written in the Transmit Data Register (TDR).*
- *When the Transfer Byte Error is disabled (TBEEN=0), transmission is stopped until the transmit data is written. If the transmit data is written, the transmission is restarted.*

5. Serial Chip Select Operation

This section shows the serial chip select operation.

■ Master mode operation (SCR:MS=0)

In master mode (SCR:MS=0), the Serial Chip Select pin operates as follows:

1. When the transmit data is written at serial chip select operation enabled (SCSCR:CSENn="1") and transmission enabled (SCR:TXE="1"), the Serial Chip Select pin becomes active.
2. After the elapse of setup time of the Serial Chip Select pin, the transmission and reception operation is started.
3. After the data transmit and reception of the times specified with TBYTE, the serial clock is stopped.
4. After the elapse of the hold time of the Serial Chip Select pin following the serial clock stop, the Serial Chip Select pin becomes active.

Figure 5-1 Serial Chip Select Operation (Master Transmission(MS=0), Normal Transfer(SPI=0), SCINV=0)

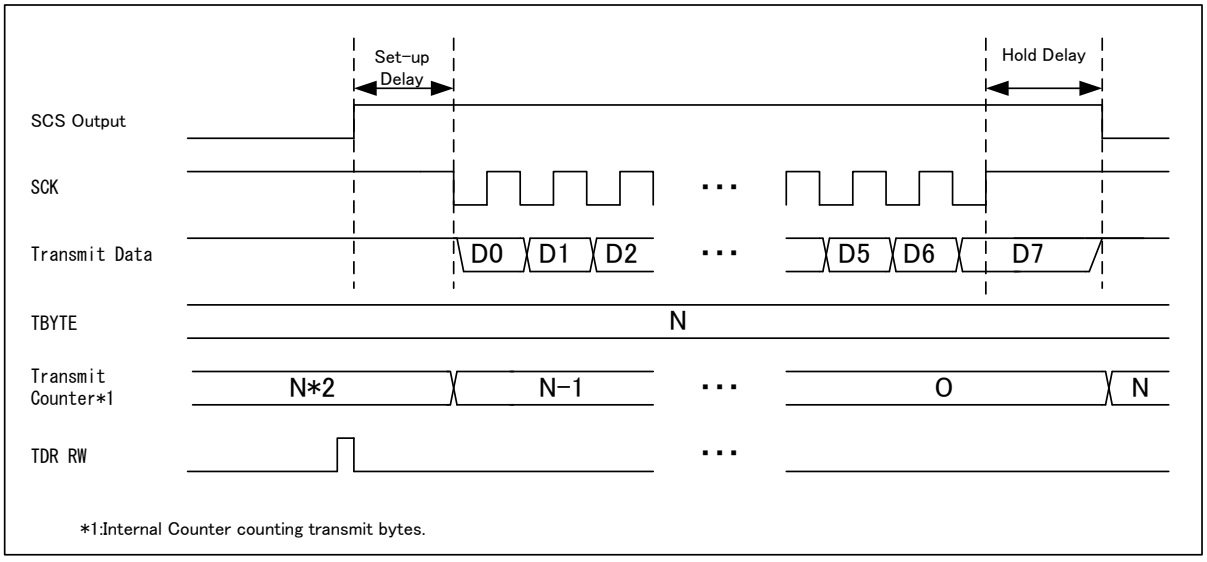
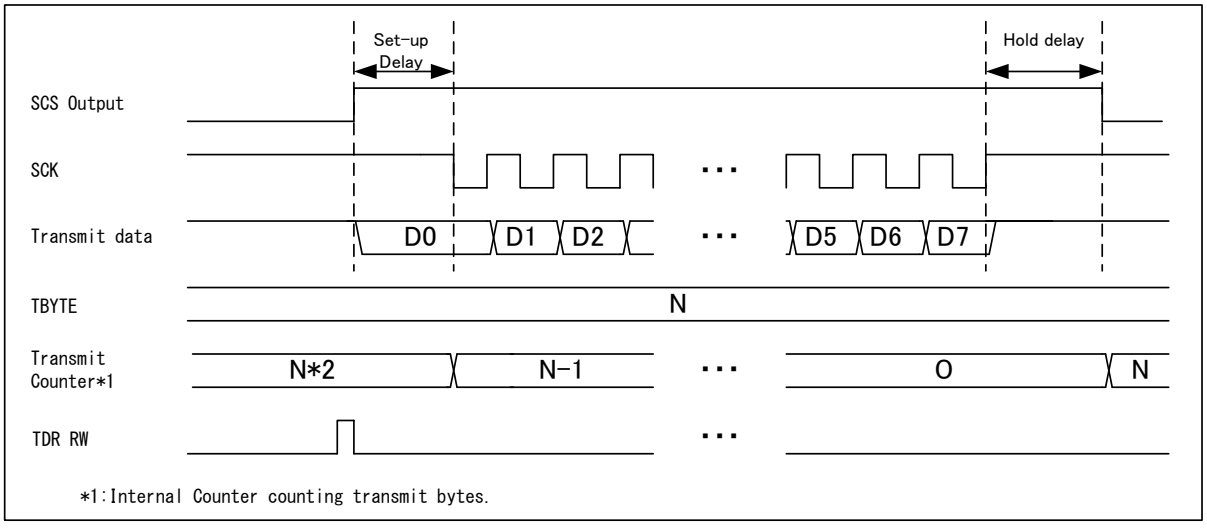


Figure 5-2 Serial Chip Select Operation (Master Transmission (MS=0), SPI Transfer (SPI=1), SCINV=0)



Notes:

- *If the transmission is disabled (SCR:TXE="1") and software reset is executed (SCR:UPCL=1) when the Serial Chip Select pin is active, the Serial Chip Select pin becomes inactive.*
- *When the Serial Chip Select pin does not hold “active state” (SCSCR:SCAM=0), the Serial Chip Select pin becomes inactive and the transmission bus becomes idle state (SSR:TBI=1) if the transmit data does not exist (SSR:TDRE=1) after the elapse of deselect time.*
- *When SCSCR:CSEN0 is set to “0” in the master mode operation (SCR:MS=0), the transmission and reception operation is executed irrespective of the Serial Chip Select pin state.*
- *When the frames of count less than the value specified with TBYTE have been transmitted, the following operations are executed if no valid transmit data exists in the Transmit Data Register (TDR) (SSR:TDRE=1), the following operations are executed:*
  - *The Chip Select Error occurs (SACSR:CSE=1) when the Transfer Byte Error is enabled (TBEEN=1). The Serial Chip Select pin becomes inactive after the elapse of the hold delay time following the Chip Select Error (SACSR:CSE=1). When the Chip Select Error Flag (SACSR:CSE) is set to”1”, the transmission operation is not executed even if the transmit data is written in the Transmit Data Register (TDR).*
  - *When the Transfer Byte Error is disabled (TBEEN=0), the transmission operation is stopped until transmit data is written in the Transmit Data Register (TDR). At this time, the Serial Chip Select pin is in active state. After the transmit data is written in the Transmit Data Register (TDR), the transmission operation is restarted.*

■ Serial Chip Select Timing Adjustment

When the Serial Chip Select Operation is enabled (SCSCR:CSENn=”1”) in Master mode operation (SCR:MS=0), setup delay, hold delay, and deselect time can be adjusted by changing the Serial Chip Select Timing Register (SCSTR3:0).

- Setup Delay Time

  This is the period from the time when the Serial Chip Select pin becomes active to the time when serial clock is output. For the details of setup delay time, see Figure 5-3 and Figure 5-4.
  This time is adjusted with Chip select setup delay bits (SCSTR0:CSSU7:0).

- Hold Delay Time

  This is the period from the time when the serial Clock output is finished to the time when the Serial Chip Select pin becomes inactive. For the details of hold delay time, see Figure 5-3 and Figure 5-4.
  This time is adjusted with Chip select hold delay bits (SCSTR1:CSHD7:0)

- Deselect time

  This is the minimum period from the time when the Serial Chip Select pin becomes inactive to the time when the Serial Chip Select pin becomes active again. Even if transmit data is written in the Transmit Data Register (TDR) during deselecting, the Serial Chip Select pin does not become active until the deselect time is finished. For details of deselect time, see Figure 5-3 and Figure 5-4.
  This time is adjusted with Chip select deselect bits (SCSTR3:2:CSDS15:0)

Figure 5-3 Timing Adjustment (Normal Transfer (SPI=0), SCINV=0)

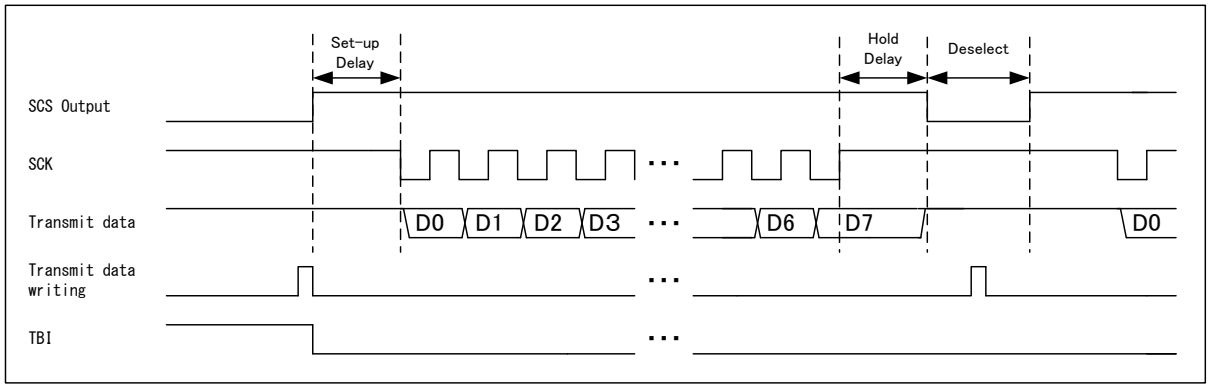
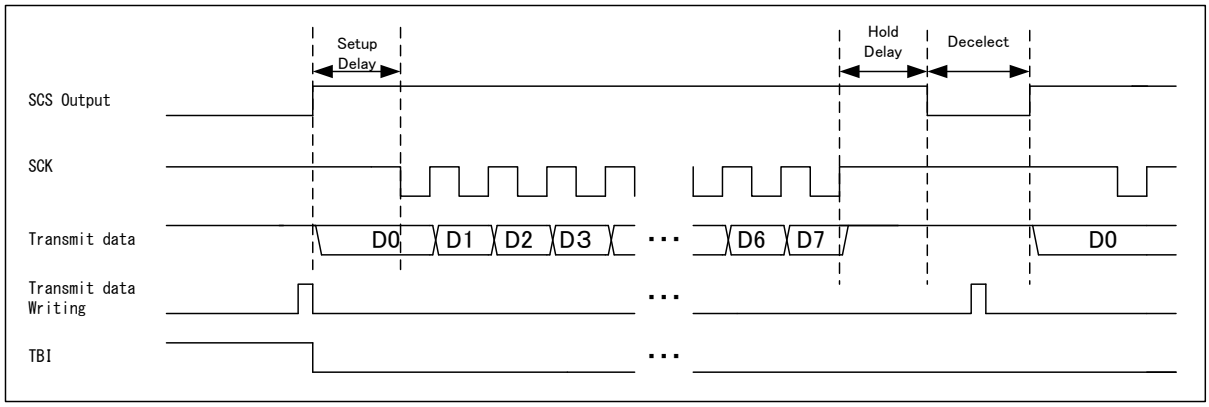


Figure 5-4 Timing Adjustment (SPI Transfer (SPI=1), SCINV=0)



Notes:

- *When no hold delay time exist (SCSTR1:CSHD7:0=0x00 in normal transfer(SCR:SPI=0), the Chip Select pin may become inactive before the sampling of the last bit. In such case, increase the values SCSTR1:CSHD7:0 to adjust the above timing.*
- *When no setup delay time exist (SCSTR0:CSSU7:0=0x00 in normal transfer(SCR:SPI=0), the Chip Select pin may become inactive before the sampling of the first bit. In such case, increase the values SCSTR0:CSSU7:0 to adjust the above timing.*

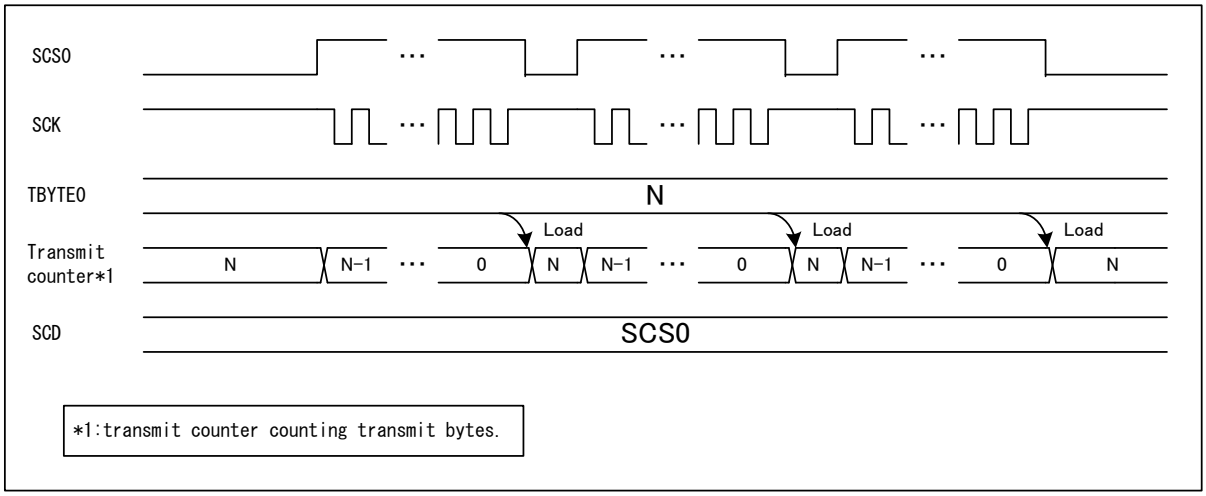
■ Chip Select Pin Independent Operation (Available only in Mater mode operation (SCR:MS=0))

For TYPE3-M4 product, when Serial Chip Select Start bit(SCSCR:SST1-0) is equivalent with Serial Chip select End bit(SCSCR:SED1-0), Chip Select at the only pin set by these bits operates.

When Serial Chip Select Active is not held (SCSCR:SCAM=0), the Serial Chip Select pin becomes inactive every time when the data transmission and reception of times specified withy TBYTE is executed.

For the operation of the Serial Chip Select pin when Serial Chip Select Active is held (SCSCR:SCAM=1), see “Serial Chip Select Active Held Operation”.

Figure 5-5 Chip Select Independent Operation (SST1-0=0, SED1-0=0, CSEN0=1, SCAM=0)



Note:

- *At the independent operation, the timing adjustment (for setup time, hold time, and deselect time) of the Serial Chip Select pin is available.*

■ Chip Select Pin Round-Robin Operation(Available only Master mode operation(SCR:MS=0))

TYPE3-M4 product can use this function. TYPE1-M4 and TYPE2-M4 products can't use this function.

When Serial Chip Select Start bit (SCSCR:SST1-0) is not equivalent with Serial Chip Select End bit(SCSCR:SED1-0), some Chip Select pins become active by rotation.

1. When writing transmission data to TDR during Serial Chip Select output enabled(SCSCR:CSOE=”1”) and transmission enabled(SCR:TXE=”1”), Serial Chip Select becomes active from the pin set by Serial Chip Select Start bit(SCSCR:SST1-0)
2. When Serial Chip Select Active Hold bit is not enabled (SCSCR:SCAM=0), Serial Chip Select pin becomes inactive after transmitting/receiving data by setting the number times at TBYTE. Then, next number Serial Chip Select pin becomes active. *1
   However, when next number Serial Chip Select pin is disabled (SCSCR:CSENn=0), that Serial Chip Select pin is skipped.
3. When the number of active Chip Serial Select pin is equivalent with the number of Serial Chip Select pin set by Serial Chip Select End bit, Serial Chip Select Pin set by Serial Chip Select start bit becomes active.

*1 : After SCS0 becomes active, SCS1 becomes active. After SCS3 becomes active, SCS0 becomes active.

When Serial Chip Select Active Hold bit is enabled (SCSCR:SCAM=”1”), see “Serial Chip Select Active Held Operation” for that operation

Figure 5-6 is the timing chart explaining the operation when Serial Chip Select Start pin is SCS0(SST1-0=0) and Serial Chip Select End pin is SCS3(SED1-0=3).

Figure 5-6 Chip Select Pin Round-Robin Operation(SST1-0=0, SED1-0=3, CSEN3=1, CSEN2=1, CSEN1=1, CSEN0=1, SCAM=0)

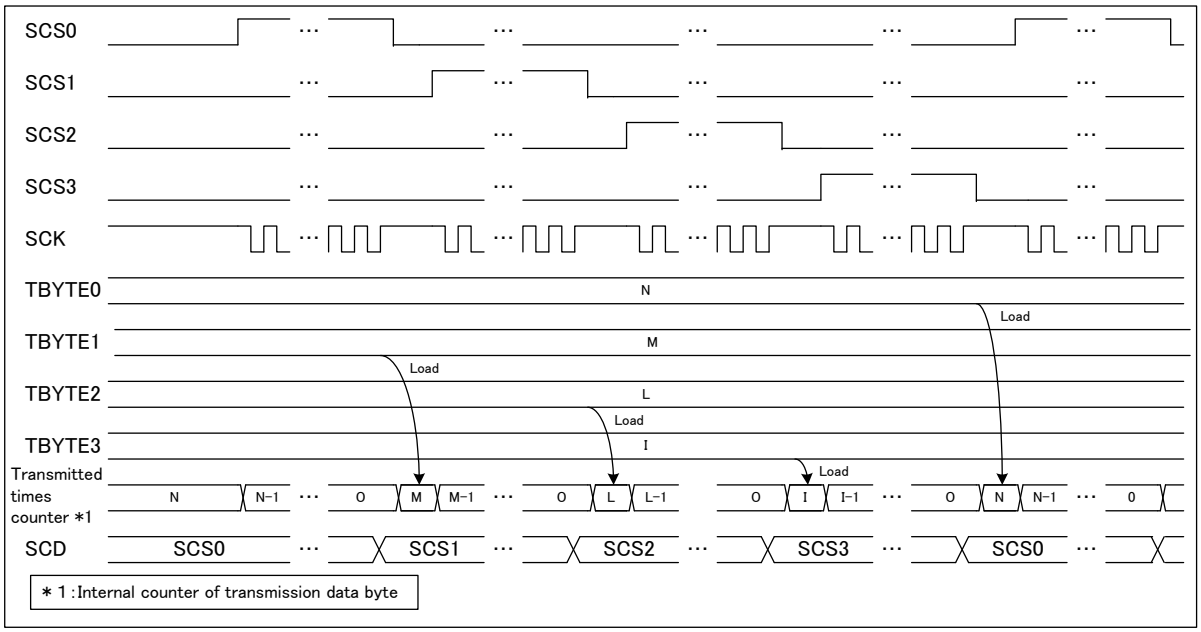


Figure 5-7 is the timing chart explaining the operation when Serial Chip Select Start pin is SCS1(SST1-0=1) and Serial Chip Select End pin is SCS2(SED1-0=2).

Figure 5-7 Chip Select Pin Round-Robin Operation (SST1-0=1, SED1-0=2, CSEN3=0, CSEN2=1, CSEN1=1, CSEN0=0, SCAM=0)

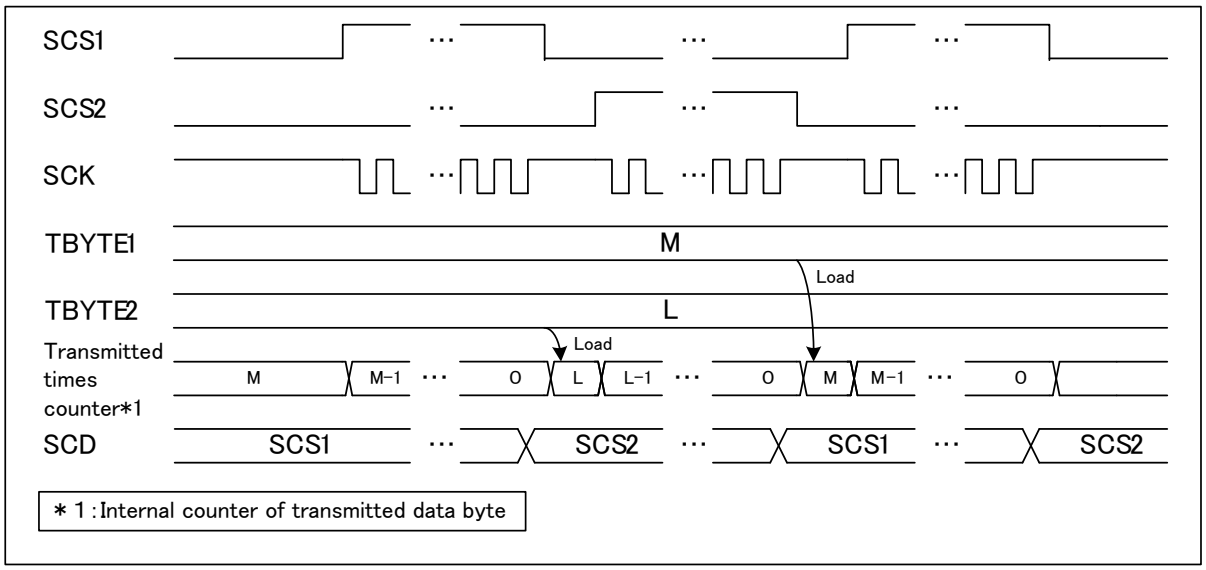
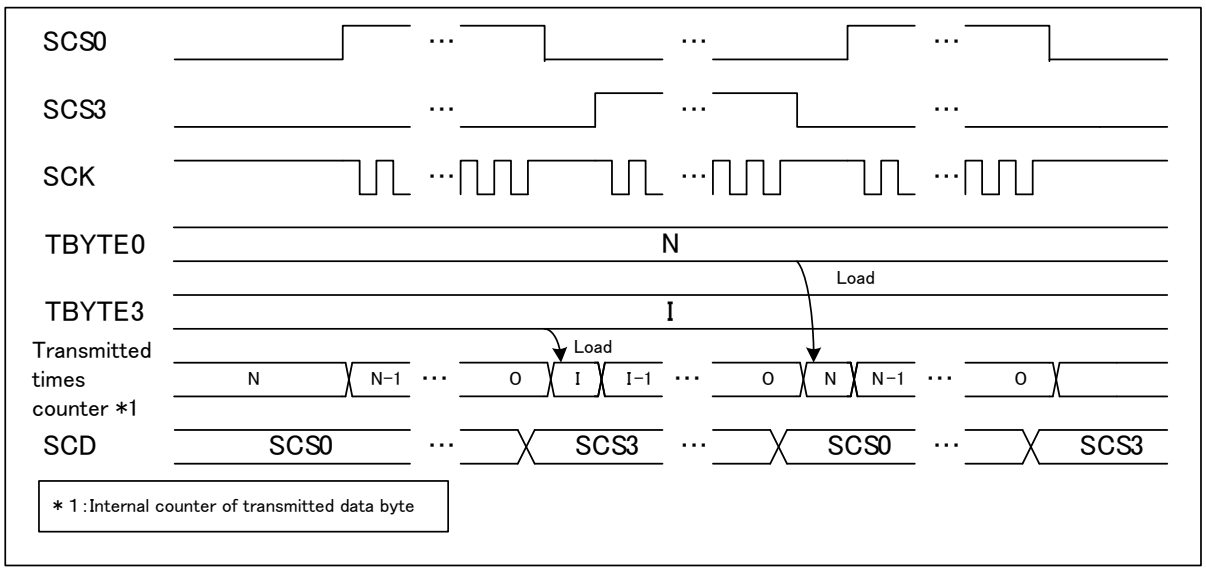


Figure 5-8 is the timing chart explaining the operation when Serial Chip Select Start pin is SCS0(SST1-0=0) and Serial Chip Select End pin is SCS3(SED1-0=3) and SCS1 and SCS3 are disable(CSEN1-2="00").

Figure 5-8 Chip Select Pin Round-Robin Operation (SST1-0=1, SED1-0=3, CSEN3=1, CSEN2=0, CSEN1=0, CSEN0=0, SCAM=0)



Notes:

- *At any following condition, Serial Chip Select Pin becomes active set by Serial Chip Select Start bit(SCSCR:SST1-0)*
  - *When changing transmission operation from disabled to enabled.*
  - *When software reset is executed (SCR:UPCL="1").*
- *During Round-Robin operation, Serial Chip Select Pin Timing Adjustment (Set up timing, Hold timing, Deselect timing) is enabled.*

■ Serial Chip Select Active Held Operation (SCSCR:SCAM=1) (Available only in master mode operation (SCR:MS=0))

When the transmission is started with setting the Serial Chip Select Active Holding bit (SCSCR:SCAM) to “1”, the Serial Chip Select pin is held in “Active State”.

Table 5-1 Serial Chip Select Active Holding bit (SCSCR:SCAM)

Present State	Present SCSCR: SCAM bit	Present SSR: TDRE bit	Next State
Transmitting (Transmit count < TBYTE)	0	-	The Serial Chip Select pin is held in “Active state” until the frames of count specified with TBYTE are transmitted.
	1		
the transmission of frames of count specified with TBYTE are finished.	0	0	After the hold delay time, sets the Serial Chip Select pin to “inactive”. After the elapse of deselect time, the next transmission is started.
		1	After the hold delay time, sets the Serial Chip Select pin to “inactive”. After the elapse of deselect time, the transmission is stopped until the next transmit data is written.
	1	1	Holds the Serial Chip Select to be “active”.
		0	In active state of Serial Chip Select pin, the transmission continues. The Serial Chip Select pin holds to be active until the frames of count specified with TBYTE again.
Chip Select Error occurs (SACSR:CSE=1)	-	-	Irrespective of SCAM setting, the Serial Chip Select is set to be inactive after the hold delay time is elapsed.
Software reset is executed (SCR:UPCL=1)	-	-	Irrespective of SCAM setting, the Serial Chip Select is set to be inactive immediately.
Transmission disabled (SCR:TXE=0)			

Note:

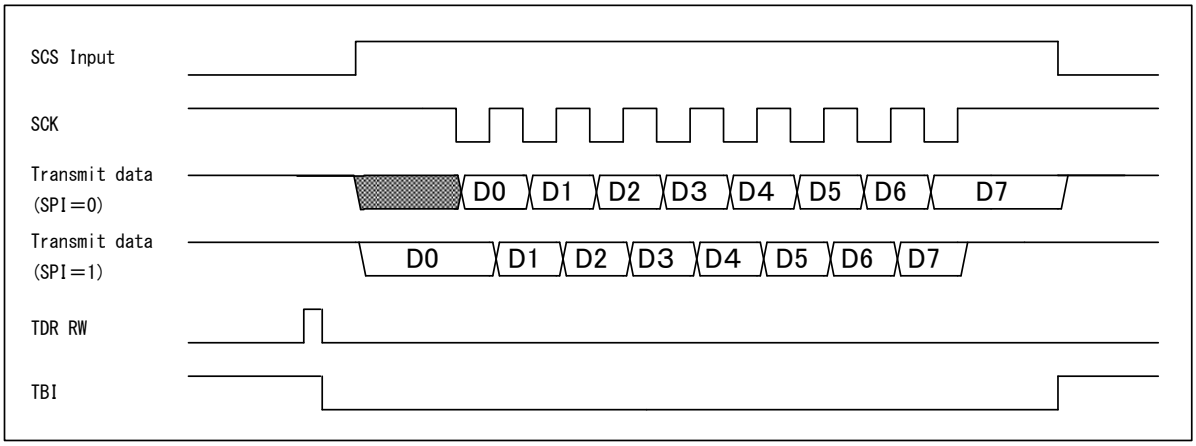
When all the following conditions are met, the Serial Chip Select pin is not held, and the Serial Chip Select pin becomes inactive after the elapse of the hold delay time, and Chip Select Error occurs (SACSR:CSE=1).

- *Transfer byte error is enabled (SACSR:TBEEN=1).*
- *The data transmission and reception of counts specified with TBYTE is not finished.*
- *The transmit data register (TDR) is empty (SSR:TDRE=1).*

■ Slave Mode Operation (SCR:MS=1)

When the Serial Chip Select pin0 (SCS0) is enabled (SCSCR:CSEN0="1") and the input of the Serial Chip Select pin becomes active, the transmission or reception operation is executed in synchronization of serial clock (SCK). Then, when the input of Serial Chip Select pin becomes inactive, the transmission or reception operation is finished.

Figure 5-9 Serial Chip Select Operation in Slave Mode Operation (Slave Transmission, SCINV=0)



Notes:

- *While the Serial Chip Select pin input is in “inactive state”, the operation is not started even if the serial clock is input.*
- *During reception operation, the Serial Chip Select input becomes inactive state before the last bit is sampled, the data received is deleted.*
- *During transmission operation, the Serial Chip Select input becomes inactive state, the data transmitted is deleted and chip select error (SACSR:CSE) occurs.*
- *When TDR is empty (SSR:TDRE=1) and the Serial Chip Select input becomes inactive state, transmit bus idle state occurs(SSR:TBI=1).*
- *In Slave Mode Operation (SCR:MS=1), when SCSCR:CSEN0 is set to “0”, the data transmission and reception is executed irrespective of the Serial Chip Select pin state.*

■ Format setting of Serial Chip Select Pin

TYPE1-M4 and TYPE2-M4 products have Serial Chip Select pin 0.
TYPE3-M4 product has Serial Chip Select pin 0-3.

Note:

- *For details, see "Data Sheet" of the product used.*

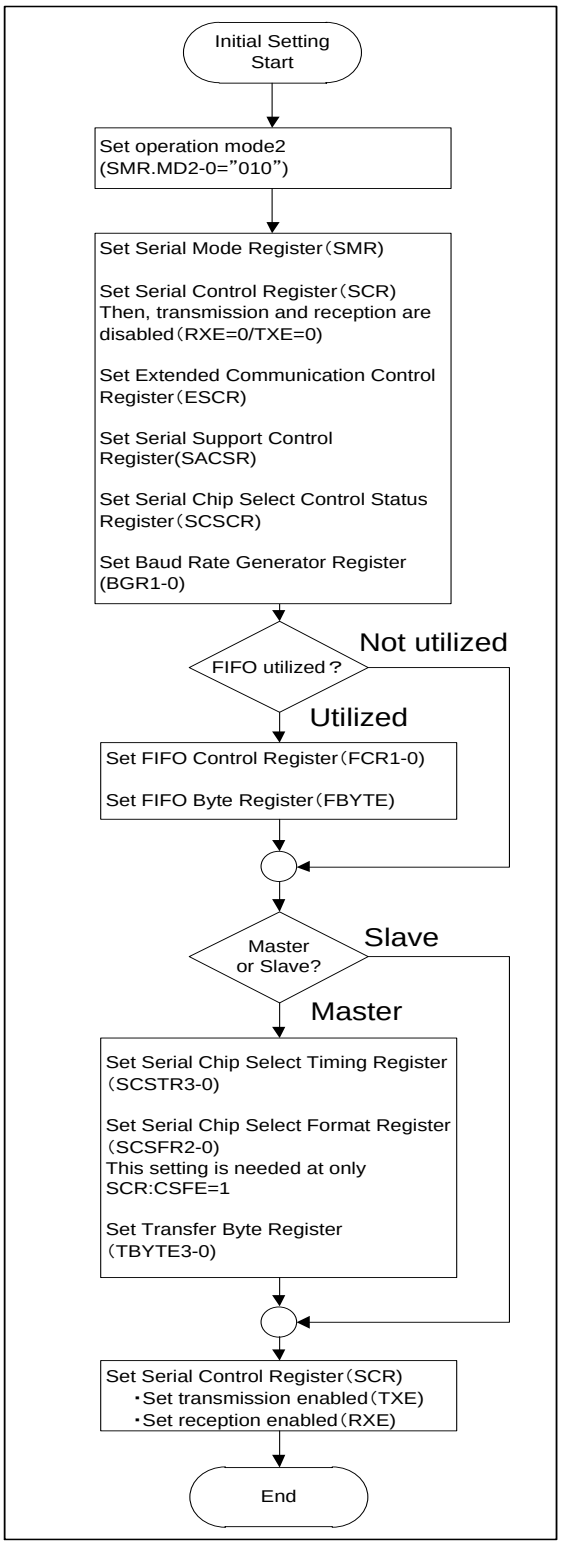
These can be set to each Chip Select Pin by the bits showed at Table 5-2, are chip select active level, clock inversion, SPI mode enabled/disabled, serial data direction and data length.

Table 5-2 Format setting of Serial Chip Select Pin

Conditions		Active level of chip select	Clock Inversion	SPI mode	Data direction	Data length
Chip select format enabled (SCR: CSFE=1) and Master mode (SCR:MS=0)	SCS0 output	SCSCR0:SCLVL	SMR:SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
	SCS1 output	SCSFR0: CS1SCLVL	SCSFR0: CS1SCINV	SCSFR0: CS1SPI	SCSFR0: CS1BDS	SCSFR0: CS1L3-0
	SCS2 output	SCSFR1: CS2SCLVL	SCSFR1: CS2SCINV	SCSFR1: CS2SPI	SCSFR1: CS2BDS	SCSFR1: CS2L3-0
	SCS3 output	SCSFR2: CS3SCLVL	SCSFR2: CS3SCINV	SCSFR2: CS3SPI	SCSFR2: CS3BDS	SCSFR2: CS3L3-0
Chip select format disabled (SCR:CSFE=0)		SCSCR0:SCLVL	SMR:SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
Slave mode (SCR:MS=1)						
Chip select disabled (CSEN3-0="0000")						

■ Initial Setting Flowchart

Figure 5-10 Initial Setting Flowchart of Chip Select



6. Dedicated baud rate generator

The dedicated baud rate generator functions in the master mode operation only. However, if received FIFO is used, set the dedicated baud rate generator in the slave mode operation, too.

CSIO (Clock Synchronous Serial Interface) baud rate selection

The dedicated baud rate generator settings vary depending on the master or slave mode operation.

[1] During master mode operation

- Divide the internal clock frequency using the dedicated baud rate generator, and select a baud rate.
  - This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).
  - The internal clock frequency is divided by the reload counter set value.

[2] During slave mode operation

The dedicated baud rate generator does not function in the slave mode operation (SCR:MS=1).
(An external clock, entered from the SCK clock input pin, is used directly.)

Note:

- *If received FIFO is used, set the dedicated baud rate generator even in the slave mode operation.*

6.1 Baud rate settings

This section explains how to set the baud rate. Also, the calculation result of serial clock frequency is shown.

Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$V = \phi / b - 1$

V : Reload value; b : Baud rate; ϕ: Bus clock frequency

(2) Calculation example

To set the 16 MHz bus clock, use the internal clock, and set the 19200 bps baud rate, set the reload value as follows:

Reload value:

$V = (16 \times 1000000) / 19200 - 1 = 832$

Therefore, the baud rate is:

$b = (16 \times 1000000) / (832 + 1) = 19208$ bps

(3) Baud rate error

The baud rate error can be calculated by the following equation.

Error (%) = (Calculated value – Target value) / Target value 100

Example: To set the 20 MHz bus clock and 153600 bps target baud rate:

Reload value = $(20 \times 1000000) / 1536009 - 1=129$

Buad rate (Calculated value) = $(20 \times 1000000) / (129 + 1) = 153846$ (bps)

Error (%) = $(153846 - 153600) / 153600 \times 100= 0.16$ (%):

Notes:

- *If the reload value is set to "0", the reload counter is stopped.*
- *If the reload value is even, the "HIGH" and "LOW" width of serial clock changes as follows, depending on SMR:SCIN bit and SCR:SPI bit settings. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.*
- *When in normal transfer (SCR:SPI=0) and the mark level of the serial clock is "HIGH" (SMR:SCINV=0), or when in SPI transfer (SCR:SPI=1) and the mark level of the serial clock is "LOW" (SMR:SCINV=1), the "HIGH" width of serial clock is longer for 1 cycle of bus clock.*
- *When in normal transfer (SCR:SPI=0) and the mark level of the serial clock is "LOW" (SMR:SCINV=1), or when in SPI transfer (SCR:SPI=1) and the mark level of the serial clock is "HIGH" (SMR:SCINV=0), the "LOW" width of serial clock is longer for 1 cycle of bus clock.*
- *Set the reload value to 3 or more.*

Reload values and baud rates for each bus clock frequency

Table 6-1 Reload values and baud rates

Baud rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	7	0	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	-	-
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.06	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	0.06	554	-0.01	693	0.06	832	0.03	1110	0.01
19200	416	-0.08	520	-0.03	832	-0.03	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	-0.03	1666	-0.02	208	0.01	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8332	<0.01	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: BGR1/0 register set value
- ERR: Baud rate error (%)

Table 6-2 Reload values and baud rates (Continued)

Baud rate (bps)	40 MHz		48 MHz		72 MHz		80 MHz		100 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	4	0	5	0	8	0	9	0	-	-
6M	-	-	7	0	11	0	-	-	-	-
5M	7	0	-	-	-	-	15	0	19	0
4M	9	0	11	0	17	0	19	0	24	0
2.5M	15	0	-	-	-	-	31	0	39	0
2M	19	0	23	0	35	0	39	0	49	0
1M	39	0	47	0	71	0	79	0	99	0
500000	79	0	95	0	143	0	159	0	199	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22	216	<0.01
250000	159	0	191	0	287	0	319	0	399	0
230400	173	-0.22	207	0.16	312	-0.16	346	0.06	433	<0.01
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03	650	<0.01
125000	319	0	383	0	575	0	639	0	799	0
115200	346	0.06	416	-0.08	624	0	693	0.06	867	<0.01
76800	520	-0.03	624	0	937	-0.05	1041	-0.03	1301	<0.01
57600	693	0.06	832	0.04	1249	0	1388	<0.01	1735	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01	2603	<0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01	3471	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01	5207	<0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	0	9599	<0.01
9600	4166	<0.01	4999	0	7499	0	8332	0	10416	0
7200	5555	<0.01	6666	<0.01	9999	0	11110	0	13888	0
4800	8332	<0.01	9999	0	14999	0	16666	0	20832	0
2400	16666	<0.01	19999	0	29999	0	-	-	-	-
1200	-	-	-	-	-	-	-	-	-	-
600	-	-	-	-	-	-	-	-	-	-
300	-	-	-	-	-	-	-	-	-	-

- Value: BGR1/0 register set value
- ERR: Baud rate error (%)

For frequencies not described in Table 6-1 and Table 6-2, calculate them conforming to the formula in “6.1 Baud rare settings”. (However, for the maximum frequency, see “Data Sheet” of the product used because it is varied by products.)

Functions of reload counter

There are two types of reload counter: the transmit reload counter and the received reload counter. They function as the dedicated baud rate generators. Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from internal clocks.

Starting counting

When the reload value is written to the Baud Rate Generator Register (BGR1 or BGR0), the reload counter starts counting.

Restarting

The reload counter restarts counting in the following conditions.

■ Common to transmit and received reload counters

A programmable reset (SCR:UPCL bit)

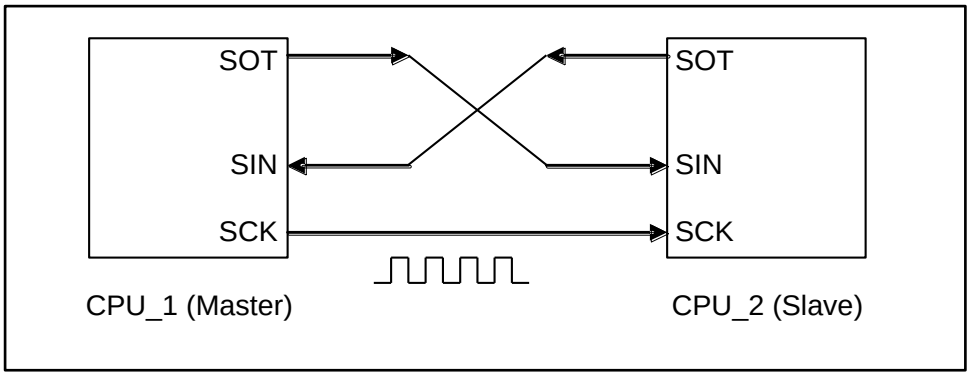
6.2 CSIO (Clock Synchronous Serial Interface) setup procedure and program flow

The CSIO (Clock Synchronous Serial Interface) allows bidirectional and synchronous serial data transmission.

■ CPU-to-CPU connection

Select the bidirectional communication for the CSIO (Clock Synchronous Serial Interface). Connect two CPUs to each other as shown in Figure 6-1.

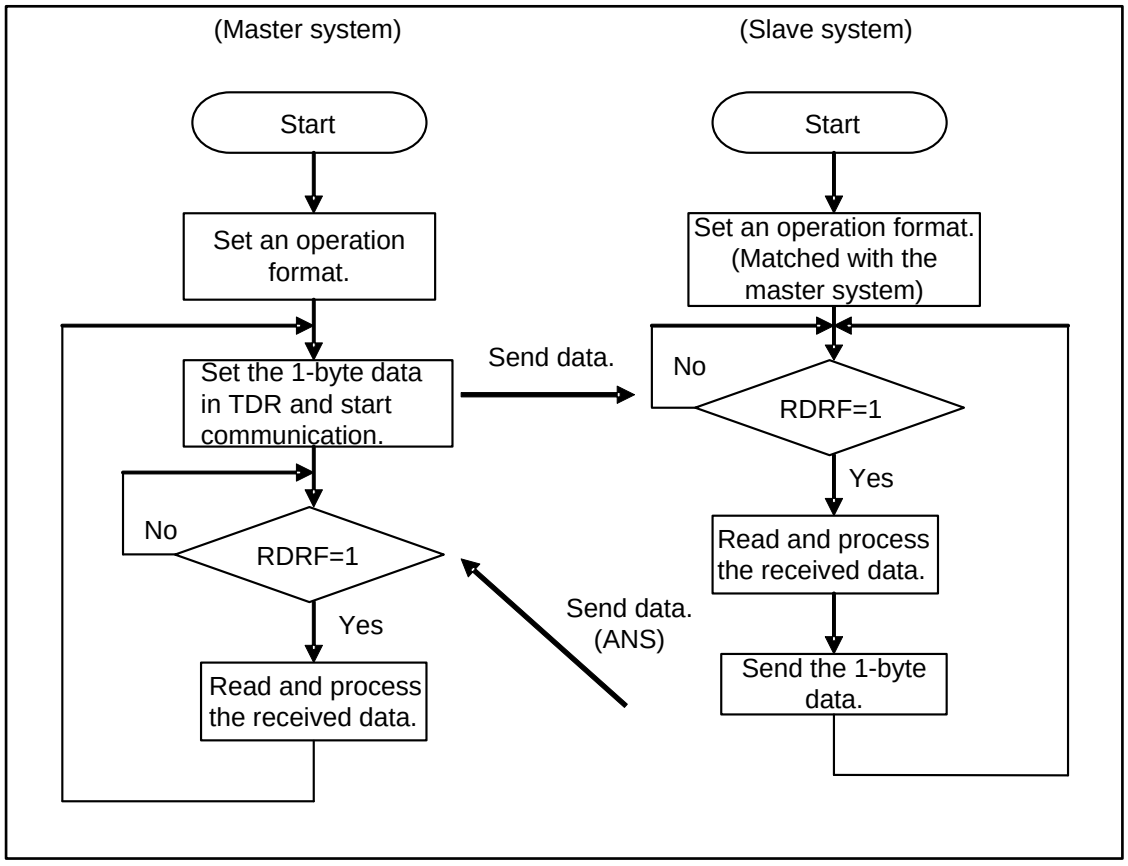
Figure 6-1 Connection example for CSIO (Clock Synchronous Serial Interface) bidirectional communication



Flowcharts

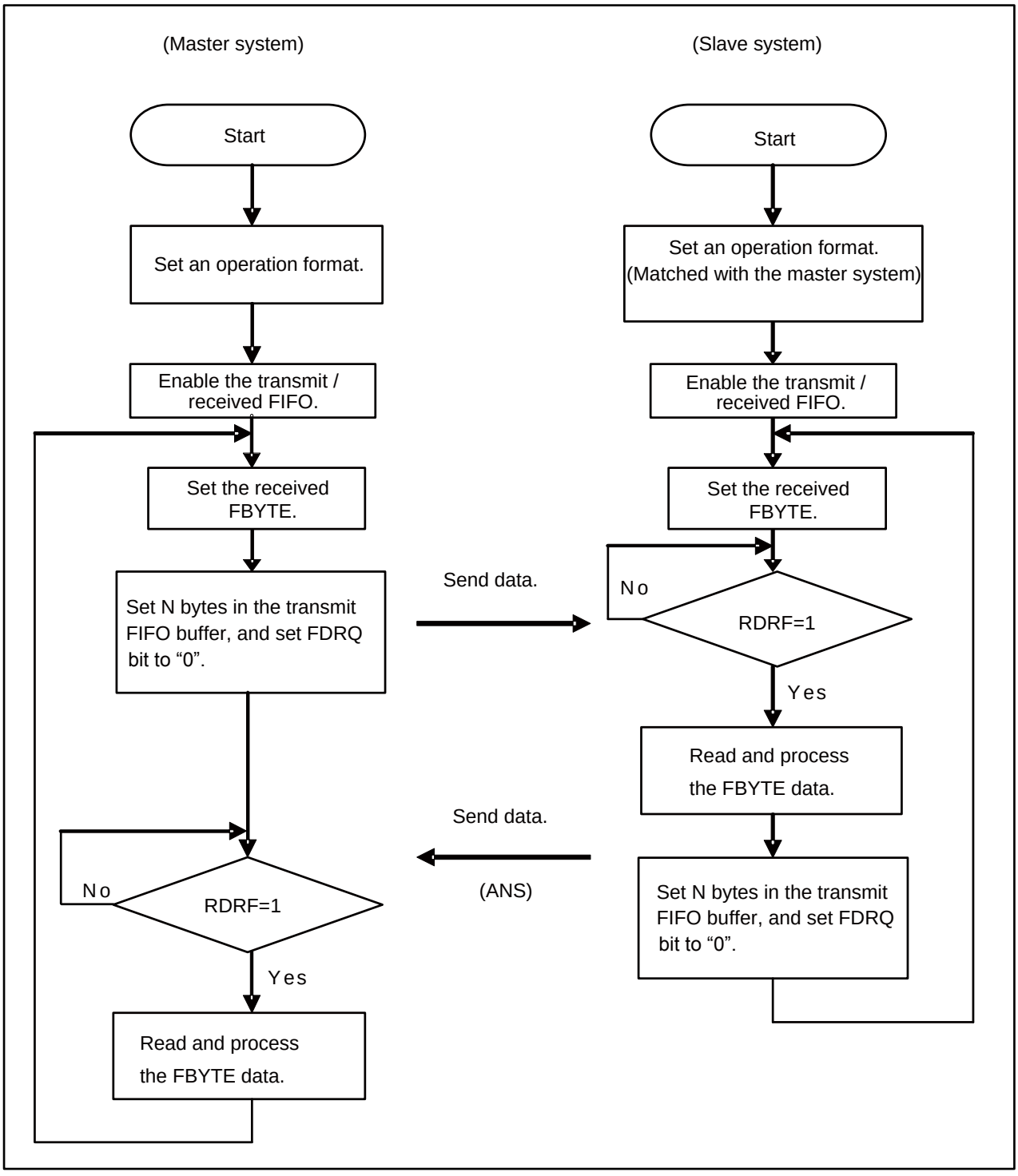
■ If FIFO is not used

Figure 6-2 Example of bidirectional communication flowchart (if FIFO is not used)



■ If FIFO is used

Figure 6-3 Example of bidirectional communication flowchart (if FIFO is used)



7. CSIO (Clock Synchronous Serial Interface) registers

This section provides a list of CSIO (Clock Synchronous Serial Interface) registers.

CSIO (Clock Synchronous Serial Interface) register list

Table 7-1 CSIO (Clock Synchronous Serial Interface) register list (TYPE1-M4, TYPE2-M4 product)

	bit15 bit8	bit7 bit0
CSIO	SCR (Serial Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	ESCR (Extended Communication Control Register)
	RDR1/TDR1 (Transmit/Received Data register 1)	RDR0/TDR0 (Transmit/Received Data register 0)
	SACSR (Serial Support Control Status Register)	
	STMR (Serial Timer Register)	
	STMCR (Serial Timer Comparison Register)	
	SCSCR (Serial Chip Select Control Status Register)	
	SCSTR1 (Serial Chip Select Timing Register1)	SCSTR0 (Serial Chip Select Timing Register0)
	SCSTR3 (Serial Chip Select Timing Register3)	SCSTR2 (Serial Chip Select Timing Register2)
	-	TBYTE0(Transfer Byte Register0)
	BGR1 (Baud Rate Generator Register 1)	BGR0 (Baud Rate Generator Register 0)
FIFO	FCR1 (FIFO Control Register 1)	FCR0 (FIFO Control Register 0)
	FBYTE2 (FIFO2 Byte Register)	FBYTE1 (FIFO1 Byte Register)

Table 7-2 CSIO (Clock Synchronous Serial Interface) register list (TYPE3-M4 product)

	bit15 bit8	bit7 bit0
CSIO	SCR (Serial Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	ESCR (Extended Communication Control Register)
	RDR1/TDR1 (Transmit/Received Data register 1)	RDR0/TDR0 (Transmit/Received Data register 0)
	SACSR (Serial Support Control Status Register)	
	STMR (Serial Timer Register)	
	STMCR (Serial Timer Comparison Register)	
	SCSCR (Serial Chip Select Control Status Register)	
	SCSTR1 (Serial Chip Select Timing Register1)	SCSTR0 (Serial Chip Select Timing Register0)
	SCSTR3 (Serial Chip Select Timing Register3)	SCSTR2 (Serial Chip Select Timing Register2)
	SCSFR1 (Serial Chip Select Format Register1)	SCSFR0 (Serial Chip Select Format Register0)
	-	SCSFR2 (Serial Chip Select Format Register2)
	TBYTE1(Transfer Byte Register1)	TBYTE0(Transfer Byte Register0)
	TBYTE3(Transfer Byte Register3)	TBYTE2(Transfer Byte Register2)
	BGR1 (Baud Rate Generator Register 1)	BGR0 (Baud Rate Generator Register 0)
FIFO	FCR1 (FIFO Control Register 1)	FCR0 (FIFO Control Register 0)
	FBYTE2 (FIFO2 Byte Register)	FBYTE1 (FIFO1 Byte Register)

Table 7-3 CSIO (Clock Synchronous Serial Interface) bit assignment (TYPE1-M4, TYPE2-M4 product)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	I3	-	WT1	WT0	L2	L1	L0
TDR1/0 (RDR1/0)	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
TDR3/2 (RDR3/2)	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
SACSR	-	-	TBEEN	CSEIE	CSE	-	-	TINT	TINTE	TSYNE	-	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
STMR	TM15	TM4	TM3	TM2	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
STMCR	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
SCSCR	-	-	-	-	-	-	SCAM	CDIV2	CDIV1	CDIV0	CSLVL	-	-	-	CSEN0	CSOE
SCSTR 1/0	CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0
SCSTR 3/2	CSDS 15	CSDS 14	CSDS 13	CSDS 12	CSDS 11	CSDS 10	CSDS9	CSDS8	CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0
TBYTE0	-	-	-	-	-	-	-	-	CS0 TD7	CS0 TD6	CS0 TD5	CS0 TD4	CS0 TD3	CS0 TD2	CS0 TD1	CS0 TD0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

Table 7-4 CSIO (Clock Synchronous Serial Interface) bit assignment (TYPE3-M4 product)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	I3	CSFE	WT1	WT0	L2	L1	L0
TDR1/0 (RDR1/0)	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
TDR3/2 (RDR3/2)	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
SACSR	-	-	TBEEN	CSEIE	CSE	-	-	TINT	TINTE	TSYNE	-	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
STMR	TM15	TM4	TM3	TM2	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
STMCR	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
SCSCR	SST1	SST0	SED1	SED0	SCD1	SCD0	SCAM	CDIV2	CDIV1	CDIV0	CSLVL	CSEN3	CSEN2	CSEN1	CSEN0	CSOE
SCSTR 1/0	CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0
SCSTR 3/2	CSDS 15	CSDS 14	CSDS 13	CSDS 12	CSDS 11	CSDS 10	CSDS9	CSDS8	CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0
SCSFR 1/0	CS2 LVL	CS2 SCINV	CS2 SPI	CS2 BDS	CS2 L3	CS2 L2	CS2 L1	CS2 L0	CS1 LVL	CS1 SCINV	CS1 SPI	CS1 BDS	CS1 L3	CS1 L2	CS1 L1	CS1 L0
SCSFR2	-								CS3 LVL	CS3 SCINV	CS3 SPI	CS3 BDS	CS3 L3	CS3 L2	CS3 L1	CS3 L0
TBYTE 1/0	CS0 TD7	CS0 TD6	CS0 TD5	CS0 TD4	CS0 TD3	CS0 TD2	CS0 TD1	CS0 TD0	CS0 TD7	CS0 TD6	CS0 TD5	CS0 TD4	CS0 TD3	CS0 TD2	CS0 TD1	CS0 TD0
TBYTE 3/2	CS3 TD7	CS3 TD6	CS3 TD5	CS3 TD4	CS3 TD3	CS3 TD2	CS3 TD1	CS3 TD0	CS2 TD7	CS2 TD6	CS2 TD5	CS2 TD4	CS2 TD3	CS2 TD2	CS2 TD1	CS2 TD0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

7.1 Serial Control Register (SCR)

The Serial Control Register (SCR) is used to enable/disable a transmit/received interrupt, enable/disable a transmit idle interrupt, and enable/disable data transmission and reception. Also, the register can set the SPI connection and reset the CSIO settings.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	(SMR)
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

[bit15] UPCL: Programmable clear bit

Initializes the CSIO internal state.

If set to "1":

- The CSIO is reset directly (software reset). However, the current register settings are kept. The transmit or received state is disconnected immediately.
- The baud rate generator reloads the BGR1/0 register value and restarts operation.
- All of transmit/received interrupt factors (SSR:TDRE, TBI, RDRF, ORE, TINT, CSE) are initialized.
- All of Serial Chip Select pins become inactive state.

If set to "0":

No effect on the operation.

"0" is always read from this bit.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	Programmable clear	

Notes:

- *Disable an interrupt first, and then execute the programmable clear instruction.*
- *If the FIFO operation is used, disable it (FCR0:FE[2:1]=00) first and then execute the programmable clear instruction.*

[bit14] MS: Master/Slave function select bit

Selects the master or slave mode.

bit	Description
0	Master mode
1	Slave mode

Notes:

- *If the slave mode is selected and if SMR:SCKE=0, the external clock is entered directly.*
- *Set this bit when the data transmission and reception is disabled (TXE=RXE=0).*
- *After you have set the MS bit, enable data reception (RXE=1).*

[bit13] SPI: SPI corresponding bit

This bit allows the SPI communication.

bit	Description
0	Normal synchronous transfer
1	SPI correspond

Notes:

- *Set this bit when the data transmission and reception is disabled (TXE=RXE=0).*
- *This bit is used for one of the following cases[*]: (TYPE1-M4, TYPE2-M4 products)*
  - *When the Chip Select pin is disabled (SCSCR:CSEN0="0").*
  - *When in Slave Mode Operation (SCR:MS=1)*
- *This bit is used for any of the following cases. (TYPE3-M4 product)*
  - *When the Chip Select pin is disabled (SCSCR:CSEN3-0="0000").*
  - *When in Slave Mode Operation (SCR:MS=1)*
  - *When the data format of chip select pin is disabled (ESCR:CSFE=0).*
  - *When the data format of chip select pin is enabled (ESCR:CSFE=1) and chip select pin0 is active.*

[bit12] RIE: Received interrupt enable bit

- This bit enables or disables an output of received interrupt request to the CPU.
- If the RIE bit and the received data flag bit (SSR:RDRF) are "1", or if any of error flag bits (ORE) is "1", a received interrupt request is output.

bit	Description
0	Disables the received interrupt.
1	Enables the received interrupt.

[bit11] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of transmit interrupt request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a transmit interrupt request is output.

bit	Description
0	Disables a transmit interrupt.
1	Enables a transmit interrupt.

[bit10] TBIE: Transmit bus idle interrupt enable bit

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If the TBIE bit and SSR:TBI bit are "1", a transmit bus idle interrupt request is output.

bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit9] RXE: Data received enable bit

Enables or disables a CSIO data reception.

bit	Description
0	Disables data reception.
1	Enables data reception.

Notes:

- *If data reception is disabled (RXE=0), the current data reception is stopped immediately.*
- *After you have set the MS bit and SMR:SCINV bit, enable the data reception (RXE=1).*

[bit8] TXE: Data transmission enable bit

Enables or disables a CSIO data transmission.

bit	Description
0	Disables the transmission.
1	Enables the transmission.

Notes:

- *If data transmission is disabled (TXE=0), the current data transmission is stopped immediately.*
- *When the Serial Chip Select is used (SCSCR:CSEN=1) in Master Mode Operation(SCR:MS=1), execute the programmable reset. (SCR:UPCL=1)*

7.2 Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to select an operation mode, to set a transmission direction, data length and serial clock inversion, and to enable or disable an output of serial data and clock to their pins.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCR)	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
Attribute		R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial value		0	0	0	-	0	0	0	0

[bit7:5] MD2, MD1, MD0: Operation mode set bits

These bits set an operation mode.

"0b000": Sets operation mode 0 (asynchronous normal mode).
"0b001": Sets operation mode 1 (asynchronous multiprocessor mode).
"0b010": Sets operation mode 2 (clock synchronous mode).
"0b011": Sets operation mode 3 (LIN communication mode).
"0b100": Sets operation mode 4 (I^2C mode).

*This chapter explains the registers and their operation in operation mode 2 (clock synchronous mode).

bit7	bit6	bit5	Description
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multiprocessor mode)
0	1	0	Operation mode 2 (clock synchronous mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I^2C mode)
Values other than the above			Setting is prohibited.

Notes:

- *Any bit setting other than above is prohibited.*
- *To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) and switch the operation mode continuously.*
- *After the operation mode has been set, set each register correctly.*

[bit4] Reserved: Reserved bit

The read value is "0". Be sure to write "0".

[bit3] SCINV: Serial clock invert bit

Inverts the serial clock format. This bit is used for the communication of the Serial Chip Select pin0 when the chip select is used in the master mode operation (SCR:MS=0).

If set to "0":

- The signal mark level of serial clock output is set to "HIGH".
- The transmit data is output at a falling edge of serial clock during normal transfer, but it is output in synchronization with a rising edge of serial clock during SPI transfer.
- The received data is sampled at a rising edge of serial clock during normal transfer, but it is sampled at a falling edge of serial clock during SPI transfer.

If set to "1":

- The signal mark level of serial clock output is set to "LOW".
- The transmit data is output at a rising edge of serial clock during normal transfer, but it is output in synchronization with a falling edge of serial clock during SPI transfer.
- The received data is sampled at a falling edge of serial clock during normal transfer, but it is sampled at a rising edge of serial clock during SPI transfer.

bit	Description
0	Signal mark level "HIGH" format
1	Signal mark level "LOW" format

Notes:

- *Always set this bit when transmission and reception are disabled (TXE=RXE=0).*
- *After setting the SCINV bit, enable data reception (SCR:RXE=1).*
- *This bit is used in the one of the following cases: (TYPE1-M4, TYPE2-M4 products)*
  - *When the chip select pin is disabled (SCSCR:CSEN=”0”)*
  - *In slave mode operation (SCR:MS=1)*
- *This bit is used in the any of the following cases: (TYPE3-M4 product)*
  - *When the chip select pin is disabled (SCSCR:CSEN3-0=”0000”)*
  - *In slave mode operation (SCR:MS=1)*
  - *When the data format of chip select pin is disabled (ESCR:CSFE=0).*
  - *When the data format of chip select pin is enabled (ESCR:CSFE=1) and chip select pin0 is active.*

[bit2] BDS: Transfer direction select bit

Specifies to transfer the least significant bit of the transfer serial data first (LSB first; BDS=0) or the most significant bit first (MSB first; BDS=1). This bit is utilized for the communication of chip select pin0 when chip select is enabled during Master mode (SCR:MS=0).

bit	Description
0	LSB first (The least significant bit is first transferred.)
1	MSB first (The most significant bit is first transferred.)

Notes:

- *Always set this bit when transmission and reception are disabled (SCR:TXE=RXE=0).*
- *This bit is used in the one of the following cases: (TYPE1-M4, TYPE2-M4 products)*
  - *When the chip select pin is disabled (SCSCR:CSEN="0")*
  - *In slave mode operation (SCR:MS=1)*
- *This bit is used in the any of the following cases: (TYPE3-M4 product)*
  - *When the chip select pin is disabled (SCSCR:CSEN3-0="0000")*
  - *In slave mode operation (SCR:MS=1)*
  - *When the data format of chip select pin is disabled (ESCR:CSFE=0).*
  - *When the data format of chip select pin is enabled (ESCR:CSFE=1) and chip select pin0 is active.*

[bit1] SCKE: Master mode serial clock output enable bit

This bit controls the serial clock I/O port.

bit	Description
0	Disables a serial clock output.
1	Enables a serial clock output.

Note:

- *If this bit is used as the SCK pin, the GPIO must also be set.*

[bit0] SOE: Serial data output enable bit

This bit enables or disables a serial data output.

bit	Description
0	Disables a serial data output.
1	Enables a serial data output.

Note:

- *If this bit is used as the SOT pin, the GPIO must also be set.*

7.3 Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the current transmission/reception state, check the Received Error flag, and clear the Received Error flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	-	-	Reserved	ORE	RDRF	TDRE	TBI	(ESCR)
Attribute	R/W	-	-	-	R	R	R	R	
Initial value	0	-	-	-	0	0	1	1	

[bit15] REC: Received error flag clear bit

This bit clears the ORE flag of the Serial Status Register (SSR).

- If this bit is set to "1", the error flag is cleared.
- This bit has no effect on the operation if set to "0".

"0" is always read.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	Clears the Received Error flag (FRE, ORE).	

[bit14:13] - : Unused bits

The values of these bits are undefined when read.
These bits have no effect on the operation when written.

[bit12] Reserved : Reserved bits

The read value is "0". Be sure to write "0".

[bit11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the ORE and SCR:RIE bits are "1", a received interrupt request is output.
- If this flag is set, data of the Received Data Register (RDR) is invalid.
- If this flag is set when received FIFO is used, the received FIFO enable bit is cleared and the received data is not stored in received FIFO.

bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit10] RDRF: Received data full flag bit

- This flag shows the state of Received Data Register (RDR).
- When the received data is loaded in the RDR, this bit is set to "1". When data is read from the Received Data Register (RDR), this bit is cleared to "0".
- If the RDRF bit and SCR:RIE bit are "1", a received interrupt request is output.
- If received FIFO is used and if the preset amount of data is received in received FIFO, the RDRF bit is set to "1".
- If received FIFO is used, if both of the following conditions are satisfied, and if the Received Idle state continues more than 8 baud rate clocks, the RDRF bit is set to "1".
  - The received FIFO idle detect enable bit (FCR1:FRIIE) is "1".
  - The preset data amount is not received and some data remains in received FIFO.

If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted.

- If the received FIFO is used and if this buffer is emptied, this bit is cleared to "0".

bit	Description
0	The Received Data Register (RDR) is empty.
1	The Received Data Register (RDR) contains data.

[bit9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When data is loaded to the transmit shift register and when the transmission is started, this bit is set to "1" to indicate that the TDR does not have the valid data.
- If the TDRE bit and SCR:TIE bit are "1", a transmit interrupt request is output.
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- For the TDRE bit set/reset timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register (TDR) is empty.

[bit8] TBI: Transmit bus idle flag bit

- This bit indicates that the CSIO is not transmitting data.
- When data is written in the Transmit Data Register (TDR), this bit is set to "0".
- If the Transmit Data Register (TDR) is empty (TDRE=1) and if no transmission is started at the state that Serial Chip Select pin is deselected, this bit is set to "1".
- When the UPCL bit of the Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- If this bit is "1" and if a transmit bus Idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

bit	Description
0	During data transmission
1	No data transmission

7.4 Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to set a transmit/received data length and to fix the serial data output to the "HIGH" state.

■ TYPE1-M4, TYPE2-M4 products

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	SOP	L3	Reserved	WT1	WT0	L2	L1	L0
Attribute		R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	-	0	0	0	0	0

■ TYPE3-M4 product

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	SOP	L3	CSFE	WT1	WT0	L2	L1	L0
Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7] SOP: Serial output pin set bit

■ This bit sets the serial data output pin to the "HIGH" state. When this bit is set to "1", the SOT pin is set to "HIGH". After that, this bit needs not be set to "0".

■ When it is read, "0" is always read.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	Sets the SOT pin to "HIGH" state.	

Note:

- *Do not set this bit during serial data transmission.*

[bit5] Reserved : Reserved bit (TYPE1-M4, TYPE2-M4 products)

For TYPE1-M4 and TYPE2-M4 products, the read value is "0". Be sure to write "0"

[bit5] CSFE : Serial Chip Select Format enable bit (TYPE3-M4 product)

TYPE3-M4 product has this bit.

This bit enables/disables the serial chip select format.When this bit sets to "1", each serial chip select pin format can be set as following.

- Active level of serial chip select
- Mark level of serial clock
- Selection of SPI transfer/Normal transfer
- Serial data direction
- Data length of serial data

bit	Description
0	Set to same data format and clock format in all serial chip select pin
1	Enable to different formats and clock formats in each serial chip select pin

Notes:

- *This bit is disabled in the any of the following cases:*

- *When the chip select pin is disabled (SCSCR:CSEN3-0="0000")*
- *In slave mode operation (SCR:MS=1)*
- *Set this bit when transmission is disabled (SCR:TXE=0).*

[bit4:3] WT1, WT0: Data transmit/received wait select bits

In master mode operation, these bits set a wait count for continuous data transmission or reception. In slave mode operation, these bits are set to "00".

- When "00" is set, SCK is output continuously.
- When "01" is set, SCK is output after 1-bit time wait.
- When "10" is set, SCK is output after 2-bit time wait.
- When "11" is set, SCK is output after 3-bit time wait.

bit4	bit3	Description
0	0	0 bit
0	1	1 bit
1	0	2 bits
1	1	3 bits

[bit6, bit2:0] L3, L2, L1, L0: Data length select bits

These bits set a length of transmit/received data.

L3	L2	L1	L0	Description
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length

Notes:

- *Any bit setting other than above is prohibited.*
- *These bits are used in one of the following conditions: (TYPE1-M4, TYPE2-M4 products)*
  - *When the Chip Select pin is disabled (SCSCR:CSEN3-0="0000"b).*
  - *At slave mode operation (SCR:MS=1)*
- *These bits are used in any of the following conditions: (TYPE3-M4 product)*
  - *When the Chip Select pin is disabled (SCSCR:CSEN3-0="0000").*
  - *At slave mode operation (SCR:MS=1)*
  - *When the data format of chip select pin is disabled (ESCR:CSFE=0).*
  - *When the data format of chip select pin is enabled (ESCR:CSFE=1) and chip select pin0 is active.*

7.5 Received Data Register/Transmit Data Register (RDR/TDR)

The Received and Transmit Data Registers are allocated at the same address. This register functions as the Received Data Register when data is read from it. This register functions as the Transmit Data Register when data is written in it.

Received Data Register (RDR)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Attribute	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Received Data Register (RDR) is a 16-bit data buffer register for serial data reception.

- When serial data signals are sent to the Serial input pin (SIN), they are converted by a shift register and stored in the Received Data Register (RDR).
- Considering data length, the received data is stored from the lower bit and other bits are set to”0”. Example: “45”h is received in 8-bit data length, D7 to D0 =”45”h, D31 to D8 =0
- When the received data is stored in the Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is set to "1". If a received interrupt is enabled (SCR:RIE=1), a received interrupt request is generated.
- The Received Data Register (RDR) must be read only when the received data full flag bit (SSR:RDRF) is "1". When data is read from the Serial Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is cleared to "0" automatically.
- If a received error occurs (SSR:ORE), data in the Received Data Register (RDR) is invalid.

Notes:

- *If the received FIFO is used and if a certain count of data is received by the received FIFO, the RDRF bit is set to "1".*
- *If received FIFO is used and if this buffer is emptied, the RDRF bit is cleared to "0".*
- *If received FIFO is used and if a received error occurs (SSR:ORE), the received FIFO enable bit is cleared and the received data is not stored in received FIFO.*

Transmit Data Register (TDR)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Attribute	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a 16-bit data buffer register for serial data transmission.

- If data transmission is enabled (SCR:TXE=1) and if the transmit data is written in the Transmit Data Register (TDR), the transmit data is transferred to the transmit shift register. Then, the data is converted into serial data, and output at the serial data output pin (SOUT).
- Considering the bit length, the transmit data is stored from the lower bit and other bits are invalid. Example: “0x45” is received in 8-bit data length, D7 to D0 =”0x45”, D15 to D8 =0
- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When the transmit data is transferred to the transmit shift register and data transmission is started, and if transmit FIFO is disabled or if transmit FIFO is empty, the transmit data empty flag (SSR:TDRE) is set to "1".
- If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written in the buffer. If a transmit interrupt is enabled, a transmit interrupt occurs. The next transmit data must be written only after the transmit interrupt has occurred or when the transmit data empty flag (SSR:TDRE) is "1".
- If the transmit data empty flag (SSR:TDRE) is "0" and if transmit FIFO is disabled or transmit FIFO is full, the transmit data cannot be written in the Transmit Data Register (TDR).

Notes:

- *The Transmit Data Register is a write-only register. While the Received Data Register is a read-only register. As these two registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) operation cannot be used.*
- *For the transmit data empty flag (SSR:TDRE) set timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".*

7.6 Serial Support Control Register (SACSR)

Serial Support Control Register (SACSR) is used to control the serial test, select the starting method of serial timer, enable/disable the timer interrupt, enable/disable the synchronous transmission, set the division ratio for the operation clock of serial timer, and enable/disable the serial timer.

bit	15	14	13	12	11	10	9	8
Field	Reserved		TBEEN	CSEIE	CSE	-	-	TINT
Attribute	-		R/W	R/W	R/W	-	-	R/W
Initial Value	00		0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TINTE	TSYNE	-	TDIV3	TDIV2	TDIV1	TDIV0	TMRE
Attribute	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

[bit15:14] Reserved: Reserved bits

At reading: The read value is “0”.
At writing: Always write “0”.

[bit13] TBEEN: Transfer Byte Error Enable bit

In Master mode operation (SCR:MS=0), enables/disables the real chip select error occurrence.

For details, see “2.6 Chip select error occurrence and flag set timing”.

bit	Transfer Byte Error Enable bit
0	Disables the chip select error occurrence in Master mode operation (SCR:MS=0).
1	Enables the chip select error occurrence in Master mode operation (SCR:MS=0).

Note:

- *Change this bit when Data transmission and reception is disabled (SCR:TXE=RXE=”0”).*

[bit12] CSEIE: Chip Select Error Interrupt Enable bit

- This bit is used to enable/disable the chip select error interrupt request output.
- When CSEIE bit and Chip Select Error Flag bit (CES) are “1”, outputs the transmission interrupt request.

bit	Chip Select Error Interrupt Enable bit
0	Disables the Chip Select Error Interrupt.
1	Enables the Chip Select Error Interrupt Enable bit.

[bit11] CES: Chip Select Error Flag

This bit is used to indicate the presence or absence of the Chip Select Error occurrence.

For details, see “2.6 Chip select error occurrence and flag set timing”.

When this bit is “1” and the Chip Select Error Interrupt Enable bit (CSEIE) is “1”, outputs the data transmission interrupt request.
When this bit is set to”1”, this bit is reset to “0”.
Setting “1” to this bit is invalid.

bit	Chip Select Error Flag
0	Chip Select Error occurs
1	No Chip Select Error occurs..

Notes:

- *This bit is reset to “0” by executing the software reset (SCR:UPCL=”1”).*
- *“1” is read by reading with Read-Modify-Write instruction.*
- *When the Serial Chip Select is not used (SCSCR:CSEN0=0) in Slave mode operation (SCR:MS=1), this bit cannot be set to “1”.*
- *When a Chip Select Error occurs, disable the data transmission and then write “0” to this bit. To restart the data transmission, write “0” to this bit to enable the data transmission (SCR:TXE=1) and write the transmit data to the Transmission Data Register (TDR).*
- *If a noise of one bus clock or more occurs on the Serial Chip Select input in the slave mode transmission, this bit may be set to “1”. In such case, restart the transmission after the completion of the master mode transmission.*

[bit8] TINT: Timer Interrupt Flag

When the values of the Serial Timer Register (STMR) and the Serial Timer Comparison Register (STMCR) match, the Serial Timer Register (STMR) is set to “0” and this register is set to “1”.

When this bit is set to “1” and the Timer Interrupt Enable bit (TINTE) is set to “1”, the stats interrupt request is output.

When this bit is set to”1”, this bit is reset to “0”.
Setting “1” to this bit is invalid.

bit	Description
0	No Timer Interrupt Request exists.
1	Timer Interrupt Request exists.

Notes:

- *This bit is reset to “0” by executing the software reset (SCR:UPCL=”1”).*
- *“1” is read by reading with Read-Modify-Write command.*
- *When the Synchronous Transmission Enable bit (TSYNE) is “1”, this bit is not set to “1”.*

[bit7] TINTE: Timer Interrupt Enable bit

This bit is used to enable/disable the Timer Interrupt to CPU.

When this bit is “1” and Timer Interrupt Flag (TINT) is “1”, the Status Interrupt Request is output.

bit7	Description
0	Disables an interrupt with serial timer.
1	Enables an interrupt with serial timer.

[bit6] TSYNE: Synchronous Transmission Enable bit

This bit enables/disables the synchronous transmission.

When this bit is “1” and the following condition is met, the transmission is started.

- The values of Serial Timer Register (STMR) and Serial Timer Comparison Register (STMCR) meet at the transmission synchronizing with a timer.

bit	Description
0	Disables the synchronous transmission. The serial timer is used as a timer.
1	Enables the synchronous transmission. The serial timer is not used as a timer.

Notes:

- *Only when the Serial Timer Enable bit (TMRE) is “0”, this bit can be changed.*
- *When the transmission is disabled (SCR:TXE=0) at Synchronous Transmission enabled (TSYNE=1), the transmission is not started even the following condition is met.*
- *The values of Serial Timer Register (STMR) and Serial Timer Comparison Register (STMCR) meet.*
- *In Slave mode operation (SCR:MS=”1”), this bit is fixed to “0” internally.*

[bit4:1] TDIV3:0: Timer Operation Clock Division bit

This bit is used to set the serial timer division ratio.

bit4	bit3	bit2	bit1	Timer Operation Clock						
				Division Ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	ϕ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	ϕ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

ϕ: Bus clock

Notes:

- *This bit can be changed only when the Serial Timer Enable bit (TMRE) is “0”.*
- *Other than the above change is disabled.*

[bit0] TMRE: Serial Timer Enable bit

This bit enables/disables the serial timer operation.

bit	Serial Timer Enable bit
0	Stops the Serial Timer operation. At the time of stop, the value of the Serial Timer 8STMR) is held.
1	When this bit is changed from “0” to “1”, initialize the Serial Timer Register (STMR) to “0” and start the operation of the Serial Timer.

Note:

- *When the synchronous transmission with the Serial Timer is executed, change this bit from “0” to “1” at transmission disabled.*

7.7 Serial Timer Register (STMR)

The Serial Timer Register (STMR) is used to indicate the timer value of the serial timer.

Bit Configuration of Serial Timer Register (STMR)

bit	15	14	13	12	11	10	9	8
Field	TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TM[15:0]: Timer Data bits

These bits indicate the timer value of the serial timer.

During the timer operation, the timer value of the serial timer is incremented by 1 every timer operation clock (SACSR:TDIV3:0).

Note:

- *At starting the timer operation, this bit is initialized to “0”.*

7.8 Serial Timer Comparison Register (STMCR)

This register is used to set the timer comparison value of the serial timer.

Bit Configuration of Serial Timer Register (STMCR)

bit	15	14	13	12	11	10	9	8
Field	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] TC15:0: Compare bits

Set the comparison values of the serial timer.

This bit is compared with the Serial Timer Register (STMR) and the Serial Timer Register (STMR) is set to “0” if the values of this bit and the Serial Timer Register (STMR) meet when the Serial Timer Register (STMR) is revised. At that time, when the synchronous transmission is disabled (SACSR:TSYNE=”0”), the Timer Interrupt Flag (SACSR: TINT) is set to “I” and when the synchronous transmission is enabled (SACSR:TSYNE=”1”), the transmission is started.

The interval of executing the following operations is (STMCR:TC+1) × Timer Operation Clock (specified with SACSR:TDIV3:0.)

- SACSR:TINT is set to “1”.
- The transmission is started with the transmission synchronizing with the serial clock.

Notes:

- *When all the following conditions are met, the Timer Interrupt Flag (SACSR:TINT) is fixed to “1”.*
- *Synchronous transmission is disabled (SACSR:TSYNE=”0”).*
- *This register is set to “0x0000”.*
- *Timer is operating.*
- *Timer Operation Clock Division value (SACSR:TDIV) is set to “0b0000”.*
- *Only when the Serial Timer is disabled (SACSR:TMRE=”0”), this register can be changed.*

7.9 Serial Chip Select Control Status Register (SCSCR)

This register is used to select the start pin and end pin of the Serial Chip Select, to display the output pin of the Serial Chip Select, to hold the active level of the Serial Chip Select, to reverse the Serial Chip Select, and to enable/disable the output of the Serial Chip Select.

(TYPE1-M4, TYPE2-M4 products)
Bit Configuration of Serial Chip Select Control Status Register (SCSCR)

bit	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	SCAM	CDIV2
Attribute	-	-	-	-	-	-	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CDIV1	CDIV0	CSLVL	-	-	-	CSEN0	CSOE
Attribute	R/W	R/W	R/W	-	-	-	R/W	R/W
Initial Value	0	0	1	0	0	0	0	0

(TYPE3-M4 product)
Bit Configuration of Serial Chip Select Control Status Register (SCSCR)

bit	15	14	13	12	11	10	9	8
Field	SST1	SST0	SED1	SED0	SCD1	SCD0	SCAM	CDIV2
Attribute	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CDIV1	CDIV0	CSLVL	CSEN3	CSEN2	CSEN1	CSEN0	CSOE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	1	0	0	0	0	0

[bit15:14] SST1-0: Serial Chip Select Active Start bit (TYPE3-M4 product)

Selects the starting pin of serial chip select.

Serial chip select becomes active by rotation from the pin set by this bit after transmission is changed from disabled (SCR:TXE= "0") to enabled (SCR:TXE= "1") when transmission data is written in TDR.

bit15:14	Description
00	Serial chip select starting pin is SCS0
01	Serial chip select starting pin is SCS1
10	Serial chip select starting pin is SCS2
11	Serial chip select starting pin is SCS3

Notes:

- *Always set this bit when transmission and reception are disabled(SCR:TXE=RXE= "0")*
- *When Serial Chip Select Start bit(SCSCR:SST1-0) is equivalent with Serial Chip select End bit(SCSCR:SED1-0), Chip Select operates at the only pin set by these bits.*
- *This bit is disabled in slave mode operation (SCR:MS=1)*
- *Only serial chip select pins set enabled (CSEN=1) becomes active.*
- *Set the serial chip select pin enabled (CSEN=1) which is set by this bit when serial chip select pin is enabled during Master mode(SCR:MS= "0").*

[bit13:12] SED1-0: Serial Chip Select Active End bit (TYPE3-M4 product)

Selects the ending pin of serial chip select.

After serial chip select pin set by this bit becomes active, serial chip select pin set by Serial Chip Select Active Start bit(SST1,SST0) becomes active.

bit13:12	Description
00	Serial chip select ending pin is SCS0
01	Serial chip select ending pin is SCS1
10	Serial chip select ending pin is SCS2
11	Serial chip select ending pin is SCS3

Notes:

- *Always set this bit when transmission and reception are disabled(SCR:TXE=RXE= ”0”)*
- *When Serial Chip Select Start bit (SCSCR:SST1-0) is equivalent with Serial Chip select End bit(SCSCR:SED1-0), chip select operates at the only pin set by these bits.*
- *This bit is disabled in slave mode operation (SCR:MS=1)*
- *Only serial chip select pins set enabled (CSEN=1) become active.*
- *Set the serial chip select pin enabled (CSEN=1) which is set by this bit when serial chip select pin is enabled during Master mode(SCR:MS= "0").*

[bit11:10] SCD1-0: Serial Chip Select Active Display bit (TYPE3-M4 product)

Display the active serial chip select pin.

bit11:10	Description
00	SCS0 is active
01	SCS1 is active
10	SCS2 is active
11	SCS3 is active

Notes:

- *When serial chip select pin is inactive, display next serial chip select pin becomes active.*
- *This bet becomes "00" in slave mode(SCR:MS="1") or transmission disabled(SCR:TXE=”0”) or software reset(SCR:UPCL=1).*

[bit9] SCAM: Serial Chip Select Active Hold bit

Selects the holding or not-holding the active status of Serial Chip Select pin.

For details, see “Serial Chip Select Active Holding Operation (SCSCR:SCAM=1)(Available only in Master mode operation (SCR:MS=0)) in “5. Serial Chip Select Operation”.

bit	Serial Chip Select Active Holding bit
0	Dose not hold the Active Status of Serial Chip Select pin.
1	Holds the Active Status of Serial Chip Select pin.

Notes:

- *When the transmission is disabled (SCR:TXE=”0”) and Software reset is executed (SCR:UPCL=”1”), the Serial Chip Select pin becomes inactive irrespective of the value of this bit.*
- *When a Serial Chip Error occurs (SACSR:CSE=1), the Serial Chip Select pin becomes inactive irrespective of the value of this bit.*

[bit8:6] CDIV2:0: Serial Chip Select Timing Operation Clock Division bit

Set the division ratio of the Serial Chip Select Timing Operation Clock.

bit8	bit7	bit6	Serial Chip Select Timing Operation Clock						
			Division Ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs

ϕ: Bus clock

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”).*
- *The setting of this bit is invalid in Slave mode operation (SCR:MS="1").*
- *The settings other the above are prohibited.*

[bit5] CSLVL: Serial Chip Select Level Setting bit

Selects “High” or “Low” for the Serial Chip Select pin level in inactive state.

This bit is available for Chip Select pin0.

bit	Serial Chip Select Level Setting bit
0	Sets the Inactive Level to “Low”.
1	Sets the Inactive Level to "High".

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”).*
- *This bit is used in the following condition: (TYPE1-M4, TYPE2-M4 products)*
  - *In Slave mode operation (SCR:MS=1)*
- *This bit is used in the following condition: (TYPE3-M4 product)*
  - *In Slave mode operation (SCR:MS=1)*
  - *When the data format of chip select pin is disabled (ESCR:CSFE=0).*
  - *When the data format of chip select pin is enabled (ESCR:CSFE=1) and chip select pin0 is active.*

[bit4:2] CSEN3-1: Serial Chip Select Enable bit (TYPE3-M4 product)

TYPE3-M4 product has these bits.

These bits are used to enable or disable the Serial Chip Select pin.

CSEN3 bit is equivalent with SCS3 pin, CSEN2 bit is equivalent with SCS2 pin, CSEN1 bit is equivalent with SCS1 pin.

bit	Serial Chip Select Enable bit
0	Disables the operation of Serial Chip Select pin.
1	Enables the operation of Serial Chip Select pin.

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE="0")*
- *When CSEN3-0 is set to "0000" in Master mode operation (SCR:MS=0), the transmission and reception operations are executed irrespective of the Serial Chip Select pin.*
- *Disable the Serial Chip Select pin not used.*

[bit1] CSEN0: Serial Chip Select Enable bit

This bits is used to enable or disable the Serial Chip Select pin.

CSEN0 bit is equivalent with SCS0 pin.

In Slave mode operation (SCR:MS=1), only CSEN0 bit can enable or disable the Serial Chip pin.

bit	Serial Chip Select Enable bit
0	Disables the operation of Serial Chip Select pin.
1	Enables the operation of Serial Chip Select pin.

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE="0")*
- *When CSEN0 is set to "0" in Master mode operation (SCR:MS=0), the transmission and reception operations are executed irrespective of the Serial Chip Select pin.*
- *When CSEN0 is set to "0" in Slave mode operation (SCR:MS=1), the transmission and reception operations are executed irrespective of the Serial Chip Select pin.*
- *Disable the Serial Chip Select pin not used.*

[bit0] CSOE: Serial Chip Select Output Enable bit

This bit is used to enable or disable the Serial Chip Select pin Output.

bit	Serial Chip Select Output Enable bit
0	Disables all the Serial Chip Select pins.
1	Enables all the Serial Chip Select pins.

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE="0")*
- *In Slave mode operation (SCR:MS="1"), This bit is set to "0".*

7.10 Serial Chip Select Timing Register (SCSTR3:0)

These registers are used to set the setup delay time, the hold delay time, and deselect time of Serial Chip Select.

Bit Configuration of Serial Chip Select Timing Registers (SCSTR1, SCSTR0)

bit	15	14	13	12	11	10	9	8
Field	CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

[bit15:8] CSSU[7:0]: Serial Chip Select Setup Delay bit

Set the period from the time when the Serial Chip Select pin becomes active to the time when the Serial Clock is output. When these bits are set to “00”h, the time when the Serial Chip Select pin becomes active becomes the same as the time when the Serial Clock is output.

bit15:8	Setup Delay Time
0x00	The Serial Chip Select pin becomes active on starting the output of the Serial Clock.
0x01	1×Serial Chip Select Timing Operation Clock
0x02	2×Serial Chip Select Timing Operation Clock
: :	: :
0xFE	254×Serial Chip Select Timing Operation Clock
0xFF	255×Serial Chip Select Timing Operation Clock

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”)*
- *In Slave mode operation (SCR:MS=”1”), this bit cannot be set.*

[bit7:0] CSHD[7:0]: Serial Chip Select Hold Delay bits

Set the period from the time when the Serial Clock output is finished to the time when the Serial Chip Select pin becomes inactive. When these bits are set to “00”h, the time when the Serial Chip Select pin becomes inactive becomes the same as the time when the Serial Clock output is finished.

bit7:0	Hold Delay Time
0x00	The time when the Serial Chip Select pin becomes inactive becomes the same as the time when the Serial Clock output is finished.
0x01	1×Serial Chip Select Timing Operation Clock
0x02	2×Serial Chip Select Timing Operation Clock
: :	: :
0xFE	254×Serial Chip Select Timing Operation Clock
0xFF	255×Serial Chip Select Timing Operation Clock

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”)*
- *In Slave mode operation (SCR:MS=”1”), this bit cannot be set.*

Bit Configuration of Serial Chip Select Timing Register (SCSTR3, SCSTR2)

bit	15	14	13	12	11	10	9	8
Field	CSDS15	CSDS14	CSDS13	CSDS12	CSDS11	CSDS10	CSDS9	CSDS8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

[bit15:0] CSDS[15:0]: Serial Chip Deselect bits

Set the minimum period from the time when the Serial Chip Select pin becomes inactive to the time when the Serial Chip Select pin becomes active again.

bit15:0	Deselect Minimum Time
0x0000	No Deselect minimum time (5 bus clock time)
0x0001	1×Serial Chip Select Timing Operation clock
0x0002	2×Serial Chip Select Timing Operation clock
: :	: :
0xFFFE	65534×Serial Chip Select Timing Operation clock
0xFFFF	65535×Serial Chip Select Timing Operation clock

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE="0")*
- *In Slave mode operation (SCR:MS="1"), this bit cannot be set.*
- *Irrespective of the deselect time setting, 5 bus clock times or more are required for the period the time when the Serial Chip Select pin becomes inactive to the time when the Serial Chip Select pin becomes active again.*
- *Do not set SCSTR2:CSDS=0x0001 and SCSCR:CDIV=0b000 at the same time.*

7.11 Serial Chip Select Format Register(SCSFR2-0)

TYPE3-M4 product has this register.

This register is used to set the active level, clock Inversion, SPI mode, data direction and data length in each serial chip select pin.

(TYPE3-M4 product)
Bit Configuration of Serial Chip Select Format Register (SCSFR2-0)

Figure 7-1 and Figure 7-2 show the bit configuration of Serial Chip Select Format Register(SCSFR2-0).

Figure 7-1 Bit configuration of Serial Chip Select Format Register (SCSFR1-0)

bit	15	14	13	12	11	10	9	8
Field	CS2 CSLVL	CS2 SCINV	CS2 SPI	CS2 BDS	CS2 L3	CS2 L2	CS2 L1	CS2 L0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	CS1 CSLVL	CS1 SCINV	CS1 SPI	CS1 BDS	CS1 L3	CS1 L2	CS1 L1	CS1 L0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	0	0	0	0	0	0	0

[bit15]CS2CSLVL: Serial Chip Select 2 Level Setting bit

Selects “High” or “Low” for the Serial Chip Select pin level in inactive state of Serial Chip Select pin when Serial Chip Select Format is enabled(ESCR:CSFE=1).

This bit is available for Chip Select pin2.

bit	Serial Chip Select Level Setting bit
0	Sets the Inactive Level to “L”.
1	Sets the Inactive Level to "H".

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”).*
- *This bit is disabled in Slave mode operation (SCR:MS=1).*
- *This bit is disabled when the data format of chip select pin is disabled (ESCR:CSFE=0).*

[bit14]CS2SCINV: Serial Clock Invert bit of Serial Chip Select 2

Inverts the serial clock format. This bit is used in active state of Serial Chip Select pin when Serial Chip Select Format is enabled (ESCR:CSFE=1).

This bit is available for Chip Select pin2.

If set to "0":

- The signal mark level of serial clock output is set to "HIGH".
- The transmit data is output at a falling edge of serial clock during normal transfer, but it is output in synchronization with a rising edge of serial clock during SPI transfer.
- The received data is sampled at a rising edge of serial clock during normal transfer, but it is sampled at a falling edge of serial clock during SPI transfer.

If set to "1":

- The signal mark level of serial clock output is set to "LOW".
- The transmit data is output at a rising edge of serial clock during normal transfer, but it is output in synchronization with a falling edge of serial clock during SPI transfer.
- The received data is sampled at a falling edge of serial clock during normal transfer, but it is sampled at a rising edge of serial clock during SPI transfer.

bit	Description
0	Signal mark level "HIGH" format
1	Signal mark level "LOW" format

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”).*
- *This bit is disabled in Slave mode operation(SCR:MS=1).*
- *This bit is disabled when the data format of chip select pin is disabled(ESCR:CSFE=0).*

[bit13] CS2 SPI: SPI corresponding bit of Serial Chip Select 2

This bit allows the SPI communication in active state of Serial Chip Select pin when Serial Chip Select Format is enabled (ESCR:CSFE=1). .

If set to "0": Normal synchronous transfer

If set to "1": SPI correspond

This bit is available for Chip Select pin2.

bit	Description
0	Normal synchronous transfer
1	SPI correspond

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE=”0”).*
- *This bit is disabled in Slave mode operation (SCR:MS=1).*
- *This bit is disabled when the data format of chip select pin is disabled (ESCR:CSFE=0).*

[bit12] CS2BDS: Transfer direction select bit of Serial Chip Select 2

Specifies to transfer the least significant bit of the transfer serial data first (LSB first; BDS=0) or the most significant bit first (MSB first; BDS=1) in active state of Serial Chip Select pin. This bit is utilized when Serial Chip Select Format is enabled (ESCR:CSFE=1).

This bit is available for Chip Select pin2.

bit	Description
0	LSB first (The least significant bit is first transferred.)
1	MSB first (The most significant bit is first transferred.)

Notes:

- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE="0").*
- *This bit is disabled in Slave mode operation (SCR:MS=1).*
- *This bit is disabled when the data format of chip select pin is disabled (ESCR:CSFE=0).*

[bit11:8] CS2 L3, L2, L1, L0: Data length select bits of Serial Chip Select 2

These bits set a length of transmit/received data in active state of Serial Chip Select pin when Serial Chip Select Format is enabled (ESCR:CSFE=1).

This bit is available for Chip Select pin2.

bit11	bit10	bit9	bit8	Description
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length

Notes:

- *Any bit setting other than above is prohibited.*
- *This bit can be changed only when the transmission and reception operations are disabled (SCR:TXE=RXE="0").*
- *This bit is disabled in Slave mode operation (SCR:MS=1).*
- *This bit is disabled when the data format of chip select pin is disabled (ESCR:CSFE=0).*

[bit7:0] CS1CSLVL, CS1SCINV, CS1SPI, CS1BDS, CS1L3-0 :Setting bits of Serial Chip Select 1

These bits set serial chip select1.Refer to the description of serial chip select pin2 each bits for the details.

Figure 7-2 Bit configuration of Serial Chip Select Format Register (SCSFR2)

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	-	CS3 CSLVL	CS3 SCINV	CS3 SPI	CS3 BDS	CS3 L3	CS3 L2	CS3 L1	CS3 L0
Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Vaue		1	0	0	0	0	0	0	0

[bit7:0] CS3CSLVL, CS3SCINV, CS3SPI, CS3BDS, CS3L3-0 :Setting bits of Serial Chip Select 3

These bits set serial chip select3.Refer to the description of serial chip select pin2 each bits for the details.

7.12 Transfer Byte Register (TBYTE3-0)

This register is used to set the transfer data count at Serial Chip Select pin in active mode.

(TYPE1-M4, TYPE2-M4 products)
Bit Configuration of Transfer Byte (TBYTE0)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-								(TBYTE0)							
Attribute	-	-	-	-	-	-	-	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(TYPE3-M4 product)
Bit Configuration of Transfer Byte (TBYTE3-0)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(TYBTE1)								(TBYTE0)							
Attribute	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
Initial Value	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
Field	(TYBTE3)								(TBYTE2)							
Attribute	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
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Transfer Byte Register sets the transfer data count at Serial Chip Select pin in active mode. After the Serial Chip Select pin become active, the data of the count specified with this register is transferred and then the Serial Chip Select pin becomes inactive.

Serial Chip Select pin 0(SCS0) is corresponds to TBYTE0, Serial Chip Select pin 1(SCS1) is corresponds to TBYTE1, Serial Chip Select pin 2(SCS2) is corresponds to TBYTE2, Serial Chip Select pin 3(SCS3) is corresponds to TBYTE3.

When the Serial Chip Select is disabled (SCSCR:CSEN0=”0”), the Transfer Byte Register0 (TBYTE0) is used for the transmission synchronizing with a timer. After starting the transmission synchronizing with a timer, the data of count specified with TBYTE0 is transferred.

When this bit is changed during transfer operation (SSR:TBI=0), the setting of transfer data count changed becomes valid after the data of count initially specified has been finished.

TBYTE	Transfer Byte Register
Write	Write to TBYTE.
Read	TBYTE Setting Value

Notes:

- *When this bit is set to (00)h, the transfer count is eight times.*
- *In Slave mode operation (SCR:MS=1), this bit cannot be set.*

7.13 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-	(BGR1)							(BGR0)							
Attribute	-	R/W							R/W							
Initial value	-	0000000							0x00							

- Set a clock frequency division to the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).
- The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/BGR0 set value can be read.
- When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the reload counter starts its counting.

[bit15] - : Unused bit

This bit value is undefined when read.
This bit has no effect on the operation when written.

[bit14:8] BGR1: Baud Rate Generator Register 1

bit14:8	Description
Write	Writes data in bit8 to bit14 of reload counter.
Read	Reads the BGR1 set value.

[bit7:0] BGR0: Baud Rate Generator Register 0

bit7:0	Description
Write	Write data in bit0 to bit7 of reload counter.
Read	Reads the BGR0 set value.

Notes:

- *Data must be written in the Baud Rate Generator Register1, 0(BGR1 and BGR0) by 16-bit data accessing.*
- *If the reload value is even, the "HIGH" and "LOW" width of serial clock are as follows. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.*
  If SMR:SCINV="0", the "HIGH" width of serial clock is longer for 1 cycle of bus clock.
  If SMR:SCINV="1", the "LOW" width of serial clock is longer for 1 cycle of bus clock.
- *Set the reload value to 3 or more.*
- *If the current values of Baud Rate Generator Register1, 0(BGR1, BGR0) are changed, the new values are reloaded only after the counter value has reached "15h00". In order to validate the new set values immediately, change the BGR1/BGR0 set values and execute the CSIO reset instruction (SCR:UPCL).*
- *If received FIFO is used and if you wish to set the received FIFO idle detect enable bit (FCR1:FRIIE) to "1" and starts the slave mode operation, set the desired baud rate in BGR1/BGR0.*

7.14 FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select the transmit or received FIFO, enable the transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
Attribute	-	-	-	R/W	R/W	R/W	R/W	R/W	
Initial value	-	-	-	0	0	1	0	0	

[bit15:13] Reserved : Reserved bits

The read value is "0". Be sure to write "0".

[bit12] FLSTE: Re-transmit data lost detect enable bit

This bit enables the FLST bit detection.

If set to "0": The FLST bit detection is disabled.
If set to "1": The FLST bit detection is enabled.

bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

Note:

- *If you wish to set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".*

[bit11] FRIIE: Received FIFO idle detection enable bit

This bit sets to detect the received idle state if the received FIFO contains valid data and if it continues more than 8-bit hours. If the received interrupt is enabled (SCR:RIE=1), a received interrupt is generated when the received idle state is detected.

bit	Description
0	Disables the received FIFO idle detection.
1	Enables the received FIFO idle detection.

Note:

- *In case of using Received FIFO, set this bit to "1".*

[bit10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. If the transmit FIFO interrupt is enabled (FTIE=1) during this time, a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).
- Transmit FIFO is reset.

The FDRQ bit is reset when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

Notes:

- *If the FBYTE (for transmission) is "0", this bit cannot be set to "0".*
- *If this bit is "0", the FSEL bit state cannot be changed.*
- *If this bit is set to "1", it has no effect on the operation.*
- *If a read-modify-write instruction is issued, "1" is read.*

[bit9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit8] FSEL: FIFO select bit

This bit selects the transmit FIFO and received FIFO.

bit	Description
0	Transmit FIFO:FIFO1; Received FIFO:FIFO2
1	Transmit FIFO:FIFO2; Received FIFO:FIFO1

Notes:

- *This bit is not cleared by FIFO reset (FCR0:FCL[2:1]=11).*
- *To change this bit state, first disable the FIFO operation (FCR0:FE[2:1]=00).*

7.15 FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute		-	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7] - : Unused bit

"0" is always read.

"0" must always be written.

[bit6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- The FLSTE bit of FIFO Control Register 1 (FCR1) is "1", the write pointer of transmit FIFO matches the read pointer which has been saved by the FSET bit, and data is written in FIFO.

The FLST bit is reset when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data identified by the read pointer (saved by the FSET bit) is overwritten. Therefore, the FLD bit cannot set the data re-transmission even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in the FIFO buffer again.

bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit 5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.
When the re-transmission setting has finished, this bit is set to "0".

bit	Description
0	Not reloaded
1	Reloaded

Notes:

- *If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.*
- *When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".*
- *After you have set the SCR:TIE bit and SCR:TBIE bit to "0", set this bit to "1". After you have enabled transmit FIFO, set the SCR:TIE bit and SCR:TBIE bit to "1".*

[bit4] FSET: FIFO pointer save bit

This bit saves the transmit FIFO read pointer.

If the read pointer is saved before transmission and if the FLST bit is "0", data can be re-transmitted even when a communication error or others occur.

If set to "1": The current read pointer value is saved.

If set to "0": No effect on the operation.

bit	Description	
	At writing	At reading
0	Not saved	"0" is always read.
1	Saved	

Note:

- *This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".*

[bit3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

When this bit is set to "1", the FIFO2 internal state is initialized.

Only the FCR1:FLST2 bit is initialized, but the other bits of FCR1/FCR0 registers are kept.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	FIFO2 is reset.	

Notes:

- *Disable the transmission and reception first, and then reset FIFO2.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The valid data count of the FBYTE2 register is set to "0".*

[bit2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 value.

When this bit is set to "1", the FIFO1 internal state is initialized.

Only the FCR1:FLST1 bit is initialized, but the other bits of FCR1/FCR0 registers are kept.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	FIFO1 is reset.	

Notes:

- *Disable the transmission and reception first, and then reset FIFO1.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The valid data count of the FBYTE1 register is set to "0".*

[bit1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If FIFO2 is set as transmit FIFO (FCR1:FSEL=1) and if data exists in FIFO2 when this bit is set to "1", the data transmission starts immediately when the UART is enabled to transmit data (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both SCR:TIE bit and SCR:TBIE bit to "1".
- If received FIFO is selected by the FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- If FIFO2 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO2 is used as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and no valid data exists in received FIFO (FBYTE2=0x00) after reception is disabled (SCR:RXE=0).
- If FIFO2 is used as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO2 state is held even if the FIFO2 operation is disabled.

bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

[bit0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- If FIFO1 is set as transmit FIFO (FCR1:FSEL=0) and if data exists in FIFO1 when this bit is set to "1", the data transmission starts immediately when the UART is enabled to transmit data (SCR:TXE=1). During this time, set both SCR:TIE bit and SCR:TBIE bit to "0". Then, set this bit to "1" and set both TIE bit and TBIE bit to "1".
- If received FIFO is selected by the FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- If FIFO1 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (SSR:TDRE=1).
- If FIFO1 is used as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and no valid data exists in received FIFO (FBYTE2=0x00) after reception is disabled (SCR:RXE=0).
- If FIFO1 is used as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO1 state is held even if the FIFO1 operation is disabled.

bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

7.16 FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count of FIFO. The following shows the settings of the FCR1:FSEL bit.

Table 7-5 Display of data count

FCR1:FSEL	FIFO selection	Byte count display
0	FIFO2: Received FIFO, FIFO1: Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2: Transmit FIFO, FIFO1: Received FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to generate a received interrupt flag for the FBYTE register of received FIFO. If this transfer data count matches the FBYTE register display, the received data full flag bit (RDRF) is set to "1".
- If the following two conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the received data full flag bit (RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.

  If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to "0". If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.
- To receive data in the master mode operation (master mode reception), set both SCR:TIE and SCR:TBIE bits to "0", set the received data count in the FBYTE register of transmit FIFO, and set the FCR1:FDRQ bit to "0". After that, when the SCR:TXE bit is "1", the serial clock is output for the preset data amount, and the preset amount of data can be received. Set the SCR:TIE bit and SCR:TBIE bit to "1" only after the FCR1:FDRQ bit has been set to "1".

[bit15:8] FBYTE2: FIFO2 data count display bits

[bit7:0] FBYTE1: FIFO1 data count display bits

Writing	Sets the transfer data count.
Reading	Reads the effective count of data.

Read (Effective data count)

During transmission : The number of data sets already written in FIFO but not transmitted yet
During reception : The number of data sets received in FIFO

Write (Transfer data count)

During transmission : Set "0x00".
During reception : Set the data count to generate a received interrupt.

Table 7-6 Data Count to be Saved in FIFO

FIFO Capacity	Data Length	Max. FBYTE count	Count of Data to be saved
16 BYTEs	5 bits to 16 bits	8	8
32 BYTEs	5 bits to 16 bits	16	16
64BYTEs	5 bits to 16 bits	32	32
64 BYTEs	5 bits to 16 bits	64	64

Notes:

- *The FBYTE register of transmit FIFO must be "0x00" except when data is received in the master mode operation.*
- *During the master mode data reception, the transmit data count must be set only when transmit FIFO is empty and both SCR:TIE bit and SSR:TBIE bit are "0".*
- *To disable the reception (SCR:RXE=0) when data is being received in the master mode operation, disable transmit FIFO first, and then disable the transmission and reception.*
- *The FBYTE bit of received FIFO must be set to "1" or larger.*
- *Change the FBYTE data of received FIFO only after you have disabled the data reception.*
- *A read-modify-write instruction cannot be used for this register.*
- *Any setting exceeding the FIFO capacity is prohibited.*

8. Restrictions on CSIO (Clock Synchronous Serial Interface)

This section shows the restrictions on CSIO (Clock Synchronous Serial Interface).

- When the Chip Select is used in normal transmit mode (SCR:SPI="0") and master mode (SCR:MS="0"), set the setup hold delay to meet the one of the following conditions:
  - Hold Delay + Setup Delay < Baud rate conversion value – 2 × t_{CYCP}
  - Baud rate conversion value/2 < Hold Delay + 3 × t_{CYCP}

Baud rate conversion value: Inverse number of Baud rate (Definition)
t_{CYCP} : APB Bus clock frequency

<Calculation Example>

When Baud rate: 1 [Mbps] (Baud rate conversion value: 1 [µs]), Peripheral bus clock: 48 [MHz] (Cycle: about 20 [ns]) and SCSCR:CDIV=”0”, Hold Delay and Setup Delay conditions are calculated as follows:

- Hold Delay:

  SCSTR:CSHD value $\times t_{CYCP} \times 2^{\text{SCSCR:CDIV Value}}$ = SCSTR:CSHD Value × 20[ns]

- Setup Delay:

  SCSTR:CSSU value $\times t_{CYCP} \times 2^{\text{SCSCR:CDIV Value}}$ = SCSTR:CSSU Value × 20[ns]

From the above condition formulas, set SCSTR:CSHD Value and SCSTR:CSSU Value conforming to the combination in Table 8-1.

Table 8-1 Setting Conditions of Hold Delay and Setup Delay (Calculation Example)

SCSTR:CSHD Value	SCSTR:CSSU Value
23 or more	Arbitrary value
22	25 or less
21	26 or less
20	27 or less
:	:
1	46 or less
0	47 or less

In master mode (SCR:MS=0) and SPI Transfer mode (SCR:SPI=1), when transfer data count is “1” (TBYTE=1) is set and Serial Chip Select Hold Function is used, use CSIO under the following condition:

- Set “No Serial Data Transmit and Reception Wait” (ESCR:WT1, WT0 ="00")

CHAPTER 1-4: LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1)

This chapter explains the LIN communication function, a part of multifunction serial interface functions and supported in Operation Mode 3.

1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1)
2. LIN Interface (Ver. 2.1) Interrupts
3. Dedicated Baud Rate Generator
4. LIN Interface (Ver. 2.1) Operations
5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow
6. LIN Interface (ver. 2.1) Registers

CODE: 9BFLIN_FM4-E01.0_FM15L-J05.4

1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1)

The LIN interface (ver. 2.1) (LIN communication control interface ver. 2.1) supports functions complying with the LIN bus. It also has transmit/received FIFO (up to 128 bytes) installed.

Functions of LIN interface (ver. 2.1) (LIN communication control interface ver. 2.1)

		Function
1	Data buffer	− Full duplex double buffer (when FIFO is not used) − Transmit/received FIFO (max 128 bytes) * (when FIFO is used)
2	Serial input	− Run oversampling three times with the bus clock and determine the value of received data based on the majority sampling value.
3	Transfer mode	Asynchronous
4	Baud rate	− A dedicated baud rate generator (constructed with a 15-bit reload counter) − The external clock can be adjusted with the reload counter.
5	Data length	8 bits
6	Signaling system	NRZ (Non Return to Zero)
7	Start bit detection	Synchronized with the falling edge of the start bit
8	Received error detection	− Framing error − Overrun error
9	Interrupt request	− Received interrupts (reception completed, framing error, overrun error) − Transmit interrupts (transmit data empty, transmit bus idle) − Status interrupts (LIN break field detection) − Interrupt request to ICU (LIN Sync field detection: LSYN) − Transmit FIFO interrupt (when transmit FIFO is empty) − DMA (Transmit/Received) transferring support function is available.
10	LIN bus option	− Supports LIN Protocol Revision 2.1 − Master device operations − Slave device operations − LIN break field generation (with variable bit length ranging from 13 to 16 bits) − LIN break delimiter generation (with variable data length ranging from 1 to 4 bits) − LIN break field detection − Detection of LIN sync field start/stop edges connected to input capture
11	FIFO options	− Transmit/received FIFO installed (maximum capacity: 128 bytes for transmit FIFO, 128 bytes for received FIFO) * − Transmit FIFO or received FIFO can be selected. − Transmit data can be resent. − Received FIFO interrupt timing can be changed via software. − FIFO resetting is supported independently.

* : The FIFO quantity varies depending on the products type.

2. LIN Interface (Ver. 2.1) Interrupts

Received interrupts and transmit interrupts are provided for LIN interface (ver. 2.1). These interrupt requests can be generated if:

- Received data is set in the Received Data Register (RDR) or a data received error occurs.
- Transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register and the data transmission is started.
- The transmit bus is idle (No data transmission occurs).
- Transmit FIFO data is requested.
- A LIN break field is detected.

LIN interface (ver. 2.1) interrupts

Table 2-1 shows the interrupt control bits and the interrupt factors of LIN interface (ver. 2.1).

Table 2-1 LIN interface (ver. 2.1) interrupt control bits and interrupt factors

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Operation to clear interrupt request flag
Reception	RDRF	SSR	A single-byte reception	SCR:RIE	Reading from the received data register (RDR)
			Reception of a data volume matching the value set for FBYTE.		Reading from the Received Data Register (RDR) until received FIFO is emptied
			While the FRIIE bit is "1" and the received FIFO contains valid data, a received idle state continues for 8 bits or longer period.		
	ORE	SSR	Overrun error		Setting the Reception Error Flag Clear bit (SSR:REC) to "1"
	FRE	SSR	Framing error		
Transmission	TDRE	SSR	The Transmit Data Register is empty	SCR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *1
	TBI	SSR	No data transmission	SCR:TBIE	Writing to the Transmit Data Register (TDR), setting the LIN break field setting bit (LBR) to "1", or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data).*1
	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit (FCR1:FDRQ) is set to "0" or transmit FIFO is full.
Status	LBD	SSR	LIN break field is detected	ESCR:LBIE	The SSR:LBD bit is set to "0".
Input capture*2	ICP0/ ICP1	ICSA10/ICSA32	The first rising edge in the LIN Sync field	ICSA10.ICE0 ICSA10.ICE1 ICSA32.ICE0 ICSA32.ICE1	Disables ICP0 and ICP1
	ICP0/ ICP1	ICSA10/ICSA32	The fifth falling edge in the LIN Sync field		

*1: Set the TIE bit to "1" only after the TDRE bit has been set to "0".

*2: For the correspondence between the channel number of Input capture and that of LIN, see the descriptions of EPFR01/EPFR02/EPFR03 register.

2.1 Received interrupt and flag set timing

Data reception can be interrupted by a received completion (SSR:RDRF = 1), a received error occurrence (SSR:ORE, FRE = 1), or a LIN break field detection.

Received interrupt and flag set timing

Upon detection of the first stop bit, received data are stored in the Received Data Register (RDR). When the data reception is completed (SSR:RDRF = 1) or when a data received error occurs (SSR:ORE, FRE = 1), each flag is set. If received interrupts are enabled (SCR:RIE = 1) during this time, a received interrupt occurs.

Note:

- *If a received error occurs, data in the Received Data Register (RDR) is invalidated.*

Figure 2-1 RDRF (Received Data Full flag bit) set timing

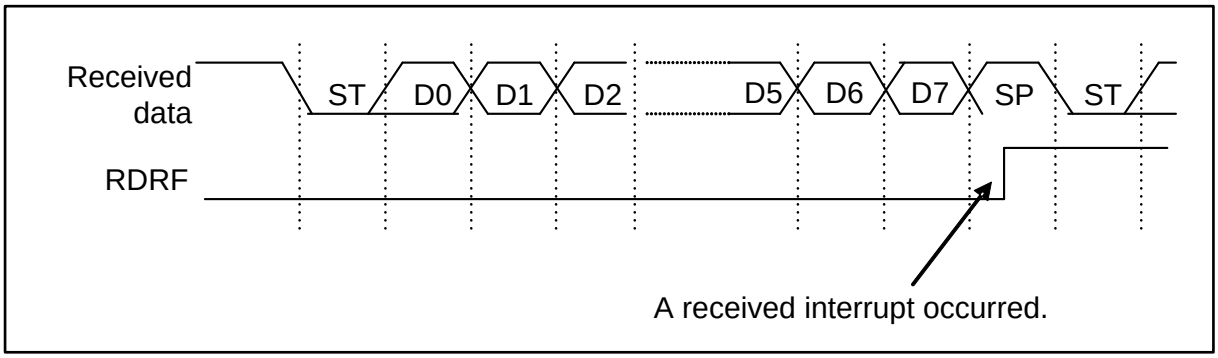
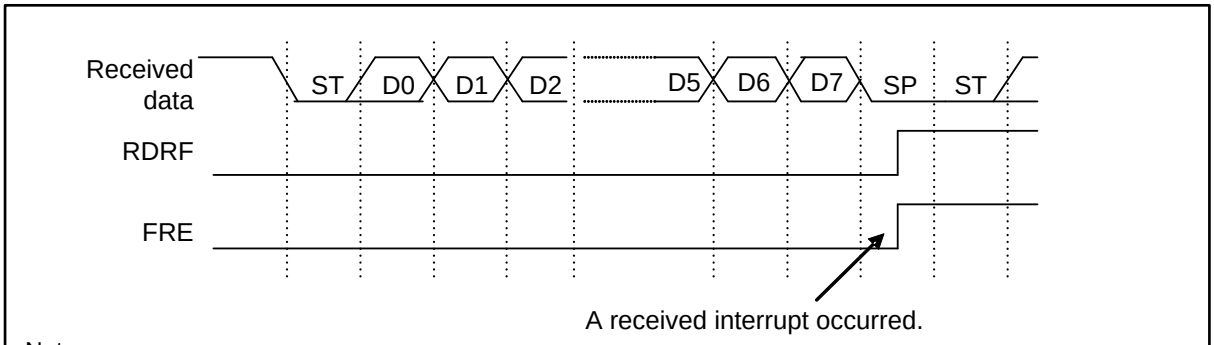


Figure 2-2 FRE (Framing Error flag bit) set timing

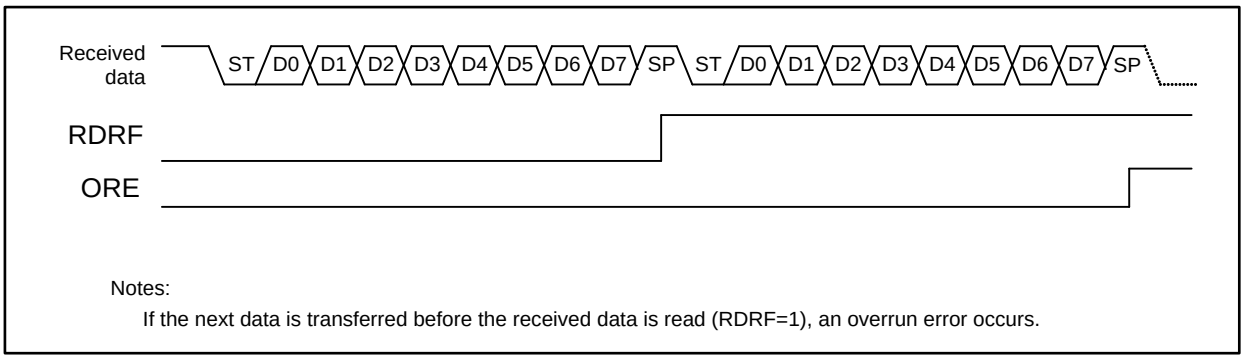


Notes:
- When the first stop bit is at "LOW" level, a framing error occurs.
- The RDRF bit is set to "1" and data can be received even if a framing error has occurred. However, the received data is invalid.

Note:

- *During reception, if a falling edge of the serial data is detected concurrently with, or 1 to 2 bus clocks before the sampling point of the stop bit, the edge is ignored and the next data may not be received successfully. To output frames continuously, adequate intervals are required between frames.*

Figure 2-3 ORE (Overrun Error flag bit) set timing



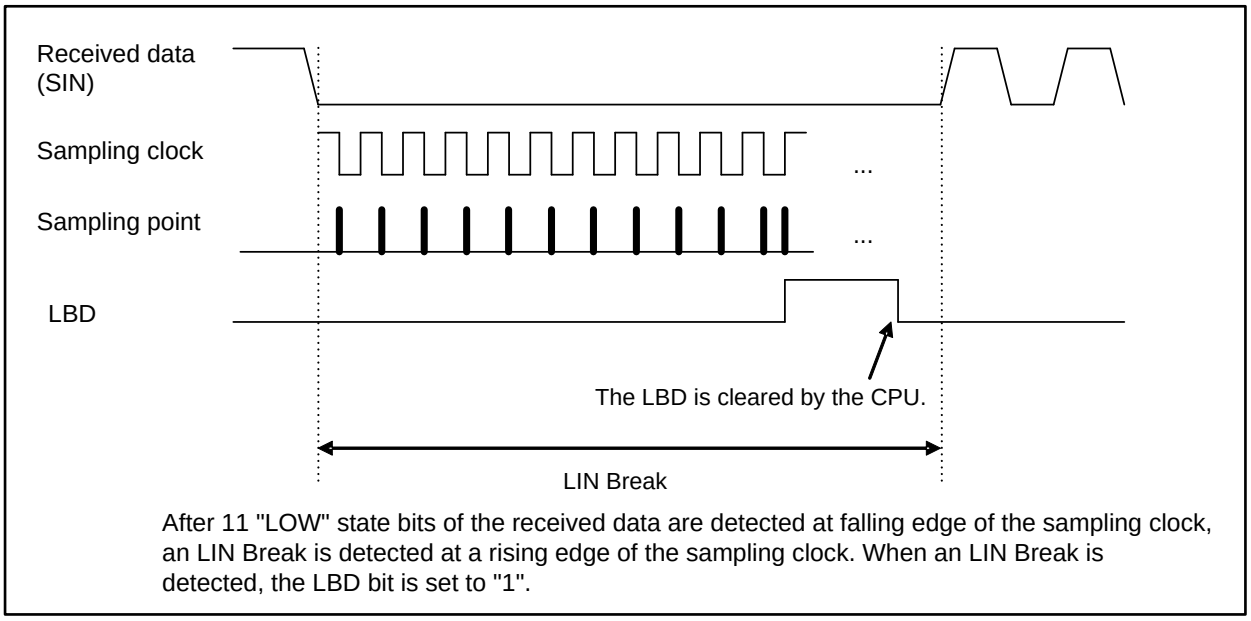
Notes:

If the next data is transferred before the received data is read (RDRF=1), an overrun error occurs.

LIN break field detection flag (LBD) set timing

If "0" is input for a width of 11 bits or more as serial input (SIN), the LBD bit is set to "1". If LIN break field interrupts are enabled (ESCR:LBIE = 1) then, a received interrupt occurs.

Figure 2-4 LBD (LIN Break field Detection flag) set timing



After 11 "LOW" state bits of the received data are detected at falling edge of the sampling clock, an LIN Break is detected at a rising edge of the sampling clock. When an LIN Break is detected, the LBD bit is set to "1".

2.2 Interrupt and flag set timing when received FIFO is used

If received FIFO is used, an interrupt occurs when the FBYTE data (preset for the FBYTE register (FBYTE)) is received.

Received interrupt and flag set timing when received FIFO is used

If the received FIFO is used, an interrupt occurs depending on the value set for the FBYTE register.

- When the amount of data set for transfer count in the FBYTE register is received, the received data full flag (SSR:RDRF) of the Serial Status register is set to "1". If received interrupts are enabled (SCR:RIE) during this time, a received interrupt occurs.
- If both of the following conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the received data full flag (SSR:RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FCR:FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.
    If the RDR data is read during counting of 8 clocks, this counter is reset to 0 and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to "0". If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.
- When the received data (RDR) is all read and received FIFO is emptied, the received data full flag (SSR:RDRF) is cleared.
- If the display of the valid received data amount is the same as the FIFO capacity and if the next data is received, an overrun error (SSR:ORE = 1) occurs.

Figure 2-5 Received interrupt occurrence timing when received FIFO is used

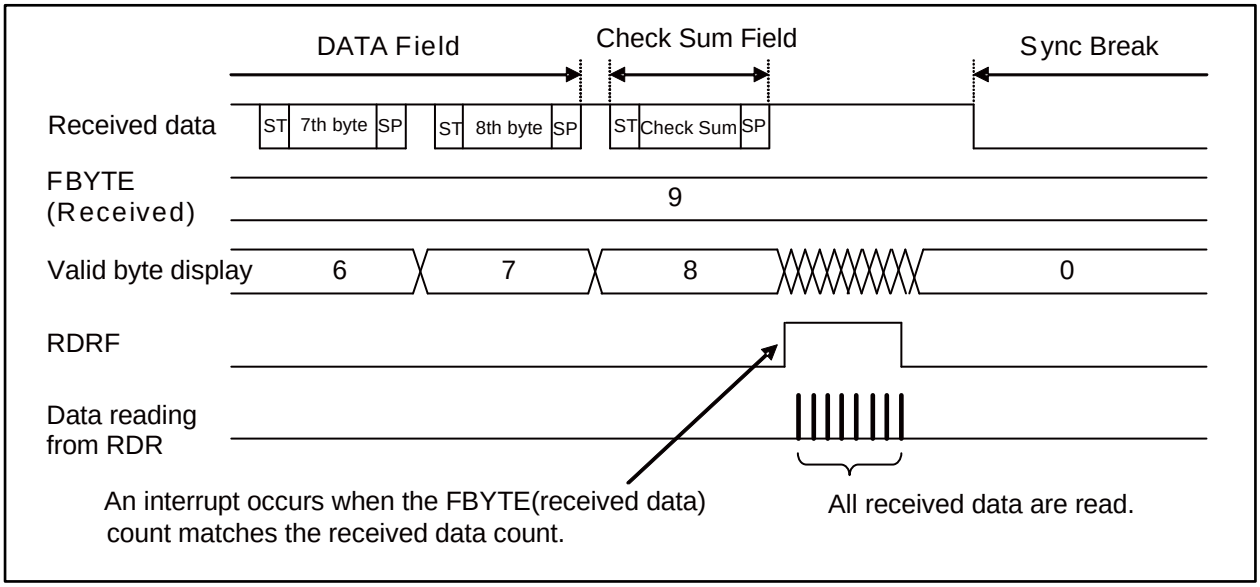
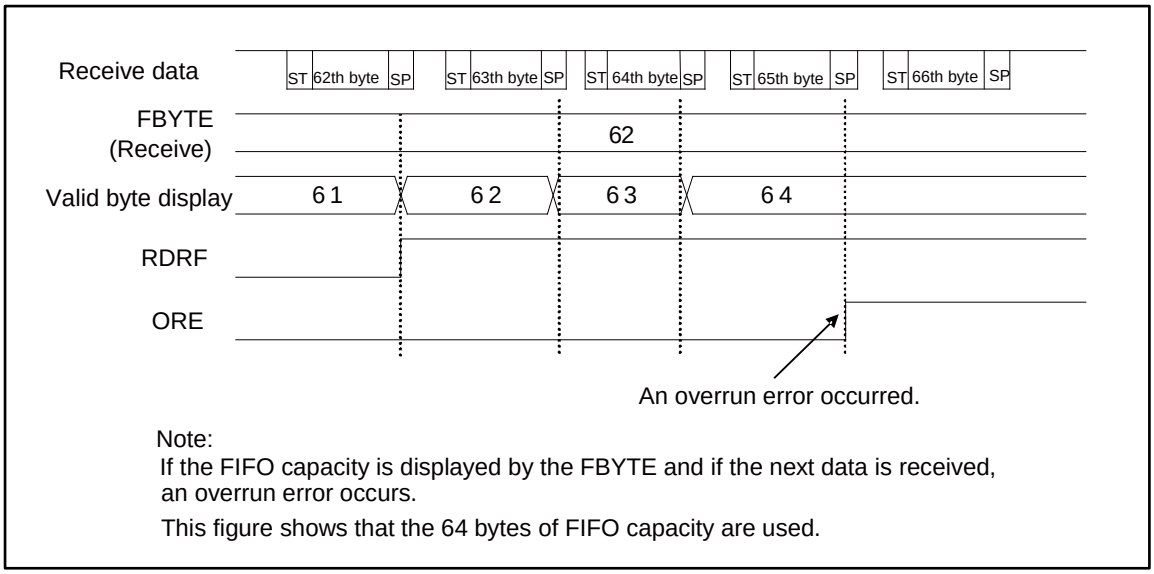


Figure 2-6 ORE (Overrun Error) flag bit set timing



Note:
If the FIFO capacity is displayed by the FBYTE and if the next data is received, an overrun error occurs.

This figure shows that the 64 bytes of FIFO capacity are used.

2.3 Transmit interrupt and flag set timing

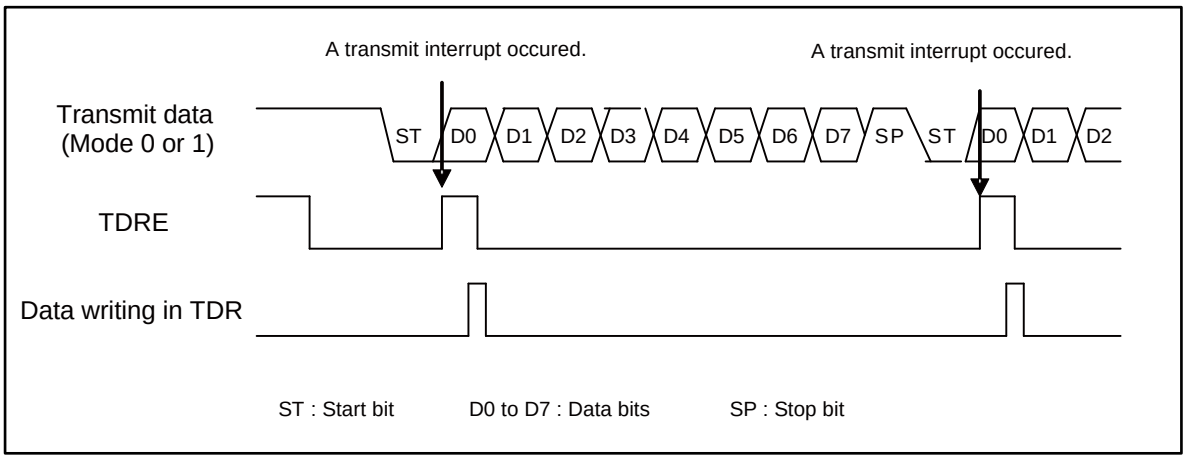
A transmit interrupt occurs when transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE = 1) and transmission starts and when no transmission is performed (SSR:TBI = 1).

Transmit interrupt and flag set timing

■ Transmit data empty flag (TDRE) set timing

After data has been transferred from the Transmit Data Register (TDR) to the transmit shift register, the next data can be written (SSR:TDRE=1). If transmit interrupts are enabled (SCR:TIE=1) during this time, a transmit interrupt occurs. As the TDRE bit is read only, the SSR:TDRE bit is cleared to "0" when data is written to the Transmit Data Register (TDR).

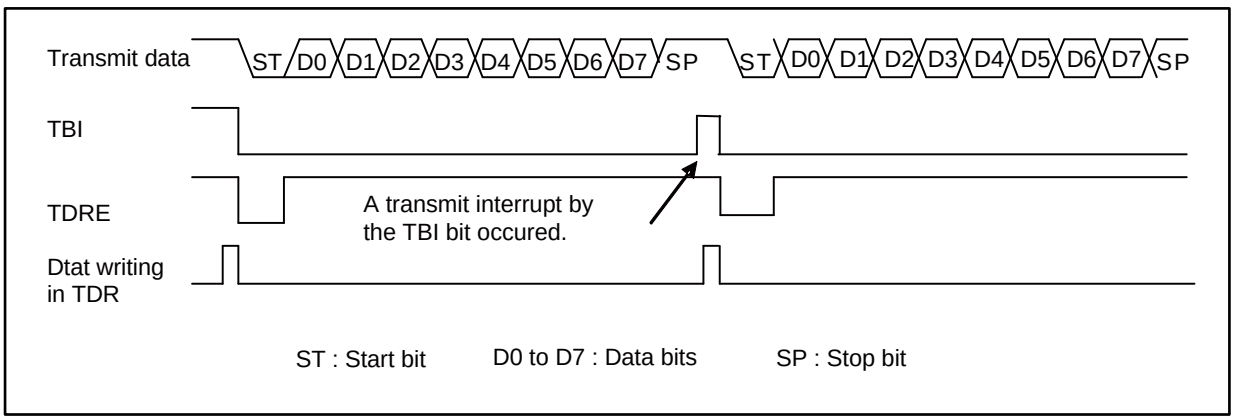
Figure 2-7 Transmit data empty flag (SSR:TDRE) set timing



■ Transmit bus idle flag (TBI) set timing

If the Transmit Data Register is empty (TDRE=1) and no data is transmitted, the SSR:TBI bit is set to "1". If transmit bus idle interrupts are enabled (SCR:TBIE=1) during this time, a transmit interrupt occurs. When transmit data is written to the Transmit Data Register (TDR), both the TBI bit and the transmit interrupt request are cleared.

Figure 2-8 Transmit bus idle flag (TBI) set timing



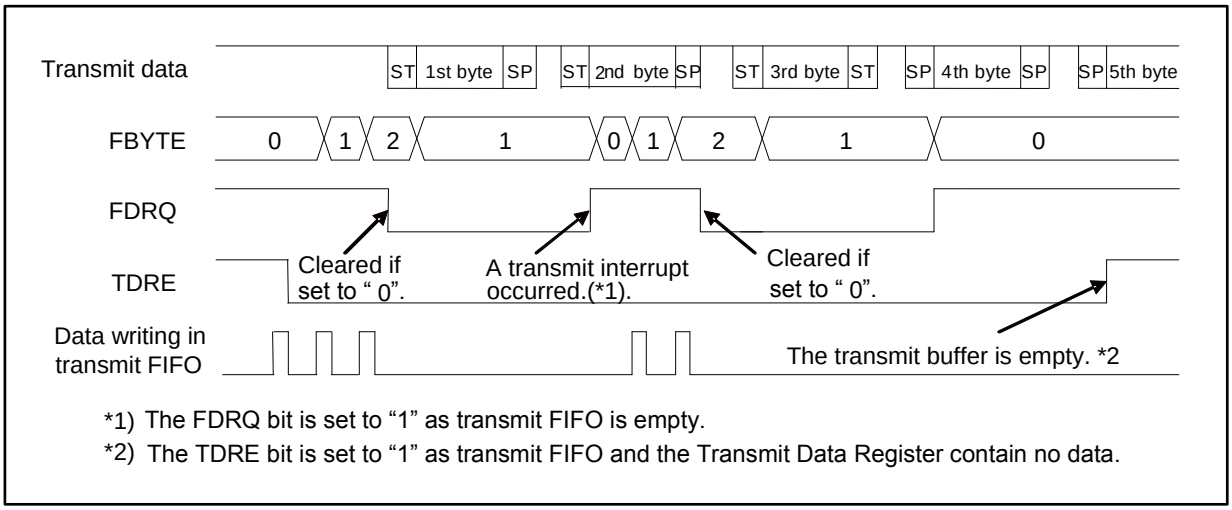
2.4 Interrupt and flag set timing when transmit FIFO is used

When the transmit FIFO is used, an interrupt occurs if the transmit FIFO contains no data.

Transmit interrupt and flag set timing when transmit FIFO is used

- If the transmit FIFO contains no data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "1". If FIFO transmit interrupts are enabled (FCR1:FTIE=1) during this time, a transmit interrupt occurs.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- When transmit FIFO is filled with data, the FIFO transmit data request bit (FCR1:FDRQ) is set to "0".
- To check to see if transmit FIFO contains any data, read from the FIFO Byte Register (FBYTE). If FBYTE=0x00, no data exists in the transmit FIFO.

Figure 2-9 Transmit interrupt occurrence timing when transmit FIFO is used



*1) The FDRQ bit is set to “1” as transmit FIFO is empty.
*2) The TDRE bit is set to “1” as transmit FIFO and the Transmit Data Register contain no data.

3. Dedicated Baud Rate Generator

For the LIN interface (ver. 2.1) transmitting/receiving clock source, either of the following can be selected.

- Dedicated baud rate generator (reload counter)
- An external clock input to the baud rate generator (reload counter)

LIN interface (ver. 2.1) baud rate

Select one of the following two baud rates.

■ Baud rate obtained by dividing an internal clock using the dedicated baud rate generator (reload counter)

This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an internal clock by the set value.

To set the clock source, select an internal clock (SMR:EXT = 0).

■ Baud rate obtained by dividing an external clock using the dedicated baud rate generator (reload counter)

Use an external clock for the clock source of the reload counter.

To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an external clock by the set value.

To set the clock source, select use of an external clock and the baud rate generator clock (SMR:EXT = 1).

This mode is designed for cases where an oscillator with a divided non-standard frequency is used.

Notes:

- *Set the external clock (EXT = 1) while the reload counter is stopped (BGR1/BGR0 = 15h00).*
- *If an external clock is selected (EXT = 1), its HIGH and LOW signals must have a width at least of two bus clocks.*

3.1 Baud rate settings

The following explains how to set the baud rate, and also a result of serial clock frequency calculation.

Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$V = \phi / b - 1$

V : Reload value b: Baud rate ϕ: Bus clock frequency or external clock frequency

(2) Calculation example

To set the 16 MHz bus clock, use the internal clock, and set the 19200 bps baud rate, set the reload value as follows:

Reload value:

$V = (16 \times 1000000) / 19200 - 1 = 832$

Therefore, the baud rate is:

$b = (16 \times 1000000) / (832 + 1) = 19208$ bps

(3) Baud rate error

The baud rate error can be obtained from the following equation.

Error (%) = (Calculated value – Target value) / Target value 100

Example: To set the 20 MHz bus clock and 153600 bps target baud rate:

Reload value =:$(20 \times 1000000) / (129 + 1)$

Buad rate (Calculated value) = $(20 \times 1000000) / (129 + 1) = 153846$ (bps)

Error (%) = $(153846 - 153600) / 153600 \times 100 = 0.16$ (%):

Notes:

- *If the reload value is set to "0", the reload counter is stopped.*
- *If the reload value is even, the "LOW" signal width of serial clock is longer than the "HIGH" signal width for a single cycle of bus clock. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.*
- *Set the reload value to 3 or more. Note that data may not be received normally due to the baud rate error and reload value setting.*

Reload value and baud rate for each bus clock frequency

Table 3-1 Reload values and baud rates

Baud rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
(bps)	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	7	0	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	138	-0.08
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.16	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	0.06	554	-0.01	693	0.06	832	0.04	1110	0.01
19200	416	-0.08	520	-0.03	832	0.04	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	<0.01	1666	-0.02	2082	0.01	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	0.01	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8332	<0.01	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

Value : BGR1/0 register set value

ERR : Baud rate error (%)

Table 3-2 Reload values and baud rates (Continued)

Baud rate (bps)	40 MHz		48 MHz		72 MHz		80 MHz		100 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	4	0	5	0	8	0	9	0	-	-
6M	-	-	7	0	11	0	-	-	-	-
5M	7	0	-	-	-	-	15	0	19	0
4M	9	0	11	0	17	0	19	0	24	0
2.5M	15	0	-	-	-	-	31	0	39	0
1M	39	0	47	0	71	0	79	0	49	0
500000	79	0	95	0	143	0	159	0	99	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22	199	0
250000	159	0	191	0	287	0	319	0	216	<0.01
230400	173	-0.22	207	0.16	312	-0.16	346	0.06	399	0
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03	433	<0.01
125000	319	0	383	0	575	0	639	0	650	<0.01
76800	520	-0.03	624	0	937	-0.05	1041	-0.03	799	0
57600	693	0.06	832	0.04	1249	0	1388	<0.01	867	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01	1301	<0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01	1735	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01	2603	<0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	0	3471	<0.01
9600	4166	<0.01	4999	0	7499	0	8332	0	5207	<0.01
7200	5555	<0.01	6666	<0.01	9999	0	11110	0	9599	<0.01
4800	8332	<0.01	9999	0	14999	0	16666	0	10416	0
2400	16666	<0.01	19999	0	29999	0	-	-	13888	0
1200	-	-	-	-	-	-	-	-	20832	0
600	-	-	-	-	-	-	-	-	-	-
300	-	-	-	-	-	-	-	-	-	-

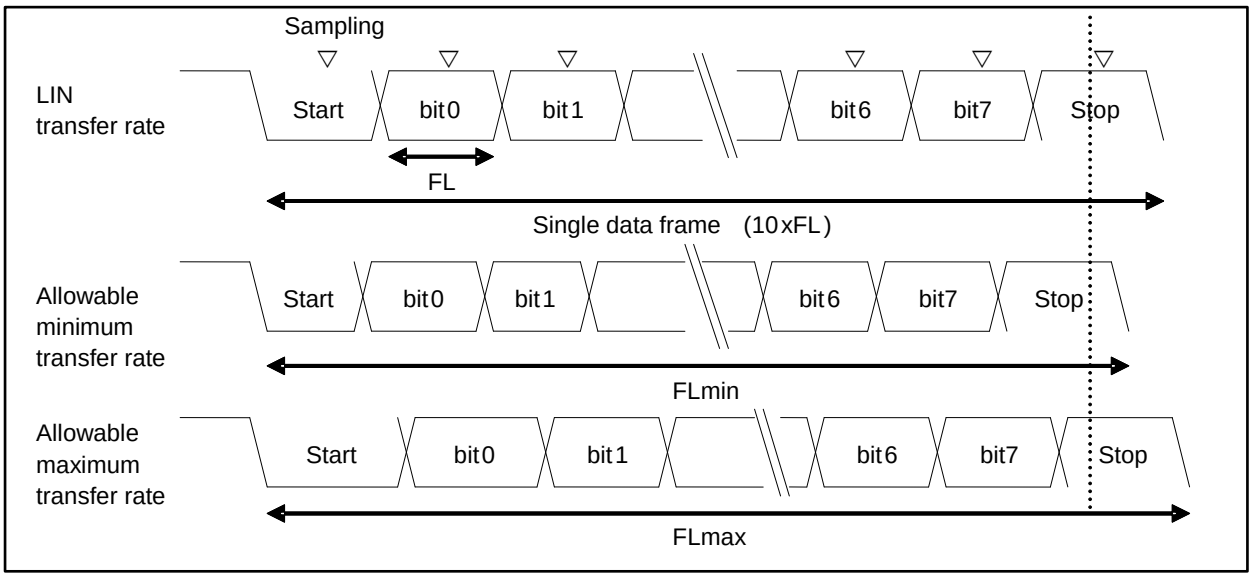
For frequencies not described in Table 3-1 and Table 3-2, calculate them by using formulas in “3.1 Baud rate settings”. (However, for the maximum frequencies, see “Data Sheet” of the product used because they are differed by products)

Allowable baud rate range for data reception

The following shows the range of baud rate error allowed for the destination to receive data.

Set the reception baud rate error by using the following formulas to ensure that the value falls within the allowable range.

Figure 3-1 Allowable baud rate range for data reception



As shown in Figure 3-1, after detection of the start bit, the sampling timing of received data is determined by the counter set in the BGR1/BGR0 register. Data can be received successfully if the last data including the stop bit matches the sampling timing.

If this applies to a reception of 10 bits, a theoretical explanation can be given in the following.

Assuming that the sampling timing margin is one bus clock (ϕ), the minimum allowable transfer rate (FLmin) is determined as follows:

FLmin = (10bit × (V+1) – (V+1)/2 + 2)/ ϕ = (19V + 23)/2 ϕ (s) V: Reload value, ϕ: Bus clock

Thus, the maximum baud rate that allows the destination to receive data (BGmax) is determined as follows.

BGmax = 10/FLmin = 20ϕ/(19V+23) (bps) V: Reload value, ϕ: Bus clock

When data is received at the maximum allowable transfer rate (FLmax), the starting point of the received data 10th bit is sampled.

Thus, the maximum allowable transfer rate (FLmax) is determined as follows:

9/10×FLmax = (10bit × (V+1) – (V+1)/2)/ ϕ V: Reload value, ϕ: Bus clock

FLmax = (19/18 × 10 × (V+1))/ϕ

Assuming that the sampling timing margin (ϕ) is two clocks, the maximum allowable transfer rate (FLmax) is determined as follows:

9/10×FLmax = (10bit × (V+1) – (V+1)/2 – 2)/ ϕ V: Reload value, ϕ: Bus clock

FLmax = (19/18 × 10 × (V+1) – 40/18)/ ϕ = (190V + 150)/18 ϕ (s) V: Reload value, ϕ: Bus clock

Accordingly, the minimum baud rate that allows the destination to receive data (BGmin) is determined as follows:

BGmin = 10/FLmax = 18ϕ/(19V+15) (bps) V: Reload value, ϕ: Bus clock

From the above formulas that yields the minimum/maximum baud rates, the allowable baud rate errors between the LIN interface (ver. 2.1) and the destination can be obtained as shown in the following table.

Reload value (V)	Maximum allowable baud rate error	Minimum allowable baud rate error
3	0%	0
10	+3.28%	-3.41%
50	+4.83%	-4.87%
100	+5.04%	-5.07%
200	+5.15%	-5.16%
32767	+5.26%	-5.26%

Note:

- *Reception accuracy depends on the number of bits per frame, bus clock, and reload value. The higher the bus clock and frequency division ratio are, the higher the accuracy becomes.*

External clock

Writing "1" to the EXT bit of the Baud Rate Generator Register (BGR) causes the baud rate generator to divide the external clock's frequency.

Note:

- *The external clock signal is synchronized with the internal clock on the LIN interface (ver. 2.1). Therefore, an external clock that does not allow synchronization causes unstable operation.*

Functions of reload counter

There are two types of reload counters: The transmit reload counter and the received reload counter, both functioning as a dedicated baud rate generator. Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from the external or internal clock.

Starting counting

When the reload value is written to the Baud Rate Generator Register1, 0 (BGR1 or BGR0), the reload counter starts counting.

Restarting

The reload counter restarts counting in the following conditions.

- Common to transmit and received reload counters

  A programmable reset (SCR:UPCL bit)

- Received reload counter

  Detection of the start bit's falling edge in asynchronous mode

4. LIN Interface (Ver. 2.1) Operations

The LIN interface (ver. 2.1) performs bi-directional LIN communication of master and slave.

Master mode operations

■ Selecting master mode

To operate the LIN interface as a master, set the SCR:MS bit to "0".

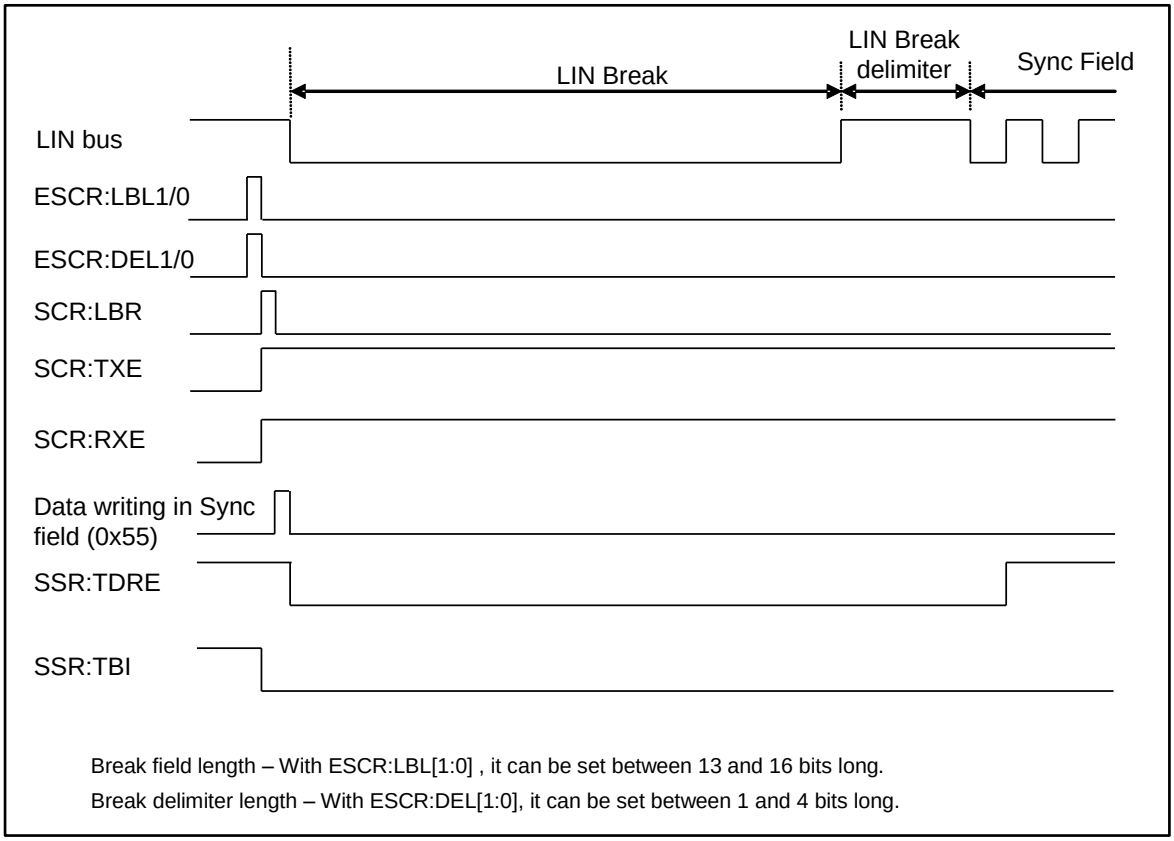
■ Break field transmission-sync field transmission

- The break field length (ESCR:LBL1, LBL0) and the break field delimiter length (ESCR:DEL1, DEL0) can be selected.
- If transmission is enabled (SCR:TXE=1), and the SCR:LBR bit (LIN Break field setting bit) is set to "1", then the break field is transmitted.
- The sync field is transmitted when "0x55" is written to the Transmit Data Register (TDR).

Notes:

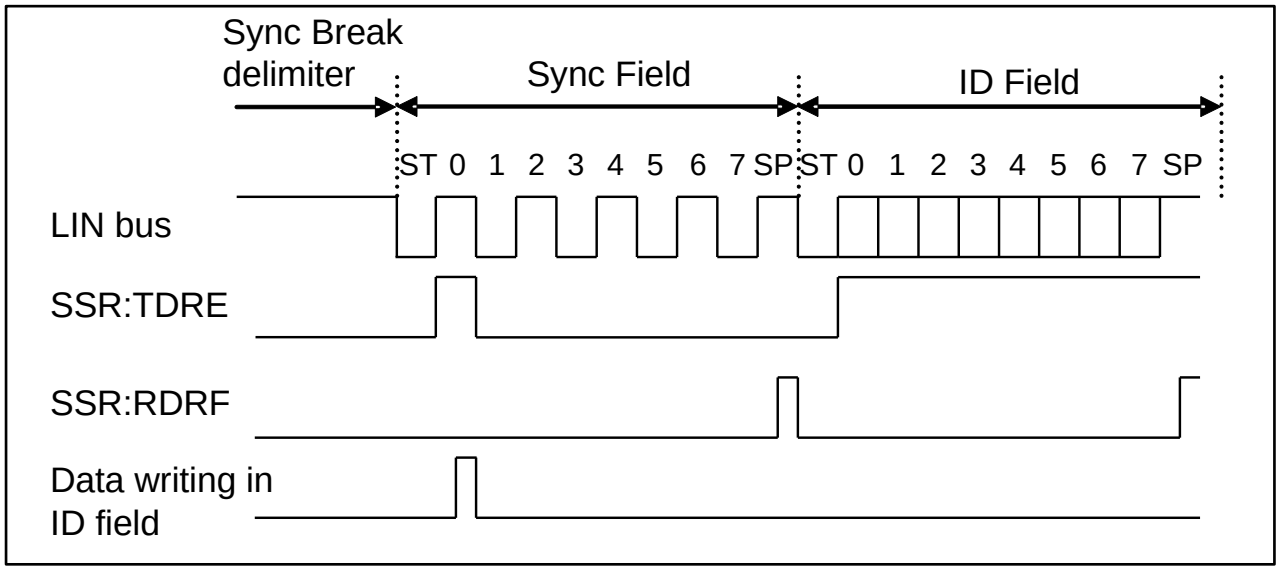
- *Before setting the Transmit Data Register (TDR) to "0x55", set the SCR:LBR bit (LIN break field setting bit) to "1".*
- *Setting the SCR:RXE bit (reception enable bit) to "1" does not enable the Break field to perform reception.*

Figure 4-1 Break field-sync field transmission



■ Sync field transmission - ID field transmission

- When the first bit of the sync field (0x55) is transmitted, the SSR:TDRE (transmit data empty) bit is set to "1".
  If transmit interrupts are enabled (SCR:TIE = 1) during this time, a transmit interrupt occurs.
- If a transmit interrupt occurs, the ID field can be written to the Transmit Data Register (TDR).
- If a received interrupt occurs, compare the received data with the transmit data to make sure that no error has occurred.
- The ID field is output in 8-bit data length and LSB-first order.



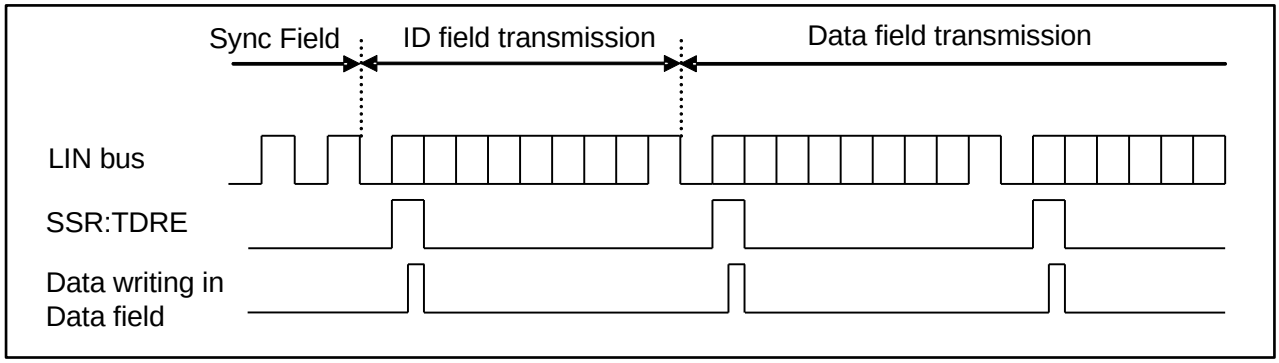
■ ID field transmission - DATA field transmission/reception

Select whether to transmit the DATA field to a slave device or to receive the DATA field.

(To transmit the DATA field)

When the first bit of the ID field is transmitted, the SSR:TDRE bit is set to "1". Then data can be written to the DATA field.

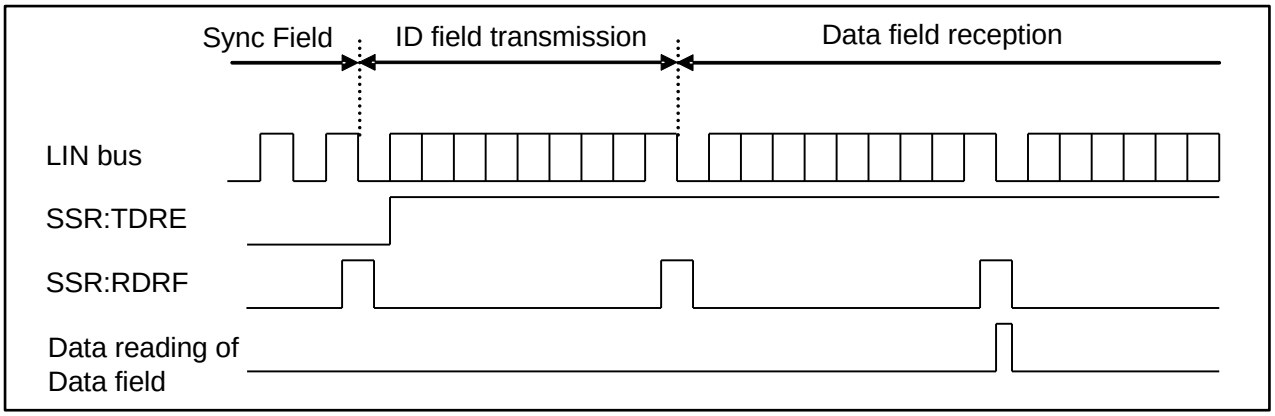
Figure 4-2 ID field transmission-DATA field transmission



(To receive the DATA field)

- When the first bit of the ID field is transmitted, the SSR:TDRE bit is set to "1". However, do not write any transmit data then.
  Also disable transmit interrupts (SCR:TIE = 0).
- When the DATA field is received, SSR:RDRF is set to "1". If received interrupts are enabled (SSR:RIE = 1) then, a received interrupt occurs.
- A start bit is detected when a falling edge is detected after data passes the noise filter (with the majority value applied after sampling serial data input three times with the bus clock) and a LOW level is detected for the data passing the sampling point.

Figure 4-3 ID field transmission - DATA field reception



Notes:

- *The LIN interface (Ver. 2.1) includes noise filter (with the majority value applied after sampling serial data input three times with the bus clock). However, design the board so as not to allow noise to pass through this filter or perform communications so that any noise that has passed does not cause any problems (e.g., by adding a data checksum to the end and resending the data if any error occurs).*
- *During reception, if a falling edge of the serial data is detected concurrently with, or 1 to 2 bus clocks before the sampling point of the stop bit, the edge is ignored and the next data cannot be received successfully. To output frames continuously, adequate intervals should be considered between frames.*

■ Master mode operation timing chart (when FIFO is not used)

Figure 4-4 LIN bus timing (when DATA field is transmitted and FIFO is not used)

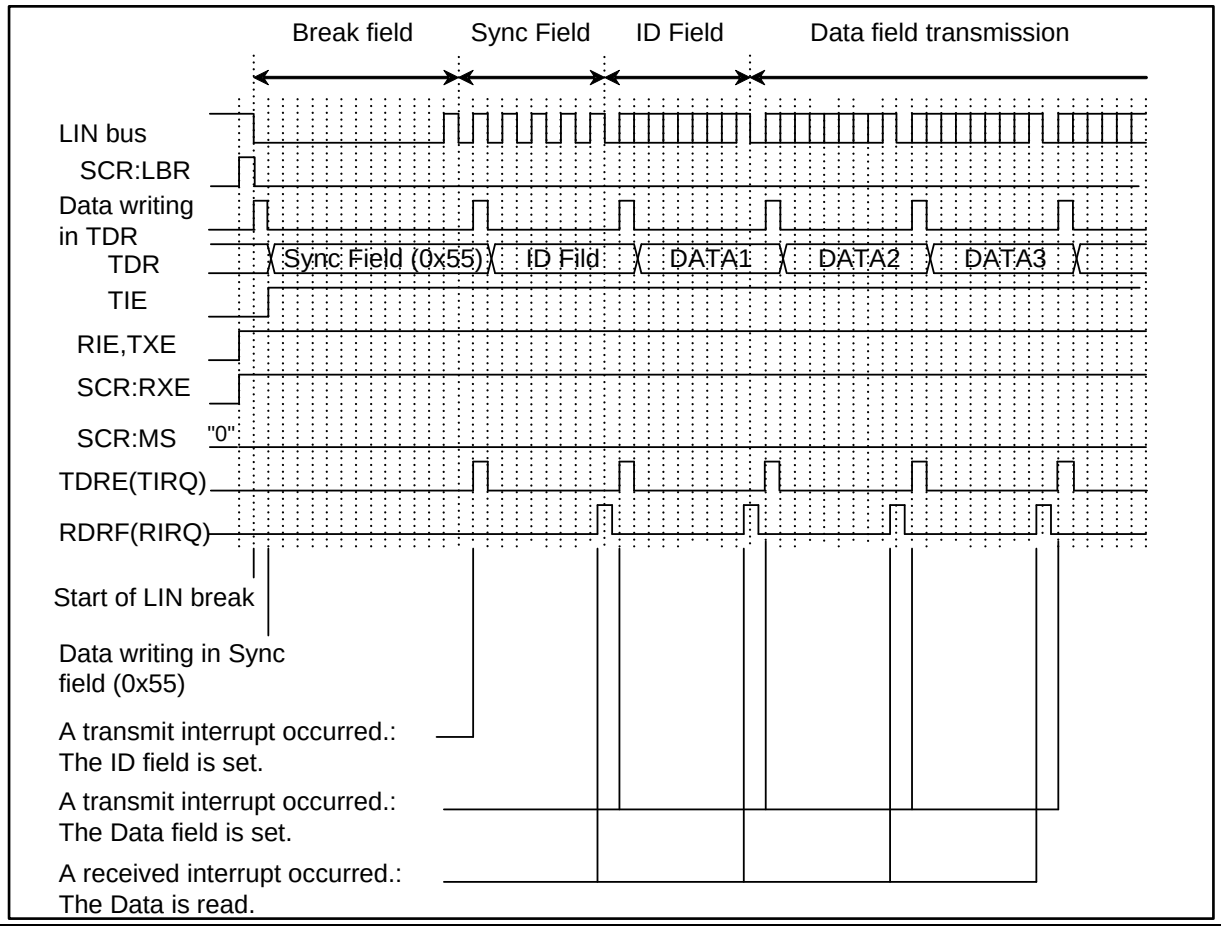
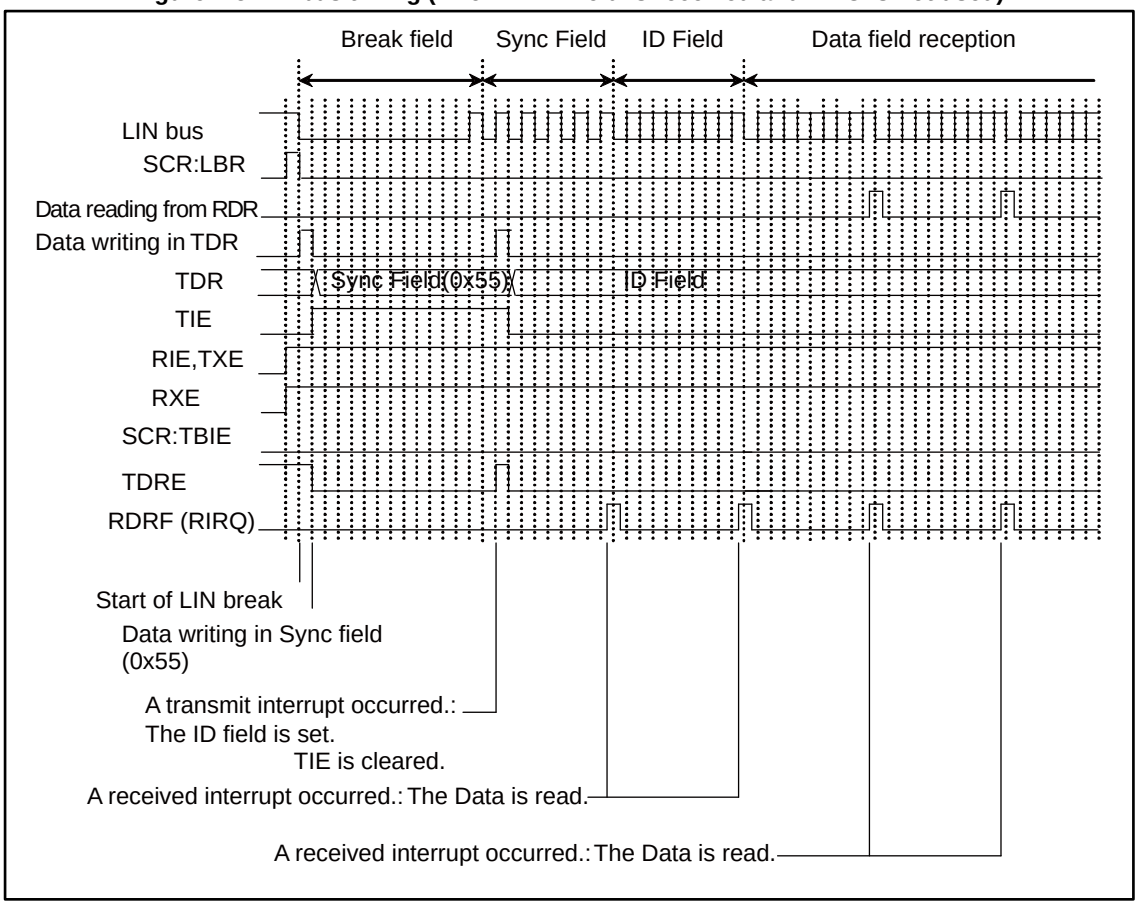


Figure 4-5 LIN bus timing (when DATA field is received and FIFO is not used)



■ Master mode operation timing chart (when FIFO is used)

Figure 4-6 LIN bus timing (when DATA field is transmitted and FIFO is used)

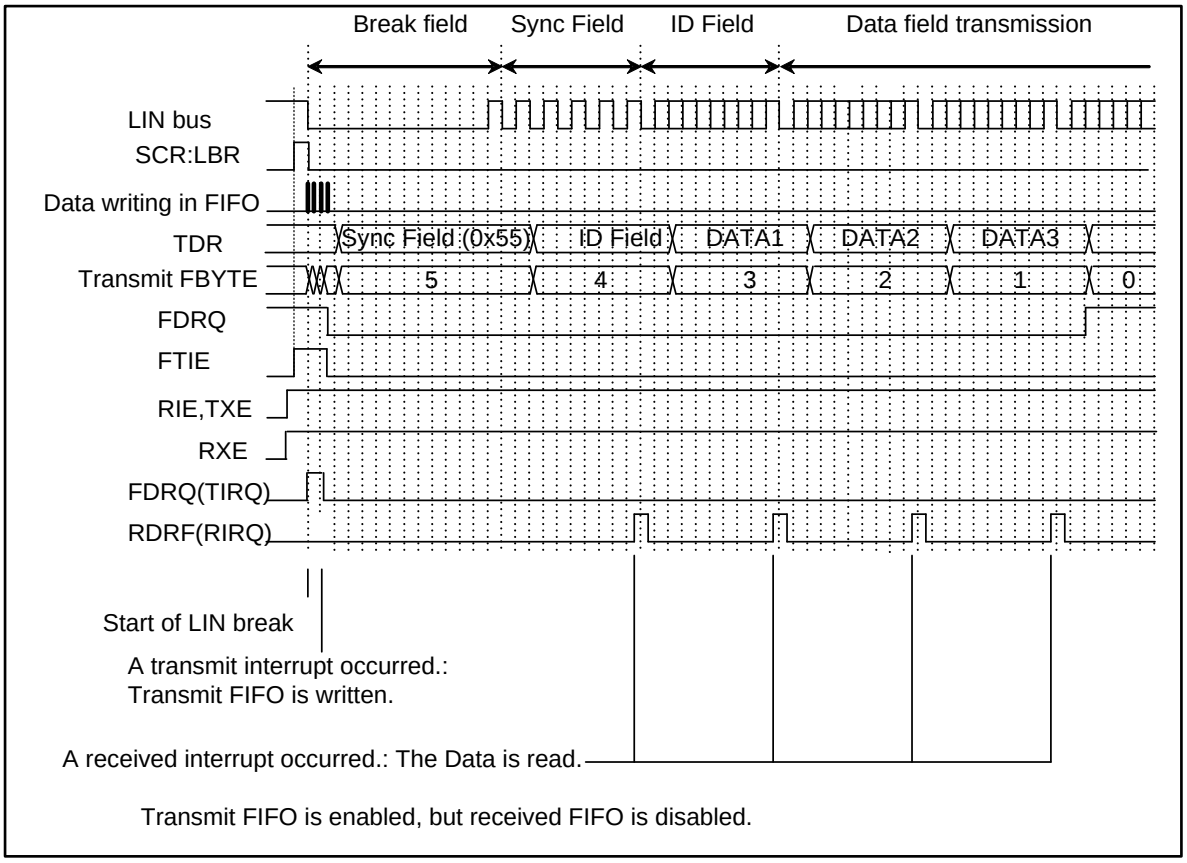
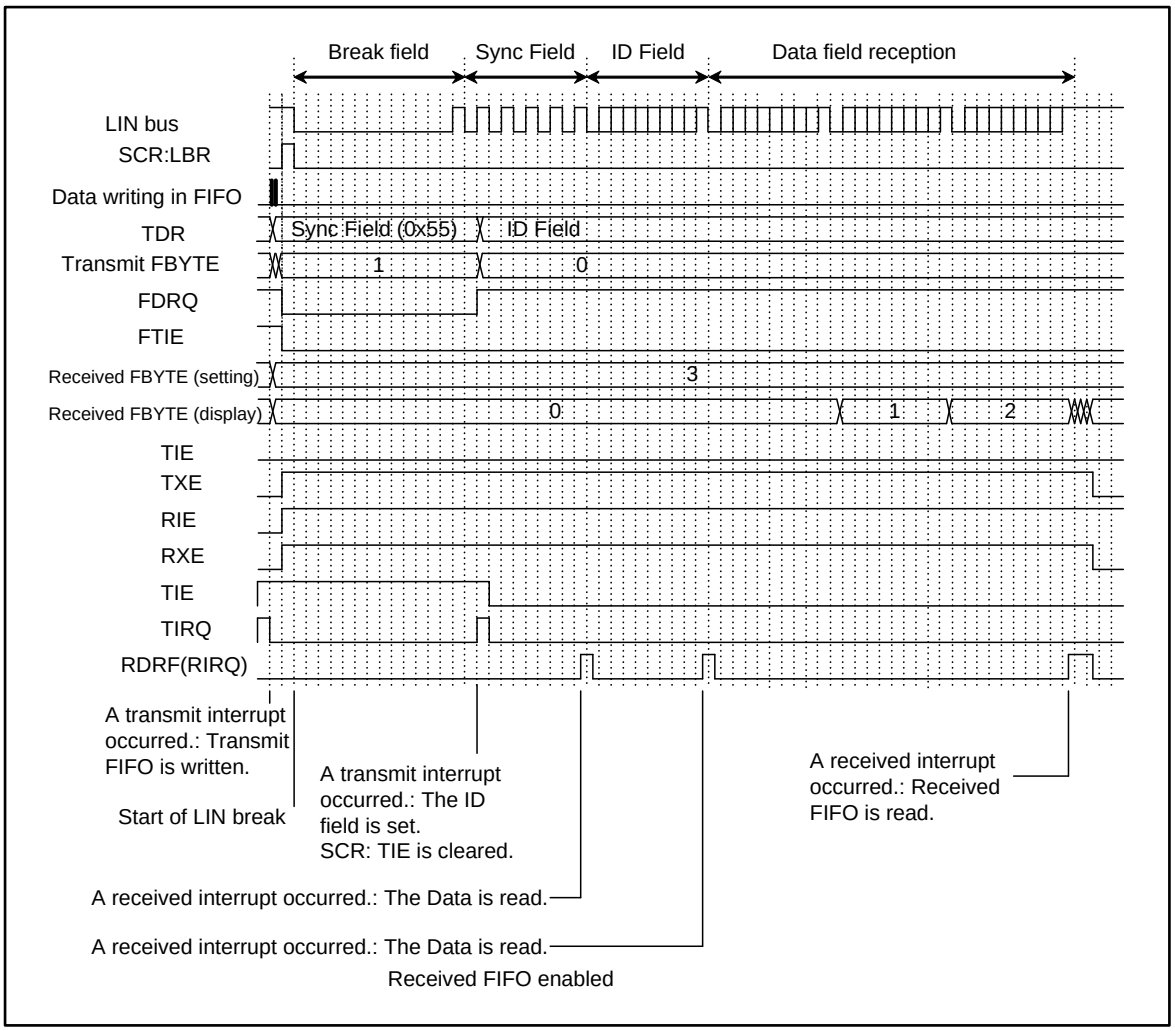


Figure 4-7 LIN bus timing (when DATA field is received and FIFO is used)



Slave mode operations

■ Selecting slave mode

To operate the LIN interface as a slave, set the SCR:MS bit to "1".

■ Break field reception - sync field reception

1. If the break field is input, the break field is detected (SSR:LBD = 1) at the 11th bit.
   If the ESCR:LBIE bit is set to "1" then, a received interrupt occurs.
2. Enable ICU interrupts then to detect both edges.
3. The LIN interface (ver. 2.1), upon the detection of the first falling edge in the sync field, sets the internal signal (LSYN) input to ICU to HIGH to start the ICU. This internal signal (LSYN) turns to LOW at the fifth falling edge.
4. The internal signal (LSYN) input to ICU is a value that the HIGH period multiplies the baud rate by eight. The baud rate set value is obtained as follows:

If the free run timer is not overflowed:

BGR value = (b - a) × Fe/(8 × ϕ) - 1

If the free run timer is overflowed:

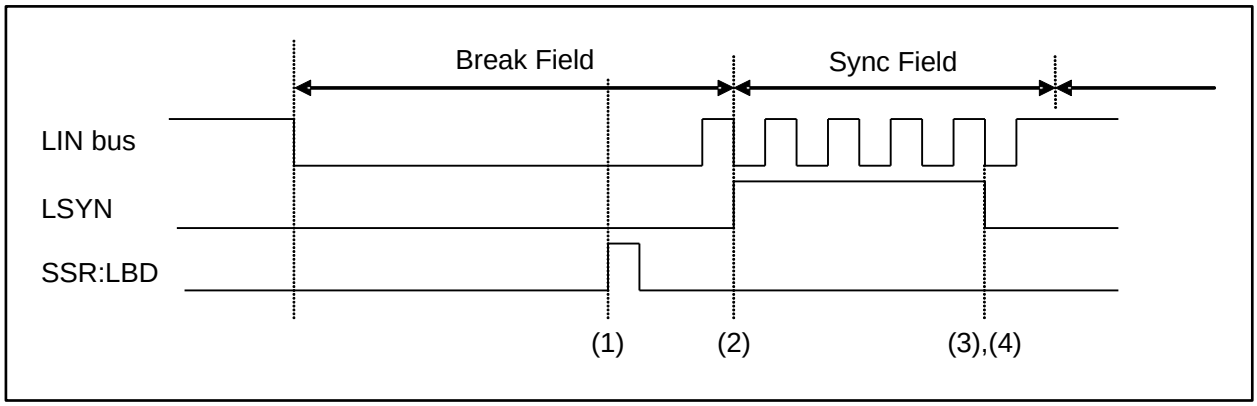
BGR value = (max + 1 + b - a) × Fe/(8 × ϕ) - 1

max	: Maximum value of the free run timer
a	: The ICU data register value after the first interrupt
b	: The ICU data register value after the second interrupt
ϕ	: Bus clock frequency (MHz)
Fe	: External clock frequency (MHz). When the internal clock is used (EXT = 0), Fe = ϕ is assumed.

Note:

- *To operate the break field and the sync field, disable the reception (SCR:RXE = 0).*

Figure 4-8 Break field reception-sync field reception



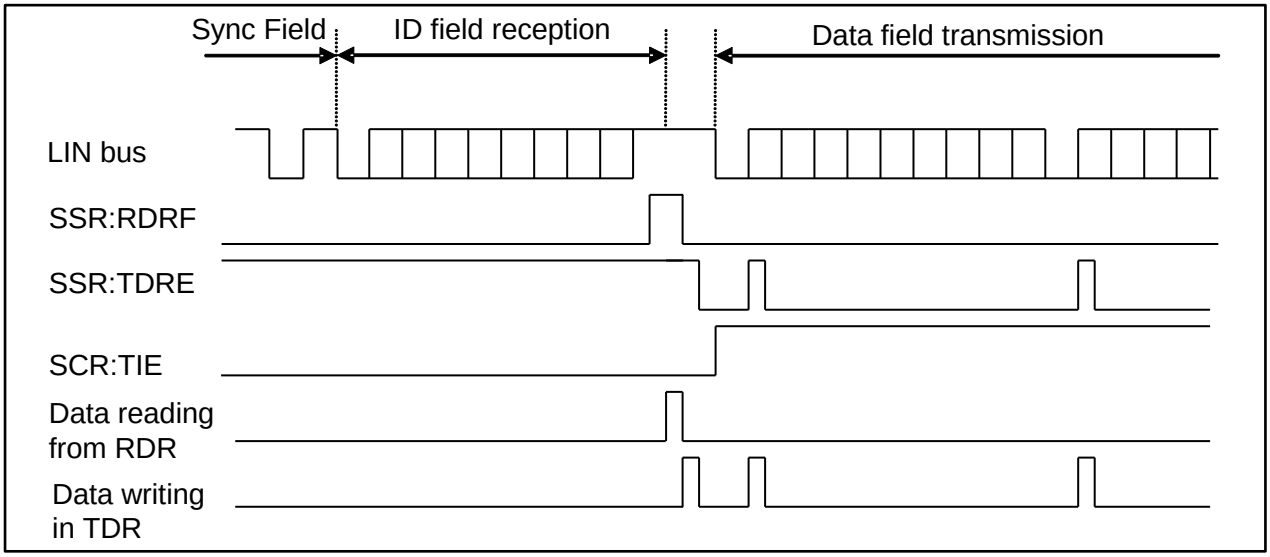
■ ID field reception - DATA field transmission/reception

After reception of the ID field, whether to transmit or to receive the DATA field to master can be selected.

(To transmit the DATA field)

After reception of the ID field, write data to the Transmit Data Register (TDR). Enable transmit interrupts (SCR:TIE = 1) during this time.

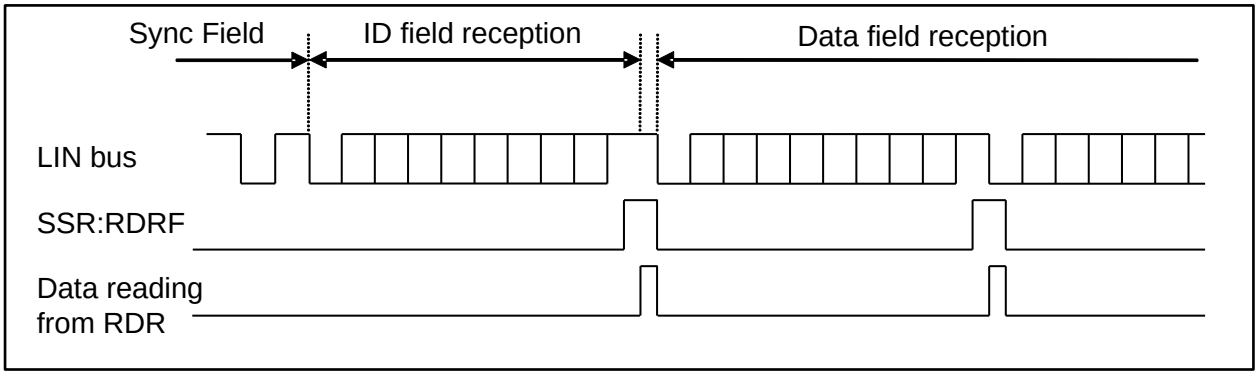
Figure 4-9 ID field reception - DATA field transmission



(To receive the DATA field)

- Every time the DATA field is received, SSR:RDRF is set to "1". If received interrupts are enabled (SCR:RDRF = 1) then, a received interrupt occurs.
- A start bit is detected when a falling edge is detected after data passes the noise filter (with the majority value applied after sampling serial data input three times with the bus clock) and a LOW level is detected for the data passing the sampling point.

Figure 4-10 ID field reception - DATA field reception



Notes:

- *The LIN interface (Ver. 2.1) includes noise filter (with the majority value applied after sampling serial data input three times with the bus clock). However, design the board so as not to allow noise to pass through this filter or perform communications so that any noise that has passed does not cause any problems (e.g., by adding a data checksum to the end and resending the data if any error occurs).*
- *During reception, if a falling edge of the serial data is detected concurrently with, or 1 to 2 bus clocks before the sampling point of the stop bit, the edge is ignored and the next data cannot be received successfully. To output frames continuously, adequate intervals should be considered between frames.*

■ Slave mode operation timing chart

Figure 4-11 LIN bus timing (when DATA field is transmitted and FIFO is not used)

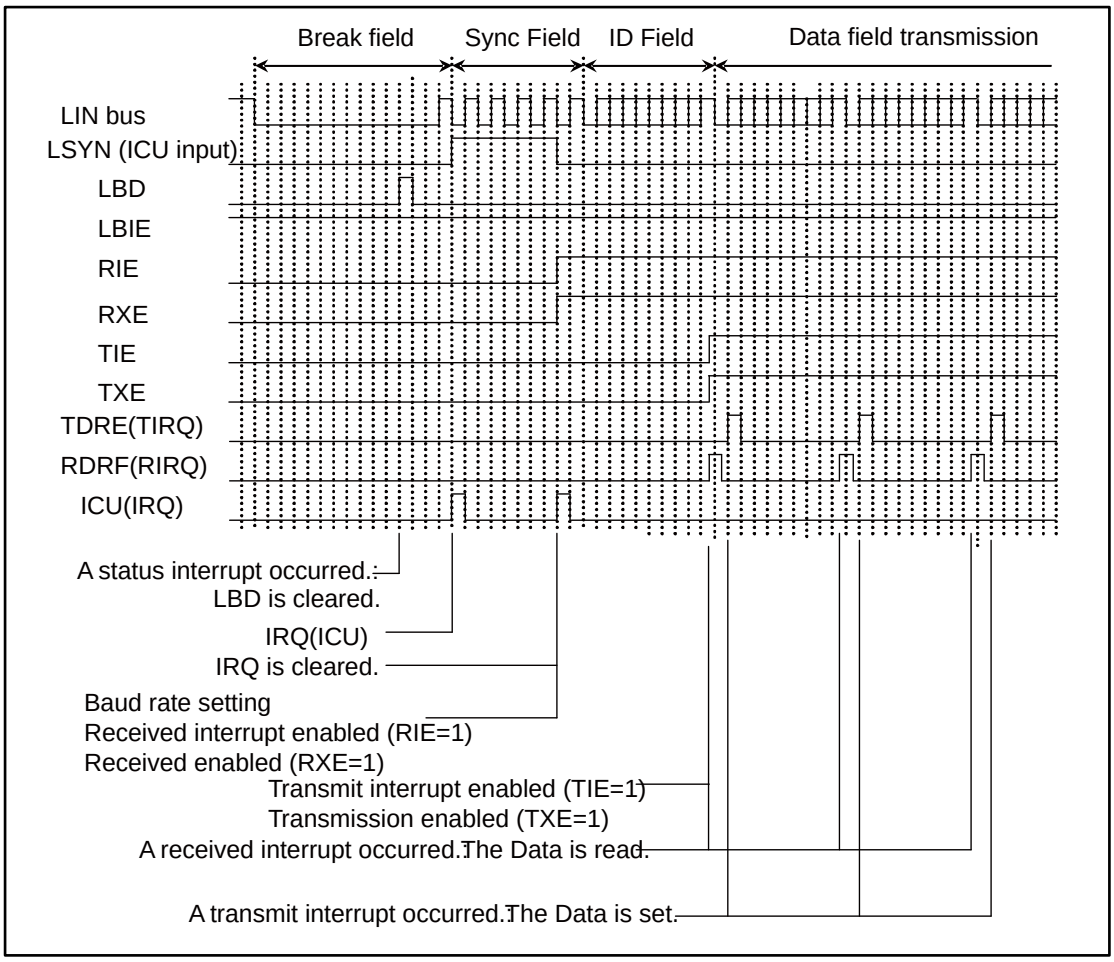
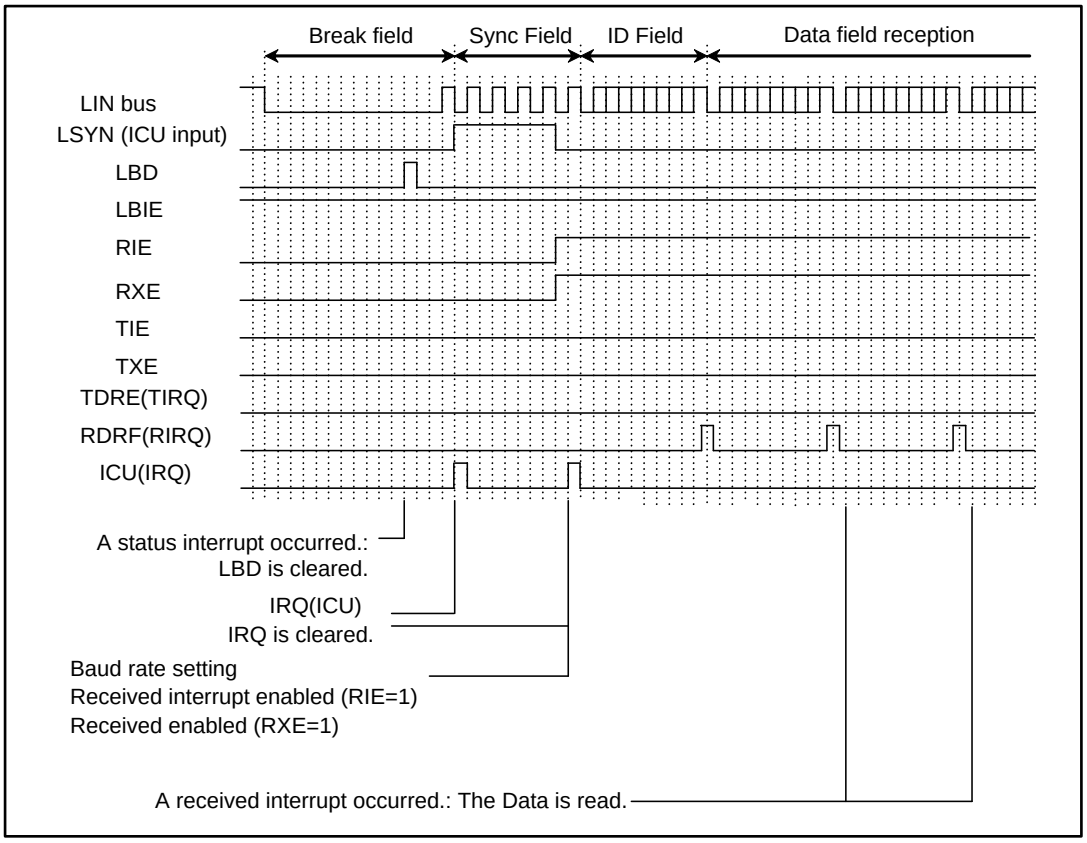


Figure 4-12 LIN bus timing (when DATA field is received and FIFO is not used)



■ If FIFO is used

Figure 4-13 LIN bus timing (when DATA field is transmitted and FIFO is used)

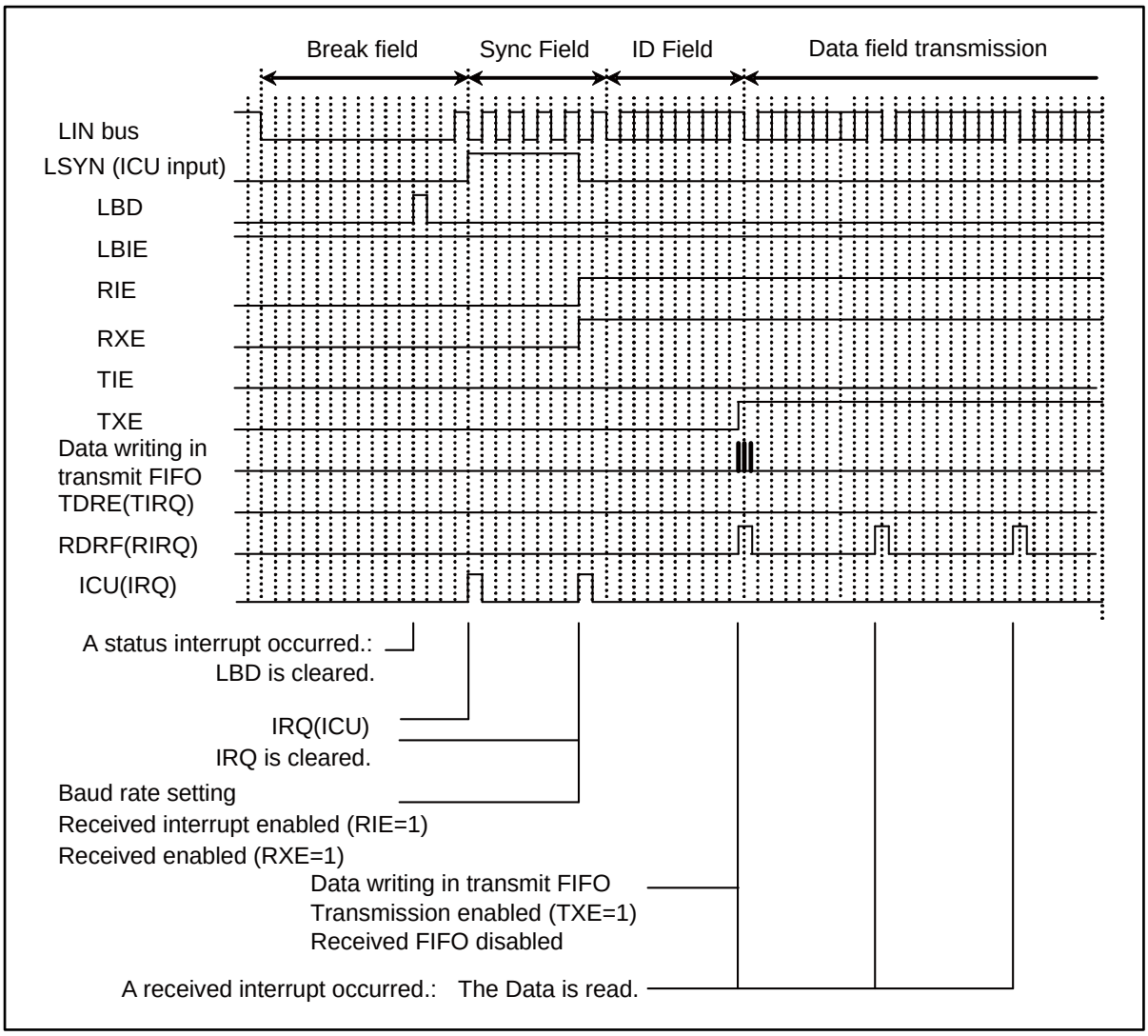
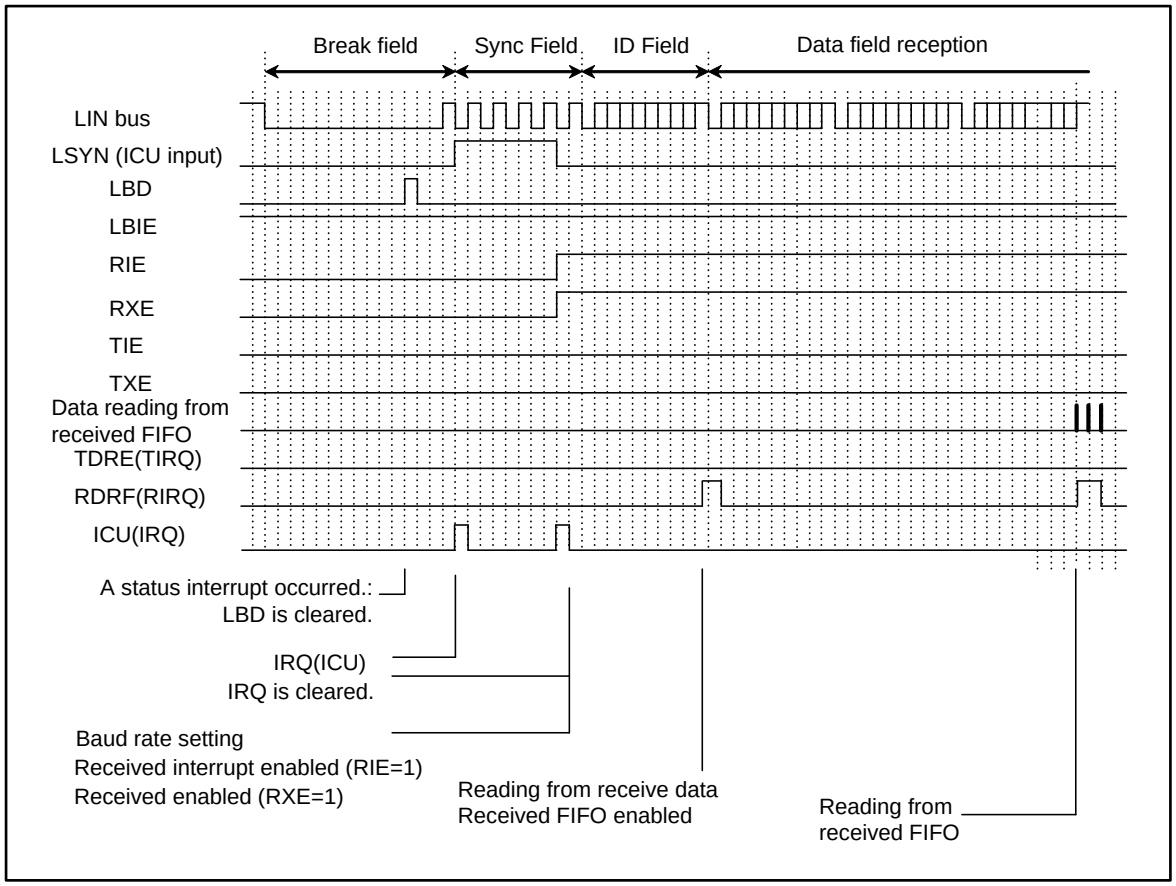


Figure 4-14 LIN bus timing (when DATA field is received and FIFO is used)



5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow

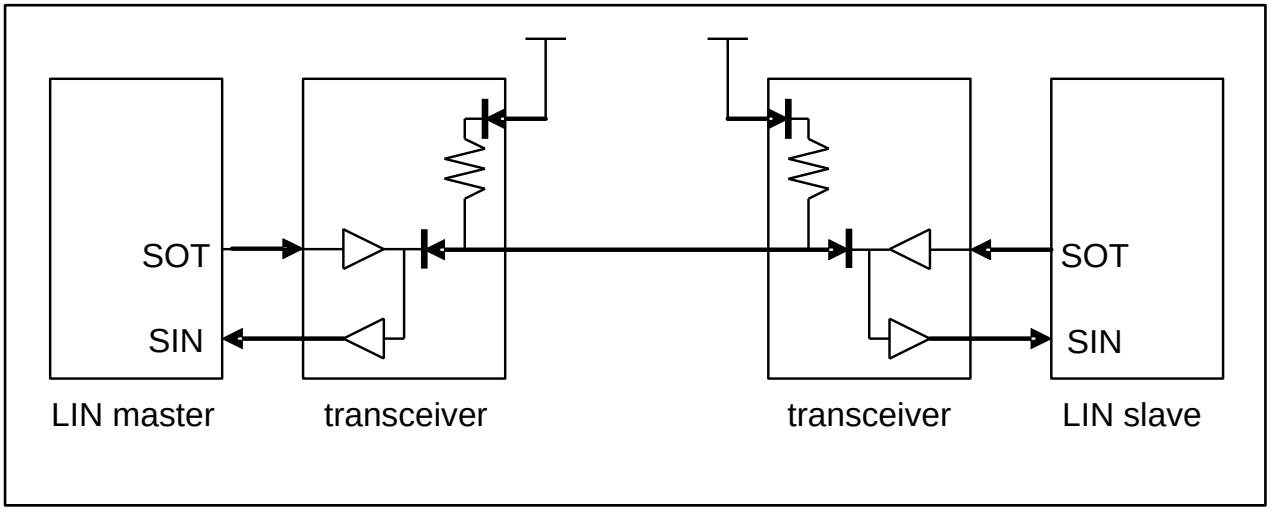
In Operation Mode 3 (LIN communication mode), the LIN interface (Ver. 2.1) can be used for a LIN master or LIN slave system.

Register settings

■ CPU-to-CPU connection

Figure 5-1 shows a communication system consisting of one LIN master and one LIN slave. The LIN interface (ver. 2.1) can work as a LIN master or a LIN slave.

Figure 5-1 Example of LIN bus system communication



Example flowchart

■ Master mode operations

Figure 5-2 Example flowchart of LIN communication in master mode (when FIFO is not used)

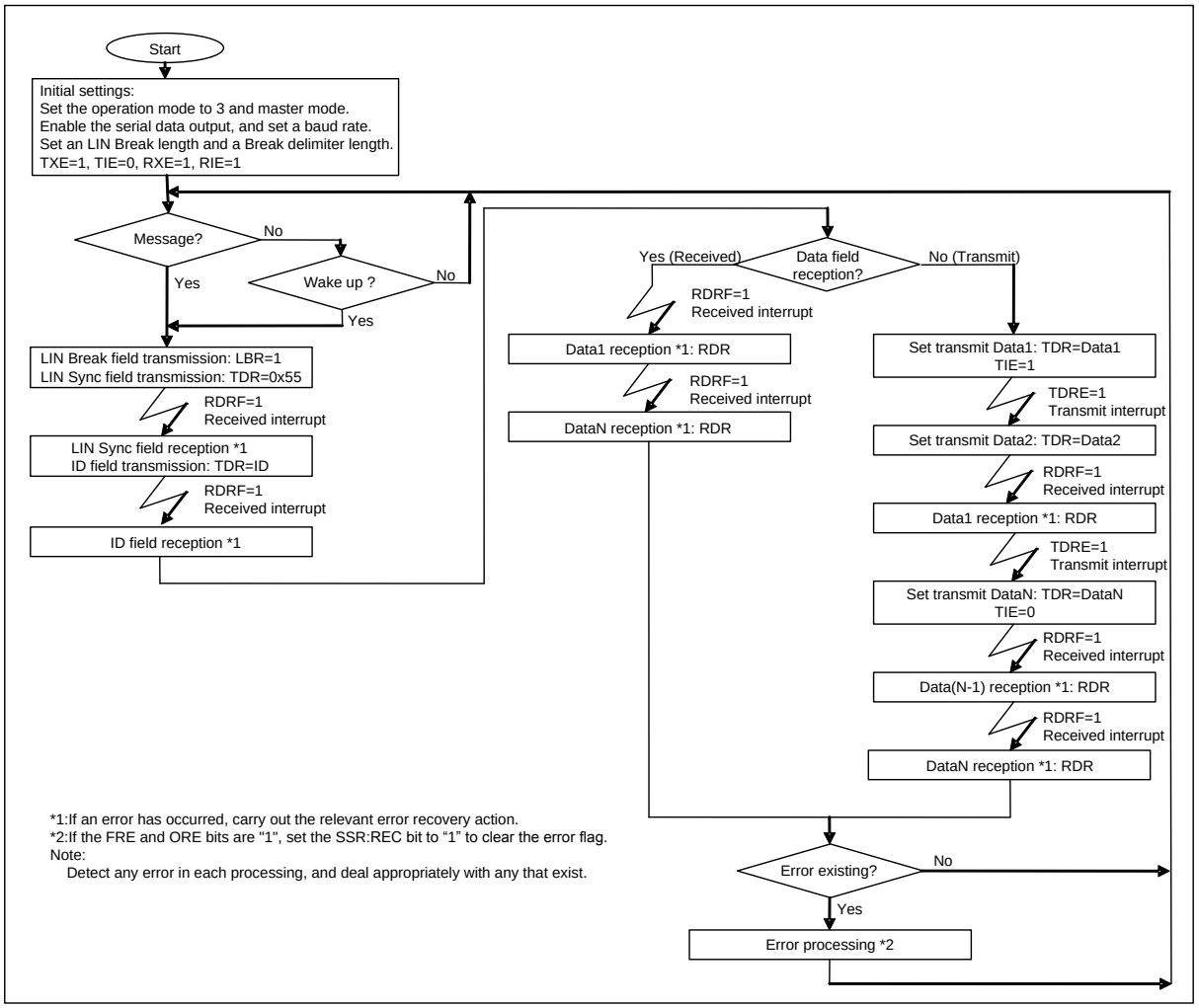
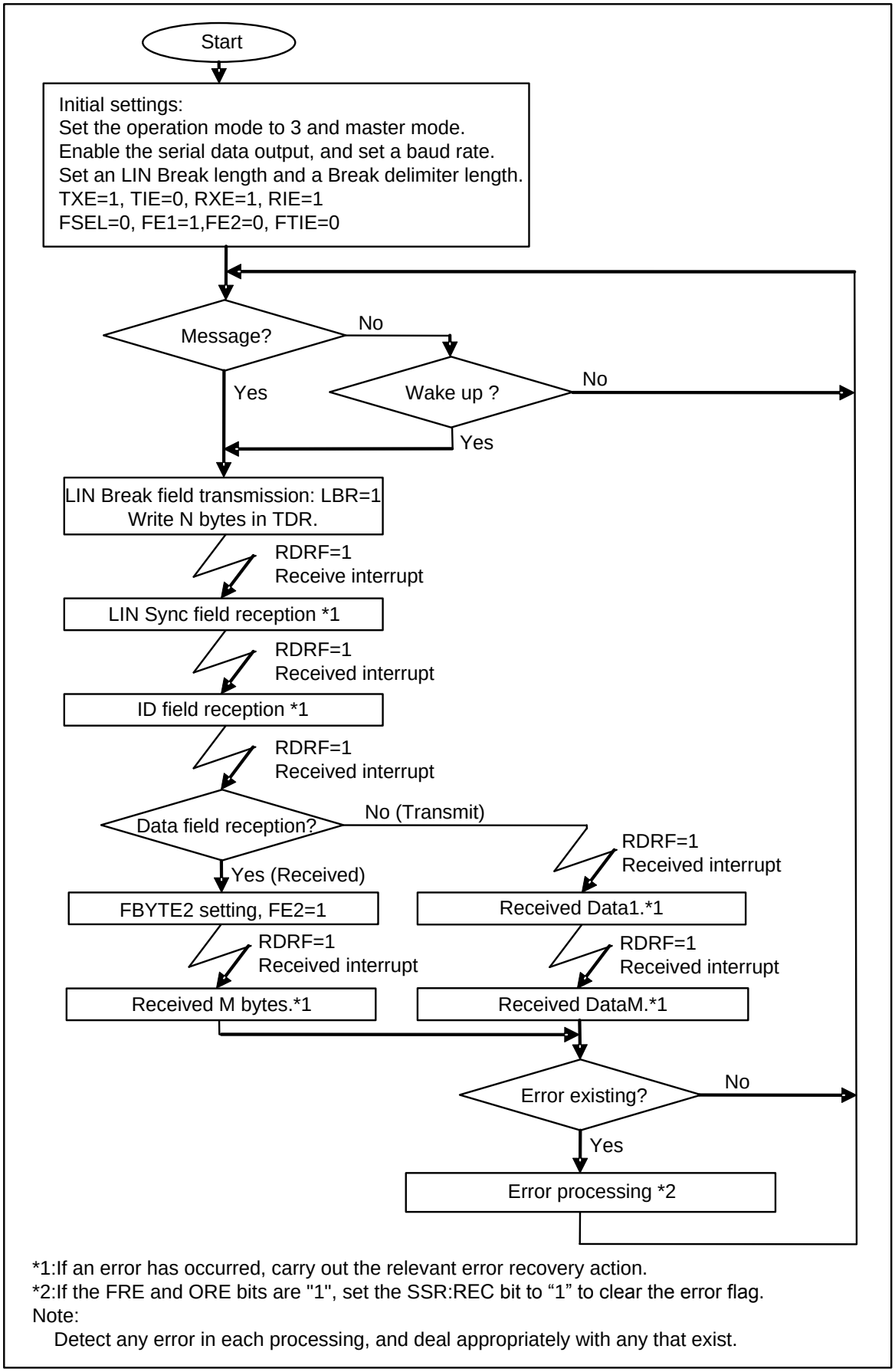


Figure 5-3 Example flowchart of LIN communication in master mode (when FIFO is used)



*1:If an error has occurred, carry out the relevant error recovery action.
*2:If the FRE and ORE bits are "1", set the SSR:REC bit to “1” to clear the error flag.
Note:
Detect any error in each processing, and deal appropriately with any that exist.

■ Slave mode operations

Figure 5-4 Example flowchart of LIN communication in slave mode (when FIFO is not used)

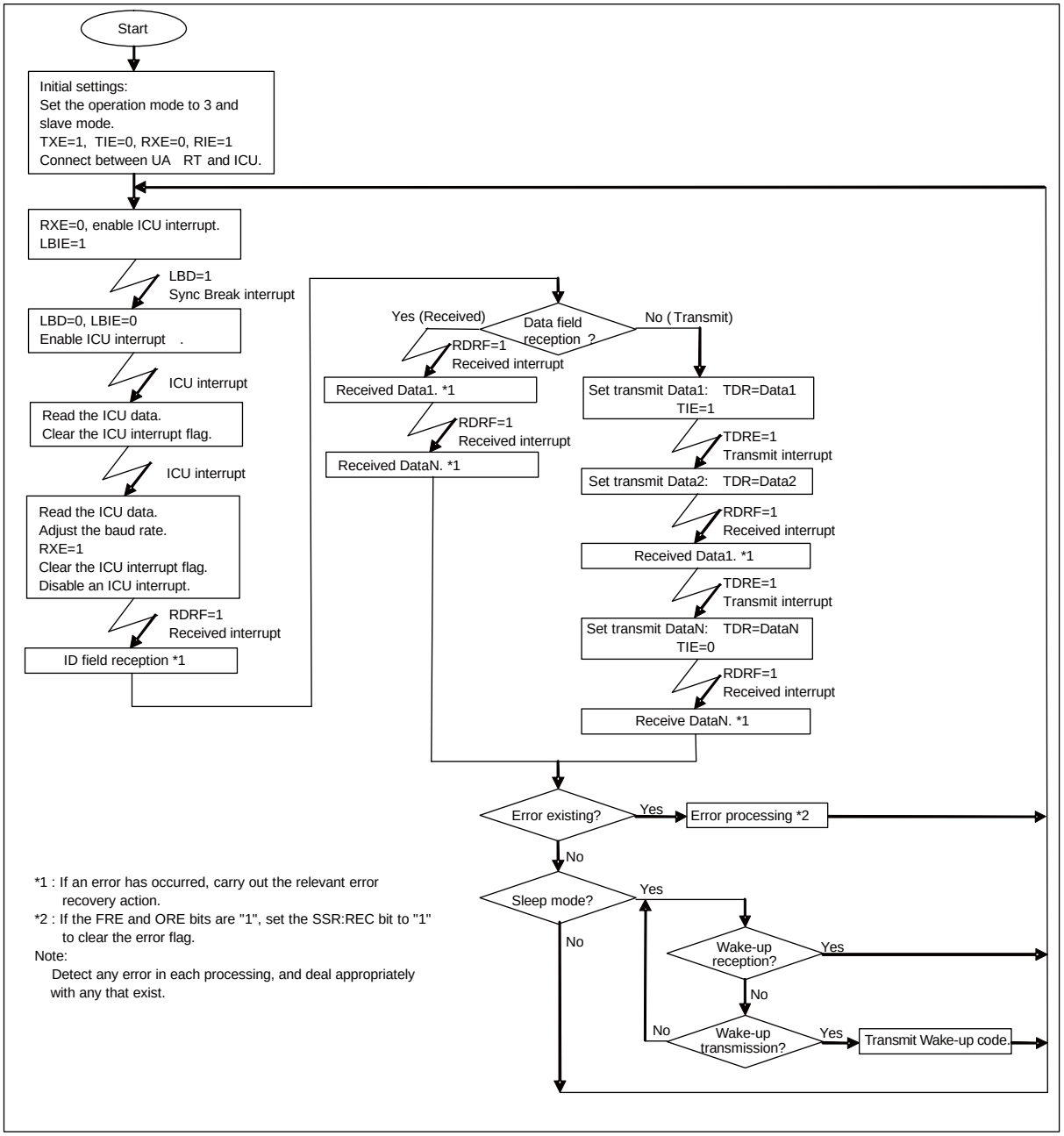
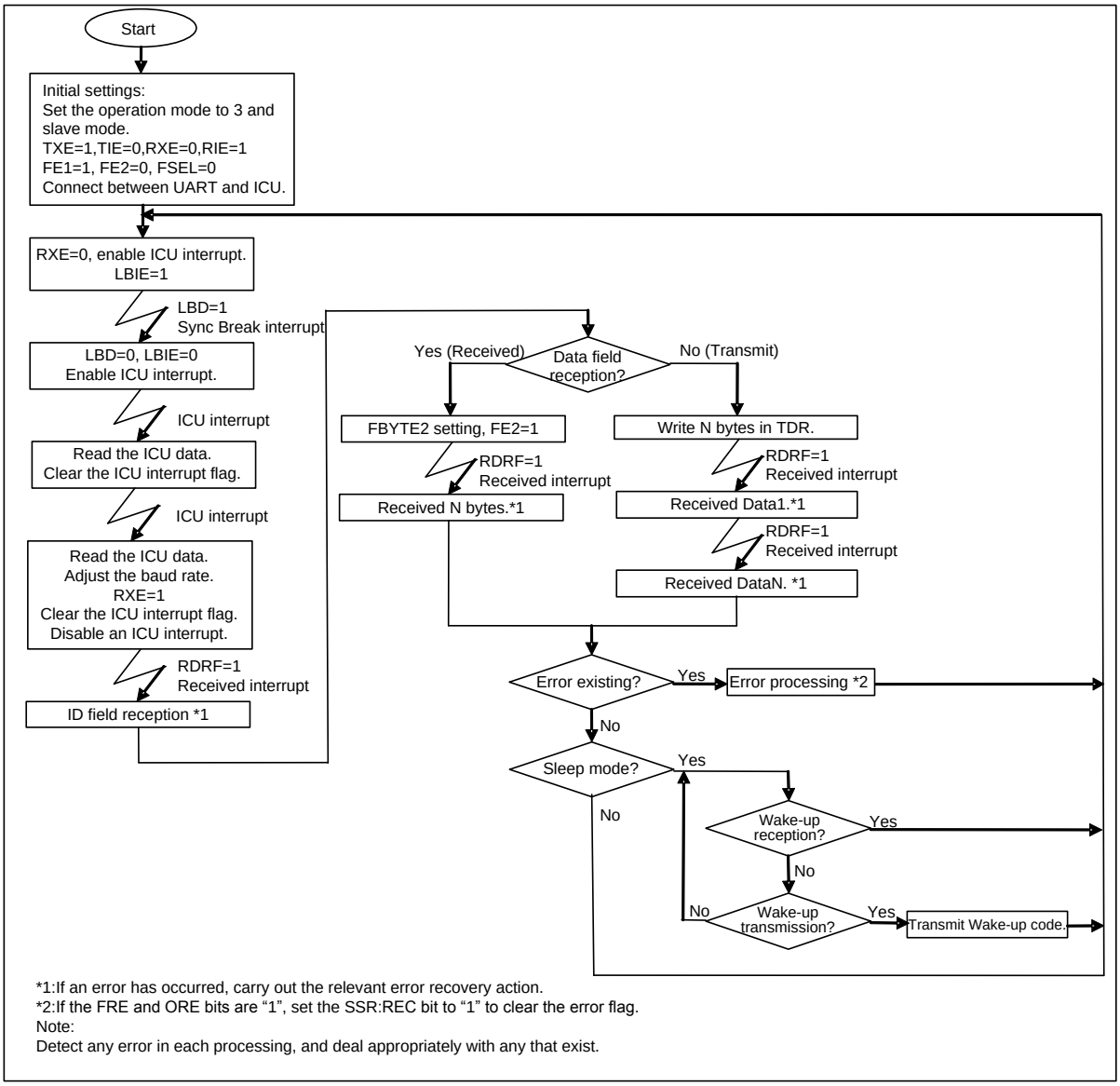


Figure 5-5 Example flowchart of LIN communication in slave mode (when FIFO is used)



*1:If an error has occurred, carry out the relevant error recovery action.
*2:If the FRE and ORE bits are “1”, set the SSR:REC bit to “1” to clear the error flag.
Note:
Detect any error in each processing, and deal appropriately with any that exist.

6. LIN Interface (ver. 2.1) Registers

The following shows a list of LIN interface (ver. 2.1) registers.

List of LIN interface (ver. 2.1) registers

Table 6-1 List of LIN interface (ver. 2.1) registers

	bit15 bit8	bit7 bit0
LIN interface (ver. 2.1)	SCR (Serial Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	ESCR (Extended Communication Control Register)
	-	RDR/TDR (Transmit/Received Data Register)
	BGR1 (Baud Rate Generator Register 1)	BGR0 (Baud Rate Generator Register 0)
FIFO	FCR1 (FIFO Control Register 1)	FCR0 (FIFO Control Register 0)
	FBYTE2 (FIFO2 Byte Register)	FBYTE1 (FIFO1 Byte Register)

Table 6-2 LIN interface (ver. 2.1) bit assignment

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	WUCR	SBL	-	-	SOE
SSR/ ESCR	REC	-	LBD	FRE	ORE	RDRF	TDRE	TBI	-	ESBL	-	LBIE	LBL1	LBL0	DEL1	DEL0
TDR/ RDR	-								D7	D6	D5	D4	D3	D2	D1	D0
BGR1	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

6.1 Serial Control Register (SCR)

The Serial Control Register (SCR) is used to enable/disable a transmit/received interrupt, enable/disable a transmit idle interrupt, and enable/disable data transmission and reception. Also, the SCR can be used to generate a LIN Break field and reset the LIN interface (ver. 2.1).

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	(SMR)
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	-	-	0	0	0	0	0	

[bit15] UPCL: Programmable clear bit

Initializes the internal state of LIN interface (ver. 2.1).

If set to "1":

- The LIN interface (ver. 2.1) is reset directly (Software reset). However, the current register settings are maintained. The transmit or received state is disconnected immediately.
- The baud rate generator reloads the BGR1/0 register value and restarts operation.
- All of transmit/received interrupt factors (SSR:TDRE, TBI, RDRF, FRE, ORE, LBD) are initialized.

If set to "0":

No effect on the operation.

"0" is always read.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	Programmable clear	

Notes:

- *Disable an interrupt first, and then execute the programmable clear instruction.*
- *If the FIFO operation is used, disable it (FCR0:FE[2:1]:=00) first and then execute the programmable clear instruction.*
- *To switch from reception operation to transmit operation continuously, execute the programmable clear instruction after data is received and write transmit data to the Transmit Data Register (TDR).*

[bit14] MS: Master/Slave function select bit

Selects the master or slave mode.

bit	Description
0	Master mode
1	Slave mode

[bit13] LBR: LIN Break Field setting bit (valid in master mode only)

If this bit is set to "1", a LIN Break field (having the length set by the ESCR:LBL1/LBL0 bit) is generated. Also, a LIN Break delimiter (set by the ESCR:DEL1/DEL0 bit) is generated.

When written:

When "0" is written: No effect on the operation.
When "1" is written: A LIN Break field is generated.

When read:

"0" is always read.

bit	Description	
	At writing	At reading
0	No effect on the operation.	"0" is always read.
1	A LIN Break field is generated.	

Notes:

- *This bit setting is valid in the master mode operation only (MS=0).*
- *Do not set this bit to "1" when a LIN Break field is being generated.*

[bit12] RIE: Received interrupt enable bit

- This bit enables or disables an output of received interrupt request to the CPU.
- If the RIE bit and the received data flag bit (SSR:RDRF) are "1", or if any of the error flag bits (SSR:FRE, ORE) is "1", a received interrupt request is output.

bit	Description
0	Disables the received interrupt.
1	Enables the received interrupt.

[bit11] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of transmit interrupt request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a transmit interrupt request is output.

bit	Description
0	Disables a transmit interrupt.
1	Enables a transmit interrupt.

[bit10] TBIE: Transmit bus idle interrupt enable bit

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If the TBIE bit and SSR:TBI bit are "1", a transmit bus idle interrupt request is output.

bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit9] RXE: Data reception enable bit

This bit enables or disables a data reception by the LIN interface (ver. 2.1).

bit	Description
0	Disables data frame reception.
1	Enables data frame reception.

Notes:

- *Data reception is not started unless a falling edge of the start bit is input even if the data reception is enabled (RXE=1).*
- *When a LIN Break field is being sent in the master mode operation, no data is received even if data reception is enabled (RXE=1).*
- *If data reception is disabled (RXE=0), the current data reception is stopped immediately.*

[bit8] TXE: Data transmission enable bit

This bit enables or disables a data transmission by the LIN interface (ver. 2.1).

bit	Description
0	Disables data frame transmission.
1	Enables data frame transmission.

Note:

- *If data transmission is disabled (TXE=0), the current data transmission is stopped immediately.*

6.2 Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to set an operation mode, to select a transmission direction, data length, and stop bit length, and enable or disable an output of serial data to their pins.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCR)	MD2	MD1	MD0	WUCR	SBL	Reserved		SOE
Attribute		R/W	R/W	R/W	R/W	R/W	-	-	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7:5] MD2, MD1, MD0: Operation mode setting bits

These bits set an operation mode.

*This chapter explains the registers and their operation in operation mode 3 (LIN communication mode).

bit7	bit6	bit5	Description
0	0	0	Operation mode 0 (asynchronous normal mode)
0	0	1	Operation mode 1 (asynchronous multiprocessor mode)
0	1	0	Operation mode 2 (clock synchronous mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I^2C mode)
Values other than the above			Setting is prohibited.

Notes:

- *Any bit setting other than above is inhibited.*
- *To switch the current operation mode, issue a programmable clear instruction (SCR:UPCL=1) and switch the operation mode continuously.*
- *After the operation mode has been set, set each register correctly.*

[bit4] WUCR: Wake-up control bit

Selects a pin to be used for an external interrupt.

If set to "0": The INT pin is set as an external interrupt pin.
If set to "1": The SIN pin is set as an external interrupt pin.

bit	Description
0	Disables the Wake-up function.
1	Enables the Wake-up function.

[bit3] SBL: Stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

bit	Description	
0	ESCR:ESBL=0	Stop bit is set to 1 bit
	ESCR:ESBL=1	Stop bit is set to 3 bits
1	ESCR:ESBL=0	Stop bit is set to 2 bits
	ESCR:ESBL=1	Stop bit is set to 4 bits

Notes:

- *In reception operation, only the first bit of the stop bit data is detected.*
- *Always set this bit when transmission is disabled (SCR:TXE=0).*

[bit2:1] Reserved : Reserved bits

The read value is "0". Be sure to write "0".

[bit0] SOE: Serial data output enable bit

This bit enables or disables a serial data output.

bit	Description
0	Disables a serial data output.
1	Enables a serial data output.

Note:

- *If this bit is used as the SOT pin, the GPIO must also be set.*

6.3 Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the current transmission/reception state, check the Received Error flag, detect a LIN Break field, and clear the Received Error flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	-	LBD	FRE	ORE	RDRF	TDRE	TBI	(ESCR)
Attribute	R/W	-	R/W	R	R	R	R	R	
Initial value	0	-	0	0	0	0	1	1	

[bit15] REC: Received Error flag clear bit

This bit clears the FRE and ORE flags of the Serial Status Register (SSR).

bit	Description	
	Writing	Reading
0	No effect on the operation.	"0" is always read.
1	Clears the Received Error flag (FRE, ORE).	

[bit14] - : Unused bit

This bit value is undefined when read.
This bit has no effect on the operation when written.

[bit13] LBD: LIN Break field detection flag bit

This bit shows a detection of LIN Break field.
When 11-bit wide or more of serial input (SIN) are "LOW", the LBD bit is set to "1". If the LIN Break field interrupt enable bit (LBIE) is "1" during this time, a status interrupt occurs.

bit	Description	
	At writing	At reading
0	Clears the LBD flag.	A Break field was not detected.
1	No effect on the operation.	A Break field was detected.

Note:

- *If a read-modify-write instruction is issued, "1" is read.*

[bit12] FRE: Framing error flag bit

- If a framing error occurs during data reception, this bit is set to "1". If the REC bit of Serial Status Register (SSR) is set to "1", this flag is cleared.
- If the FRE and RIE bits are "1", a received interrupt request is output.
- If this flag is set, data of the Received Data Register (RDR) is invalid.
- If this flag is set when received FIFO is used, the received FIFO enable bit is cleared and the received data is not stored in received FIFO.

bit	Description
0	No framing error occurred.
1	A framing error occurred.

[bit11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". If the REC bit of Serial Status Register (SSR) is set to "1", this flag is cleared.
- If the ORE and RIE bits are "1", a received interrupt request is output.
- If this flag is set, data in the Received Data Register (RDR) is invalid.
- If this flag is set when received FIFO is used, the received FIFO enable bit is cleared and the received data is not stored in received FIFO.

bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit10] RDRF: Received data full flag bit

- This flag shows the state of Received Data Register (RDR).
- When the received data is loaded in the RDR, this bit is set to "1". When the Received Data Register (RDR) is read, this bit is cleared to "0".
- If the RDRF and RIE bits are "1", a received interrupt request is output.
- If received FIFO is used, the RDRF bit is set to "1" when the preset amount of data is received in received FIFO.
- If received FIFO is used, this bit is cleared to "0" when received FIFO is emptied.

bit	Description
0	The Received Data Register (RDR) is empty.
1	The Received Data Register (RDR) contains data.

[bit9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If the transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When the data is loaded to the transmit shift register and when the transmission is started, this bit is set to "1" to indicate that the TDR does not contain the valid data.
- If the TDRE and TIE bits are "1", a transmit interrupt request is output.
- When the UPCL bit of Serial Control Register (SCR) is set to "1", the TDRE bit is set to "1".
- For the TDRE bit set/clear timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".

bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register (TDR) is empty.

[bit8] TBI: Transmit bus idle flag bit

- This bit indicates that the LIN interface (ver. 2.1) is not transmitting data.
- When transmit data is written in the Transmit Data Register (TDR), this bit is set to "0".
- When the LIN Break field is set (SMR:LBR=1), this bit is set to "0".
- If the Transmit Data register (TDR) is empty (TDRE=1) and if no transmission is started, this bit is set to "1".
- If the Transmit Data Register is emptied after the LIN Break field has been transmitted, this bit is set to "1".
- If this bit is "1" and if a transmit bus idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

bit	Description
0	Data being transmitted
1	No data transmission

6.4 Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to enable/disable a LIN Break field interrupt, detect a LIN Break field, set a LIN Break field length and a Break delimiter length, and select a stop bit length.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSR)	Reserved	ESBL	-	LBIE	LBL1	LBL0	DEL1	DEL0
Attribute		-	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	-	0	0	0	0	0

[bit7] Reserved : Reserved bit

The read value is "0". Be sure to write "0".

[bit6] ESBL: Extended stop bit length select bit

This bit sets a stop bit length (the frame end mark of the transmit data).

bit	Description	
0	SMR:SBL=0	Stop bit length is set to 1 bit
	SMR:SBL=1	Stop bit length is set to 2 bits
1	SMR:SBL=0	Stop bit length is set to 3 bits
	SMR:SBL=1	Stop bit length is set to 4 bits

Notes:

- *In reception operation, only the first bit of the stop bit data is detected.*
- *Always set this bit when transmission is disabled (TXE=0).*

[bit5] - : Unused bit

This bit value is undefined when read.

This bit has no effect on the operation when written.

[bit4] LBIE: LIN Break field detect interrupt enable bit

This bit enables or disables a LIN Break field detect interrupt.

If the LIN Break field detect flag (LBD) is "1", a received interrupt occurs when an interrupt is enabled (LBIE=1).

bit	Description
0	Disables a LIN Break field detect interrupt.
1	Enables a LIN Break field detect interrupt.

[bit3:2] LBL1/LBL0: LIN Break field length select bits (valid in master mode only)

- These bits set a LIN Break field generation time (in number of bits).
- This bit must be set before the LBR bit of Serial Control Register (SCR) is set to "1" (for LIN Break field transmission).
- A LIN Break field is always detected at the 11th bit in the slave mode operation regardless of this bit setting.

bit3	bit2	Description
0	0	13 bits length
0	1	14 bits length
1	0	15 bits length
1	1	16 bits length

Note:

- *This bit setting is valid in the master mode operation only (SMR:MS="0").*

[bit1:0] DEL1/DEL0: LIN Break delimiter length select bits (valid in master mode only)

- These bits set a LIN Break delimiter length (in number of bits).
- These bits must be set before the LBR bit of Serial Control Register (SCR) is set to "1" (for LIN Break field transmission).

bit1	bit0	Description
0	0	1 bit length
0	1	2 bits length
1	0	3 bits length
1	1	4 bits length

Note:

- *This bit setting is valid in the master mode operation only (SMR:MS=0).*

6.5 Received Data Register/Transmit Data Register (RDR/TDR)

The Received and Transmit Data Registers are allocated at the same address. This register functions as the Received Data Register when data is read from it. This register functions as the Transmit Data Register when data is written in it.

Received Data Register (RDR)

bit	15 ... 8	7	6	5	4	3	2	1	0
Field		D7	D6	D5	D4	D3	D2	D1	D0
Attribute		R	R	R	R	R	R	R	R
Initial value		0	0	0	0	0	0	0	0

The Received Data Register (RDR) is a data buffer register for serial data reception.

- When serial data signals are sent to the Serial Input pin (SIN), they are converted by a shift register and stored in the Received Data Register (RDR).
- When the received data is stored in the Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is set to "1". If a received interrupt is enabled (SSR:RIE=1), a received interrupt request is generated.
- The Received Data Register (RDR) must be read only when the received data full flag bit (SCR:RDRF) is "1". When data is read from the Serial Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is cleared to "0" automatically.
- If a received error occurs (when SSR:ORE or FRE is "1"), data in the Received Data Register (RDR) becomes invalid.

Notes:

- *If received FIFO is used and if the preset amount of data is received in received FIFO, the RDRF bit is set to "1".*
- *If received FIFO is used and if this buffer is emptied, the RDRF bit is cleared to "0".*
- *If a received error occurs when received FIFO is used (SSR:ORE or FRE is "1"), the received FIFO enable bit is cleared and the received data is not stored in received FIFO.*

Transmit Data Register (TDR)

bit	15 ... 8	7	6	5	4	3	2	1	0
Field		D7	D6	D5	D4	D3	D2	D1	D0
Attribute		W	W	W	W	W	W	W	W
Initial value		1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a data buffer register for serial data transmission.

- If data transmission is enabled (SCR:TXE=1) and if the transmit data is written in the Transmit Data Register (TDR), the transmit data is transferred to the transmit shift register. Then, the data is converted into serial data, and output at the serial data output pin (SOT).
- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When the transmit data is transferred to the serial transmit shift register and data transmission is started, and if transmit FIFO is disabled or if transmit FIFO is empty, the transmit data empty flag (SSR:TDRE) is set to "1".
- If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written in the buffer. If a transmit interrupt is enabled, a transmit interrupt occurs. The next transmit data must be written only after the transmit interrupt has occurred or when the transmit data empty flag (SSR:TDRE) is "1".
- If the transmit data empty flag (SSR:TDRE) is "0" and transmit FIFO is disabled or transmit FIFO is full, no transmit data can be written in the Transmit Data Register (TDR).

Notes:

- *The Transmit Data Register is a write-only register. While the Received Data Register is a read-only register. As these two registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) operation cannot be used.*
- *For the transmit data empty flag (SSR:TDRE) set timing when transmit FIFO is used, see "2.4 Interrupt and flag set timing when transmit FIFO is used".*

6.6 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks. Also, an external clock can be selected as the clock source of the reload counter.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXT	(BGR1)							(BGR0)							
Attribute	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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- The Baud Rate Generator Registers are used to set a frequency division ratio of serial clocks.
- The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/BGR0 set value can be read.
- When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the reload counter starts its counting.
- The EXT bit (bit15) specifies to use the clock source of reload counter as the internal clock or the external clock. If EXT=0 is set, an internal clock is used. If EXT=1 is set, an external clock is used.

[bit15] EXT: External clock select bit

bit	Description
0	Uses the internal clock.
1	Uses an external clock.

[bit14:8] BGR1: Baud Rate Generator Register 1

bit14:8	Description
Write	Writes data in bit8 to bit14 of reload counter.
Read	Reads the BGR1 set value.

[bit7:0] BGR0: Baud Rate Generator Register 0

bit7:0	Description
Write	Writes data in bit0 to bit7 of reload counter.
Read	Reads the BGR0 set value.

Notes:

- *Data must be written in the Baud Rate Generator Register1, 0 (BGR1 and BGR0) in 16-bit data access mode.*
- *If the current values of Baud Rate Generator Register1, 0 (BGR1, BGR0) are changed, the new values are reloaded only after the counter value has reached "15h00". In order to validate the new set values immediately, change the BGR1/BGR0 set values and execute the programmable clear (UPCL).*
- *If the reload value is even, the "LOW" signal width of serial clock is longer than the "HIGH" signal width for a single cycle of bus clock. If the value is odd, the serial clock has the same "HIGH" and "LOW" signal width.*
- *Set the reload value to 3 or more. Note that data may not be received normally due to the baud rate error and reload value setting.*
- *When the baud rate generator is operating and if you need to switch to the external clock (EXT=1), first set the baud rate generators 1 and 0 (BGR1 and BGR0) to "0". Then, execute the programmable clear instruction (UPCL) and select the external clock (EXT=1).*

6.7 FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to set the FIFO test, select transmit or received FIFO, enable transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
Attribute	-			R/W	R/W	R/W	R/W	R/W	
Initial value	-			0	0	1	0	0	

[bit15:13] Reserved : Reserved bits

The read value is "0". Be sure to write "0".

[bit12] FLSTE: Re-transmit data lost detect enable bit

This bit enables the FLST bit detection.

bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

Note:

- *To set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".*

[bit11] FRIIE: Received FIFO idle detect enable bit

This bit sets to detect the received idle state if received FIFO contains valid data for more than 8-bit hours. If the received interrupt is enabled (SCR:RIE=1), a received interrupt is generated when the received idle state is detected.

bit	Description
0	Disables the received FIFO idle detection.
1	Enables the received FIFO idle detection.

Note:

- *In case of using Received FIFO, set this bit to "1".*

[bit10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. If the Transmit Interrupt is enabled (FTIE=1) during this time, a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).
- Transmit FIFO is reset.

The FDRQ bit is cleared when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

Notes:

- *If the FBYTE (for transmission) is "0", this bit cannot be set to "0".*
- *If this bit is "0", the FSEL bit state cannot be changed.*
- *If this bit is set to "1", it has no effect on the operation.*
- *If a read-modify-write instruction is issued, "1" is read.*

[bit9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit8] FSEL: FIFO select bit

This bit selects the transmit FIFO and received FIFO.

bit	Description
0	Transmit FIFO:FIFO1; Received FIFO:FIFO2
1	Transmit FIFO:FIFO2; Received FIFO:FIFO1

Notes:

- *This bit is not cleared by FIFO reset (FCR0:FCL[2:1]=11).*
- *To change this bit state, first disable the FIFO operation (FCR0:FE[2:1]=00).*

6.8 FIFO Control Register 0 (FCR0)

FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute		-	R	R/W	W	R/W	R/W	R/W	R/W
Initial value		-	0	0	0	0	0	0	0

[bit7] - : Unused bit

This bit value is undefined when read.

This bit has no effect on the operation when written.

[bit6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- The FLSTE bit of FIFO Control Register 1 (FCR1) is "1", the write pointer of transmit FIFO matches the read pointer which has been saved by the FSET bit, and data is written in FIFO.

The FLST bit is cleared when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data identified by the read pointer (saved by the FSET bit) is overwritten. Therefore, the FLD bit cannot set the data re-transmission even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in FIFO again.

bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.

When the re-transmission setting has finished, this bit is set to "0".

bit	Description
0	Not reloaded
1	Reloaded

Notes:

- *If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.*
- *When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".*
- *After you have set the TIE and TBIE bits to "0", set this bit to "1". After you have enabled transmit FIFO, set the TIE and TBIE bits to "1".*

[bit4] FSET: FIFO pointer save bit

This bit saves the transmit FIFO read pointer.

If the read pointer is saved before transmission and if the FLST bit is "0", data can be re-transmitted even when a communication error or others occur.

bit	Description
0	Not saved
1	Saved

Note:

- *This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".*

[bit3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

If this bit is set to "1", the FIFO2 internal state is initialized.

Only the FCR1:FLST2 bit is initialized, but the other bits of FCR1/FCR0 registers are kept.

bit	Description	
	Writing	Reading
0	No effect on the operation.	"0" is always read.
1	FIFO2 is reset.	

Notes:

- *Disable the transmission and reception first, and then reset FIFO2.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The valid data count of the FBYTE2 register is set to "0".*

[bit2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 value.

If this bit is set to "1", the FIFO1 internal state is initialized.

Only the FCR1:FLST1 bit is initialized, but the other bits of FCR1/FCR0 registers are kept.

bit	Description	
	Writing	Reading
0	No effect on the operation.	"0" is always read.
1	FIFO1 is reset.	

Notes:

- *Disable the transmission and reception first, and then reset FIFO1.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The valid data count of the FBYTE1 register is set to "0".*

[bit1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If FIFO2 is set as transmit FIFO and if data exists in FIFO2 when this bit is set to "1", the data transmission starts immediately when the LIN interface (ver. 2.1) is enabled to transmit data (TXE=1). During this time, set both TIE and TBIE bits to "0". Then, set this bit to "1" and set both TIE and TBIE bits to "1".
- If received FIFO is selected by the FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- If FIFO2 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (TDRE=1).
- If FIFO2 is used as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and no valid data exists in received FIFO (FBYTE2=0x00) after reception is disabled (SCR:RXE=0).
- If FIFO2 is used as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO2 state is held even if the FIFO2 operation is disabled.

bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

[bit0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- If FIFO1 is set as transmit FIFO and if data exists in FIFO1 when this bit is set to "1", the data transmission starts immediately when the LIN interface (ver. 2.1) is enabled to transmit data (TXE=1). During this time, set both TIE and TBIE bits to "0". Then, set this bit to "1" and set both TIE and TBIE bits to "1".
- If received FIFO is selected by the FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- If FIFO1 is used as transmit FIFO, this bit must be set to "1" or "0" when the transmit buffer is empty (TDRE=1).
- If FIFO1 is used as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and no valid data exists in received FIFO (FBYTE2=0x00) after reception is disabled (SCR:RXE=0).
- If FIFO1 is used as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) after reception is disabled (SCR:RXE=0).
- The FIFO1 state is held even if the FIFO1 operation is disabled.

bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

6.9 FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer. Also, this register can be used to generate a received interrupt when a certain number of data sets is received in the received FIFO.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count of FIFO. The following shows the settings of the FCR1:FSEL bit.

Table 6.9-1 Display of data count

FCR1:FSEL	FIFO selection	Data count display
0	FIFO2:Received FIFO, FIFO1:Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2:Transmit FIFO, FIFO1:Received FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to the FBYTE register of received FIFO to generate a received interrupt flag. If this transfer data count matches the FBYTE register display, the received data full flag bit (RDRF) is set to "1".
- If the following two conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the received data full flag (SSR:RDRF) is set to "1".
  - The received FIFO idle detect enable bit (FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.

If the RDR data is read during counting of 8 clocks, this counter is reset to "0", and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to "0". If data remains in received FIFO and if received FIFO is enabled, the data counting is restarted.

[bit15:8] FBYTE2: FIFO2 data count display bits

[bit7:0] FBYTE1: FIFO1 data count display bits

Writing	Sets the transfer data count.
Reading	Reads the effective count of data.

Read (Effective data count)

During transmission: The number of data sets already written in FIFO but not transmitted yet
During reception: The number of data sets received in FIFO

Write (Transfer data count)

During transmission: Set "0x00".
During reception: Set the data count to generate a received interrupt.

Table 6-3 Data Count to be Saved in FIFO

FIFO Capacity	Max. FBYTE Count	Max. Data Count to be Saved in FIFO
16 BYTEs	16	16
32 BYTEs	32	32
64 BYTEs	64	64
128 BYTEs	128	128

Notes:

- *Set "0x00" in the FBYTE register of transmit FIFO.*
- *Set data equal to or greater than "1" in the FBYTE register of received FIFO.*
- *This state can be changed only after the data transmission or reception has been disabled.*
- *A read-modify-write instruction cannot be used for this register.*
- *Any setting exceeding the FIFO capacity is prohibited.*
- *After setting FIFO select bit (FCR1:FSEL), set FIFO byte register (FBYTE).*
- *FIFO select bit (FCR1:FSEL) and FIFO byte register (FBYTE) cannot be set at the same time.*
- *In the FIFO data count display at transmit, the data count which is made by subtracting "1" from transmit data written count is displayed. This is because data transmitted is written to be saved in transmit FIFO when the data not transmitted to TDR register exists. When data in TDR register is transmitted, the data not transmitted in transmit FIFO is transferred to TDR register.*
- *In the FIFO data count display at reception, the count of data which is received but not read is displayed. The data under receiving at TDR register is no included.*

CHAPTER 1-5: I²C Interface (I²C Communications Control Interface)

This chapter explains the I²C function supported in operation mode 4 of the multifunction serial interface.

1. Overview of I²C Interface (I²C Communications Control Interface)
2. I²C Interface interrupt
3. Dedicated Baud Rate Generator
4. I²C communication operation flowchart examples
5. I²C Interface Registers

CODE: 9BFI2C_FM4-E01.0_FM15I-J05.4

1. Overview of I²C Interface (I²C Communications Control Interface)

The I²C interface (I²C communications control interface) supports the I²C bus and operates as a master/slave device on the I²C bus. It also has transmit/received FIFO (up to 128 bytes each)*1 installed.

Functions of I²C interface (I²C communications control interface)

		Function
1	Data buffer	− Full duplex double buffer (when FIFO is not used) − Transmit/received FIFO (max 128 bytes each) * (when FIFO is used)
2	Serial input	Removes noise from 2 clocks to 32 clocks in the bus clock for serial clock/serial data input.
3	Transfer mode	Synchronous
4	Baud rate	− A dedicated baud rate generator (constructed with a 15-bit reload counter) − The external clock can be adjusted with the reload counter. − Supports Standard-mode/Fast-mode/ Fast-mode Plus*2
5	Data length	8 bits
6	Signaling system	NRZ (Non Return to Zero)
7	Interrupt request	− Received interrupt − Transmit interrupt − Request of status interrupt/interrupt to ICU − Transmit FIFO interrupt (when transmit FIFO is empty) − DMA(Transmit/Received) transferring support function is available.
8	I²C	− Master/slave transmission and reception functions − Arbitration function − Clock synchronization function − Transmission direction detection function − Function to generate and detect iteration start condition − Bus error detection function − General call addressing function − 7-bit addressing as master/slave − Generation of interrupt enabled during transmission or a bus error − The 10-bit addressing function can be programmatically enabled.
9	FIFO	− Transmit/received FIFO installed (maximum capacity: 128bytes for transmit FIFO, 128 bytes for received FIFO) *1 − Transmit FIFO or received FIFO can be selected. − Transmit data can be resent. − Received FIFO interrupt timing can be changed via software. − FIFO resetting is supported independently.

*1 : The FIFO capacity size varies depending on the product type.

*2 : For Fast-mode Plus operation, the dedicated I/O settings are required. For details, see Chapter I/O Port.

2. I²C Interface interrupt

I²C interface interrupt request is generated due to the following factors.

- After transmission/reception of the first byte and after data transmission/reception is completed
- Stop condition
- Iteration start condition
- FIFO transmit data request
- FIFO received data completed

I²C Interface Interrupt

Table 2-1 shows the interrupt control bits and interrupt factors for the I²C interface.

Table 2-1 Interrupt control bits and interrupt factors for the I²C interface

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Operation to clear interrupt request flag
Status	INT	IBCR	The first byte has been transmitted/received*1 (except for master operation when SSR:DMA=1)	IBCR:INTE	Setting the interrupt flag bit (IBCR:INT) to "0"
			Data has been transmitted/received*1 (When SSR:DMA=0)		
			Bus Error detection (EIBCR:BCE=0)		
			Detection of a bus error		
			Detection of arbitration lost		
			Detection of reserved address		
			Reception of NACK		
			Received FIFO being full during reception as a slave (When SSR:DMA=0)		Setting IBCR:INT to "0" after reading received data until received FIFO is emptied
	SPC	IBSR	Stop condition	IBCR:CNDE	Setting SPC to "0"
	RSC		Detection of iteration start		Setting RSC to "0"
Reception	RDRF	SSR	Reception of reserved address	SMR:RIE	Reading from the received data register (RDR)
			Completion of data reception		
			Reception of a data volume matching the value set for FBYTE.		Reading from the Received Data Register (RDR) until received FIFO is emptied
			Detection of reception idling when FRIIE=1		
	ORE	SSR	Overrun error		Setting the reception error flag bit (SSR:REC) to "1"

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Operation to clear interrupt request flag
Transmission	TDRE	SSR	The Transmit Data Register is empty. Setting the transmit buffer empty flag set bit (SSR:TSET) to "1"	SMR:TIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *2
	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	The FIFO transmit data request bit is set to "0" or transmit FIFO is full.
	TBI (SSR: DMA=1)	SSR	No transmission operation Setting the transmit buffer empty flag set bit (SSR:TSET) to "1"	SCR:TBIE	Writing to the Transmit Data Register (TDR) or setting the transmit FIFO operation enable bit to "1" when the transmit FIFO operation enable bit is set to "0" and valid data are present in transmit FIFO (re-transmitting data) *3

*1 : If normal data can be transmitted/received and SSR:TDRE is "0", no interrupt is generated. This to support DMA transfers.
To generate the IBCR:INT bit at a time of data transmission/reception, the SSR:TDRE bit needs to be set to "1" before the IBCR:INT bit is set.

*2 : Be sure to check that the SSR:TDRE bit is set to "0" and then set the SMR:TIE bit to "1".

*3 : Be sure to check that the SSR:TBI bit is set to "0" and then set the SSR:TBIE bit to "1".

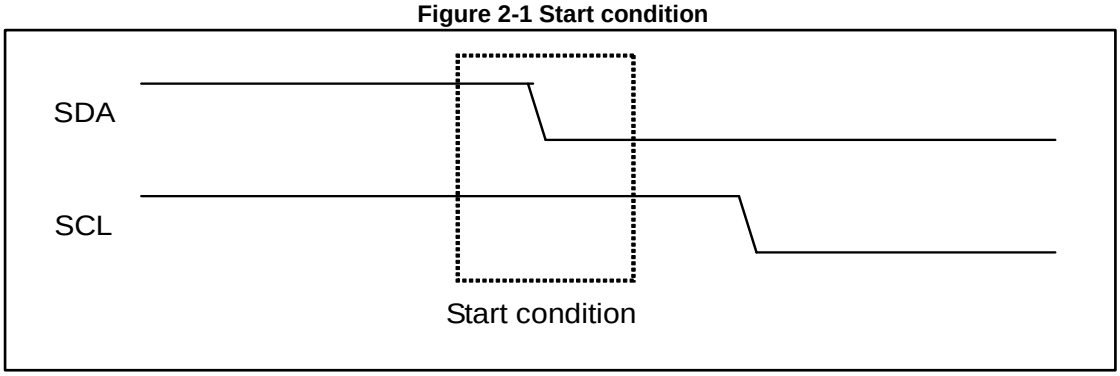
2.1 I²C interface operation

The I²C interface performs communications using two two-way bus lines, a serial data line (SDA) and a serial clock line (SCL).

I²C bus start condition

The following shows the I²C bus start condition.

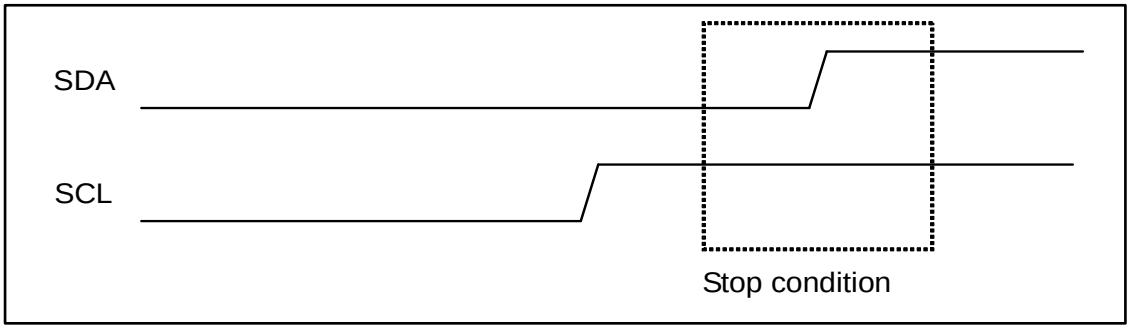
Figure 2-1 Start condition



I²C bus stop condition

The following shows the I²C bus stop condition.

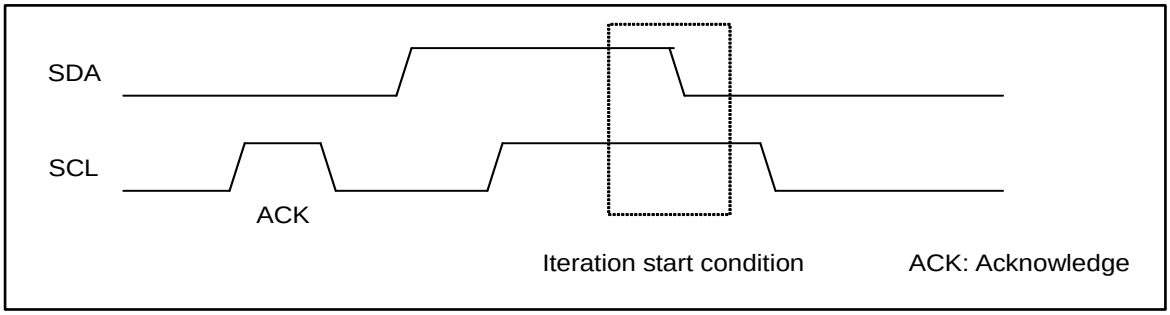
Figure 2-2 Stop condition



I²C bus iteration start condition

The following shows the I²C bus iteration start condition.

Figure 2-3 Iteration start condition



2.2 Master mode

Master mode generates the start condition on the I^2C bus and outputs clocks to the I^2C bus. When the MSS bit in the IBCR register is set to "1" while the I^2C bus is in idle state (SCL=HIGH, SDA=HIGH), master mode is activated, causing the ACT bit in the IBCR register to be set to "1".

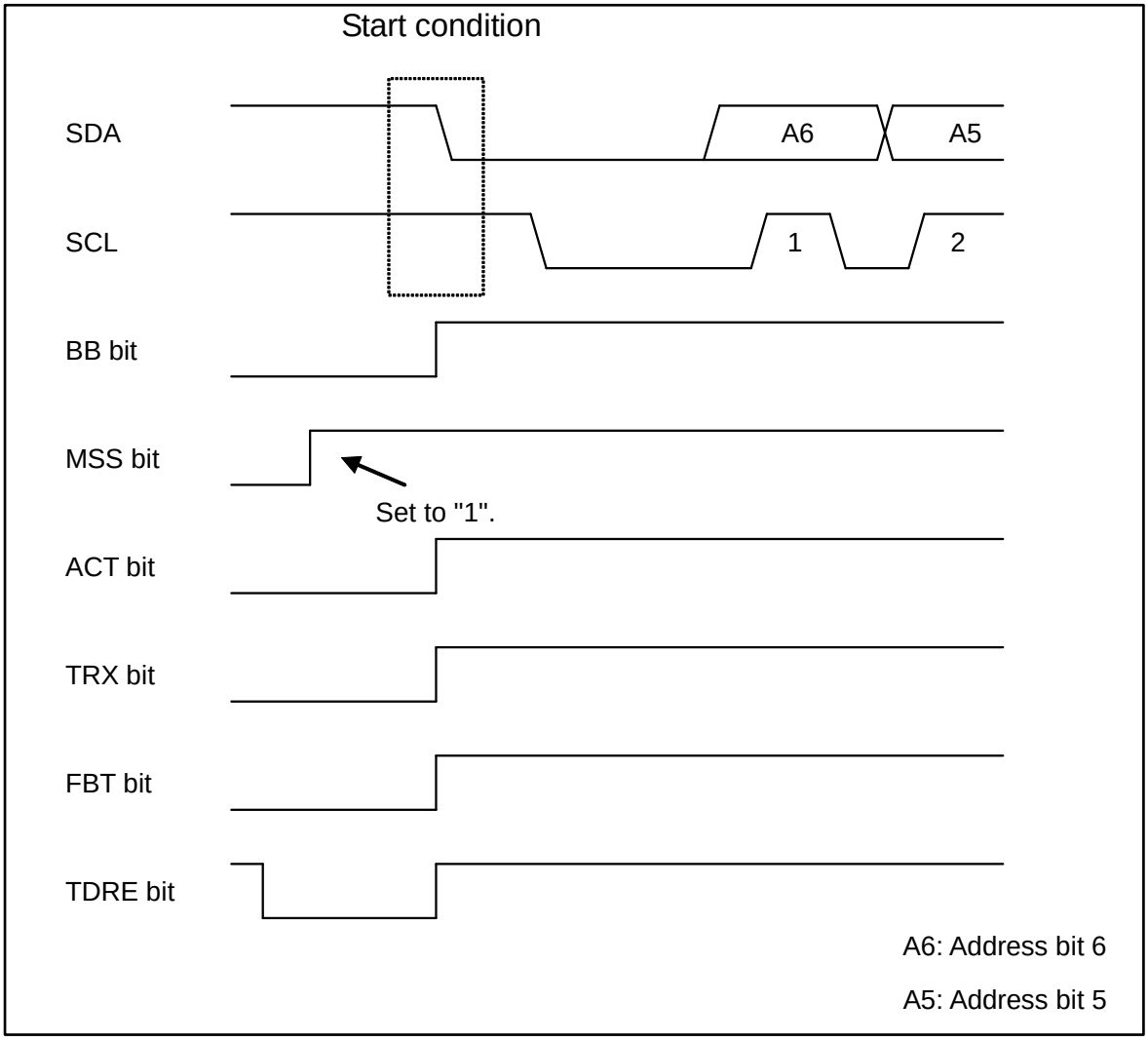
Generating start condition

The start condition is generated under the following condition.

- When SDA="H", SCL="H", ISMK:EN=1 and IBSR:BB=0, the IBCR:MSS bit is set to "1".

Outputting the start condition to the I^2C bus causes the IBCR:ACT bit to be set to "1". After that, when the start condition is received, the IBSR:BB bit is set to "1" to indicate that the I^2C bus is carrying out communications. (See Figure 2-4.)

Figure 2-4 Start condition output and relationships with respective bits



Note:

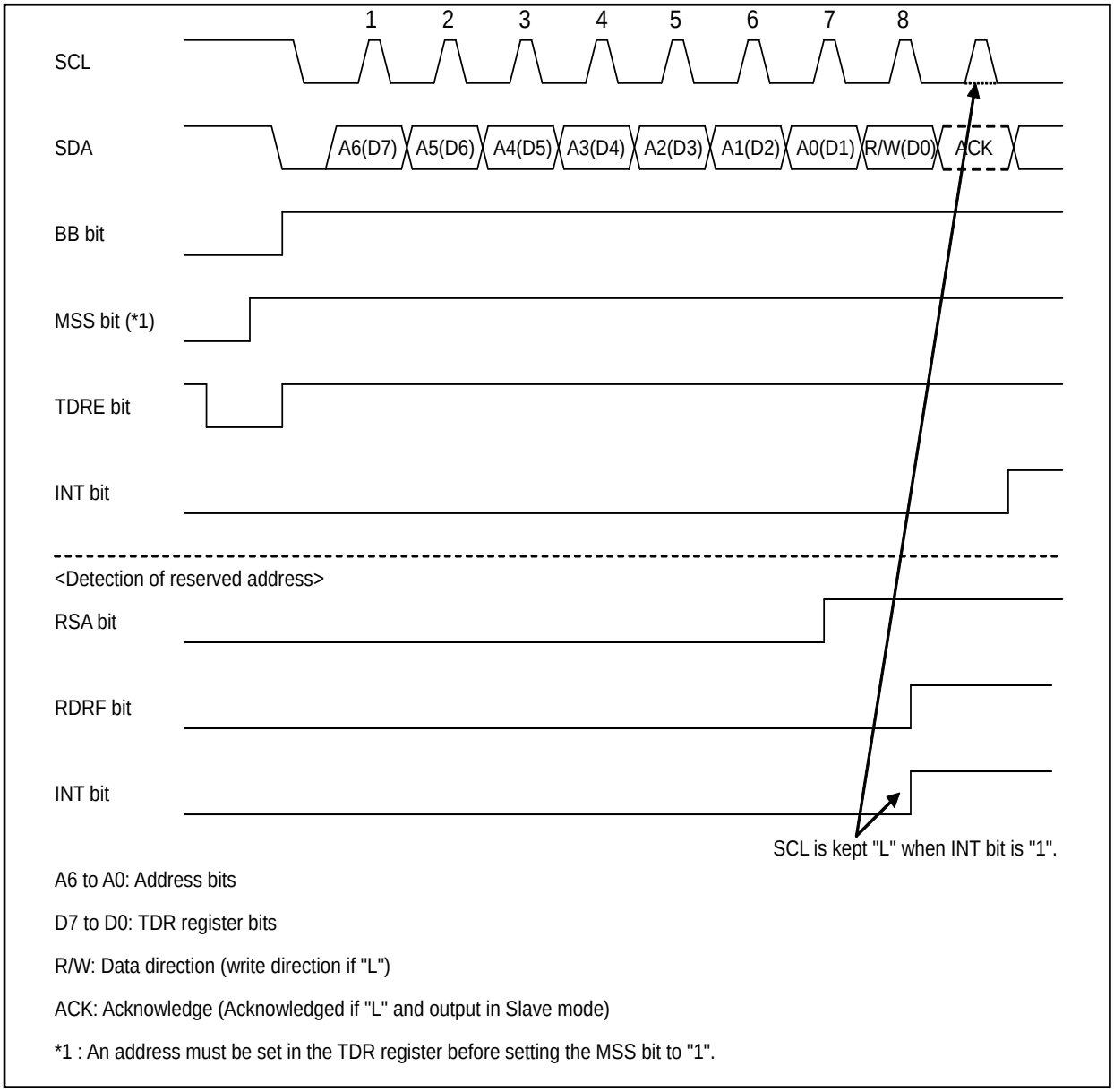
- *In operation mode 4 (I^2C mode), the bus clock is used at a frequency no lower than 8 MHz. Also note that setting of a baud rate generator that exceeds 400 kbps is prohibited.*

Slave address output

Outputting the start condition causes data that are set in the TDR register to be output as the address, starting with bit 7. When FIFO is enabled, the data in the TDR register that is written the earliest is output. bit 0 is used as the data direction bit (R/W). When the data direction bit (R/W) is "0", it indicates that data flow in the write direction (from the master to a slave). Set the address to the TDR register before setting the IBCR:MSS bit to "1" or IBCR:SCC bit to "1".

For the output timing of the address and the data direction, see Figure 2-5, Figure 2-6.

Figure 2-5 Address and data direction (when FIFO is disabled)



A6 to A0: Address bits

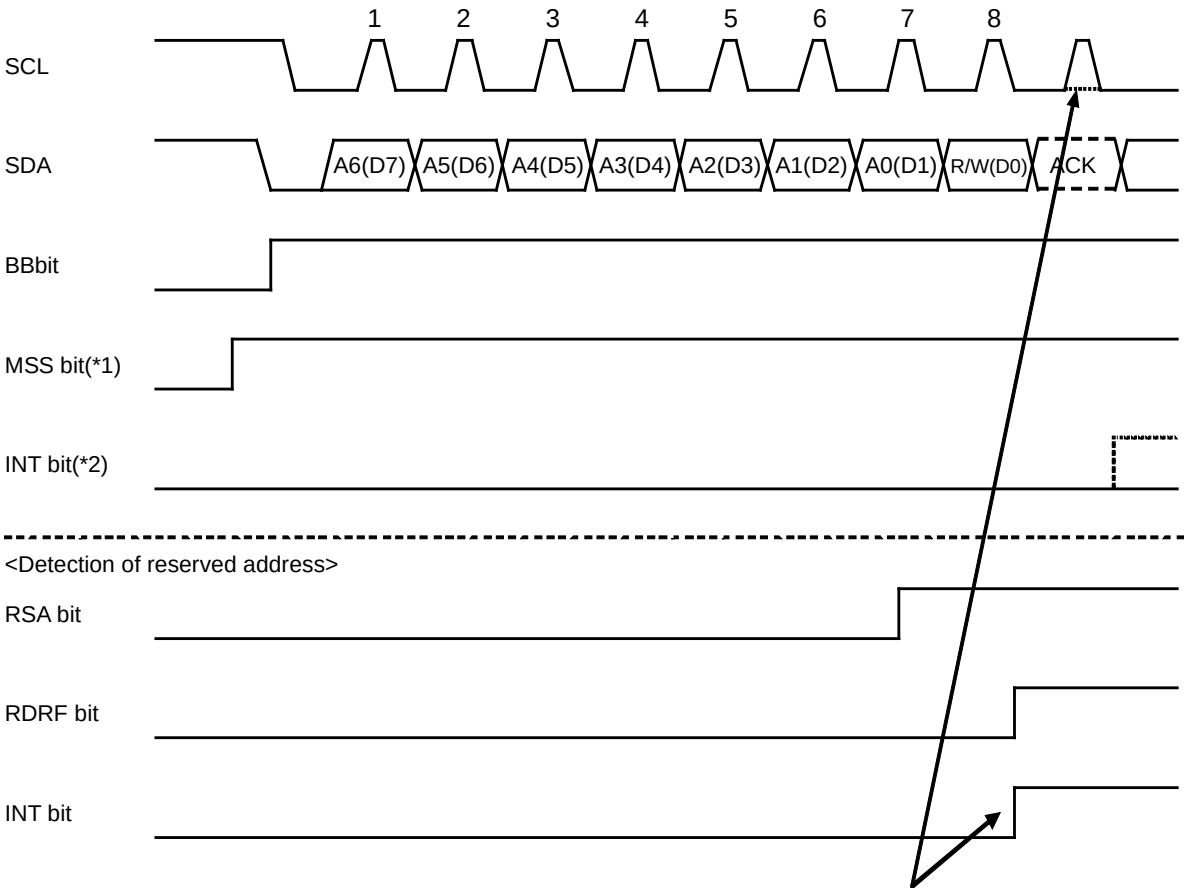
D7 to D0: TDR register bits

R/W: Data direction (write direction if "L")

ACK: Acknowledge (Acknowledged if "L" and output in Slave mode)

*1 : An address must be set in the TDR register before setting the MSS bit to "1".

Figure 2-6 Address and data direction (when transmit/received FIFO is enabled)



A6 to A0: Address bits

D7 to D0: TDR register bits

R/W: Data direction (write direction if "L")

ACK: Acknowledge (Acknowledged if "L" and output in Slave mode)

*1 : An address must be set in the TDR register before setting the MSS bit to “1”.

*2 : If acknowledged with "L" and if R/W="L", the Send FIFO buffer has data. If acknowledged with "L" and if R/W="H", the Receive FIFO buffer has no data, the INT bit is not set to "1".

Acknowledgement reception by first byte transmission

When the data direction bit (R/W) is output, the I^2C interface receives acknowledgement from a slave. The following lists operations to enable/disable FIFO.

Table 2-2 Operations after acknowledgement reception with DMA mode disabled (IBSR:RSA="0", SSR:DMA="0")

Transmit FIFO	Received FIFO	Transmit FIFO status	Received FIFO status	Data direction bit (R/W)	Operation immediately after receiving acknowledgement	
					Acknowledgement: ACK	Acknowledgement: NACK
Disable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1		
Disable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data		Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	
Enable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1		
Enable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data		Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	

Table 2-3 Operations after acknowledgement reception with DMA mode enabled (IBSR:RSA="0", SSR:DMA="1")

Transmit FIFO	Received FIFO	Transmit FIFO status	Received FIFO status	Data direction bit (R/W)	Operation immediately after receiving acknowledgement	
					Acknowledgement: ACK	Acknowledgement: NACK
Disable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1		
Disable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data		Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	
Enable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
				1		
Enable	Enable	-	Without data	0	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	Sets the IBCR:INT bit to "1" with the wait state.
			With data		Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the SSR:TBI bit to "1" and waits. If the SSR:TDRE bit is set to "0", SSR:TBI bit stays "0" without the wait state.	

■ When DMA mode is disabled (SSR:DMA=0)

To disable FIFO (To disable both transmit FIFO and received FIFO)

- When the IBSR:RSA bit is set to "0", after receiving acknowledgement, the interface sets the interrupt flag (IBCR:INT) to "1" if the SSR:TDRE bit is set to "1" and waits while maintaining SCL at LOW. Writing "0" to the interrupt flag sets the interrupt flag to "0", which releases wait. If the SSR:TDRE bit is set to "0", the interface generates a clock on SCL upon reception of ACK without setting the interrupt flag to "1".
- When the IBSR:RSA bit is set to "1", after receiving a reserved address (before acknowledgement), the interface sets the interrupt flag (IBCR:INT) to "1" and waits while maintaining SCL at LOW. After reading from the RDR register, setting the IBCR:ACKE bit and transmit data and writing "0" to the interrupt flag causes the interrupt flag to be set to "0", which releases wait.
- The received acknowledgement is set to the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

To enable FIFO

- Before setting "1" to the IBCR:MSS bit, it is needed to set the following for FIFO.
  - When transmitting to a slave (the data direction bit=0), data including the slave address must be set to transmit FIFO.
  - When receiving data from a slave (the data direction bit=1), the FIFO Byte Register must be set with the number of data sets to be received, and dummy data must be written to the Transmit Data Register for the slave address, data direction bit and the data volume for the number of bytes to be received.
- When the IBSR:RSA bit is set to "0", after receiving acknowledgement and if it is ACK, the interface transmits/receives data according to the data direction bit without setting the interrupt flag (IBCR:INT) to "1" (with no wait occurring). If it is NACK, the interface sets the interrupt flag (IBCR:INT) to "1", and waits while maintaining SCL at LOW.
- The received acknowledgement is stored in the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

■ When DMA mode is enabled (SSR:DMA=1)

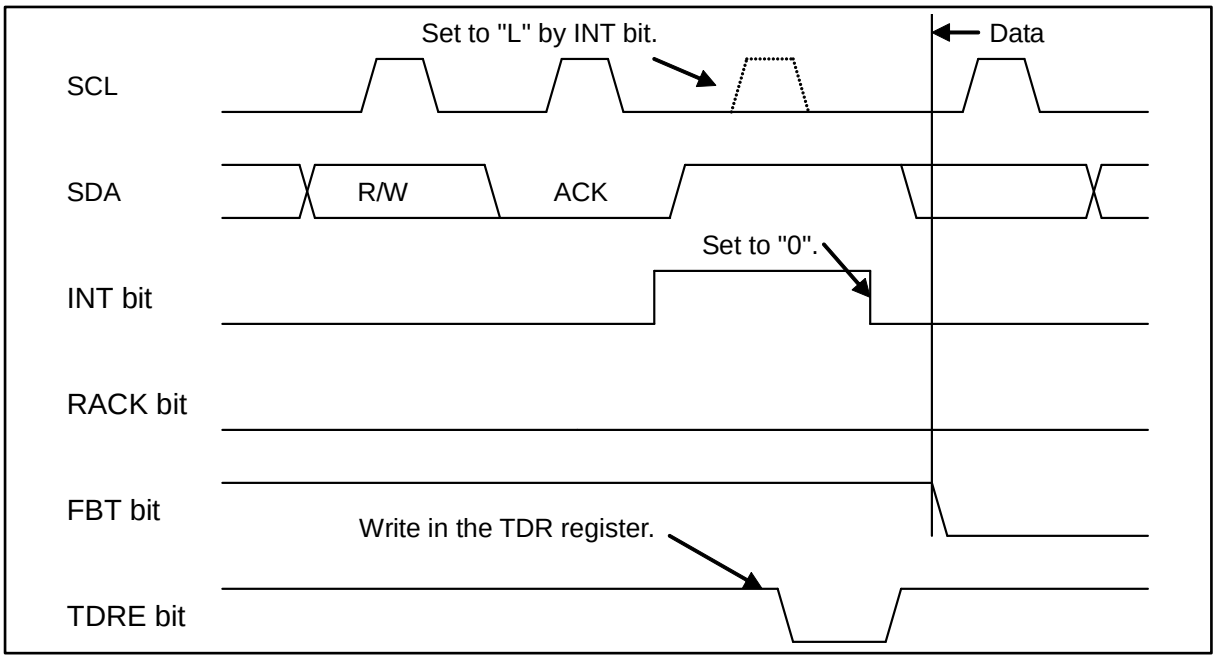
To disable FIFO (To disable both transmit FIFO and received FIFO)

- When the IBSR:RSA bit is set to "0", after receiving acknowledgement, the interface sets the transmit bus idle flag (SSR:TBI) to "1" if the SSR:TDRE bit is set to "1" and waits while maintaining SCL at LOW. Writing data to be transmitted to the TDR register causes the transmit bus idle flag to be set to "0", which releases wait. If the SSR:TDRE bit is set to "0", the interface generates a clock on SCL upon reception of ACK without setting the transmit bus idle flag (SSR:TBI) to "1".
- When the IBSR:RSA bit is set to "1", after receiving a reserved address (before acknowledgement), the interface sets the interrupt flag (IBCR:INT) to "1" and waits while maintaining SCL at LOW. After reading from the RDR register, setting the IBCR:ACKE bit and transmit data and writing "0" to the interrupt flag causes the interrupt flag to be set to "0", which releases wait.
- The received acknowledgement is set to the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

To enable FIFO

- Before setting "1" to the IBCR:MSS bit, it is needed to set the following for FIFO.
  - When transmitting to a slave (the data direction bit=0), data including the slave address must be set to transmit FIFO.
  - When receiving data from a slave (the data direction bit=1), the FIFO Byte Register must be set with the number of data sets to be received, and dummy data must be written to the Transmit Data Register for the slave address, data direction bit and the data volume for the number of bytes to be received.
- When the IBSR:RSA bit is set to "0", after receiving acknowledgement and if it is ACK, the interface transmits/receives data according to the data direction bit without setting the interrupt flag (IBCR:INT) to "1" (with no wait occurring). If it is NACK, the interface sets the interrupt flag (IBCR:INT) to "1", and waits while maintaining SCL at LOW.
- The received acknowledgement is stored in the IBSR:RACK bit. The interface checks the IBSR:RACK bit during wait, and, in case of NACK, it writes "0" to the IBCR:MSS bit or "1" to the IBCR:SCC bit to generate a stop condition or iteration start condition. At this time, the IBCR:INT bit is cleared to "0" automatically.

Figure 2-7 Acknowledgement (when FIFO is disabled, IBSR:RSA="0", and ACK response is selected)



The following describes the wait timing for an address.

- After receiving acknowledgment if the IBSR:RSA bit is "0".
- Before receiving acknowledgment if the IBSR:RSA bit is "1".

Not dependent on the setting of the IBCR:WSEL.

Figure 2-8 Acknowledgement (when FIFO is disabled, IBSR:RSA="0", and NACK response is selected)

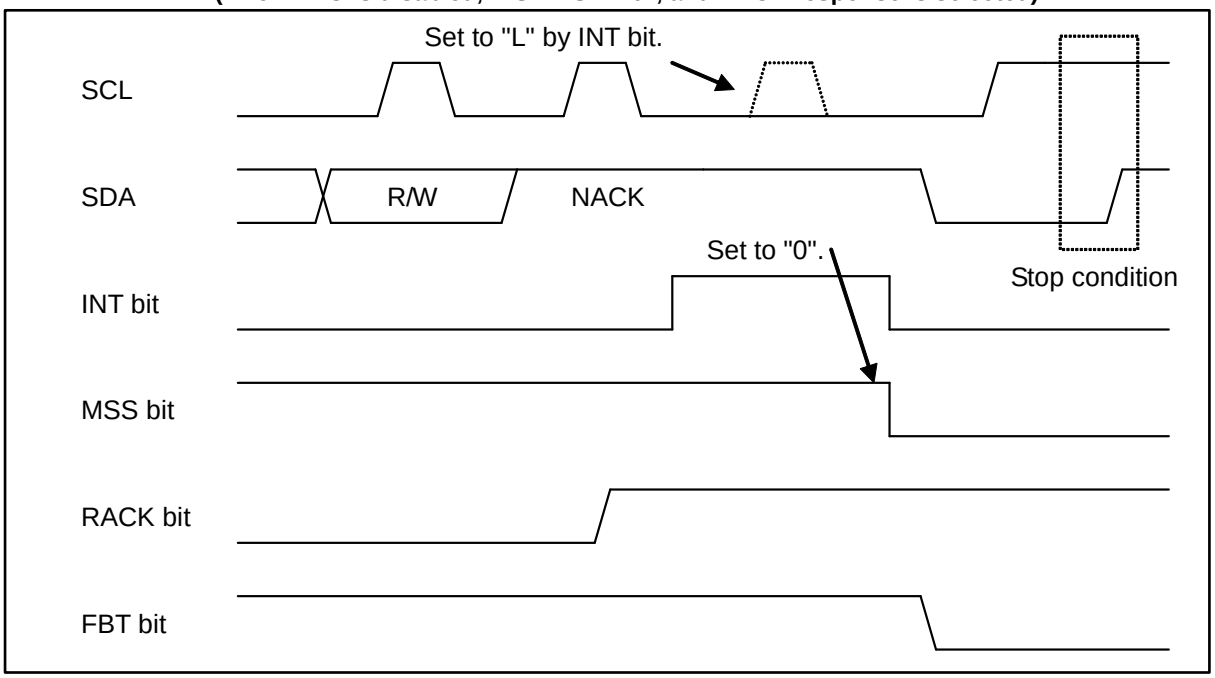


Figure 2-9 Acknowledgement
(when FIFO is disabled, IBSR:RSA="1", and ACK response is selected)

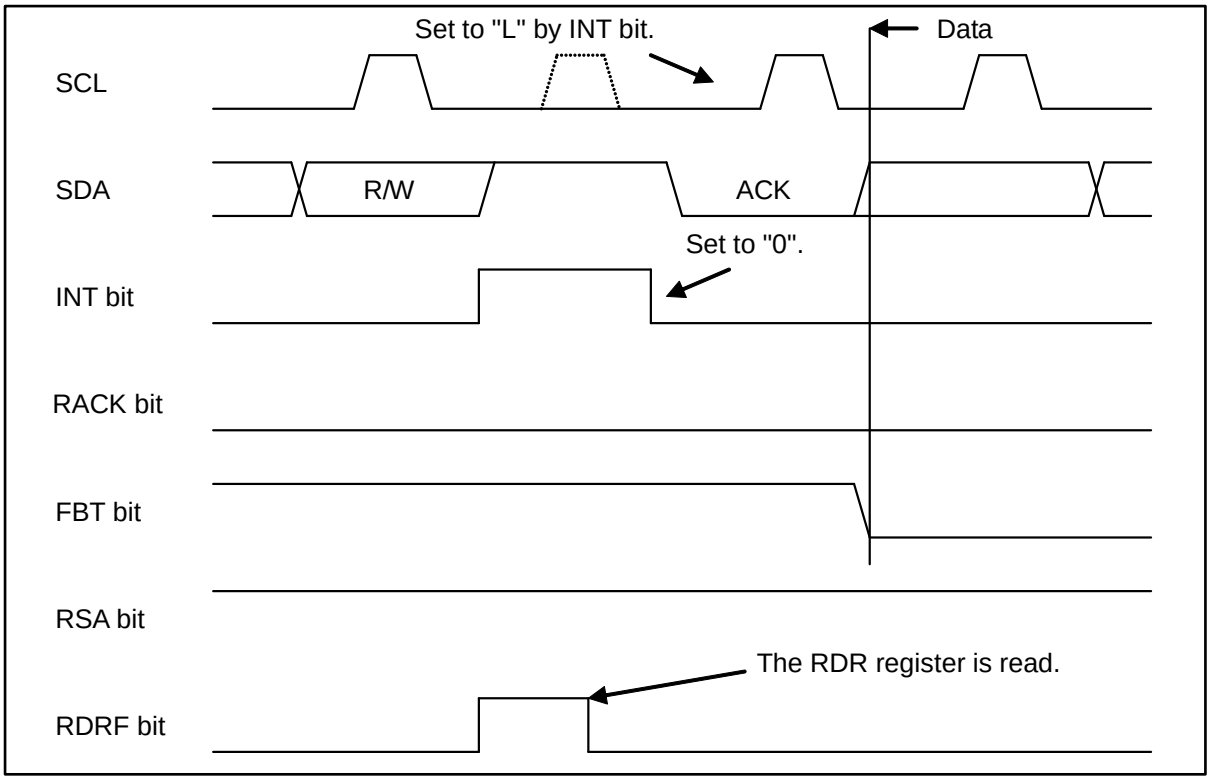


Figure 2-10 Acknowledgement
(when FIFO is disabled, IBSR:RSA="1", and NACK response is selected)

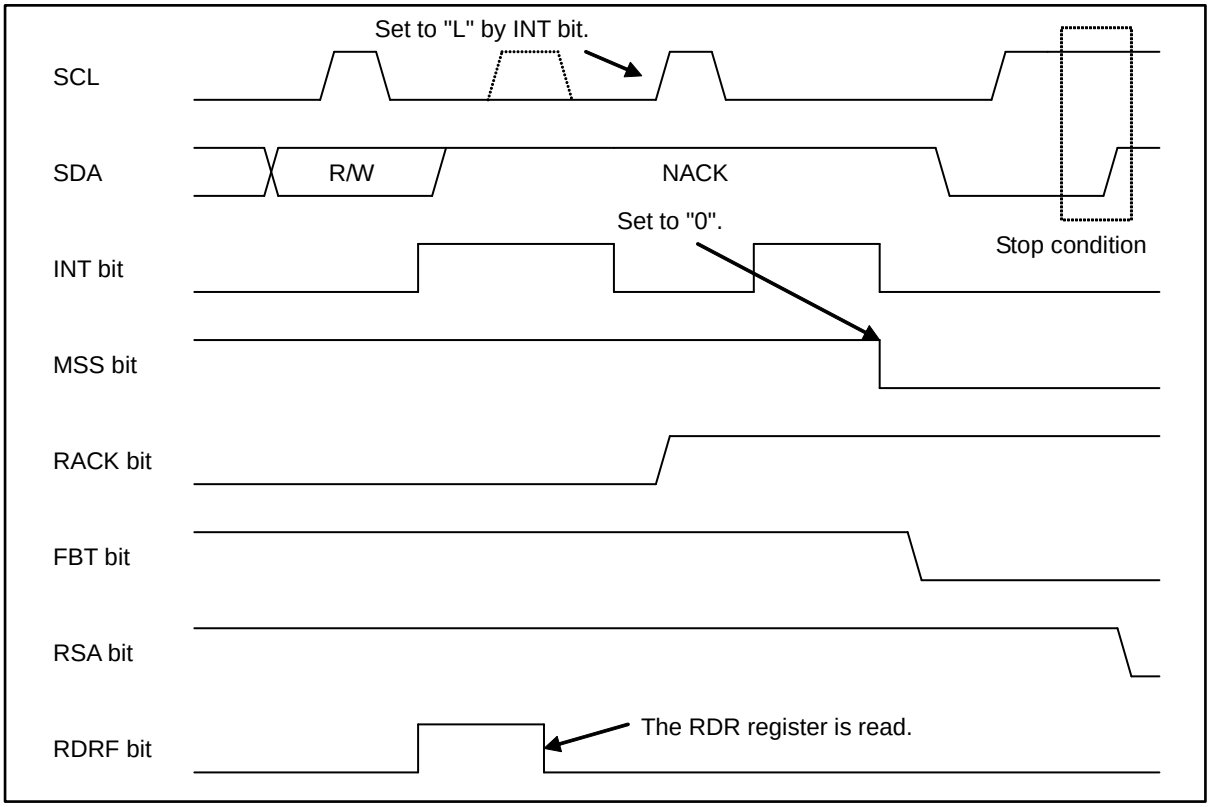
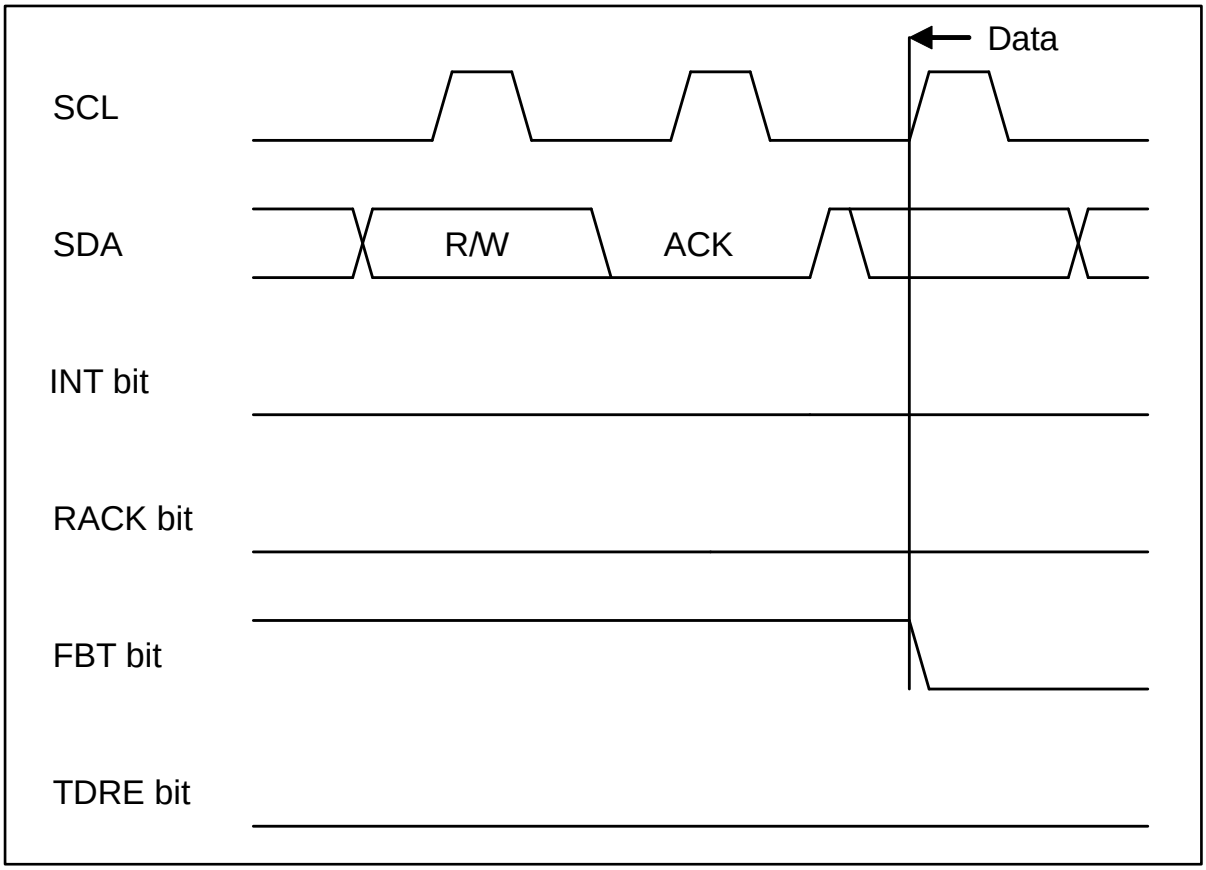


Figure 2-11 Acknowledgement (when FIFO is enabled, transmit FIFO has data, received FIFO has no data, IBSR:RSA=0, and ACK response is selected)



Data transmission by the master

When the data direction bit (R/W) is set to "0", data are transmitted from the master. The slave gives response either with ACK or NACK for each one-byte transmission.

The following shows the wait timing by IBCR:WSEL setting.

Table 2-4 IBCR:WSEL bit status for master data transmission when DMA mode is disabled (SSR:DMA=0)

WSEL bit	Operation
0	<When FIFO is not used> After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit or upon detection of arbitration lost, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state. <When FIFO is used> Starts the wait state by setting the interrupt flag (IBCR:INT) to "1" after acknowledgement upon detection of arbitration lost or when no more valid data remain in the Transmit Data Register (SSR:TDRE=1).
1	<When FIFO is not used> After the second byte, after the master has transmitted one-byte data with "1" set for the SSR:TDRE bit or upon detection of arbitration lost, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state. <When FIFO is used> Starts the wait state by setting the interrupt flag (IBCR:INT) to "1" when data transmission has taken place after detection of arbitration lost or no more valid data in the Transmit Data Register (SSR:TDRE=1).

Table 2-5 IBCR:WSEL bit status for master data transmission when DMA mode is enabled (SSR:DMA=1)

WSEL bit	Operation
0	<When FIFO is not used> After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit, the transmit bus idle flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state. <When FIFO is used> Starts the wait state by setting the transmit bus idle flag (SSR:TBI) to "1" after acknowledgment when no more valid data remain in the Transmit Data Register (SSR:TDRE=1).
1	<When FIFO is not used> After the second byte, after the master has transmitted one-byte data with "1" set for the SSR:TDRE bit, the transmit bus idle flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state. <When FIFO is used> Starts the wait state by setting the transmit bus idle flag (SSR:TBI) to "1" after the master has transmitted one-byte data when no more valid data remain in the Transmit Data Register (SSR:TDRE=1).

In the following case, however, the interrupt flag (IBCR:INT) is set after acknowledgement, regardless of the IBCR:WSEL setting:

- If NACK is received when the stop condition (IBCR:MSS=0, ACT=1) is not set.

The following shows an example procedure for transmitting data to a slave.

■ Data Transmission to slave when DMA mode is disabled (SSR:DMA=0)

1. To transmit data to an address other than the reserved:

- When transmit FIFO is disabled:

1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
2. ACK is received after the Slave Address setting is transmitted, and then the interrupt flag (IBCR:INT) is set to "1".
3. Writes transmit data to the TDR register.
4. Writes "0" to the interrupt flag (IBCR:INT) upon updating of the IBCR:WSEL bit and releases the wait state of the I^2C bus.
5. After transmitting one byte, the interrupt flag is set to "1", which puts the I^2C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL=0, and directly after transmitting one byte in case IBCR:WSEL=1. Repeats steps 3 to 5 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, another interrupt is generated after receiving acknowledgement and the bus enters the wait state.
6. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

- When transmit FIFO is enabled:

1. Writes Slave Address (including the data direction bit) and transmit data to the TDR register.
2. Writes "1" to the IBCR:MSS bit upon setting of the IBCR:WSEL bit.
3. If NACK is received during transmission, sets the interrupt flag (IBCR:INT) to "1" immediately after that to put the I^2C bus in the wait state. If ACK responses are received for all bytes, sets the interrupt flag to "1" according to the setting of IBCR:WSEL after the last byte is transmitted to put the I^2C bus in the wait state.
4. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

2. To transmit data to a reserved address:

- When transmit FIFO is disabled:

1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
3. Reads from the RDR register and confirms the reserved address.(*1)
4. Writes transmit data to the TDR register.
5. Writes "0" to the interrupt flag (IBCR:INT) upon updating of the IBCR:WSEL bit and releases the wait state of the I^2C bus.
6. After transmitting one byte, the interrupt flag is set to "1", which puts the I^2C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL=0, and directly after transmitting one byte in case IBCR:WSEL=1. Repeats steps 4 to 6 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, another interrupt is generated after receiving acknowledgement and the bus enters the wait state.
7. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

- When transmit FIFO is enabled:

1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
3. Reads from the RDR register and confirms the reserved address.(*1)
4. Writes all transmit data to the TDR register (until transmit FIFO becomes full if it is the case).
5. If NACK is received during transmission, the interrupt flag (IBCR:INT) is set to "1" immediately after that to put the I^2C bus in the wait state.
   If ACK responses are received for all bytes, sets the interrupt flag to "1" according to the setting of IBCR:WSEL after the last byte is transmitted to put the I^2C bus in the wait state.
6. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

*1 : When any one of the following conditions is met, the IBCR:ACKE and IBCR:WSEL bits must be set to "1" and to check which is needed for the next data, operation as a master or operation as a slave.

- Multi-master mode is activated and the reserved address is a general call.
- Arbitration lost has been detected and the interface may operate as a slave.

■ Data Transmission to slave when DMA mode is enabled (SSR:DMA=1)

1. To transmit data to an address other than the reserved:

   − When transmit FIFO is disabled:

   1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
   2. ACK is received after the Slave Address setting is transmitted, and then the transmit bus idle flag (SSR:TBI) is set to "1".
   3. Writes data to be transmitted to the TDR register to release the wait state of the I^2C bus.
   4. After transmitting one byte, sets the transmit bus idle flag (SSR:TBI) to "1" to put the I^2C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL=0, and directly after transmitting one byte in case of IBCR:WSEL=1.
   5. Writes data to be transmitted to the TDR register to release the wait state of the I^2C bus.
   6. After transmitting one byte, sets the transmit bus idle flag to "1" to put the I^2C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL=0, and directly after transmitting one byte in case of IBCR:WSEL=1. Repeats steps 5 to 6 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, the interrupt flag (IBCR:INT) is set to "1" after receiving acknowledgement and the bus enters the wait state.
   7. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"*2 to generate the stop condition or iteration start condition.

   − When transmit FIFO is enabled:

   1. Writes Slave Address (including the data direction bit) and transmit data to the TDR register.
   2. Writes "1" to the IBCR:MSS bit upon setting of the IBCR:WSEL bit.
   3. If NACK is received during transmission, sets the interrupt flag (IBCR:INT) to "1" immediately after that to put the I^2C bus in the wait state. If ACK responses are received for all bytes, sets the transmit bus idle flag (SSR:TBI) to "1" according to the setting of IBCR:WSEL after the last byte is transmitted to put the I^2C bus in the wait state.
   4. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"*2 to generate the stop condition or iteration start condition.

2. To transmit data to a reserved address:

   − When transmit FIFO is disabled:

   1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
   2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
   3. Reads from the RDR register and confirms the reserved address.(*1)
   4. Writes transmit data to the TDR register.
   5. Writes "0" to the interrupt flag (IBCR:INT) upon updating of the IBCR:WSEL bit and releases the wait state of the I^2C bus.
   6. After transmitting one byte, the interrupt flag is set to "1", which puts the I^2C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL=0, and directly after transmitting one byte in case IBCR:WSEL=1.
   7. Writes data to be transmitted to the TDR register to release the wait state of the I^2C bus.
   8. After transmitting one byte, sets the transmit bus idle flag to "1" to put the I^2C bus in the wait state after receiving acknowledgment in case of IBCR:WSEL=0, and directly after transmitting one byte in case of IBCR:WSEL=1. Repeats steps 7 to 8 until all the specified number of data sets have been transmitted. However, if NACK is received after the wait state is released when IBCR:WSEL=1, the interrupt flag (IBCR:INT) is set to "1" after receiving acknowledgement and the bus enters the wait state.
   9. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"*2 to generate the stop condition or iteration start condition.

- When transmit FIFO is enabled:

1. Sets the reserved address for Slave Address in the TDR register and writes "1" to the IBCR:MSS bit.
2. After the Slave Address setting is transmitted, the interrupt flag (IBCR:INT) is set to "1".
3. Reads from the RDR register and confirms the reserved address.(*1)
4. Writes all transmit data to the TDR register (until transmit FIFO becomes full if it is the case).
5. If NACK is received during transmission, sets the interrupt flag (IBCR:INT) to "1" immediately after that to put the I^2C bus in the wait state. If ACK responses are received for all bytes, sets the interrupt flag (IBCR:INT) to "1" according to the setting of IBCR:WSEL after the last byte is transmitted, which puts the I^2C bus in the wait state.
6. Sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1"(*2) to generate the stop condition or iteration start condition.

*1 : When any one of the following conditions is met, the IBCR:ACKE and IBCR:WSEL bits must be set to "1" and to check which is needed for the next data, operation as a master or operation as a slave.

- Multi-master mode is activated and the reserved address is a general call.
- Arbitration lost has been detected and the interface may operate as a slave.

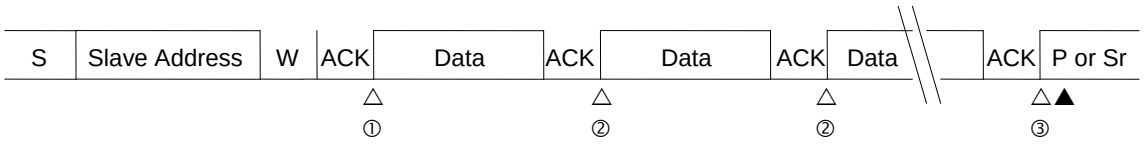
*2 : When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.

1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

Notes:

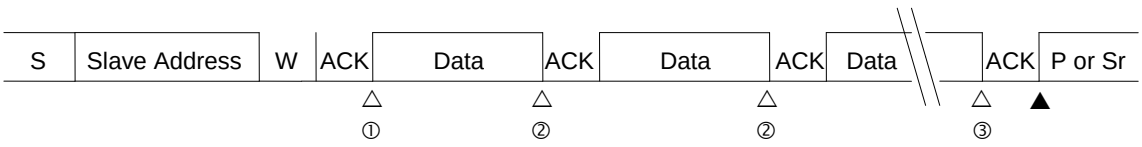
- *When seven-bit slave address detection is enabled (ISBA:SAEN=1), it is prohibited to specify a seven-bit slave address in master mode.*
- *To change the IBCR register during transmission/reception, do so when the interrupt flag (IBCR:INT) is "1".*
- *If the IBCR:WSEL bit is changed, the update is used as a condition for generating the transmit bus idle flag (SSR:TBI) when the interrupt flag (IBCR:INT) is enabled and DMA mode is also enabled (SSR:DMA=1) for the next data.*
- *The master operates as follows when transmit data are written to the TDR register during data transmission with SSR:TDRE set to "1" and an ACK response is detected.*
- *When DMA mode is disabled (SSR:DMA=0), the interrupt flag (IBCR:INT) does not attain "1", and the written data are transmitted.*
- *When DMA mode is enabled (SSR:DMA=1), the transmit bus idle flag (SSR:TBI) does not attain "1", and the written data are transmitted.*
- *The master operates as follows when transmit data are written to the TDR register during data reception with SSR:TDRE set to "1" and an ACK response is detected.*
- *When DMA mode is disabled (SSR:DMA=0), the interrupt flag (IBCR:INT) does not attain "1" and only SSR:RDRF attains "1" (when received FIFO is enabled, and the number of bytes set in the FBYTE register have been received).*
- *When DMA mode is enabled (SSR:DMA=1), the transmit bus idle flag (SSR:TBI) does not attain "1" and only SSR:RDRF attains "1" (when received FIFO is enabled, and the number of bytes set in the FBYTE register have been received).*

Figure 2-12 Master mode transmit interrupt 1 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



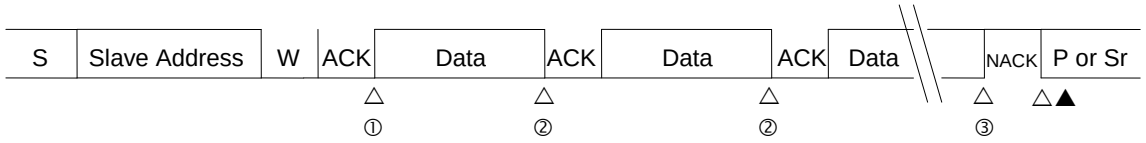
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
③ An interrupt occurs when a single byte is sent and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-13 Master mode transmit interrupt 2 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", ACK response)



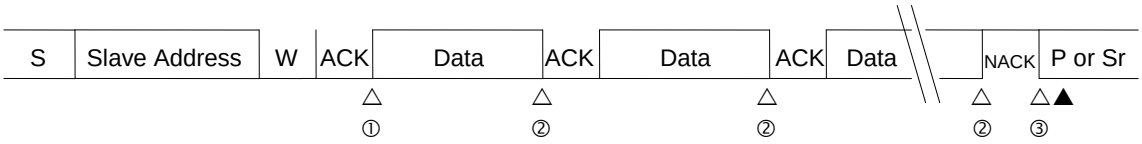
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register, and the INT bit is set to "0".
③ An interrupt occurs when a single byte is sent.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-14 Master mode transmit interrupt 3 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



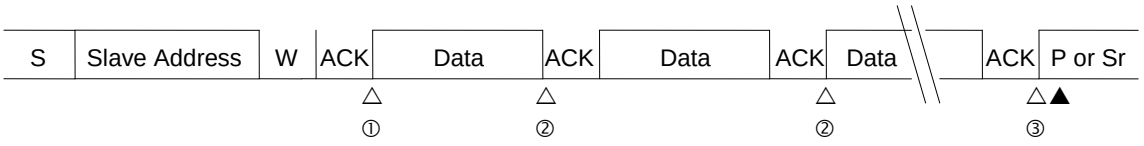
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register, and the INT bit is set to "0".
③ An interrupt occurs when a single byte is sent.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-15 Master mode transmit interrupt 4 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK response during transmission)



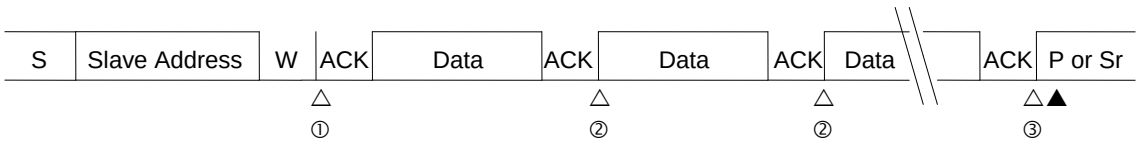
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register, and the INT bit is set to "0".
③ An interrupt occurs when a NACK is responded.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-16 Master mode transmit interrupt 5 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1" -> "0", IBSR:RSA="0", ACK response)



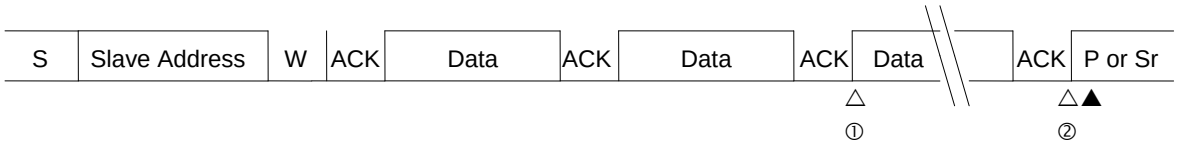
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the send buffer, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent.
- The send data is written in the send buffer, and both WSEL and INT bits are set to "0".
③ An interrupt occurs when a single byte is sent.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-17 Master mode transmit interrupt 6 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



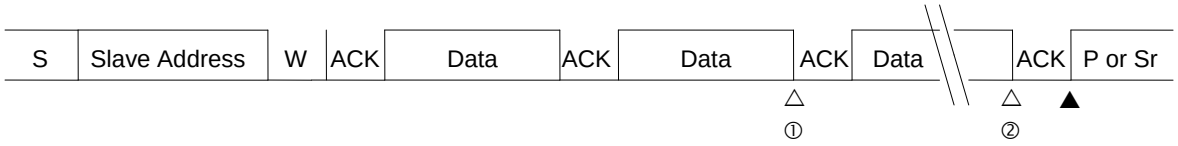
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs when the slave address (reserved address) is sent, a direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
③ An interrupt occurs when a single byte is sent and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-18 Master mode transmit interrupt 7 by enabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0", ACK response)



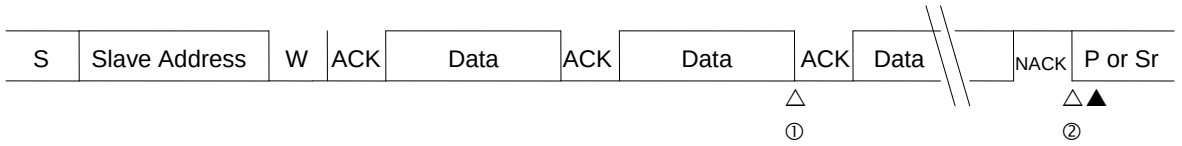
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer, and INT bit is set to "0".
② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied) and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-19 Master mode transmit interrupt 8 by enabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



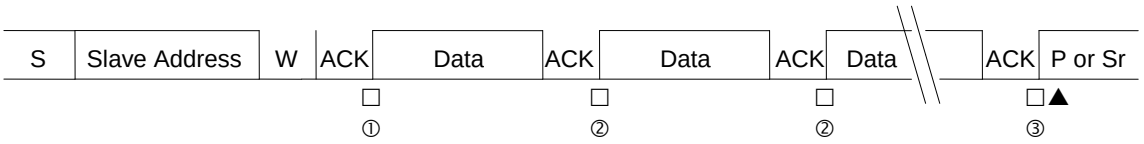
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer, and INT bit is set to "0".
② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied).
- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-20 Master mode transmit interrupt 9 by enabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



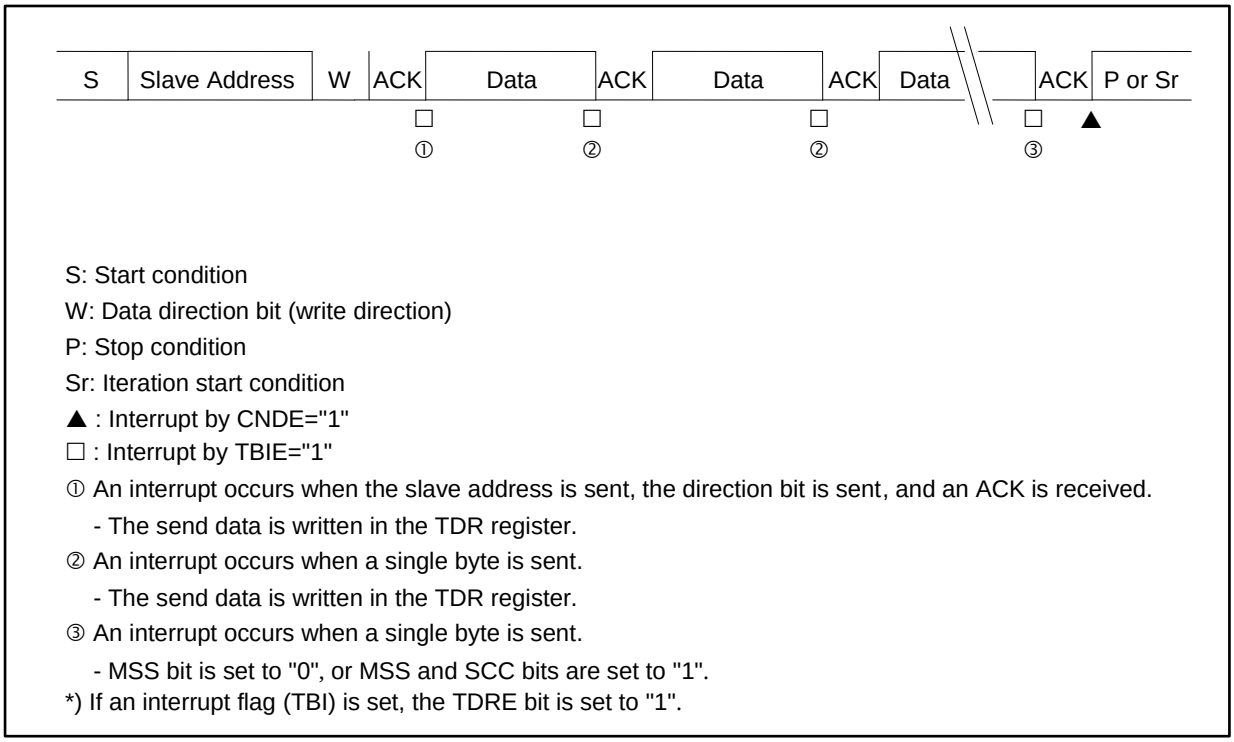
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer, and INT bit is set to "0".
② An interrupt occurs when a NACK is responded.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-21 Master mode transmit interrupt 10 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



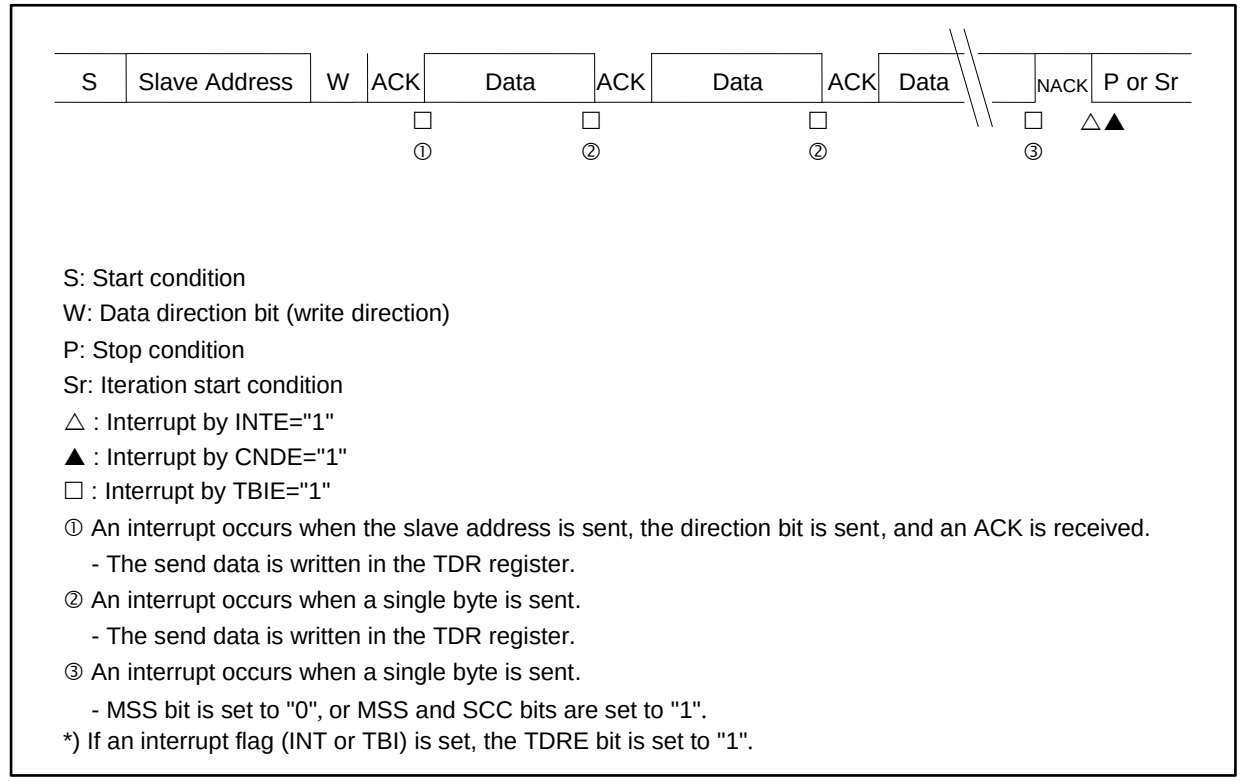
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register.
② An interrupt occurs when a single byte is sent and an ACK is received.
- The send data is written in the TDR register.
③ An interrupt occurs when a single byte is sent and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (TBI) is set, the TDRE bit is set to "1".

Figure 2-22 Master mode transmit interrupt 11 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", ACK response)



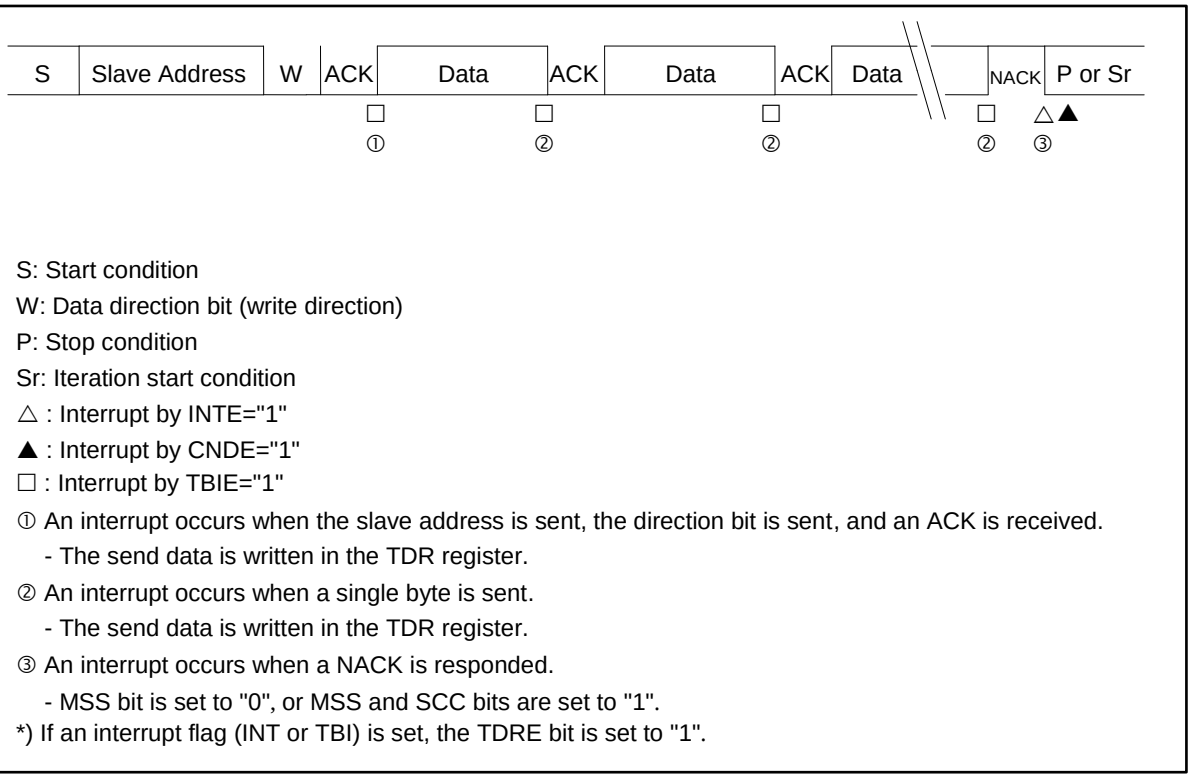
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register.
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register.
③ An interrupt occurs when a single byte is sent.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (TBI) is set, the TDRE bit is set to "1".

Figure 2-23 Master mode transmit interrupt 12 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



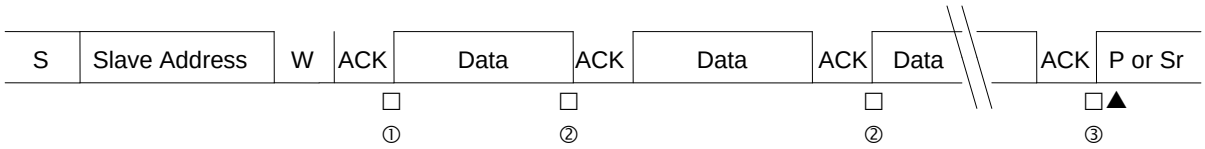
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register.
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register.
③ An interrupt occurs when a single byte is sent.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-24 Master mode transmit interrupt 13 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK response during transmission)



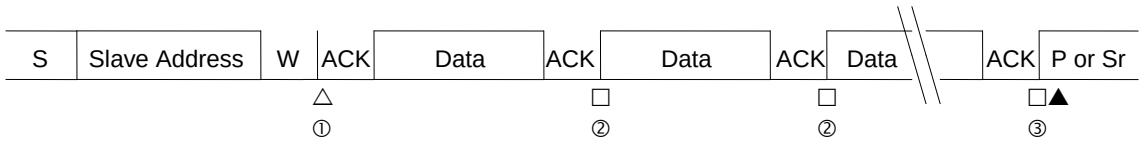
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
☐ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register.
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register.
③ An interrupt occurs when a NACK is responded.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-25 Master mode transmit interrupt 14 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1" -> "0", IBSR:RSA="0", ACK response)



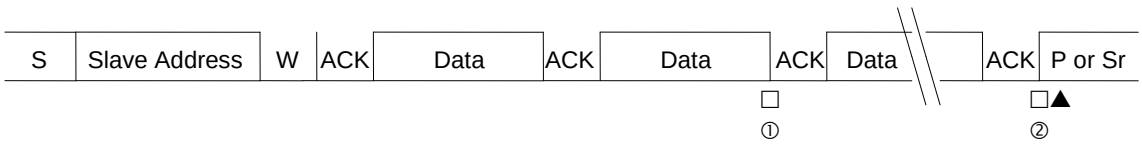
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
▲ : Interrupt by CNDE="1"
☐ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the send buffer.
② An interrupt occurs when a single byte is sent.
- The WSEL bit is set to "0" and the send data is written in the send buffer.
③ An interrupt occurs when a single byte is sent.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (TBIE) is set, the TDRE bit is set to "1".

Figure 2-26 Master mode transmit interrupt 15 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")



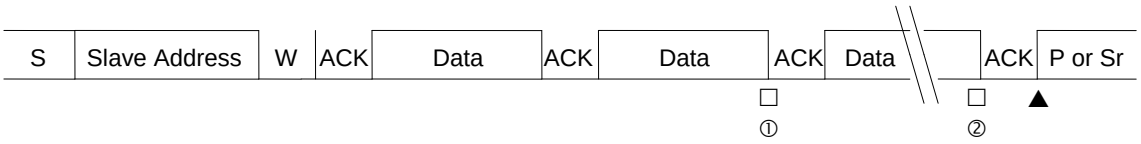
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address (reserved address) is sent, a direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".
② An interrupt occurs when a single byte is sent and an ACK is received.
- The send data is written in the TDR register.
③ An interrupt occurs when a single byte is sent and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".
*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-27 Master mode transmit interrupt 16 by enabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK response)



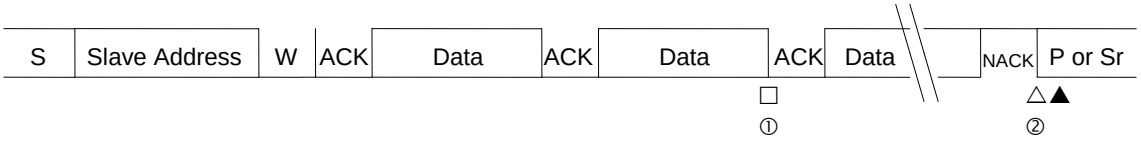
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer.
② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied) and an ACK is received.
- MSS bit is set to "0", or MSS and SCC bits are set to "1".

Figure 2-28 Master mode transmit interrupt 17 by enabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer.
② An interrupt occurs when the last byte is sent (the Send FIFO buffer is emptied).
- MSS bit is set to "0", or both MSS and SCC bits are set to "1".

Figure 2-29 Master mode transmit interrupt 18 by enabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK response)



S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer.
② An interrupt occurs when a NACK is responded.
- MSS bit is set to "0", or both MSS and SCC bits are set to "1".

Data reception by the master

■ When DMA mode is disabled (SSR:DMA=0)

When the data direction bit (R/W) is set to "1", the master receives data transmitted from a slave.

When FIFO is disabled, the master operates as follows.

- If the SSR:TDRE bit is set to "1", wait is generated (IBCR:INT=1, SSR:RDRF=1) each time one byte is received . At this time, an ACK or NACK response is returned, according to the setting of the ACKE bit in the IBCR register, before wait if the IBCR:WSEL bit is "1", and after wait if the IBCR:WSEL bit is "0".
- If the SSR:TDRE bit is set to "0", the next data is received without generating wait (IBCR:INT=0) when an ACK response is set for the ACKE bit in the IBCR register while wait is generated when the NACK response is set (IBCR:INT=1).

When FIFO is enabled, the SSR:RDRF bit is set to "1" upon reception of data in the same number of bytes set for the number of bytes to be received. The interrupt flag is set to "1" when the SSR:TDRE bit is "1", which puts the I^2C bus in the wait state. At this time, acknowledgement operates as follows. Even if NACK is output, it is stored in received FIFO as received data.

- In case of IBCR:WSEL=0, an NACK response is returned when the SSR:TDRE bit is set to "1" if NACK is set for the ACKE bit.
- In case of IBCR:WSEL=1, the interrupt flag is set to "1" after receiving the final byte, which generates wait. During that wait, an ACK or NACK response is returned according to the IBCR:ACKE setting after the IBCR:ACKE bit is set and the interrupt flag is cleared to "0".

For interrupt-generated wait, refer to the following.

Table 2-6 IBCR:WSEL bit status for master data reception when DMA mode is disabled (SSR:DMA=0)

WSEL bit	Operation
0	After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state.
1	After the second byte, after the master has received one-byte data with "1" set for the SSR:TDRE bit, the interrupt flag (IBCR:INT) is set to "1" and SCL to LOW for the wait state.

The following shows an example procedure for receiving data from a slave.

- When received FIFO is disabled:

1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
2. ACK is received after the Slave Address setting is transmitted, and then the interrupt flag (IBCR:INT) is set to "1".
3. Writes "0" to the interrupt flag bit (IBCR:INT) upon updating of the IBCR:WSEL bit to release the wait state of the I^2C bus.
4. After receiving one byte, sets the interrupt flag to "1" to set the I^2C bus in the wait state after transmitting acknowledgment in case of IBCR:WSEL=0 and directly after receiving one byte in case of IBCR:WSEL=1. Repeats steps 3 to 4 until all the specified number of data sets have been received.
5. After receiving the last data, outputs NACK and sets the IBCR:MSS bit to "0" or sets the IBCR:SCC bit to "1" to generate the stop condition or iteration start condition.

- When transmit/received FIFO is enabled:

1. Sets the number of bytes to be received to the FBYTE register.
2. Writes Slave Address (including the data direction bit) and dummy data in the number of bytes to be received to the TDR register.
3. Writes "1" to the IBCR:MSS bit.
4. An ACK response is returned and data reception continues as long as the SSR:TDRE bit stays "0". During that reception operation, SSR:RDRF is set to "1" when the number of bytes set up in FBYTE have been received. When SSR:RDRF is set to "1", starts reading from the RDR register.
5. When SSR:TDRE bit is "1", sets the interrupt flag to "1" to set the I^2C bus in the wait state after outputting NACK if IBCR:WSEL=0, and directly after one-byte reception if IBCR:WSEL=1.
6. In case of IBCR:WSEL=1, sets the IBCR:ACKE bit to "0". In case of IBCR:WSEL=0, no setting is needed for the IBCR:ACKE bit, Setting the IBCR:MSS bit to "0" or setting the IBCR:SCC bit to "1" generates the stop condition or iteration start condition.

■ When DMA mode is enabled (SSR:DMA=1)

When the data direction bit (R/W) is set to "1", the master receives data transmitted from a slave.

When FIFO is disabled, the master operates as follows.

- If the SSR:TDRE bit is set to "1", wait is generated (SSR:TBI=1, SSR:RDRF=1) each time one byte is received. At this time, an ACK or NACK response is returned, according to the setting of the ACKE bit in the IBCR register, before wait if the IBCR:WSEL bit is "1", and after wait if the IBCR:WSEL bit is "0".
- If the SSR:TDRE bit is set to "0", wait is generated (SSR:RDRF=1) each time one byte is received. At this time, an ACK or NACK response is returned, according to the setting of the ACKE bit in the IBCR register, before wait if the IBCR:WSEL bit is "1", and after wait if the IBCR:WSEL bit is "0".

When FIFO is enabled, the SSR:RDRF bit is set upon reception of data in the same number of bytes set for the number of bytes to be received. The transmit bus idle flag (SSR:TBI) is set when the SSR:TDRE bit is "1", which puts the I^2C bus in the wait state. At this time, acknowledgement operates as follows. Even if NACK is output, it is stored in received FIFO as received data.

- In case of IBCR:WSEL=0, an NACK response is returned when the SSR:TDRE bit is set to "1" if NACK is set for the ACKE bit.
- In case of IBCR:WSEL=1, wait is generated (SSR:TBI=1) after receiving the last byte. During that wait, the master sets the IBCR:ACKE bit and returns ACK or NACK response, according to the IBCR:ACKE setting, after clearing the transmit bus idle flag (SSR:TBI).

For interrupt-generated wait, refer to the following.

Table 2-7 IBCR:WSEL bit status for master data reception when DMA mode is enabled (SSR:DMA=1)

WSEL bit	Operation
0	After the second byte, after acknowledgement with "1" set for the SSR:TDRE bit, the transmit bus idle flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state. After the second byte, after acknowledgement with received FIFO is unused, if the received data full flag (SSR:RDRF) is set to "1", SCL is set to LOW for the wait state.
1	After the second byte, after the master has received one-byte data with "1" set for the SSR:TDRE bit, the interrupt flag (SSR:TBI) is set to "1" and SCL to LOW for the wait state. After the second byte, after the received data full flag (SSR:RDRF) is set to "1" when received FIFO is not used, SCL is set to LOW for the wait state.

The following shows an example procedure for receiving data from a slave.

- When received FIFO is disabled:

1. Sets Slave Address (including the data direction bit) to the TDR register and writes "1" to the IBCR:MSS bit.
2. ACK is received after the Slave Address setting is transmitted, and then the transmit bus idle flag (SSR:TBI) is set to "1".
3. Writes data to be transmitted to the TDR register to release the wait state of the I^2C bus.
4. After one byte is received, sets the transmit bus idle flag (SSR:TBI) and the received data full flag (SSR:RDRF)(*2) to "1" under the following conditions to put the I^2C bus in the wait state.
   - In case of IBCR:WSEL=0, after transmitting acknowledgement
   - In case of IBCR:WSEL=1, after receiving one byte
5. Updates the IBCR:WSEL bit, reads from the RDR register and writes dummy data to the TDR register.
6. After one byte is received, sets the transmit bus idle flag (SSR:TBI) and the received data full flag (SSR:RDRF)(*2) to "1" under the following conditions to put the I^2C bus in the wait state.
   - In case of IBCR:WSEL=0, after transmitting acknowledgement
   - In case of IBCR:WSEL=1, after receiving one byte

   Repeats steps 5 to 6 until all the specified number of data sets have been received.
7. After receiving the last data, outputs NACK and sets the IBCR:MSS bit to "0" or sets the IBCR:SCC(*1) bit to "1" to generate the stop condition or iteration start condition.

- When transmit/received FIFO is enabled:

1. Sets the number of bytes to be received to the FBYTE register.
2. Writes Slave Address (including the data direction bit) and dummy data in the number of bytes to be received to the TDR register.
3. In case of IBCR:WSEL=0, sets NACK for the ACKE bit, and writes "1" to the IBCR:MSS bit.
4. An ACK response is returned and data reception continues as long as the SSR:TDRE bit stays "0". During that reception operation, SSR:RDRF is set to "1" when the number of bytes set up in FBYTE have been received. When SSR:RDRF is set to "1", starts reading from the RDR register.
5. When the SSR:TDRE bit is set to "1", sets the interrupt flag to "1" to set the I^2C bus in the wait state after outputting NACK if IBCR:WSEL=0. In case of IBCR:WSEL=1, directly after one byte is received, sets the transmit bus idle flag (SSR:TBI) to "1" to put the I^2C bus in the wait state.
6. In case of IBCR:WSEL=1, sets the IBCR:ACKE bit to "0". In case of IBCR:WSEL=0, no setting is needed for the IBCR:ACKE bit, Set the IBCR:MSS bit to "0" or set the IBCR:SCC(*1) bit to "1" to generate the stop condition or iteration start condition.

*1 : When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.

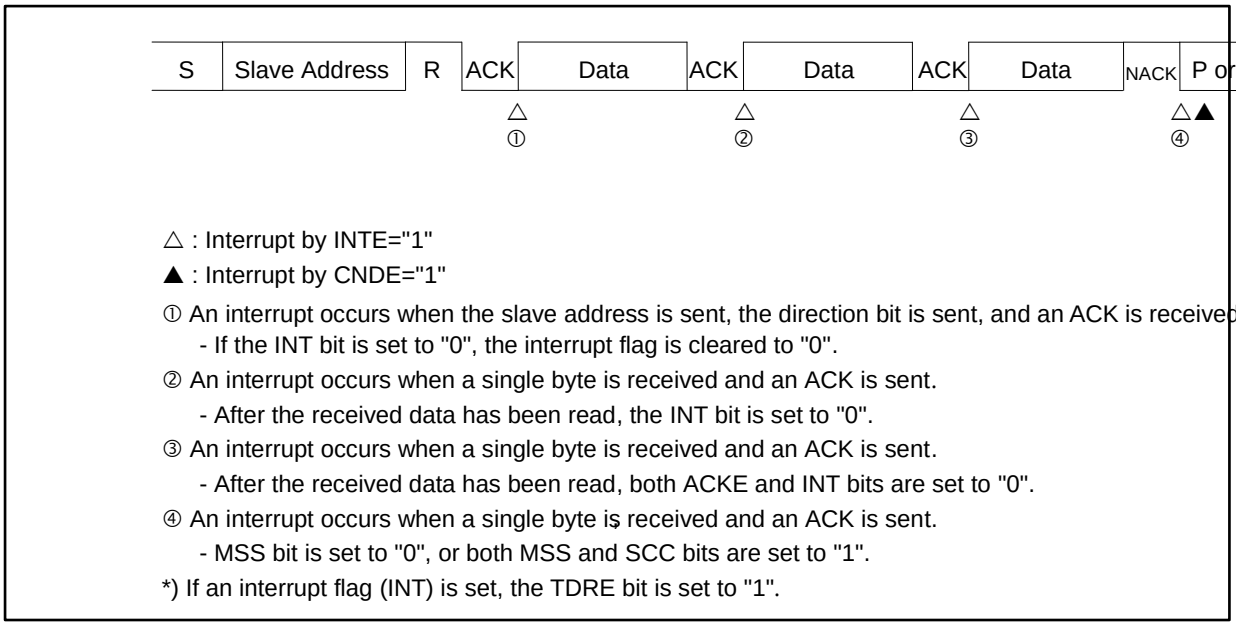
1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

*2 : Directly after receiving one byte, the received data full flag (SSR:RDRF) is set to "1" regardless of the setting for IBCR:WSEL. When the received data full flag (SSR:RDRF) is set to "1" in the second byte or later, put the I^2C bus in the wait state after transmitting acknowledgment in case of IBCR:WSEL=0, and directly after receiving one byte in case of IBCR:WSEL=1.

Notes:

- *When seven-bit slave address detection is enabled (ISBA:SAEN=1), it is prohibited to specify a seven-bit slave address in master mode.*
- *When SSR:TDRE is "0", even if an overrun error occurs, acknowledgement is output according to the setting for the IBCR:ACKE bit, and then the next process should follow.*
- *To change the IBCR register during transmission/reception, do so when the interrupt flag (IBCR:INT) is "1" or when the transmit bus idle flag (SSR:TBI) is "1" during DMA mode being enabled (SSR:DMA=1).*
- *In the master mode reception with DMA disabled (SSR:DMA=0), write dummy data to the TDR register, and then, if the SSR:TDRE bit is "0" when the interrupt flag (IBCR:INT) is turned to "1", receive the next data with the interrupt flag (IBCR:INT) kept at "0".*
- *In the master mode reception with DMA enabled (SSR:DMA=1), write dummy data to the TDR register, and then, if the SSR:TDRE bit is "0" when the transmit bus idle flag (SSR:TBI) is turned to "1", receive the next data with the transmit bus idle flag (SSR:TBI) kept at "0".*
- *To receive data when received FIFO is enabled and IBCR:WSEL=0, the SSR:RDRF bit is set to "1" after receiving the last bit and the interrupt flag (IBCR:INT) is set to "1" after transmitting ACK.*

Figure 2-30 Master mode received interrupt 1 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- If the INT bit is set to "0", the interrupt flag is cleared to "0".

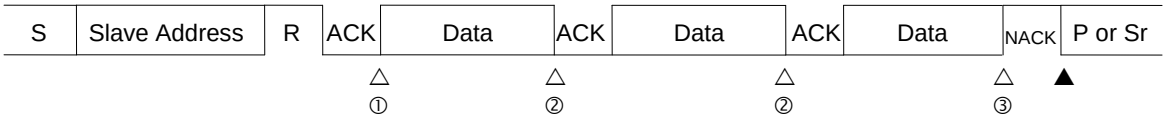
② An interrupt occurs when a single byte is received and an ACK is sent.
- After the received data has been read, the INT bit is set to "0".

③ An interrupt occurs when a single byte is received and an ACK is sent.
- After the received data has been read, both ACKE and INT bits are set to "0".

④ An interrupt occurs when a single byte is received and an ACK is sent.
- MSS bit is set to "0", or both MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-31 Master mode received interrupt 2 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- If the INT bit is set to "0", the interrupt flag is cleared to "0".

② An interrupt occurs when a single byte is received.
- After the received data has been read, the INT bit is set to "0".

③ An interrupt occurs when a single byte is received.
- After the received data has been read, ACKE bit are set to "0". MSS bit is set to "0", or both MSS and SCC bits are set to "1".

*) If an interrupt flag (INT) is set, the TDRE bit is set to "1".

Figure 2-32 Master mode received interrupt 3 by enabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")

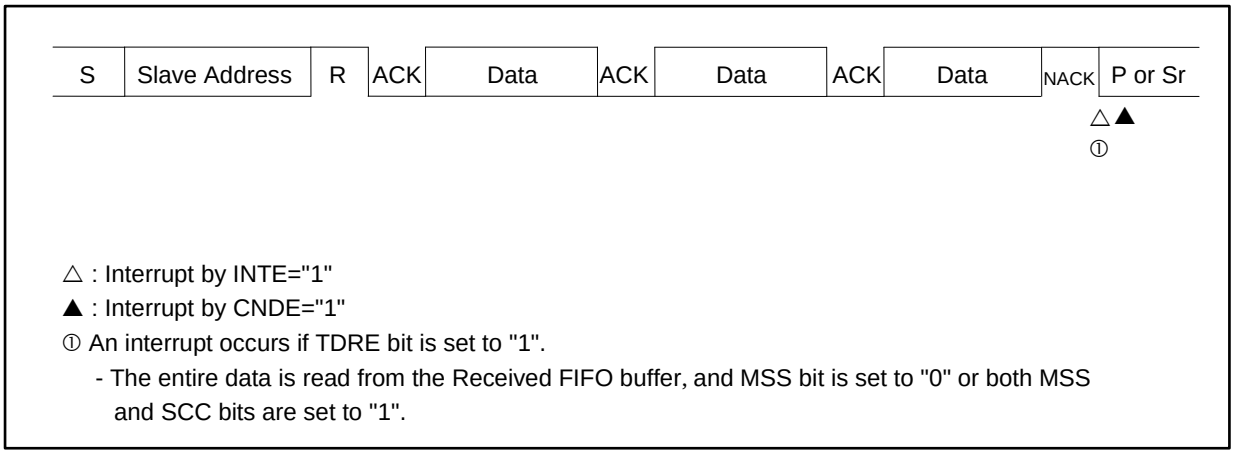


Figure 2-33 Master mode received interrupt 4 by enabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")

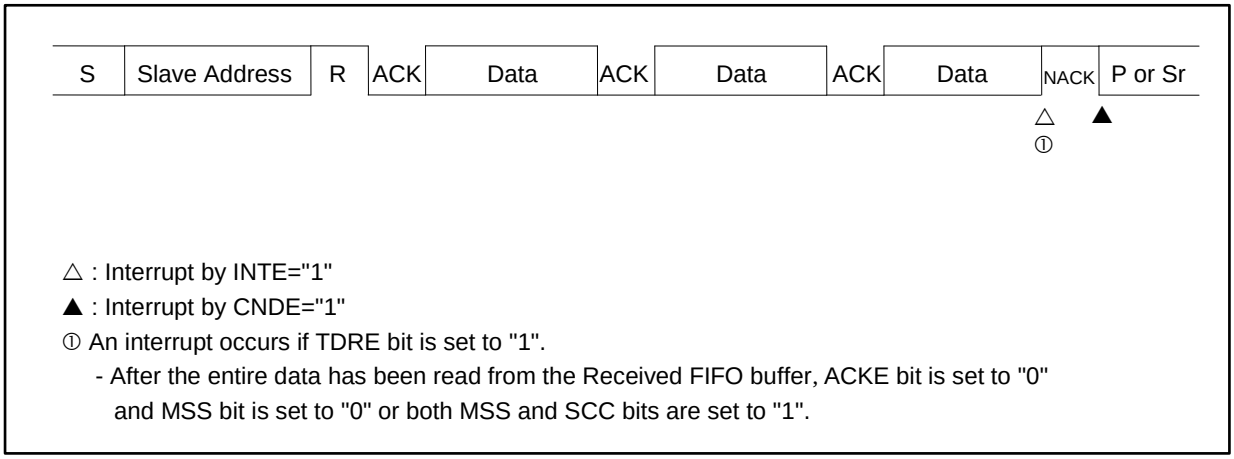
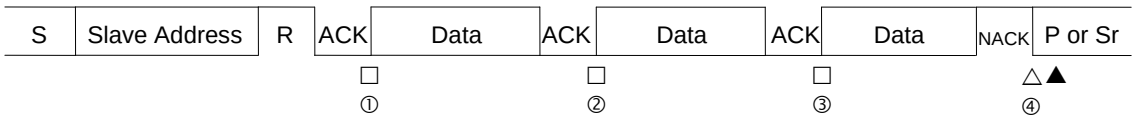


Figure 2-34 Master mode received interrupt 5 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- Dummy data is written in the TDR register.

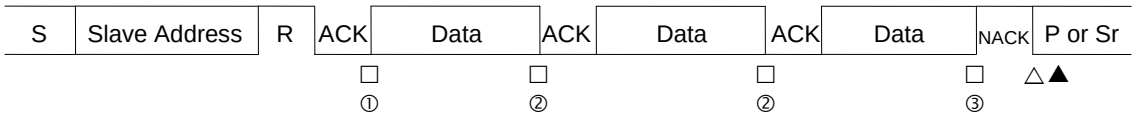
② An interrupt occurs when a single byte is received and an ACK is sent.
- After the received data has been read, the dummy data is written in the TDR register.

③ An interrupt occurs when a single byte is received and an ACK is sent.
- After the received data has been read, the ACKE bit are set to "0" and the dummy data is written in the TDR register.

④ An interrupt occurs when a single byte is received and an ACK is sent.
- MSS bit is set to "0", or both MSS and SCC bits are set to "1".

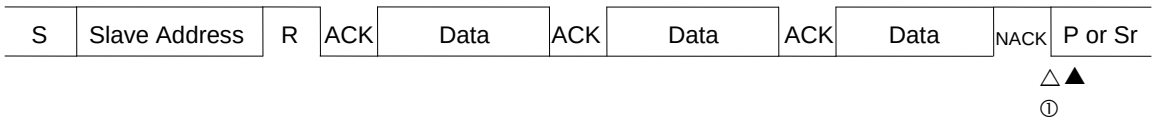
*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-35 Master mode received interrupt 6 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



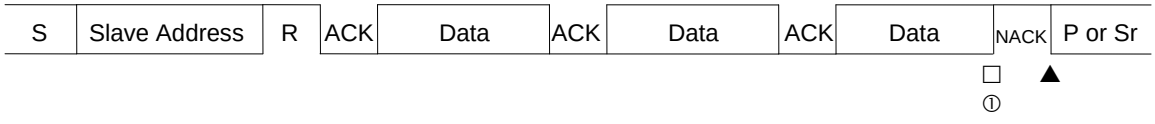
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
□ : Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- Dummy data is written in the TDR register.
② An interrupt occurs when a single byte is received.
- After the received data has been read, the dummy data is written in the TDR register.
③ An interrupt occurs when a single byte is received.
- After the received data has been read, ACKE bit are set to "0" MSS bit is set to "0", or both MSS and SCC bits are set to "1".
*) If an interrupt flag (INT or TBI) is set, the TDRE bit is set to "1".

Figure 2-36 Master mode received interrupt 7 by enabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")



△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs if TDRE bit is set to "1".
- The entire data is read from the Received FIFO buffer, and MSS bit is set to "0" or both MSS and SCC bits are set to "1".

Figure 2-37 Master mode received interrupt 8 by enabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



□ : Interrupt by TBIE="1"
▲ : Interrupt by CNDE="1"
① An interrupt occurs if TDRE bit is set to "1".
- After the entire data has been read from the Received FIFO buffer, ACKE bit is set to "0" and MSS bit is set to "0" or both MSS and SCC bits are set to "1".

Arbitration lost

If the master receives the data different from sent data, due to collision of data from another master, the master judges the situation as arbitration lost. At this time, the IBCR:MSS bit is set to "0" and the IBSR:AL bit to "1", enabling operation in slave mode.

The IBSR:AL bit can be cleared to "0" under the following conditions:

- The IBCR:MSS bit is set to "1".
- The IBCR:INT bit is set to "0".
- The IBSR:SPC bit is set to "0" when the IBSR:AL bit and IBSR:SPC bit are "1".
- The I^2C interface operation is disabled (ISMK:EN=0).

Upon an occurrence of arbitration lost, the interrupt flag (IBCR:INT) is set to "1" according to the setting of the IBCR:WSEL bit, and sets SCL of the I^2C bus to LOW.

Wait state for master mode

When both conditions below are satisfied, master mode is put in the wait state while the IBSR:BB bit stays "1". After the IBSR:BB bit attains "0", start condition is transmitted.

- When the IBCR:MSS is set to "1" while the IBSR:BB bit is "1"
- When the interface is not operating as a slave

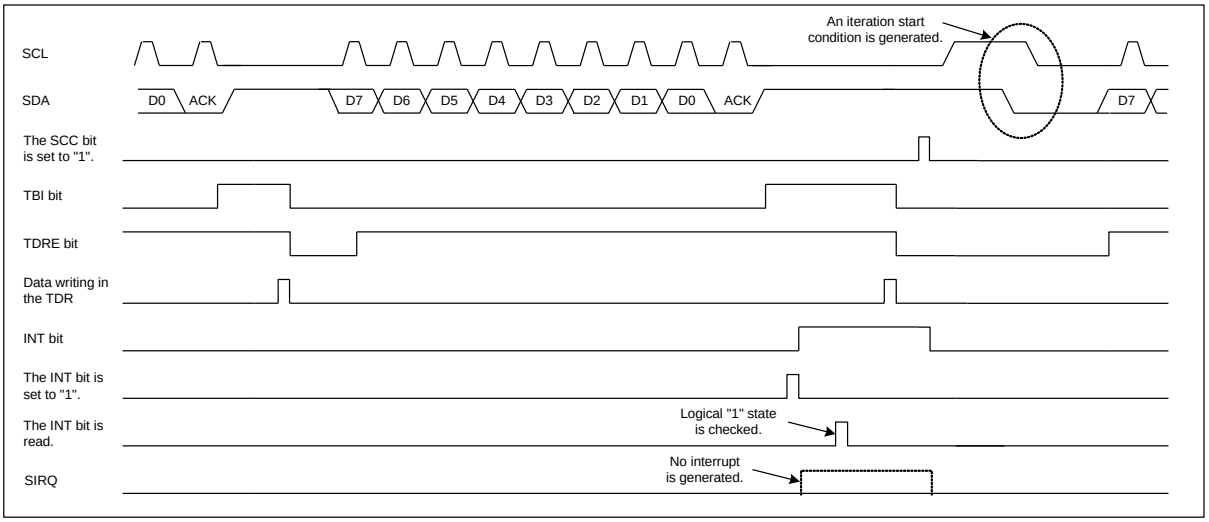
Refer to the IBCR:MSS bit and IBCR:ACT bit to check if master mode is in the wait state or not (in the wait state if the IBCR:MSS=1 and IBCR:ACT=0). After setting the IBCR:MSS bit to "1" and to operate in slave mode, set the IBSR:AL bit to "1", the IBCR:MSS bit to "0", and the IBCR:ACT bit to "1".

Issuing iteration start condition when DMA mode is enabled (SSR:DMA=1)

When writing a slave address to the TDR register while the transmit bus is idle (SSR:TBI=1) and the interrupt flag (IBCR:INT) is "0", transmission starts and the iteration start condition cannot be issued. Therefore, to issue the iteration start condition while the transmit bus is idle (SSR:TBI=1) and the interrupt flag (IBCR:INT) is "0", follow the steps below.

1. Set the IBCR:INT bit to "1". At this time, no SIRQ interrupt is generated.
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Issue the iteration start condition (IBCR:SCC=1).

Figure 2-38 Issuing iteration start condition when DMA mode is enabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK response)



2.3 Slave mode

If the (iteration) start condition is detected and a combination of the ISBA and ISMK registers matches the received address, the interface outputs an ACK response and acts in slave mode.

Note:

- *When EIBCR:BEC set to "0", If a start condition is detected again while transferring address data after a start condition is detected or while transferring bit2 to bit19 (acknowledge bits), the next data cannot be received since a bus error (IBCR:BER = 1) is detected and reception is stopped. In such a case, a start condition must be retransmitted from the master after clearing the interrupt flag (IBCR:INT).*

Slave address match detection

After the (iteration) start condition is detected, subsequent seven bits are received as the address. For each of the bits that are set to "1" in the ISMK register, the ISBA register is compared with the received address. If they match, ACK is output.

Table 2-8 Operation immediately after outputting acknowledgement to a slave address

Transmit FIFO	Received FIFO	Transmit FIFO status	Received FIFO status	Data direction bit (R/W)	Operation immediately after receiving acknowledgement	
					Acknowledgement: ACK	Acknowledgement: NACK
Disable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
				1		
Disable	Enable	-	Without data	0	Holds the IBCR:INT bit to "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
			With data		Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	
Enable	Disable	-	-	0	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
				1		
Enable	Enable	-	Without data	0	Holds the IBCR:INT bit to "0" without the wait state.	Holds the IBCR:INT bit to "0" without the wait state.
			With data		Sets the IBCR:INT bit to "1" with the wait state.	
			-	1	If the SSR:TDRE bit is set to "1", the interface sets the IBCR:INT bit to "1" and waits. If the SSR:TDRE bit is set to "0", IBCR:INT bit stays "0" without the wait state.	

- Detection of reserved address

  If the first byte matches the reserved address ("0000xxxx" or "1111xxxx"), the value of 8th bit is received regardless of whether or not transmit/received FIFO is enabled, and the IBCR:INT bit is set to "1", causing the I^2C bus to be placed into the wait state. After the received data has been read, configure the following settings.

- To run the interface as a slave device, set the IBCR:ACKE bit to "1" and check the value of the data direction bit (IBSR:TRX). If the transmitting direction is set, write the transmit data to TDR, and clear the IBCR:INT bit. The interface then acts as a slave device.
- When not running the interface as a slave device, set the IBCR:ACKE bit to "0", and clear the IBCR:INT bit. After acknowledgement has been output, the interface does not act as a slave device.

Data direction bit

After receiving the address, the interface receives the data direction bit to determine whether to transmit or receive data. If this bit is "0", it means that data is transmitted from the master device, and the interface receives data as a slave device.

Reception in slave mode

If the received data matches the slave address and the data direction bit is "0", it means that data is received in slave mode. The following shows a procedure example to receive data in slave mode.

- When DMA mode is disabled (SSR:DMA=0)
  - When received FIFO is disabled:

1. After transmitting ACK, set the interrupt flag (IBCR:INT) to "1", and place the I^2C bus into the wait state. Based on the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits, judge that the event is an interrupt by a slave address match. Then write "1" to the IBCR:ACKE bit and "0" to the interrupt flag (IBCR:INT), and release the wait state of the I^2C bus (see Table 2-8).
2. After receiving 1-byte data, set the interrupt flag (IBCR:INT) to "1" according to setting of the IBCR:WSEL bit, and place the I^2C bus into the wait state.
3. Read the data received from the RDR register, set the IBCR:ACKE bit, write "0" to the interrupt flag (IBCR:INT), and release the wait state of the I^2C bus.
4. Repeat steps 2 and 3 to detect the stop or iteration start condition.

  - When received FIFO is enabled:

1. If NACK is detected or received FIFO becomes full, the interrupt flag (IBCR:INT) is set to "1", and the I^2C bus is placed into the wait state. If the stop or iteration start condition is detected, the interrupt flag (IBCR:INT) is not set to "1" (the I^2C bus is not placed into the wait state) by setting the IBSR:SPC and IBSR:RSC bits to "1". Received FIFO sets the SSR:RDRF bit to "1" when the set value of the FBYTE register matches the number of data sets received. If the SMR:RIE bit is then "1", a received interrupt is generated.
2. When the interrupt flag (IBCR:INT) is set to "1", read the received data from the RDR register. After all data has been read, write "0" to the interrupt flag to release the wait state of the I^2C bus. If the stop or iteration start condition is detected, read all the received data from the RDR register, and clear the IBSR:SPC or IBSR:RSC bit to "0".

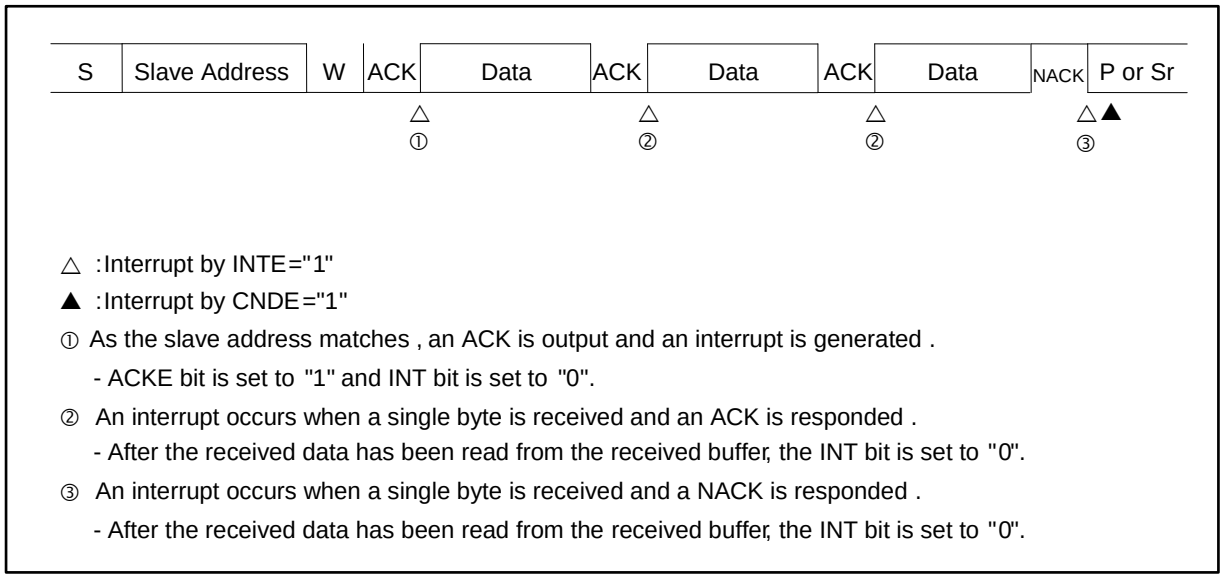
- When DMA mode is enabled (SSR:DMA=1)
  - When received FIFO is disabled:

1. After transmitting ACK, set the interrupt flag (IBCR:INT) to "1", and place the I^2C bus into the wait state. Based on the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits, judge that the event is an interrupt by a slave address match. Then write "1" to the IBCR:ACKE bit and "0" to the interrupt flag (IBCR:INT), and release the wait state of the I^2C bus (see Table 2-8).
2. Set "1" to the received data full flag (SSR:RDRF) immediately after receiving 1-byte data. When the received data full flag (SSR:RDRF) is set to "1", if IBCR:WSEL=0, place the I^2C bus into the wait state after transmitting acknowledgement. If IBCR:WSEL=1, place the I^2C bus into the wait state immediately after receiving the 1-byte data.
3. After setting the IBCR:ACKE bit, read the data received from the RDR register, and clear the received data full flag (SSR:RDRF) to "0" to release the wait state of the I^2C bus.
4. Repeat steps 2 and 3 to detect the stop or iteration start condition.

- When received FIFO is enabled:

1. If NACK is detected, the interrupt flag (IBCR:INT) is set to "1", and the I^2C bus is placed into the wait state. When received FIFO becomes full, place the I^2C bus into the wait state. If the stop or iteration start condition is detected, the IBSR:SPC and IBSR:RSC bits are set to "1", and the interrupt flag (IBCR:INT) is not set to "1" (the I^2C bus is not placed into the wait state). Received FIFO sets the SSR:RDRF bit to "1" when the set value of the FBYTE register matches the number of data sets received. If the SMR:RIE bit is then "1", a received interrupt is generated.
2. When the interrupt flag (IBCR:INT) is set to "1", read the received data from the RDR register. After all data has been read, write "0" to the interrupt flag to release the wait state of the I^2C bus. When received FIFO is full, release the wait state of the I^2C bus if the received data is read from the RDR register even once. If the stop or iteration start condition is detected, read all the received data from the RDR register, and clear the IBSR:SPC or IBSR:RSC bit to "0".

Figure 2-39 Slave mode received interrupt 1 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① As the slave address matches , an ACK is output and an interrupt is generated .

- ACKE bit is set to "1" and INT bit is set to "0".

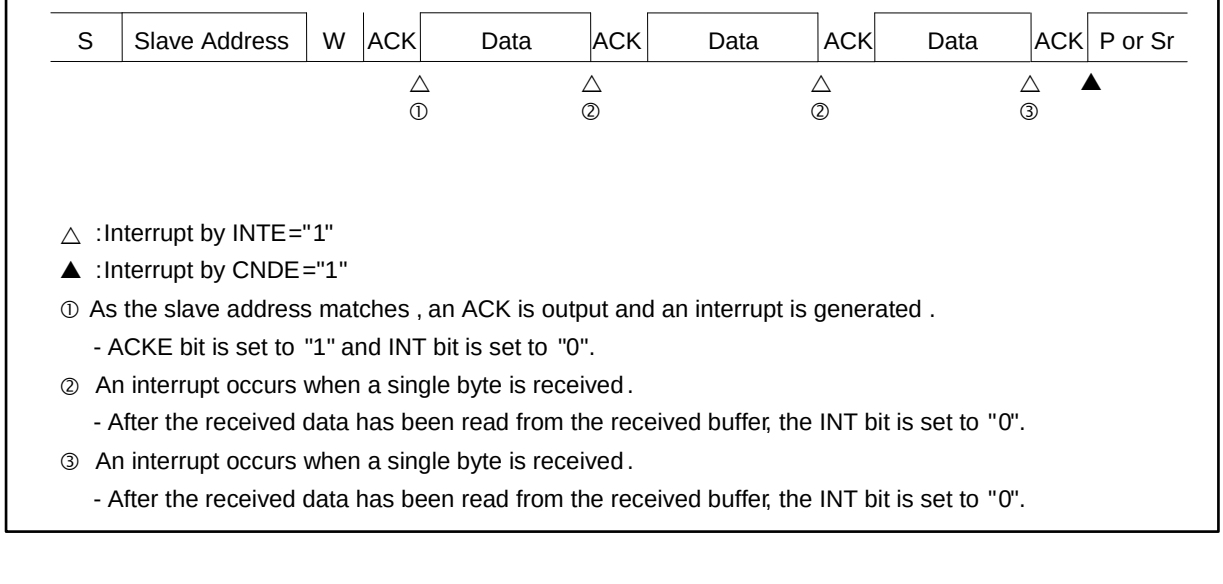
② An interrupt occurs when a single byte is received and an ACK is responded .

- After the received data has been read from the received buffer, the INT bit is set to "0".

③ An interrupt occurs when a single byte is received and a NACK is responded .

- After the received data has been read from the received buffer, the INT bit is set to "0".

Figure 2-40 Slave mode received interrupt 2 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① As the slave address matches , an ACK is output and an interrupt is generated .

- ACKE bit is set to "1" and INT bit is set to "0".

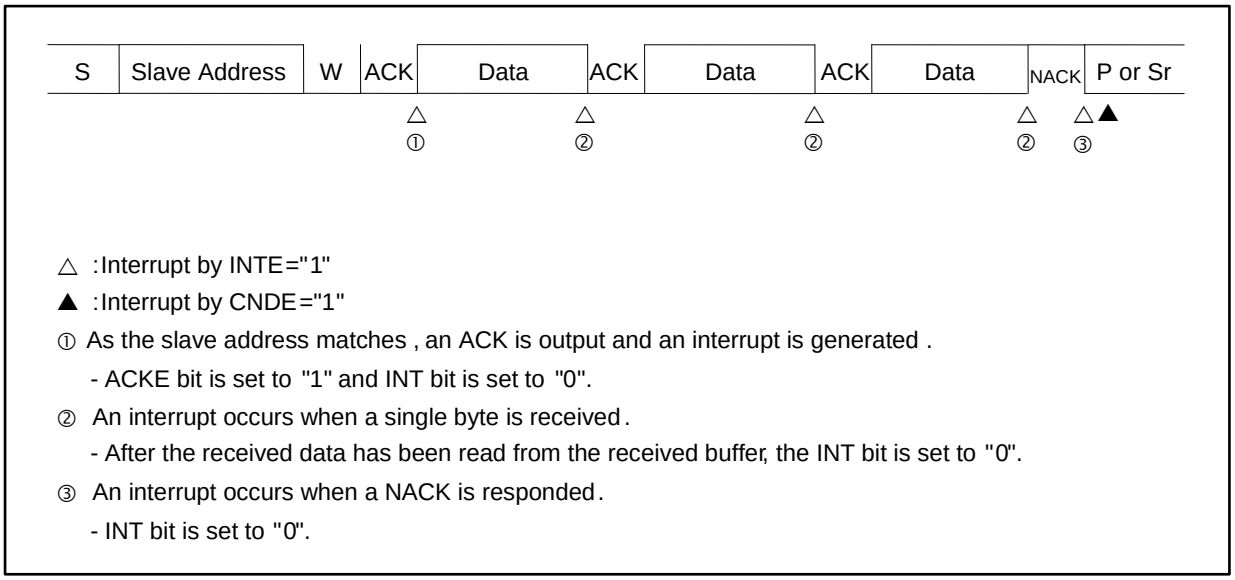
② An interrupt occurs when a single byte is received.

- After the received data has been read from the received buffer, the INT bit is set to "0".

③ An interrupt occurs when a single byte is received.

- After the received data has been read from the received buffer, the INT bit is set to "0".

Figure 2-41 Slave mode received interrupt 3 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① As the slave address matches , an ACK is output and an interrupt is generated .

- ACKE bit is set to "1" and INT bit is set to "0".

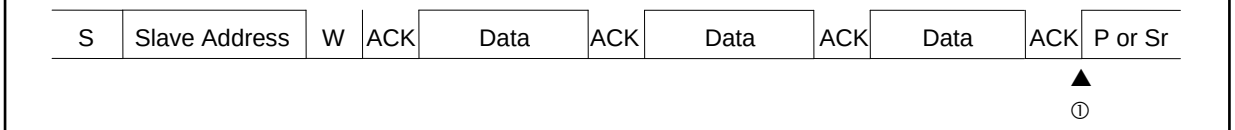
② An interrupt occurs when a single byte is received.

- After the received data has been read from the received buffer, the INT bit is set to "0".

③ An interrupt occurs when a NACK is responded.

- INT bit is set to "0".

Figure 2-42 Slave mode received interrupt 4 by enabling received FIFO (SSR:DMA="0", IBSR:RSA="0")



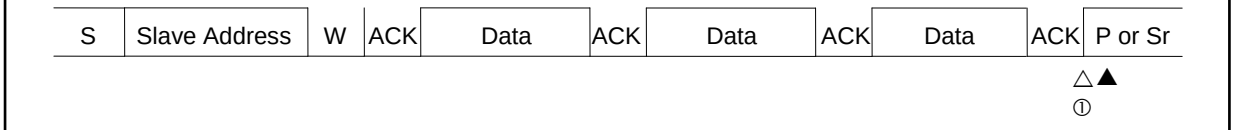
△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① An interrupt occurs when the stop condition or the iteration start condition is detected .

- The entire data is read from the Received FIFO buffer

Figure 2-43 Slave mode received interrupt 5 by enabling received FIFO (SSR:DMA="0", IBSR:RSA="0")



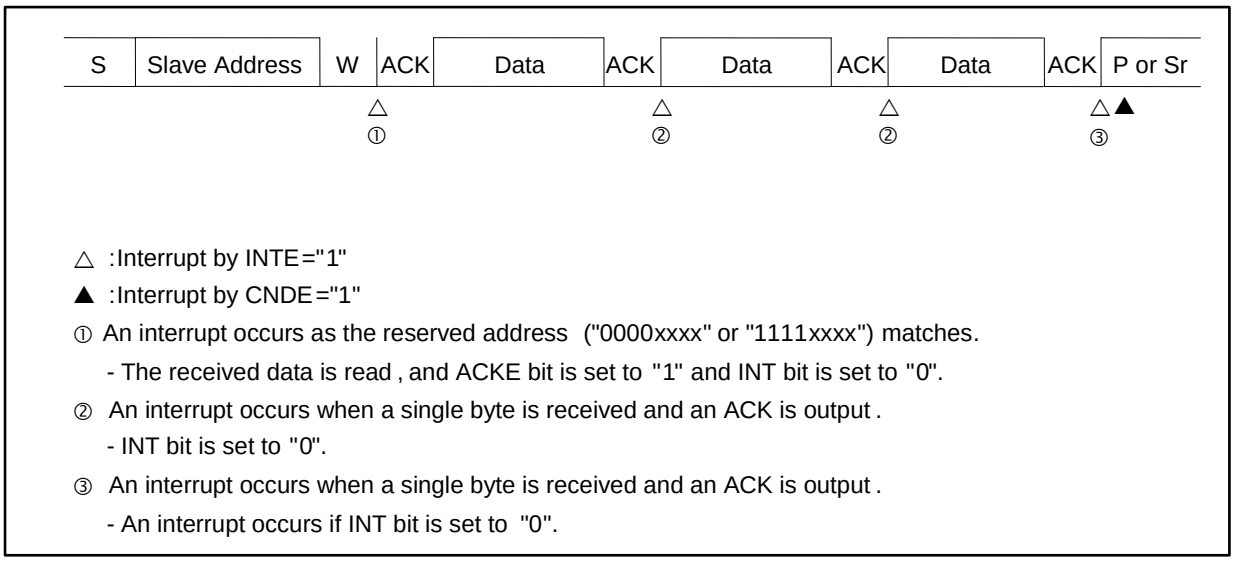
△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① An interrupt occurs when the Received FIFO buffer is filled with data

- The entire data is read from the Received FIFO buffer and the INT bit is set to "0".

Figure 2-44 Slave mode received interrupt 6 by disabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

① An interrupt occurs as the reserved address ("0000xxxx" or "1111xxxx") matches.

- The received data is read , and ACKE bit is set to "1" and INT bit is set to "0".

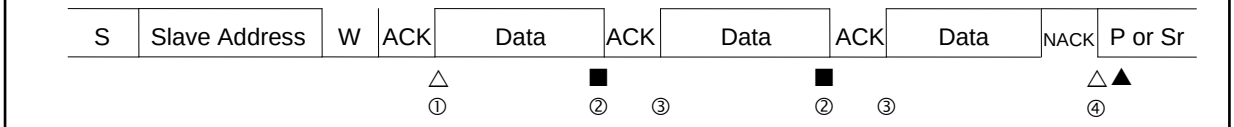
② An interrupt occurs when a single byte is received and an ACK is output .

- INT bit is set to "0".

③ An interrupt occurs when a single byte is received and an ACK is output .

- An interrupt occurs if INT bit is set to "0".

Figure 2-45 Slave mode received interrupt 7 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



△ :Interrupt by INTE="1"

▲ :Interrupt by CNDE="1"

■ :Interrupt by RIE="1"

① As the slave address matches , an ACK is output and an interrupt is generated .

- ACKE bit is set to "1" and INT bit is set to "0".

② An interrupt occurs (but the I^2C bus is not waited) when a single byte is received.

- The received data is read from the received buffer

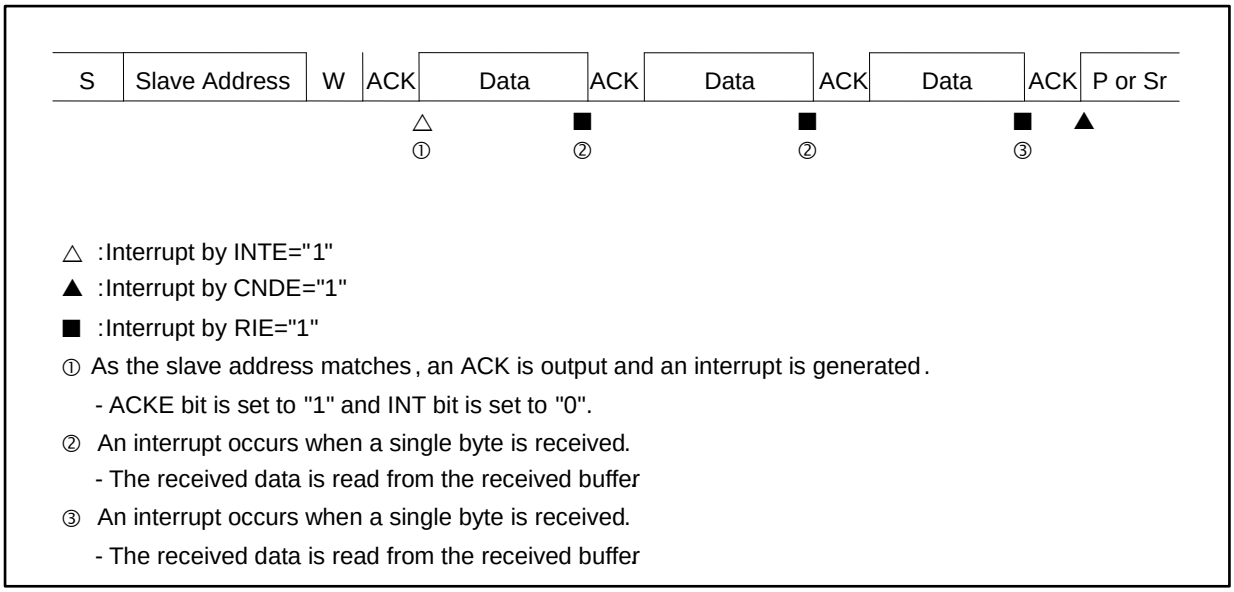
③ The I^2C bus is waited when an ACK is responded.

- The received data is read from the received buffer

④ An interrupt occurs when a single byte is received and a NACK is responded.

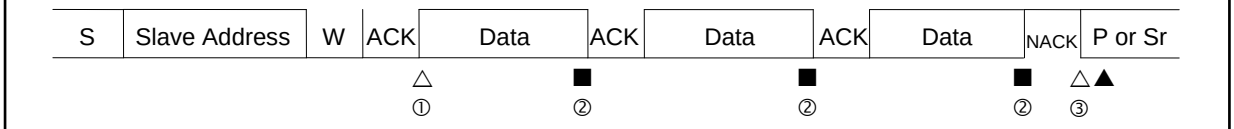
- After the received data has been read from the received buffer, the INT bit is set to "0".

Figure 2-46 Slave mode received interrupt 8 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



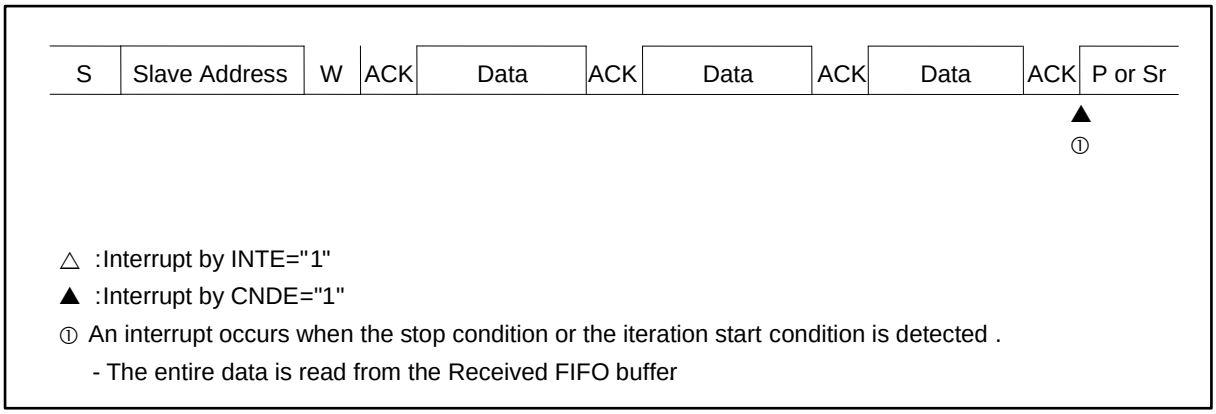
△ :Interrupt by INTE="1"
▲ :Interrupt by CNDE="1"
■ :Interrupt by RIE="1"
① As the slave address matches, an ACK is output and an interrupt is generated.
- ACKE bit is set to "1" and INT bit is set to "0".
② An interrupt occurs when a single byte is received.
- The received data is read from the received buffer
③ An interrupt occurs when a single byte is received.
- The received data is read from the received buffer

Figure 2-47 Slave mode received interrupt 9 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



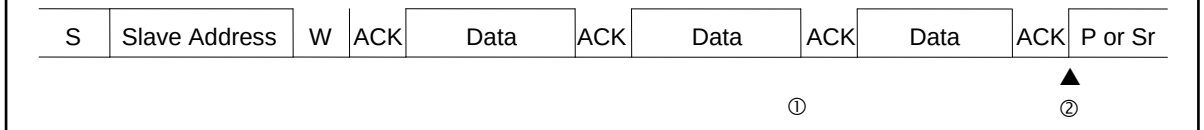
△ :Interrupt by INTE="1"
▲ :Interrupt by CNDE="1"
■ :Interrupt by RIE="1"
① As the slave address matches, an ACK is output and an interrupt is generated.
- ACKE bit is set to "1" and INT bit is set to "0".
② An interrupt occurs when a single byte is received.
- The received data is read from the received buffer
③ An interrupt occurs when a NACK is responded.
- INT bit is set to "0".

Figure 2-48 Slave mode received interrupt 10 by enabling received FIFO (SSR:DMA="1", IBSR:RSA="0")



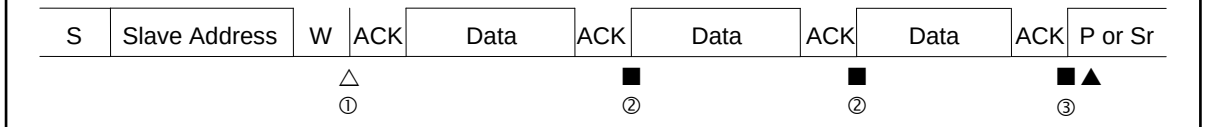
△ :Interrupt by INTE="1"
▲ :Interrupt by CNDE="1"
① An interrupt occurs when the stop condition or the iteration start condition is detected .
- The entire data is read from the Received FIFO buffer

Figure 2-49 Slave mode received interrupt 11 by enabling received FIFO (SSR:DMA="1", IBSR:RSA="0")



▲ :Interrupt by CNDE="1"
① As the Received FIFO buffer is filled with data, the I^2C bus is waited.
- The waiting is released when data is read even once from the Received FIFO buffer
② An interrupt occurs when the stop condition or the iteration start condition is detected .
- The entire data is read from the Received FIFO buffer

Figure 2-50 Slave mode received interrupt 12 by disabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")



△ :Interrupt by INTE="1"
▲ :Interrupt by CNDE="1"
■ :Interrupt by RIE="1"
① An interrupt occurs as the reserved address ("0000xxxx" or "1111xxxx") matches.
- The received data is read, and ACKE bit is set to "1" and INT bit is set to "0".
② An interrupt occurs when a single byte is received and an ACK is output.
- The received data is read.
③ An interrupt occurs when a single byte is received and an ACK is output.
- The received data is read.

Transmission in slave mode

If the received data matches the slave address and the data direction bit is "1", it means that data is transmitted in slave mode. If FIFO is disabled, set the interrupt flag (IBCR:INT) to "1" after transmitting one byte or outputting an acknowledgement response depending on setting of the IBCR:WSEL bit. Then place the I^2C bus into the wait state (see Table 2-8).

Using the IBSR:RACK bit, check the acknowledgement output from the master device. If NACK response is returned from the master device, it means that the master device could not receive data correctly or data receiving was ended. If NACK is detected at IBCR:WSEL=1, an interrupt is generated to place the I^2C bus into the wait state.

2.4 Bus error

If the stop or (iteration) start condition is detected while transmitting or receiving data on the I^2C bus, it is handled as a bus error.

Bus error occurrence condition

If a bus error occurs, the IBCR:BER bit is set to "1" in the following conditions.

- The (iteration) start or stop condition is detected while transferring the first byte.
- The (iteration) start condition or stop condition is detected at bit2 to bit9 (acknowledgement) of data.

Bus error operation

■ EIBCR:BEC=0

If the interrupt flag (IBCR:INT) is set to "1" by transmitting or receiving data, check the IBCR:BER bit. When the IBCR:BER bit is "1", perform error processing. The IBCR:BER bit is cleared by writing "0" to the IBCR:INT bit.

If a bus error occurs, the IBCR:INT bit is set to "1"; however, the I^2C bus is not placed into the wait state by setting its SCL to LOW.

■ EIBCR:BEC=1

If the interrupt flag (IBCR:SPC or IBCR:RSC) is set to "1" by transmitting or receiving data, check the IBCR:BER bit. When the IBCR:BER bit is "1", perform error processing. The IBCR:BER bit is cleared by flowing operations.

- When IBCR:INT=1, write "0" in IBCR:INT.
- When IBCR:SPC=1, write "0" in IBCR:SPC.
- When IBCR:RSC=1, write "0" in IBCR:RSC.

3. Dedicated Baud Rate Generator

The dedicated baud rate generator configures the setting of the serial clock frequency.

Selecting the baud rate

■ Baud rate obtained by dividing an internal clock using the dedicated baud rate generator (reload counter)

This generator provides two internal reload counters, which support transmitting and receiving serial clocks respectively. To select the baud rate, specify the 15-bit reload value using Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0).

Each reload counter divides an internal clock by the set value.

Calculating the baud rate

Two 15-bit reload counters are set using the Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0). The baud rate is obtained in the following formulas.

(1) Reload value

$V = \phi / b - 1$

V: Reload value b: Baud rate ϕ : Bus clock frequency or external clock frequency

Note that the preset baud rate may not be generated at a rising edge of signal on I^2C bus. In such case, adjust the reload value.

(2) Calculation example

To set the 16 MHz bus block and 400 kbps baud rate, set the reload value as follows.

Reload value:

$V = (16 \times 1000000)/400000 - 1 = 39$

Therefore, the baud rate is:

$b=(16 \times 100000) / (39+1) = 400$ kbps

Notes:

- *Write Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) by 16-bit access operation.*
- *When the ISMK:EN bit in the ISMK register is "0", set the value of each Baud Rate Generator Register.*
- *In operation mode 4 (I^2C mode), operate the bus clock at a frequency no lower than 8 MHz. Also note that setting of a baud rate generator that exceeds 400 kbps is prohibited.*
- *If the reload value is set to "0", the reload counter is stopped.*

Reload values and baud rates for each bus clock frequency

Table 3-1 Reload values and baud rates

Baud rate [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32 MHz
	Value	Value	Value	Value	Value	Value
1000000	setting is prohibited.					
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

Baud rate [bps]	40 MHz	48 MHz	72 MHz	80 MHz	100 MHz
	Value	Value	Value	Value	Value
1000000	setting is prohibited		71	79	99
400000	99	119	179	199	249
200000	199	239	359	399	499
100000	399	479	719	799	999

The numeric values above are available when the SCL rising timing of the I^2C bus is 0s. If the SCL rising timing of the I^2C bus is late, the baud rate is set to the value later than the numeric values above.

For frequencies not described in Table 3-1, calculate them by using the formulas in "n Calculating the baud rate" of 3. "Dedicated Baud Rate Generator". (However, for the maximum frequency, see the Data Sheet" of the product used because it differs depending on products.)

Functions of reload counter

Each reload counter consists of a 15-bit register for the reload value, and generates transmitting and receiving clocks from internal clocks. The count value of the transmit reload counter can be read from the Baud Rate Generator Registers (BGR1 and BGR0).

Starting counting

When the reload value is written to the Baud Rate Generator Register (BGR1 or BGR0), the reload counter starts counting.

4. I²C communication operation flowchart examples

This section shows I²C communication operation flowchart examples.

I²C flowchart example (FIFO not used) when DMA mode is disabled (SSR:DMA=0)

Figure 4-1 I²C flowchart example (FIFO not used) when DMA mode is disabled (SSR:DMA=0) 1/3

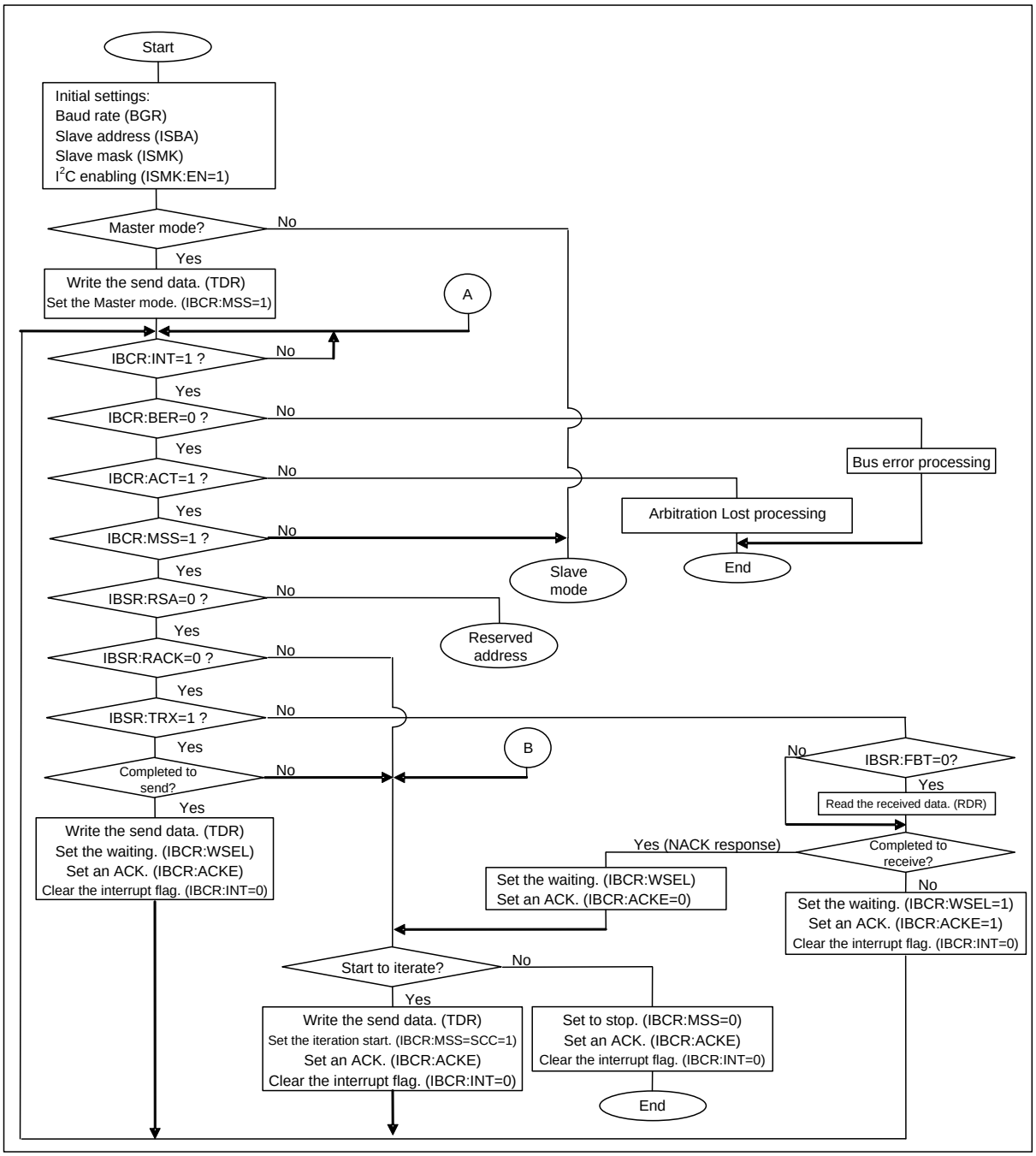


Figure 4-2 I²C flowchart example (FIFO not used) when DMA mode is disabled (SSR:DMA=0) 2/3

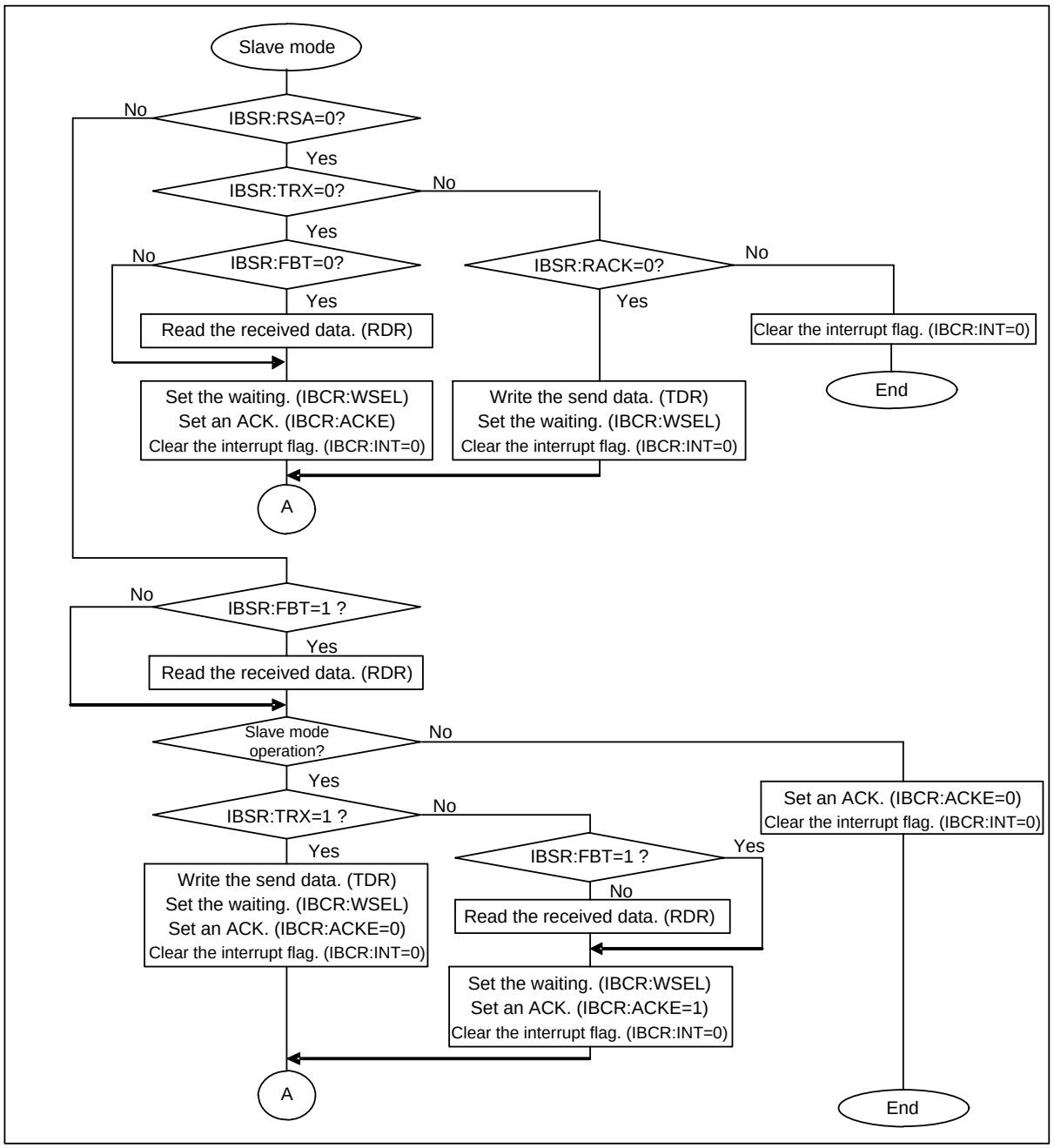
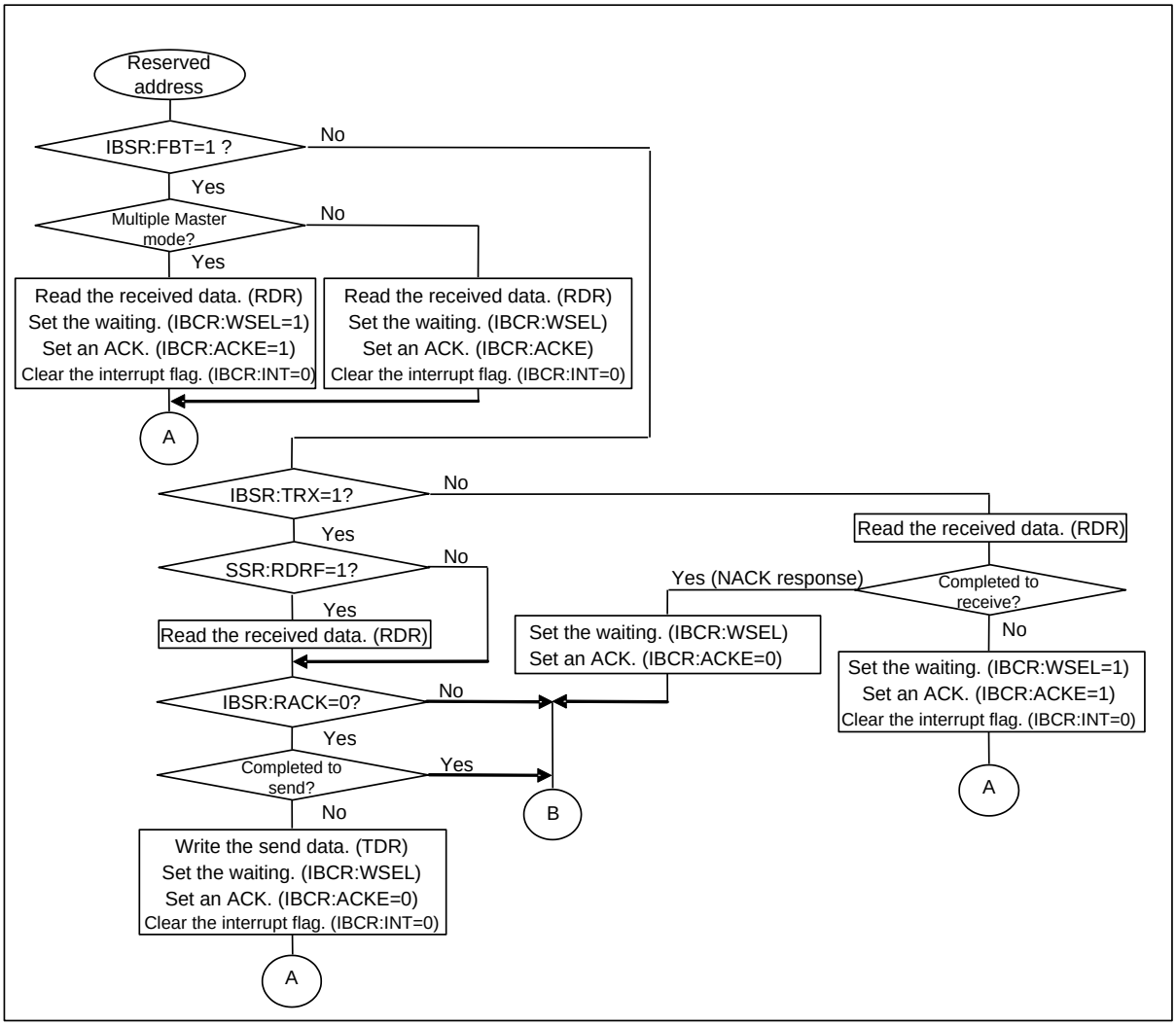


Figure 4-3 I²C flowchart example (FIFO not used) when DMA mode is disabled (SSR:DMA=0) 3/3



I²C flowchart examples (FIFO not used) when DMA mode is enabled (SSR:DMA=1)

Figure 4-4 I²C flowchart example (FIFO not used) when DMA mode is enabled (SSR:DMA=1) 1/4

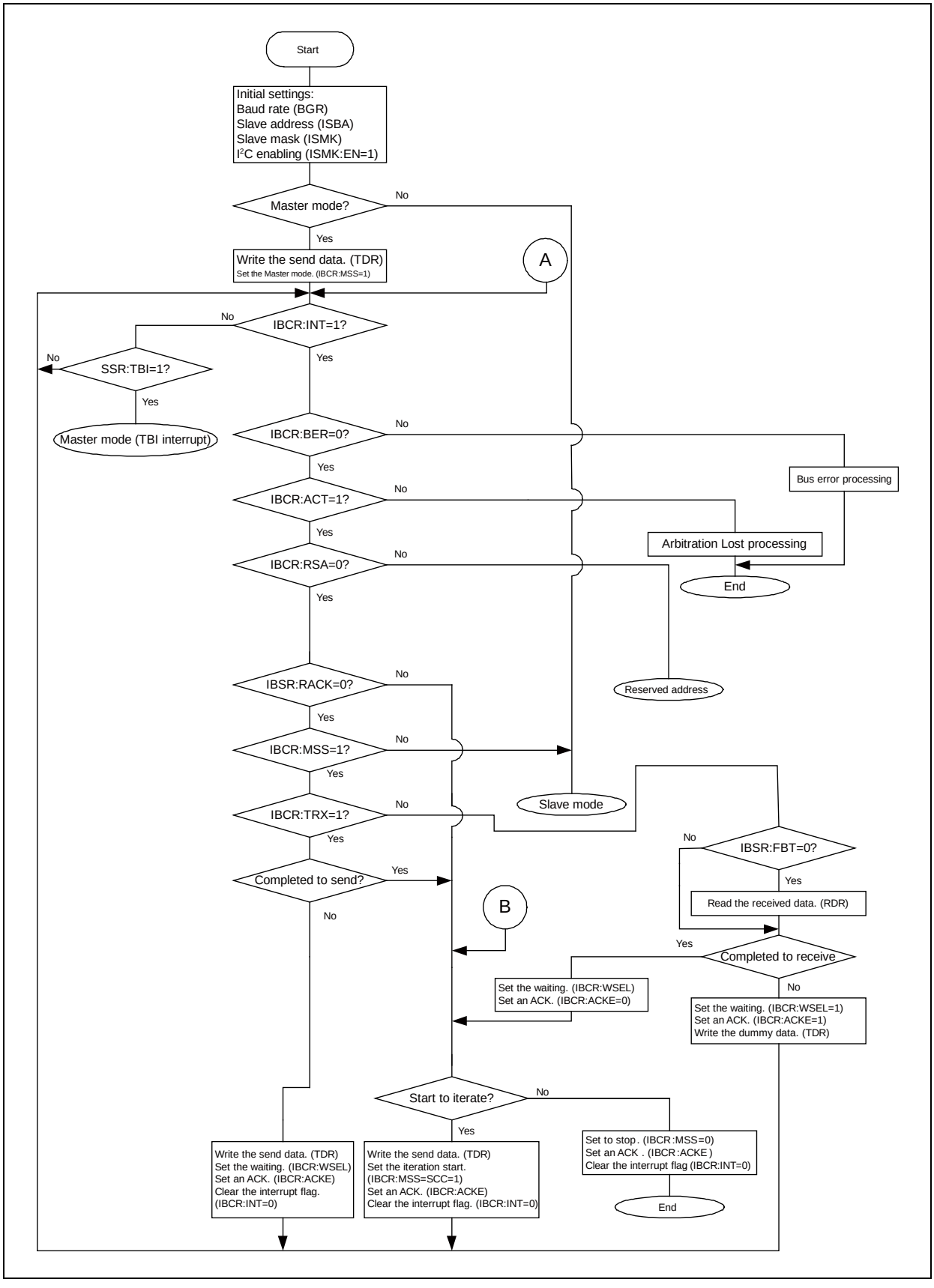


Figure 4-5 I²C flowchart example (FIFO not used) when DMA mode is enabled (SSR:DMA=1) 2/4

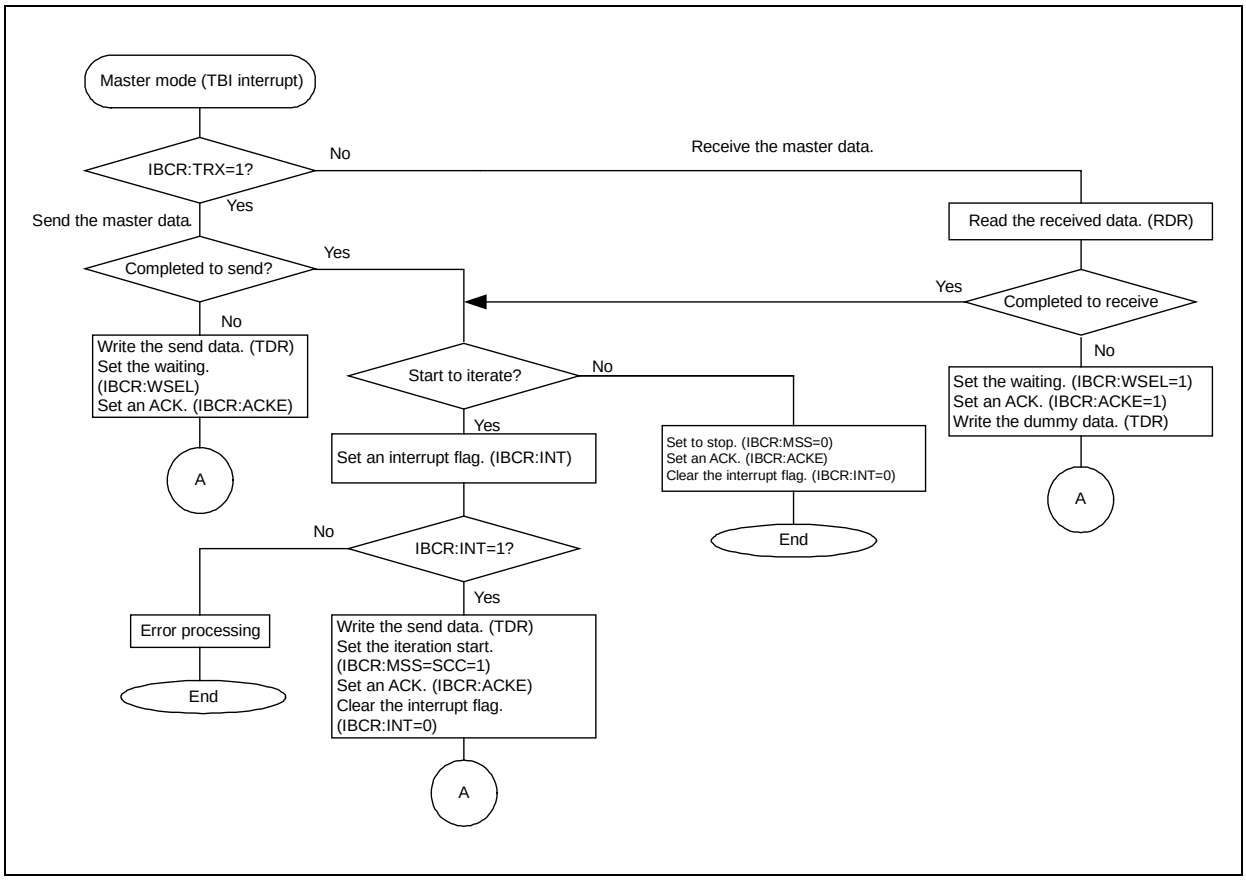


Figure 4-6 I²C flowchart example (FIFO not used) when DMA mode is enabled (SSR:DMA=1) 3/4

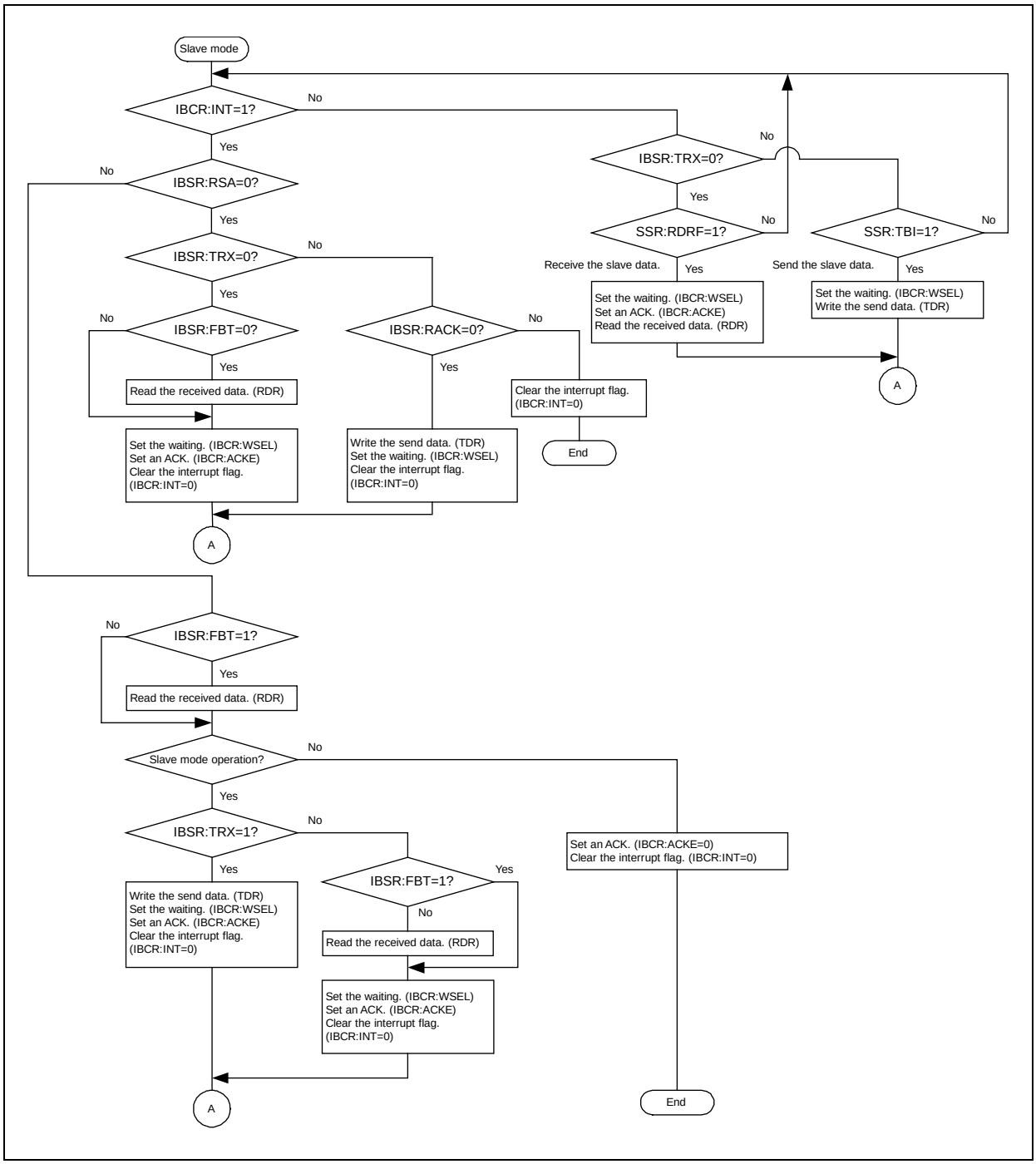
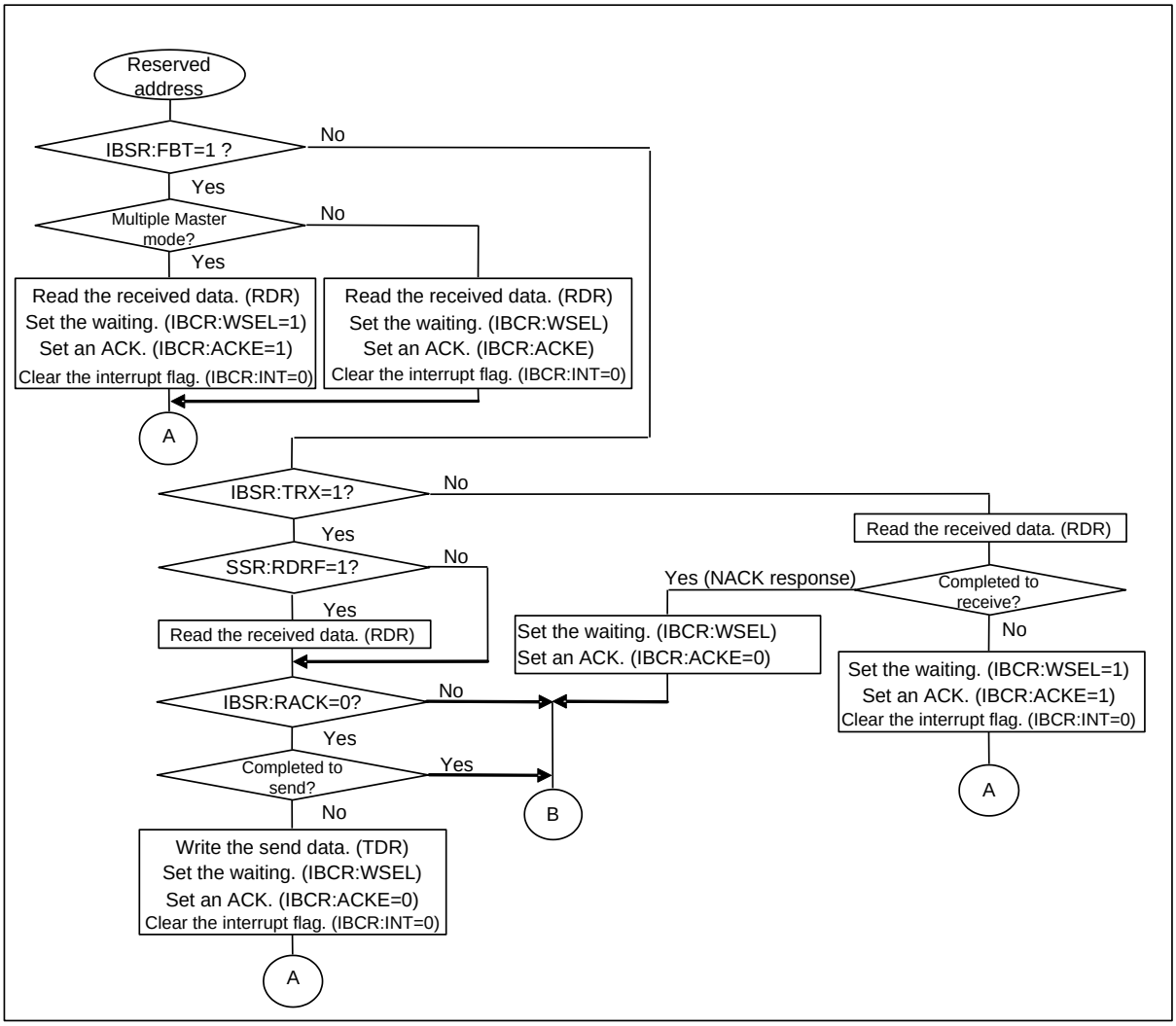


Figure 4-7 I²C flowchart example (FIFO not used) when DMA mode is enabled (SSR:DMA=1) 4/4



Note:

- *The flow shows an outline of operation settings in I²C mode. To perform the appropriate operations, take into account error processing based on applications.*

5. I²C Interface Registers

The following lists the I²C interface registers.

List of I²C interface registers

Table 5-1 List of I²C interface registers

	bit15 bit8	bit7 bit0
I²C	IBCR (I²C Bus Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	IBSR (I²C Bus Status Register)
	-	RDR/TDR (Transmit/Received Data Register)
	EIBCR (Extension I²C Bus control Register)	NFCR (Noise Filter Control Register)
	BGR1 (Baud Rate Generator Register 1)	BGR0 (Baud Rate Generator Register 0)
	ISMK (7-bit Slave Address Mask Register)	ISBA (7-bit Slave Address Register)
FIFO	FCR1 (FIFO Control Register 1)	FCR0 (FIFO Control Register 0)
	FBYTE2 (FIFO2 Byte Register)	FBYTE1 (FIFO1 Byte Register)

Table 5-2 I²C Interface bit assignment

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IBCR/ SMR	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	-	RIE	TIE	-	-
SSR/ IBSR	REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
TDR1/ TDR0	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
EIBCR/ NFCR	-	-	SDAS	SCLS	SDAC	SCLC	SOCE	BEC	-	-	-	NFT4	NFT3	NFT2	NFT1	NFT0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
ISMK/ ISBA	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

5.1 I²C Bus Control Register (IBCR)

The I²C Bus Control Register (IBCR) is used to select master or slave mode, generate an iteration start condition, enable an acknowledgement, enable an interrupt, and display an interrupt flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	(SMR)
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

[bit15] MSS: Master/slave select bit

- If this bit is set to "1" when the I²C bus is in idle state (ISMK:EN=1, IBSR:BB=0), master mode is selected.
- If this bit is set to "1" when the BB bit of IBSR register is "1", the occurrence of start condition is waited until the IBSR:BB bit is set to "0". If the slave address matches and the slave operation is started during waiting, this bit is set to "0" and the AL bit of IBSR register is set to "1".
- When master mode is selected (MSS=1, ACT=1) and the interrupt flag (INT) is "1", a stop condition is generated when this bit is set to "0".

The MSS bit is cleared in any of the following conditions.

1. When the I²C interface operation is disabled (ISMK:EN=0)
2. When an arbitration lost occurs
3. When a bus error is detected (BER=1) and when EIBCR:BEC=0.
4. When the MSS bit is set to "0" if INT=1
5. When DMA mode is enabled (SSR:DMA=1), SSR:TBI=1, and when the MSS bit is set to "0"

The following provides the relationship between MSS and ACT bits.

MSS bit	ACT bit	State
0	**0**	**Idle**
0	1	The slave address matching or ACK is responded to the reserved address (*1), and slave mode is in operation (in slave mode).
1	0	The master mode operation is waited.
1	1	During master mode operation (in master mode)

*1) ACK response: The SDA is LOW on the I²C bus during acknowledgement.

bit	Description
0	Selects slave mode.
1	Selects master mode.

Notes:

- *If DMA mode is disabled (SSR:DMA=0) and the MSS bit is set to "1", the MSS bit must be set to "0" only when the MSS bit is "1" and the INT bit is "1". If the MSS bit is set to "0" when the ACT bit is "1", the INT bit is also cleared to "0".*
- *If DMA mode is enabled (SSR:DMA=1) and the MSS bit is set to "1", the MSS bit must be set to "0" only when the MSS bit is "1" and the INT bit is "1", or the SSR:TBI bit is "1". If the MSS bit is set to "0" when the ACT bit is "1", the INT bit is also cleared to "0".*
- *When master mode is selected, the MSS bit is read to be "1" even when it is set to "0" while the ACT bit is "1".*

[bit14] ACT/SCC : Operation flag/iteration start condition generation bit

This bit setting has a different meaning when it is written and read.

Reading	Writing
ACT bit	SCC bit

The ACT bit indicates the current operation in master or slave mode.

The ACT bit is set when:

1. The start condition is output onto the I^2C bus (master mode)
2. The slave address matches the address sent from the master device (slave mode)
3. The reserved address is detected and it is acknowledged (If MSS is "0", slave mode is selected.)

The ACT bit is reset when:

<Master mode>

1. The stop condition is detected.
2. An arbitration lost is detected.
3. When a bus error is detected and when EIBCR:BEC=0.
4. The I^2C interface operation is disabled (ISMK:EN=0)

<Slave mode>

1. The (iteration) start condition is detected
2. The stop condition is detected.
3. The reserved address is detected (IBSR:RSA=1) but not acknowledged
4. The I2C interface operation is disabled (ISMK:EN=0)
5. When a bus error is detected (BER=1) and when EIBCR:BEC=0.

If this bit is set to "1" in master mode, the iteration start is executed. This bit is disabled to set to "0".

bit	Description	
	At writing	At reading
0	No effect	No effect
1	Generates an iteration start condition.	During the I2C operation

Notes:

- *The SCC bit must be set to "1" during an interrupt of master mode (when MSS=1, ACT=1 and INT=1) only. If the SCC bit is set to "1" when the ACT bit is "1", the INT bit is cleared to "0".*
- *This bit must not be set to "1" in slave mode (when MSS=0 and ACT=1).*
- *If the SCC bit is set to "1" and if the MSS bit is set to "0" simultaneously, the MSS bit setting is preceded.*
- *When data is read by a read-modify-write instruction, the SCC bit is read.*
- *If both of the following conditions are satisfied, the INT bit is set to "1" and the I2C bus is waited (SCL=LOW). To generate an iteration start condition, clear the INT bit by setting the SCC bit to "1" again.*
- *The SCC bit is set to "1" during master mode interrupt at 8th bit (MSS=1, ACT=1, INT=1 and WSEL=1).*
- *A negative acknowledgement (NACK) is received at 9th bit.*
- *When DMA mode is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.*

1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set this bit to "1".

[bit13] ACKE: Data byte acknowledge enable bit

- If this bit is set to "1", LOW is output when acknowledged.
- This bit must be changed if any of the following conditions has occurred:
  - If DMA mode is disabled (SSR:DMA=0), the ACT bit is "1", and the INT bit is "1"
  - If DMA mode is enabled (SSR:DMA=1), the ACT bit is "1", and the SSR:TBI bit is "1"
  - If DMA mode is enabled (SSR:DMA=1), the ACT bit is "1", the slave mode reception is selected, and the SCR:RDRF is "1"
  - If the ACT bit is "0"

This bit is invalid in the following conditions.

1. During acknowledgement to an address field other than the reserved address (automatic generation)
2. During data transmission (IBSR:RSA=0, IBSR:TRX=1, IBSR:FBT=0)
3. If the received FIFO is enabled and the slave mode reception is selected (FCR0:FE=1, MSS=0, ACT=1), an ACK is returned.
4. If the received FIFO is enabled, the WSEL bit is "0", the master mode reception is selected (FCR0:FE=1, MSS=1, ACT=1, WSEL=0), and the SSR:TDRE bit is "0", an ACK is always returned. If the SSR:TDRE bit is "1", a NACK is returned.
5. If the received FIFO is enabled, WSEL=0, the reserved address is detected and the slave transmission is selected (IBSR:RSA=1, IBSR:TRX=1, IBSR:FBT=1), an ACK is always returned. To respond with a NACK, disable the received FIFO and set the ACKE bit to "0" during interrupt after detection of the reserved address.
6. The received FIFO is enabled, the WSEL bit is "1", the master mode reception is selected, and the Transmit Data Register has data (FCR0:FE=1, MSS=1, ACT=1, WSEL=1, SSR:TDRE=0).

bit	Description
0	Disables acknowledgment.
1	Enables acknowledgement.

[bit12] WSEL: Wait selection bit

- If DMA mode is disabled (SSR:DMA=0), this bit selects a generation time of interrupt before or after acknowledgement (INT=1) and selects to wait the I2C bus or not.
- If DMA mode is enabled (SSR:DMA=1), this bit selects a generation time of interrupt before or after acknowledgement (INT=1, and SSR:TBI=1 for transmission or SSR:RDRF=1 for reception) and selects to wait the I2C bus or not.
- The WSEL bit is invalid in the following conditions.
  1. An interrupt occurs (INT=1) for the first byte. (*1)
  2. The reserved address is detected (IBSR:FBT=1, IBSR:RSA=1).
  3. The NACK response is detected during FIFO data transfer (FCR0:FE=1, IBSR:RACK=1, ACT=1). (*2)
  4. The received FIFO is filled with data during FIFO reception.

  *1) The first byte indicates data after the (iteration) start condition.

  *2) NACK response: The SDA bit of I2C bus is HIGH during acknowledgement.

bit	Description
0	Waits (9 bits) after acknowledgement.
1	Waits (8 bits) after data transmission or reception.

[bit11] CNDE: Condition detection interrupt enable bit

This bit enables an interrupt if a stop condition or an iteration start condition is detected in master or slave mode (ACT=1). An interrupt occurs if the RSC or SPC bit of IBSR register is "1" and if this bit is set to "1".

bit	Description
0	Disables an interrupt due to the iteration start or stop condition.
1	Enables an interrupt due to the iteration start or stop condition.

[bit10] INTE: Interrupt enable bit

This bit enables an interrupt (INT=1) due to a data transmission and reception or bus error in master or slave mode.

bit	Description
0	Disables an interrupt.
1	Enables an interrupt.

[bit9] BER: Bus error flag bit

This bit indicates that an error has been detected on the I2C bus.

The BER bit is set when:

1. The start or stop condition is detected during transfer of the first byte. (*1)
2. The (iteration) start condition or the stop condition is detected at bit2 to bit9 (acknowledgement) of data after the 2nd or subsequent byte.

The BER bit is reset when:

1. The INT bit is set to "0" if EIBCR:BEC=0 and BER=1.
2. The I2C interface operation is disabled (ISMK:EN=0).
3. The IBCR:INT bit is set to "0" when EIBCR:BEC=1 and IBCR:INT=1.
4. The IBCR:SPC bit is set to "0" when EIBCR:BEC=1 and IBCR:SPC=1.
5. The IBCR:RSC bit is set to "0" when EIBCR:BEC=1 and IBCR:RSC=1.

*1) The first byte indicates data after the (iteration) start condition.

bit	Description
0	No error
1	An error was detected.

Note:
In the following cases, check this bit state if the interrupt flag (INT bit) is "1". If it is "1", the normal data transmission and reception fail. Retransmit the data.

- *The interrupt flag(INT bit) is "1" when EIBCR:BEC=0*
- *The iteration start condition confirmation bit(IBSR:RSC bit) is "1" when EIBCR:BEC=1*
- *The stop condition confirmation bit(IBSR:SPC bit) is "1" when EIBCR:BEC=1*

[bit8] INT: interrupt flag bit

The interrupt flag bit is set to "1" after 8 or 9 bits (ACK) of data have been transmitted and received or when a bus error has occurred in master or slave mode. During operation other than bus error, if the INT bit is set to "1", the SCL flag is set to LOW. If the INT bit is set to "0", the SCL is released from the LOW state.

The INT bit is set when:

<8th bit>

<If DMA mode is not related>

1. The reserved address is detected in the first byte.
2. The WSEL bit is "1" and an arbitration lost is detected in the 2nd or subsequent byte.

<If DMA mode is disabled (SSR:DMA=0)>

1. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", master mode is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.
2. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", slave mode is selected, the received FIFO is disabled, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.
3. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", the slave mode transmission is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.
4. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "1", the received FIFO is disabled, and the slave mode reception is selected.

<If DMA mode is enabled (SSR:DMA=1)>

1. If DMA mode is enabled (SSR:DMA=1), WSEL bit is "1", master mode is selected , the SSR:TBI bit is "1" in the 2nd or subsequent byte, and the INT bit is set to "1".

<9th bit>

<If DMA mode is not related>

1. An arbitration lost is detected in the first byte.
2. The NACK signal is received during the time other than stop condition output setting (the MSS bit is set to "0" during the master mode operation).
3. The WSEL bit is "0" and an arbitration lost is detected in the 2nd or subsequent byte.
4. The reserved address is not detected in the 1st byte, and data is found in the received FIFO when the received FIFO is enabled and data is received in master or slave mode (IBSR:TRX=0).
5. EIBCR:BEC=1 and IBSR:BER=1

<If DMA mode is disabled (SSR:DMA=0)>

1. If DMA mode is disabled (SSR:DMA=0), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when data is transmitted (IBSR:TRX=1) in master or slave mode.
2. If DMA mode is disabled (SSR:DMA=0), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when the received FIFO is disabled for data reception (IBSR:TRX=0) in master or slave mode.
3. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "0", and the SSR:TDRE bit is "1" in the 2nd or subsequent byte during the master mode operation.
4. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "0", and the SSR:TDRE bit is "1" in the 2nd or subsequent byte during the slave mode transmission.
5. If DMA mode is disabled (SSR:DMA=0), WSEL bit is "0", the received FIFO is disabled, and the slave mode reception is selected. However, if the reserved address is detected in the 1st byte during the slave mode reception, no interrupt is generated by bit 9.
6. If DMA mode is disabled (SSR:DMA=0), the received FIFO is enabled, data is received in slave mode, and the received FIFO is filled with data.

<If DMA mode is enabled (SSR:DMA=1)>

1. If DMA mode is enabled (SSR:DMA=1), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when data is transmitted (IBSR:TRX=1) in slave mode.
2. If DMA mode is enabled (SSR:DMA=1), the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1" when the received FIFO is disabled for data reception (IBSR:TRX=0) in slave mode.
3. If DMA mode is enabled (SSR:DMA=1), WSEL bit is "0", the SSR:TBI bit is "1" in the 2nd or subsequent byte during the master mode operation, and the INT bit is set to "1".

<Others>

1. When a bus error is detected and EIBCR:BEC=0.

The INT bit is reset when:

1. The INT bit is set to "0".
2. The INT bit is "1" and the ACT bit is "1", the MSS bit is set to "0".
3. The INT bit is "1" and the ACT bit is "1", the SCC bit is set to "1".

If the DMA mode is disabled (SSR:DMA=0), it is invalid to set the INT bit to "1".

bit	Description	
	At writing	At reading
0	Clears the INT bit.	Does not issue an interrupt request.
1	No effect	Issues an interrupt request.

Notes:

- *When DMA mode is enabled (SSR:DMA=1) and the SSR:TBI bit is "1" in the 2nd or subsequent byte during the master mode operation, a status interrupt (SIRQ=1) is not generated even when the INT bit is set to "1".*
- *When DMA is enabled (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below to issue the iteration start condition.*

1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

- *If the INT flag is changed from "1" to "0", the I2C bus is released from waiting.*
- *If the ISMK:EN bit is set to "0", the SSR:RDRF and INT bits may be set to "1" in certain received timing. If so, read the received data and clear the INT bit.*
- *When a read-modify-write instruction is issued, "1" is read.*
- *If the received FIFO is enabled, the INT bit is not set to "1" even when the received FIFO is filled with data during the master mode reception.*
- *Set this bit to "1" when the start condition is issued (IBCR:MSS=1).*

5.2 Serial Mode Register (SMR)

The Serial Mode Register (SMR) is used to set an operation mode, and to enable or disable the transmit/received interrupt.

bit	15 ... 8	7	6	5	4	3	2	1 0
Field	(SCR)	MD2	MD1	MD0	Reserved	RIE	TIE	Reserved
Attribute		R/W	R/W	R/W	-	R/W	R/W	-
Initial value		0	0	0	-	0	0	-

[bit7:5] MD2, MD1, MD0: operation mode set bits

These bits set an operation mode.

* This chapter explains the registers and their operation in operation mode 4 (I2C mode).

bit7	bit6	bit5	Description
0	0	0	Operation mode 0 (async normal mode)
0	0	1	Operation mode 1 (async multiprocessor mode)
0	1	0	Operation mode 2 (clock sync mode)
0	1	1	Operation mode 3 (LIN communication mode)
1	0	0	Operation mode 4 (I2C mode)
Values other than the above			Setting disabled.

Notes:

- *Any bit setting other than above is prohibited.*
- *To switch the current operation mode, disable the I2C (ISMK:EN=0) and change the operation mode continuously.*
- *After the operation mode has been set, set each register correctly.*

[bit4] Reserved: Reseved bit

The read value is "0". Be sure to write "0".

[bit3] RIE: Received interrupt enable bit

- This bit enables or disables an output of received interrupt request to the CPU.
- If the RIE bit and the received data flag bit (SSR:RDRF) are "1", or if any of error flag bits (SSR:ORE) is "1", a received interrupt request is output.

bit	Description
0	Disables the received interrupt.
1	Enables the received interrupt.

Note:

To receive data using the INT bit of I2C Bus Control Register (IBCR) when DMA mode is disabled (SSR:DMA=0), set this bit to "0".

[bit2] TIE: Transmit interrupt enable bit

- This bit enables or disables an output of transmit interrupt request to the CPU.
- If the TIE and SSR:TDRE bits are "1", a transmit interrupt request is output.

bit	Description
0	Disables the transmit interrupt.
1	Enables the transmit interrupt.

Note:

- *To transmit data using the INT bit of I2C Bus Control Register (IBCR) when DMA mode is disabled (SSR:DMA=0), set this bit to "0".*

[bit1:0] Reserved : Reserved bits

The read value is "0". Be sure to write "0".

5.3 I2C Bus Status Register (IBSR)

The I2C Bus Status Register (IBSR) shows the iteration start, acknowledgement, data direction, arbitration lost, stop condition, I2C bus status, and bus error detection.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SSR)	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
Attribute		R	R	R	R	R	R/W	R/W	R
Initial value		0	0	0	0	0	0	0	0

[bit7] FBT: First byte bit

This bit indicates the first byte.

The FBT bit is set when:

1. The (iteration) start condition is detected.

The FBT bit is cleared when:

1. The second byte is sent or received.
2. The stop condition is detected.
3. The I2C interface operation is disabled (ISMK:EN=0).
4. When a bus error is detected (IBCR:BER=1) and EIBCR:BEC=0.

bit	Description
0	Other than 1st byte
1	The 1st byte is being sent or received.

[bit6] RACK: Acknowledge flag bit

This bit shows acknowledgement being received in the 1st byte or in master or slave mode.

The RACK bit is updated when:

1. Acknowledged in the 1st byte.
2. Data is acknowledged in master or slave mode.

The RACK bit is cleared (RACK=0) when:

1. The (iteration) start condition is detected.
2. The I2C interface operation is disabled (ISMK:EN=0).
3. When a bus error is detected (IBCR:BER=1) and EIBCR:BEC=0.

bit	Description
0	LOW is received.
1	HIGH is received.

[bit5] RSA: Reserved address detection bit

This bit shows that the reserved address has been detected.

The RSA bit is set (RSA=1) when:

1. The 1st byte is "0000xxxx" or "1111xxxx". Where "x" can be "0" or "1".

The RSA bit is reset (RSA=0) when:

1. The (iteration) start condition is detected.
2. The stop condition is detected.
3. The I2C interface operation is disabled (ISMK:EN=0).
4. When a bus error is detected (IBCR:BER=1) and EIBCR:BEC=0.

If the RSA bit is set to "1" in the 1st byte, the interrupt flag (IBCR:INT) is set to "1" and the SCL flag is set to "L" at the falling edge of SCL (8th bit) of the 1st byte regardless of FIFO enable or disable state. To read the received data and start the slave mode operation during this time, set the IBCR:ACKE bit to "1" and clear the interrupt flag (IBCR:INT) to "0". If the TRX bit is "0" after that, data is received in slave mode. To stop the data reception, set the IBCR:ACKE bit to "0". No data is received after that.

bit	Description
0	The reserved address is not detected.
1	The reserved address is detected.

Notes:

- *If the IBCR:ACKE bit is set to "0" during data transfer, this IBCR:ACKE bit cannot be set to "1" until the stop condition or the iteration start condition is detected.*
- *If the slave mode transmission is detected during an interrupt by reserved address detection and if the received FIFO is enabled, an ACK response is returned. In this case, disable the received FIFO and set the IBCR:ACKE bit to "0".*

[bit4] TRX: Data direction bit

This bit indicates the data direction.

The TRX bit is set when:

1. The (iteration) start condition is sent in master mode.
2. 8th bit of the 1st byte is "1" in slave mode (in the slave mode transmission direction).

The TRX bit is reset when:

1. An arbitration lost occurs (AL=1).
2. 8th bit of the 1st byte is "0" in slave mode (in the slave mode reception direction).
3. 8th bit of the 1st byte is "1" in master mode (in the master mode reception direction).
4. The stop condition is detected.
5. The (iteration) start condition is detected in any mode other than master mode.
6. The I2C interface operation is disabled (ISMK:EN=0).
7. When a bus error is detected (IBCR:BER=1) and EIBCR:BEC=0.

bit	Description
0	Received direction
1	Transmission direction

[bit3] AL: Arbitration lost bit

This bit indicates an arbitration lost.

The AL bit is set when:

1. The output data does not match the received data in master mode.
2. The IBCR:MSS bit is set to "1" but the slave mode operation is selected.
3. The iteration start condition is detected by 1st bit of the 2nd or subsequent byte data in master mode when EIBCR:BEC=0.
4. The iteration start condition is detected in master mode and when EIBCR:BEC=0.
5. The stop condition is detected by 1st bit of the 2nd or subsequent byte data in master mode when EIBCR:BEC=1.
6. The stop condition is detected in master mode when EIBCR:BEC=1 (except the case where the stop condition is detected in the acknowledge field.)
7. The iteration start condition cannot be generated in master mode.
8. The stop condition cannot be generated in master mode.

The AL bit is reset when:

1. The IBCR:MSS bit is set to "1".
2. The IBCR:INT bit is set to "0".
3. The SPC bit is set to "0" when both AL and SPC bits are "1".
4. The I2C interface operation is disabled (ISMK:EN=0).
5. When a bus error is detected (IBCR:BER=1) and EIBCR:BEC=0.

bit	Description
0	No arbitration lost has occurred.
1	An arbitration lost has occurred.

[bit2] RSC: Iteration start condition check bit

This bit shows that an iteration start condition is detected in master or slave mode.

The RSC bit is set when:

1. When an iteration start condition is detected after acknowledgement, during the master or slave mode operation when EIBCR:BEC=0.
2. When an iteration start condition is detected in the first byte, during the master or slave mode, in the first bit when EIBCR:BEC=1.

The RSC bit is reset when:

1. The RSC bit is set to "0".
2. The IBCR:MSS bit is set to "1".
3. The I2C interface operation is disabled (ISMK:EN=0).

It is invalid to set this bit to "1".

bit	Description
0	No iteration start condition has been detected.
1	An iteration start condition has been detected.

Notes:

- *If no acknowledgement response is sent while data is received in slave mode due to the reserved address being detected, slave mode is released. In this case, this bit is not set to "1" even if the next iteration start condition is detected.*
- *When a read-modify-write instruction is issued, "1" is read.*

[bit1] SPC: Stop condition check bit

This bit shows that a stop condition is detected in master or slave mode.

The SPC bit is set when:

1. When the stop condition is detected in the master or slave mode operation, when EIBCR:BEC=0.
2. The stop condition is detected in the one of the following cases when EIBCR:BEC=1.
   - In the first byte when IBCR:ACT=0
   - In the slave operation mode
   - In the master mode(except the case where the stop condition is detected in the acknowledge field)
3. In master mode, the stop condition has occurred and, therefore, an arbitration lost has occurred.

The SPC bit is reset when:

1. This bit is set to "0".
2. The IBCR:MSS bit is set to "1".
3. The I2C interface operation is disabled (ISMK:EN=0).

It is invalid to set this bit to "1".

bit	Description	
0	No stop condition is detected.	
1	Master mode	An arbitration lost has occurred when the stop condition is detected or when it is output.
	Slave mode	The stop condition is detected.

Notes:

- *If no acknowledgement response is sent while data is received in slave mode due to the reserved address being detected, slave mode is released. In this case, this bit is not set to "1" even if the next stop condition is detected.*
- *When a read-modify-write instruction is issued, "1" is read.*
- *When all the following conditions are met, this bit is not set to"1" and the master operation is continued even if the stop condition is detected:*
- *When EIBCR:BEC=1*
- *In the master operation*
- *In the acknowledge field*

[bit0] BB: Bus state bit

This bit shows the bus state.

The BB bit is set when:

1. LOW is detected in SDA or SCL of the I2C bus.

The BB bit is reset when:

1. The stop condition is detected.
2. The I2C interface operation is disabled (ISMK:EN=0).
3. When a bus error is detected (IBCR:BER=1) and EIBCR:BEC=0.

bit	Description
0	The bus is in idle state.
1	The bus is in transmission and reception state.

5.4 Serial Status Register (SSR)

The Serial Status Register (SSR) is used to check the transmission or reception state.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI	(IBSR)
Attribute	R/W	R/W	R/W	R/W	R	R	R	R	
Initial value	0	0	0	0	0	0	1	1	

[bit15] REC: Received error flag clear bit

This bit clears the ORE bit of Serial Status Register (SSR).

- If this bit is set to "1", the ORE bit is cleared.
- This bit has no effect on the operation if set to "0".

When it is read, "0" is always read.

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	Clears the Received Error flag (ORE).	

[bit14] TSET: Transmit empty flag set bit

This bit sets the TDRE bit of Serial Status Register (SSR).

- If it is set to "1" and if the TDRE bit and DMA mode are enabled (DMA=1), the TBI bit is set.
- This bit has no effect on the operation if set to "0".

When it is read, "0" is always read.

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	The TDRE bit is set.	

Note:

- *Set this bit to "1" only when the IBCR:INT bit is "1".*

[bit13] DMA: DMA mode enable bit

This bit enables or disables DMA mode.

- If this bit is set to "1", an interrupt condition is generated during DMA transfer.
- If this bit is set to "0", an interrupt condition is generated during normal data transfer.

For details, see Table 2-1.

bit	Description
0	Disables DMA mode.
1	Enables DMA mode.

Note:

- *This bit state can be changed only when the ISMK:EN bit is "0".*

[bit12] TBIE: Transmit bus idle interrupt enable bit (Effective only when DMA mode is enabled)

- This bit enables or disables an output of transmit bus idle interrupt request to the CPU.
- If DMA mode is enabled (DMA=1) and both TBIE and TBI bits are "1", a transmit bus idle interrupt request is output.
- If DMA mode is disabled (DMA=0), this bit is set to "0". If data is written, this writing is ignored and the "0" is maintained.

bit	Description
0	Disables the transmit bus idle interrupt.
1	Enables the transmit bus idle interrupt.

[bit11] ORE: Overrun error flag bit

- If an overrun occurs during data reception, this bit is set to "1". This is cleared if the REC bit of Serial Status Register (SSR) is set to "1".
- If the ORE and SMR:RIE bits are "1", a received interrupt request is output.
- If this flag is set, the Received Data Register (RDR) is invalid.
- If the received FIFO is used and if this flag is set, the received data is not stored in the received FIFO.

bit	Description
0	No overrun error occurred.
1	An overrun error occurred.

[bit10] RDRF: Received data full flag bit

- This flag shows the state of Received Data Register (RDR).
- If the SMR:RIE bit and the received data flag bit (RDRF) are "1", a received interrupt request is issued.
- When the received data is loaded in the RDR, this bit is set to "1". When data is read from the Received Data Register (RDR), this bit is cleared to "0".
- This bit is set at the falling edge of SCL signal (8th bit of data).
- This bit is also set even when a NACK is responded. (*1)
- If the received FIFO is used and if a certain count of data is received by the received FIFO, the RDRF bit is set to "1".
- If the received FIFO is used and if received FIFO is emptied, this bit is cleared to "0".
- If all of the following conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the interrupt flag (SSR:RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FCR:FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.
  - The IBCR:BER bit is "0".

If the RDR data is read during counting of 8 clocks, this counter is reset to 0 and counting for 8 clocks is restarted.

*1) NACK response: The SDA bit of I2C bus is "H" during acknowledgement.

bit	Description
0	The Received Data Register (RDR) is empty.
1	The Received Data Register (RDR) contains data.

Notes:

- *If all of the following conditions are satisfied, the SCL flag is set to LOW after ACK is transmitted was transmitted. If the RDRF bit is set to "0", the SCL flag is released from the LOW state.*
  - *The received FIFO is not used.*
  - *DMA mode is enabled (SSR:DMA=1).*
  - *Data is received in the 2nd or subsequent byte (IBSR:TRX=0), and the RDRF bit is "1".*
  - *The IBCR:WSEL bit is "0".*
- *If all of the following conditions are satisfied, the SCL flag is set to LOW immediately after single-byte data reception. If the RDRF bit is set to "0", the SCL flag is released from the LOW state.*
  - *The received FIFO is not used.*
  - *DMA mode is enabled (SSR:DMA=1).*
  - *Data is received in the 2nd or subsequent byte (IBSR:TRX=0), and the RDRF bit is "1".*
  - *The IBCR:WSEL bit is "1".*
- *If the received FIFO is used and DMA mode is enabled for data reception (DMA=1), the SCL flag is set to LOW when the received FIFO is filled with data. If data is read from the RDR even once, the SCL flag is released from the LOW state.*

[bit9] TDRE: Transmit data empty flag bit

- This flag shows the state of Transmit Data Register (TDR).
- If the SMR:TIE and TDRE bits are "1", a Transmit Interrupt Request is output.
- If transmit data is written in the TDR, this bit is set to "0" to indicate that the TDR contains valid data. When data is loaded to a shift register for transmission and its transmission is started, this bit is set to "1" to indicate that the TDR does not have the valid data.
- If the TSET bit of Serial Status Register (SSR) is set to "1", this flag is set. If an arbitration lost or a bus error is detected, use this flag to set the TDRE bit to "1".

bit	Description
0	The Transmit Data Register (TDR) contains data.
1	The Transmit Data Register is empty.

[bit8] TBI: Transmit bus idle flag bit (Effective only when DMA mode is enabled)

This bit shows that no data is sent by the I2C when DMA mode is enabled (DMA=1). If DMA mode is enabled (DMA=1) and the TBI bit is set to "1" in the 2nd or subsequent byte, the SCL flag is set to LOW. If the TBI bit is set to "0", the SCL flag is cleared from the LOW state.

The TBI bit is set when:

<8th bit>

1. The WSEL bit is "1", master mode is selected, and the TDRE bit is "1" in the 2nd or subsequent byte.
2. The WSEL bit is "1", the slave mode transmission is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.

<9th bit>

1. Master mode is selected, the reserved address is not detected in the 1st byte, and the SSR:TDRE bit is "1".
2. The WSEL bit is "0", master mode is selected, and the TDRE bit is "1" in the 2nd or subsequent byte.
3. The WSEL bit is "0", the slave mode transmission is selected, and the SSR:TDRE bit is "1" in the 2nd or subsequent byte.

<Others>

The transmit buffer empty flag set bit (TSET) is set to "1".

The TBI bit is reset when:

1. The transmit data is written in the Transmit Data Register (TDR).

If this bit is "1" and if the transmit bus idle interrupt is enabled (SCR:TBIE=1), a transmit interrupt request is output.

- If DMA mode is disabled (DMA=0), this bit is undefined.

bit	Description
0	During data transmission
1	No data transmission

5.5 Received Data Register/Transmit Data Register (RDR/TDR)

The Received and Transmit Data Registers are allocated at the same address. This register functions as the Received Data Register when data is read from it. This register functions as the Transmit Data Register when data is written in it.

Received Data Register (RDR)

bit	15 ... 8	7	6	5	4	3	2	1	0
Field		D7	D6	D5	D4	D3	D2	D1	D0
Attribute		R	R	R	R	R	R	R	R
Initial value		0	0	0	0	0	0	0	0

The Received Data Register (RDR) is a data buffer register for serial data reception.

- When a serial data signal is sent to the serial data line (SDA pin), it is converted by a shift register and stored in the Received Data Register (RDR).
- When the first byte (*1) is received, a received address is not stored in the Received Data Register (RDR). However, when the first byte is a reserved address, a received address is stored in the Received Data Register (RDR). In this case, the least significant bit (RDR:D0) is the data direction bit.
- When the received data is stored in the Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is set to "1".
- When data is read from the Received Data Register (RDR), the received data full flag bit (SSR:RDRF) is cleared to "0" automatically.

  *1) The first byte indicates data after the (iteration) start condition.

Notes:

- *If the received FIFO is used and if a certain count of data is received by the received FIFO, the SSR:RDRF bit is set to "1".*
- *If the received FIFO is used and if received FIFO is emptied, the SSR:RDRF bit is cleared to "0".*

Transmit Data Register (TDR)

bit	15 ... 8	7	6	5	4	3	2	1	0
Field		D7	D6	D5	D4	D3	D2	D1	D0
Attribute		W	W	W	W	W	W	W	W
Initial value		1	1	1	1	1	1	1	1

The Transmit Data Register (TDR) is a data buffer register for serial data transmission.

- Data of the Transmit Data register (TDR) is output to the serial data line (SDA pin) with the MSB first order.
- When the first byte is transmitted, the least significant bit (TDR:D0) indicates the data direction.
- When the transmit data is written in the Transmit Data Register (TDR), the transmit data empty flag (SSR:TDRE) is cleared to "0".
- When data is transferred to a shift register for transmission, the transmit data empty flag (SSR:TDRE) is set to "1".
- If transmit FIFO is disabled and if the data empty flag (SSR:TDRE) is "0", the transmit data cannot be written in the Transmit Data Register (TDR).
- If transmit FIFO is used, the transmit data can be written until transmit FIFO is filled with it even if the transmit data empty flag (SSR:TDRE) is "0".

Note:

- *The Transmit Data Register is a write-only register. While the Received Data Register is a read-only register. As these two registers are allocated at the same address, the write and read values differ from each other. Therefore, the INC/DEC instruction and other read-modify-write (RMW) instruction cannot be used.*

5.6 Noise Filter Control Register (NFCR)

The Noise Filter Control Register (NFCR) is used to set te noise filter time.

Noise Filter Control Register (NFCR)

Bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(EIBCR)	-	-	-	NFT4	NFT3	NFT2	NFT1	NFT0
Attribute		-	-	-	R/W	R/W	R/W	R/W	R/W
Initial Value		-	-	-	0	0	0	0	0

[bit7:5] Reserved: Reserved bits

The read value is "0". Be sure to write "0"

[bit4:0] NFT4 to NFT0: Noise Filter Time Select bits

Selects the Noise Filter Times of Serial clock input (SCL) and Serial data input (SDA).

The formula of calculating the noise filter time is as follows:

Noise Filter Time = (NFT+1) × 2 × Bus Clock Frequency Time

For the relationship between noise filter time select bit setting and bus clock frequency, see Table 5-3. Set the noise filter time select bits according to the frequency.

Notes:

- *When ISMK:EN bit of ISMK register is “0”, set these bits.*
- *Any bit setting other than those in Table 5-3 is prohibited.*

Table 5-3 Relationship between Noise Filter Time Select bits and Bus Clock Frequency

bit4	bit3	bit2	bit1	bit0	Bus Clock Frequency［MHz］
0	0	0	0	0	8 MHz or more and less than 40 MHz *1
0	0	0	0	1	40 MHz or more and less than 60 MHz
0	0	0	1	0	60 MHz or more and less than 80 MHz
0	0	0	1	1	80 MHz or more and less than 100 MHz
0	0	1	0	0	100 MHz or more and less than 120 MHz
0	0	1	0	1	120 MHz or more and less than 140 MHz
0	0	1	1	0	140 MHz or more and less than 160 MHz
0	0	1	1	1	160 MHz or more and less than 180 MHz
0	1	0	0	0	180 MHz or more and less than 200 MHz
0	1	0	0	1	200 MHz or more and less than 220 MHz
0	1	0	1	0	220 MHz or more and less than 240 MHz
0	1	0	1	1	240 MHz or more and less than 260 MHz
0	1	1	0	0	260 MHz or more and less than 280 MHz
0	1	1	0	1	280 MHz or more and less than 300 MHz
0	1	1	1	0	300 MHz or more and less than 320 MHz
0	1	1	1	1	320 MHz or more and less than 340 MHz
1	0	0	0	0	340 MHz or more and less than 360 MHz
1	0	0	0	1	360 MHz or more and less than 380 MHz
1	0	0	1	0	380 MHz or more and less than 400 MHz

*1 : In Standard –mode, 2 MHz or more and less than 40 MHz

5.7 Extension I2C Bus Control Register (EIBCR)

The Extension I2C Bus Control Register (EIBCR) is used to control the output of SDA/SCL and set the operation continuity after a bus error occurs.

bit	15	14	13	12	11	10	9	8	7	...	0
Field	Reserved		SDAS	SCLS	SDAC	SCLC	SOCE	BEC	-		
Attribute	-		R	R	R/W	R/W	R/W	R/W			
Initial value	-		0	0	1	1	0	0			

[bit15:14] Reserved: Reserved bits

The read value is "0". Be sure to write "0".

[bit13] SDAS: SDA status bit

This bit indicates the signal level of SDA line after a noise filter.

bit	Description
0	SDA line is in "Low" level.
1	SDA line is in "High" level.

Note:

- *This bit is valid only when I2C is enabled (ISMK:EN=1). When I2C is disabled (ISMK:EN=0), "0" is always read from this bit.*

[bit12] SCLS: SCL status bit

This bit indicates the signal level of SCL line after a noise filter.

bit	Description
0	SCL line is in "Low" level.
1	SCL line is in "High" level.

Note:

- *This bit is valid only when I2C is enabled (ISMK:EN=1). When I2C is disabled (ISMK:EN=0), "0" is always read from this bit.*

[bit11] SDAC: SDA output control bit

When the serial output control is enabled (SOCE=1), this bit controls SDA output.

bit	Description
0	SDA output is in "Low" level.
1	SDA output is in "High" level.

[bit10] SCLC: SCL output control bit

When the serial output control is enabled (SOCE=1), this bit controls SCL output.

bit	Description
0	SCL output is in "Low" level.
1	SCL output is in "High" level.

[bit9] SOCE: Serial output enabled bit

This bit enables the serial output.

When this bit is set to "1", the following operations are executed:

- SDA output is controlled with SDA output control bit (SDAC).
- SCL output is controlled with SCL output control bit (SCLC)

bit	Description
0	Serial output control is disabled.
1	Serial output control is enabled.

Note:

- *Only when IBCR:MSS=0 and IBCR:ACT=0, this bit must be set to "1".*

[bit8] BEC: Bus error control bit

After a bus error occurs (IBSR:BER=1), this bit selects the continuity or abortion of I2C operation.

bit	Description
0	I2C operation is aborted.
1	I2C operation is continued.

Note:

- *When EIBCR:BEC=0, if the restart condition is detected while the address data is being transferred or bit2 to bit9(acknowledge bits) are being transferred after the start condition is detected, a bus error is detected(IBCR:BER=1) and reception is aborted. So, the next data is not received. In this case, after clearing the interrupt flag (IBCR:INT), the re-processing of the start condition from master is required.*

5.8 7-bit Slave Address Mask Register (ISMK)

The 7-bit Slave Address Mask Register (ISMK) is used to compare or set each bit of the slave address.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	(ISBA)
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	1	1	1	1	1	1	1	

[bit15] EN: I2C interface operation enable bit

This bit enables or disables the I2C interface operation.

If set to "0": The I2C interface operation is disabled.
If set to "1": The I2C interface operation is enabled.

bit	Description
0	Disable
1	Enable

Notes:

- *This bit is not cleared to "0" even if the BER bit of IBSR register is set to "1".*
- *The baud rate generator must be set only when this bit is "0".*
- *When this bit is "0", set both the 7-bit Slave Address Register and the 7-bit Slave Address Mask Register.*
- *If the I2C interface operation is disabled (EN=0), data transmission and reception is inhibited immediately.*
- *If you have set the IBCR:MSS bit to "0" to generate a Stop condition and if you wish to disable the I2C interface operation, make sure that the stop condition has occurred. Then, disable the operation (EN=0).*
- *If the EN bit is set to "0" during data transmission, a pulse may be generated on the SDA/SCL signal of the I2C bus.*

[bit14:8] SM6 to SM0: Slave address mask bits

These bits specify to exclude the 7-bit slave address and the received address from comparison.

If set to "1", the address is compared.
If set to "0", the address matching is assumed.

bit14:8	Description
0	Does not compare the bits.
1	Compares the bits.

Note:

- *This register must be set only when the EN bit is "0".*

5.9 7-bit Slave Address Register (ISBA)

The 7-bit Slave Address Register (ISBA) is used to set the slave address.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(ISMK)	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
Attribute		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7] SAEN: Slave address enable bit

This bit enables the slave address detection.

If set to "0": The slave address is not detected.
If set to "1": The ISBA and ISMK settings and the received 1st byte are compared.

bit	Description
0	Disable
1	Enable

[bit6:0] SA6 to SA0: 7-bit slave address

- If the slave address detection is enabled (SAEN=1), the 7-bit Slave Address Register (ISBA) compares the 7-bit data, which has been received after detection of (iteration) start condition, with this register value. If all bits match each other, slave mode is selected and an ACK is output. At this time, the received slave address is set in this register (if SAEN=0, no ACK is output).
- If an address bit is set to "0" in the ISMK register, it is not compared.

bit6:0	Description
	7-bit slave address

Notes:

- *The reserved address cannot be set.*
- *This register must be set only when the EN bit of ISMK register is "0".*

5.10 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0)

Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) are used to set a frequency division ratio of serial clocks.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-	(BGR1)							(BGR0)							
Attribute	-	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
Initial value	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Baud Rate Generator Registers are used to set a frequency division ratio of serial clocks.
The BGR1 register corresponds to the high-order bits, and the BGR0 register corresponds to the low-order bits. The reload value to be counted can be written, and the BGR1/BGR0 set value can be read.
When the reload value is written in Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0), the Reload counter starts its counting.

[bit15] -: Unused bit

This bit value is undefined when read.
This bit has no effect on the operation when written.

[bit14:8] BGR1: Baud Rate Generator Register 1

bit14:8	Description
Write	Writes data in bit8 to bit14 of reload counter.
Read	Reads the BGR1 set value.

[bit7:0] BGR0: BAUD RATE GENERATOR REGISTER 0

bit7:0	Description
Write	Writes data in bit0 to bit7 of reload counter.
Read	Reads the BGR0 set value.

Notes:

- *Data must be written in the Baud Rate Generator Registers (BGR1 and BGR0) by 16-bit data accessing.*
- *The Baud Rate Generator Registers must be set when the EN bit of ISMK register is "0".*
- *The baud rate must be set regardless of master or slave mode selection.*
- *In operation mode 4 (I2C mode), operate the bus clock at a frequency no lower than 8 MHz for Standard-mode/Fast-mode and note that setting of a baud rate generator that exceeds 400 kbps is prohibited. Moreover, for Fast-mode Plus, operate the bus clock at a frequency no lower than 64 MHz for Standard-mode/Fast-mode and note that setting of a baud rate generator that exceeds 1000 kbps is prohibited.*

5.11 FIFO Control Register 1 (FCR1)

The FIFO Control Register (FCR1) is used to select the transmit or received FIFO, enable the transmit FIFO interrupt, and control the interrupt flag.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL	(FCR0)
Attribute	-			R/W	R/W	R/W	R/W	R/W	
Initial value	-			0	-	1	0	0	

[bit15:13] Reserved: Reserved bits

The read value is "0". Be sure to write "0".

[bit12] FLSTE: Re-transmit data lost detection enable bit

This bit enables the FCR0:FLST bit detection.

If set to "0", the FCR0:FLST bit detection is disabled.
If set to "1", the FCR0:FLST bit detection is enabled.

bit	Description
0	Disables the Data Lost detection.
1	Enables the Data Lost detection.

Note:

- *To set this bit to "1", set the FSET bit to "1" first, and then set this bit to "1".*

[bit11] FRIIE: Received FIFO idle detection enable bit

This bit sets to detect the received idle state if the received FIFO contains valid data and if it continues more than 8-bit hours. If the received interrupt is enabled (SCR:RIE=1), a received interrupt is generated when the received idle state is detected.

bit	Description
0	Disables the received FIFO idle detection.
1	Enables the received FIFO idle detection.

Note:

- *In case of using Received FIFO, set this bit to "1".*

[bit10] FDRQ: Transmit FIFO data request bit

This bit requests for the transmit FIFO data.
If this bit is "1", the transmit data is being requested. If the Transmit Interrupt is enabled (FTIE=1) during this time, a transmit FIFO interrupt request is output.

The FDRQ bit is set when:

- The FBYTE (for transmission) is "0" (Transmit FIFO is empty).
- Transmit FIFO is reset.

The FDRQ bit is reset when:

- This bit is set to "0".
- Transmit FIFO is filled with data.

bit	Description
0	Does not request for the transmit FIFO data.
1	Requests for the transmit FIFO data.

Notes:

- *If the FBYTE (for transmission) is "0", this bit cannot be set to "0".*
- *If this bit is "0", the FSEL bit state cannot be changed.*
- *If this bit is set to "1", it has no effect on the operation.*
- *If a read-modify-write instruction is issued, "1" is read.*
- *If a transmit interrupt has occurred and the required data have been written in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".*

[bit9] FTIE: Transmit FIFO interrupt enable bit

This bit enables a transmit FIFO interrupt. If this bit is set to "1", an interrupt occurs when the FDRQ bit is set to "1".

bit	Description
0	Disables the transmit FIFO interrupt.
1	Enables the transmit FIFO interrupt.

[bit8] FSEL: FIFO buffer selection bit

This bit selects the transmit or received FIFO.

bit	Description
0	Set transmit FIFO as FIFO1, and the received FIFO as FIFO2.
1	Set transmit FIFO as FIFO2, and the received FIFO as FIFO1.

Notes:

- *This bit is not cleared by FIFO reset (FCR0:FCL[2:1]=11).*
- *To change this bit state, first disable the FIFO operation (FCR0:FE[2:1]=00).*

5.12 FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO, save the read pointer, and set the data re-transmission.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(FCR1)	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Attribute		-	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		0	0	0	0	0	0	0	0

[bit7] - : Unused bit

When read, "0" is always read.
When writing, always set to "0".

[bit6] FLST: FIFO re-transmit data lost flag bit

This bit shows that the re-transmit data of transmit FIFO has been lost.

The FLST bit is set when:

- If the FLSTE bit of FIFO Control Register 1 (FCR1) is "1", the write pointer of transmit FIFO matches the read pointer which has been saved by the FSET bit, and data is written in the FIFO buffer.

The FLST bit is reset when:

- FIFO is reset (FCL bit is set to "1").
- The FSET bit is set to "1".

If this bit is set to "1", the data which has been saved by the FSET bit and identified by the read pointer is overwritten. The data re-transmission cannot be set by the FLD bit even if an error has occurred. If this bit is set to "1" and if you wish to re-transmit data, first reset FIFO. Then, write data in the FIFO buffer again.

bit	Description
0	No Data Lost has occurred.
1	Data Lost has occurred.

[bit5] FLD: FIFO pointer reload bit

This bit reloads the data, being saved in transmit FIFO by the FSET bit, to the reload pointer. This bit can be used to re-transmit data after a communication error or others have occurred.
When the re-transmission setting has finished, this bit is set to "0".

bit	Description
0	Not reloaded
1	Reloaded

Notes:

- *If this bit is "1", data is being reloaded in the read pointer. Therefore, data writing except for FIFO reset is disabled.*
- *When FIFO is enabled or when data is being transmitted, this bit cannot be set to "1".*
- *Set the SMR:TIE bit to "0" first, and set this bit to "1". Then, enable transmit FIFO and set the SMR:TIE bit to "1".*

[bit4] FSET: FIFO pointer save bit

This bit saves the read pointer value of transmit FIFO.

If the read pointer value is saved before being transmitted and if the FLST bit is "0", the data can be re-transmitted even if a communication error or others have occurred.

If set to "1", the current read pointer value is saved.

If set to "0", it has no effect on the operation.

bit	Description
0	Not saved
1	Saved

Note:

- *This bit can be set to "1" only when the transmit byte count (FBYTE) is "0".*

[bit3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

If this bit is set to "1", the FIFO2 buffer is initialized.

Only the FCR0:FLST bit is initialized, but the other bits of FCR1/0 registers are kept.

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	FIFO2 is reset.	

Notes:

- *Disable the FIFO2 operation first, and then reset the FIFO2 buffer.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The FBYTE2 register has the significant data count of "0".*

[bit2] FCL1: FIFO1 reset bit

This bit resets the FIFO1 value.

If this bit is set to "1", the FIFO1 buffer is initialized.

Only the FCR0:FLST bit is initialized, but the other bits of FCR1/0 registers are kept.

bit	Description	
	At writing	At reading
0	No effect on operation.	"0" is always read.
1	FIFO1 is reset.	

Notes:

- *Disable the FIFO1 operation first, and then reset FIFO1.*
- *Set the transmit FIFO interrupt enable bit to "0" before the execution.*
- *The FBYTE1 register has the significant data count of "0".*

[bit1] FE2: FIFO2 operation enable bit

This bit enables or disables the FIFO2 operation.

- To use the FIFO2 operation, set this bit to "1".
- If received FIFO is selected by the FCR1:FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- To use FIFO2 as transmit FIFO, this bit must be set to "1" or "0" when the transmit data is empty (SSR:TDRE=1).
- To use FIFO2 as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and received FIFO contains no valid data (FBYTE2=0) while the I2C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- To use FIFO2 as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) while the I2C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- The FIFO2 state is held even if the FIFO2 operation is disabled.

bit	Description
0	Disables the FIFO2 operation.
1	Enables the FIFO2 operation.

Notes:

- *The enable or disable state must be switched only when the IBSR:BB bit is "0" or when the IBCR:INT bit is "1".*
- *If received FIFO is selected and the reserved address is detected, and if you wish to select the slave mode transmission, set this bit to "0" and set IBCR:ACKE bit to "0" with an interrupt of reserved address detection.*
- *If received FIFO is selected and if the SSR:RDRF bit of SSR is "1" when this bit is changed from "1" to "0", received FIFO is not disabled until the bit is set to "0".*
- *If transmit FIFO is selected, FIFO2 contains data, and you wish to change this bit from "0" to "1", set the SMR:TIE bit to "0" first. Then, set this bit to "1", and set the SMR:TIE bit to "1".*

[bit0] FE1: FIFO1 operation enable bit

This bit enables or disables the FIFO1 operation.

- To use the FIFO1 operation, set this bit to "1".
- If received FIFO is selected by the FCR1:FSEL bit and if a received error has occurred, this bit is cleared to "0". This bit cannot be set to "1" until the received error is cleared.
- To use FIFO1 as transmit FIFO, this bit must be set to "1" or "0" when the transmit data is empty (SSR:TDRE=1).
- To use FIFO1 as received FIFO, this bit must be set to "0" when the received buffer is empty (SSR:RDRF=0) and received FIFO contains no valid data (FBYTE2=0) while the I2C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- To use FIFO1 as received FIFO, this bit must be set to "1" when the received buffer is empty (SSR:RDRF=0) while the I2C interface operation is disabled (ISMK:EN=0), the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1".
- The FIFO1 state is held even if the FIFO1 operation is disabled.

bit	Description
0	Disables the FIFO1 operation.
1	Enables the FIFO1 operation.

Notes:

- *The enable or disable state must be switched only when the IBSR:BB bit is "0" or when the IBCR:INT bit is "1".*
- *If received FIFO is selected and the reserved address is detected, and if you wish to select the slave mode transmission, set this bit to "0" and set IBCR:ACKE bit to "0" with an interrupt of reserved address detection.*
- *If received FIFO is selected and the SSR:RDRF bit is "1" when this bit is changed from "1" to "0", received FIFO is not disabled until the bit is set to "0".*
- *If transmit FIFO is selected, FIFO1 contains data, and if you wish to change this bit from "0" to "1" state, set the SMR:TIE bit to "0" first. Then, set this bit to "1", and set the SMR:TIE bit to "1".*

5.13 FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) indicates the effective data count in the FIFO buffer. Also, this register can be used to generate a received interrupt when certain number of data sets are received in the received FIFO.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	(FBYTE2)								(FBYTE1)							
Attribute	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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FBYTE register indicates the effective data count in the FIFO buffer. The following table shows the relation between the FCR1:FSEL bit state and FBYTE.

Table 5-4 Display of data count

FSEL	FIFO selection	Data count display
0	FIFO2:Received FIFO, FIFO1: Transmit FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1
1	FIFO2:Transmit FIFO, FIFO1:Received FIFO	FIFO2:FBYTE2, FIFO1:FBYTE1

- The initial value of data transfer count is "0x08" for the FBYTE register.
- Set a data count to generate a received interrupt flag for the FBYTE register of received FIFO. If this transfer data count matches the FBYTE register display, the received data full flag bit (SSR:RDRF) is set to "1".
- If both of the following conditions are satisfied and if the received idle state continues for more than 8 baud rate clocks, the received data full flag bit (SSR:RDRF) is set to "1".
  - The received FIFO idle detection enable bit (FCR:FRIIE) is "1".
  - The number of data sets stored in the received FIFO does not reach the transfer count.

  If the RDR data is read during counting of 8 clocks, this counter is reset to 0 and counting for 8 clocks is restarted. If received FIFO is disabled, this counter is reset to 0. If data remains in the received FIFO and if received FIFO is enabled, the data counting is restarted.
- To receive data in the master mode operation (master mode reception), set the SMR:TIE bit to "0", set the received data count for the FBYTE register of transmit FIFO, and set the FCR1:FDRQ bit to "0". The SCL clocks are output for the specified data count, and then IBCR:INT bit is set to "1". The SMR:TIE bit must be set to "1" only after the FCR1:FDRQ bit is set to "1".

[bit15:8] FBYTE2: FIFO2 data count display bits

[bit7:0] FBYTE1: FIFO1 data count display bits

Writing	Sets the transfer data count.
Reading	Reads the effective count of data.

Read (Effective data count)

During transmission: The number of data sets already written in the FIFO buffer but not transmitted yet
During reception: The number of data sets received in FIFO

Write (Transfer data count)

During transmission: Set "0x00".
During reception: Set the data count to generate a received interrupt.

Table 5-5 DATA Count to be Saved in FIFO

FIFO Capacity	Max. FBYTE Count	Data Count to be Saved
16 BYTEs	16	16
32 BYTEs	32	32
64 BYTEs	64	64
128 BYTEs	128	128

Notes:

- *The FBYTE value of transmit FIFO must be "0x00" except when data is received in the master mode operation.*
- *During the master mode data reception, the transmit data count must be set only when transmit FIFO is empty and the SMR:TIE bit is "0".*
- *When data is being received in the master mode operation, the I^2C interface operation can be disabled (ISMK:EN=0) only after transmit/received FIFO has been disabled.*
- *Setting of a send data number when receiving the data by master operation must be executed when the transmit FIFO is empty and SMR:TIE bit is "0".*
- *The FBYTE bit of received FIFO must be set to "1" or larger.*
- *Change this register under one of the following conditions:*
- *When the I^2C interface operation is disabled (ISMK:EN=0)*
- *When IBCR:INT=1 in case of SSR:DMA=0 and master mode reception*
- *When SSR:TBI=1 in case of SSR:DMA=1 and master mode reception*
- *A read-modify-write instruction cannot be used for this register.*
- *Any setting exceeding the FIFO capacity is inhibited.*
- *To receive data in the master mode operation (master mode reception), do not write dummy data to the Transmit Data Register (TDR) when setting the SMR:TIE bit to "0" and setting the received data count for the FBYTE register of transmit FIFO.*

CHAPTER 2-1: USB/Ethernet Clock Generation Block

This chapter explains the USB/Ethernet clock generation.

1. Overview and Configuration

CODE: 9BFBSPLL_FM4-E01.0

1. Overview and Configuration

Generating USB clock and Ethernet clock

This block generates a 48 MHz USB clock used in USB macro communication and a 50 MHz (RMII)/25 MHz (MII) Ethernet clock used in Ethernet communication.

Since the function and configuration differ by products, see the chapter "USB Clock Generation" for the products other than Ethernet equipped products, and see the chapter "USB/Ethernet Clock Generation" for Ethernet equipped products.

Furthermore, for logic macros of USB and Ethernet mounted in this family, the operation clocks (HCLK) are gated in the logic macro at the initial state for low power consumption.
To use USB or Ethernet function, be sure to change the following register settings to release the clock gating:

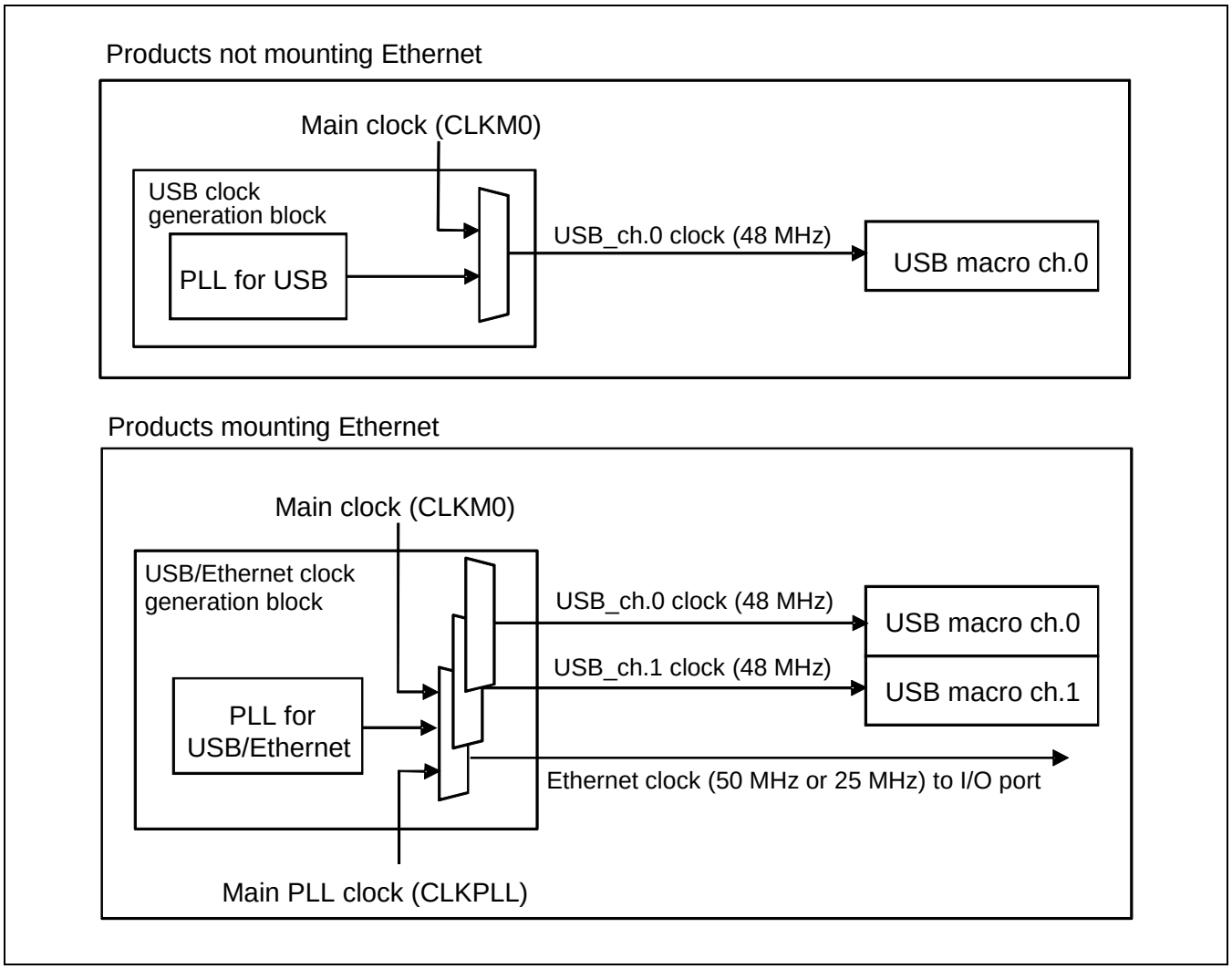
USB ch.0: For details, see 4.5 "peripheral clock control register (CKEN2)" in "Peripheral clock gating function" of "Peripheral Manual".

USB ch.1: For details, see 4.5 "peripheral clock control register (CKEN2)" in "Peripheral clock gating function" of "Peripheral Manual".

Figure 1-1 shows a block diagram of a USB clock and a USB/Ethernet clock generation block.

Block diagram of USB clock and USB/Ethernet clock generation block

Figure 1-1 USB clock and USB/Ethernet clock generation block



CHAPTER 2-2: USB Clock Generation

This chapter explains the USB clock generation.

1. Overview
2. Configuration and Block Diagram
3. Explanation of Operation
4. Setup Procedure Example
5. Register List
6. Usage Precautions

1. Overview

This section provides an overview of the USB clock generation.

The USB clock runs at 48 MHz and is used by USB macro for communication.

The USB clock generating method is selected from the following two methods:

- 48 MHz main clock (hereinafter CLKMO) is used as it is.
- PLL for USB (hereinafter USB-PLL) is used for the clock source..

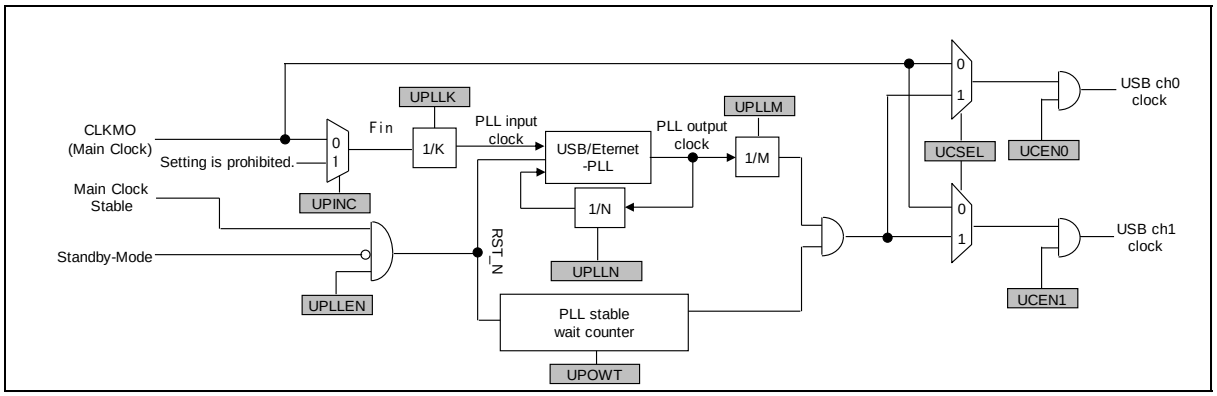
The USB clock generation unit is responsible for the following functions:

- Enables or stops output of the USB clock.
- Selects the USB clock.
- Enables or stops oscillation of USB-PLL.
- Selects the input clock of USB-PLL.
- Sets the input clock frequency division of USB-PLL.
- Sets the output clock multiplication of USB-PLL.
- Sets the stabilization wait time of USB-PLL.
- Stops the USB clock in standby mode.

2. Configuration and Block Diagram

This section explains the configuration and block diagram of the USB clock generation unit.

Figure 2-1 Block diagram of USB clock generation unit



USB-PLL Control Register (UPLLEN)

The control register can enable USB-PLL oscillation.

Input Clock Select Register (UPINC)

Be sure to select the CLKMO.

USB-PLL

- Frequency division setting register (UPLLK, UPLLN, UPLLM)

  To generate 48 MHz as USB clock, the settings of K frequency division, N frequency division and M frequency division are required.

  For the specification range of the USB-PLL input clock frequency, output clock frequency, and multiplier (N division setting value), refer to the PLL use conditions of "PLL input clock frequency", "PLL macro oscillation clock frequency", and "PLL multiplier" in "Data Sheet" of the product used.

- Oscillation stabilization wait time setting (UPOWT)

  Oscillation stabilization wait time for USB-PLL can be specified.

Output clock

- Output Clock Select Register (UCSEL)

  Can be selected from CLKMO or USB-PLL output clock.

- PLL Clock Output Enable Register (UCEN0, UCEN1))

  Can set the USB clock output enable.

Standby mode setting

- The Standby-Mode signal shown in Figure 2-1 turns to be active in the following modes.
  The USB clock stops in the following standby modes.
  - Stop mode
  - TIMER mode
- The Main Clock stable signal shown in Figure 2-1 is an oscillation stabilization signal for each mode.

3. Explanation of Operation

This section explains the operation of the USB clock generation unit.

Selecting the USB clock

The following two types of clocks can be selected for the USB clock.

■ CLKMO

CLKMO can be used directly as the USB clock. In this case, CLKMO must be input externally at 48 MHz, or must oscillate at 48 MHz. Enable the output of the USB clock after confirming stabilization of the CLKMO oscillation.

■ Selecting the USB-PLL output clock

The USB-PLL output clock can be used as the source clock of USB clock.

The USB-PLL output clock must be output at 240 MHz or 288 MHz to generate a 48 MHz clock after M division.

Table 3-1 below shows the setting example of the division ratio.

Table 3-1 Example of PLL frequency division ratio settings

Fin (MHz)	USB Clock Output 48 MHz		
	PLL Output Frequency 240 MHz		
	K	N	M
4	1	60	5
8	1	30	5
8	2	60	5
16	1	15	5
16	2	30	5
16	4	60	5
24	2	20	5
24	4	40	5
24	6	60	5
48	*		

* : Without using USB-PLL, use CLKMO directly as USB clock.

Changing to standby mode

■ When changing to standby mode

Before changing to standby mode (STOP mode, or TIMER mode), set UCEN0 and UCEN1 of UCCR register to "0" to stop the USB clock supply.

1. Set UCCR:UCEN0=0 and UCCR:UCEN1=0.
2. Read the UCCR Register to check that UCEN0 and UCEN1 are set to "0".
3. Changing to standby mode.

When returning from standby mode, set UCEN0 and UCEN1 bits to "1", if required. The supply starts when the USB clock oscillation has been stabilized. Take either of the following actions to confirm whether or not the USB clock oscillation has been stabilized.

a) When USB-PLL is used

Check that UP_STR:UPRDY is "1", or use the USB-PLL oscillation stabilization wait interrupt.

b) When CLKMO (48 MHz) is used

After the CLKMO oscillation has been stabilized, supply the USB clock.

USB-PLL oscillation stabilization wait settings

■ Oscillation stabilization wait time for USB-PLL can be specified

After CLKMO oscillation has been stabilized, the oscillation stabilization wait time for USB-PLL begins to be counted.

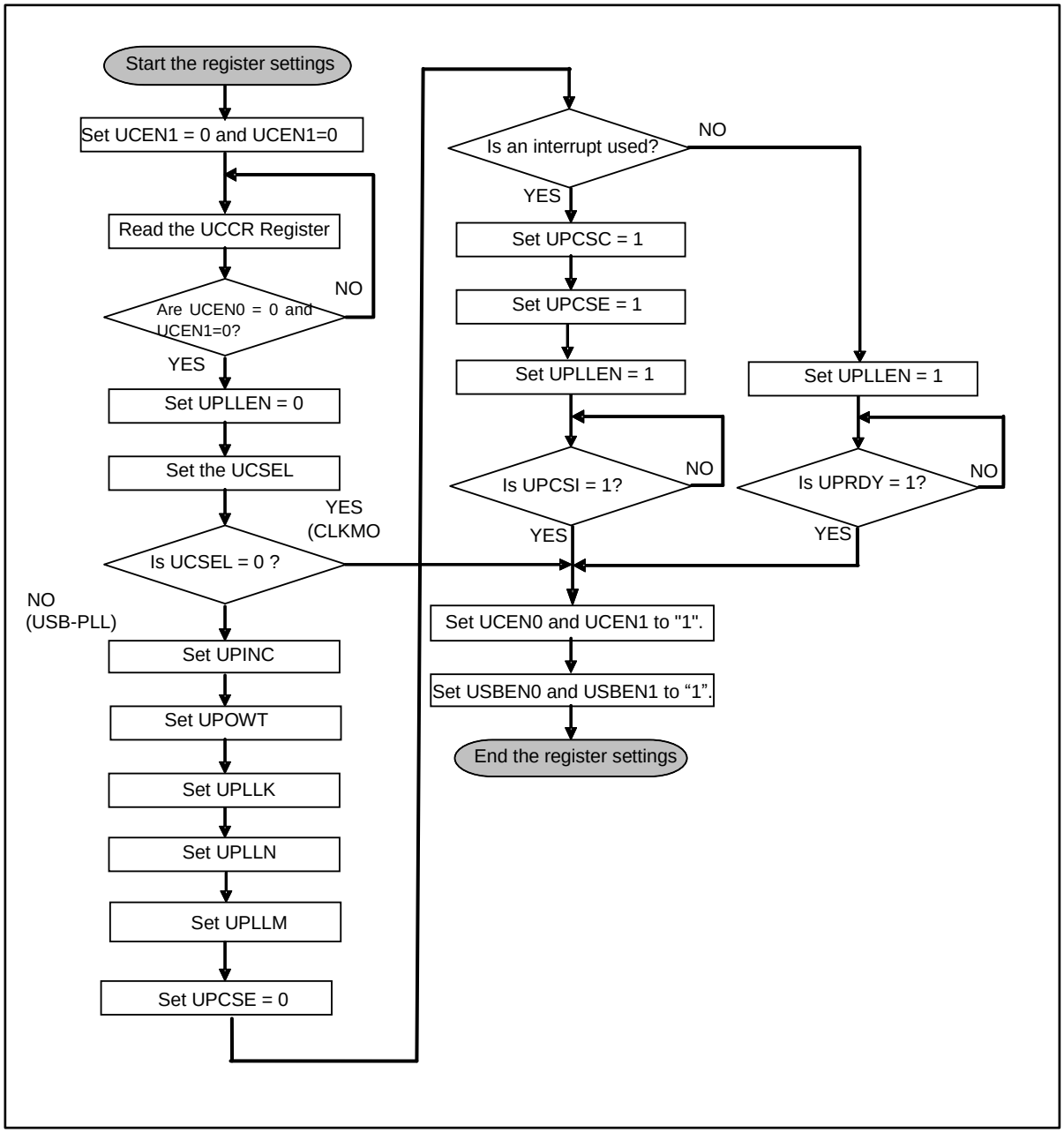
Before enabling the USB-PLL oscillation, configure the oscillation stabilization wait time for USB-PLL and the oscillation stabilization complete interrupt. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

4. Setup Procedure Example

This section explains an example of setting up the USB clock generation unit.

Figure 4-1 shows an example of setting up the USB clock.

Figure 4-1 USB clock generation procedure



5. Register List

This section explains the register list of the USB clock generation unit.

The register list of the USB clock generation unit

Abbreviation	Register name	Reference
UCCR	USB Clock Control Register	5.1
UPCR1	USB-PLL Control Register 1	5.2
UPCR2	USB-PLL Control Register 2	5.3
UPCR3	USB-PLL Control Register 3	5.4
UPCR4	USB-PLL Control Register 4	5.5
UPCR5	USB-PLL Control Register 5	5.6
UP_STR	USB-PLL Status Register	5.7
UPINT_ENR	USB-PLL Interrupt factor Enable Register	5.8
UPINT_STR	USB-PLL Interrupt factor Status Register	5.9
UPINT_CLR	USB-PLL Interrupt factor Clear Register	5.10
USBEN0	USB (ch.0) Enable Register	5.11
USBEN1	USB (ch.1) Enable Register	5.12

5.1 USB Clock Control Register (UCCR)

The UCCR selects the USB clock and enables/disables the USB clock output.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				UCEN1	Reserved	UCSEL	UCEN0
Attribute	-				R/W	-	R/W	R/W
Initial value	-				0	-	0	0

Register functions

[bit7:4] Reserved: Reserved bits

"0b0000" is read from these bits.

Set these bits to "0b0000" when writing.

[bit3]UCEN1: USB(ch.1) clock output enable bit

bit	Description
0	Disables USB(ch.1) clock output. [Initial value]
1	Enables USB(ch.1) clock output.

[bit2]Reserved: Reserved bit

"0" is read from this bit.

Set this bit to "0" when writing.

[bit1] UCSEL: USB clock selection bit

bit	Description
0	CLKMO [Initial value]
1	USB-PLL oscillation clock

[bit0] UCEN: USB clock output enable bit

bit	Description
0	Disables the USB (ch.0) clock output [Initial value]
1	Enables the USB (ch.0) clock output

Notes:

- *When selecting CLKMO as USB clock with UCSEL bit, the 48 MHz frequency must be input from an external main oscillation.*
- *This register is not initialized by software reset.*

5.2 USB-PLL Control Register1 (UPCR1)

The UPCR1 sets USB-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						UPINC	UPLLEN
Attribute	-						R/W	R/W
Initial value	-						0	0

Register functions

[bit7:2] Reserved: Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit1] UPINC: USB-PLL input clock selection bit

bit	Description
0	CLKMO [Initial value]
1	Setting is prohibited.

[bit0] UPLLEN: USB-PLL oscillation enable bit

bit	Description
0	Stops USB-PLL [Initial value]
1	Enables the USB-PLL oscillation

Notes:

- *Be sure to set UPINC to "0". Operation is not guaranteed when UPINC is set to "1".*
- *This register is not initialized by software reset.*

5.3 USB-PLL Control Register 2 (UPCR2)

The UPCR2 sets the oscillation stabilization wait time of USB-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					UPOWT		
Attribute	-					R/W		
Initial value	-					000		

Register functions

[bit7:3] Reserved: Reserved bits

"0b00000" is read from these bits.

Set these bits to "0b00000" when writing.

[bit2:0] UPOWT: USB-PLL oscillation stabilization wait time setting bits

bit2	bit1	bit0	Description
0	0	0	2^9/Fin : Approx. 128 μs * [Initial value]
0	0	1	2^{10}/Fin : Approx. 256 μs *
0	1	0	2^{11}/Fin : Approx. 512 μs *
0	1	1	2^{12}/Fin : Approx. 1.02 ms *
1	0	0	2^{13}/Fin : Approx. 2.05 ms *
1	0	1	2^{14}/Fin : Approx. 4.10 ms *
1	1	0	2^{15}/Fin : Approx. 8.20 ms *
1	1	1	2^{16}/Fin : Approx. 16.4 ms *

* : When F_{in} = 4 MHz

Notes:

- *Fin is the clock (CLKMO) selected by UPINC.*
- *This register is not initialized by software reset.*
- *Since the oscillation stabilization wait time for PLL macro differs by products, refer to the use conditions of "PLL oscillation stabilization wait time" in "Data Sheet" of the product used.*

5.4 USB-PLL Control Register 3 (UPCR3)

The UPCR3 sets the frequency division ratio (K) of USB-PLL macro.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved			UPLLK				
Attribute	-			R/W				
Initial value	-			00000				

Register functions

[bit7:5] Reserved: Reserved bits

"0b000" is read from these bits.

Set these bits to "0b000" when writing.

[bit4:0] UPLLK: Frequency division ratio (K) setting bits of the USB-PLL clock

bit4:0	Description
00000	Divides the frequency by (UPLLK+1). The division ratio of 1 to 32 can be set by using the UPLLK value. (Example) UPLLK = "00000" => 1/1 frequency [Initial value]
00001	
•	
•	
11111	

Note:

- *This register is not initialized by software reset.*

5.5 USB-PLL Control Register 4 (UPCR4)

The UPCR4 Register sets the frequency division ratio (N) of USB-PLL.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved	UPLLN						
Attribute	-	R/W						
Initial value	-	0111011						

■ Register functions

[bit7] Reserved: Reserved bit

"0b0" is read from this bit.

Set this bit to "0b0" when writing.

[bit6:0] UPLLN: Frequency division ratio (N) setting bits of the USB-PLL clock

bit6:0	Description
0000000	Setting is prohibited.
•	
0001100	
0001101	Divides the frequency by (UPLLN+1). The division ratio of 14 to 100 can be set by using the UPLLN value. (Example) UPLLN = "0111011" => 1/60 frequency [Initial value]
•	
•	
1100011	
1100100	Setting is prohibited.
•	
1111111	

Note:

- *This register is not initialized by software reset.*

5.6 USB-PLL Control Register 5 (UPCR5)

The UPCR5 sets the frequency division ratio (M) of USB-PLL.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				UPLLM			
Attribute	-				R/W			
Initial value	-				0100			

■ Register functions

[bit7:4] Reserved: Reserved bits

"0b0000" is read from these bits.

Set these bits to "0b0000" when writing.

[bit3:0] UPLLM: Frequency division ratio (M) setting bits of the USB-PLL clock

bit3:0	Description
0000	Divides the frequency by (UPLLM+1). The division ratio of 1 to 16 can be set by using the UPLLM value. (Example) UPLLM = "0100" => 1/5 frequency [Initial value]
0001	
•	
•	
1111	

Note:

- *This register is not initialized by software reset.*

5.7 USB-PLL Status Register (UP_STR)

The UP_STR indicates the macro status of USB-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPRDY
Attribute	-							R
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] UPRDY: USB-PLL oscillation stabilization bit

bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stabilized state

Note:

- *This register is not initialized by software reset.*

5.8 USB-PLL Interrupt Factor Enable Register (UPINT_ENR)

The UPINT_ENR enables/disables the USB-PLL oscillation stabilization wait complete interrupt.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPCSE
Attribute	-							R/W
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] UPCSE: USB-PLL oscillation stabilization wait complete interrupt enable bit

bit	Description
0	Disables the interrupt [Initial value]
1	Enables the interrupt

5.9 USB-PLL Interrupt Factor Status Register (UPINT_STR)

The UPINT_STR indicates the status of USB-PLL oscillation stabilization wait interrupts.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPCSI
Attribute	-							R
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] UPCSI: USB-PLL interrupt factor status bit

bit	Description
0	No interrupt has occurred [Initial value]
1	An interrupt has occurred

5.10 USB-PLL Interrupt Factor Clear Register (UPINT_CLR)

The UPINT_CLR is used to clear the USB-PLL interrupt factor.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPCSC
Attribute	-							W
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] UPCSC: USB-PLL oscillation stabilization interrupt factor clear bit

bit	Description
0	Disabled [Initial value]
1	Clears the USB-PLL oscillation stabilization wait interrupt.

Note:

- *Writing "1" to UPCSC bit of this register to clear the UPINT_STR Register.*

5.11 USB (ch.0) Enable Register (USBEN0)

The USBEN0 enables/disables USB (ch.0) controller operation.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							USBEN0
Attribute	-							R/W
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000010" is read from these bits.

Set these bits to "0b0000010" when writing.

[bit0] USBEN0: USB (ch.0) enable bit

bit	Description
0	Disables the USB(ch.0) operation (Resets the USB controller) [Initial value]
1	Enables the USB(ch.0) operation

Notes:

- *When using USB(ch.0), set this bit to "1" previously.*
- *Supply at least five cycles of USB clocks to the USB controller before setting this bit to "1".*

5.12 USB (ch.1) Enable Register (USBEN1)

The USBEN1 enables/disables USB (ch.1) controller operation.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							USBEN1
Attribute	-							R/W
Initial Value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000010" is read from these bits.

Set these bits to "0b0000010" when writing.

[bit0] USBEN1: USB (ch.1) enable bit

bit	Description
0	Disables the USB(ch.1) operation (Resets the USB controller) [Initial value]
1	Enables the USB(ch.1) operation

Notes:

- *When using USB (ch.1), set this bit to "1" previously.*
- *Supply at least five cycles of USB clocks to the USB controller before setting this bit to "1".*

6. Usage Precautions

This section explains the precautions for using the clock generation unit.

■ USB clock output setting and USB clock selection

Do not disable the USB (ch.0) clock output (UCEN = 0) and select the USB clock (UCSEL), or disable the USB (ch.1) clock output (UCEN = 1) and select the USB clock (UCSEL) at the same time.
Be sure to disable the USB clock output before selecting the USB clock.

■ Setting the frequency division ratio of USB-PLL oscillation

When the PLL frequency division ratio is changed after stabilization of PLL oscillation, stop the PLL oscillation once, change the frequency division ratio, and then enable the PLL oscillation again.

■ Selecting CLKMO

By writing "0" to the UCSEL bit, CLKMO is selected as the USB clock.
The main clock should be selected when CLKMO oscillates at 48 MHz.

■ Setting the PLL oscillation stabilization wait time

Set the oscillation stabilization wait time with the PLL Oscillation Stabilization Wait Time Setting Register, and then enable PLL. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

■ Selecting the USB-PLL input clock

By writing "1" to the UCSEL bit, the USB-PLL oscillation clock is selected as the USB clock.
Write "0" to the UPINC bit of the USB-PLL Control Register 1 (UPCR1), and be sure to select CLKMO as the USB-PLL input clock.

The following Table 6-1 shows relationship among the USB clock and UCSEL/UPLLEN/UPINC.

Table 6-1 USB clock and register settings

		UCSEL	UPLLEN	UPINC
When using the 48 MHz main clock		0	0	-
When using the PLL macro oscillation clock	CLKMO	1	1	0
	Setting is prohibited.	1	1	1

■ Standby mode and the USB-PLL oscillation stabilization wait counter

If the mode changes to TIMER/STOP mode while waiting for the USB-PLL oscillation to stabilize, USB-PLL stops and the stabilization wait counter is cleared.

■ Setting the USB enable bit and USB controller

To use the USB controller, enable the USB enable bit (USBEN). Supply the USB clock to the USB controller before enabling the USB enable bit (USBEN). For details on USB controller settings, see Chapters "USB Function" and "USB Host".

CHAPTER 2-3: USB/Ethernet Clock Generation

This chapter explains the USB/Ethernet clock generation.

1. Overview
2. Configuration and Block Diagram
3. Description of operation
4. Example of setting procedure
5. List of Registers
6. Usage Precautions

CODE: 9BFUSBETHERPLL_FM4-E01.0

1. Overview

This section explains the overview of the USB/Ethernet clock generation.

The USB clock is a 48 MHz clock used by USB macro to communicate. The Ethernet clock is a 50 MHz (RMII)/25 MHz (MII) clock used for Ethernet communication.

By using this function, a USB (48 MHz) clock and Ethernet (50 MHz/25 MHz) clock can be generated simultaneously.

The following three methods are used to generate a USB/Ethernet clock:

- Using a 48 MHz or 50 MHz/25 MHz main clock (hereafter CLKMO) without change
- Using PLL for USB/Ethernet (hereafter USB/Ethernet-PLL) as a clock source
- Using a main PLL clock (hereafter CLKPLL) as a clock source

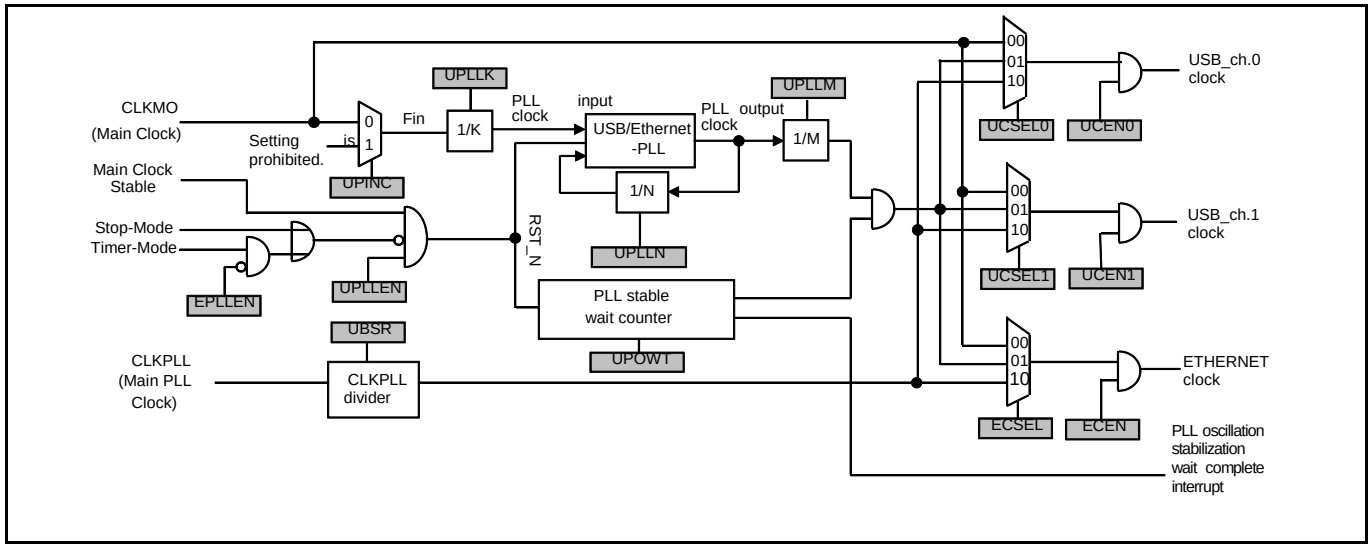
USB/Ethernet clock generation block has the following functions:

- USB/Ethernet clock output enable/disable setting
- Selection of USB/Ethernet clock
- USB/Ethernet-PLL oscillation enable/disable setting
- Selection of USB/Ethernet-PLL input clock
- USB/Ethernet-PLL input clock division setting
- USB/Ethernet-PLL output clock multiplication setting
- USB/Ethernet-PLL stabilization wait time setting
- USB/Ethernet clock stop in standby mode

2. Configuration and Block Diagram

This section describes the configuration of the USB/Ethernet clock generation block and block diagram.

Figure 2-1 Block diagram of USB/Ethernet clock generation block



USB/Ethernet-PLL control register (UPLLEN)

USB/Ethernet-PLL oscillation enable can be set by the control register.

Input clock selection register (UPINC)

CLKMO must be selected.

USB/Ethernet-PLL

- Division setting register (UPLLK, UPLLN, UPLLM)

  To generate 48 MHz as a USB clock or 50 MHz/25 MHz as an Ethernet clock, settings of K division, N division, and M division are required.

  Refer to the use conditions of "PLL input clock frequency", "PLL macro oscillation clock frequency", and "PLL multiplier" in "Data Sheet" of the product used for the specification range of input clock frequency, output clock frequency, and multiplier (N division setting value) of USB/Ethernet-PLL.

- Oscillation stabilization wait time setting register (UPOWT)

  Oscillation stabilization wait time of the USB/Ethernet-PLL can be set.

CLKPLL input

- Division setting register (UBSR)

  Division setting of CLKPLL must be executed.

Output clock

- Output clock selection register (UCSEL0, UCSEL1, ECSEL)

  It can be selected from a CLKMO, USB/Ethernet-PLL output clock, or CLKPLL division clock.

- USB/Ethernet clock output enable register (UCEN0, UCEN1, ECEN)

  USB/Ethernet clock output enable can be set.

Standby mode setting

- Oscillation of USB/Ethernet-PLL stops in TIMER mode or STOP mode. However, if USB/Ethernet-PLL is used as an Ethernet clock (ECSEL[1:0] = 01) and is set to EPLLEN = 1, oscillation stop of USB/Ethernet-PLL will not be executed in TIMER mode.
- The Main Clock stable signals described in Figure 2-1 are oscillation stabilization signals.

3. Description of operation

This section explains the operation of the USB/Ethernet clock generation block.

USB/Ethernet clock selection

A source clock of the USB/Ethernet clock can be selected from the following three types.

■ CLKMO

CLKMO can be directly used as a USB clock or Ethernet clock. In this case, CLKMO needs to be externally input in 48 MHz or 50 MHz/25 MHz, or it needs to oscillate in 48 MHz or 50 MHz/25 MHz. Also, wait for output enable of the USB clock or Ethernet clock after confirming the oscillation stabilization of CLKMO.

■ USB/Ethernet-PLL output clock

The USB/Ethernet-PLL output clock can be used as the source clock of the USB/Ethernet clock.

- When used as USB clock

  USB/Ethernet-PLL output clock must be output in 240 MHz or 288 MHz to generate a 48 MHz clock by M division.

- When used as an Ethernet clock

  The USB/Ethernet-PLL output clock must be output from 200 MHz to 300 MHz to generate a 50 MHz clock or 25 MHz clock by M division.

Note:

- *If it is used as an Ethernet clock, the output clock of USB/Ethernet-PLL must not be divided by three (UPLLM = 0010) due to the specification restriction of Ethernet communication clock duty.*

Table 3-1 shows the setting example of the PLL division ratio.

Table 3-1 Setting example of PLL division ratio

Fin (MHz)	Ethernet clock output 50MHz			Ethernet clock output 25MHz			USB clock output 48MHz		
	PLL output frequency 200MHz			PLL output frequency 200MHz			PLL output frequency 240MHz		
	K	N	M	K	N	M	K	N	M
4	1	50	4	1	50	8	1	60	5
8	1	25	4	1	25	8	1	30	5
16	2	25	4	2	25	8	1	15	5
24	3	25	4	6	50	8	2	20	5
25	5	40	4	*			5	48	5
48	6	25	4	6	25	8	*		
50	*			5	20	8	10	48	5

*: Use CLKMO directly as a USB clock or Ethernet clock without using USB/Ethernet-PLL.

■ CLKPLL

CLKPLL can be divided to be used as a USB clock or Ethernet clock if needed.

Note:

- *If this clock generation block is used as an Ethernet clock, CLKPLL must not be divided by three (UBSR = 0010) due to the specification restriction of Ethernet communication clock duty.*

Transition to standby mode

■ When executing a transition to standby mode

Before executing a transition to standby mode (STOP mode or TIMER mode), set "0" to all UCEN0, UCEN1, and ECEN bits of UCCR register to stop supplying the USB clock and Ethernet clock.

1. Set UCCR:UCEN = 0, UCCR:UCEN1 = 0, and UCCR:ECEN = 0
2. Read UCCR register and confirm that UCEN0, UCEN1, and ECEN bits are "0".
3. Transition to the standby mode

When returning from standby mode, set UCEN0, UCEN1, and ECEN back to "1" if needed. When oscillation of the USB/Ethernet clock stabilizes, it starts supplying. Check the following to know if oscillation of the USB/Ethernet clock stabilizes.

a) When USB/Ethernet-PLL is used

Check if UPRDY = 1, or use USB/Ethernet-PLL oscillation stabilization wait interrupt.

b) When CLKMO (50 MHz/25 MHz or 48 MHz) is used

After stabilization of CLKMO oscillation, the USB/Ethernet clock is provided.

c) When CLKPLL is used

Check if SCM_STR:PLRDY = 1, or use PLL oscillation stabilization wait interrupt (see the chapter "Clock" in "PERIPHERAL MANUAL").

USB/Ethernet-PLL oscillation stabilization wait

■ USB/Ethernet-PLL oscillation stabilization wait time setting

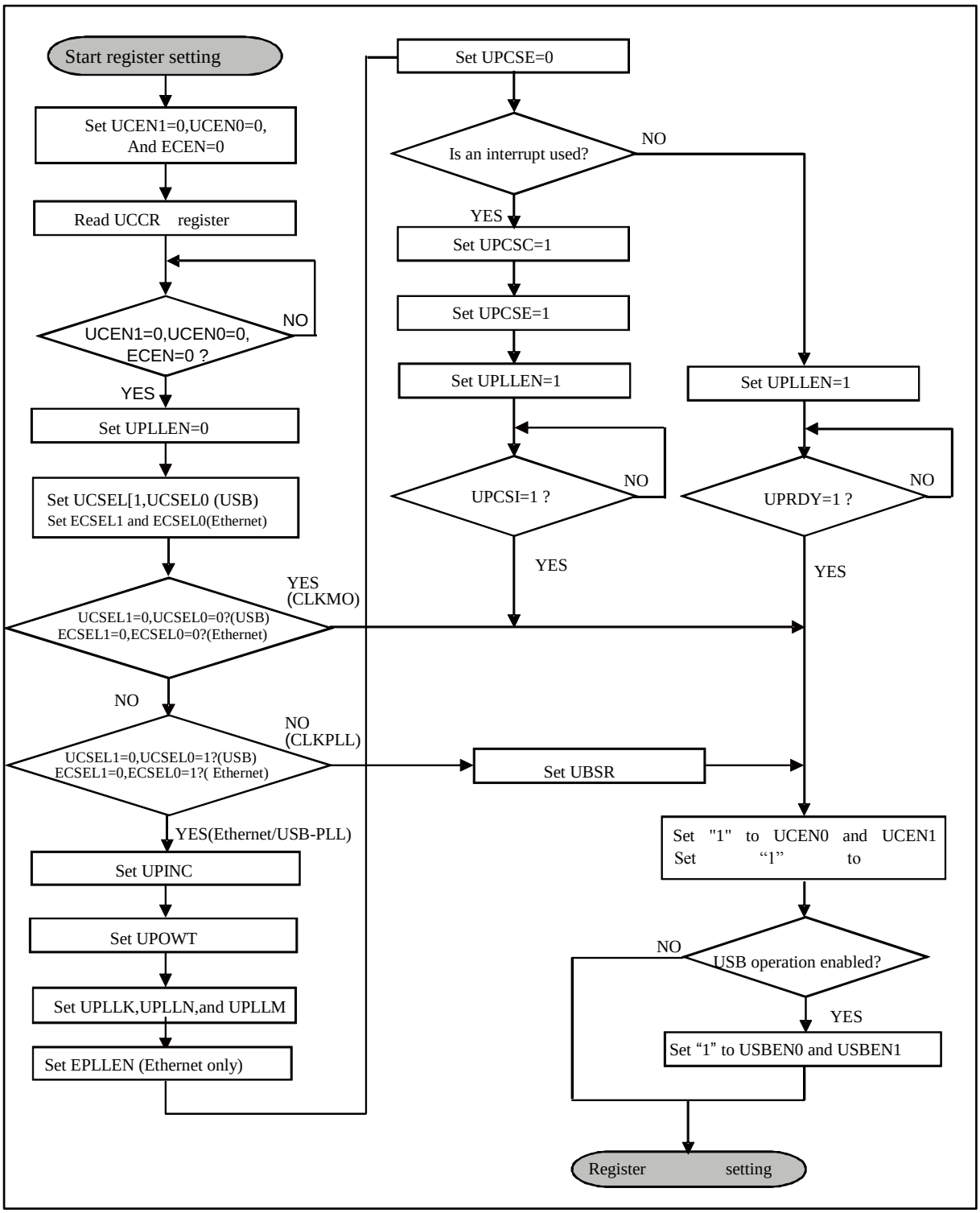
After stabilization of CLKMO oscillation, start counting USB/Ethernet-PLL oscillation stabilization wait time. Before executing USB/Ethernet-PLL oscillation enable, set the USB/Ethernet-PLL oscillation stabilization wait time and oscillation stabilization complete interrupt. Do not change the oscillation stabilization wait time during the oscillation stabilization wait.

4. Example of setting procedure

This section describes the example of the setting procedure for the USB/Ethernet clock generation block.

Figure 4-1 shows the example of the setting procedure for the USB/Ethernet clock.

Figure 4-1 Procedure for USB/Ethernet clock generation



5. List of Registers

This section describes the list of registers for the USB/Ethernet clock generation block.

List of registers for the USB/Ethernet clock generation block

Abbreviation	Register Name	Reference
UCCR	USB/Ethernet clock control register	5.1
UPCR1	USB/Ethernet-PLL control register1	5.2
UPCR2	USB/Ethernet-PLL control register2	5.3
UPCR3	USB/Ethernet-PLL control register3	5.4
UPCR4	USB/Ethernet-PLL control register4	5.5
UPCR5	USB/Ethernet-PLL control register5	5.6
UPCR6	USB/Ethernet-PLL control register6	5.7
UPCR7	USB/Ethernet-PLL control register7	5.8
UP_STR	USB/Ethernet-PLL state register	5.9
UPINT_ENR	USB/Ethernet-PLL interrupt factor enable register	5.10
UPINT_STR	USB/Ethernet-PLL interrupt factor state register	5.11
UPINT_CLR	USB/Ethernet-PLL interrupt factor clear register	5.12
USBEN0	USB (ch.0) enable register	5.13
USBEN1	USB (ch.1) enable register	5.14

5.1 USB/Ethernet Clock Control Register (UCCR)

The UCCR register sets selection for the USB/Ethernet clock and output enable for the USB/Ethernet clock.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved	ECSEL1	ECSEL0	ECEN	UCEN1	UCSEL1	UCSEL0	UCEN0
Attribute	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	0	0	0	0	0	0	0

Register function

[bit7] Reserved: Reserved bit

From this bit, "0" is read.

When writing, set "0".

[bit6:5] ECSEL1/ECSEL0: Ethernet clock selection bits

bit6:5	Description
00	CLKMO [initial value]
01	USB/Ethernet-PLL oscillation clock
10	CLKPLL division clock
11	Reserved

[bit4] ECEN: Ethernet clock output enable bit

bit	Description
0	Disable Ethernet clock output [initial value]
1	Enable Ethernet clock output

[bit3] UCEN1: USB (ch.1) clock output enable bit

bit	Description
0	Disable USB (ch.1) clock output [initial value]
1	Enable USB (ch.1) clock output

[bit2:1] UCSEL1/UCSEL0: USB clock selection bits

bit2:1	Description
00	CLKMO [initial value]
01	USB/Ethernet-PLL oscillation clock
10	CLKPLL division clock
11	Reserved

[bit0] UCEN0: USB (ch.0) clock output enable bit

bit	Description
0	Disable USB (ch.0) clock output [initial value]
1	Enable USB (ch.0) clock output

Notes:

- *To select CLKMO as the USB clock in UCSEL[1:0] bits, input 48 MHz signal from the external main oscillation. Also, to select it as the Ethernet clock, input 50 MHz or 25 MHz signal from the external main oscillation.*
- *This register is not initialized in software reset.*

5.2 USB/Ethernet-PLL Control Register1 (UPCR1)

The UPCR1 register sets PLL for USB/Ethernet.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						UPINC	UPLLEN
Attribute	-						R/W	R/W
Initial value	-						0	0

Register function

[bit7:2] Reserved: Reserved bits

From these bits, "0b000000" is read.

When writing, set "0b000000".

[bit1] UPINC: USB/Ethernet-PLL input clock selection bit

bit	Description
0	CLKMO [initial value]
1	Setting is disabled

[bit0] UPLLEN: USB/Ethernet-PLL oscillation enable bit

bit	Description
0	Stop USB/Ethernet-PLL [initial value]
1	Enable USB/Ethernet-PLL oscillation

Notes:

- *"0" must be set in UPINC. If "1" is set, the operation will not be guaranteed.*
- *This register is not initialized in software reset.*

5.3 USB/Ethernet-PLL Control Register2 (UPCR2)

The UPCR2 register sets the oscillation stabilization wait time of PLL for USB/Ethernet.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					UPOWT		
Attribute	-					R/W		
Initial value	-					000		

Register function

[bit7:3] Reserved: Reserved bits

From these bits, "0b00000" is read.

When writing, set "0b00000".

[bit2:0] UPOWT: USB/Ethernet-PLL oscillation stabilization wait time setting bits

bit2	bit1	bit0	Description
0	0	0	2^9/Fin : Approx. 128 μs* [initial value]
0	0	1	2^{10}/Fin : Approx. 256 μs*
0	1	0	2^{11}/Fin : Approx. 512 μs*
0	1	1	2^{12}/Fin : Approx. 1.02 ms*
1	0	0	2^{13}/Fin : Approx. 2.05 ms*
1	0	1	2^{14}/Fin : Approx. 4.10 ms*
1	1	0	2^{15}/Fin : Approx. 8.20 ms*
1	1	1	2^{16}/Fin : Approx. 16.4 ms*

* : When Fin = 4 MHz.

Notes:

- *Fin is the clock selected in UPINC.*
- *This register is not initialized in software reset.*
- *Since the oscillation stabilization wait time for PLL macro differs by products, refer to the use conditions of "PLL oscillation stabilization wait time" in "Data Sheet" of the product used.*

5.4 USB/Ethernet-PLL Control Register3 (UPCR3)

The UPCR3 register sets the division ratio (K) of PLL for Ethernet/USB.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved			UPLLK				
Attribute	-			R/W				
Initial value	-			00000				

Register function

[bit7:5] Reserved: Reserved bits

From these bits, "0b000" is read.

When writing, set "0b000".

[bit4:0] UPLLK: USB/Ethernet-PLL clock division ratio (K) setting bits

bit4:0	Description
00000	Divided by (UPLLK + 1). The division ratio of 1 to 32 can be set by using the UPLIK value. (example) UPLLK = "00000" ⇒ 1 division [initial value]
00001	
.	
.	
11111	

Note:

- *This register is not initialized in software reset.*

5.5 USB/Ethernet-PLL Control Register4 (UPCR4)

The UPCR4 register sets the division ratio (N) of PLL for USB/Ethernet.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved	UPLLN						
Attribute	-	R/W						
Initial value	-	0111011						

■ Register function

[bit7] Reserved: Reserved bit

From this bit, "0" is read.

When writing, set "0".

[bit6:0] UPLLN: USB/Ethernet-PLL clock division ratio (N) setting bits

bit6:0	Description
0000000	Setting is prohibited.
.	
0001100	
0001101	Divided by (UPLLN + 1). The division ratio of 14 to 100 can be set by using the UPLLN value. (example) UPLLN = "0111011" ⇒ 60 division [initial value]
.	
.	
1100011	
1100100	Setting is prohibited.
.	
1111111	

Note:

- *This register is not initialized in software reset.*

5.6 USB/Ethernet-PLL Control Register5 (UPCR5)

The UPCR5 register sets the division ratio (M) of PLL for USB/Ethernet.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				UPLLM			
Attribute	-				R/W			
Initial value	-				0100			

Register function

[bit7:4] Reserved: Reserved bits

From these bits, "0b0000" is read.

When writing, set "0b0000".

[bit3:0] UPLLM: USB/Ethernet-PLL clock division ratio (M) setting bits

bit3:0	Description
0000	Divided by (UPLLM + 1). The division ratio of 1 to 16 can be set by using the UPLLM value. (example) UPLLM = "0100" ⇒ 5 division [initial value]
0001	
.	
.	
1111	

Note:

- *This register is not initialized in software reset.*

5.7 USB/Ethernet-PLL Control Register6 (UPCR6)

The UPCR6 register sets the division ratio of CLKPLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				UBSR			
Attribute	-				R/W			
Initial value	-				0010			

Register function

[bit7:4] Reserved: Reserved bits

From these bits, "0b0000" is read.

When writing, set "0b0000".

[bit3:0] UBSR: CLKPLL division ratio setting bits

bit3:0	Description
0000	Divided by (UBSR + 1). The division ratio of 1 to 16 can be set by using the USBR value. (example) UBSR = "0010" □ 3 division [initial value]
0001	
.	
.	
1111	

Note:

- *This register is not initialized in software reset.*

5.8 USB/Ethernet-PLL Control Register7 (UPCR7)

The UPCR7 register controls USB/Ethernet-PLL in TIMER mode.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							EPLLEN
Attribute	-							R/W
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000000" is read.

When writing, set "0b0000000".

[bit0] EPLLEN: USB/Ethernet-PLL control bit in Timer mode

bit	Description
0	Stop USB/Ethernet-PLL in TIMER mode.
1	Does not stop USB/Ethernet-PLL in TIMER mode.

Note:

- *This register is not initialized in software reset.*

5.9 USB/Ethernet-PLL State Register (UP_STR)

The UP_STR register indicates the state of USB/Ethernet-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPRDY
Attribute	-							R
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000000" is read.

When writing, set "0b0000000".

[bit0] UPRDY: USB/Ethernet-PLL oscillation stabilization bit

bit	Description
0	Stabilization wait or oscillation stop state [initial value]
1	Stabilization state

Note:

- *This register is not initialized in software reset.*

5.10 USB/Ethernet-PLL Interrupt Factor Enable Register (UPINT_ENR)

The UPINT_ENR register sets enable/disable of USB/Ethernet-PLL oscillation stabilization wait complete interrupt.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPCSE
Attribute	-							R/W
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000000" is read.
When writing, set "0b0000000".

[bit0] UPCSE: USB/Ethernet-PLL oscillation stabilization wait complete interrupt enable bit

bit	Description
0	Disable generation of interrupt [initial value]
1	Enable generation of interrupt

5.11 USB/Ethernet-PLL Interrupt Factor State Register (UPINT_STR)

The UPINT_ENR register indicates the state of USB/Ethernet-PLL oscillation stabilization wait interrupt.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPCSI
Attribute	-							R
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000000" is read.

When writing, set "0b0000000".

[bit0] UPCSI: USB/Ethernet-PLL interrupt factor state bit

bit	Description
0	Interrupt does not occur [initial value]
1	Interrupt occurs

5.12 USB/Ethernet-PLL Interrupt Factor Clear Register (UPINT_CLR)

The UPINT_ENR register sets USB/Ethernet-PLL interrupt factor clear.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							UPCSC
Attribute	-							W
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000000" is read.

When writing, set "0b0000000".

[bit0] UPCSC: USB/Ethernet-PLL oscillation stabilization interrupt generation factor clear bit

bit	Description
0	Null [initial value]
1	Clear the USB/Ethernet-PLL oscillation stabilization wait interrupt

Note:

- *If this register is written and cleared, the UPINT_STR register will be cleared.*

5.13 USB (ch.0) Enable Register (USBEN0)

The USBEN0 register sets operation enable of USB (ch.0) controller.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							USBEN0
Attribute	-							R/W
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000010" is read.

When writing, set "0b0000010".

[bit0] USBEN0: USB (ch.0) enable bit

bit	Description
0	Disable USB (ch.0) operation (reset USB controller block) [initial value]
1	Enable USB (ch.0) operation

Notes:

- *To use a USB (ch.0), firstly set "1" to this bit.*
- *Set "1" after supplying more than 5 cycles of the USB clock to the USB controller.*

5.14 USB (ch.1) Enable Register (USBEN1)

The USBEN register sets operation enable of USB (ch.1) controller.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							USBEN1
Attribute	-							R/W
Initial value	-							0

Register function

[bit7:1] Reserved: Reserved bits

From these bits, "0b0000010" is read.

When writing, set "0b0000010".

[bit0] USBEN1: USB (ch.1) enable bit

bit	Description
0	Disable USB (ch.1) operation (reset USB controller block) [initial value]
1	Enable USB (ch.1) operation

Notes:

- *To use a USB (ch.1), firstly set "1" to this bit.*
- *Set "1" after supplying more than 5 cycles of the USB clock to the USB controller.*

6. Usage Precautions

■ This section describes precautions for the clock generation block.

USB clock output setting and selection of USB clock

Do not execute USB (ch.0) clock output disable (UCEN0 = 0) and USB clock selection (UCSEL0, UCSEL1), or USB (ch.1) clock output disable (UCEN1 = 0) and USB clock selection (UCSEL0, UCSEL1) simultaneously.
Make sure to execute first USB clock output disable and then USB clock selection.

■ Division ratio setting of USB/Ethernet-PLL oscillation

To change PLL division ratio after stabilization of PLL oscillation, firstly stop PLL oscillation. After changing the division ratio, enable PLL oscillation again.

■ Selection of CLKMO

If UCSEL0 = 0 and UCSEL1 = 0 are set, CLKMO will be selected for the USB/Ethernet clock. Select CLKMO only when CLKMO is oscillating at 48 MHz (used in USB) or 50 MHz/25 MHz (used in Ethernet).

■ USB/Ethernet-PLL oscillation stabilization wait time setting

Enable PLL after setting the oscillation stabilization wait time in the PLL oscillation stabilization wait time setting register. Do not change the oscillation stabilization wait time while in oscillation stabilization wait.

■ Selection of USB/Ethernet-PLL input clock

A source clock of the USB clock and Ethernet clock can be selected by UCSEL0 and UCSEL1 settings and ECSEL0 and ECSEL1 settings. Also, a separate source clock can be specified for the USB clock and Ethernet clock.

Table 6-1 shows the setting values of registers related to source clock selection.

Table 6-1 List of register settings for each USB/Ethernet clock source selection

USB clock source	CLKMO (48 MHz)		USB/Ethernet-PLL output clock		CLKPLL	
Ethernet clock source	USB/Ethernet-PLL output clock	CLKPLL	CLKMO (50 MHz/25 MHz)	CLKPLL	CLKMO (50MHz/25 MHz)	USB/Ethernet-PLL output clock
Setting value	UCSEL1 = 0 UCSEL0 = 0 ECSEL1 = 0 ECSEL0 = 1 UPLLEN = 1	UCSEL1 = 0 UCSEL0 = 0 ECSEL1 = 1 ECSEL0 = 0 UPLLEN = 1	UCSEL1 = 0 UCSEL0 = 1 ECSEL1 = 0 ECSEL0 = 0 UPLLEN = 1	UCSEL1 = 0 UCSEL0 = 1 ECSEL1 = 1 ECSEL0 = 0 UPLLEN = 1	UCSEL1 = 1 UCSEL0 = 0 ECSEL1 = 0 ECSEL0 = 0 UPLLEN = 1	UCSEL1 = 1 UCSEL0 = 0 ECSEL1 = 0 ECSEL0 = 1 UPLLEN = 1

■ Standby mode and USB/Ethernet-PLL oscillation stabilization wait counter

By executing a transition to TIMER/STOP mode during USB/Ethernet-PLL oscillation stabilization wait time, PLL stops and the stabilization wait counter is cleared (except for TIMER mode when EPLLEN = 1 and ECSEL[1:0] = 01).

■ Settings of USB enable bit and USB controller

When using the USB controller, enable USB enable bit (USBEN). Also, enable USB enable bit (USBEN) after supplying the USB clock to the USB controller. As for the details on the USB controller setting, see the chapters "USB Function" and "USB Host".

CHAPTER 3-1: USB Function

This chapter explains the USB function.

1. Overview of USB Function
2. Configuration of USB Function
3. Operations of USB Function
4. Examples of USB Function Setting Procedures
5. USB Function Registers

CODE: FW03F-E19.5

1. Overview of USB Function

The USB function is an interface supporting the USB (Universal Serial Bus) communication protocol. It supports full-speed transfer mode (12 Mbps), and has the following features.

1.1 Features of USB function

- Full-speed (12 Mbps) transfer supported.
- Auto answered device status.
- Automatic generation and check of bit stripping, bit stuffing, CRC5, and CRC16.
- Toggle check by data synchronization bit.
- Auto-answer to all standard commands other than the Get/SetDescriptor and SynchFrame commands (these three commands can be processed similarly as class vendor commands).
- The class vendor commands can be received as data and responded by firmware.
- Up to 6 Endpoints supported. (Endpoint 0 is fixed to control transfer)
- Each Endpoint includes 2 buffers for data transfer.
  (Endpoint 0 includes each buffer exclusively for IN and OUT directions)
- Automatic data transfer via DMA supported (except Endpoint 0 buffers).

Note:

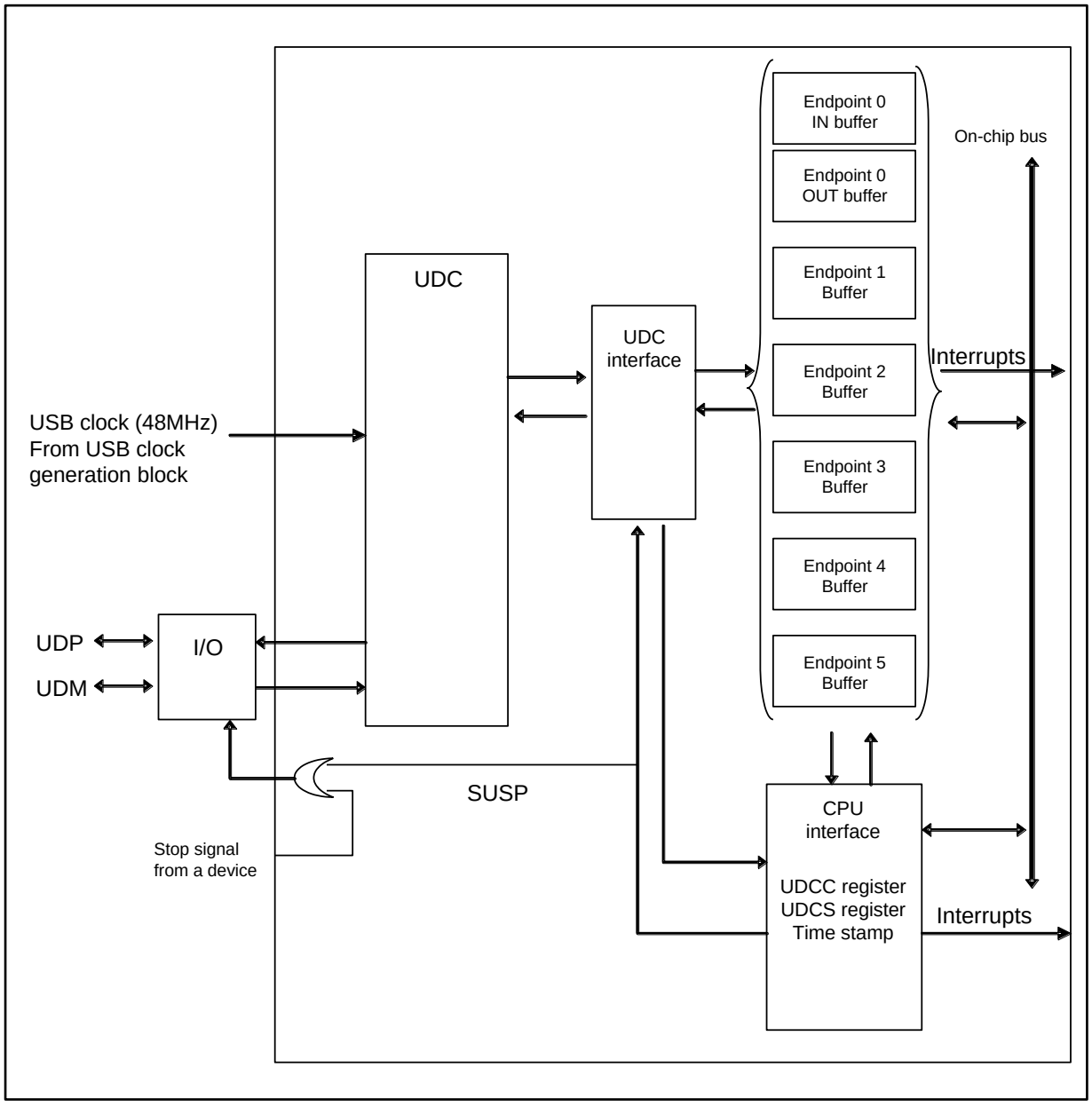
- *Set the base clock (HCLK) to 13 MHz or higher when using the USB function.*

2. Configuration of USB Function

Figure 2-1 shows the block diagram of the USB function.

USB function block diagram

Figure 2-1 USB function block diagram



Configuration of endpoint for USB function

Configuration combination	Configuration	Interface	Alternate	Endpoint	Type
Comb1	-	-	-	0	CTRL
	1	0	0	1	Bulk/Interrupt
		0	0	2	Bulk/Interrupt
		0	0	3	Bulk/Interrupt
		0	0	4	Bulk/Interrupt
		0	0	5	Bulk/Interrupt
Comb2	-	-	-	0	CTRL
	1	1	0	-	- (*1)
		1	1	1	ISO
		0	0	2	Bulk/Interrupt
		0	0	3	Bulk/Interrupt
		0	0	4	Bulk/Interrupt
		0	0	5	Bulk/Interrupt
Comb3	-	-	-	0	CTRL
	1	1	0	-	- (*1)
		1	1	1	ISO
		2	0	-	- (*1)
		2	1	2	ISO (*2)
		0	0	3	Bulk/Interrupt
		0	0	4	Bulk/Interrupt
		0	0	5	Bulk/Interrupt

Comb1: Configuration when ISO is not set to Types of Endpoint1 and Endpoint2

Comb2: Configuration when ISO is set to Type of Endpoint1

Comb3: Configuration when ISO is set to Types of Endpoint1 and Endpoint2

*1 : When isochronous is set, the endpoint does not exist for Alternate=0.
Set "0" for the number of interface descriptor endpoints for Alternate=0.

*2 : When ISO is set to Type of Endpoint2, ISO must be also set to Type of Endpoint1.

3. Operations of USB Function

The USB function supports the USB (Universal Serial Bus) communication protocol. Its hardware supports the basic protocol operation (handshake). Therefore, USB communication can be implemented by processing only transfer data.

3.1. USB function operation
3.2. Detection of connection and disconnection
3.3. Operation of each register in response to a command
3.4. Suspend function
3.5. Wake-up function
3.6. DMA transfer function
3.7. NULL transfer function
3.8. STALL response/release of Endpoint 0
3.9. STALL response/release of Endpoint 1 to Endpoint 5

3.1 USB function operation

To use the USB function, take the following steps for setup.

1. Configure the USB clock generation block while the USB Enable Register (USBEN) disables USB operation (USBEN = 0).
2. Enable the USB clock output.
3. Enable USB operation (USBEN = 1).

The USB function transfers packets bi-directionally to/from a host controller that supports the USB protocol. Connection with the host and devices, and configuration are enumerated. Communications are implemented subsequently in different transfer types using device drivers.

The following explains the operation of USB communication between the host and devices by taking an enumeration for example.

Behaviors of registers and USB packets are shown here to provide details of the entire process.

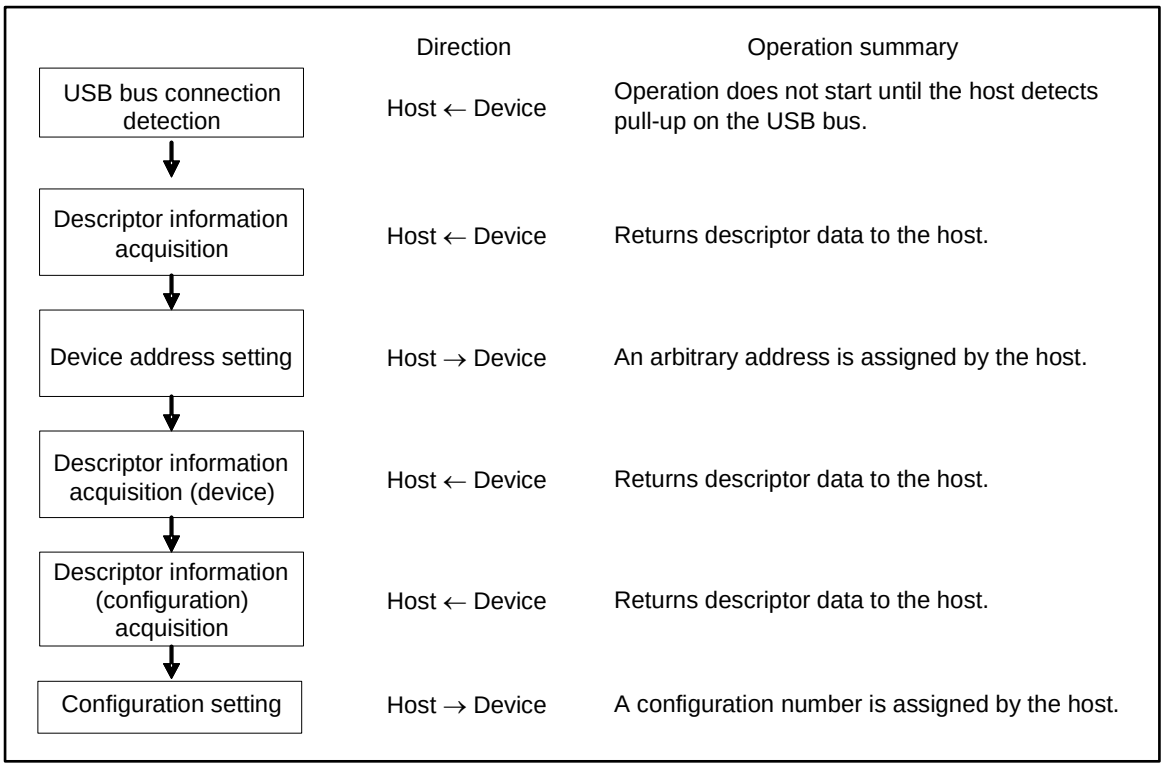
Enumeration

Enumeration is the first process for USB operation to establish connection between the host and devices. The host investigates what types of devices are connected on the USB bus by using USB control transfer (a USB transfer type). (Defined in the USB specification) This process uses EP0 (Endpoint 0) from the six Endpoints (as defined in the USB specification).

To use EP1 to EP5, reception and processing on the USB bus are required in the following order:

1. Resetting the USB bus
2. Setting the address by SET_Address
3. Setting configuration by SET_Config

Figure 3-1 Example of USB cable pin connection



- USB bus connection detection

  The connection is reported from a device to the host.
  The host monitors two signal lines (D+ and D-) on the USB bus, and finds the connection of a device if either of the signals turns to HIGH level.
  For a detailed procedure explaining how to use the device in self-powered mode, see "3.2 Detection of connection and disconnection". To use the device as bus-powered, follow the procedure given in "I Initial register setting and operation start procedures".

■ Initial register setting and operation start procedures

The following shows an example initial setting procedure of USB function registers.

1. Set EP0 configuration (such as packet size) by the EP0C register.
2. Set EPEN, DIR, or TYPE of each Endpoint by the EP1C to EP5C registers.
3. Clear the RST bit in the UDCC register.
4. Clear BFINI in the EP0IS, EP0OS, and EP1S to EP5S registers.
5. Clear the HCONX bit in the UDCC register.

■ USB bus reset

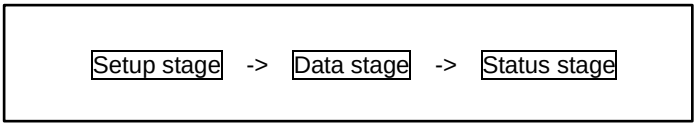
The USB device core is initialized when the host executes a bus reset on the device, but register and buffer states are not initialized.
Take the following steps to process the device. (The process is not required in the initial bus reset after USB connection.)

1. Initialize the buffer by the BFINI bit in the EP0I Status Register (EP0IS), the BFINI bit in the EP0O Status Register (EP0OS), and the BFINI bit in the EP1 to EP5 Status Registers (EP1S to EP5S).
2. Return firmware control to the state before the enumeration.

■ Descriptor acquisition

When the host requests a device, the device reports data to the host in reply to the request.
The communication is broken up into the following three stages.

Figure 3-2 Communication stages



The setup stage checks whether the device has received the packets from the host successfully and decodes the command. The descriptor information to be returned in the next data stage is prepared in the send buffer in this stage. The data stage checks whether the host has sent data successfully. In the status stage, the host sends a packet without data to end the transfer.

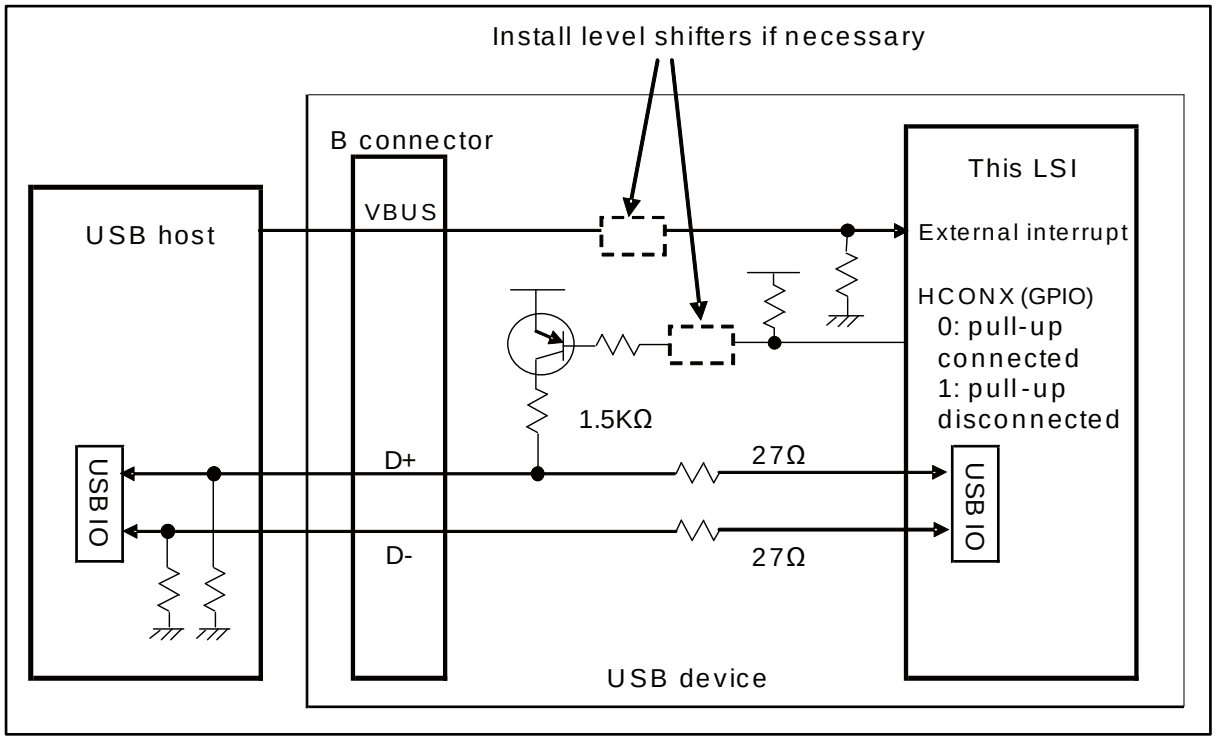
3.2 Detection of connection and disconnection

The following explains about detecting connection and disconnection to/from the USB host.

Example of USB system connection

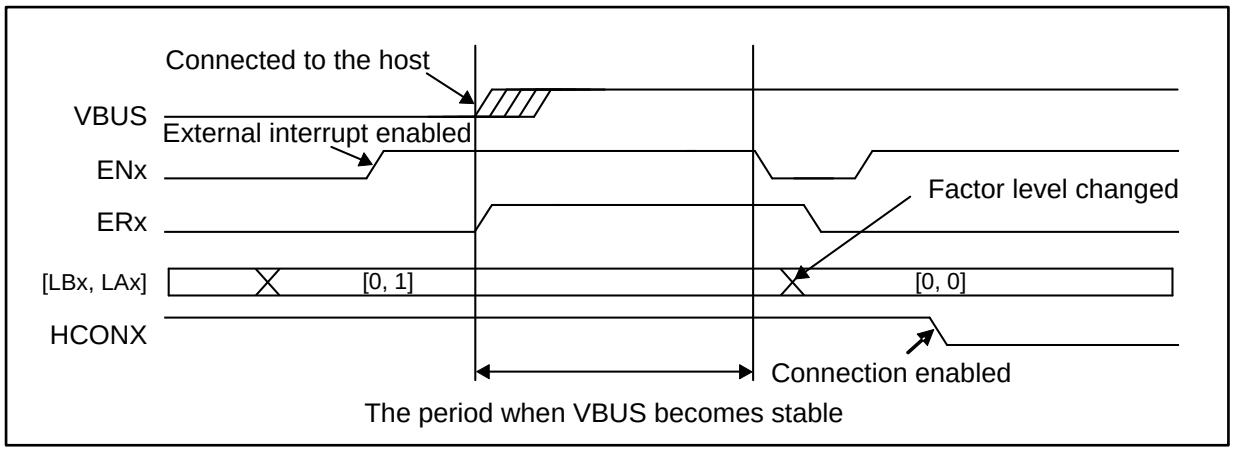
By connecting an external interrupt pin to the VBUS pin of the USB connector, and installing a pull-down resistor onto the VBUS signal, disconnection from the USB host can be detected. Figure 3-3 shows an example connection of USB connector with D+, D- and VBUS.

Figure 3-3 Example of USB system configuration



■ Connection detection

Figure 3-4 Connection detecting operation



A device finds and processes the connection with the host in the following sequence:

1. The HCONX bit in the UDCC register must be set to "1". (When controlling a pull-up resistor on a general-purpose port, set the port to the pull-up resistor disconnection.)
2. Set the source level of external interrupts connected with VBUS to HIGH level detection to enable interrupts.
3. Find the USB host connection by the detection of HIGH level of the external interrupt pin, and waits for the period the VBUS becomes stable.
4. Disable external interrupts once. Change the external interrupt factor level to LOW to clear the interrupt source, and enable external interrupts again.
5. Configure the initial settings (Initialize all components including the USB function registers.)See "Initial register setting and operation start procedures" in this section.
6. Connect the pull-up resistor to D+ by clearing[*1] the HCONX bit in the UDCC register.[*2]

*1 : When control the pull-up resistor on a general-purpose port, clear the HCONX bit in the UDCC register, and set the pull-up resistor control general-purpose port to the pull-up resistor connection.

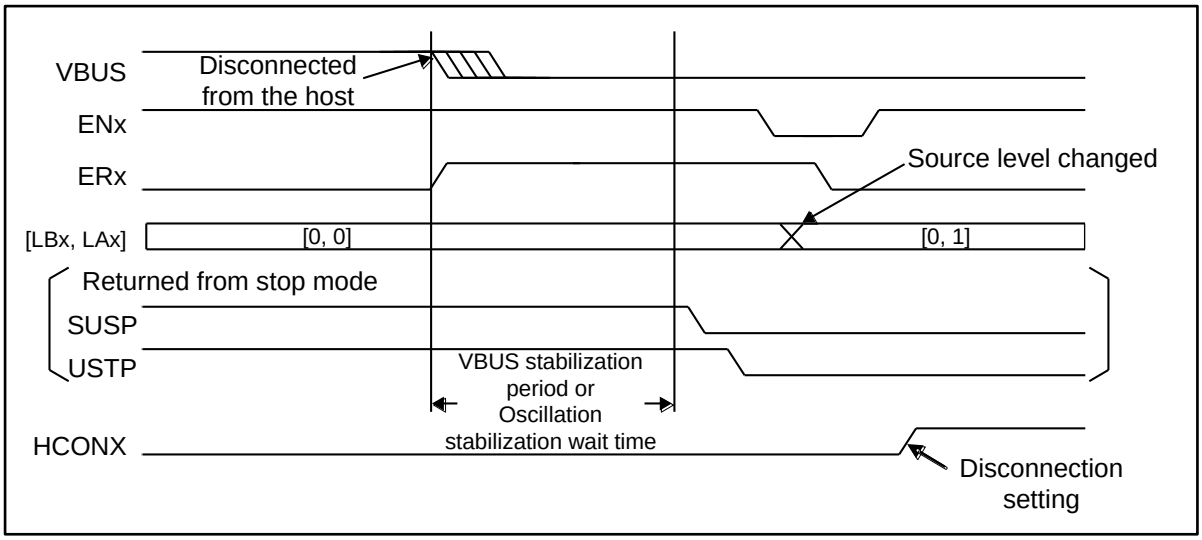
*2 : Clear the HCONX bit even if the pull-up resistor is not controlled.

Note:

- *If an external noise filter is installed on the external interrupt pin, the above VBUS stabilization period does not need to be set by the program.*

■ Disconnection detection

Figure 3-5 Disconnection detecting operation



A device finds and processes the disconnection from the host in the following sequence:

1. Find the disconnection of the USB host by detecting LOW level of the external interrupt pin connected to VBUS.
2. When returned from stop mode or timer mode
   After the oscillation stabilization wait time, clear in the order of SUSP in the UDCS register and USTP in the UDCC register.
   In other than stop mode and timer mode wait for the period the VBUS becomes stable.
3. Disable external interrupts once. Change the external interrupt factor level to HIGH to clear the interrupt factor, and enable external interrupts again.
4. Disconnect the pull-up resistor from D+ by setting[*1] the HCONX bit in the UDCC register.[*2]

*1 : When controlling the pull-up resistor on a general-purpose port, set the HCONX bit in the UDCC register, and set the pull-up resistor control general-purpose port to the pull-up resistor disconnection.

*2 : Set the HCONX bit even if the pull-up resistor is not controlled.

Note:

- *If an external noise filter is installed on the external interrupt pin, the above VBUS stabilization period does not need to be set by the program.*

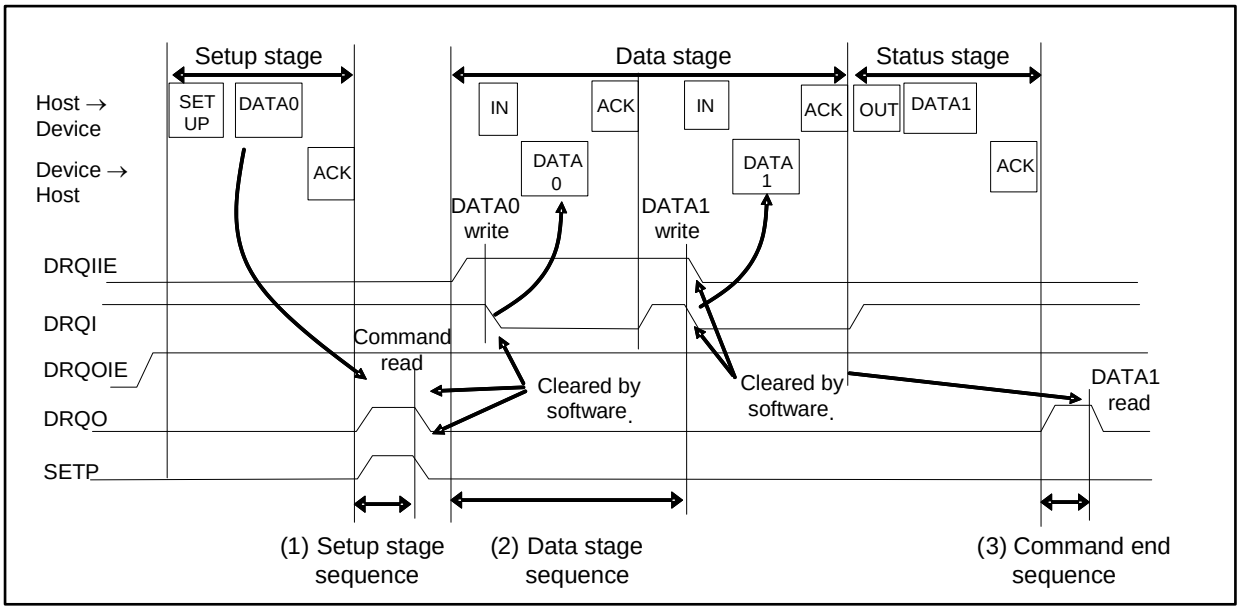
3.3 Operation of each register in response to a command

The following explains the method (architecture) to process USB packets. Responding to CPU interrupts, the firmware sequence is processed for each handshake. This is equivalent to the processing of each packet on the stage basis.

Operation of each register in response to a read command

The following explains the case of GetDescripter, SynchFrame, and class vendor commands.

Figure 3-6 Operation of Each Register in Response to a Read Command



(1) Setup stage sequence

Upon the receipt of the setup stage, DRQO changes to "1". Immediately when DRQO has changed, enter the CPU interrupt and check the SETP flag. If the flag is "1", read required bits of the command in the receive buffer. (Not necessarily read all the eight bytes.) Subsequently, decode the command, configure required settings, clear the SETP flag and the DRQO interrupt factor, and return.

(2) Data stage sequence

If the command decoding concludes that the data stage is in the IN direction, enable DRQIE,* and transfer outgoing data to the send buffer by the CPU interrupt. When the transfer has finished, clear the DRQI interrupt factor, and return.

*: The DRQI interrupt factor is initially set to "1", and is only used to enable interrupts.

DRQI is set when the data packet to the IN direction has finished. The CPU interrupt is entered immediately when DRQI has been set, and outgoing data is transferred to the send buffer in preparation for the next data packet. When the transfer has finished, clear the interrupt source DRQI, and return.

(3) Command end sequence

DRQO is set when the status stage to OUT direction has finished. Immediately when DRQO is set, enter the CPU interrupt and check that the number of received data units is 0. In preparation for the next setup stage, clear the interrupt factor DRQO, and return.

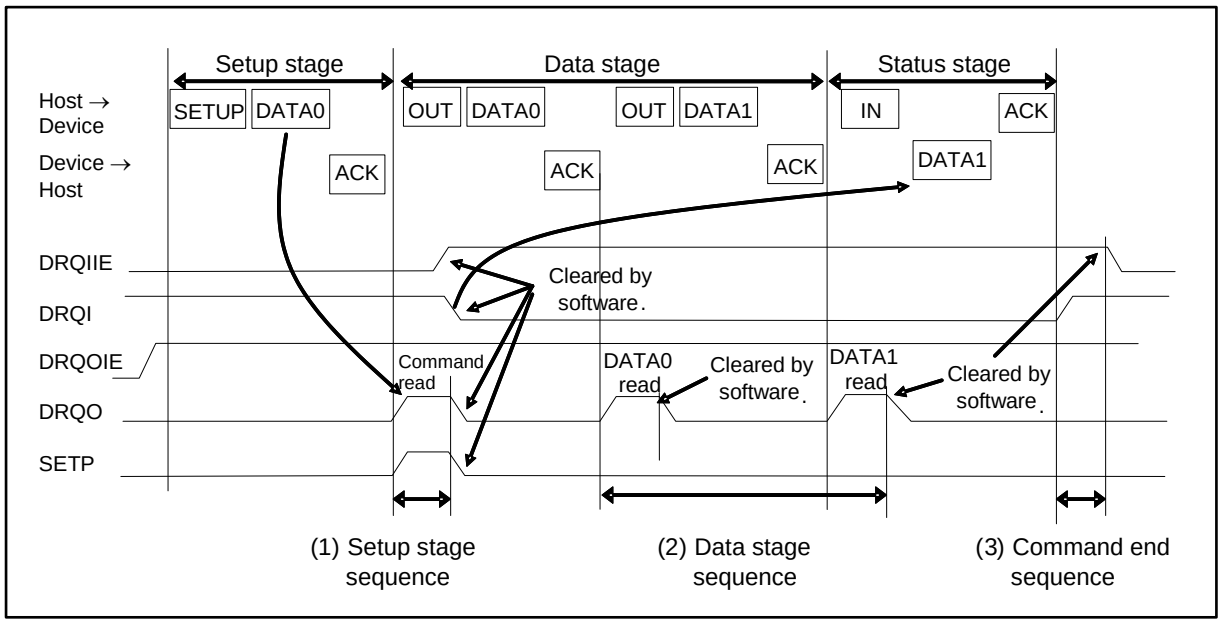
Note:

- *When next setup stage is received without "(3) Command end sequence" being carried out due to the process of an interrupt which has higher priority than USB, the device makes no response to the next setup stage. In order to avoid this phenomenon, carry out any of the following.*
  - *Increase the interrupt priority of the Setup stage, Data stage and Command end sequence*
  - *Continue the process of the IN transfer interrupt in the Data stage sequence until DRQO is cleared in the Command end sequence.*

Operation of each register in response to a write command

The following explains the case of SetDescripter and class vendor commands.

Figure 3-7 Operation of Each Register in Response to a write Command



(1) Setup stage sequence

Upon the receipt of the setup stage, DRQO changes to "1". Immediately when DRQO has changed to "1", enter the CPU interrupt and check the SETP flag. If the flag is "1", read required bits of the command in the receive buffer. (Not necessarily read all the eight bytes.) Subsequently, decode the command, configure required settings.

In preparation of 0-byte response in the status stage, do not write data to the send buffer, and set DRQI to "0" (as the DRQI interrupt factor is initially set to "1"). Set the DRQIIE to "1" to check a successful completion of the status stage. Clear the SETP flag and the DRQO interrupt factor to return from the interrupt.

(2) Data stage sequence

DRQO is set when the data packed to OUT direction has finished. Immediately when DRQO is set, enter the CPU interrupt and check SIZE in the EP0 Status Register. Use DMA limited to received data, or use CPU read access to read data from the receive buffer. Subsequently, clear interrupt factor DRQO to return from the interrupt.

(3) Command end sequence

DRQI is set when the status stage to the IN direction has finished. Immediately when DRQI is set, enter the CPU interrupt and check that the status stage has finished successfully. Subsequently, clear interrupt factor DRQI, and return.

3.4 Suspend function

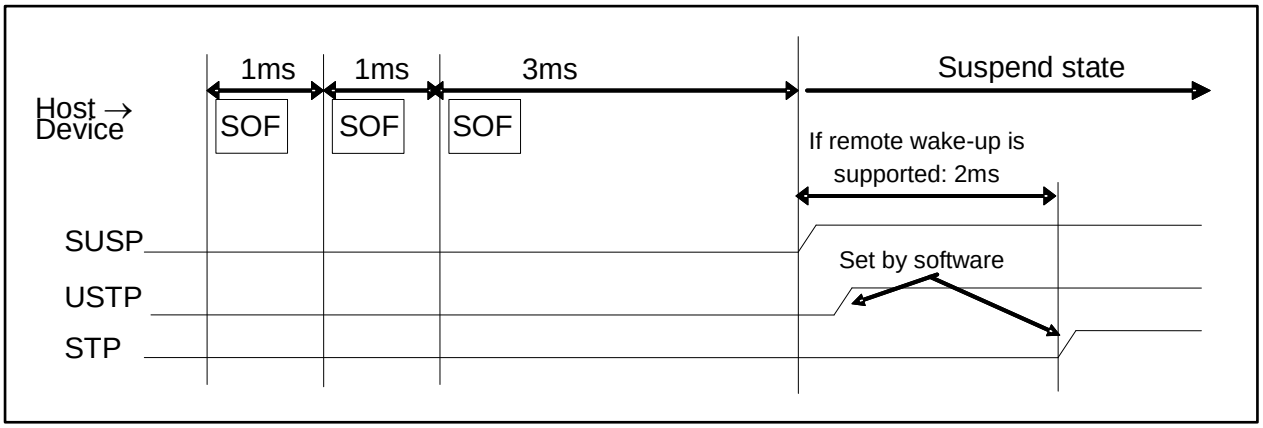
Depending on the bus power configuration, USB devices must drop the power consumption to 500 μA or less in suspend state. The following explains the sequence the USB device makes transition to suspend state, and then stop mode or timer mode.

Suspend sequence

When the USB device core detects a suspend state, SUSP bit in the UDCS register is enabled.

The following provides an example sequence.

Figure 3-8 Suspend operation



- Suspend sequence

  When there is a 3 ms or longer period of inactivity on the USB bus, the USB function detects a suspend state, and sets the SUSP bit interrupt factor in the UDCS register. For devices supporting remote wake-up function, the USB function waits 2 ms more * and sets stop mode or timer mode.

  *: This period is required to block remote wake-up.

Note:

- *Before stop mode or timer mode is entered, set UDCIE:SUSPIE = 0 and UDCC:USTP = 1 in this order.*

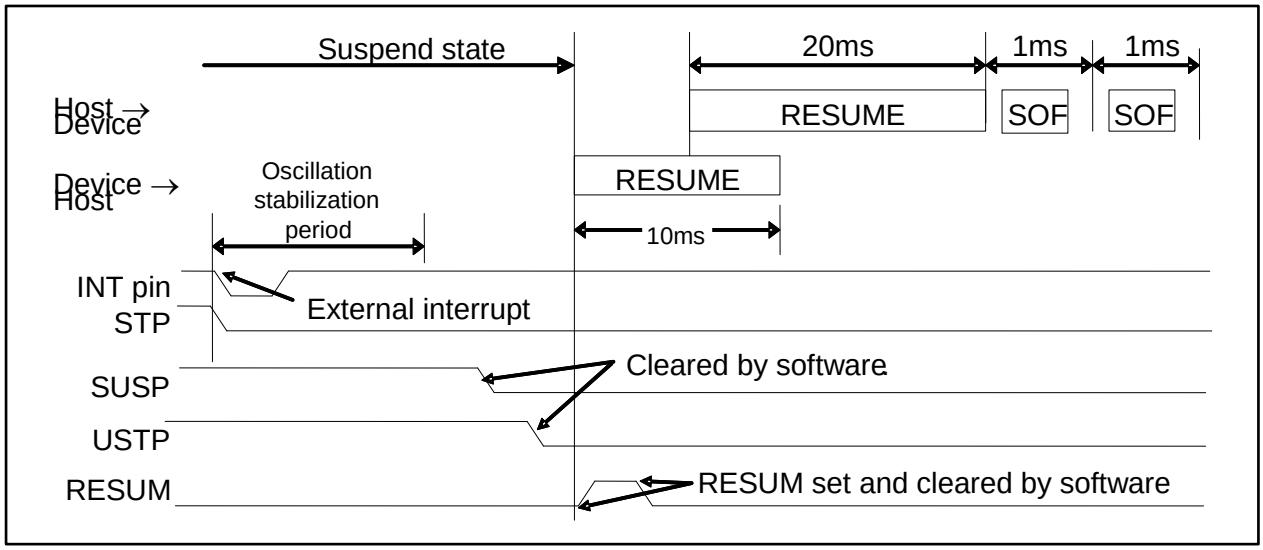
3.5 Wake-up function

To recover a USB device from suspend state to wake-up state, the USB protocol provides two ways.

- Remote wake-up from the device
- Wake-up from the host

Remote wake-up

Figure 3-9 Remote wake-up operation

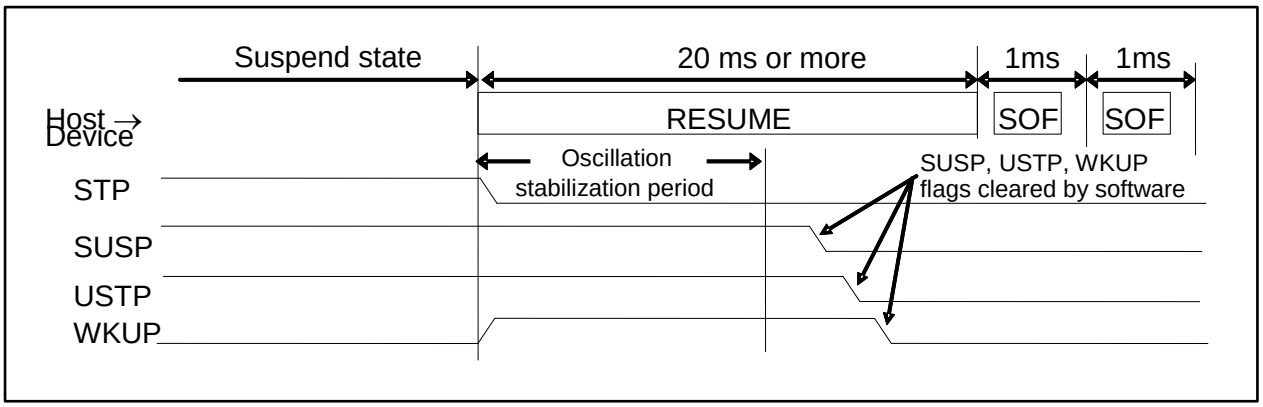


The device must be processed in the following sequence:

1. Recover the device from stop mode or timer mode by an external interrupt.
2. Check that the USB generation clock is stable.
3. Clear the SUSP bit in the UDCS register to "0".
4. Perform a dummy read from the UDCS register.
5. Clear the USTP bit of the UDCC register to "0".
6. Perform a dummy read from the UDCC register.
7. Set the RESUM bit in the UDCC register to "0".
8. Clear the RESUM bit in the UDCC register to "0".

Wake-up from the host

Figure 3-10 Wake-up operation from the host



Process the USB device in the following sequence.

1. Set the oscillation stabilization time so that it will not exceeds 10 ms.
2. Check that the USB clock is stable.
3. Clear SUSP bit in the UDCS register, and USTP bit in the UDCC register to "0" in this order.
4. Clear WKUP bit in the UDCS register to "0".

3.6 DMA transfer function

Data handled by the USB function can be transferred via DMA between the send/receive buffer and embedded RAM. The following two modes are available for the DMA transfer.

- Packet transfer mode, in which CPU starts DMA for each packet.
- Automatic data size transfer mode, in which DMA is automatically started for every packet.

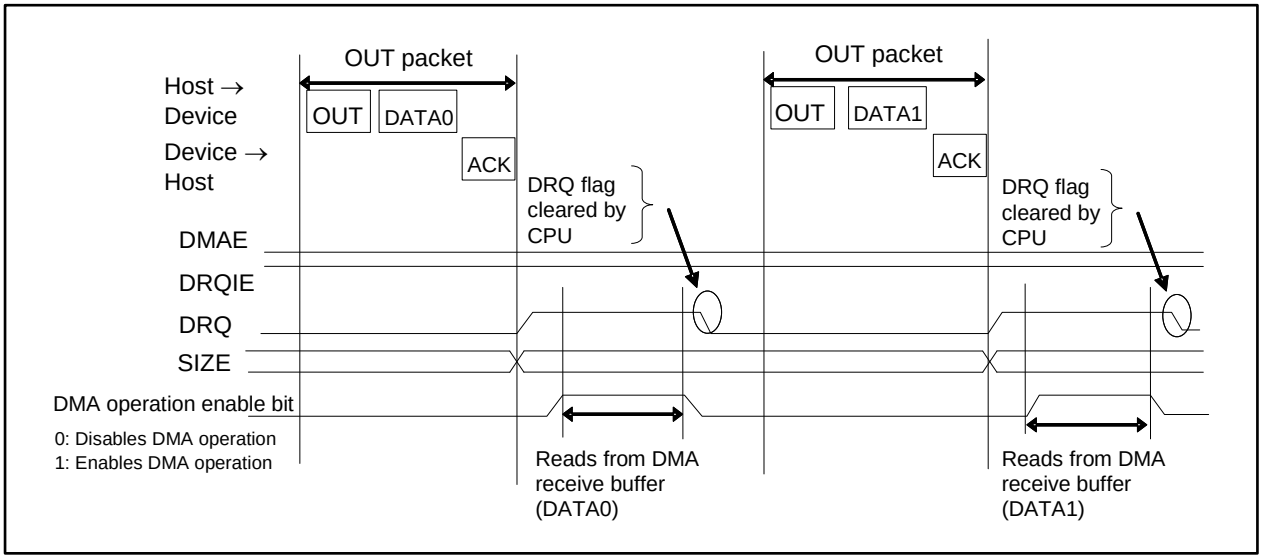
Packet transfer mode

The packet transfer mode transfers each packet according to the data size set in DMA and, each time the transfer of a packet finished, clears the interrupt factor (DRQ) for the next packet transfer. This transfer mode can access buffers of Endpoint 1 to Endpoint 5. Before using DMA, set the interrupt output destination by the DREQ Select Register. (Connect the interrupt output to CPU.NVIC.)

Figure 3-11 and Figure 3-12 show the timing to access buffers in each OUT direction and IN direction.

■ Transfer in the OUT direction (Host -> Device)

Figure 3-11 OUT packet transfer

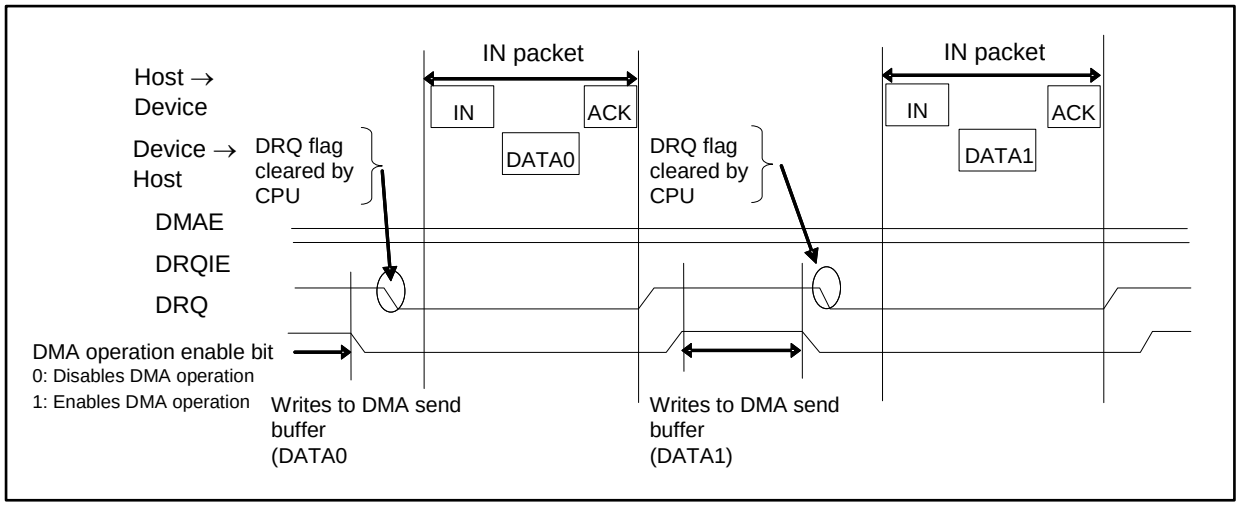


In the OUT direction transfer, the device must be processed in the following sequence:

1. Once the DRQ flag is set and the interrupt handling is entered, check the transfer data size.
2. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the transfer data size, and then enable DMA to start the transfer.
3. After the transfer, clear the pertinent DRQ flag in the EP1S to EP5S registers and the pertinent interrupt factor flag in the DMAC status register, and return from the interrupt handling.

■ Transfer in the IN direction (Device -> Host)

Figure 3-12 IN packet transfer



In the IN direction transfer, the device must be processed in the following sequence:

1. Once the DRQ flag is set and the interrupt handling is entered, configure the DMA register settings relevant to the number of transfers and block size corresponding to the data size to be transferred in the next IN packet, and then enable DMA to start the transfer.
2. After the DMA transfer, clear the pertinent DRQ flag in the EP1S to EP5S registers and the pertinent interrupt factor flag in the DMAC status register, and return from the interrupt handling.

Automatic data size transfer mode

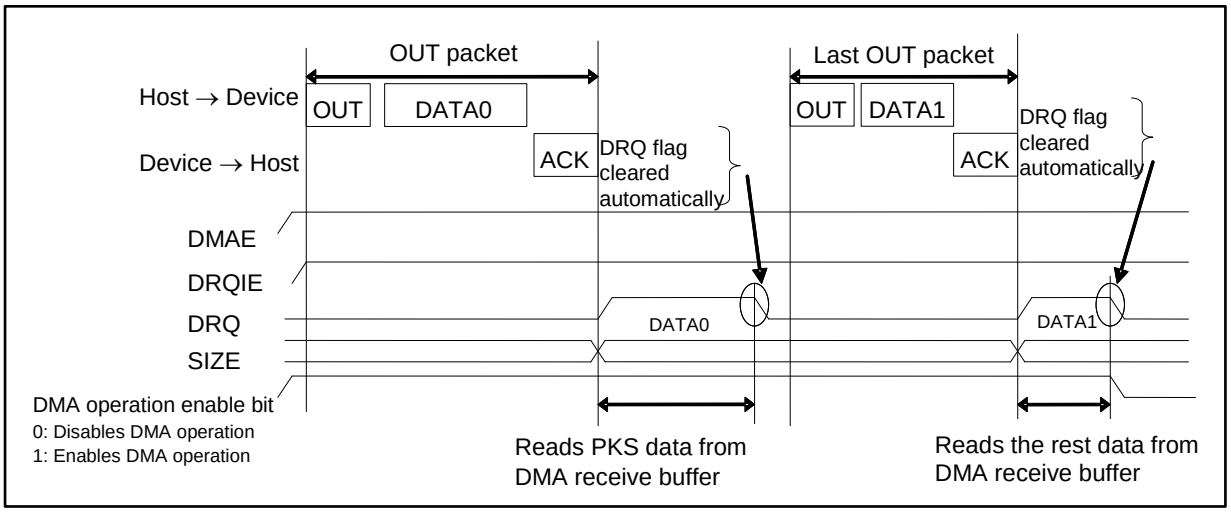
This mode can transfer even bytes. To transfer odd bytes in the OUT direction transfer, a CPU transfer sequence is required. (See Figure 3-14.) To transfer odd bytes in the IN direction transfer, see the following information.

- For TYPE0 products
  Odd bytes cannot be transferred in the IN direction transfer.
- For products other than TYPE0
  To transfer odd bytes in the IN direction transfer via DMA, set the ODDPKS register.(See chapter "Interrupts (A)".)

Before using DMA, set the interrupt output destination by the DREQ Select Register.(Connect the interrupt output to DMAC.) Configure in DMA the total data size to transfer, and also set the transfer enable bit previously. If DRQ is set after transfer from the host while DMAE is enabled, the interrupt factor (DRQ) is automatically cleared when the data size corresponding to PKS in the EP1 to EP5 Control Registers (EPxC) has been transferred. Afterward, the same sequence is repeated after transfer from the host until the transfer data size configured previously in DMA is reached. Meanwhile, configuration by the CPU is not required at all. Thus this mode can transfer data automatically by a single setting. The CPU interrupt is entered after the transfer of the last data. To perform the next transfer, therefore, reconfigure DMAC then to enable DMA and return from the interrupt. The automatic data size transfer mode uses DMAE as "1", buffer access to Endpoints 1 to 5 is only enabled. The following shows the timing to access the buffer in each of the OUT and IN directions.

■ Transfer in the OUT direction (Host -> Device)

Figure 3-13 Transfer in the OUT direction (Host -> Device)



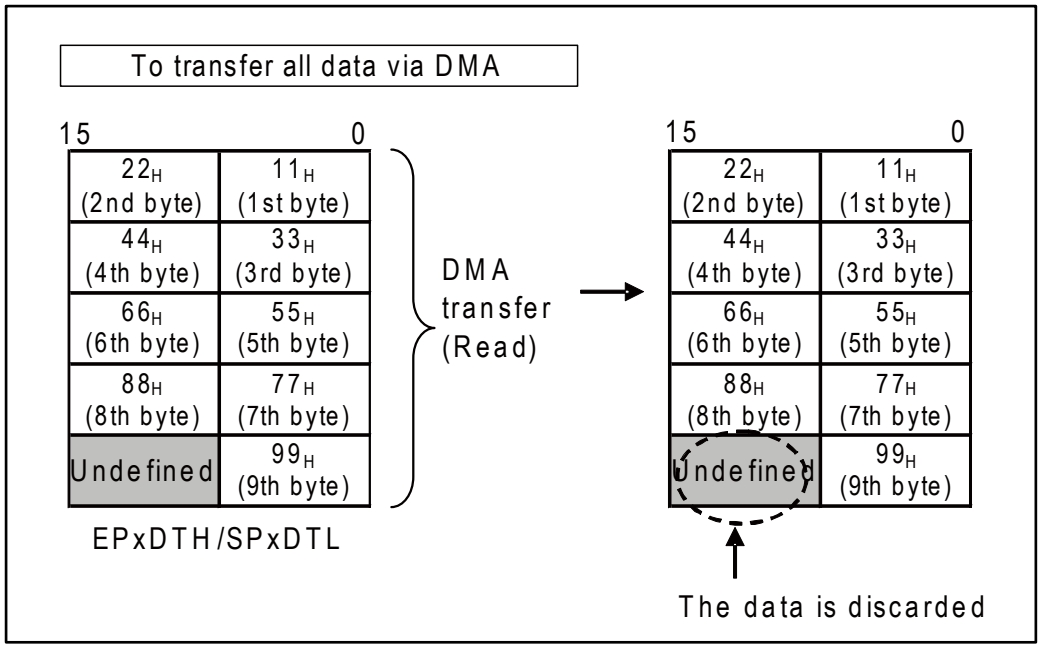
In the OUT direction transfer, the device must be processed in the following sequence:

1. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the total data size, and then enable DMA to start the transfer.
2. Enable DMAE and DRQIE.
3. After the transfer, reconfigure the DMAC using an interrupt generated by the interrupt factor pertinent to the DMAC status register, and clear the flag to return from the interrupt handling.

To transfer the data size corresponding to the odd bytes via DMA, the following methods are available:

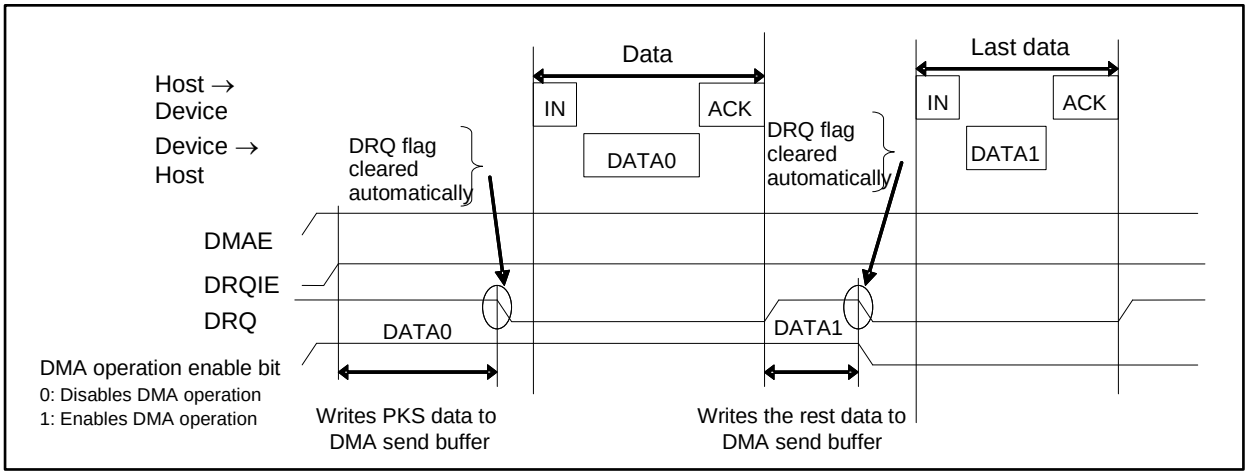
- Transfer all the data + 1 byte via DMA, and discard the last data after an endian conversion.

Figure 3-14 Example odd bytes transfer in the OUT direction



■ Transfer in the IN direction (Device -> Host)

Figure 3-15 Transfer in the IN direction (Device -> Host)



In the IN direction transfer, the device must be processed in the following sequence:

1. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the total data size, and then enable DMA to start the transfer.
2. Enable DMAE and DRQIE.
3. After the transfer, reconfigure the DMAC using an interrupt generated by the interrupt factor pertinent to the DMAC status register, and clear the flag to return from the interrupt handling.

3.7 NULL transfer function

If data sent from the USB function is the last packet and satisfies the maximum packet size, then the 0-byte can be automatically transferred via the next packet transfer. DMAE must be enabled to use this function. This function is valid only in IN transfer.

NULL transfer mode

NULL transfer mode sends 0-byte in reply to the next host's data request in the IN direction after the last data in the IN direction has been transferred.

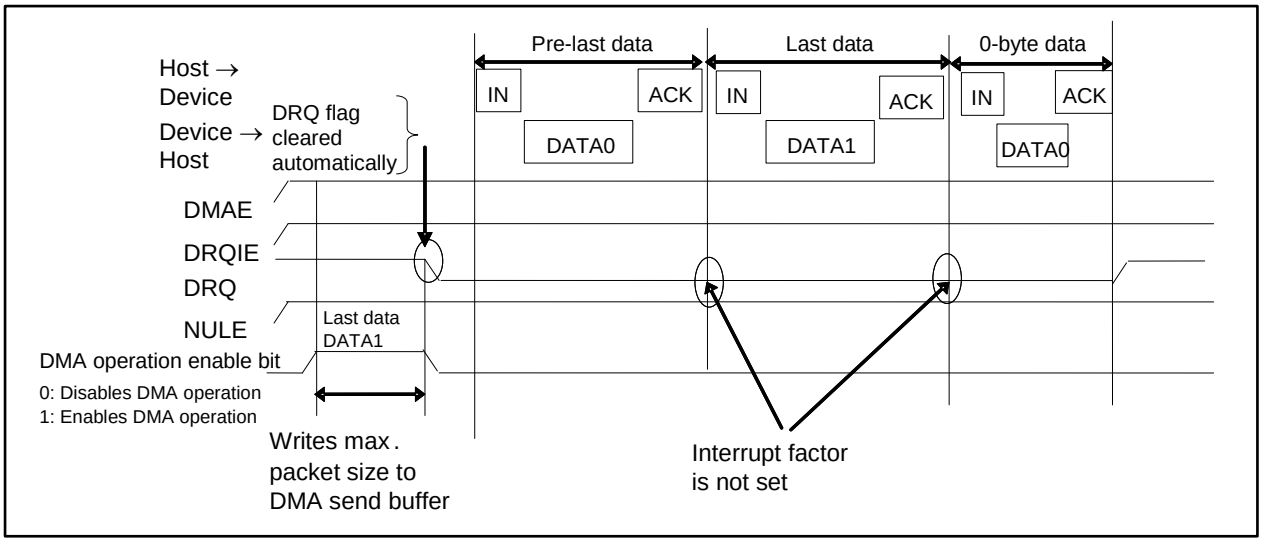
NULL transfer mode works when the following conditions are met:

- Automatic buffer transfer mode is set (DMAE = 1)
- The last data transfer writes the maximum packet size to the DMA buffer
- DMA data units are counted as 0 by writing the last data

After the last data has been written to buffer via DMA, the DRQ interrupt flag is not set until the 0-byte data is read from the host. The following shows the timing to access the buffer.

Only the transfer in the IN direction (Device -> Host) is explained.

Figure 3-16 NULL data transfer operation



The device must be processed as follows:

1. Enable EPxC:DMAE, EPxS:DRQIE, and EPxC:NULE bits by setting to "1".

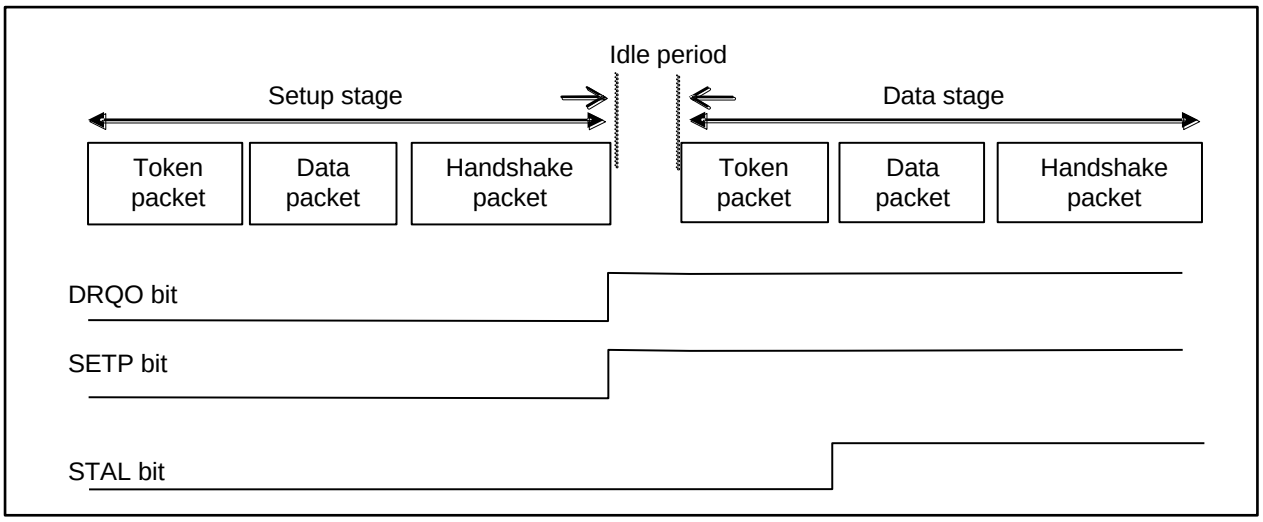
3.8 STALL response/release of Endpoint 0

The STAL bit in the EP0 Control Register (EP0C) controls the STALL response and release of Endpoint 0.

STAL bit set timing

To perform the STALL response, interpret the command at the setup stage (SETP = 1 detection) of control transfer. If the STALL response is required, set the STAL bit. (See Figure 3-17) After setting the STAL bit, clear the interrupt factor (DRQO bit).

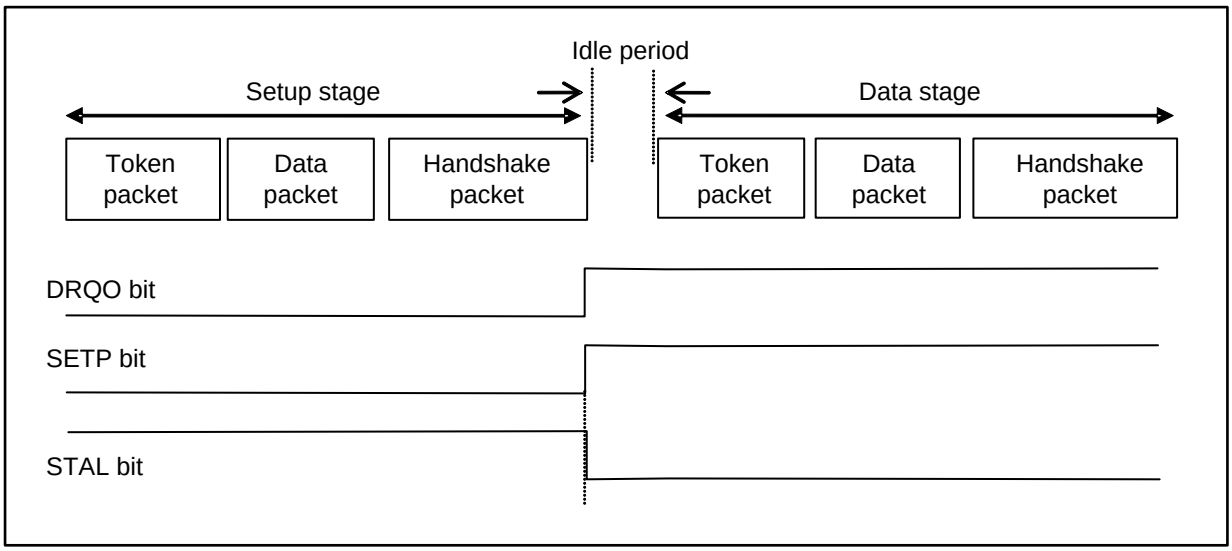
Figure 3-17 STAL bit set timing



STAL bit clear timing

Upon the detection of SETP = 1, pointing to the setup stage of control transfer, the STAL bit is automatically cleared and the STALL state is released. (See Figure 3-18)

Figure 3-18 STAL bit clear timing



Note:

- *Upon the detection of SETP = 1 (DRQO = 1 interrupt), the STAL bit is cleared to "0". To enable the STALL response again, set the STAL bit to "1".*

3.9 STALL response/release of Endpoint 1 to Endpoint 5

The STAL bit and the internal status bit in the EP1 to EP5 Control Registers (EP1C to EP5C) controls the STALL response and release of Endpoints 1 to 5.

STALL response processed by software

Figure 3-19 and Figure 3-20 show the procedures to process the STALL response by software. To perform the STALL response, configure the STAL bit of relevant Endpoint by software. The internal status bit does not change then.

When a transaction occurs from the host to the Endpoint to which the STAL bit is set, the hardware automatically sets the internal status bit of the relevant Endpoint to perform the STALL response to the host. Once the internal status bit is set, it remains set even when the STAL bit cleared. As the internal state bit remains set until the host issues the Clear Feature command, the STALL response remains running. While the STALCLREN bit of the UDC Control Register (UDCC) is set to "0", the STALL response also remains running in the following condition:

The STAL bit remains set even after the internal status bit is cleared by the Clear Feature command.

This is because the internal status bit is set each time a transaction occurs to the relevant Endpoint. To release the STALL response, therefore, the STAL bit must be cleared, and the internal status bit must be cleared by the Clear Feature command. If the STALCLREN bit in the UDC Control Register (UDCC) is set to "1", the STAL bit is cleared at the same time the internal status bit is cleared by the Clear Feature command, and the STALL response is not performed for the next transaction.

Figure 3-19 To process the STALL response by software (the STAL bit is cleared by software) UDCC.STALCLREN=0

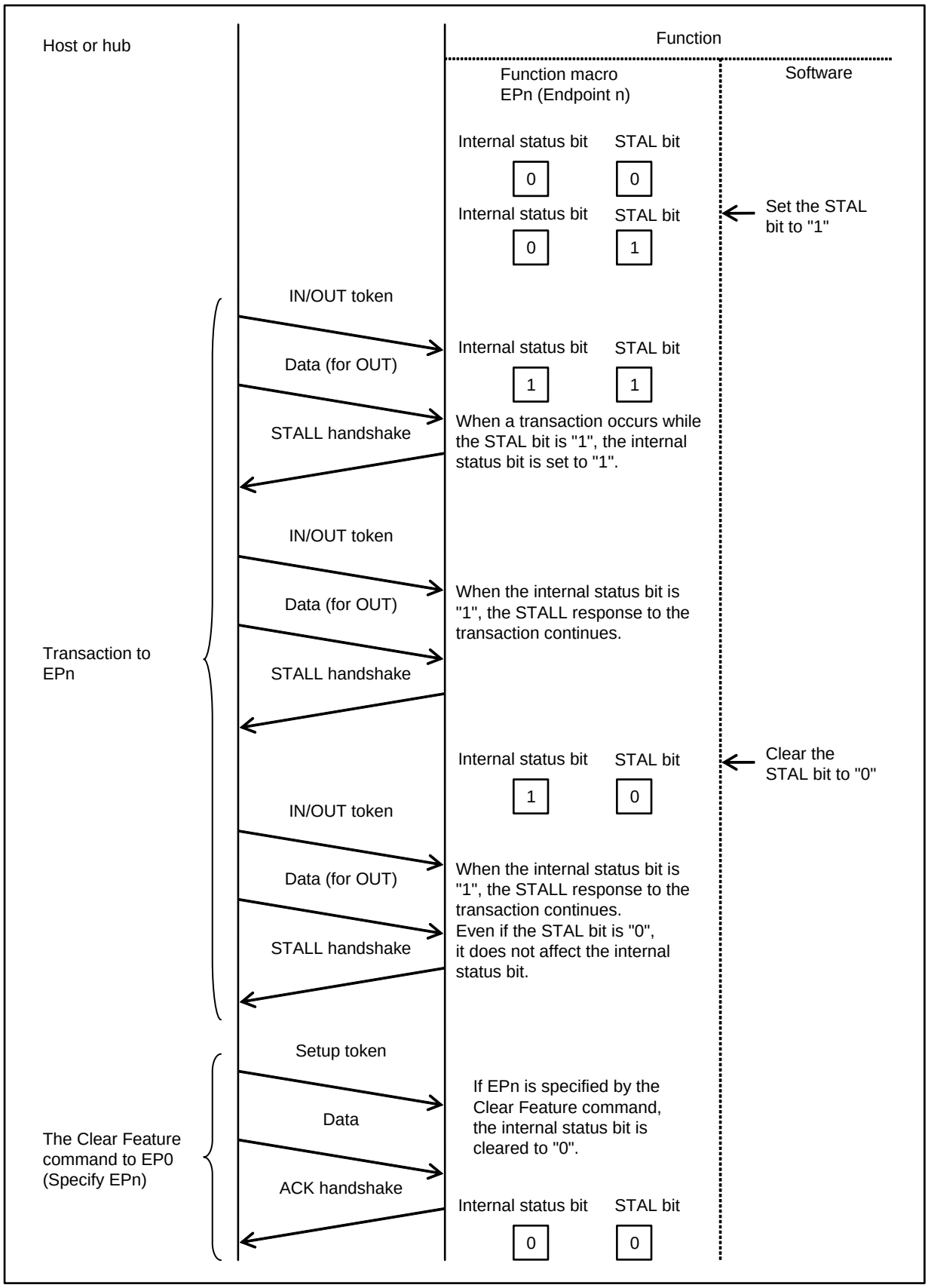
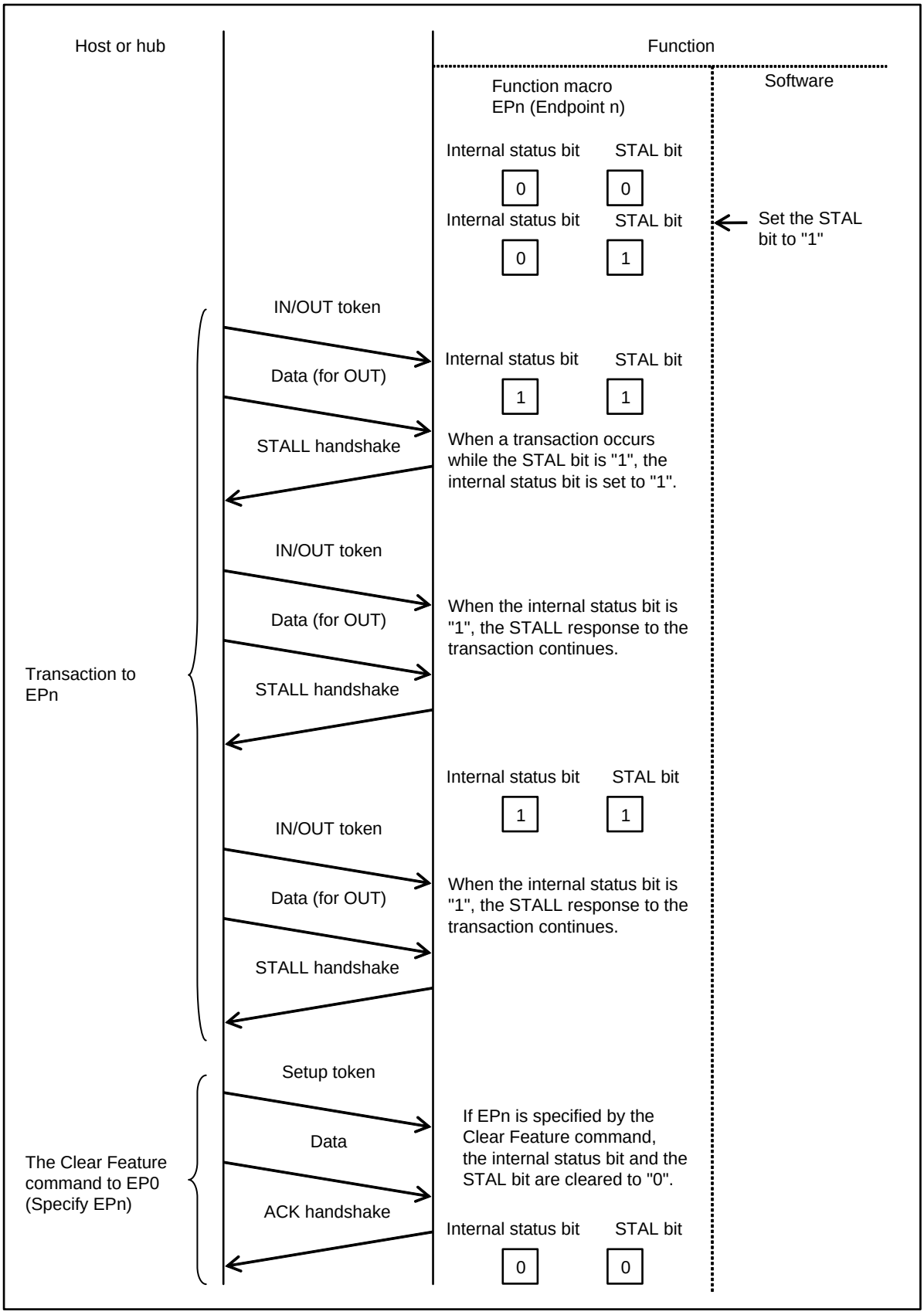


Figure 3-20 To process the STALL response by software (the STAL bit is cleared by hardware) UDCC.STALCLREN=1



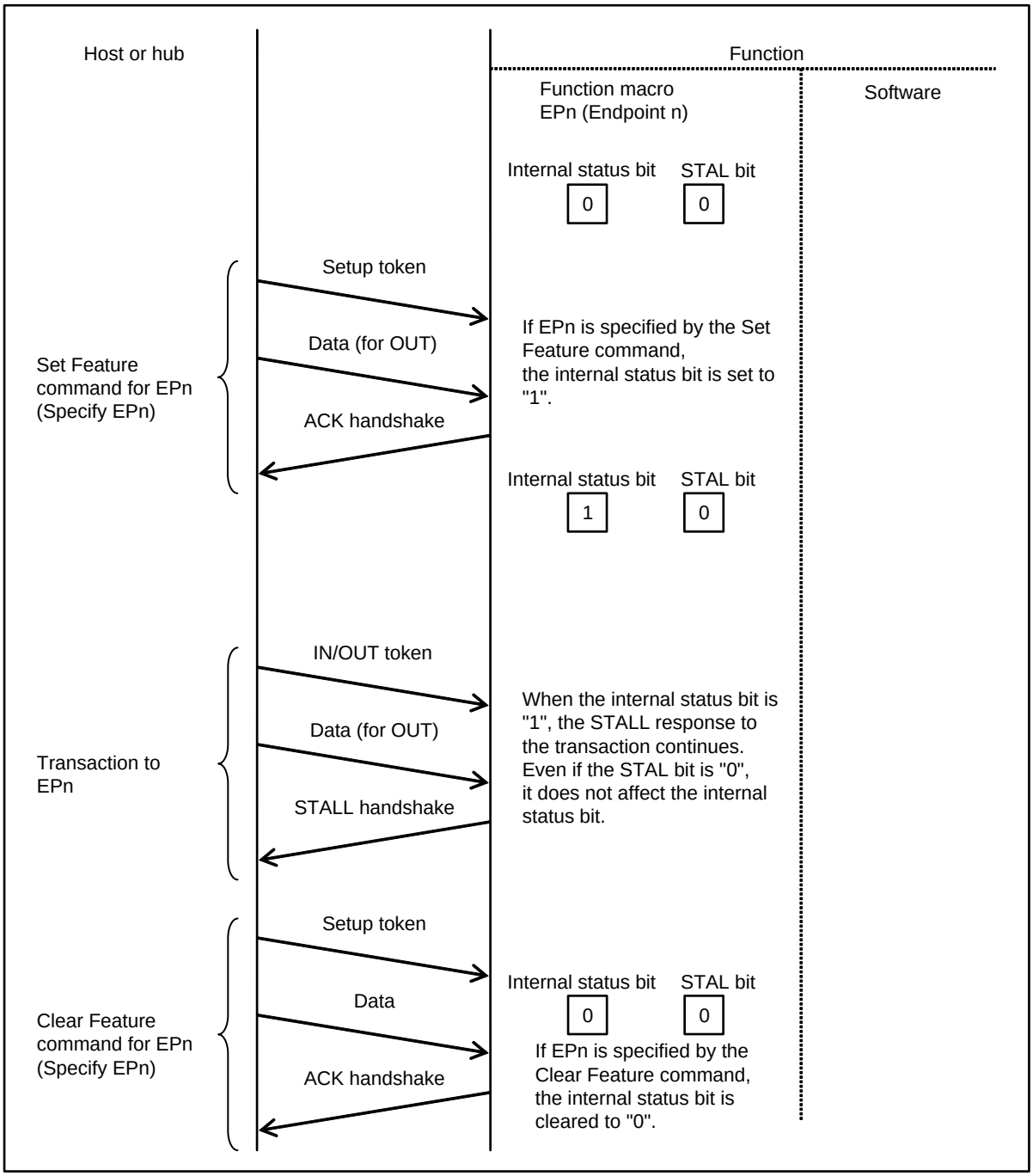
Automatic STALL response by hardware

Figure 3-21 shows the procedure for the automatic STALL response by hardware.

When the STALL response is set by the Set Feature command, the hardware automatically set the internal status bit of the relevant Endpoint, irrespective of the STAL bit setting, and perform the STALL response. Once the internal bit is set, the value is retained until cleared by the Clear Feature command from the host irrespective of the STAL bit setting.

The STAL bit is referred to even after the internal status bit is cleared by the Clear Feature command. To release the STALL response, therefore, the internal status bit must be cleared by the Clear Feature command.

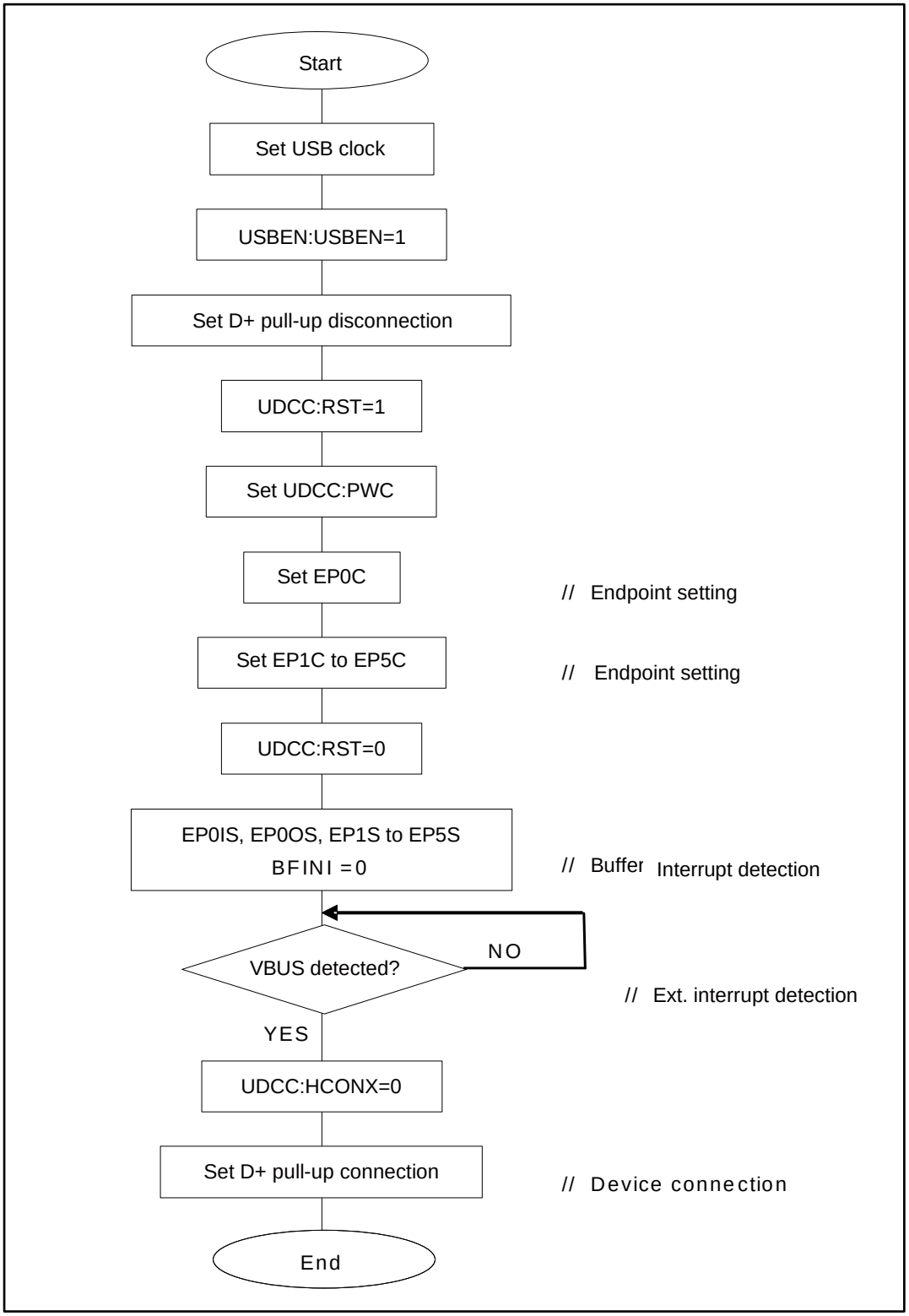
Figure 3-21 Automatic STALL Response by Hardware



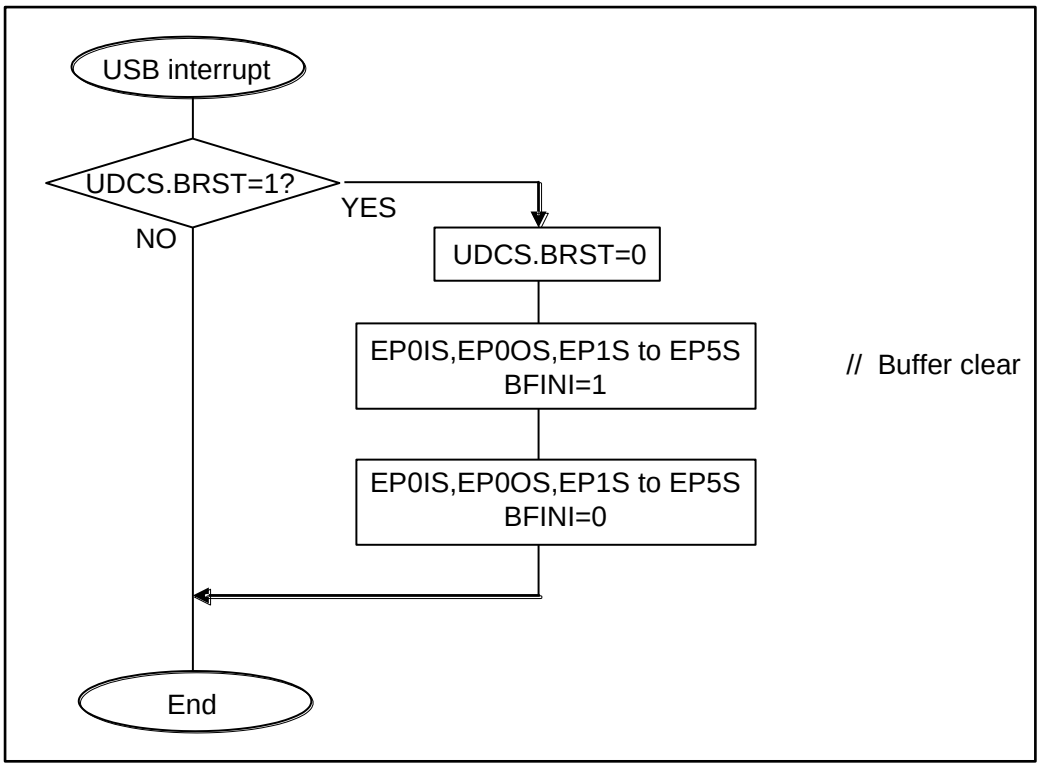
4. Examples of USB Function Setting Procedures

This section provides flowcharts for initialization, bus reset, CPU transfer, packet transfer (IN/OUT) and automatic data size transfer (IN/OUT).

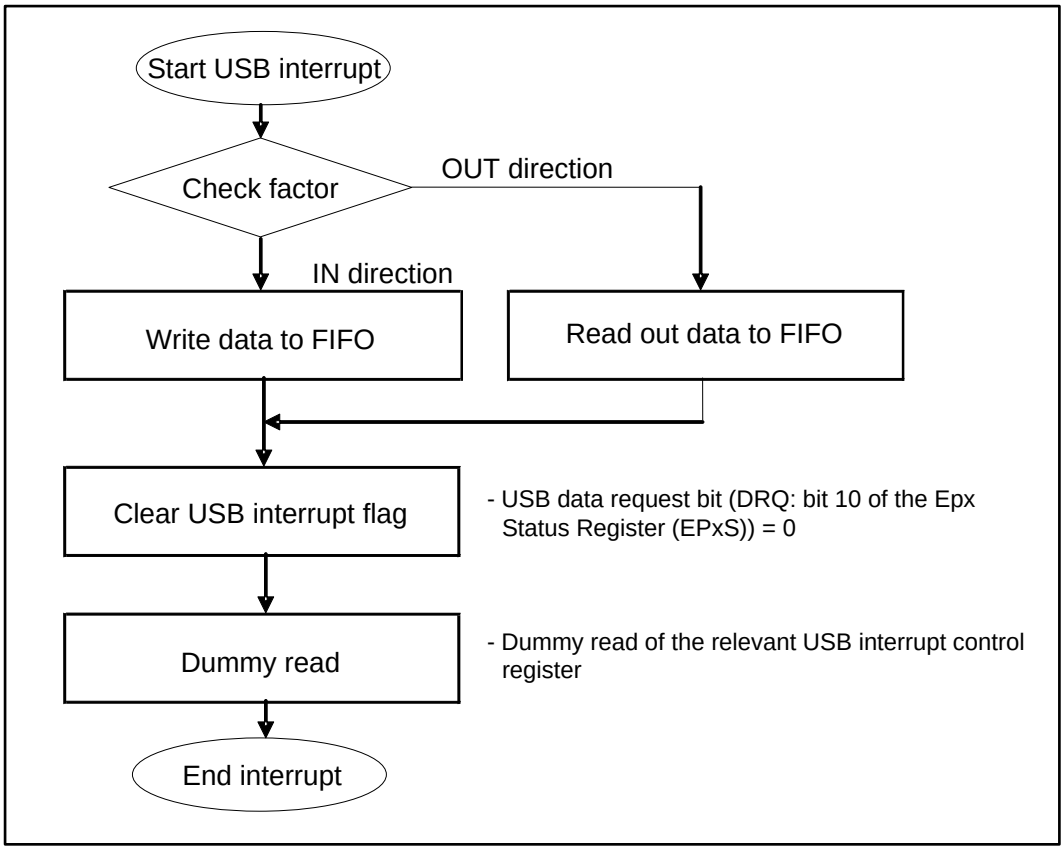
Initialization



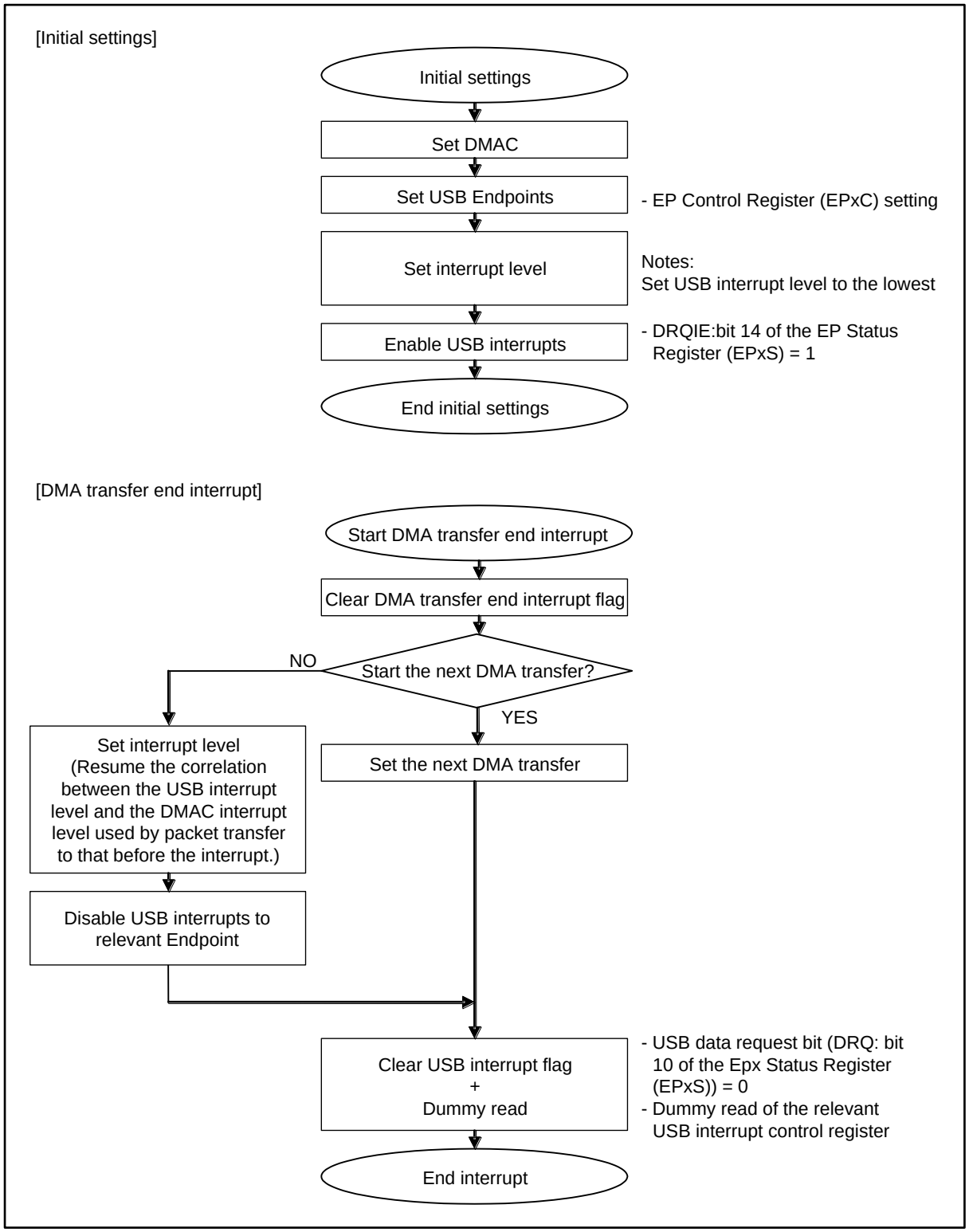
Bus reset


USB interrupt
UDCS.BRST=1?
YES
NO
UDCS.BRST=0
EP0IS,EP0OS,EP1S to EP5S
BFINI=1
// Buffer clear
EP0IS,EP0OS,EP1S to EP5S
BFINI=0
End

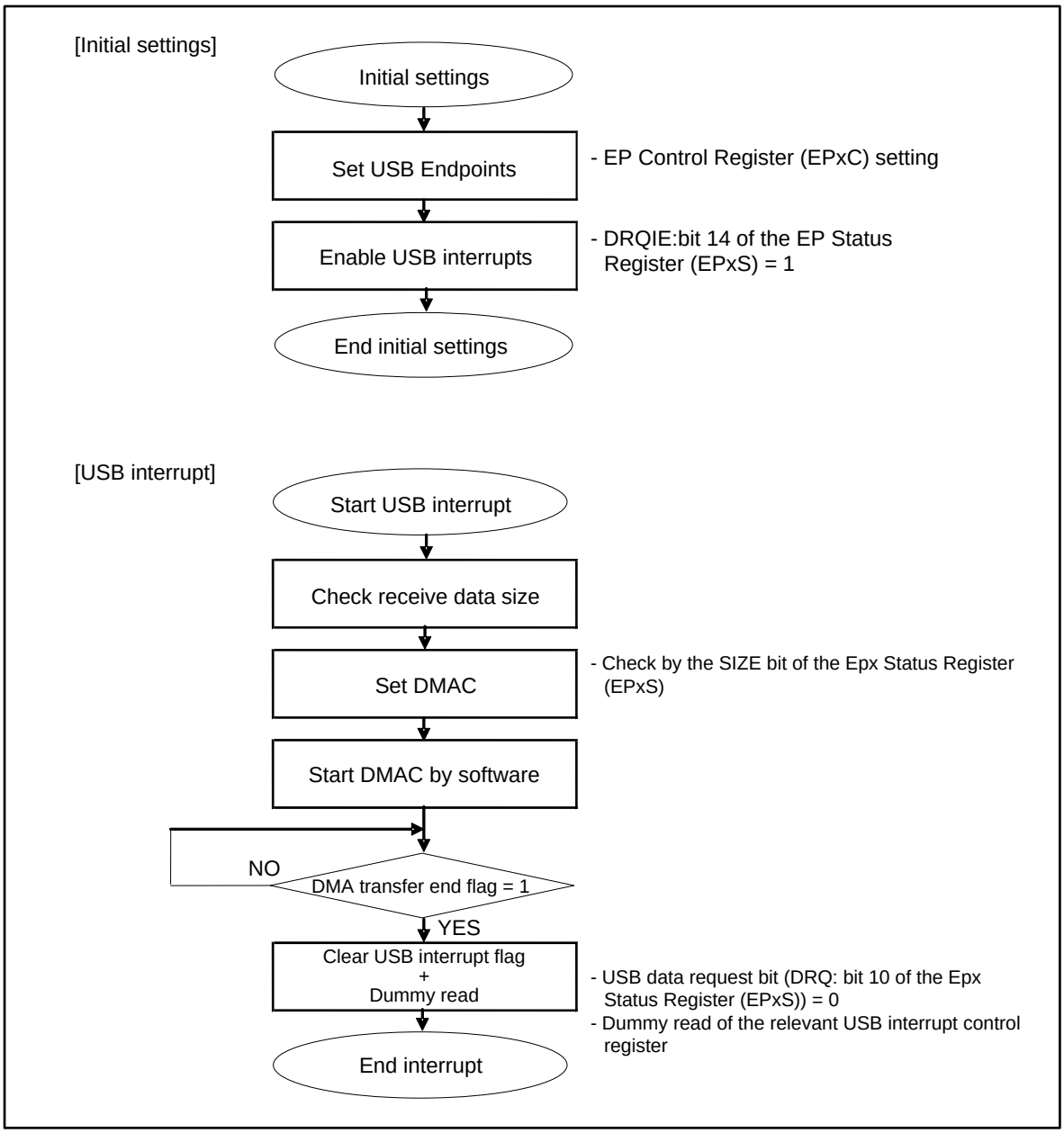
Example control for CPU transfer



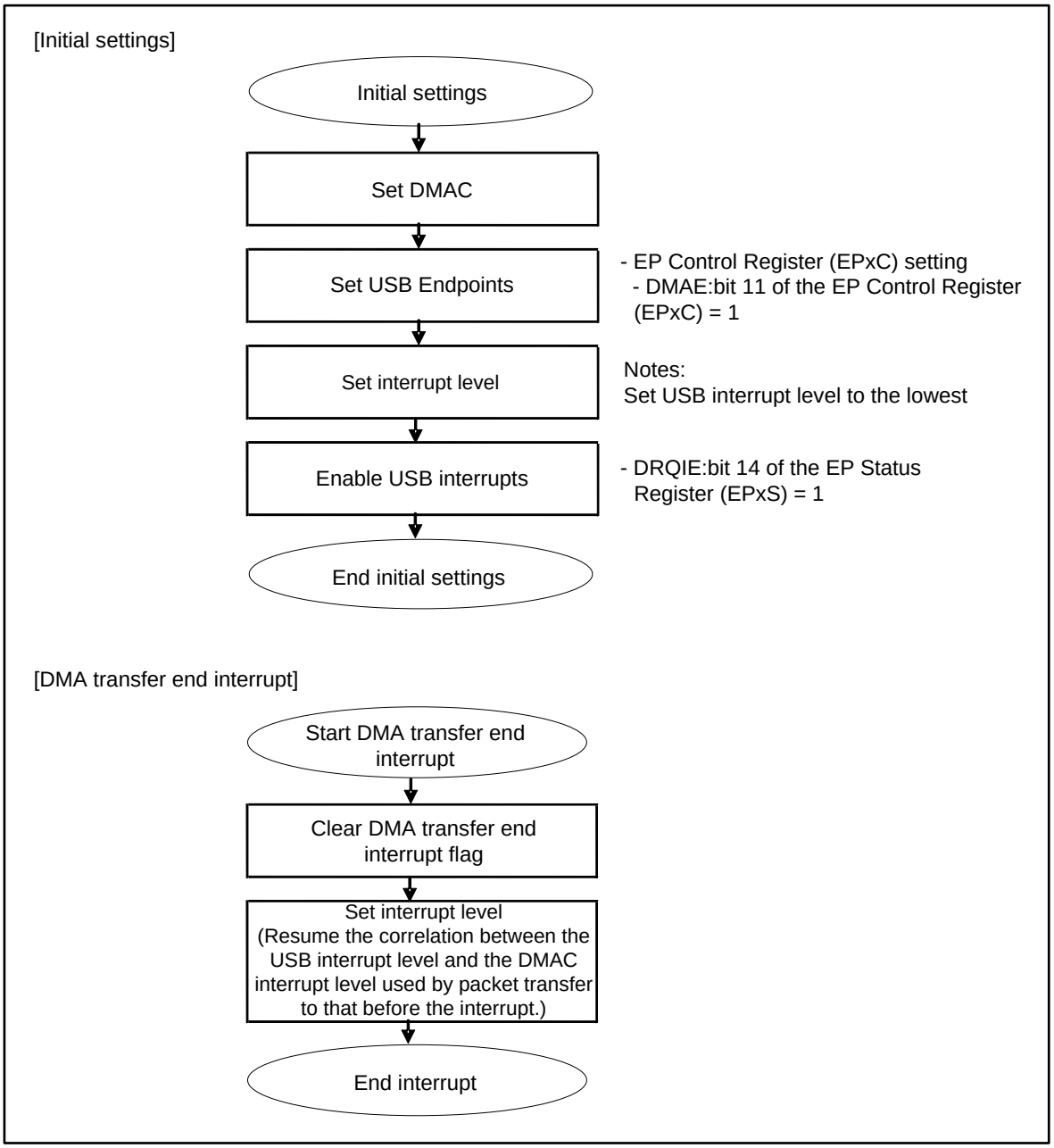
Example control for packet transfer in IN direction



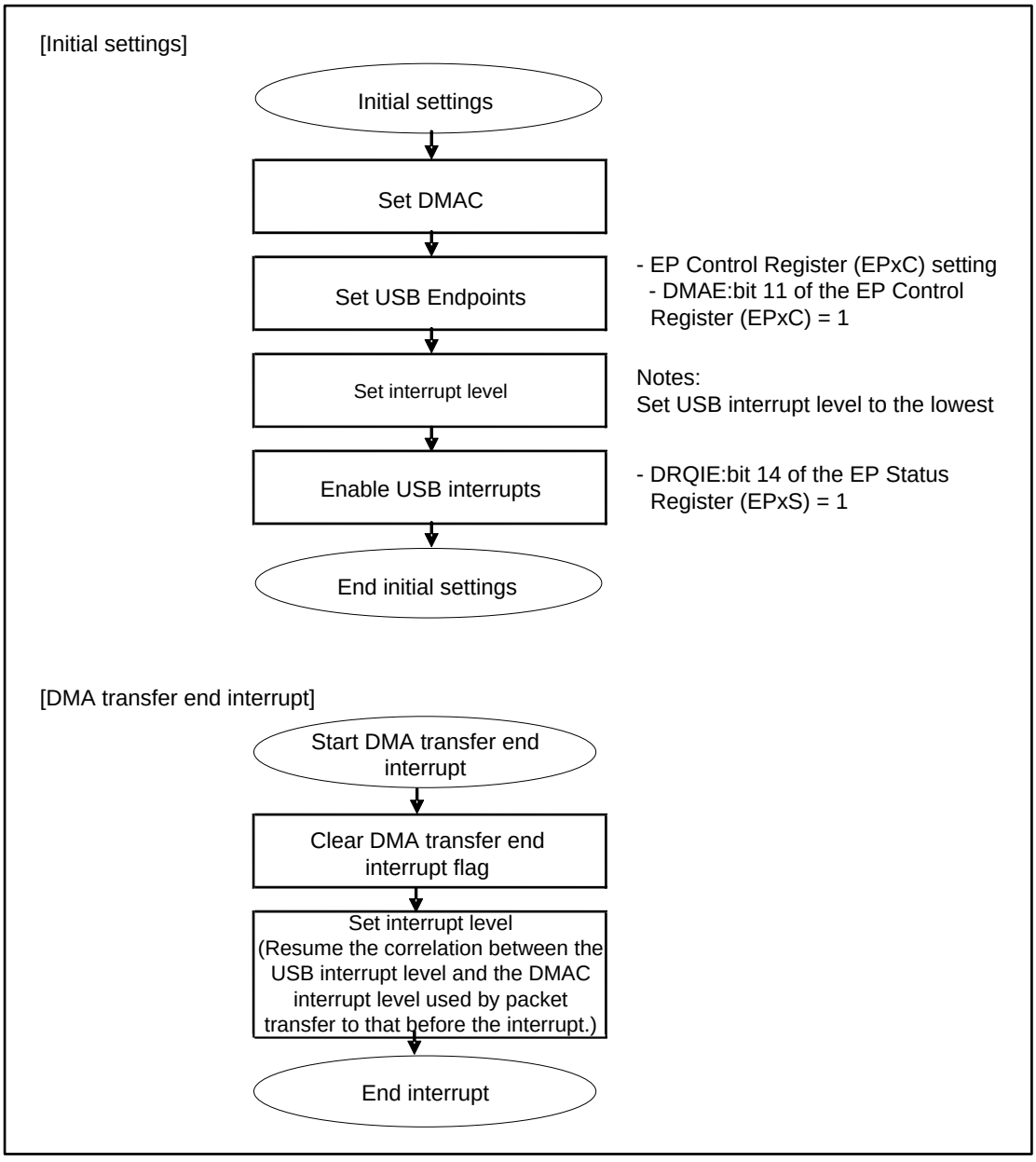
Example control for packet transfer in OUT direction



Example control for automatic data size transfer in IN direction



Example control for automatic data size transfer in OUT direction



5. USB Function Registers

This section explains the configurations and functions of the registers used for the USB function.

USB function register list

Abbreviation	Register name	Reference
UDCC	UDC Control Register	5.1
EP0C	EP0 Control Register	5.2
EP1C	EP1 Control Register	5.3
EP2C	EP2 Control Register	
EP3C	EP3 Control Register	
EP4C	EP4 Control Register	
EP5C	EP5 Control Register	
TMSP	Time Stamp Register	5.4
UDCS	UDC Status Register	5.5
UDCIE	UDC Interrupt Enable Register	5.6
EP0IS	EP0I Status Register	5.7
EP0OS	EP0O Status Register	5.8
EP1S	EP1 Status Register	5.9
EP2S	EP2 Status Register	
EP3S	EP3 Status Register	
EP4S	EP4 Status Register	
EP5S	EP5 Status Register	
EP0DTH	EP0 Data Register high-order	5.10
EP0DTL	EP0 Data Register low-order	
EP1DTH	EP0 Data Register high-order	
EP1DTL	EP0 Data Register low-order	
EP2DTH	EP0 Data Register high-order	
EP2DTL	EP0 Data Register low-order	
EP3DTH	EP0 Data Register high-order	
EP3DTL	EP0 Data Register low-order	
EP4DTH	EP0 Data Register high-order	
EP4DTL	EP0 Data Register low-order	
EP5DTH	EP0 Data Register high-order	
EP5DTL	EP0 Data Register low-order	

UDCC:RST dependent register bit update timing list

	Register	bit
Register bits to be updated when UDCC:RST=1	UDCC	HCONTX, PFBK, PWC
	EP0C	PKS0
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
	EP3C	EPEN, TYPE, DIR, PKS3
	EP4C	EPEN, TYPE, DIR, PKS4
	EP5C	EPEN, TYPE, DIR, PKS5
Register bits initialized when UDCC:RST=1	EP0IS	BFINI, DRQI
	EP0OS	BFINI, DRQ, SPK
(Update when UDCC:RST=0)	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
	EP3S	BFINI, DRQ, SPK
	EP4S	BFINI, DRQ, SPK
	EP5S	BFINI, DRQ, SPK
	TMSP	TMSP
	UDCS	SUSP, SOF, BRST, WKUP, SETP, CONF
	UDCIE	SUSPIE, SOFIE, BRSTIE, WKUPIE, CONFN, CONFIE
Register bits unaffected by UDCC:RST	UDCC	RESUME, USTP
	EP0C	STAL
	EP1C	DMAE, NULE, STAL
	EP2C	DMAE, NULE, STAL
	EP3C	DMAE, NULE, STAL
	EP4C	DMAE, NULE, STAL
	EP5C	DMAE, NULE, STAL
	EP1DTH/L	BFDT
	EP2DTH/L	BFDT
	EP3DTH/L	BFDT
	EP4DTH/L	BFDT
	EP5DTH/L	BFDT

5.1 UDC Control Register (UDCC)

The UDC Control Register (UDCC) controls the UDC core circuit.

The following figure shows the bit configuration of the UDC Control Register (UDCC).

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	RST	RESUM	HCONX	USTP	STALCLREN	Reserved	RFBK	PWC
Attribute	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
Initial value	1	0	1	0	0	0	0	0

Note:

- *The UDC Control Register (UDCC), except bit6 RESUM and bit4 USTP, should be configured while bit7 RST = 1, and should not be rewritten while USB is running. bit6 RESUM must be set or reset in USB suspend mode and while the remote wake-up is enabled by the following command.*
  Set bit4 USTP to "1" before stop mode or timer mode is entered.
  When those modes have been released, set the SUSP of UDCS and USTP of UDCC to "0" in this order after confirmation of stabilized USB supply clock.

The following explains the function of each bit in the UDC Control Register (UDCC).

[bit15:8] Reserved: Reserved bits

Always write "0" to these bits. They are always read as "0".

[bit7] RST: Function Reset bit (function ReSeT)

This bit is ORed with the chip system reset to individually reset the USB function. The USB function is reset by the RST bit when connected with the host via cable. As the initial value is "1", reset enabled, write "0" to release the state.

bit	Description
0	Releases USB Function reset
1	Resets the USB function

Note:

- *This bit initializes the relevant bit of the Time Stamp Register (TMSP), UDC Status Register (UDCS), UDC Interrupt Enable Register (UDCIE) at the same time. It also sets the BFINI of the EP0I, EP0O, and EP1 to EP5 Status Register concurrently. After the initial settings, therefore, clear the RST bit (BFINI bit is not cleared) and clear BFINI bit of the Endpoints used in this order.*

[bit6] RESUM: Resume Setting bit (RESUMe set)

In suspend state while remote wake-up is enabled *, the resume is started when writing "1" to the RESUM bit. To instruct to resume, set the RESUM bit to "1", and then write "0" to it to clear.

* : The DEVICE_REMOTE_WAKEUP bit is set by the SET_FEATURE command from the host.

bit	Description
0	Resets the USB resume start instruction bit
1	Instructs to start the USB resume

[bit5] HCONX: Host Connection bit (Host CONnection)

This bit controls the switch between an external pull-up resistor and the USB data line to make the connection with the host or HUB recognized.

bit	Description
0	Connected to the host or HUB
1	Disconnected from the host or HUB

Note:

- *Even if the connection is found by the host or HUB while the external pull-up resistor is kept ON, the bus reset command on the USB bus is ignored while this bit is "1".*

[bit4] USTP: USB Operating Clock Stop bit (Udc SToP)

Setting this bit stops the clock for the USB operating unit. When USB is not operated, power consumption can be reduced by configuring this bit.

bit	Description
0	Normal mode
1	Stops the clock for the USB operating unit

Note:

- *If stop mode and timer mode is not set, the USTP bit must be configured after setting RST to "1", and also after 3 cycles at full speed or 43 cycles at low speed (supported only in host mode) so that the reset can be ensured. This bit can be cleared at the same time RST is cleared.*

[bit3] STALCLREN: Endpoint 1 to Endpoint 5 STAL bit Clear Select bit (STALl CLeaR Enable)

This bit selects the method to clear the STAL bit of Endpoint 1 to Endpoint 5 using the Clear Feature command. The STALCLREN bit sets whether to automatically clear the STAL bit to "0" by hardware, a bit of EP1 to EP5 Control Registers (EP1C to EP5C) for Endpoints (1 to 5) specified by the Clear Feature command. This bit selects the method to clear the STAL bit of the Endpoint Control Registers (EP1C to EP5C), either by software or hardware.

bit	Description
0	Clears the STAL bit of the EP1 to EP5 Control Registers (EP1C to EP5C) by software.
1	Automatically clears the STAL bit of the EP1 to EP5 Control Registers (EP1C to EP5C) by hardware.

Note:

- *The STALCLREN bit should be configured while the RST of the UDC Control Register (UDCC) is "1", and should not be rewritten while USB is running.*

[bit2] Reserved: Reserved bit

Always write "0" to this bit. It is always read as "0".

[bit1] RFBK: Data Toggle Mode Select bit (Rate Feed BacK mode)

This bit selects the data toggle mode for USB interrupt transfer.

bit	Description
0	Selects the alternating data toggle mode. Toggles data PID when the transfer has finished successfully.
1	Selects the data toggle mode. Unconditionally toggles data PID.

[bit0] PWC: Power Control bit (PoWer Control)

This bit specifies the operating power mode (self power or bus power) of the USB function.
(Configuration of this bit applies to standard command GetStatus.)

bit	Description
0	Bus power
1	Self power

5.2 EP0 Control Register (EP0C)

The EP0 Control Register (EP0C) controls Endpoint 0.

The following figure shows the bit configuration of the EP0 Control Register (EP0C).

bit	15	14	13	12	11	10	9	8
Field	-				Reserved		STAL	Reserved
Attribute	-				-		R/W	-
Initial value	XXXX				00		0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	PKS0						
Attribute	-	R/W						
Initial value	0	1000000						

Note:

- *Except bit9 STAL, the EP0 Control Register (EP0C) must be configured while both of the bit7 RST in the UDC Control Register (UDCC) and bit7 BFINI in the EP0I/O Status Register (EP0I/EP0OS) are "1".*
  It must not be rewritten while USB is running.

The following explains the function of each bit in the EP0 Control Register (EP0C).

[bit15:12] -: Undefined bits

The written value has no effect. The read value is undefined.

[bit11:10] Reserved: Reserved bits

Always write "0" to these bits.
They are always read as "0".

[bit9] STAL: Endpoint 0 STALL Setting bit (STALl ep0 set)

This bit can set Endpoint 0 to the STALL state (STALL response).

This bit is automatically cleared by hardware. If a SETUP packet is received by Endpoint 0 after the STALL response to Endpoint 0 is performed, this bit is cleared to "0". For the timing to clear this bit, see "n STAL bit clear timing" of "3.8 STALL response/release of Endpoint 0".

bit	Description
0	Ignored
1	Sets the STALL state (STALL response)

Notes:

- *If the STALCLREN bit of UDC Control Register (UDCC) is "0", the STALL response remains operating to the host while the STAL bit is set to "1". Upon the receipt of a normal SETUP packet after STAL bit reset, Endpoint 0 resumes from the STALL state.*
- *A read-modify-write instruction reads this bit as "0".*

[bit8:7] Reserved: Reserved bits

Write value should always be "0".
They are always read as "0".

[bit6:0] PKS0: Packet Size Endpoint 0 Setting bits (PacKet Size ep0 set)

These bits specify the maximum number of bytes transferred by one packet. For Endpoint 0, the maximum number of bytes is 64, and the set value is valid both for IN and OUT directions.

Example: "0x08" => 8 bytes, "0x40" => 64 bytes (maximum value)

Notes:

- *These bits must be configured when both of the RST bit in the UDC Control Register (UDCC) and the BFINI bit in the EP0I/O Status Register (EP0IS/EP0OS) are "1". Do not rewrite while USB is running.*
- *A value exceeding the maximum number of transferable bytes (0x40), and "0x00" must not be written.*

5.3 EP1 to EP5 Control Registers (EP1C to EP5C)

The EP1 to EP5 Control Registers (EP1C to EP5C) control Endpoint 1 to Endpoint 5.

The following figure shows the bit configuration of the EP1 to EP5 Control Registers (EP1C to EP5C).

EP1 Control Register (EP1C)

bit	15	14	13	12	11	10	9	8
Field	EPEN	TYPE		DIR	DMAE	NULE	STAL	PSK1
Attribute	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Initial value	0	11		0	0	0	0	1

bit	7	6	5	4	3	2	1	0
Field	PSK1							
Attribute	R/W							
Initial value	0x00							

EP2 to EP5 Control Registers (EP2C to EP5C)

bit	15	14	13	12	11	10	9	8
Field	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved
Attribute	R/W	R/W		R/W	R/W	R/W	R/W	-
Initial value	0	11		0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	PKS5 to PKS2						
Attribute	-	R/W						
Initial value	0	1000000						

Note:

- *Except DMAE, NULE, and STAL bits, the EP1 to EP5 Control Registers (EP1C to EP5C) must be configured while both of the bit 7 RST in the UDC Control Register (UDCC) and bit 15 BFINI in the EP0 to EP5 Status Registers (EP1S to EP5S) are "1". They must not be rewritten while USB is running.*

The following explains the function of each bit in the EP1 to EP5 Control Registers (EP1C to EP5C).

[bit15] EPEN: Endpoint 1 to Endpoint 5 Enable bits (EndPoint1 to EndPoint5 ENable)

This bit enables the Endpoint. Based on the EPEN bit setting, the Endpoint is configured by the host as those used by the function. TYPE, DIR and PKS bits in the EP1 to EP5 Control Registers are valid as the configuration information.

bit	Description
0	Disables the Endpoint
1	Enables the Endpoint

[bit14:13] TYPE: Endpoint Transfer Type Select bits (Endpoint TYPE)

These bits specify the transfer type that the Endpoint supports.

bit14:13	Description
00	Setting is prohibited.
01	Iso transfer (Function operating mode)
10	Bulk transfer
11	Interrupt transfer

Note:

- *Iso transfer can be set in function operating mode for Endpoint 1 only or for both Endpoint 1 and Endpoint 2. Setting for Endpoint 2 only, setting for other than Endpoint 1/ Endpoint 2 or setting in host operating mode is disabled.*

[bit12] DIR: Endpoint Transfer Direction Select bit (endpoint DIRection)

This bit specifies the transfer direction that the Endpoint supports.

bit	Function operating mode	Host operating mode (EP1 and EP2 only)
0	OUT Endpoint	IN Endpoint
1	IN Endpoint	OUT Endpoint

[bit11] DMAE: DMA Automatic Transfer Enable bit (DMA Enable)

This bit sets a mode that uses DMA for writing or reading transfer data to/from send/receive buffer, and automatically transfers the send/receive data synchronized with an data request in the IN or OUT direction by the host. Until the data size set in the DMA is reached, the data is transferred.

bit	Description
0	Releases the automatic buffer transfer mode
1	Sets the automatic buffer transfer mode

Note:

- *The CPU must not access the send/receive buffer while the DMAE bit is set to "1".*

[bit10] NULE: NULL Automatic Transfer Enable bit (NULl Enable set)

When a data transfer request in IN direction is received while automatic buffer transfer mode is set (DMAE = 1), this bit sets a mode that transfers 0-byte data automatically upon the detection of the last packet transfer.

bit	Description
0	Releases the NULL automatic transfer mode
1	Sets the NULL automatic transfer mode

Note:

- *For data transfer in the OUT direction or when automatic buffer transfer mode is not set, the NULL bit configuration does not affect communication.*

[bit9] STAL: Endpoint 1 to Endpoint 5 Stall Setting bit (STALl set)

This bit can set Endpoint to the STALL state (STALL response).

- When the STALCLREN bit of the UDC Control Register (UDCC) is "0"

  This bit is not cleared to "0" by the Clear Feature command. This bit must be cleared by software. For the timing to clear this bit, see "n STALL response processed by software" of "3.9 STALL response/release of Endpoint 1 to Endpoint 5".

bit	Description
0	Release the STALL state
1	Sets the STALL state (STALL response)

- When the STALCLREN bit of the UDC Control Register (UDCC) is "1"

  This bit is cleared by hardware. It is cleared to "0" for the Endpoint specified by the Clear Feature command. For the timing to clear this bit, see "n STALL response processed by software" of "3.9 STALL response/release of Endpoint 1 to Endpoint 5".

bit	Description
0	Ignored
1	Sets the STALL state (STALL response)

Notes:

- *If the STALCLREN bit of the UDC Control Register (UDCC) is "0", the STALL response remains operating to the host while the STAL bit is set to "1". Return from the STALL state is possible by the Clear Feature command after resetting the STAL bit.*
- *The value read by a read-modify-write instruction differs depending on the value set in STALCLREN.*
- *When STALCLREN = 0, the value at that time is read.*
- *When STALCLREN = 1,"0" is read.*

[EP2 to EP5: bit8:7] EP2 to EP5 reserved bits

In EP2 to EP5, these bits are reserved. Write value should always be "0". They are always read as "0".

[(EP1: bit8:7) bit6:0] PKS: Packet Size Setting bits (PacKet Size ep1 set)

These bits specify the maximum size transferred by one packet. The following shows the maximum packet size that can be specified for Endpoint 1 to Endpoint 5.

EndPoint	Maximum transfer size	Configurable range
1	256 bytes (Odd numbers allowed)	0x001 to 0x100
2 to 5	64 bytes (Odd numbers allowed)	0x01 to 0x40

Notes:

- *A value exceeding the maximum number of transferable bytes (0x100 or 0x40), and "0x00" must not be written. For Endpoint 2 to Endpoint 5, write "00" to bit8 and bit 7. Also when automatic buffer transfer mode (DMAE = 1) is used, 0 to 2 must not be written to the relevant Endpoint.*
- *Set even bytes for PKS.*

5.4 Time Stamp Register (TMSP)

The Time Stamp Register (TMSP) indicates the frame number upon the receipt of SOF packets.

The following figure shows the bit configuration of the Time Stamp Register (TMSP).

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved			TMSP		
Attribute	-	-	-			R	R	R
Initial value	X	X	XXX			0	0	0
RST reset	0	0	Irrelevant			0	0	0

bit	7	6	5	4	3	2	1	0
Field	TMSP							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0
RST reset	0	0	0	0	0	0	0	0

The following explains the function of each bit in the Time Stamp Register (TMSP).

[bit15:11] Reserved: Reserved bits

The written value has no effect on operation. The read value is undefined.

[bit10:0] TMSP: Time Stamp bits (TiMe StamP)

These bits indicate the frame number of a received SOF packet. The frame number is updated upon the receipt of a SOF packet.

5.5 UDC Status Register (UDCS)

The UDC Status Register (UDCS) indicates the bus status during USB communication or the reception of specific commands. Each bit except the SETP bit is an interrupt factor, and so can generate an interrupt to the CPU if the correspondent interrupt enable bit is enabled.

The following figure shows the bit configuration of the UDC Status Register (UDCS).

bit	7	6	5	4	3	2	1	0
Field	-	-	SUSP	SOF	BRST	WKUP	SETP	CONF
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	0	0	0	0	0	0
RST reset	X	X	0	0	0	0	0	0

The following explains the function of each bit in the UDC Status Register (UDCS).

[bit7:6] -: Undefined bits

The written value has no effect on operation. The read value is undefined.

[bit5] SUSP: Suspend detection bit (SUSPend)

This bit indicates that the USB function makes transition to suspend state. The SUSP bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	No suspend has been detected or interrupt factor has been cleared.
1	Suspend has been detected

[bit4] SOF: SOF Detection bit (Start Of Freame)

This bit indicates that a SOF packet has been received, and then the Time Stamp Register value is updated. The SOF bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	No SOF has been received or interrupt factor has been cleared.
1	SOF packet has been received

[bit3] BRST: Bus Reset Detection bit (Bus ReSeT)

This bit indicates the detection of a USB bus reset. The BRST bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	No USB bus reset has been detected or interrupt factor has been cleared.
1	USB bus reset has been detected

Note:

- *When this bit is detected, initialize the buffer by the BFINI bit in the EP0I Status Register (EP0IS), the BFINI bit in the EP0O Status Register (EP0OS), and the BFINI bit in the EP1 to EP5 Status Registers (EP1S to EP5S).*

[bit2] WKUP: Wake-up Detection bit (WaKe UP)

This bit indicates that the USB function has resumed from suspend state. Remote wake-up caused by the RESUM bit setting, and wake-up caused by a request from the host are the resume factors, but the WKUP bit is automatically set only by a resume request by the host. The WKUP bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0".A read-modify-write access reads the bit as "1".

bit	Description
0	No host-caused resume has been detected or interrupt factor has been cleared.
1	Host caused resume has been detected

Note:

- *Even when wake-up caused by a host request occurs, this bit is not set if the RESUM bit in the UDCC register has been set.*

[bit1] SETP: Setup Stage Detection bit (SETuP)

This bit indicates that the received data is the setup stage of USB control transfer. Writing "1" to this bit is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	No SETUP stage has been received or factor has been cleared.
1	Setup stage of control transfer has been received

Note:

- *The SETP bit is not set during standard command automatic response. This bit is not an interrupt factor.*

[bit0] CONF: Configuration Detection bit (CONFigration)

This bit indicates that the USB function has been configured. The CONF bit is set when SetConfig of a USB command is received successfully. The CONF bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	No SetConfig has been detected or interrupt factor has been cleared.
1	SetConfig has been detected

5.6 UDC Interrupt Enable Register (UDCIE)

The UDC Interrupt Enable Register (UDCIE) enables interrupts generated by the factors of the UDC Status Register with respective bits (except for CONFN bit).

The following figure shows the bit configuration of the UDC Interrupt Enable Register (UDCIE).

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	SUSPIE	SOFIE	BRSTIE	WKUPIE	CONFN	CONFIE
Attribute	-	-	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0
RST reset	0	Irrelevant	0	0	0	0	0	0

The following explains the function of each bit in the UDC Interrupt Enable Register (UDCIE).

[bit15:14] Reserved: Reserved bits

Always write "0" to these bits. They are always read as "0".

[bit13] SUSPIE: Suspend Interrupt Enable bit (SUSP Interrupt Enable)

This bit enables interrupts generated by the "SUSP" interrupt factor of the UDC Status Register.

bit	Description
0	Disables interrupts generated by the SUSP factor
1	Enables interrupts generated by the SUSP factor

[bit12] SOFIE: SOF Reception Interrupt Enable bit (SOF Interrupt Enable)

This bit enables interrupts generated by the "SOF" interrupt factor of the UDC Status Register.

bit	Description
0	Disables interrupts generated by the SOF factor
1	Enables interrupts generated by the SOF factor

[bit11] BRSTIE: Bus Reset Enable bit (BRST Interrupt Enable)

This bit enables interrupts generated by the "BRST" interrupt factor of the UDC Status Register.

bit	Description
0	Disables interrupts generated by the BRST factor
1	Enables interrupts generated by the BRST factor

[bit10] WKUPIE: Wake-up Interrupt Enable bit (WKUP Interrupt Enable)

This bit enables interrupts generated by the "WKUP" interrupt factor of the UDC Status Register.

bit	Description
0	Disables interrupts generated by the WKUP factor
1	Enables interrupts generated by the WKUP factor

[bit9] CONFN: Configuration Number Indication bit (CONFiguration Number)

This bit indicates the configuration number. The information is updated when the CONF interrupt factor of the UDC Status Register is set.

bit	Description
0	CONFIG number 0
1	CONFIG number 1

[bit8] CONFIE: Configuration Interrupt Enable bit (CONFiguration)

This bit enables interrupts generated by the "CONF" interrupt factor of the UDC Status Register.

bit	Description
0	Disables interrupts generated by the CONF factor.
1	Enables interrupts generated by the CONF factor.

5.7 EP0I Status Register (EP0IS)

The EP0I Status Register (EP0IS) indicates the status of the Endpoint 0 transfer in the IN direction.

The following figure shows the bit configuration of the EP0I Status Register (EP0IS).

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQIIE	-	-	-	DRQI	-	-
Attribute	R/W	R/W	-	-	-	R/W	-	-
Initial value	1	0	X	X	X	1	X	X
BFINI reset	1	Irrelevant	X	X	X	1	X	X

bit	7	6	5	4	3	2	1	0
Field	-	-	-	-	-	-	-	-
Attribute	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X
BFINI reset	X	X	X	X	X	X	X	X

The following explains the function of each bit in the EP0I Status Register (EP0IS).

[bit15] BFINI: Send Buffer Initialization bit (BuFfer INItial)

This bit initializes the send buffer of transfer data. In addition, this bit is automatically set to "1" when the RST bit in the UDC Control Register (UDCC) is set to "1". If the RST bit was used for resetting, therefore, set the RST bit to "0" before clearing this bit.

bit	Description
0	Clears the initialization
1	Initializes the send buffer

Note:

- *Initialization by the BFINI bit initializes the buffer and the DRQI bit. Before initializing the buffer, make sure that the DRQI or DRQO bit is set, and there is no access from the host, and then configure the STAL bit if necessary.*

[bit14] DRQIIE: Send Data Interrupt Enable bit (Data ReQuest In Interrupt Enable)

This bit enables interrupts generated by the "DRQI" interrupt factor of the EP0I Status Register.

bit	Description
0	Disables interrupts generated by the DRQI factor.
1	Enables interrupts generated by the DRQI factor.

[bit13:11] -: Undefined bits

The written value has no effect on operation. The read value is undefined.

[bit10] DRQI: Send/Receive Data Interrupt Request bit (Data ReQuest In)

This bit indicates that the IN packet transfer from the EP0 host normally ended and data was read out from the send buffer, so that the next send data can be written. The DRQI bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	Clears the interrupt factor
1	Send data can be written to the send buffer

Note:

- *This bit must be cleared after data has been written to the send buffer. Also while this bit is not set, "0" must not be written.*
  Data can be written to the send buffer when DRQI bit is "1". Also when the DRQI bit is cleared, data has been set to the send buffer. When an IN packet request is received while the DRQI bit is "1", therefore, NAK is sent automatically to the host.

[bit9:0] -: Undefined bits

The written value has no effect on operation. The read value is undefined.

5.8 EP0O Status Register (EP0OS)

The EP0O Status Register (EP0OS) indicates the status of the Endpoint 0 transfer in the OUT direction.

The following figure shows the bit configuration of the EP0O Status Register (EP0OS).

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQOIE	SPKIE	-	-	DRQO	SPK	Reserved
Attribute	R/W	R/W	R/W	-	-	R/W	R/W	-
Initial value	1	0	0	X	X	0	0	0
BFINI reset	1	Irrelevant	Irrelevant	X	X	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	SIZE						
Attribute	-	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X
BFINI reset	X	X	X	X	X	X	X	X

The following explains the function of each bit in the EP0O Status Register (EP0OS).

[bit15] BFINI: Receive Buffer Initialization bit (BuFfer INItial)

This bit initializes the receive buffer for transfer data. This bit is also automatically set by setting the RST bit of the UDC Control Register (UDCC). If the RST bit was used for resetting, therefore, set the RST bit to "0" before clearing this bit.

bit	Description
0	Clears the initialization
1	Initializes the receive buffer

Note:

- *Initialization by the BFINI bit initializes the DRQO and SPK bits. Before initializing the buffer, make sure that the DRQI or DRQO bit is set, and there is no access from the host, and then configure the STAL bit if necessary.*

[bit14] DRQOIE: Receive Data Interrupt Enable bit (Data ReQuest Out Interrupt Enable)

This bit enables interrupts generated by the "DRQO" interrupt factor of the EP0O Status Register.

bit	Description
0	Disables interrupts generated by the DRQO factor
1	Enables interrupts generated by the DRQO factor

[bit13] SPKIE: Short Packet Interrupt Enable bit (SPK Interrupt Enable)

This bit enables interrupts generated by the "SPK" interrupt factor of the EP0O Status Register.

bit	Description
0	Disables interrupts generated by the SPK factor
1	Enables interrupts generated by the SPK factor

[bit12:11] -: Undefined bits

The written value has no effect on operation. The read value is undefined.

[bit10] DRQO: Receive Data Interrupt Request bit (Data ReQuest Out)

This bit indicates that the OUT packet transfer from the EP0 host normally ended, and data has been written to the receive buffer, which can be read out. This bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	Clears the interrupt factor
1	Received data can be read from the receive buffer

Note:

- *This bit must be cleared after data has been read from the receive buffer. Also while this bit is not set, "0" must not be written.*
  The receive buffer is not updated when DRQO is "1". The update is allowed when DRQO is cleared. When an OUT packet request is received while the DRQO bit is "1", therefore, NAK is sent automatically to the host.

[bit9] SPK: Short Packet Interrupt Request bit (Short PacKet)

This bit indicates that the data size transferred from the host does not satisfy the maximum packet size (including 0-byte) set by PKS in the EP0 Control Register (EP0C) when the data has been received successfully. This bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	Received data size satisfies the maximum packet size
1	Received data size does not satisfy the maximum packet size

[bit8:7] Reserved: Reserved bits

The written value has no effect on operation. They are always read as "0".

[bit6:0] SIZE: Packet Size Indication bits (packet SIZE)

These bits indicate the number of data bytes written to the receive buffer after EP0's OUT packet transfer has finished. The SIZE bits are updated to a valid value when the DRQO interrupt factor of the EP0O Status Register (EP0OS) has been set.

Example: 8 bytes => "0x08", 64 bytes => "0x40" (maximum value)

5.9 EP1 to EP5 Status Registers (EP1S to EP5S)

The EP1 to EP5 Status Registers (EP1S to EP5S) indicate the status of the Endpoint 1 to Endpoint 5.

The following figure shows the bit configuration of the EP1 to EP5 Status Registers (EP1S to EP5S).

EP1 Status Register (EP1S)

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	SIZE1
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	R
Initial value	1	0	0	X	0	0	0	X

bit	7	6	5	4	3	2	1	0
Field	SIZE1							
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

EP2 to EP5 Status Registers (EP2S to EP5S)

bit	15	14	13	12	11	10	9	8
Field	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	-
Initial value	1	0	0	X	0	0	0	X

bit	7	6	5	4	3	2	1	0
Field	Reserved	SIZE2 to SIZE5						
Attribute	-	R	R	R	R	R	R	R
Initial value	0	X	X	X	X	X	X	X

The following explains the function of each bit in the EP1 to EP5 Control Registers (EP1S to EP5S).

[bit15] BFINI: Send/Receive Buffer Initialization bit (BuFfer INItial)

This bit initializes the send/receive buffer of transfer data. The BFINI bit is also automatically set by setting the RST bit of the UDC Control Register (UDCC). If the RST bit was used for resetting, therefore, set the RST bit to "0" before clearing the BFINI bit.

bit	Description
0	Clears the initialization
1	Initializes the send/receive buffer

Note:

- *The EP1 to EP5 send/receive buffers have a double-buffer configuration. The BFINI bit initialization initializes the double buffers concurrently and also initializes the DRQ and SPK bits. Before initializing the buffer, make sure that the DRQ bit is set, and check the BUSY bit to make sure that there is no access from the host, and then configure the STAL bit.*

[bit14] DRQIE: Packet Transfer Interrupt Enable bit (Data ReQuest Interrupt Enable)

This bit enables interrupts generated by the "DRQ" interrupt factor of the EP1 to EP5 Status Registers.

bit	Description
0	Disables interrupts generated by the DRQ factor
1	Enables interrupts generated by the DRQ factor

Note:

- *To use the automatic buffer transfer mode (DMAE = 1), set DMA and enable transfer before enabling the DRQIE bit.*

[bit13] SPKIE: Short Packet Interrupt Enable bit (SPK Interrupt Enable)

This bit enables interrupts generated by the "SPK" interrupt factor of the EP1 to EP5 Status Registers.

bit	Description
0	Disables interrupts generated by the SPK factor
1	Enables interrupts generated by the SPK factor

[bit12] Reserved: Reserved bit

The written value has no effect on operation. The read value is undefined.

[bit11] BUSY: Busy Flag bit (BUSY flag)

This bit indicates that the host is currently gaining write or read access to the send/receive buffer. The BUSY bit is automatically set or reset.

bit	Description
0	No access from the host
1	Write or read access from the host is in process

Note:

If the BUSY bit is set to "1" while the DRQ bit is set to "1", it indicates that the host is currently accessing either of the double buffers that is not accessed by the CPU or via DMA.
Usually, control using the BUSY bit is not required. To initialize the buffer by setting BFINI, however, take the following steps previously.

1. *Make sure that the DRQ bit has been set, and check the BUSY bit to make sure that there is no access from the host.*
2. *Set the STAL bit.*

[bit10] DRQ: Packet Transfer Interrupt Request bit (Data ReQuest)

This bit indicates that the EP1 to EP5 packet transfer has normally ended, and processing of the data is required. The DRQ bit is an interrupt factor, and writing "1" is ignored. Clear the DRQ bit by writing "0" while it is "1". A read-modify-write access reads the bit as "1".

bit	Description
0	Clears the interrupt factor
1	Packet transfer normally ended

Note:

- *If automatic buffer transfer mode (DMAE = 1) is not used, "0" must be written to the DRQ bit after data has been written or read to/from the send/receive buffer. Switch the access buffers once the DRQ bit is cleared. That DRQ = 0 may not be read after the DRQ bit is cleared. If the transfer direction is set to IN, and the DRQ bit is cleared without writing buffer data while the DRQ bit is "1", it implies that 0-byte data is set. If DIR of the EP1 to EP5 Control Registers (EP1C to EP5C) is set to "1" at initial settings, the DRQ bit of corresponding Endpoint is set at the same time. Also while the DRQ bit is not set, "0" must not be written.*

[bit9] SPK: Short Packet Interrupt Request bit (Short PacKet)

This bit indicates that the data size transferred from the host does not satisfy the maximum packet size (including 0-byte) set by PKS in the EP1 to EP5 Control Registers (EP1C to EP5C) when the data has been received successfully. This bit is an interrupt factor, and writing "1" is ignored. Clear it by writing "0". A read-modify-write access reads the bit as "1".

bit	Description
0	Received data size satisfies the maximum packet size
1	Received data size does not satisfy the maximum packet size

Note:

- *The SPK bit is not set during data transfer in the IN direction.*

[EP2 to EP5: bit8:7] Reserved: Reserved bits

In EP2 to EP5, these bits are reserved. The written value has no effect on operation. They are always read as "0".

[(EP1: bit8:7) bit6:0] SIZE: packet SIZE

These bits indicate the number of data bytes written to the receive buffer when OUT packet transfer of EP1 to EP5 has finished. The SIZE bit is updated to a valid value when the DRQ interrupt factor of the EP1 to EP5 Status Registers (EP1S to EP5S) has been set.

The maximum transfer data size of Endpoint 1 to Endpoint 5 is as follows:

EndPoint	Maximum transfer size	Indication range
1	256 bytes	0x000 to 0x100
2 to 5	64 bytes	0x00 to 0x40

Note:

- *These bits are set to the data size transferred from the host in the OUT direction and written to the buffer. Therefore, a value read during transfer in the IN direction has no effect on operation.*

5.10 EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)

The EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) control writing or reading transfer data to/from the send/receive buffer for Endpoint 0 to Endpoint 5.

The following figure shows the bit configuration of the EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

EP0DTH to EP5DTH

bit	15	14	13	12	11	10	9	8
Field	BFDT							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

EP0DTL to EP5DTL

bit	7	6	5	4	3	2	1	0
Field	BFDT							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

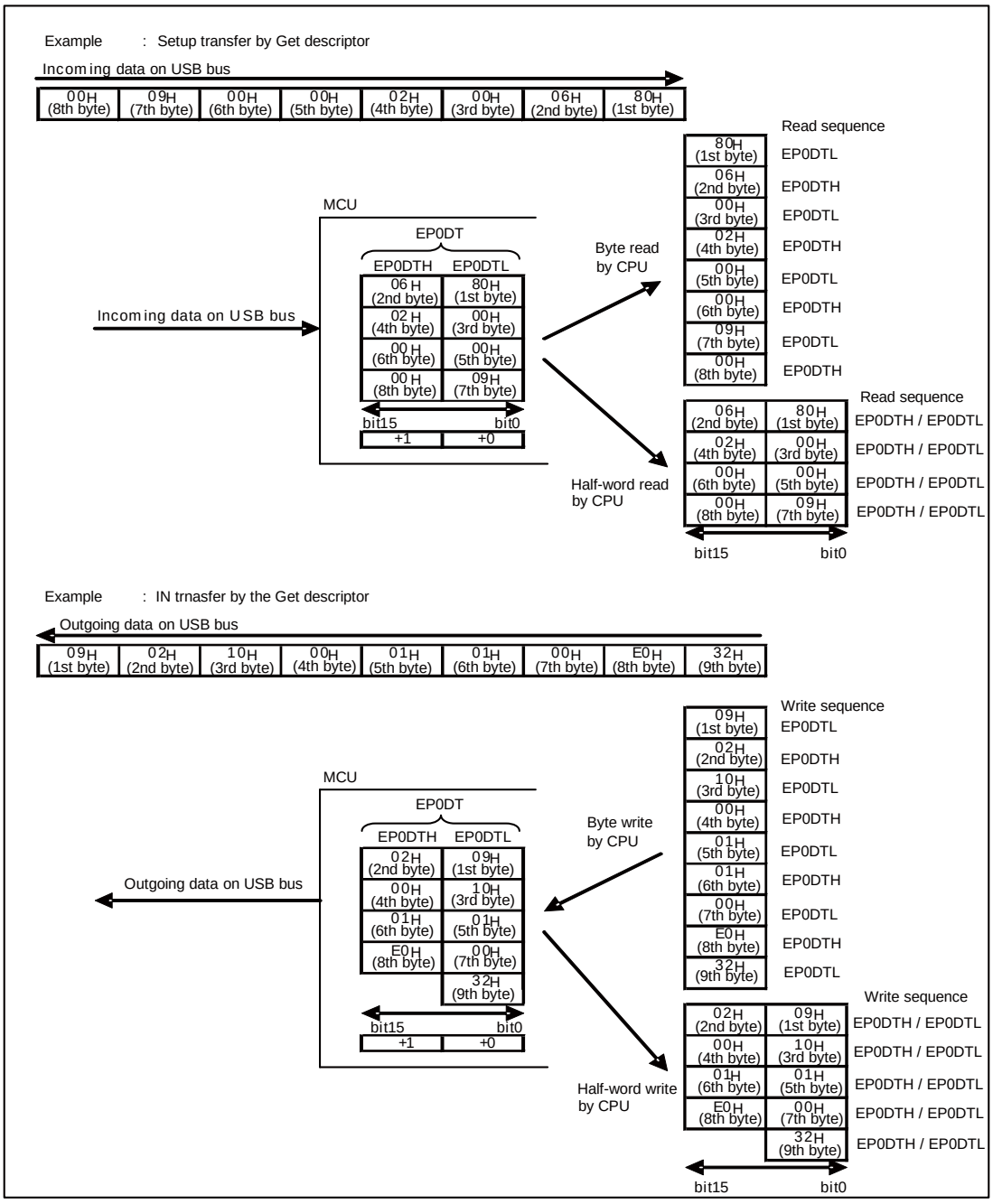
The following explains the function of each bit in the EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

[bit15:0] BFDT: Endpoint Send/Receive Buffer Data bits (BuFfer DaTa)

A register used for data write/read to/from the send/received buffer for each end point.

Notes:

- *The CPU can access the EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) either by the byte or by the half-word.*
- *Byte access*
  First access low-order (EPxDTL) and then high-order (EPxDTH). Subsequently, access low-order (EPxDTL) and high-order (EPxDTH) alternately.
- *This register must not be accessed by the bit operation instruction.*



The DMA transfer can only access the EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) by the half-word. (See "n Automatic data size transfer mode" of "3.6 DMA transfer function".)

CHAPTER 3-2: USB Host

This chapter explains the functions and operations of the USB host.

1. Overview of USB host
2. USB host configuration
3. USB host operations
4. USB host setting procedure examples
5. USB host registers

CODE: FW03H-E18.4

1. Overview of USB host

This section explains the functions and operations of the USB host.

Features of USB host

The USB host has the following features:

- Automatic detection of full-speed or low-speed transfer
- Support of full-speed or low-speed transfer
- Automatic detection of device connection or disconnection
- Support of USB bus reset sending function
- Support of IN, OUT, SETUP, and SOF tokens
- Automatic sending of handshake packet for IN token (excluding STALL)
- Automatic detection of handshake packet for OUT token
- Support of maximum packet length of up to 256 bytes
- Support of actions against errors (CRC error, toggle error, and timeout)
- Support of Wake-up function
- Support of Spansion's original USB host functions which can also be operated as USB functions by switching the operation mode. (For restrictions in the USB host specifications, see Table 1-1.)

Note:

- *Set the base clock to 13 MHz or higher when using the USB host.*

Table 1-1 Restrictions in USB host specifications

		Host
Hub support		❍*1
Transfer functions	Bulk transfer	❍
	Control transfer	❍
	Interrupt transfer	❍
	Isochronous transfer	❍
Transfer speed modes	Low Speed	❍
	Full Speed	❍
PRE packet support		×
SOF packet support		❍
Error types	CRC error	❍
	Toggle error	❍
	Timeout	❍
	Max. packet < Received data	❍
Detection of device connection or disconnection		❍
Detection of transfer speed		❍

❍: Supported.

×: Not supported.

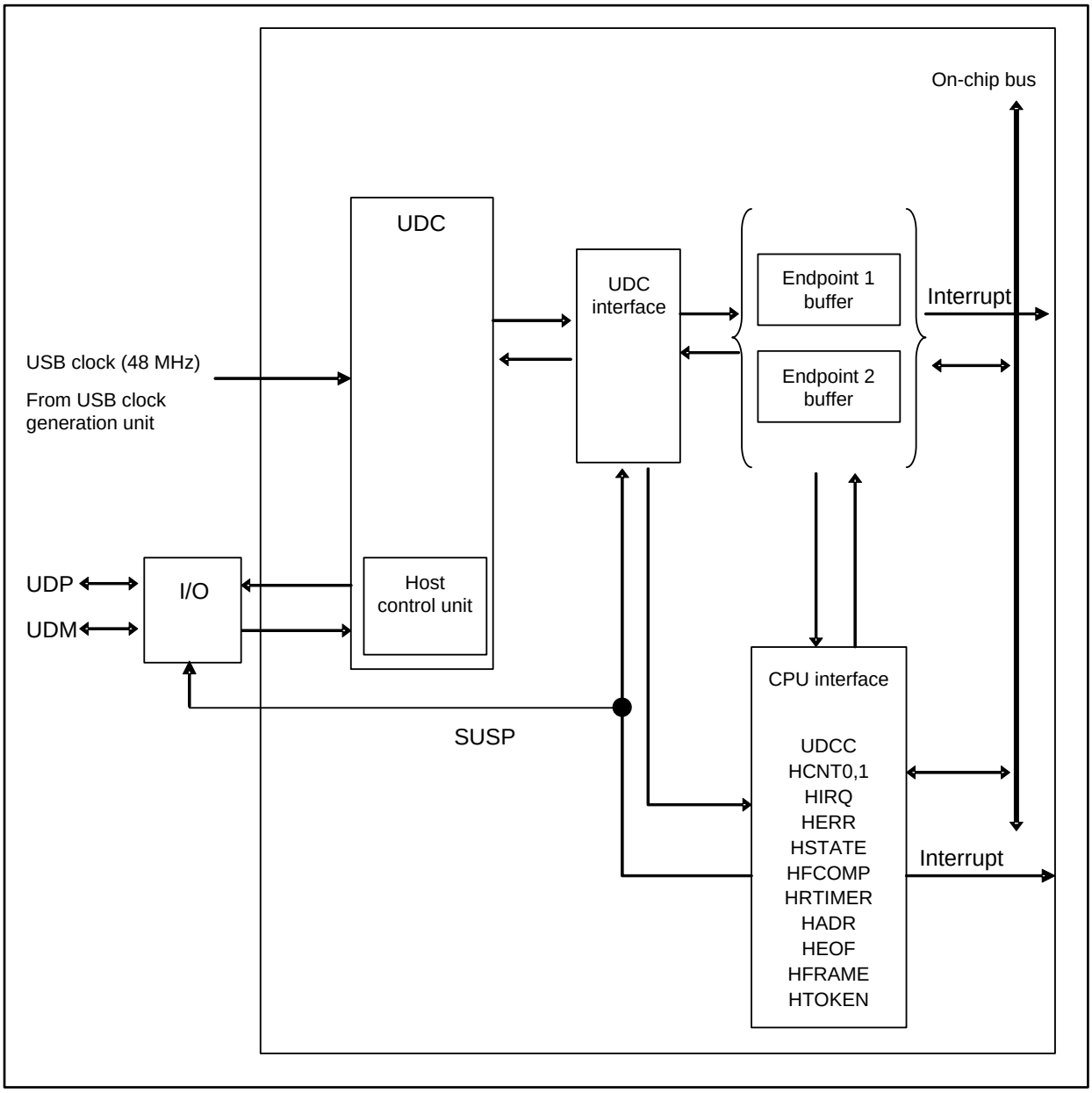
*1 : Supports a hub of up to one stage in only the full-speed mode.

2. USB host configuration

Figure 2-1 shows the USB host block diagram.

USB host block diagram

Figure 2-1 USB host block diagram



3. USB host operations

This section explains the operations of the USB host.

3.1. Device connection
3.2. USB bus resetting
3.3. Token packet
3.4. Data packet
3.5. Handshake packet
3.6. Retry function
3.7. SOF interrupt
3.8. Error status
3.9. End of packet
3.10. Suspend and resume operations
3.11. Device disconnection

3.1 Device connection

This section shows how to detect that an external USB device is connected using software.

Host function setting

To carry out USB operation, configure the setting of the USB clock generation unit and enable the USB clock output while the USBEN bit of the USB Enable Register (USBEN) is "0" (USB operation disabled). Next, set the USBEN bit to "1" (USB operation enabled). Then, to operate the USB as a host, set "1" to the HOST bit of Host Control Register 0 (HCNT0).

States whether or not an external USB device is connected

When an external USB device is not connected, both of host pins D+ and D- are set to "LOW" by the pull-down resistor. In this case, the CSTAT bit of the Host Status Register (HSTATE) is "0" and the TMODE bit is undefined. When an external USB device is connected, the CSTAT bit of the Host Status Register (HSTATE) is changed to "1".

Detection of external USB device connection

When a connection of an external USB device is detected, the CNNIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". If "1" is set to the CNNIRE bit of Host Control Register 0 (HCNT0), a device connection interrupt occurs. To clear this interrupt, write "0" to the CNNIRQ bit of the Host Interrupt Register (HIRQ). When detecting a device connection by polling, instead of an interrupt, use the following steps to create a program.

1. Set the CNNIRE bit of Host Control Register 0 (HCNT0) to "0".
2. Check that the CNNIRQ bit of the Host Interrupt Register (HIRQ) changes to "1".

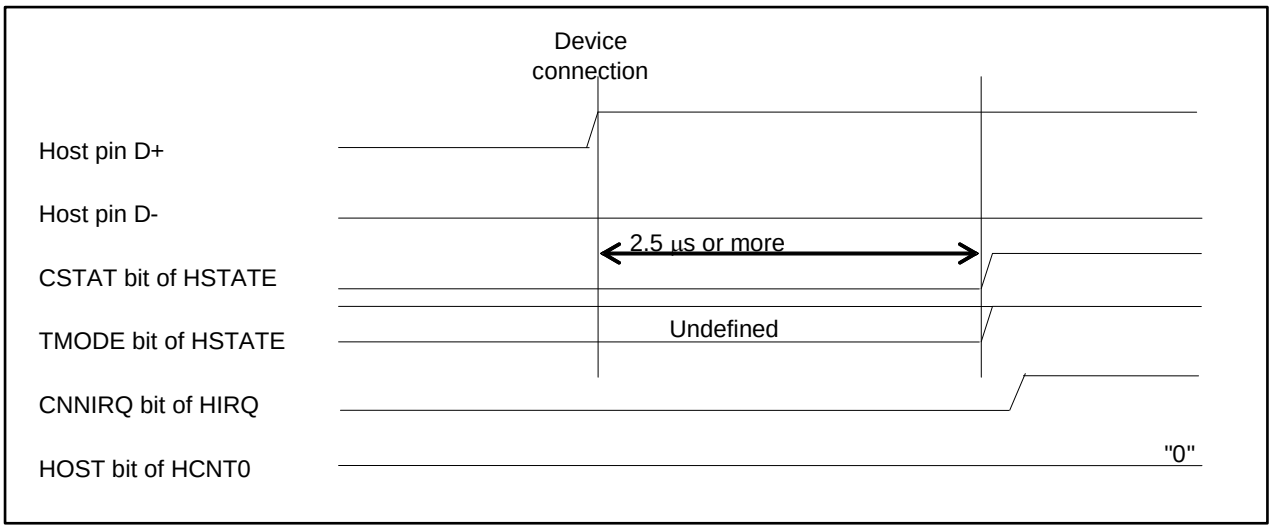
Obtaining the transfer speed of the remote USB device and selecting clocks

To obtain the possible transfer speed of the remote USB device after detecting a connection, check the value of the TMODE bit of the Host Status Register (HSTATE). The following shows the relationships between the transfer speed and the value of the TMODE bit of the Host Status Register (HSTATE).

- The destination is a device in the full-speed mode. -> TMODE="1"
- The destination is a device in the low-speed mode. -> TMODE="0"

If the RST bit of the UDC control register (UDCC) is "1" after obtaining the transfer speed of an external USB device, update the CLKSEL bit of the Host Status Register (HSTATE) according to the obtained transfer speed.

Figure 3-1 Full-speed device connection detection timing example (HCNT0:HOST="0")



Notes:

- *When 2.5µs or more lapsed after an external USB device was connected, the CSTAT bit of the Host Status Register (HSTATE) is changed to "1".*
- *The TMODE and CSTAT bits of the Host Status Register (HSTATE) are updated regardless of the setting of the HOST bit of Host Control Register 0 (HCNT0). The CNNIRQ and DIRQ bits of the Host Interrupt Register (HIRQ) are set to "1" if conditions are satisfied.*

3.2 USB bus resetting

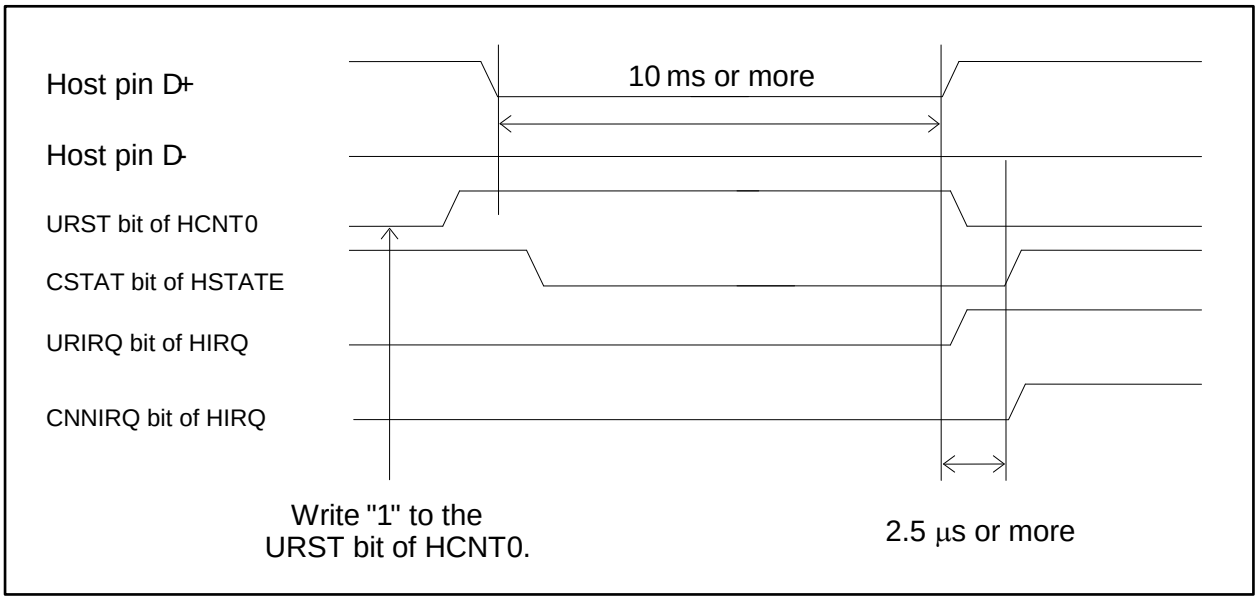
The USB bus is reset by sending SE0 for 10 ms or more if the URST bit of Host Control Register 0 (HCNT0) is set to "1" in the host mode. After USB bus resetting has been completed, the URST bit of Host Control Register 0 (HCNT0) is set to "0", and the URIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". If the URIRE bit of Host Control Register 0 (HCNT0) is then set to "1", an interrupt occurs. To clear this interrupt, write "0" to the URIRQ bit of the Host Interrupt Register (HIRQ).

Notes on before and after resetting the USB bus

Note the following points when resetting the USB bus.

1. To check that the device is connected before resetting the USB bus, make sure that the CSTAT bit of the Host Status Register (HSTATE) is set to "1".
2. Resetting the USB bus changes the CSTAT bit of the Host Status Register (HSTATE) to "0", resulting in the USB device being disconnected. At this time, the DIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1".
3. After USB bus resetting has been completed, compare the value of the CLKSEL bit with that of the TMODE bit in the Host Status Register (HSTATE). If they do not match, update the CLKSEL bit to make a match. Update the CLKSEL bit when the RST bit of the UDC Control Register (UDCC) is "1".
4. After USB bus resetting has been completed, check that the USB device is connected using one of the bits shown below, and execute token processing.
   - CNNIRQ bit of Host Interrupt Register (HIRQ)
   - CSTAT bit of Host Status Register (HSTATE)

Figure 3-2 Device resetting timing example



Note:

- *No token is issued if a connection of the USB device is not detected after USB bus resetting has been completed.*

3.3 Token packet

When issuing an IN, OUT, or SETUP token in the host mode, use the following setting steps to send a token packet.

1. Set the Host Address Register (HADR).
2. Set the DIR and PKS bits of the EP1 Control Register (EP1C) or EP2 Control Register (EP2C).
3. Write the required data to the Host Token Endpoint Register (HTOKEN).

When issuing an SOF token, set the Frame Setup Register (HFRAME) and EOF Setup Register (HEOF), and write the required data to the Host Token Endpoint Register (HTOKEN). The setting above is not required if no change is made in the HADR, EP1C, EP2C, HFRAME, and HEOF registers.

Token packet setting

In the host mode, use endpoint 1 and endpoint 2 buffers to send and receive data.
When issuing an IN, OUT, or SETUP token, specify the destination address in the Host Address Register (HADR). Then, specify the maximum number of bytes for each packet in the PKS bit and the transfer direction of each packet in the DIR bit of the EP1 Control Register (EP1C) or EP2 Control Register (EP2C) respectively.

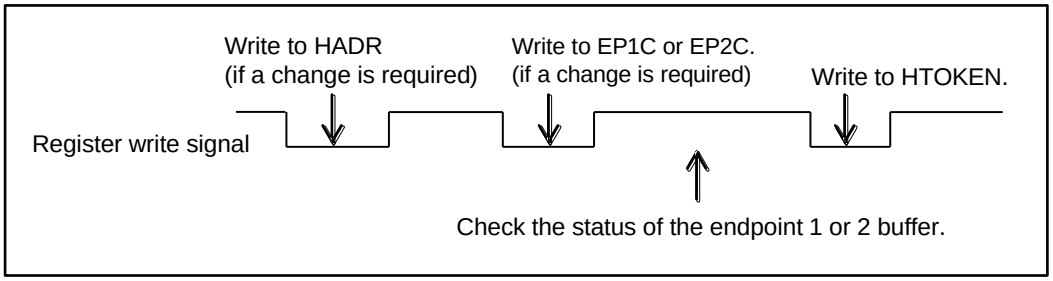
If the DIR bit of the EP1 Control Register (EP1C) is "1", the endpoint 1 buffer is used as an OUT buffer. The endpoint 2 buffer is used as an IN buffer. Then set "0" to the DIR bit of the EP2 Control Register (EP2C).
If the DIR bit of the EP1 Control Register (EP1C) is "0", the endpoint 1 buffer is used as an IN buffer. The endpoint 2 buffer is used as an OUT buffer. Then set "1" to the DIR bit of the EP2 Control Register (EP2C).

Take the following steps to execute token processing.

1. Specify the DIR and PKS bits of the EP1 Control Register (EP1C) and EP2 Control Register (EP2C).
2. If the target endpoint n (n: 1 or 2) is set to the OUT direction, write send data to the endpoint n (n: 1 or 2) buffer. Also set "0" to the DRQ bit of the EPn Status Register (EPnS: n = 1 or 2).
   If the IN direction is selected, read the DRQ bit of the EPn Status Register (EPnS: n = 1 or 2), and check that its value is "0".
3. Specify the target endpoint, token, and toggle data in the Host Token Endpoint Register (HTOKEN).

The USB circuit sends a token packet in the order of Sync, token, address, endpoint, CRC5, and EOP based on the specified token; however, Sync, CRC5, and EOP are sent automatically. After one packet has been sent, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is set to "0b000" (see "3.7 SOF interrupt"). At this time, if the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. To clear this interrupt, write "0" to the CMPIRQ bit of the Host Interrupt Register (HIRQ).

Figure 3-3 Example of register setting to issue an IN, OUT, or SETUP token



When issuing an SOF token, specify the EOF time in the EOF Setup Register (HEOF) and the frame number in the Frame Setup Register (HFRAME) respectively. Then specify an SOF token code in the TKNEN bit of the Host Token Endpoint Register (HTOKEN). After this, Sync, SOF token, frame number, CRC5, and EOP are sent, the SOFBUSY bit of the Host Status Resister (HSTATE) is set to "1", and the

Frame Setup Register (HFRAME) is incremented by one. The CMPIRQ bit of the Host Interrupt Register (HIRQ) is also set to "1", causing the TKNEN bit of the Host Token Endpoint Register (HTOKEN) to be cleared to "(000)b". If the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. When SOF is sent automatically, an interrupt by CMPIRQ does not occur. To clear a token completion interrupt, write "0" to the CMPIRQ bit of the Host Interrupt Register (HIRQ).
SOF is automatically sent every 1 ms while the SOFBUSY bit of the Host Status Register (HSTATE) is "1". The following shows the conditions (SOF stop conditions) to set the SOFBUSY bit of the Host Status Register (HSTATE) to "0".

- Write "0" to the SOFBUSY bit of the Host Status Register (HSTATE).
- Reset the USB bus (write "1" to the URST bit of HCNT0).
- Write "1" to the SUSP bit of the Host Status Register (HSTATE).
- Disconnect the USB device (when the CSTAT bit of HSTATE is "0").

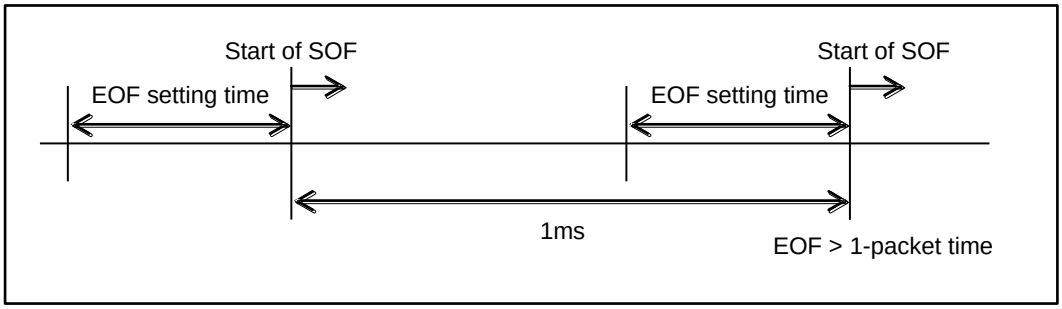
Take the following steps to change the USB from the host mode to the function mode.

1. Set "0" to the SOFBUSY bit of the Host Status Register (HSTATE).
2. Check the following conditions.
   - The SOFBUSY bit of the Host Status Register (HSTATE) is cleared to "0".
   - The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is set to "000".
   - The SUSP bit of the Host Status Register (HSTATE) is set to "0".
3. Set "1" to the RST bit of the UDC Control Register (UDCC).
4. Change the operation mode from the host mode to the function mode.

To set the SOFBUSY bit of the Host Status Register (HSTATE) to "1" again, send an SOF token once more.

The EOF Setup Register is used to prevent SOF from being sent simultaneously with other tokens. If the TKNEN bit of the Host Token Endpoint Register (HTOKEN) is written in the period from the EOF setting time to the SOF starting time, the specified token is placed into the wait state. After SOF has been sent, the token in the wait state is issued. The EOF Setup Register specifies a 1-bit time as the time unit. For example, if "0x10" is specified in the EOF Setup Register, the time is set to 16×1/12MHz=1333.3ns in the full-speed mode and 16×1/1.5MHz=10666.6ns in the low-speed mode. When the EOF setting time is shorter than the 1-packet time, SOF may be sent doubly during execution of other token. In this case, the LSTSOF bit of the Host Error Status Register (HERR) is set to "1", and SOF is not sent. If "1" is set to the LSTSOF bit of the Host Error Status Register (HERR), the value of the EOF Setting Register must be increased (see the explanation of the EOF Setup Register).

Figure 3-4 SOF timing



3.4 Data packet

When sending a data packet after a token packet, transfer toggle data based on the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN). Further, send endpoint 1 or 2 buffer data, CRC16 data, and EOP depending on the value of the DIR bit of the EP1 Control Register (EP1C).
When receiving a data packet, compare the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN) with the received toggle data. If they match, the received data is distributed to the Endpoint 1 or Endpoint 2 buffer depending on the value of the DIR bit of the EP1 Control Register (EP1C) to check for a CRC16 error.

Data packet

Take the following steps to send or receive a data packet after sending a token packet.

1. For sending
   - Automatically send Sync.
   - If the TGGL bit of the Host Token Endpoint Register (HTOKEN) is "0", send DATA0. If the TGGL bit is "1", send DATA1.
   - If the DIR bit of the EP1 Control Register (EP1C) is "1", select the Endpoint 1 buffer. If the DIR bit of the EP1 Control Register (EPIC) is "0", select the Endpoint 2 buffer. Then, send all the target data.
   - Send a 16-bit CRC.
   - Send a 2-bit EOP.
   - Send a 1-bit J State.

2. For receiving
   - Receive Sync.
   - Receive toggle data, and compare it with the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN).
   - If the toggle data matches the value of the TGGL bit, check the DIR bit of the EP1 Control Register (EP1C). If the DIR bit is "1", select the Endpoint 2 buffer. If the DIR bit of the EP1 Control Register (EP1C) is "0", select the Endpoint 1 buffer. Then, distribute the received data to the respective buffers.
   - Verify the 16-bit CRC when EOF is received.

When the HOST bit of Host Control Register 0 (HCNT0) is "1", set the inverted value to the respective DIR bits of the EP1 Control Register (EP1C) and EP2 Control Register (EP2C). For example, if "0" is set to the DIR bit of the EP1 Control Register (EP1C), set "1" to the DIR bit of the EP2 Control Register (EP2C).

3.5 Handshake packet

A handshake packet is used to notify the remote device of the status of the local device.

Handshake packet

A handshake packet sends either one of ACK, NAK, and STALL from the receiving side when it is judged that the receiving side is ready to receive data normally. If the USB circuit receives a handshake packet, the type of the received handshake packet is set to the HS bit of the Host Error Status Register (HERR). If the USB circuit sends a handshake packet, the type of the sent handshake packet is set to the HS bit of the Host Error Status Register (HERR).

3.6 Retry function

When a NAK or CRC error occurs at the end of a packet, if "1" is set to the RETRY bit of Host Control Register 1 (HCNT1), processing is retried repeatedly for the period specified in the Retry Timer Setting Register (HRTIMER).

Retry function

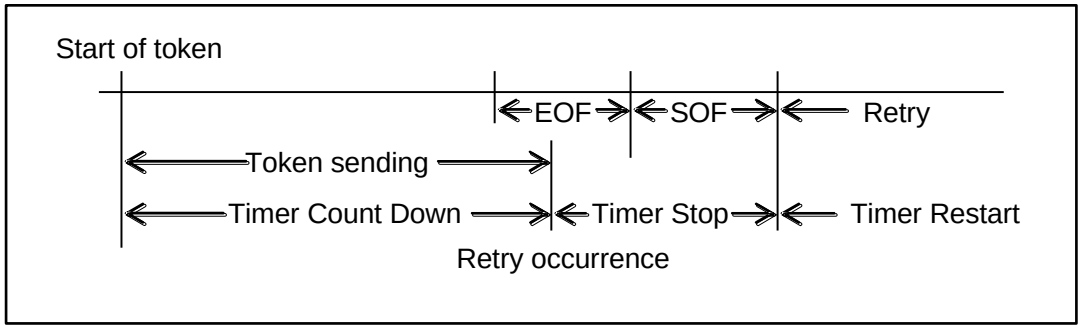
When an error* other than STALL or device disconnection occurs, the target token is retried if the RETRY bit of Host Control Register 1 (HCNT1) is "1". The following shows the conditions to end retry processing.

* : HERR:HS="01", HERR:RERR="1", HERR:TOUT="1", HERR:TGERR="1",HERR:CRC="1", HERR:STUFF="1"

- The RETRY bit of Host Control Register 1 (HCNT1) is set to "0".
- "0" is detected in the retry timer.
- The interrupt flag is generated by SOF (SOFIRQ of HIRQ = "1").
- ACK is detected.
- A device disconnection is detected.

The retry timer is activated at start of a token, and counted down by a 1-bit transfer clock. If retry occurs in the EOF area, counting stops. If a SOF token is ended while the SOFIRQ bit of HIRQ is "0", counting restarts from the timer value at the time when counting stopped. When the retry timer runs out to "0" and a packet ends, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Figure 3-5 Retry timer operation (SOFIRQ of HIRQ = "0")



When retry processing is ended, end information of the EOP is set to each register.

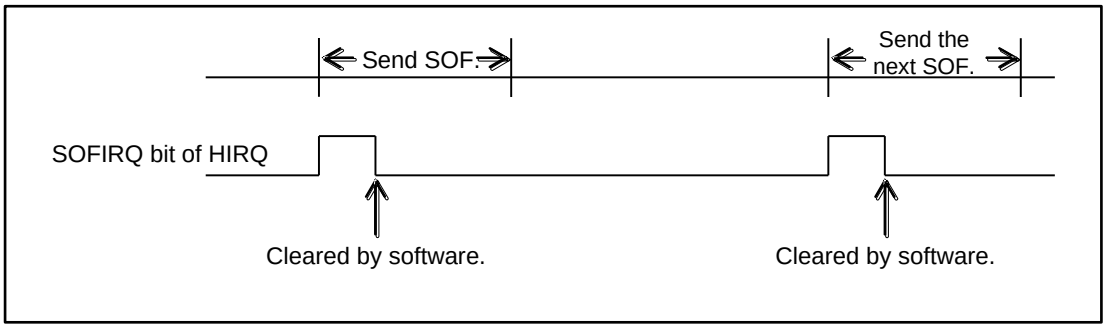
3.7 SOF interrupt

The SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1" at start of SOF depending on the setting of the SOFSTEP bit of Host Control Register 1 (HCNT1) and SOF Interrupt Frame Compare Register (HFCOMP). If the SOFIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. When SOF processing is executed using the Host Token Endpoint Register (HTOKEN), the SOFIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1".

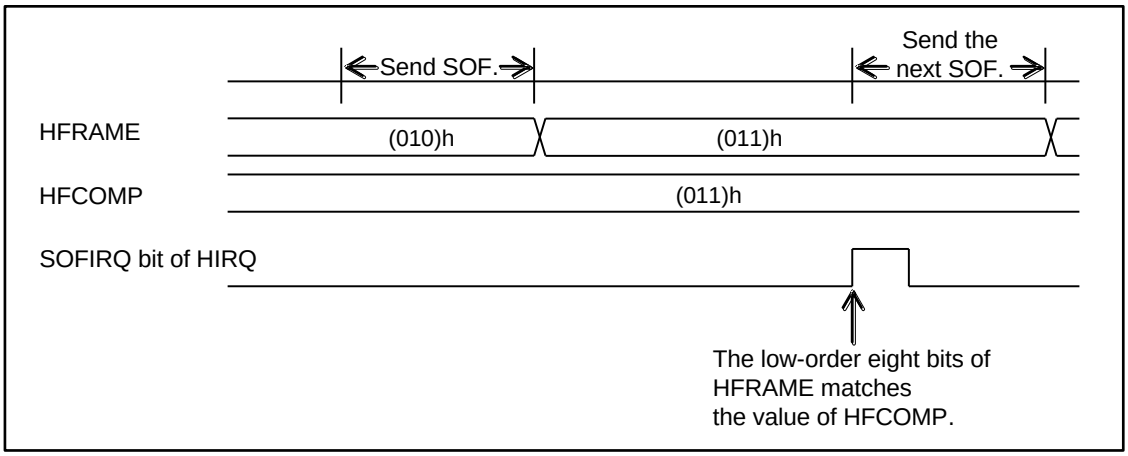
SOF interrupt

When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "0", the value of the SOF Interrupt Frame Compare Register (HFCOMP) is compared with the low-order eight bits of the frame number for SOF token. If they match, "1" is set to the SOFIRQ bit of the Host Interrupt Register (HIRQ) when sending SOF. When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "1", "1" is set to the SOFIRQ bit of the Host Interrupt Register (HIRQ) each time SOF is sent.

1. When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "1":



2. When the SOFSTEP bit of Host Control Register 1 (HCNT1) is "0":



If "1" is set to the CANCEL bit of Host Control Register 1 (HCNT1), the target token is not sent when it is set at the following timing.

- A token other than SOF is set to the Host Token Endpoint Register (HTOKEN) in the EOF area.

If a token is set at this timing, the following operations are carried out.

- If the SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1" when the next SOF is sent, the TKNEN bit of the Host Token Endpoint Register (HTOKEN) is immediately cleared to "0b000". In this case, that token is not sent.

The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is cleared at the following timing.

At this timing, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1". When the SOFIRQ bit is set to "1", the TCAN bit of the Host Interrupt Register (HIRQ) indicates that a token is canceled. When retrying to send a token, write "0" to the TCAN bit of the Host Interrupt Register (HIRQ). Then write a token to be sent to the TKNEN bit of the Host Token Endpoint Register (HTOKEN).

If "0" is set to the CANCEL bit of Host Control Register 1 (HCNT1), the token specified in the Host Token Endpoint Register (HTOKEN) is sent following SOF.

Figure 3-6 Token cancellation example at CANCEL bit of HCNT1 = "1"

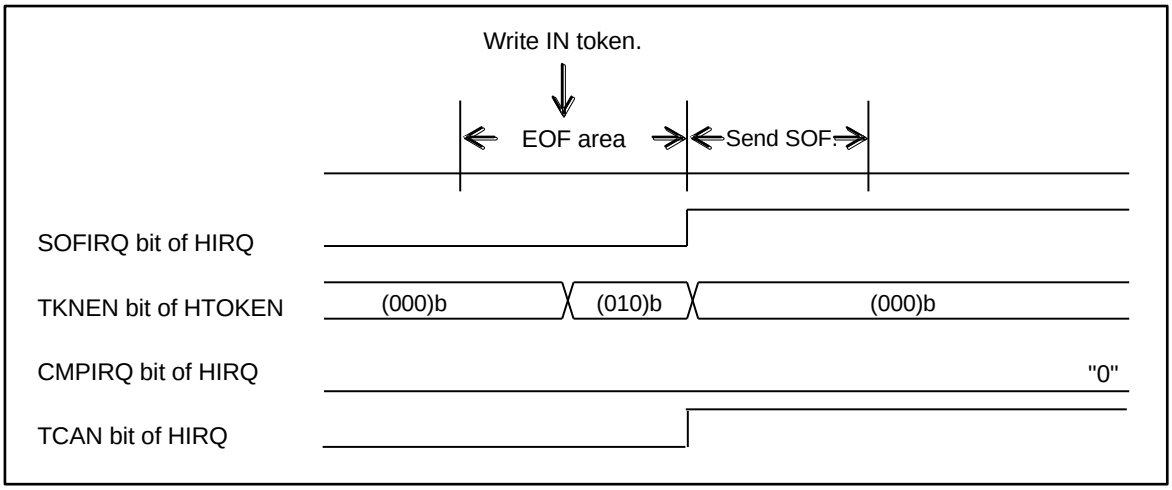
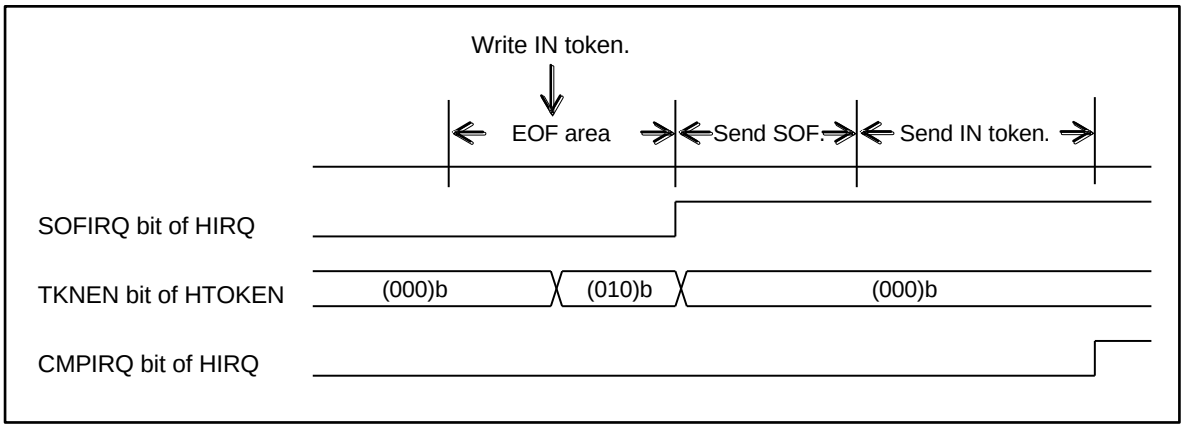


Figure 3-7 Token operation example at CANCEL bit of HCNT1 = "0"



3.8 Error status

The USB host supports error information.

Error status

1. Stuffing Error

   If "1" is successively set to six bits, "0" is inserted into one bit. If "1" is successively detected in seven bits, it is judged to be Stuffing Error, and the STUFF bit of the Host Error Status Register (HERR) is set to "1". To clear this status, write "0" to the STUFF bit. If the next token is sent without clearing the STUFF bit, a factor is reflected on the STUFF bit when the next token is ended.

2. Toggle Error

   When sending an IN token, the toggle data of a data packet is compared with the value of the TGGL bit of the Host Token Endpoint Register (HTOKEN). If they do not match, the TGERR bit of the Host Error Register (HERR) is set to "1". To clear the TGERR bit, write "0" to the TGERR bit of the Host Error Register (HERR). If the next token is sent without clearing the TGERR bit, a factor is reflected on the TGERR bit when the next token is ended.

3. CRC Error

   When receiving an IN token, data and CRC of the received data packet are obtained with the CRC polynomial "G(X) = X16 + X15 + X2 + 1". If the remainder is not "(800d)h", it means that CRC Error occurs, and the CRC bit of the Host Error Register (HERR) is set to "1". To clear the CRC bit, write "0" to the CRC bit of the Host Error Register (HERR). If the next token is sent without clearing the CRC bit, a factor is reflected on the CRC bit when the next token is ended.

4. Time Out Error

   "1" is set to the TOUT bit of the Host Error Status Register (HERR) when:

   - A data packet or handshake packet has not been input in the specified time;
   - SE0 has been detected during data receiving; or
   - Stuffing Error has been detected.

   To clear the TOUT bit, write "0" to the TOUT bit of the Host Error Register (HERR). If the next token is sent without clearing the TOUT bit, a factor is reflected on the TOUT bit when the next token is ended.

5. Receive Error

   If EP1 is used as a receive buffer, the value of the PKS bit of the EP1 Control Register (EP1C) is used as the receive packet size. If EP2 is used as a receive buffer, the value of the PKS bit of the EP2 Control Register (EP2C) is used as the receive packet size. When the received data exceeds the specified receive packet size, the RERR bit of the Host Error Status Register (HERR) is set to "1". To clear the RERR bit, write "0" to the RERR bit of the Host Error Register (HERR). If the next token is sent without clearing the RERR bit, a factor is reflected on the RERR bit when the next token is ended.

3.9 End of packet

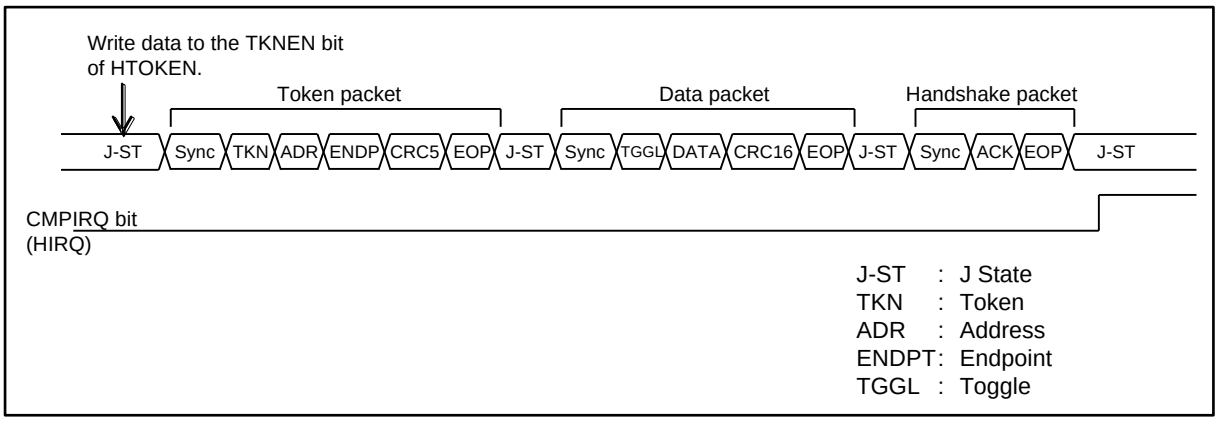
If one packet is ended in the USB host, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". At this time, if the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs.

Packet end timing

When one packet ends, the interrupt flag is generated when:

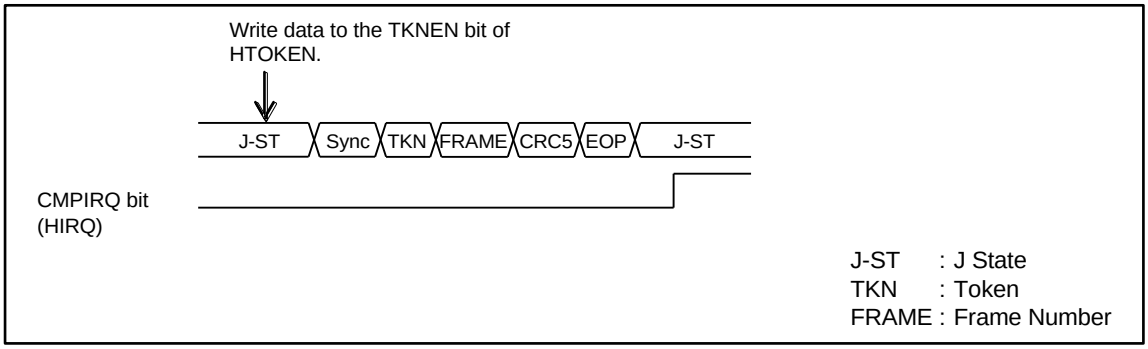
- The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is "(001)b", "(010)b", or "(011)b" (SETUP token, IN token, or OUT token).

Figure 3-8 Timing example 1 when setting the CMPIRQ bit of the Host Interrupt Register (HIRQ)



The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is "(100)b" (SOF token).

Figure 3-9 Timing example 2 (SOF token) when setting the CMPIRQ bit of the Host Interrupt Register (HIRQ)



3.10 Suspend and resume operations

The USB host supports suspend and resume operations.

Suspend operation

If "1" is set to the SUSP bit of the Host Status Register (HSTATE), the procedure below is performed, and the USB circuit is placed into the suspend state.

- The USB bus is placed in the high-impedance state.
- A circuit block with no clock required is stopped.

If the USB circuit is placed in the suspend state, the SUSP bit of the Host Status Register (HSTATE) is set to "1".

However, the following operations are prohibited while resetting the USB bus.

- "1" is set to the SOFBUSY bit of the Host Status Register (HSTATE) or the USB circuit is placed into the suspend state during data transfer.
- Clocks supplied to the USB are stopped in the suspend state.

Take the following steps to stop clocks.

1. Change to the stop or timer mode.
2. Set the UCEN bit of the USB Clock Setup Register (UCCR) to "0".

Resume operation

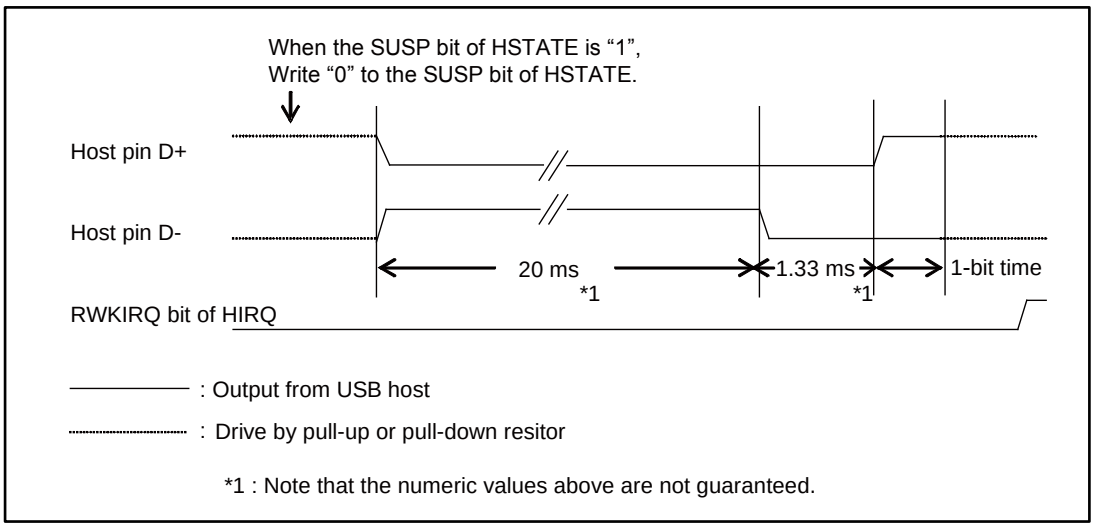
The USB bus changes from the suspend state to the resume state to resume processing when one of the following conditions is satisfied.

- "0" is set to the SUSP bit of the Host Status Register (HSTATE).
- The host pin D+ or D- is placed in the K-state mode.
- A device disconnection is detected.
- A device connection is detected.

After the RWKIRQ bit of the Host Interrupt Register (HRQ) has been set to "1", a token can be issued. The following shows the operation timing for each condition.

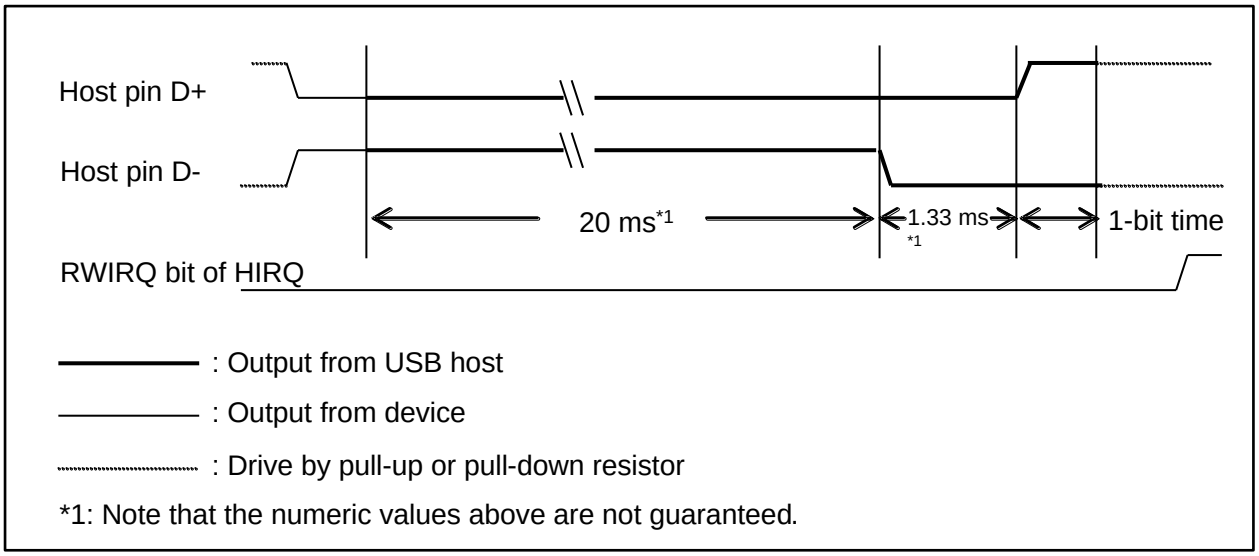
- "0" is set to the SUSP bit of the Host Status Register (HSTATE).

Figure 3-10 Resume operation with register (Full-speed mode)



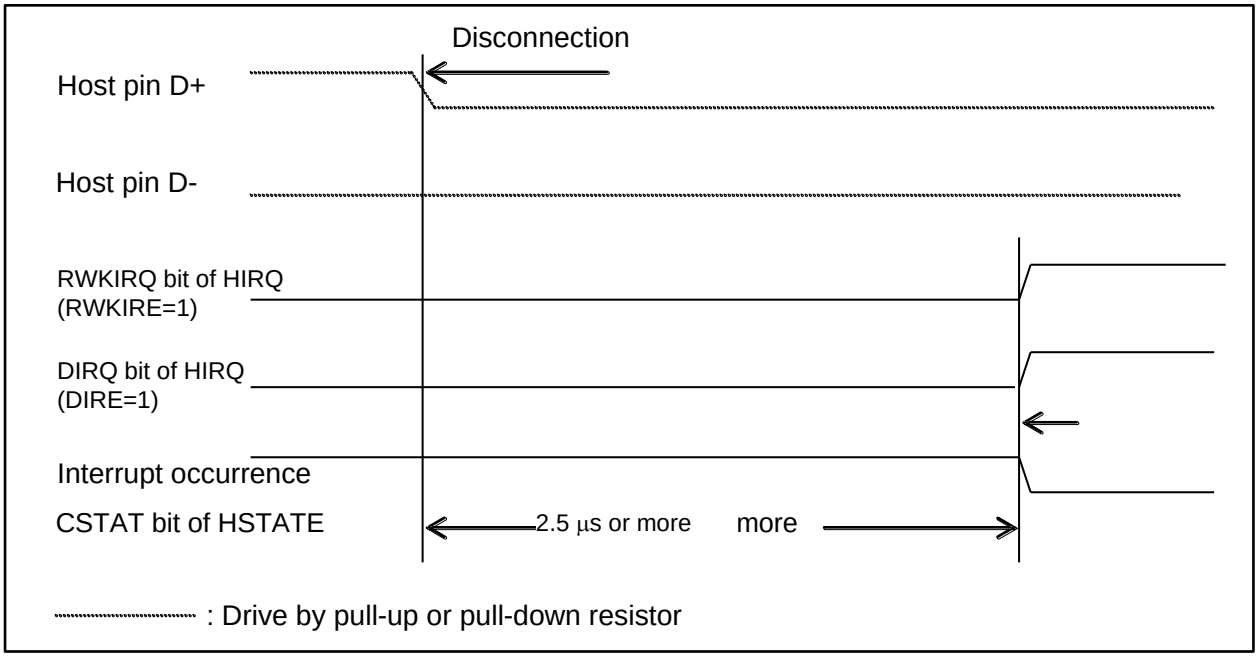
- The state that host pin D+ or D- is placed in the K-state mode has been detected.

Figure 3-11 Resume operation by device (Full-speed mode)



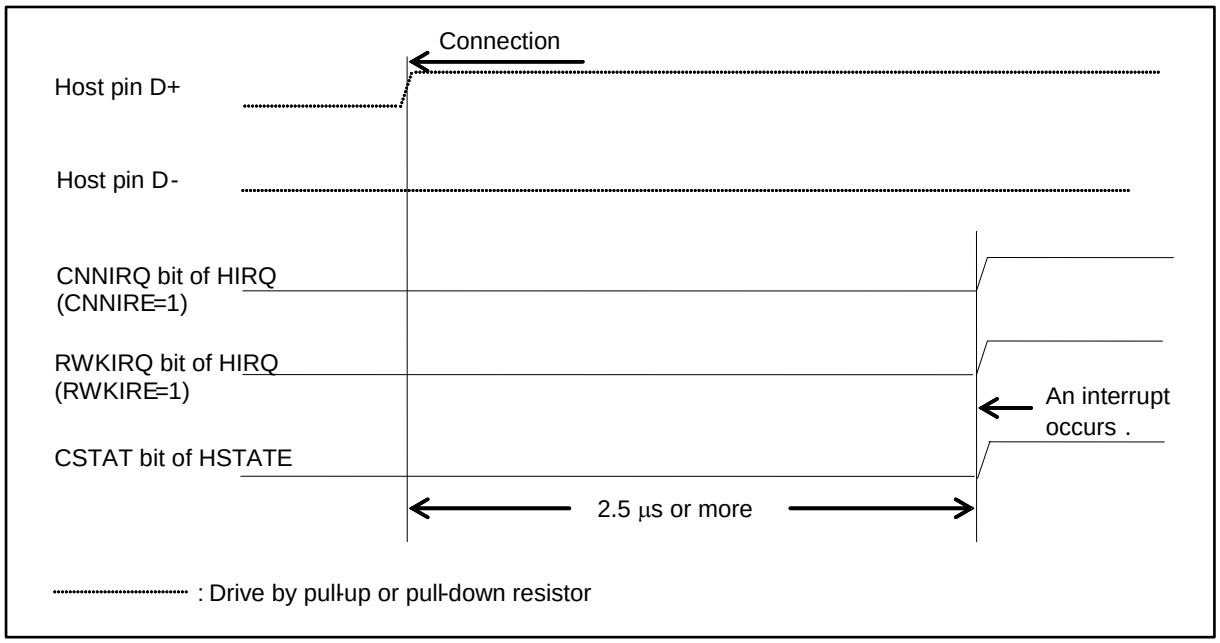
- A device disconnection is detected.

Figure 3-12 Resume operation by device disconnection



- A device connection is detected.

Figure 3-13 Resume operation by device connection (Full-speed mode)



3.11 Device disconnection

The device disconnection timer starts when both the host pins D+ and D- are set to "LOW". If "LOW" is detected for 2.5 μs or more, the CSTAT bit of the Host Status Register (HSTATE) is set to "0".

Device disconnection

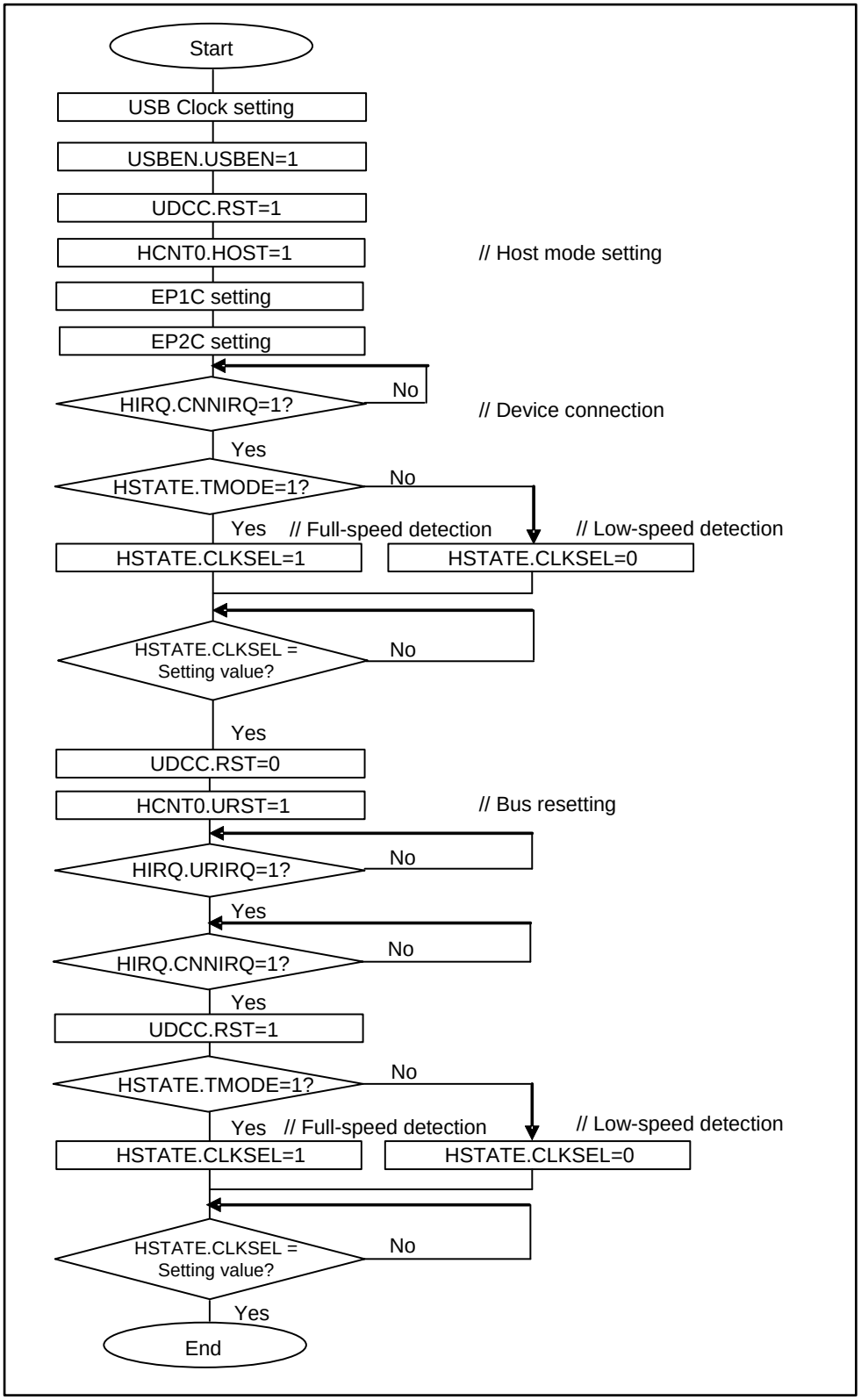
If both the host pins D+ and D- remain set to "LOW" for 2.5 μs or more regardless of the host or function mode, it is judged that the device has been disconnected. This then sets "0" to the CSTAT bit of the Host Status Register (HSTATE) and "1" to the DIRQ bit of the Host Interrupt Register (HIRQ). At this time, if the DIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs. To clear this interrupt, write "0" to the DIRQ bit of the Host Interrupt Register (HIRQ).

If the USB bus is reset, it is judged that the device has been disconnected. In this case, the CSTAT bit of the Host Status Register (HSTATE) is set to "0", but the DIRQ bit of the Host Interrupt Register (HIRQ) is not set to "1".

4. USB host setting procedure examples

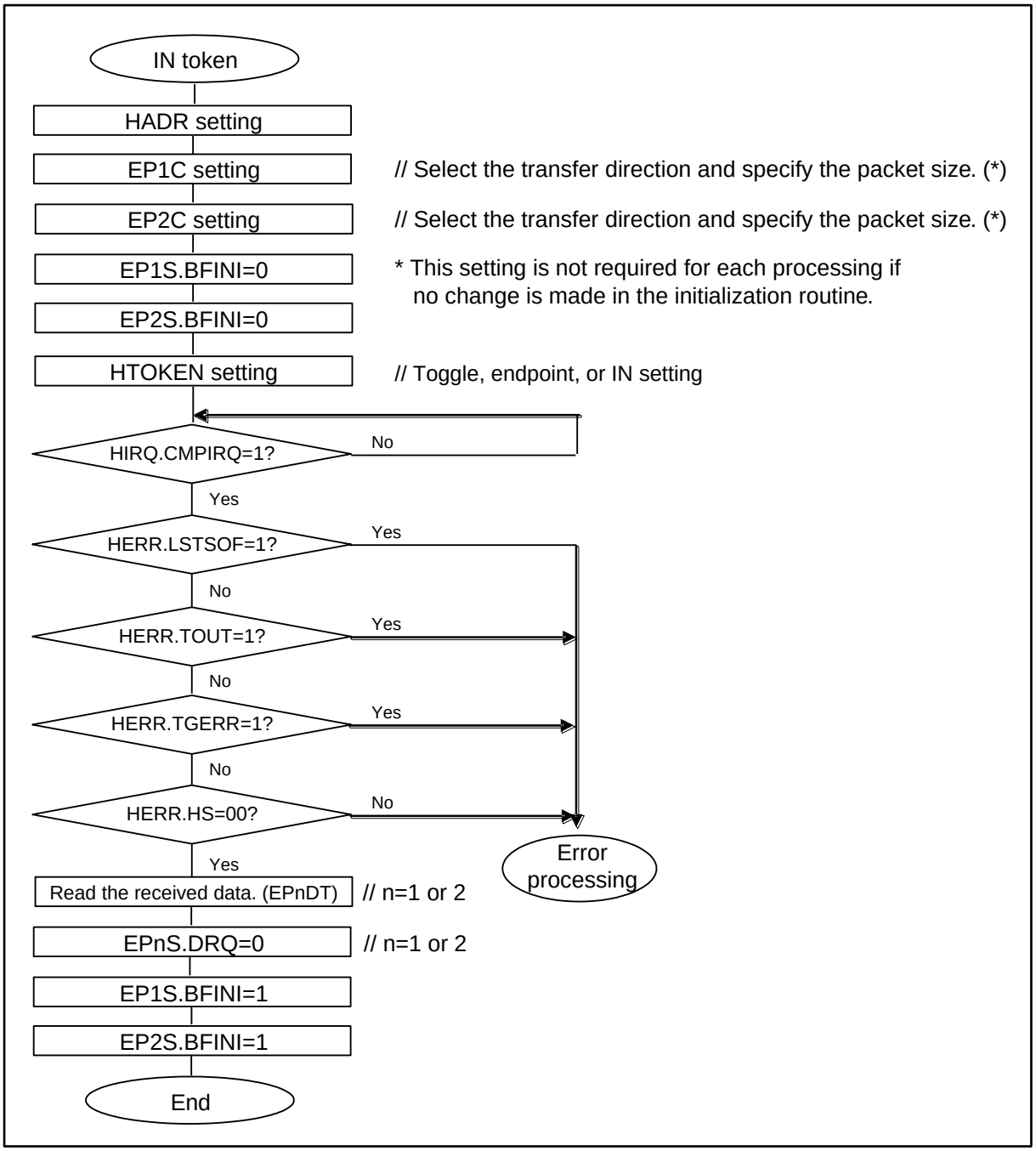
The following shows the flowchart for the USB host tokens.

Initialization and device detection

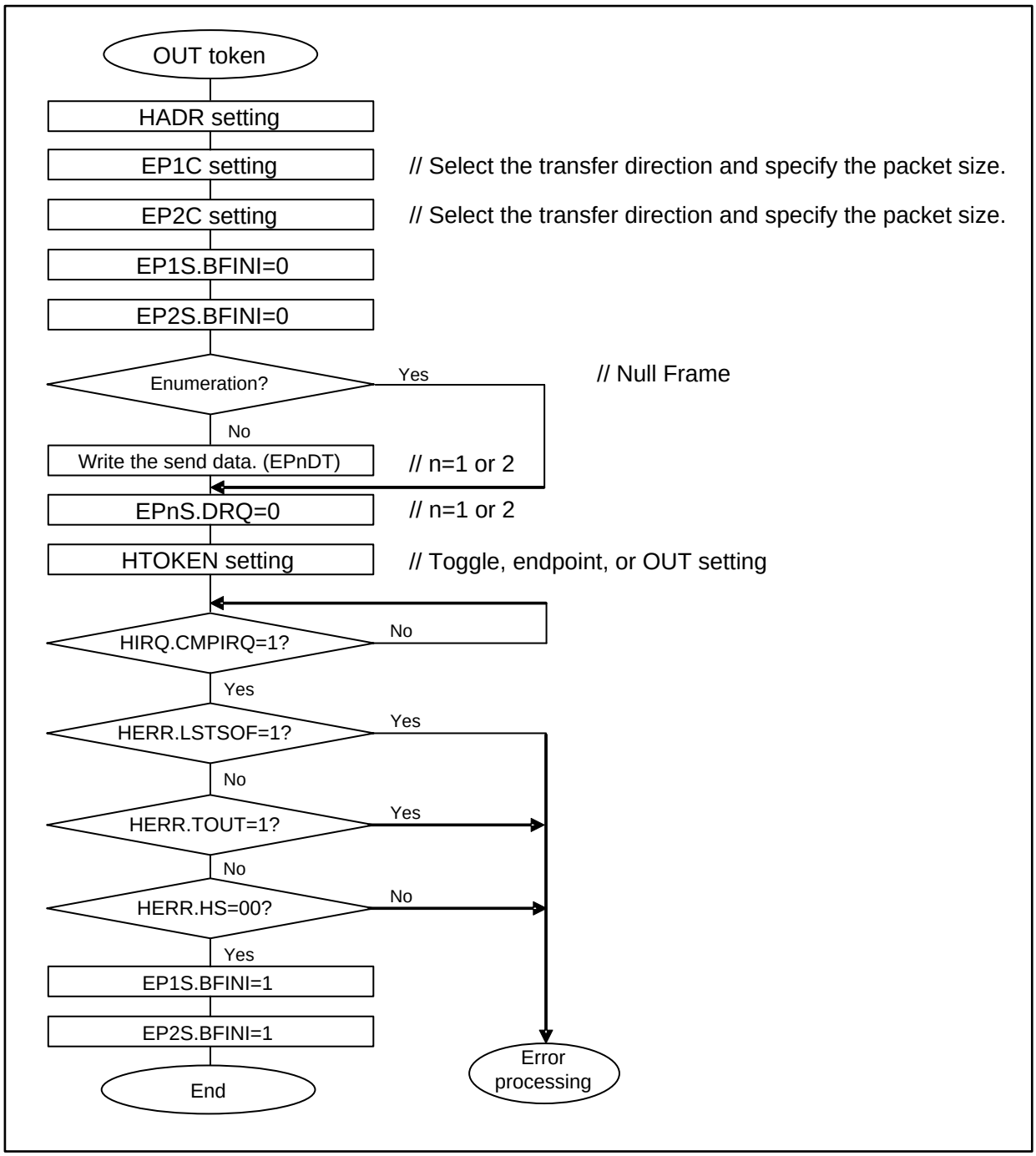


IN, OUT, or SETUP token

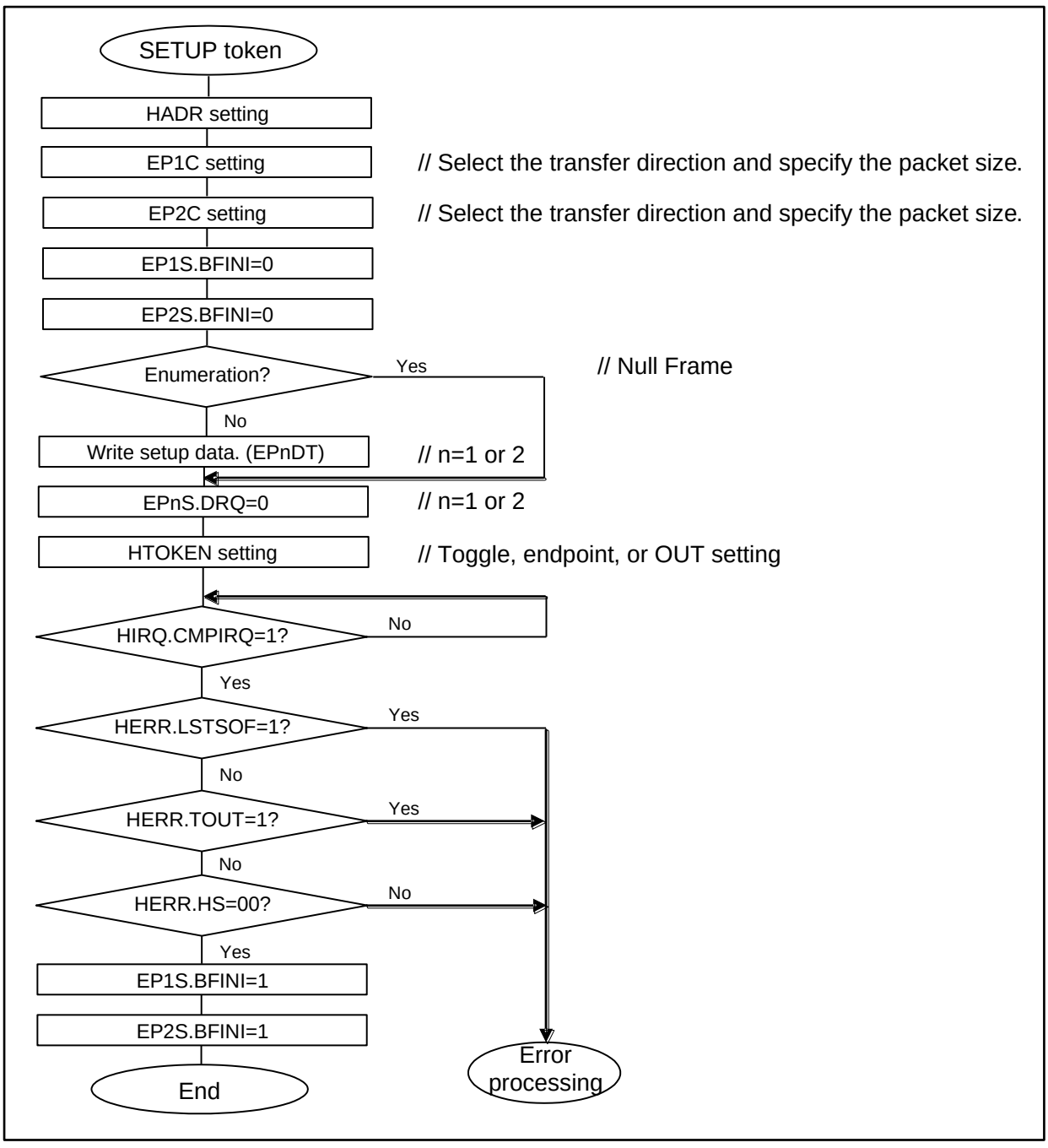
■ IN token



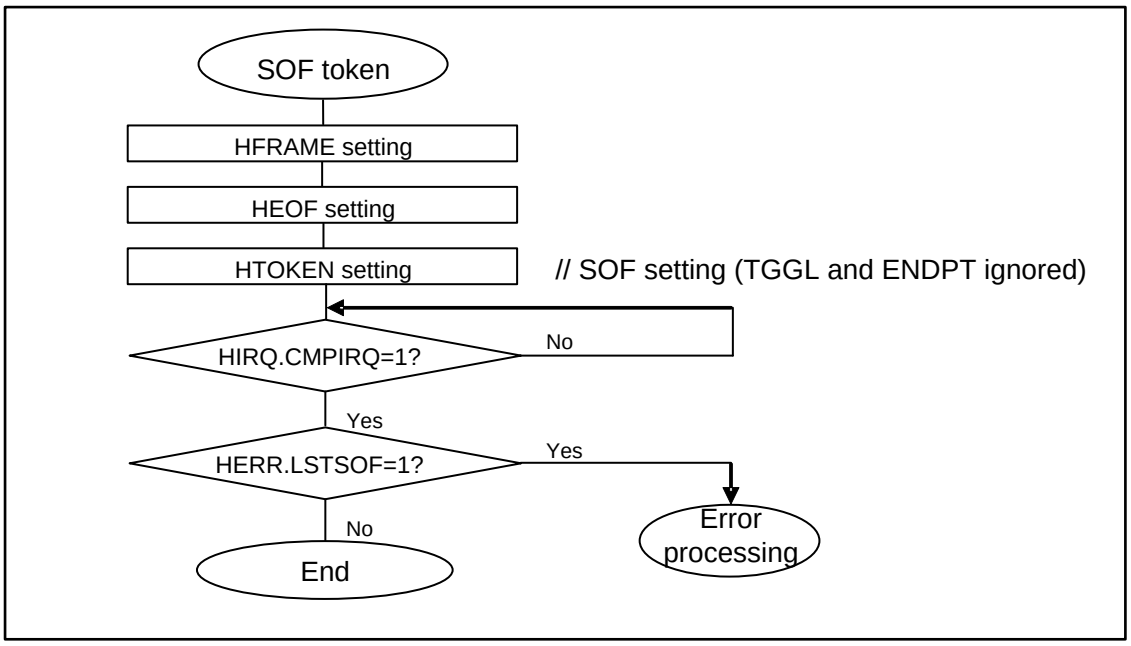
■ OUT token


OUT token
HADR setting
EP1C setting
// Select the transfer direction and specify the packet size.
EP2C setting
// Select the transfer direction and specify the packet size.
EP1S.BFINI=0
EP2S.BFINI=0
Enumeration?
Yes
// Null Frame
No
Write the send data. (EPnDT)
// n=1 or 2
EPnS.DRQ=0
// n=1 or 2
HTOKEN setting
// Toggle, endpoint, or OUT setting
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
HERR.TOUT=1?
Yes
No
HERR.HS=00?
No
Yes
EP1S.BFINI=1
EP2S.BFINI=1
End
Error processing

■ SETUP token


SETUP token
HADR setting
EP1C setting
// Select the transfer direction and specify the packet size.
EP2C setting
// Select the transfer direction and specify the packet size.
EP1S.BFINI=0
EP2S.BFINI=0
Enumeration?
Yes
// Null Frame
No
Write setup data. (EPnDT)
// n=1 or 2
EPnS.DRQ=0
// n=1 or 2
HTOKEN setting
// Toggle, endpoint, or OUT setting
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
HERR.TOUT=1?
Yes
No
HERR.HS=00?
No
Yes
EP1S.BFINI=1
EP2S.BFINI=1
End
Error processing

SOF token


SOF token
HFRAME setting
HEOF setting
HTOKEN setting
// SOF setting (TGGL and ENDPT ignored)
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
End
Error processing

5. USB host registers

This section explains the configurations and functions of the registers used for the USB host.

List of USB host registers

Abbreviation	Register name	Reference
UDCC	UDC Control Register	*
EP1C	EP1 Control Register	*
EP2C	EP2 Control Register	*
EP1S	EP1 Status Register	*
EP2S	EP2 Status Register	*
EP1DTH	EP0 Data Register high-order	*
EP1DTL	EP0 Data Register low-order	*
EP2DTH	EP0 Data Register high-order	*
EP2DTL	EP0 Data Register low-order	*
HCNT0	Host Control Register 0	5.1
HCNT1	Host Control Register 1	
HIRQ	Host Interrupt Register	5.2
HERR	Host Error Status Register	5.3
HSTATE	Host Status Register	5.4
HFCOMP	SOF Interrupt Frame Compare Register	5.5
HRTIMER	Retry Timer Setup Register	5.6
HADR	Host Address Register	5.7
HEOF	EOF Setup Register	5.8
HFRAME	Frame Setup Register	5.9
HTOKEN	Host Token Endpoint Register	5.10

* : See chapter "USB Function".

UDCC:RST dependent register bit update timing list

	Register	bit
Register bits to be updated when UDCC:RST=1	HCNT0	HOST
	HSTATE	CLKSEL
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
Register bits initialized when UDCC:RST=1 (Update when UDCC:RST=0)	HCNT0	URST
	HIRQ	TCAN, RWKIRQ, URIRQ, CMPIRQ, CNNIRQ, DIRQ, SOFIRQ
	HERR (All bits)	LSTSOF, RERR, TOUT, CRC, TGERR, STUFF, HS
	HSTATE	SOFBUSY, SUSP
	HFRAME	FRAME0, FRAME1
	HTOKEN (All bits)	TGGL, TKNEN, ENDPT
	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
Register bits unaffected by UDCC:RST	HCNT0	RWKIRE, URIRE, CMPIRE, CNNIRE, DIRE, SOFIRE
	HCNT1	SOFSTEP, CANCEL, RETRY
	HIRQ	CNNIRQ, DIRQ
	HFCOMP	HFRAMECOMP
	HSTATE	TMODE, CSTAT
	HRTIMER0, 1, 2	RTIMER0, 1, 2
	HADR	Address
	HEOF	EOF0, 1

5.1 Host Control Registers 0 and 1 (HCNT0 and HCNT1)

Host Control Registers 0 and 1 (HCNT0 and HCNT1) are used to specify the USB operation mode and interrupt.

Host Control Register 1 (HCNT1)

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	SOFSTEP	CANCEL	RETRY
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1
Reset enabled or not*	x	x	x	x	x	x	x	x

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. □: To be reset.

Host Control Register 0 (HCNT0)

bit	7	6	5	4	3	2	1	0
Field	RWKIRE	URIRE	CMPIRE	CNNIRE	DIRE	SOFIRE	URST	HOST
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Reset enabled or not*	x	x	x	x	x	x	❍	x

* : Enables or disables a reset with the RST bit of UDCC. X : Not to be reset. ❍□: To be reset.

[bit15:11] Reserved: Reserved bits

Always set it to "0".

[bit10] SOFSTEP (SOF STEP) SOF interrupt occurrence selection bit

This is a SOF interrupt occurrence selection bit.

If this bit is set to "1", the SOF interrupt flag (HIRQ:SOFIRQ) is set to "1" each time SOF is sent.

If this bit is set to "0", the set value of the SOF Interrupt Frame Compare Register (HFCOMP) is compared with the low-order eight bits of the SOF frame number. If they match, the SOF interrupt flag (HIRQ:SOFIRQ) is set to "1".

bit	Description
0	An interrupt occurred due to the HFCOMP setting.
1	An interrupt occurred.

Notes:

- *If a SOF token (TKNEN="001") is sent by the setting of the Host Token Endpoint Register (HTOKEN), the SOF interrupt flag (HIRQ:SOFIRQ) is not set to "1" regardless of the setting of this bit.*
- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit9] CANCEL (token CANCEL enable) token cancellation enable bit

This is a token cancellation enable bit.

When "1" is set to this bit, if the target token is written to the Host Token Endpoint Register (HTOKEN) in the EOF area (specified in the EOF Setting Register), its sending is canceled. When "0" is set to this bit, token sending is not canceled even if the target token is written to the register. The cancellation of token sending is detected by reading the TCAN bit of the Host Interrupt Register (HIRQ).

bit	Description
0	Continues a token.
1	Cancels a token.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit8] RETRY (RETRY enable) retry enable bit

this is a retry enable bit.

If this bit is set to "1", the target token is retried if a NAK or error* occurs. Retry processing is performed during the time that is specified in the Retry Timer Setting Register (HRTIMER).

* : HERR:RERR="1", HERR:TOUT="1", HERR:CRC="1", HERR:TGERR="1", HERR:STUFF="1"

bit	Description
0	Does not retry token sending.
1	Retries token sending.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit7] RWKIRE (Remove WaKe up Interrupt Request Enable) resume interrupt enable bit

This is a resume interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the RWKIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the RWIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

bit	Description
0	Disables an interrupt after restarting.
1	Enables an interrupt after restarting.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit6] URIRE (Usb bus Rest Interrupt Request Enable) bus reset interrupt enable bit

This is a bus reset interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the URIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the URIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

bit	Description
0	Disables an interrupt after resetting the USB bus.
1	Enables an interrupt after resetting the USB bus.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit5] CMPIRE (CoMPletion Interrupt Request Enable) token completion interrupt enable bit

This is a token completion interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

bit	Description
0	Disables an interrupt at completion.
1	Enables an interrupt at completion.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit4] CNNIRE (CoNNection Interrupt Request Enable) device connection detection interrupt enable bit

This is a device connection detection interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the CNNIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the CNNIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

bit	Description
0	Disables an interrupt at device connection.
1	Enables an interrupt at device connection.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit3] DIRE (Disconnection Interrupt Request Enable) device disconnection detection interrupt enable bit

This is a device disconnection detection interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the DIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the DIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

bit	Description
0	Disables an interrupt at device disconnection.
1	Enables an interrupt at device disconnection.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit2] SOFIRE (Start Of Frame Interrupt Request Enable) SOF interrupt enable bit

This is a SOF interrupt enable bit.

When "1" is set to this bit, an interrupt occurs if the SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". When "0" is set to this bit, an interrupt does not occur even if the SOFIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

bit	Description
0	Disables an interrupt when sending SOF.
1	Enables an interrupt when sending SOF.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit1] URST (Usb bus ReSeT) bus reset bit

This is a bus reset bit.

When "1" is set to this bit, the USB bus is reset. This bit continues to be "1" during USB bus resetting, and changes to "0" when USB bus resetting is ended. If "0" is set to this bit, no processing is performed.

bit	Description
0	Holds the status of the USB bus.
1	Resets the USB bus.

Notes:

- *No processing is performed even if this bit is set to "1" while the RST bit of the UDC Control Register (UDCC) is "1".*
- *This bit is not allowed to be set to "1" while the SUSP bit of the Host Status Register (HSTATE) is "1" or during token sending.*
- *The Host Control Register (HCNT0 or HCNT1) is not allowed to be written while this bit is "1".*

[bit0] HOST (HOST mode) host mode bit

This is a host mode bit.

When "1" is set to this bit, the USB acts as a host. When "0" is set to this bit, the USB acts as a function.

bit	Description
0	Function mode
1	Host mode

Notes:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *Change the value of this bit while the RST bit of the UDC Control Register (UDCC) is "1".*
- *The operation mode does not transition to the required one immediately after it was changed using this bit. Read this bit to check that the operation mode has changed.*
- *Before changing from the host mode to the function mode, check that the following conditions are satisfied and also set "1" to the RST bit of the UDC Control Register (UDCC).*
- *The SOFBUSY bit of the Host Status Register (HSTATE) is set to "0".*
- *The TKNEN bits of the Host Token Endpoint Register (HTOKEN) are set to "000".*
- *The SUSP bit of the Host Status Register (HSTATE) is set to "0".*
- *Before changing from the function mode to the host mode, set "1" to the HCONX bit of the UDC Control Register (UDCC), and disconnect the host or HUB.*

5.2 Host Interrupt Register (HIRQ)

The Host Interrupt Register (HIRQ) indicates the USB host interrupt request flags. A host interrupt can occur by setting the interrupt enable bit of the Host Control Register (HCNT0 or HCNT1), excluding the TCAN bit.

Be sure byte access to host interrupt register. (HIRQ).

bit	7	6	5	4	3	2	1	0
Field	TCAN	Reserved	RWKIRQ	URIRQ	CMPIRQ	CNNIRQ	DIRQ	SOFIRQ
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Reset enabled or not*	□	□	□	□	□	x	x	❍

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

[bit7] TCAN (Token CANcel flag) token cancellation flag

This is a token cancellation flag.

If this bit is set to "1", it means that token sending is canceled based on the setting of the CANCEL bit of Host Control Register 1 (HCNT1). When this bit is "0", it means that token sending is not canceled. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

A read-modify-write access reads the bit as "1".

bit	Description
0	Token has not been canceled.
1	Token has been canceled.

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *No interrupt occurs even if this bit is set. To cancel this with interrupt processing, check that token sending is canceled during SOF interrupt processing.*

[bit6] Reserved: Reserved bit

Always set it to "0".

[bit5] RWKIRQ (Remove WaKe up Interrupt ReQuest) remote Wake-up end flag

This is a remote Wake-up end flag.

If this bit is set to "1", it means that remote Wake-up is ended. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the RWKIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

A read-modify-write access reads the bit as "1".

bit	Description
0	Issues no interrupt request by restart.
1	Issues an interrupt request by restart.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit4] URIRQ (Usb bus Reset Interrupt ReQuest) bus reset end flag

This is a bus reset end flag.

If this bit is set to "1", it means that USB bus resetting is ended. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the URIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

A read-modify-write access reads the bit as "1".

bit	Description
0	Issues no interrupt request by USB bus resetting.
1	Issues an interrupt request by USB bus resetting.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit3] CMPIRQ (CoMPletion Interrupt ReQuest) token completion flag

This is a token completion flag.

If this bit is set to "1", it means that a token is completed. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

A read-modify-write access reads the bit as "1".

bit	Description
0	Issues no interrupt request by token completion.
1	Issues an interrupt request by token completion.

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *This bit is not set to "1" even if the TCAN bit of the Host Interrupt Register (HIRQ) changes to "1".*
- *Take the following steps when this bit is set to “1” by finishing IN token or Isochronous IN token.*
  1) Read HS bit of Host Error Status Register (HERR), then set CMPIRQ bit to “0”.
  2) Set DRQIE bit of EPn Status Register (EPnS) (n=1 or 2) to “1” if HS bit of Host Error Status Register (HERR) is equal to “00” and wait until DRQ bit changes to “1”.
  Finish the IN token processing if HS bit is not equal to “00”.
  3) Read the received data if DRQ bit of EPn Status Register (EPnS) changes to “1”.

[bit2] CNNIRQ (CoNNection Interrupt ReQuest) device connection detection flag

This is a device connection detection flag.

If this bit is set to "1", it means that a device connection is detected. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the CNNIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

A read-modify-write access reads the bit as "1".

bit	Description
0	Issues no interrupt request by detecting a device connection.
1	Issues an interrupt request by detecting a device connection.

Notes:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *A device connection is also detected in the function mode.*

[bit1] DIRQ (Disconnection Interrupt ReQuest) device disconnection detection flag

This is a device disconnection detection flag.

If this bit is set to "1", it means that a device disconnection is detected. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the DIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

A read-modify-write access reads the bit as "1".

bit	Description
0	Issues no interrupt request by detecting a device disconnection.
1	Issues an interrupt request by detecting a device disconnection.

Notes:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *A device disconnection is also detected in the function mode.*

[bit0] SOFIRQ (Start Of Frame Interrupt ReQuest) SOF starting flag

This is a SOF starting flag.

If this bit is set to "1", it means that SOF token sending is started. When this bit is "0", it has no meaning. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

When the SOFIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs if this bit is set to "1".

A read-modify-write access reads the bit as "1".

bit	Description
0	Issues no interrupt request by starting a SOF token.
1	Issues an interrupt request by starting a SOF token.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

5.3 Host Error Status Register (HERR)

The Host Error Status Register (HERR) indicates whether or not an error occurs while sending or receiving data in the host mode.

Host Error Status Register (HERR) should be accessed with a byte access instruction.

bit	15	14	13	12	11	10	9	8
Field	LSTSOF	RERR	TOUT	CRC	TGERR	STUFF	HS	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	11	
Reset enabled or not*	❍	❍	❍	❍	❍	❍	❍	

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

[bit15] LSTSOF (LoST SOF) lost SOF flag

This is a lost SOF flag.

If this bit is set to "1", it means that the SOF token could not be sent in the host mode because other token is in process. When this bit is "0", it means that no lost SOF error is detected. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

bit	Description
0	SOF has been sent.
1	SOF sending error

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit14] RERR (Receive Error) receive error flag

This is a receive error flag.

When this bit is set to "1", it means that the received data exceeds the specified maximum number of packets in the host mode. If a receive error is detected, bit13 (TOUT) of this register is also set to "1". When this bit is "0", it means that no error occurs. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

bit	Description
0	No receive error has occurred.
1	Maximum packet receive error has occurred.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit13] TOUT (Time OUT) timeout flag

This is a timeout flag.

If this bit is set to "1", it means that no response is returned to a token from the device within the specified time after the token has been sent in host mode. When this bit is "0", it means that no timeout is detected. When this bit is "0", it means that no error occurs. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

bit	Description
0	No timeout has occurred.
1	Timeout has occurred.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit12] CRC (CRC error) CRC error flag

This is a CRC error flag.

If this bit is set to "1", it means that a CRC error is detected in the host mode. When this bit is "0", it means that no CRC error is detected. If a CRC error is detected, bit13 (TOUT) of this register is also set to "1". When this bit is "0", it means that no CRC error is detected. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

bit	Description
0	No CRC error has occurred.
1	CRC error has occurred.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit11] TGERR (ToGgle ERRor) toggle error flag

This is a toggle error flag.

If this bit is set to "1", it means that the data of this bit does not match the value of the received toggle data. When this bit is "0", it means that no toggle error is detected. If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

bit	Description
0	No toggle error has occurred.
1	Toggle error has occurred.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit10] STUFF (STUFFing error) stuffing error flag

This is a stuffing error flag.

If this bit is set to "1", it means that a bit stuffing error is detected. When this bit is "0", it means that no stuffing error is detected. If a stuffing error is detected, bit13 (TOUT) of this register is also set to "1". If this bit is written with "0", it is set to "0". However, if this bit is written with "1", its value is ignored.

bit	Description
0	No stuffing error has occurred.
1	Stuffing error has occurred.

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

[bit9:8] HS (Hand Shake status) handshake status flags

These are handshake status flags.

These flags indicate the status of a handshake packet to be sent or received.

These flags are set to "NULL" when no handshake occurs due to an error or when a SOF token has been ended with the TKNEN bits of the Host Token Endpoint Register (HTOKEN).

These bits are updated when sending or receiving has been ended.

Table 5.3-1 Handshake

bit9	bit8	Handshake
0	0	ACK
0	1	NAK
1	0	STALL
1	1	NULL

Note:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*

5.4 Host Status Register (HSTATE)

The Host Status Register (HSTATE) indicates the state of the USB circuit such as a device connection or transfer mode. Note that the setting of the CLKSEL bit is also effective in the function mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	ALIVE	CLKSEL	SOFBUSY	SUSP	TMODE	CSTAT
Attribute	-		R/W	R/W	R/W	R/W	R	R
Initial value	X		0	1	0	0	1	0
Reset enabled or not*	-		x	x	❍	❍	x	x

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

[bit7:6] Reserved: Reserved bits

The values of these bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit5] ALIVE (keep-ALIVE)

This bit is used to specify the keep-alive function in the low-speed mode. If this bit it set to "1" while the CLKSEL bit of the Host Status Register (HSTATE) is "0", SE0 is output instead of SOF. This bit is effective when the CLKSEL bit of the Host Status Register (HSTATE) is "0". If the CLKSEL bit is "1", SOF is output regardless of the setting of the ALIVE bit.

bit	Description
0	SOF output
1	SE0 output (Keep-alive)

[bit4] CLKSEL (CLocK SELect) USB operation clock selection bit

This is a USB operation clock selection bit.

bit	Description
0	Low-speed clock
1	Full-speed clock

Notes:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *Change the value of this bit while the RST bit of the UDC Control Register (UDCC) is "1".*
- *This bit must always be set to "1". It must not be set to "0".*
- *Use the on-chip bus (HCLK) clock with 13 MHz or more.*

[bit3] SOFBUSY (SOF BUSY) SOF busy flag

This is a SOF busy flag.

When a SOF token is sent using the Host Token Endpoint Register (HTOKEN), this bit is set to "1", which means that the SOF timer is active. When this bit is "0", it means that the SOF timer is under suspension. To stop the active SOF timer, write "0" to this bit. However, if this bit is written with "1", its value is ignored.

bit	Description
0	The SOF timer is stopped.
1	The SOF timer is active.

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *The SOF timer does not stop immediately after "0" has been set to this bit to stop the SOF timer. To check whether or not the SOF timer is stopped, read this bit.*

[bit2] SUSP (SUSPend) suspend setting bit

This is a suspend setting bit.

If this bit is set to "1", the USB circuit is placed into the suspend state. If this bit is set to "0" while it is "1" or the USB bus is placed into the k-state mode, the suspend state is released, and the RWIRQ bit of the Host Interrupt Register (HIRQ) is set to "1".

Table 5.4-1 Suspend setting

bit	Operation
Set to "1".	Suspend
Set "0" while this bit is "1".	Resume
Others	Holds the state.

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *Do not set this bit to "1" while the USB is active (during USB bus resetting, data transfer, or SOF timer running).*
- *USB clock must not be stopped in the suspend state.*
- *If the value of this bit is changed, it is not immediately reflected on the state of the USB bus. To check whether or not the state is updated, read this bit.*

[bit1] TMODE (Transmission MODE) transmission mode flag

This is a transmission mode flag.

If this bit is "1", it means that the device is connected in the full-speed mode. When this bit is "0", it means that the device is connected in the low-speed mode. This bit is valid when the CSTAT bit of the Host Status Register (HSTATE) is "1".

bit	Description
0	Low Speed
1	Full Speed

Notes:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *Use the base clock (HCLK) with 13 MHz or more.*

[bit0] CSTAT (Connect STATus) connection status flag

This is a connection status flag.

When this bit is "1", it means that the device is connected. When this bit is "0", it means that the device is disconnected.

bit	Description
0	Device is disconnected.
1	Device is connected.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

5.5 SOF Interrupt Frame Compare Register (HFCOMP)

The SOF Interrupt Frame Compare Register (HFCOMP) is used to specify the data to be compared with the low-order eight bits of a frame number when sending a SOF token. When the SOFSTEP bit of Host Control Register 0 (HCNT0) is "0", the value of this register is compared with that of the low-order eight bits of a frame number. If they match, the SOFIRQ bit of the Host interrupt Register (HIRQ) is set to "1" when starting SOF sending. When the SOFIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs.

bit	15	14	13	12	11	10	9	8
Field	FRAMECOMP							
Attribute	R/W							
Initial value	00000000							
Reset enabled or not*	x							

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. □: To be reset.

[bit15:8] FRAMECOMP : frame compare data

These are frame compare data.

These bits are used to specify the data to be compared with the low-order eight bits of a frame number when sending a SOF token.

If the SOFSTEP bit of Host Control Register 0 (HCNT0) is "0", the frame number of SOF is compared with the value of this register when sending a SOF token. If they match, "1" is set to the SOFIRQ bit of the Host Interrupt Register (HIRQ).

The setting of this register is invalid when the SOFSTEP bit of Host Control Register 0 (HCNT0) is "0".

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

5.6 Retry Timer Setup Register (HRTIMER)

The Retry Timer Setup Register (HRTIMER) is used to specify the token retry time.

bit	15	14	13	12	11	10	9	8
Field	RTIMER1							
Attribute	R/W							
Initial value	00000000							
Reset enabled or not*	x							

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. □: To be reset.

bit	7	6	5	4	3	2	1	0
Field	RTIMER0							
Attribute	R/W							
Initial value	00000000							
Reset enabled or not*	x							

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. □: To be reset.

bit	7(23)	6(22)	5(21)	4(20)	3(19)	2(18)	1(17)	0(16)
Field	Reserved						RTIMER2	
Attribute	-						R/W	
Initial value	X						00	
Reset enabled or not*	-						x	

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. □: To be reset.

[bit23:18] Reserved: Reserved bits

The values of these bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit17:0] HRTIMER0, 1, 2 : Retry timer setting bits

These are retry timer setting bits.

These bits are used to specify the retry time in this register. The retry timer is activated when token sending starts while the RETRY bit of Host Control Register 1 (HCNT1) is "1". The retry time is then decremented by one when a 1-bit transfer clock (12 MHz in the full-speed mode) is output. When the retry timer reaches "0", the target token is sent, and processing is ended.

If a token retry occurs in the EOF area, the retry timer is stopped until SOF sending is ended. After SOF sending has been completed, the retry timer restarts with the value that is set when the timer stopped.

Notes:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC). If data is written while the RST bit of the UDC Control Register (UDCC) is "1", the written data is ignored.*
- *Write this register in the host mode. bit15 to bit0 of this register are set to "0" in the function mode. Even if data is written to bit15 to bit0 of this register, it is ignored.*

5.7 Host Address Register (HADR)

The Host Address Register (HADR) is used as an address field to send a token.

bit	15	14	13	12	11	10	9	8
Field	Reserved	Address						
Attribute	-	R/W						
Initial value	X	0000000						
Reset enabled or not*	-	x						

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. □: To be reset.

[bit15] Reserved: Reserved bit

The values of this bit is undefined in read mode. Even if "0" or "1" is written to this bit, it has no effect on LSI operations.

[bit14:8] Address : address bits

These are address bits.
These bits are used to specify a token address.

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

5.8 EOF Setup Register (HEOF)

The EOF Setup Register (HEOF) is used to specify the token disable time before sending a SOF token. If both the following conditions are satisfied, a request token is sent after a SOF token has been transferred.

- When the value of the SOF timer is compared with that of this register, it is less than the value of this register.

- An IN, OUT, or SETUP token sending request has been issued.

This is a function to prevent a SOF token generated by hardware from being sent together with other tokens. The time unit of this register is the 1-bit transfer time.

bit	15	14	13	12	11	10	9	8
Field	Reserved		EOF1					
Attribute	-		R/W					
Initial value	X		000000					
Reset enabled or not*	-		x					

* : Enables or disables a reset with the RST bit of UDCC. x: Not to be reset. ❍ : To be reset.

bit	7	6	5	4	3	2	1	0
Field	EOF0							
Attribute	R/W							
Initial value	00000000							
Reset enabled or not*	x							

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

[bit15:14] Reserved: Reserved bits

The values of these bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit13:0] EOF1, EOF0 (End Of Frame) : EOF bits

These are EOF bits.

These bits are used to specify the time to disable token sending before transferring SOF. Specify the time with a margin, which is longer than the one-packet length. The time unit is the 1-bit transfer time.

Setting example: MAXPKT = 64 bytes, full-speed mode
(Token_length + packet_length + header + CRC)×7/6 + Turn_around_time
= (34 bit + 546 bit)×7/6 + 36 bit = 712.7 bit
Therefore, set "0x2C9".

Note:

- *This bit is not initialized even if "1" is set to the RST bit of the UDC Control Register (UDCC).*

5.9 Frame Setup Register (HFRAME)

The Frame Setup Register (HFRAME) is used to specify a frame number when sending a SOF token. If SOF sending is set to the TKNEN bit of the Host Token Endpoint Register (HTOKEN), the SOF timer is activated. After this, SOF is sent automatically every 1 ms. The Frame Setup Register is automatically incremented by one each time SOF is ended.

bit	15	14	13	12	11	10	9	8
Field	Reserved					FRAME1		
Attribute	-					R/W		
Initial value	X					000		
Reset enabled or not*	-					❍		

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

bit	7	6	5	4	3	2	1	0
Field	FRAME0							
Attribute	R/W							
Initial value	00000000							
Reset enabled or not*	❍							

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

[bit15:11] Reserved: Reserved bits

The values of these bits are undefined in read mode. Even if "0" or "1" is written to these bits, it has no effect on LSI operations.

[bit10:0] FRAME1, FRAME0 : frame setting bits

These are frame setting bits.
These bits are used to specify a frame number of SOF.

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *Specify a frame number in this register before setting SOF in the TKNEN bit of the Host Token Endpoint Register (HTOKEN).*
- *This register is not allowed to be written while the SOFBUSY bit of the Host Status Register (HSTATE) is "1" and a SOF token is in process.*

5.10 Host Token Endpoint Register (HTOKEN)

The Host Token Endpoint Register (HTOKEN) is used to specify toggle, endpoint, and token.

bit	7	6	5	4	3	2	1	0
Field	TGGL	TKNEN			ENDPT			
Attribute	R/W	R/W			R/W			
Initial value	0	000			0000			
Reset enabled or not*	❍	❍			❍			

* : Enables or disables a reset with the RST bit of UDCC. x : Not to be reset. ❍ : To be reset.

[bit7] TGGL (ToGGLe) toggle bit

This is a toggle bit.

This bit is used to set toggle data. Toggle data is sent depending on the setting of this bit. When receiving toggle data, received toggle data is compared with the toggle data indicated by this bit to verify whether or not an error occurs.

bit	Description
0	DATA0
1	DATA1

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *Set this bit when the TKNEN bits of the Host Token Endpoint Register (HTOKEN) are "000".*

[bit6:4] TKNEN (ToKeN ENable) token enable bits

These are token enable bits.

These bits send a token according to the settings. After operation has been ended, the TKNEN bits are set to "000", and the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to "1". If the CMPIRE bit of Host Control Register 0 (HCNT0) is "1", an interrupt occurs.

The settings of the TGGL and ENDPT bits are ignored when sending a SOF token.

Table 5-1 Token setting

bit6	bit5	bit4	Operation
0	0	0	Sends no data.
0	0	1	Sends SETUP token.
0	1	0	Sends IN token.
0	1	1	Sends OUT token.
1	0	0	Sends SOF token.
1	0	1	Sends Isochronous IN.
1	1	0	Sends Isochronous OUT.
1	1	1	Reserved (Setting disabled)

Notes:

- *This bit is set to the initial value when "1" is set to the RST bit of the UDC Control Register (UDCC).*
- *The PRE packet is not supported.*
- *Do not set "100" to the TKNEN bit when the SOFBUSY bit of the Host Status Register (HSTATE) is "1".*
- *Change the USB to the host mode before writing data to this bit.*
- *When issuing a token again after the token interrupt flag (CMPIRQ) has been set to "1", wait for 3 cycles or more after a USB transfer clock (12 MHz in the full-speed mode, 1.5 MHz in the low-speed mode) was output, then write data to this bit.*
- *When the device is disconnected (CSTAT of HSTATE = "0"), token sending is not performed even if data is written to this bit.*
- *Read the value of TKNEN bit if a new value is written in it .Continue writing in this bit until a retrieved value equals a new value written in During this checking process, it is needed to prevent any interrupt.*
- *Take the following steps when CMPIRQ bit of Host Interrupt Register (HIRQ) is set to "1" by finishing IN token or Isochronous IN token.*
  1) Read HS bit of Host Error Status Register (HERR), then set CMPIRQ bit to "0".
  2) Set DRQIE bit of EPn Status Register (EPnS) (n=1 or 2) to "1" if HS bit of Host Error Status Register (HERR) is equal to "00" and wait until DRQ bit changes to "1".
  Finish the IN token processing if HS bit is not equal to "00".
  3) Read the received data if DRQ bit of EPn Status Register (EPnS) changes to "1".

[bit3:0] ENDPT (ENDPoinT) endpoint bits

These are endpoint bits.

These bits are used to specify an endpoint to send or receive data to or from the device.

Note:

- *This bit is initialized when "1" is set to the RST bit of the UDC Control Register (UDCC).*

CHAPTER 4: Ethernet

For Ethernet, refer to the separate "PERIPHERAL MANUAL Ethernet Part".

CODE: 9xETHERTOP-E01.1

CHAPTER 5-1: CAN Prescaler

This chapter explains the CAN prescaler.

1. Overview and configuration
2. CAN Prescaler Register

CODE: 9BFCANPRE-E01.5

1. Overview and configuration

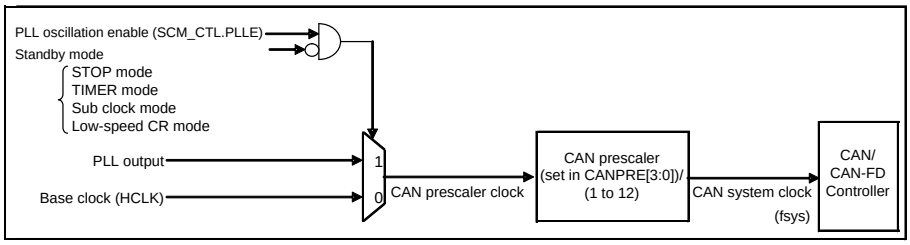
The CAN prescaler generates a CAN system clock (fsys) and supplies it to the CAN/CAN-FD.

The CAN prescaler divides a CAN prescaler clock by 1 to 12, and supplies it to the CAN as a CAN/CAN-FD system clock (fsys).

Figure 1-1 shows the block diagram of the CAN prescaler.

CAN block diagram

Figure 1-1 Generating a CAN system clock (fsys)



Explanation of Operations

The CAN prescaler selects the following as a CAN prescaler clock, and supplies it to the CAN/CAN-FD after frequency dividing.

- For PLL: PLL output
- For others (including Standby mode in Figure 1-1): Base clock (HCLK)

2. CAN Prescaler Register

This chapter describes the CAN Prescaler Register.

Abbreviation	Register name	Reference
CANPRE	CAN Prescaler Register	2.1

2.1 CAN Prescaler Register (CANPRE)

The CAN Prescaler Register is used to configure the CAN system clock (fsys) generation prescaler.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved			CANPRE			
Attribute	-	-			R/W			
Initial value	0	000			1011			

Register functions

[bit7] Reserved: Reserved bit

Be sure to write "0".

[bit6:4] Reserved: Reserved bits

Logical 0 is always read. In the write mode, set "0".

[bit3:0] CANPRE: CAN prescaler setting bits

These bits are used to specify a divided CAN prescaler. The divided clock is supplied as CAN/CAN-FD system clock to CAN macro.

bit3:0	Description
0000	CAN prescaler clock is not divided.
0001	CAN prescaler clock is divided to 1/2.
001x	CAN prescaler clock is divided to 1/4.
01xx	CAN prescaler clock is divided to 1/8.
1000	CAN prescaler clock is divided to 2/3. The clock duty is 67%.
1001	CAN prescaler clock is divided to 1/3.
1010	CAN prescaler clock is divided to 1/6.
1011	CAN prescaler clock is divided to 1/12. [Initial value]
110x	CAN prescaler clock is divided to 1/5.
111x	CAN prescaler clock is divided to 1/10.

Notes:

- *Before changing the value of the CAN prescaler setting bit, set the initialization bit (Init) of the CAN Control Register (CTRLR) to "1", and stop all bus operations.*
- *To use the PLL output as a CAN prescaler clock, set the initialization bit (Init) of the CAN Control Register (CTRLR) to "0" after PLL oscillation has been stabilized.*
- *TYPE1-M4, TYPE2-M4 products*
  Make sure that the CAN system clock output by the CAN prescaler is 16 MHz or less.
- *TYPE3-M4 product*
  Make sure that the CAN system clock output by the CAN prescaler is 40MHz or less.

CHAPTER 5-2: CAN Controller

This chapter explains CAN.

1. Overview
2. Configuration
3. CAN Controller Operations
4. CAN Registers
5. Notes

CODE: FC42L-E02.6

1. Overview

The CAN controller complies with CAN protocol version 2.0A/B, a standard protocol for serial communication. CAN is widely used in various industrial fields such as automobile and factory automation.

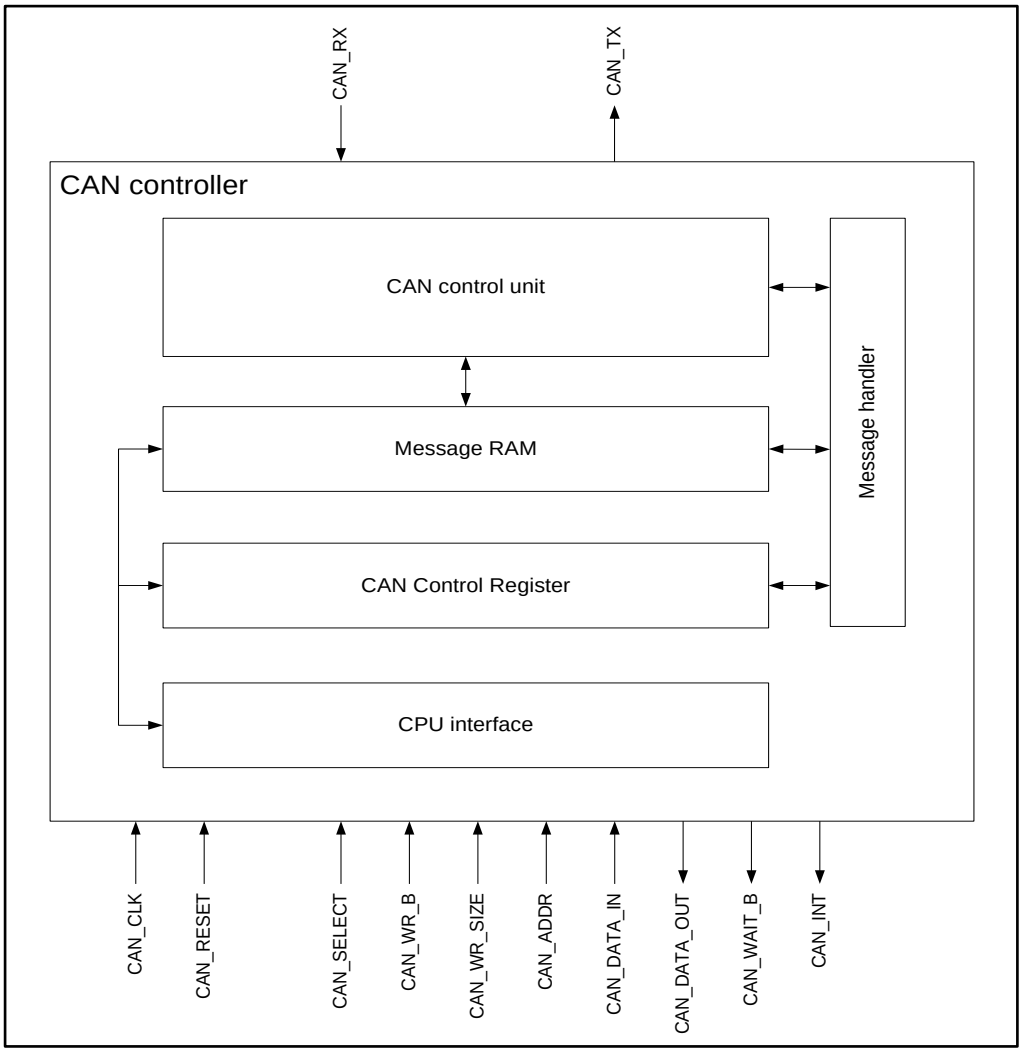
The CAN controller has the following features:

- Supports CAN protocol version 2.0A/B
- Supports a bit rate up to 1 Mbit/s
- Identifier mask for each message object
- Supports programmable FIFO mode
- Maskable interrupt
- Supports 32 message buffers
- Supports programmable loop-back mode for self-test operation
- Read and write from/to the message buffer using interface registers

2. Configuration

Figure 2-1 shows the block diagram of the CAN controller.

Figure 2-1 CAN controller block diagram



- CAN control unit
  Controls the CAN protocol and the serial registers for serial/parallel conversion to transfer send/receive messages.
- Message RAM
  Stores message objects
- Registers
  All registers used by CAN.
- Message handler
  Controls the message RAM and CAN control unit.
- CPU interface
  Controls the internal bus interface.

3. CAN Controller Operations

This section explains the operations and functions of the CAN controller.

Following functions are included:

3.1. Message objects

3.2. Message transmission

3.3. Message reception

3.4. FIFO buffer function

3.5. Interrupt function

3.6. Bit timing

3.7. Test mode

3.8. Software initialization

3.1 Message objects

The following explains message objects and the interface of the message RAM.

Message objects

The configuration of message objects in the message RAM (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits) is not initialized by a hardware reset. Initialize the message objects by the CPU, or set the MsgVal bit to disable (MsgVal = "0"). Configure the CAN Bit Timing Register (BTR) while the Init bit in the CAN Control Register (CTRLR) is "1".

To configure message objects, set the message interface registers (the IFx Mask Register, IFx Arbitration Register, IFx Message Control Register, and IFx Data Register), and then write a message number to the corresponding IFx Command Request Register. By writing the message number, the interface register data will be transferred to the addressed message object.

When the Init bit in the CAN Control Register is cleared to "0", the CAN controller starts operation. The received data that have passed acceptance filtering are stored into the message RAM. Messages with pending transmission requests are transferred from the message RAM to the shift register in the CAN controller, and then sent to the CAN bus.

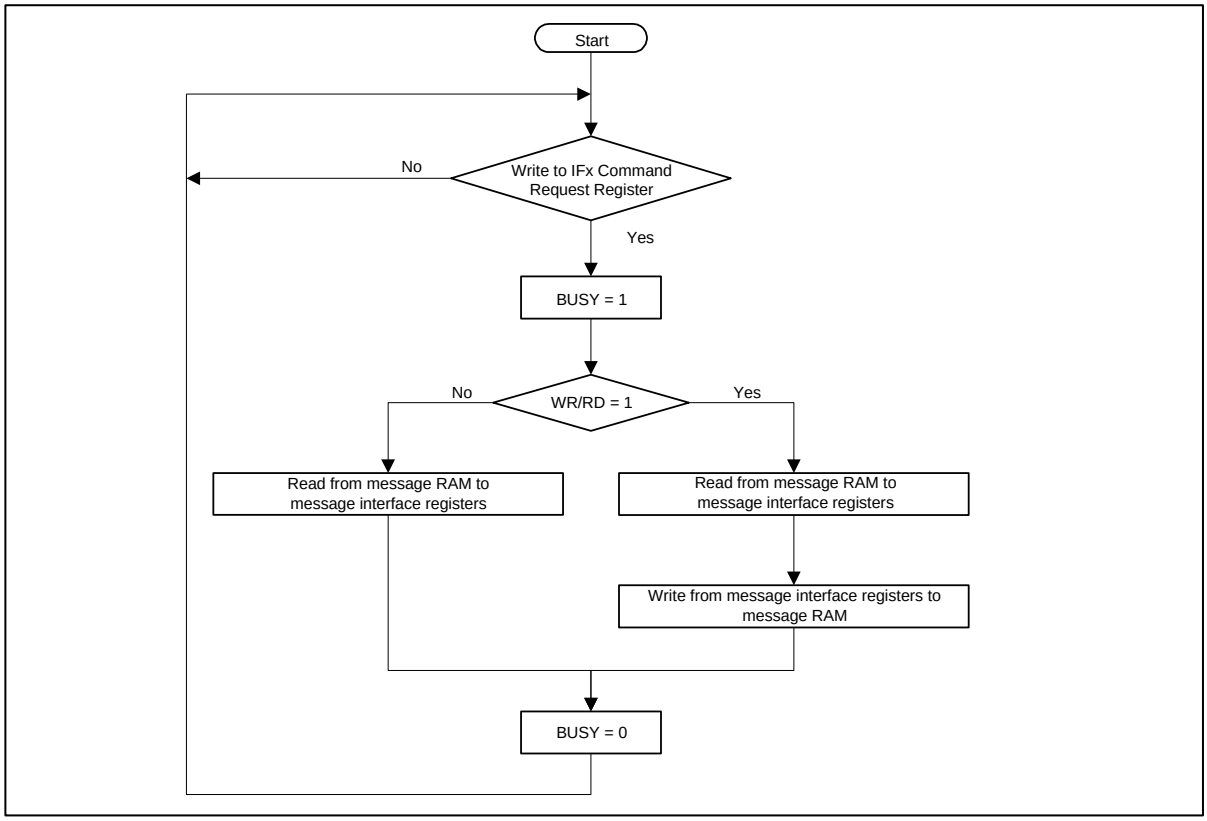
The CPU reads the received messages and updates outgoing messages via message interface registers. The CPU is interrupted according to the configuration of the CAN Control Register and IFx Message Control Register (message object).

Data transfer from/to message RAM

When data transfer starts between the message interface registers and message RAM, the BUSY bit in the IFx Command Request Register is set to "1". After the transfer has finished, the BUSY bit is cleared to "0". (See Figure 3-1)

The IFx Command Register selects whether to transfer complete data or only partial data of one message object. The structure of the message RAM does not allow the writing of single bits/bytes of one message object. The complete data of one message object is always written to the message RAM. Therefore, the data from the message interface registers to the message RAM is transferred in a read-modify-write cycle.

Figure 3-1 Data transfer between the message interface registers and message RAM



3.2 Message transmission

The following explains how to configure the send message objects, and about the transmission.

Sending messages

If there is no data transfer between the message interface registers and message RAM, the MsgVal bit in the CAN Message Valid Register and the TxRqst bit in the CAN Transmit Request Register are evaluated. A valid message object with the highest priority of pending transmission requests is transferred to the shift register for transmission. Then the NewDat bit of the message object is reset to "0".

When the transmission has finished successfully, and if there is no new data in the message object (NewDat = "0"), the TxRqst bit is reset to "0". If TxIE is set to "1", then the IntPnd bit is set to "1" after a successful transmission. If the CAN controller lost the arbitration on the CAN bus, or if an error occurred during transmission, the message is resent immediately when the CAN bus becomes idle.

Transmission priority

The transmission priority of the message objects is determined by the message number. Message object 1 has the highest priority, while message object 32 (the largest number of the installed message objects) has the lowest priority. If two or more transmission requests are pending, they are transferred in the order of corresponding message number from smallest to largest.

Notes:

- *In one of the following conditions, the messages may not be sent until any of the events described below occurs.*

  Conditions : (1) A message buffer with the lowest priority is used for transmission.
  (2) The TxRqst bit was previously set to "1", but is set to "0" to abort transmission.
  (3) The TxRqst bit is set to "1" again at the timing of (2).

  Events : - A valid message flows on the CAN bus.
  - A transmission request is issued to another message buffer.
  - CAN is initialized by the Init bit.

  If canceling the transmission is required to suit system operations, execute the following steps.

1. *Execute one of the following steps.*
   Do not use a message buffer with the lowest priority as a send message buffer.
   After aborting the transmission, generate any of the above events.

2. *Set the TxRqst bit to "1" again.*

- *If the message objects of ID28 to ID0, DLC3 to DLC0, Xtd, and Data7 to Data0 are changed while the TxRqst bit is "1", message objects before and after the change may be mixed for transmission, or the message objects after the change may not be transmitted. Therefore, be sure to change them while the TxRqst bit is "0".*

Configuring a send message object

Table 3-1 shows how a send object should be initialized.

Table 3-1 Initialization of a send message object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The IFx Arbitration Register (ID28 to ID0 and Xtd bit), given by the application, defines the ID and the type of the outgoing message.

If the standard frame (11-bit ID) is set, then ID28 to ID18 are used, and ID17 to ID0 are ignored. If the extended frame (29-bit ID) is set, then ID28 to ID0 are used.

If TxIE bit is set to "1", then the IntPnd bit is set to "1" after a successful transmission of the message object.

If the RmtEn bit is set to "1", the TxRqst bit is set to "1" after receiving the corresponding remote frame, and a data frame is sent automatically.

The data register (DLC3 to DLC0, Data0 to Data7) settings are given by the application.

When UMask is set to "1", the IFx Mask Register (Msk28 to Msk0, UMask, MXtd, and MDir bits) is used to receive remote frames with the IDs grouped by the mask setting, and then enable the transmission (by setting the TxRqst bit to "1"). For details, see Remote Frame in "3.3 Message reception".

Note:

- *The Dir bit in the IFx Mask Register must not be mask-enabled.*

Updating a send message object

The CPU can update the data of a send message object via the message interface registers.

The send message object data is written by four bytes of the corresponding IFx data register (in the unit of IFx data register A or IFx data register B). Therefore, the send message object cannot be changed by a single byte.

To update 8-byte data, write 0x0087 to the IFx Command Mask Register, and the message number to the IFx Command Request Register. This concurrently updates the send message object data (of 8-byte) and write "1" to the TxRqst bit.

If both the NewDat and TxRqst bits are set to "1", the NewDat bit is reset to "0" once the transmission is started.

Notes:

- *To update data, update it by four bytes of the IFx Data Register A or IFx Data Register B.*
- *If the message objects of ID28 to ID0, DLC3 to DLC0, Xtd, and Data7 to Data0 are changed while the TxRqst bit is "1", message objects before and after the change may be mixed for transmission, or the message objects after the change may not be transmitted. Therefore, be sure to change them while the TxRqst bit is "0".*

3.3 Message reception

The following explains how to configure the receive message object and about the reception.

Acceptance filtering for received messages

When the arbitration and control field (ID + IDE + RTR + DLC) of a message is completely shifted into the shift register of the CAN controller, scanning of the message RAM is started to compare matching with a valid message object.

Then the arbitration field and mask data (including MsgVal, UMask, NewDat, and EoB) are loaded from a message object in the message RAM, and the message object is compared with the arbitration field of the shift register including mask data.

This operation is repeated "until a matching is detected between a message object and the arbitration field of the shift register", or "until the last word of the message RAM is reached. "When a matching is detected, scanning of the message RAM is stopped, and the CAN controller processes data depending of the type of the received frame (data frame or remote frame).

Reception priority

The reception priority of the message objects is determined by the message number. Message object 1 has the highest priority, while message object 32 (the largest number of the installed message objects) has the lowest priority. If two or more objects are matched in the acceptance filtering, therefore, the object with the smallest message number becomes the receive message object.

Data frame reception

The CAN controller transfers the received message from the shift register into the message RAM of the message object matched in the acceptance filtering. The stored data includes all arbitration fields and the data length code as well as data bytes. This is implemented (to keep the ID and the data bytes) even if the IFx Mask Register is set to be masked.

The NewDat bit is set to "1" upon the reception of new data. When the CPU reads the message object, reset the NewDat bit to "0". If the NewDat bit has already been set to "1" upon the reception of a message, the MsgLst is set to "1" indicating that the previous data was lost.

If the RxIE bit has been set to "1", reception of a message buffer causes the IntPnd bit in the CAN Interrupt Pending Register to be set to "1". Then the TxRqst bit of the message object is reset to "0". This is implemented to prevent transmission of a remote frame when the requested data frame is received during the transmission.

Remote frame

One of the following three operations is selected when a remote frame is received. The selection depends on how the matching message object is configured.

1. Dir = "1" (Direction = Send), RmtEn = "1", UMask = "1" or "0"

   Receives the matched remote frame, sets only the TxRqst of this message object to "1", and automatically replies (sends) data frame to the remote frame. (Message objects other than TxRqst bit remain unchanged.)

2. Dir = "1" (Direction = Send), RmtEn = "0", UMask = "0"

   Does not receive an incoming remote frame, even if it matches the message object, and disables the remote frame. (The TxRqst bit of the message object remains unchanged.)

3. Dir = "1" (Direction = Send), RmtEn = "0", UMask = "1"

   If an incoming remote frame matches the message object, the TxRqst bit of the message object is reset to "0", and the remote frame is handled as if it were a received data frame. The received arbitration field and control field (ID + IDE + RTR + DLC) are stored into the message object in the message RAM, and the NewDat bit of this message object is set to "1", The data field of the message object remains unchanged.

Configuring a receive message object

Table 3-2 shows how a receive message object should be initialized.

Table 3-2 Initialization of a receive message object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The IFx Arbitration Register (ID28 to ID0 and Xtd bit) is given by the application. The register defines the ID and the type of a received message, used for the acceptance filtering.

If the standard frame (11-bit ID) is set, then ID28 to ID18 are used, and ID17 to ID0 are ignored. When a standard frame is received, ID17 to ID0 are reset to "0". If the extended frame (29-bit ID) is set, then ID28 to ID0 are used.

When the RxIE has been set to "1", and when a received data frame is stored into the message object, then the IntPnd bit is set to "1".

The data length code (DLC3 to DLC0) is given by the application. When the CAN controller stores the received data frame into the message object, it stores the received data length code and eight bytes data. If the data length code is less than eight, undefined data is written to the remaining bytes of the message object.

When UMask is set to "1", the IFx Mask Register (Msk28 to Msk0, UMask, MXtd, and MDir bits) is used to allow the reception of data frames with the IDs grouped by the mask setting. For details, see Data Frame Reception in "3.3 Message reception".

Note:

- *The Dir bit in the IFx Mask Register must not be mask-enabled.*

Handling a received message

The CPU can read a received message any time via the message interface registers.

The following shows an example of handling a received message. Write "0x007F" to the IFx Command Register, and a message number of the message object to the IFx Command Request Register. This procedure transfers a received message of the specified message number from the message RAM to the message interface registers. Then the NewDat bit and IntPnd bit of the message object can be cleared to "0" according to the configuration of the IFx Command Mask Register.

An incoming message is received if it is matched in the acceptance filtering. If the message object uses a mask for acceptance filtering, the masked data is excluded from the acceptance filtering to determine whether or not the message should be received.

The NewDat bit indicates whether a new message has been received since the last time the message object was read.

The MsgLst bit indicates that the previous received data was lost because the next data is received before the previous data is read from the message object. The MsgLst bit is not automatically reset.

During transmission of a remote frame, if a data frame matched in the acceptance filtering is received, the TxRqst bit is automatically reset to "0".

3.4 FIFO buffer function

The following explains the configuration of a FIFO buffer of the message object and its operations in handling received messages.

Configuration of FIFO buffer

The configuration of the receive message object belonging to a FIFO buffer is the same as that of a receive message object except the EoB bit. (See Configuring a Receive Message Object in "3.3 Message reception".)

A FIFO buffer is used by concatenating two or more receive message objects. To store received messages into this FIFO buffer, the ID and the mask settings of the receive message objects must be matched when they are used.

The first receive message object of the FIFO buffer has the lowest message number, i.e., the highest priority. In the last receive message object of the FIFO buffer, set "1" to the EoB bit to indicate that the object is the end of the FIFO buffer block. (Except in the last message object, the EoB bit in each message object that uses the FIFO buffer configuration must be set to "0".)

Notes:

- *Be sure to configure the same settings for the ID and the masks of message objects used in the FIFO buffer.*
- *When the FIFO buffer is not used, be sure to set the EoB bit to "1".*

Receiving messages using FIFO buffers

A received message, when it matches the FIFO buffer ID, is stored into the receive message object in the FIFO buffer with the lowest message number.

When a message is stored into the receive message object in the FIFO buffer, the NewDat bit of this receive message object is set to "1". When the NewDat bit is set in receive message object while the EoB bit is set to "0", the receive message object is protected until the last receive message object (with EoB bit = "1") is reached. Meanwhile, the CAN controller does not write to the FIFO buffer.

When both of the following conditions are met, the next incoming message is written to the last message object and therefore overwrites the previous message.

- Valid data is stored into the last FIFO buffer
- The NewDat bit of the receive message object is not written by "0" (to release the write protect)

If "0" is not written to the NewDat bit (to release the write protect) of the receive message object while valid data is stored into the last FIFO buffer, the next incoming message is written to the last message object and overwrites the previous message.

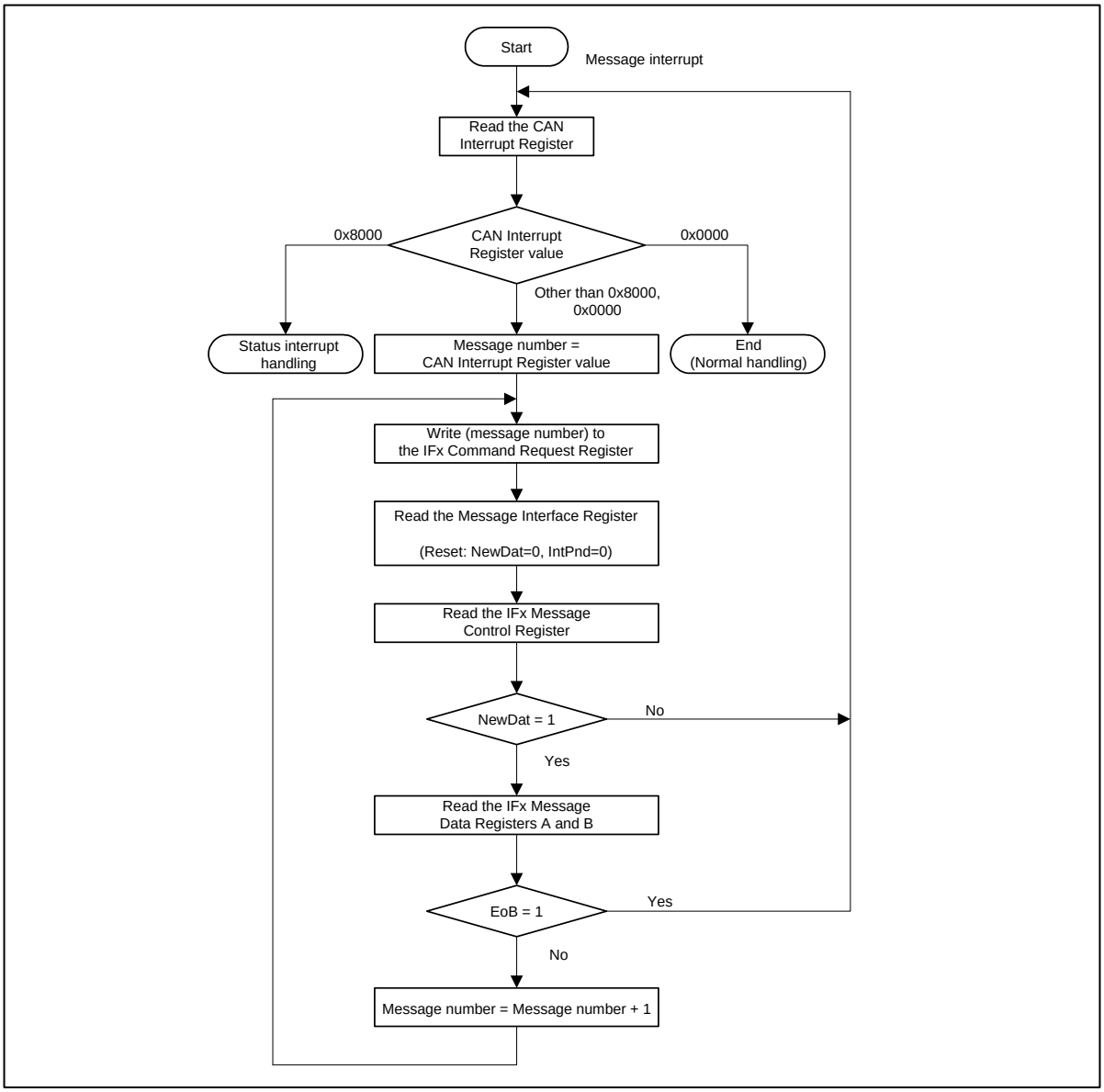
Reading from FIFO buffer

To read the contents of a receive message object, the CPU transfers the object to the Message Interface Register by writing the received message number to the IFx Command Request Register. Then, set WR/RD in the IFx Command Mask Register to "0" (read), set TxRqst/NewDat = 1, IntPnd = 1, and set the NewDat bit and IntPnd bit to "0".

To assure the correct FIFO buffer function, be sure to first read a receive message object in the FIFO buffer with the lowest message number, and then other objects in ascending order.

Figure 3-2 shows how the CPU handles the message objects the FIFO buffer concatenates.

Figure 3-2 CPU handling of FIFO buffer



3.5 Interrupt function

The following explains the interrupt handing using the status interrupt (IntId = 0x8000) and message interrupt (IntId = Message number).

If two or more interrupts are pending, the CAN Interrupt Register points to a pending interrupt code with the highest priority. The chronological order of the interrupt codes are neglected, and the interrupt code with the highest priority is always shown. The interrupt code is retained until the CPU clears it.

The status interrupt (0x8000 of the IntId bit) has the highest priority.

Priority of message interrupts is determined by the message number. A smaller number has a higher priority while the larger the lower.

A message interrupt is cleared by clearing the IntPnd bit of the message object. A status interrupt is cleared by reading the CAN Status Register.

The IntPnd bit in the CAN interrupt Pending Register indicates whether an interrupt has been caused. When no interrupts are pending, the IntPnd bit retains "0".

While the IE bit in the CAN Control Register, and the TxIE bit and RxIE bit in the IFx Message Control Register are set to "1", if the IntPnd bit turns to "1", then the interrupt line to the CPU becomes active. The interrupt line remains active until the CAN Interrupt Pending Register is cleared to "0" (the interrupt factor is reset) or the IE bit in the CAN Control Register is reset to "0".

The 0x8000 value of the CAN Interrupt Register indicates that the CAN Status Register has been updated by the CAN controller. This interrupt has the highest priority. The interrupt by updating the CAN Status Register can enable or disable the setting of the CAN Interrupt Register using the EIE bit and SIE bit in the CAN Control Register. The interrupt line to the CPU can be controlled by the IE bit in the CAN Control Register.

A write access from the CPU can update (reset) the RxOk bit, TxOk bit, and LEC bit in the CAN Status Register. However, the write access cannot generate or reset an interrupt.

Except the 0x8000 and 0x0000 values, the CAN Interrupt Register indicates that a message interrupt is pending, and that the interrupt has the highest priority.

The CAN Interrupt Register is updated even when IE is reset.

The factor of a message interrupt to the CPU can be checked from the CAN Interrupt Register or CAN Interrupt Pending Register. (See "4.5 Message handler registers") When clearing a message interrupt, the message data can be read concurrently. If a message interrupt indicated by the CAN Interrupt Register is cleared, the CAN Interrupt Register sets another interrupt with the next higher priority. This waits for the next interrupt handling. If no interrupts are pending, the CAN Interrupt Register shows the 0x0000 value.

Notes:

- *A status interrupt (IntId = 0x8000) is cleared by a read access to the CAN Status Register.*
- *A write access to the CAN Status Register will not generate a status interrupt (IntId = 0x8000).*

3.6 Bit timing

The following provides the overview of the bit timing and explains about the bit timing in the CAN controller.

Each CAN node in the CAN network has its own clock generator (usually a quartz oscillator). The time parameter of the bit time can be configured individually for each CAN node. Even if each CAN node's oscillator has a different cycle (fosc), a common bit rate can be generated.

The oscillator frequencies vary slightly because of changes in temperature or voltage, or deterioration of components. As long as the frequencies vary only within the tolerance range (df) of the oscillators, the CAN nodes can compensate for the different bit rates by resynchronizing to the bit stream.

The bit time can be divided into four segments according to the CAN specifications (see Figure 3-3), into the synchronization segment (Sync_Seg), the propagation time segment (Prop_Seg), the phase buffer segment 1 (Phase_Seg1), and the phase buffer segment 2 (Phase_Seg2). Each segment consists of the programmable number of time quanta (See Table 3-3). The basic unit of the time quantum (tq) is defined by CAN controller's system clock "fsys" and the baud rate prescaler (BRP).

tq = BRP / fsys

CAN's system clock "fsys" is the frequency of its clock input (See Figure 3-3). Synchronization segment Sync_Seg is a timing in the bit time where edges of the CAN bus level are expected to occur. Propagation time segment Prop_Seg compensates for the physical delay times within the CAN network. Phase buffer segments Phase_Seg1 and Phase_Seg2 must specify the sampling points. Resynchronization jump width (SJW) must define the width within which resynchronization can move the sampling point to compensate for edge phase errors.

Figure 3-3 Bit timing

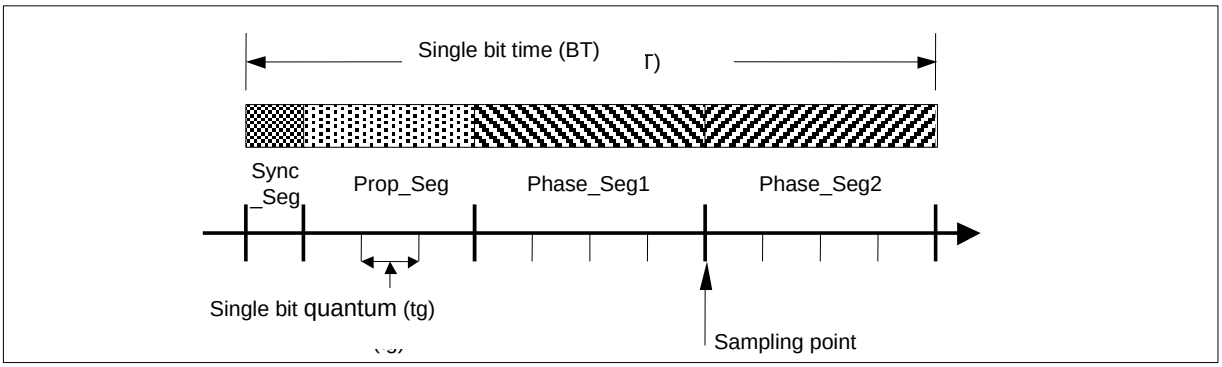


Table 3-3 CAN bit time parameters

Parameter	Range	Function
BRP	[1 to 32]	Defines the length of time quantum tq.
Sync_Seg	1 tq	Fixed length. Synchronization to system clock.
Prop_Seg	[1 to 8] tq	Compensates for the physical delay times.
Phase_Seg1	[1 to 8] tq	Assures edge phase errors before the sampling point. May be prolonged temporarily by synchronization.
Phase_Seg2	[1 to 8] tq	Assures edge phase errors after the sampling point. May be shortened temporarily by synchronization.
SJW	[1 to 4] tq	Resynchronization jump width. Will not be longer than either of the phase buffer segments.

The following shows the bit timing in the CAN controller.

Figure 3-4 The bit timing in the CAN controller

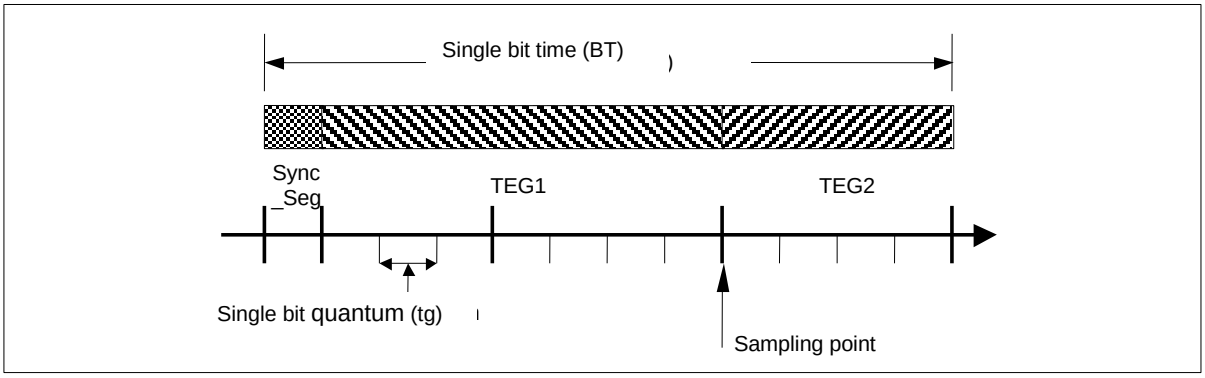


Table 3-4 CAN controller parameters

Parameter	Range	Function
BRPE, BRP	[0 to 1023]	Defines the length of time quantum tq. Can extend the prescaler by up to 1024 by the Bit Timing Register and the Prescaler Extension Register.
Sync_Seg	1 tq	Synchronization to system clock. Fixed length.
TSeg1	[1 to 15] tq	A time segment before the sampling point. Equivalent to Prop_Seg and Phase_Seg1. Can be controlled by the Bit Timing Register.
TSeg2	[0 to 7] tq	A time segment after the sampling point. Equivalent to Phase_Seg2. Can be controlled by the Bit Timing Register.
SJW	[0 to 3] tq	Resynchronization jump width. Can be controlled by the Bit Timing Register.

The following shows the relations among the parameters:

tq = ([BRPE, BRP]+1) / fsys
BT = SYNC_SEG + TEG1 + TEG2
= (1 + (TSeg1 + 1) + (TSeg2 + 1)) × tq
= (3 + TSeg1 + TSeg2) × tq

3.7 Test mode

The following explains how to configure test mode, and about its operations.

Test mode setting

Test mode is entered by setting the Test bit in the CAN Control Register to "1". In test mode, the Tx1, Tx0, LBack, Silent, and Basic bits in the CAN Test Register are enabled.

When the Test bit in the CAN Control Register is set to "0", all test register functions are disabled.

Silent mode

The CAN controller can be set in silent mode by programming the Silent bit in the CAN Test Register to "1".

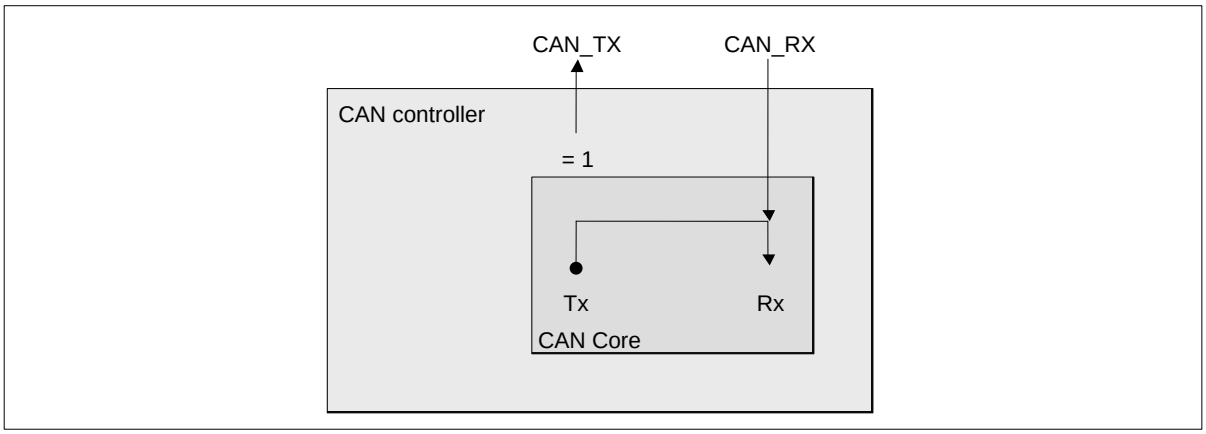
In silent mode, the CAN controller can receive data frames and remote frames, but only outputs recessive bits onto the CAN bus and does not send messages and ACK.

When the CAN controller is required to send dominant bits (ACK bits, overload flags, active error flags), the CAN controller uses the internal rerouting circuit to send them to the RX side. In this operation, the RX side can receive dominant bits rerouted inside the CAN controller even when the CAN bus remains in a recessive state.

In silent mode, the analysis of CAN bus traffic is possible without being affected by transmission of the dominant bits (ACK bits, error flags).

Figure 3-5 shows the connection of the CAN_TX and CAN_RX signals to the CAN controller in silent mode.

Figure 3-5 CAN controller in silent mode



Loop back mode

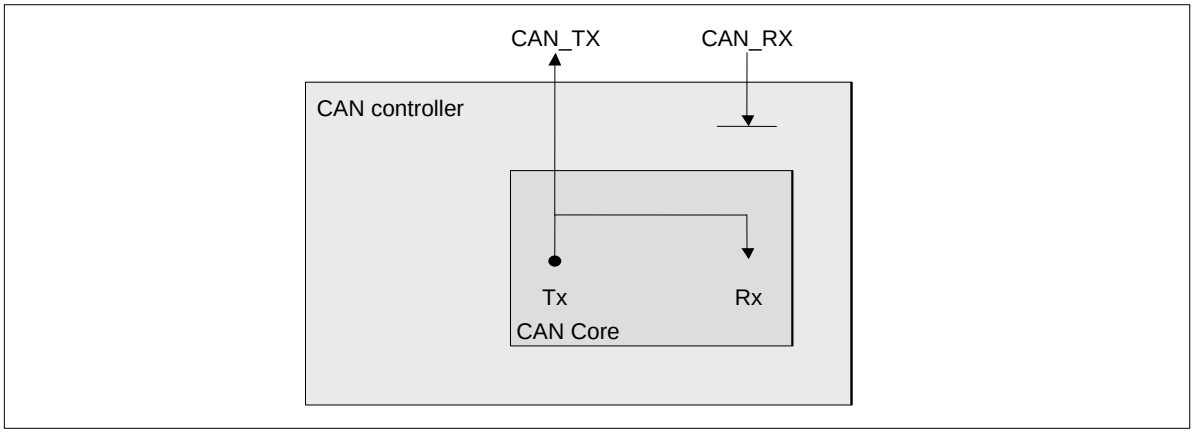
The CAN controller can be set in loop back mode by programming the LBack bit in the CAN Test Register to "1".

Loop back mode can be used for self-diagnostic functions.

In loop back mode, TX is connected with RX inside the CAN controller. The CAN controller treats the transmitted messages as messages received by RX, and stores the messages passed acceptance filtering into the receive buffer.

Figure 3-6 shows the connection of the CAN_TX and CAN_RX signals to the CAN controller in loop back mode.

Figure 3-6 CAN controller in loop back mode



Note:

- *Being independent of external signals, the CAN controller does not sample dominant bits in the acknowledgement slot of a data/remote frame. This usually causes the CAN controller to generate acknowledgement errors. In this test mode, however, the errors are not caused.*

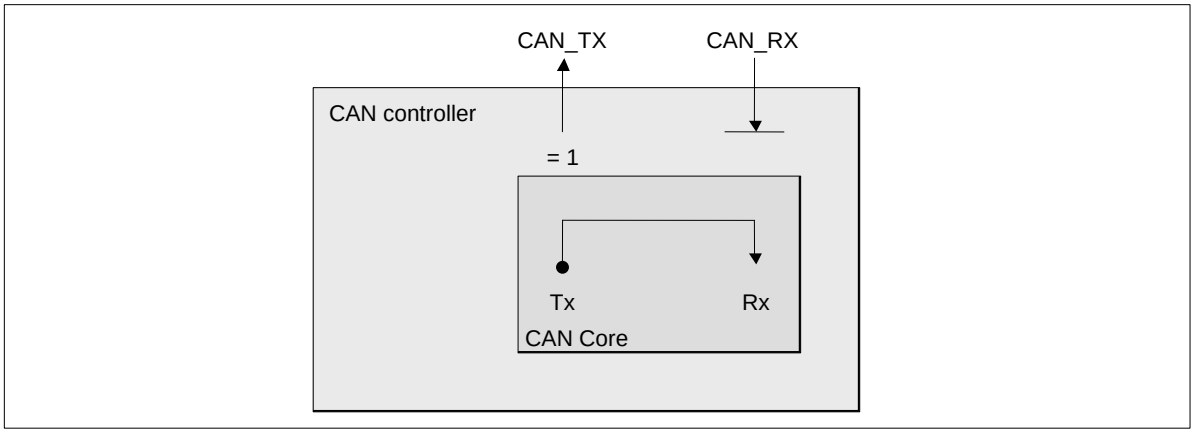
Combination of silent mode and loop back mode

Loop back mode and silent mode can be combined by setting the LBack bit and Silent bit in the CAN Test Register to "1" at the same time.

This mode can be used for "Hot self-test". The "Hot self-test" means that the CAN controller can be tested in loop back mode without affecting operation of the CAN system, because a constant recessive value is output from the CAN_TX pin and the input to the CAN_RX pin is ignored.

Figure 3-7 shows the connection of the CAN_TX and CAN_RX signals to the CAN controller when silent mode and loop back mode are combined.

Figure 3-7 CAN controller in combined silent and loop back modes



Basic mode

The CAN controller can be set in basic mode by programming the Basic bit in the CAN Test Register to "1".

In basic mode the CAN controller runs without using the message RAM.

The IF1 Message Interface Register is used to control transmission.

First when sending a message, the contents of transmission are configured in the IF1 Message Register. Then the BUSY bit in the IF1 Command Request Register is set to "1" to request transmission. While the BUSY bit is set to "1", the IF1 Message Interface Register is locked or the transmission is pending.

When the BUSY bit is set to "1", the CAN controller performs the following operation:

Immediately when the CAN bus becomes idle, the CAN controller loads the contents of the IF1 Message Interface Register to the send shift register to start transmission. When the transmission has finished successfully, the BUSY bit is reset to "0", and the locked IF1 Message Interface Register is released.

While pending, the transmission can be aborted by resetting the BUSY bit in the IF1 Command Request Register to "0". If the BUSY bit is reset to "0" during the transmission, a possible retransmission in case of lost arbitration or error detection is disabled.

The IF2 Message Interface Register is used to control reception.

All contents of the message are received without using acceptance filtering. The contents of the received message can be read by setting the BUSY bit in the IF2 Command Request Register to "1".

When the BUSY bit is set to "1", the CAN controller performs the following operation:

- Stores the received message (the contents of the receive shift register) into the IF2 Message Interface Register without any acceptance filtering.

If a new message is stored into the IF2 Message Interface Register, the CAN controller sets the NewDat bit to "1". When an additional message is received while the NewDat bit is "1", then CAN controller sets MsgLst to "1".

Notes:

- *In basic mode, all the message objects related to control and status bits are ignored as well as the control mode setting of the IFx Command Mask Register.*
- *The message number of the command request register is ignored.*
- *The NewDat bit and MsgLst bit in the IF2 Message Control Register retain their usual function, DLC3 to DLC0 indicates the received DLC, and other control bits are read as "0".*

Software control of the CAN_TX pin

CAN_TX is a CAN send pin and has four output functions:

- Outputs serial data (Usual output)
- Outputs CAN sampling point signals to monitor the bit timing of the CAN controller
- Outputs a constant dominant value
- Outputs a constant recessive value

The output of constant dominant and recessive values, combined with CAN_RX monitoring function of the CAN receive pin, can be used to check the CAN bus physical layer.

The output mode of the CAN_TX pin can be controlled by the Tx1 and Tx0 bits in the CAN Test Register.

Note:

- *When using CAN message transmission or any of the loop back, silent, or basic modes, the CAN_TX must be set to the serial data output.*

3.8 Software initialization

The following explains about initialization using software.

The sources of software initialization are as follows:

- Hardware reset
- Setting the Init bit in the CAN Control Register
- Shift to a busoff state

A hardware reset initializes all other than the message RAM (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits). The message RAM must be initialized, after the hardware reset, by the CPU or by setting the MsgVal in the message RAM to "0". The Bit Timing Register must be configured before clearing the Init bit in the CAN Control Register to "0".

The Init bit in the CAN Control Register is set to "1" in the following conditions:

- Writing "1" from the CPU
- Hardware reset
- In a busoff state

When the Init bit is set to "1", all message transfer from/to the CAN bus is stopped, and the CAN_TX pin in the CAN bus output is in a recessive state (excluding CAN_TX test mode).

Setting the Init bit to "1" does not change the error counter and any register.

When the Init bit and CCE bit in the CAN Control Register are set to "1", the Bit Timing Register for baud rate control and Prescaler Extension Register can be configured.

The software initialization is completed by resetting the Init bit to "0".

By waiting for the occurrence of a consecutive 11 recessive bits (i.e., bus idle) after the Init bit is reset to "0", the message is transferred after synchronization with data transfer on the CAN bus.

Before changing message object masks ID, Xtd, EoB, and RmtEn during normal operation, the MsgVal must be disabled.

4. CAN Registers

The following registers are provided for CAN.

- CAN Control Register (CTRLR)
- CAN Status Register (STATR)
- CAN Error Counter (ERRCNT)
- CAN Bit Timing Register (BTR)
- CAN Interrupt Register (INTR)
- CAN Test Register (TESTR)
- CAN Prescaler Extension Register (BRPER)
- IFx Command Request Register (IFxCREQ)
- IFx Command Mask Register (IFxCMSK)
- IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx Arbitration 1, 2 (IFxARB1, IFxARB2)
- IFx Message Control Register (IFxMCTR)
- IFx Data Register A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)
- CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)
- CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)
- CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)
- CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

Total control register list

Table 4-1 Total control register list

Abbreviation	Register name	Reference
CTRLR	CAN Control Register	4.2.1
STATR	CAN Status Register	4.2.2
ERRCNT	CAN Error Counter	4.2.3
BTR	CAN Bit Timing Register	4.2.4
INTR	CAN Interrupt Register	4.2.5
TESTR	CAN Test Register	4.2.6
BRPER	CAN Prescaler Extension Register	4.2.7

Message interface register list

Table 4-2 Message interface register list

Abbreviation	Register name	Reference
IF1CREQ	IF1 Command Request Register	4.3.1
IF1CMSK	IF1 Command Mask Register	4.3.2
IF1MSK1	IF1 Mask Register 1	4.3.3
IF1MSK2	IF1 Mask Register 2	4.3.3
IF1ARB1	IF1 Arbitration Register 1	4.3.4
IF1ARB2	IF1 Arbitration Register 2	4.3.4
IF1MCTR	IF1 Message Control Register	4.3.5
IF1DTA1	IF1 Data A Register 1 (Little endian)	4.3.6
IF1DTA2	IF1 Data A Register 2 (Little endian)	4.3.6
IF1DTB1	IF1 Data B Register 1 (Little endian)	4.3.6
IF1DTB2	IF1 Data B Register 2 (Little endian)	4.3.6
IF1DTA2	IF1 Data A Register 2 (Big endian)	4.3.6
IF1DTA1	IF1 Data A Register 1 (Big endian)	4.3.6
IF1DTB2	IF1 Data B Register 2 (Big endian)	4.3.6
IF1DTB1	IF1 Data B Register 1 (Big endian)	4.3.6
IF2CREQ	IF2 Command Request Register	4.3.1
IF2CMSK	IF2 Command Mask Register	4.3.2
IF2MSK1	IF2 Mask Register 1	4.3.3
IF2MSK2	IF2 Mask Register 2	4.3.3
IF2ARB1	IF2 Arbitration Register 1	4.3.4
IF2ARB2	IF2 Arbitration Register 2	4.3.4
IF2MCTR	IF2 Message Control Register	4.3.5
IF2DTA1	IF2 Data A Register 1 (Little endian)	4.3.6
IF2DTA2	IF2 Data A Register 2 (Little endian)	4.3.6
IF2DTB1	IF2 Data B Register 1 (Little endian)	4.3.6
IF2DTB2	IF2 Data B Register 2 (Little endian)	4.3.6
IF2DTA2	IF2 Data A Register 2 (Big endian)	4.3.6
IF2DTA1	IF2 Data A Register 1 (Big endian)	4.3.6
IF2DTB2	IF2 Data B Register 2 (Big endian)	4.3.6
IF2DTB1	IF2 Data B Register 1 (Big endian)	4.3.6

Message handler register list

Table 4-3 Message handler register list

Abbreviation	Register name	Reference
TREQ1	CAN Transmit Request Register 1	4.5.1
TREQ2	CAN Transmit Request Register 2	4.5.1
NEWDT1	CAN New Data Register 1	4.5.2
NEWDT2	CAN New Data Register 2	4.5.2
INTPND1	CAN Interrupt Pending Register 1	4.5.3
INTPND2	CAN Interrupt Pending Register 2	4.5.3
MSGVAL1	CAN Message Valid Register 1	4.5.4
MSGVAL2	CAN Message Valid Register 2	4.5.4

4.1 CAN register functions

An address space of 256 bytes is allocated to the CAN registers. The CPU gains access to the message RAM via the message interface registers.

This section lists CAN registers, and describes the detailed function of each register.

Total control registers

- CAN Control Register (CTRLR)
- CAN Status Register (STATR)
- CAN Error Counter (ERRCNT)
- CAN Bit Timing Register (BTR)
- CAN Interrupt Register (INTR)
- CAN Test Register (TESTR)
- CAN Prescaler Extension Register (BRPER)

Message interface registers

- IFx Command Request Register (IFxCREQ)
- IFx Command Mask Register (IFxCMSK)
- IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2)
- IFx Message Control Register (IFxMCTR)
- IFx Data Registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)

Message handler registers

- CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)
- CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)
- CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)
- CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

4.2 Total control registers

Total control registers control the CAN protocol and operating modes, and provide status information.

Total control registers

- CAN Control Register (CTRLR)
- CAN Status Register (STATR)
- CAN Error Counter (ERRCNT)
- CAN Bit Timing Register (BTR)
- CAN Interrupt Register (INTR)
- CAN Test Register (TESTR)
- CAN Prescaler Extension Register (BRPER)

4.2.1 CAN Control Register (CTRLR)

The CAN Control Register controls the operating modes of the CAN controller.

Register configuration

■ CAN Control Register (high-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

■ CAN Control Register (low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Test	CCE	DAR	Reserved	EIE	SIE	IE	Init
Attribute	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1

Register functions

[bit15:8] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

[bit7] Test: Test mode enable bit

bit	Function
0	Normal operation [Initial value]
1	Test mode

Note:

- *The Test bit can be set to "1" only while the Init bit is "1".*

[bit6] CCE: Bit Timing Register write enable bit

bit	Function
0	Disables write access to the CAN Bit Timing Register and CAN Prescaler Extension Register. [Initial value]
1	Enables write access to the CAN Bit Timing Register and CAN Prescaler Extension Register. This setting is valid while the Init bit is "1".

[bit5] DAR: Automatic retransmission disable bit

bit	Function
0	Enables automatic retransmission when arbitration is lost or an error is detected. [Initial value]
1	Disables automatic retransmission.

Based on the CAN specification (ISO11898. See 6.3.3 Recovery Sequence), the CAN controller automatically resends frames when arbitration is lost or an error is detected during transfer. To allow the automatic retransmission, set the DAR bit to "0". To operate CAN in Time Triggered CAN (TTCAN, See ISO11898-1) environments, set the DAR bit to "1".

Notes:

- *In the mode where the DAR bit is set to "1", the TxRqst bit and the NewDat bit of a message object behave differently. (For message objects, see "4.4 Message objects")*
- *When frame transmission has started, the TxRqst bit of the message object is reset to "0" while NewDat remains set.*
- *When frame transmission has finished successfully, the NewDat bit is reset to "0".*
  If arbitration is lost or an error is detected during transmission, the NewDat bit remains set.
  To restart the transmission, the CPU must set the TxRqst to "1".
- *If the DAR bit in the CAN Control Register (CTRLR) is changed from "0" to "1" during frame transmission (TxRqst = "1"), a frame being transmitted will be transmitted again. Therefore, change the DAR bit only while the Init bit is "1".*
- *A transmission using two or more message buffers while the DAR bit is set to "1" assumes the following operations:*
- *If the TxRqst in other message buffer is set to "1" before or during frame transmission (TxRqst bits in multiple message buffers are set to "1"), all the set TxRqst bits are reset to "0" upon the start of frame transmission, and data in the message buffer with the highest priority will be sent.*

  When frame transmission has finished successfully, the NewDat bit of the sent message buffer is reset to "0" and, if TxIE of the message buffer is "1" then, IntPnd of the message object is set to "1". Data in other message buffers will not be sent because their TxRqst bits have been reset to "0" upon the start of frame transmission.
  Check the message buffer sent by NewDat and IntPnd, and then set TxRqst and NewDat to "1" again for another message buffer to be sent.

[bit4] Reserved: Reserved bit

This bit is read as "0", and must be set to "0" when writing.

[bit3] EIE: Error interrupt code enable bit

bit	Function
0	A change of the BOff or EWarn bit in the CAN Status Register disables the setting of interrupt code in the CAN Interrupt Register. [Initial value]
1	A change of the BOff or EWarn bit in the CAN Status Register enables the setting of status interrupt code in the CAN Interrupt Register.

[bit2] SIE: Status interrupt code enable bit

bit	Function
0	A change of the TxOk, RxOk, or LEC bit in the CAN Status Register disables the setting of interrupt code in the CAN Interrupt Register. [Initial value]
1	A change of the TxOk, RxOk, or LEC bit in the CAN Status Register enables the setting of status interrupt code in the CAN Interrupt Register. A change of TxOk, RxOk, or LEC bit caused by write access from the CPU is not set in the CAN Interrupt Register.

[bit1] IE: Interrupt enable bit

bit	Function
0	Disables interrupt generation. [Initial value]
1	Enables interrupt generation.

[bit0] Init: Initialization bit

bit	Function
0	CAN controller operations enabled.
1	Initialization [Initial value]

Notes:

- *The busoff recovery sequence (see CAN Specification Rev. 2.0) cannot be shortened by setting or resetting the Init bit. If the device enters busoff state, the CAN controller itself sets the Init bit to "1", stopping all bus operations. If the Init bit is cleared to "0" from the busoff state, the bus operation remains stopped until 129 bus idle sequences (one bus idle sequence consists of 11 recessive bits) occur consecutively. When the bus recovery sequence has completed, the error counter is reset.*
- *If the Init bit is set to "1" and then reset to "0" during the busoff recovery sequence, the busoff recovery sequence restarts from the beginning (sends a set of 11 recessive bits 129 times).*
- *To write to the CAN Bit Timing Register, set the Init and CCE bits to "1".*
- *Setting the Init bit to "1" during transfer stops data reception immediately.*
- *To set the Init bit to "1" during transmission, set the Init bit to "1" after the transmission has finished. If you set the Init bit to "1" during transmission, set the Init bit to "0" and then wait for a two-bit time to perform the transmission setting (TxRqst="1").*
- *Before making transition to low consumption mode (stop mode or clock mode), and before changing clock supply, the Init bit must be set to "1" to initialize the CAN controller.*
- *To change the division ratio of clock supplied to the CAN interface by using the following registers, set the Init bit to "1" to stop the CAN controller previously.*
- *CAN Bit Timing Register (BTR)*
- *CAN Prescaler Extension Register (BRPER)*
- *CAN Prescaler (CANPRE)*

4.2.2 CAN Status Register (STATR)

The CAN Status Register indicates the CAN status and a CAN bus state.

Register configuration

■ CAN Status Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

■ CAN Status Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	BOff	EWarn	EPass	RxOk	TxOk	LEC		
Attribute	R,WX	R,WX	R,WX	R,W	R,W	R,W		
Initial value	0	0	0	0	0	000		

Register functions

[bit15:8] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

[bit7] BOff: Busoff bit

bit	Function
0	CAN bus is not in busoff state. [Initial value]
1	CAN bus is in busoff state.

[bit6] EWarn: Warning bit

bit	Function
0	Both the send and receive counters are below 96. [Initial value]
1	Send or receive counter has reached or exceeded 96.

[bit5] EPass: Error passive bit

bit	Function
0	Both the send and receive counters are below 128 (error active state). [Initial value]
1	The RP bit of the receive counter is "1", or the send counter is between 128 and 255 (error passive state).

[bit4] RxOk: Successful message reception bit

bit	Function
0	No message has been transferred successfully on the CAN bus, or the bus is in idle state. [Initial value]
1	A messages has been transferred successfully on the CAN bus.

[bit3] TxOk: Successful message transmission bit

bit	Function
0	The bus is in idle state, or no message has been sent successfully. [Initial value]
1	A messages has been sent successfully.

Note:

- *The RxOk and TxOk bits can be reset only by the CPU.*

[bit2:0] LEC: Last error code bits

bit2:0	State	Function
0	Normal	Successful transmission or reception. [Initial value]
1	Stuff error	Six or more dominant or recessive bits have been detected consecutively in a message.
2	Form error	A wrong fixed format part of a received frame has been detected.
3	Ack error	A sent message was not acknowledged by another node.
4	Bit 1 error	In the sent message data excluding the arbitration field, bits that have been sent as recessive data is detected as dominant data.
5	Bit 0 error	In the sent message data excluding the arbitration field, bits that have been sent as dominant data is detected as recessive data. This bit is set each time 11 recessive bits are detected during bus recovery. The bus recovery sequence can be monitored by reading this bit.
6	CRC error	The CRC data in a received message did not match the calculated CRC value.
7	Undetected	If the CPU wrote "7" to the LEC bit, and the LEC value is read as "7" afterward, it indicates that no bus event has been detected since the CPU wrote the value. (The bus is in idle state)

The LEC bit holds a code that indicates the last error occurred on the CAN bus. When a message has been transferred (sent or received) without error, this bit is cleared to "0". The undetected code "7" is written by the CPU to check for code updates.

Notes:

- *If the BOff and EWarn bits change while the EIE bit is "1", or if the RxOk, TxOk, and LEC bits change while the SIE bit is "1", the status interrupt code (0x8000) is written to the CAN Interrupt Register.*
- *Writing from the CPU updates the RxOk and TxOk bits, and this erases the RxOk and TxOk bits set by the CAN controller. If the RxOk and TxOk bits are used, clear the RxOk and TxOk bits within the time (45 × BT) after they are set to "1". BT indicates one bit time.*
- *If a change of the LEC bit causes an interrupt while the SIE bit is "1", do not write to the CAN Status Register.*
- *No interrupt is caused by a change of the EPass bit, or writing to the RxOk, TxOk, and LEC bits from the CPU.*
- *When the BOff bit has turned to "1", the EPass bit and EWarn bit are "1". When the EPass bit has turned to "1", the EWarn bit is "1".*
- *The status interrupt (0x8000) of the CAN Interrupt Register is cleared by reading this register.*

4.2.3 CAN Error Counter (ERRCNT)

The CAN Error Counter indicates the receive error passive, the receive error counter, and the send error counter.

Register configuration

■ CAN Error Counter (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	RP	REC[6:0]						
Attribute	R,WX	R,WX						
Initial value	0	0000000						

■ CAN Error Counter (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	TEC[7:0]							
Attribute	R,WX							
Initial value	0x00							

Register functions

[bit15] RP: Receive error passive indication

bit	Function
0	The receive error counter is below the error passive level. [Initial value]
1	The receive error counter has reached the error passive level defined in the CAN specification.

[bit14:8] REC[6:0]: Receive error counter

A receive error counter value. The range of the receive error counter value is between 0 and 127.
If the receive error counter reaches or exceeds 128, the RP bit is set to "1", and the counter is not refreshed.

Example: If a receive error adds 8 to REC[6:0] = 127 with RP = 0,
then REC[6:0] = 127 with RP = 1.
If a receive error adds 8 to REC[6:0] = 126 with RP = 0,
then REC[6:0] = 126 with RP = 1.
If a receive error adds 8 to REC[6:0] = 119 with RP = 0,
then REC[6:0] = 127 with RP = 0.
If reception is successful when REC[6:0] = 126 and RP = 1,
then REC[6:0] = 125 and RP = 0.

[bit7:0] TEC[7:0]: Send error counter

A send error counter value. The range of the send error counter value is between 0 and 255.
If the send error counter reaches or exceeds 256, the Init bit of the CAN Control Register is set to "1", and the counter is not refreshed.

Example: If a send error adds 8 to TEC[7:0] = 255 with Init = 0,
then TEC[7:0] = 255 with Init = 1.
If a send error adds 8 to TEC[7:0] = 254 with Init = 0,
then TEC[7:0] = 254 with Init = 1.
If a receive error adds 8 to TEC[7:0] = 247 with Init = 0,
then TEC[7:0] = 255 with Init = 0.

4.2.4 CAN Bit Timing Register (BTR)

The CAN Bit Timing Register configures the prescaler and the bit timing.

Register configuration

■ CAN Bit Timing Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved	TSeg2			TSeg1			
Attribute	-	R/W			R/W			
Initial value	0	010			0011			

■ CAN Bit Timing Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	SJW		BRP					
Attribute	R/W		R/W					
Initial value	00		000001					

Register functions

[bit15] Reserved: Reserved bit

This bit is read as "0", and must be set to "0" when writing.

[bit14:12] TSeg2: Time segment 2 setting bits

Valid programmed values are 0 to 7. The TSeg2 + 1 value is the time segment 2.

The time segment 2 is equivalent to the Phase Buffer Segment (PHASE_SEG2) in the CAN specification.

[bit11:8] TSeg1: Time segment 1 setting bits

Valid programmed values are 1 to 15. The 0 value must not be used. The TSeg1 + 1 value is the time segment 1.

The time segment 1 is equivalent to the Propagation Segment (PROP_SEG) + Phase Buffer Segment 1 (PHASE_SEG1) in the CAN specification.

[bit7:6] SJW: Resynchronization jump width setting bits

Valid programmed values are 0 to 3. The SJW + 1 value is the resynchronization jump width.

[bit5:0] BRP: Baud rate prescaler setting bits

Valid programmed values are 0 to 63. The BRP + 1 value is the baud rate prescaler.

It determines the basic unit of time quantum (tq) for the CAN controller by dividing the system clock (fsys).

Note:

- *The CAN Bit Timing Register and CAN Prescaler Extension Register must be configured while the Init bit and CCE bit in the CAN Control Register are set to "1".*

4.2.5 CAN Interrupt Register (INTR)

The CAN Interrupt Register indicates message interrupt code and status interrupt code.

Register configuration

■ CAN Interrupt Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	IntId15 to IntId8							
Attribute	R,WX							
Initial value	0x00							

■ CAN Interrupt Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	IntId7 to IntId0							
Attribute	R,WX							
Initial value	0x00							

Register functions

bit15:0	Function
0x0000	No interrupt
0x0001 to 0x0020	An interrupt factor indicates a message object number. (Message interrupt code)
0x0021 to 0x7FFF	Unused.
0x8000	Indicates an interrupt by a change in the CAN Status Register. (Status interrupt code)
0x8001 to 0xFFFF	Unused.

If two or more interrupts are pending, the CAN Interrupt Register indicates a high-priority interrupt code. If a high-priority interrupt code is generated while an interrupt code is set to the CAN Interrupt Register, the CAN Interrupt Register is updated to the high-priority interrupt code.

High-priority interrupt codes are arranged in the order of status interrupt code (0x8000), message interrupt codes (0x0001, 0x0002, 0x0003, ..., 0x0020).

When the IE bit of the CAN Control Register is set to "1" while the IntId bit is not 0x0000, a CPU interrupt signal becomes active. When the IntId bit is set to 0x0000 (an interrupt factor is reset) or the IE bit of the CAN Control Register is reset to "0", an interrupt signal becomes inactive.

To clear a message interrupt code, reset the IntPnd bit of the target message object (see "4.4 Message objects" for the message object) to "0".

A status interrupt code is cleared by reading the CAN Status Register.

Note:

- *To read the CAN Interrupt Register, access it in halfword or word mode.*

4.2.6 CAN Test Register (TESTR)

The CAN Test Register is used to set the test mode and monitor the RX pin. For operations, see "3.7 Test mode".

Register configuration

■ CAN Test Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

■ CAN Test Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Rx	Tx1	Tx0	LBack	Silent	Basic	Reserved	Reserved
Attribute	R,WX	R/W	R/W	R/W	R/W	R/W	-	-
Initial value	r	0	0	0	0	0	0	0

The initial value "r" of Rx in bit 7 indicates the level on the CAN bus.

Register functions

[bit15:8] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

[bit7] Rx: Rx pin monitor bit

bit	Function
0	Indicates that the CAN bus is in the dominant state.
1	Indicates that the CAN bus is in the recessive state.

[bit6:5] Tx1, Tx0: TX pin control bits

bit6	bit5	Function
0	0	Normal operation. [Initial value]
0	1	Outputs a sampling point to the Tx pin.
1	0	Outputs a dominant to the TX pin.
1	1	Outputs a recessive to the TX pin.

[bit4] LBack: Loop back mode

bit	Function
0	Disables loop back mode. [Initial value]
1	Enables loop back mode.

[bit3] Silent: Silent mode

bit	Function
0	Disables silent mode. [Initial value]
1	Enables silent mode.

[bit2] Basic: Basic mode

bit	Function
0	Disables basic mode. [Initial value]
1	Enables basic mode. The IF1 register is used for a sent message, and the IF2 register for a received message.

[bit1:0] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

Notes:

- *After setting "1" to the Test bit of the CAN Control Register, write data to this register. When the Test bit of the CAN Control Register is set to "1", test mode becomes valid. If the Test bit of the CAN Control Register is set to "0" during processing, test mode changes to normal mode.*
- *If the Tx bits are set to a value other than "00", no message can be sent.*

4.2.7 CAN Prescaler Extension Register (BRPER)

The CAN Prescaler Extension Register is used to extend the prescaler used in the CAN controller by combining it with the prescaler specified at a CAN bit timing.

Register configuration

■ CAN Prescaler Extension Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

■ CAN Prescaler Extension Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Reserved				BRPE			
Attribute	-				R/W			
Initial value	0000				0000			

Register functions

[bit15:4] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

[bit3:0] BRPE: Baud rate prescaler extension bits

These bits are used to extend the baud rate prescaler up to 1023 by combining BRP and BRPE in the CAN Bit Timing Register.

The value "{BRPE (MSB: 4 bits), BRP (LSB: 6 bits)} + 1" is set as the prescaler value of the CAN controller.

4.3 Message interface registers

The CAN controller provides two message interface registers to control an access from the CPU to the message RAM.

The CAN controller provides two message interface registers to control an access from the CPU to the message RAM. These two registers are used to avoid a confliction between an access from the CPU to the message RAM and an access from the CAN controller to the message RAM by buffering the data (message object) transferred or to be transferred. A message object (see "4.4 Message objects" for message object) is used to collectively transfer data between the message interface registers and message RAM.

Two message interface registers have the same functions, excluding basic test mode, and can be operated independently. For example, the IF2 Message Interface Register can be used to read data from the message RAM while the IF1 Message Interface Register is being used to write data to the message RAM. Table 4-2 shows two message interface registers.

Each Message Interface Register consists of two components: (1) Command Register (Command Request and Command Mask Registers) and (2) Message Buffer Register (Mask, Arbitration, Message Control, and Data Registers) controlled with the Command Register. The Command Mask Register indicates the data transfer direction and also which part in a message object is to be transferred. The Command Request Register is used to select a message number and perform the operation specified in the Command Mask Register.

4.3.1 IFx Command Request Register (IFxCREQ)

The IFx Command Request Register is used to select a message number of the message RAM and transfer data between the message RAM and Message Buffer Register. In basic test mode, IF1 is used to control sending and IF2 to control receiving.

Register configuration

■ IFx Command Request Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	BUSY	Reserved						
Attribute	R/W	-						
Initial value	0	0000000						

■ IFx Command Request Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Message Number							
Attribute	R/W							
Initial value	0x01							

Register functions

A message transfer starts between the message RAM and Message Buffer Register (Mask, Arbitration, Message Control, and Data Registers) immediately after a message number has been written to the IFx Command Request Register. This write operation sets the BUSY bit to "1" and continues transfer processing while the BUSY bit is "1". When transfer processing is ended, the BUSY bit is reset to "0".

If the CPU accesses the Message Interface Register while the BUSY bit is "1", the CPU waits until the BUSY bit is set to "0" (for 3 to 6 clock cycles after data has been written to the Command Request Register).

The method for using the BUSY bit is different in basic test mode. The IF1 Command Request Register, which is used as a send message, starts message sending when the BUSY bit is set to "1". When message transfer has finished successfully, the BUSY bit is reset to "0". Resetting the BUSY bit to "0" enables canceling message transfer at any time.

The IF2 Command Request Register, which is used for receiving message, stores the received message in the IF2 Message Interface Register when the BUSY bit is set to "1".

[bit15] BUSY: Busy flag bit

− Other than basic test mode

bit	Function
0	Indicates that data transfer is not performed between the Message Interface Register and message RAM. [Initial value]
1	Indicates that data transfer is being performed between the Message Interface Register and message RAM.

- Basic test mode
- IF1 Command Request Register

bit	Function
0	Disables message sending.
1	Enables message sending.

- IF2 Command Request Register

bit	Function
0	Disables message receiving.
1	Enables message receiving.

[bit14:8] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

[bit7:0] Message Number: Message number (32 message buffers)

bit7:0	Function
0x00, 0x40, 0x60, 0x80, 0xA0, 0xC0, 0xE0	Setting is prohibited. If specified, it is interpreted as 0x20, causing 0x20 to be read.
0x01 to 0x20	Specifies a message number to perform processing.
0x21 to 0x3F, 0x41 to 0x5F, 0x61 to 0x7F, 0x81 to 0x9F, 0xA1 to 0xBF, 0xC1 to 0xDF, 0xE1 to 0xFF	Setting is prohibited. If specified, it is interpreted as one of 0x01 to 0x1F, causing the interpreted value to be read.

Note:

- *The BUSY bit can be read and written. Therefore, writing any data to this bit does not affect operations, excluding in basic test mode (see "3.7 Test mode" for basic test mode).*

4.3.2 IFx Command Mask Register (IFxCMSK)

The IFx Command Mask Register is used to control the transfer direction between the Message Interface Register and message RAM and specify which data is to be updated. This register is invalid in basic test mode.

Register configuration

■ IFx Command Mask Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

■ IFx Command Mask Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	WR/RD	Mask	Arb	Control	CIP	TxRqst/NewDat	Data A	Data B
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Register functions

[bit15:8] Reserved: Reserved bits

These bits are read as "0", and must be set to "0" when writing.

[bit7] WR/RD: Writing or reading control bit

bit	Function
0	Indicates that data is read from the message RAM. Reading from the message RAM is performed by writing data to the IFx Command Request Register. What data is to be read from the message RAM depends on the setting of the Mask, Arb, Control, CIP, TxRqst/NewDat, Data A, or Data B bit. [Initial value]
1	Indicates that data is written to the message RAM. Writing to the message RAM is performed by writing data to the IFx Command Request Register. What data is to be written to the message RAM depends on the setting of the Mask, Arb, Control, CIP, TxRqst/NewDat, Data A, or Data B bit.

Note:

- *After resetting, data of the message RAM is unfixed. The message RAM cannot be read while its data is unfixed.*

The meaning of the bit6 to bit0 in the IFx Command Mask Register depends on the transfer direction specified with the WR or RD bit.

■ When the transfer direction is "writing" (WR/RD="1")

[bit6] Mask: Mask data update bit

bit	Function
0	Indicates that mask data (ID mask + MDir + MXtd) of a message object *1 is not updated. [Initial value]
1	Indicates that mask data (ID mask + MDir + MXtd) of a message object *1 is updated.

*1: See "4.4 Message objects".

[bit5] Arb: Arbitration data update bit

bit	Function
0	Indicates that arbitration data (ID + Dir + Xtd + MsgVal) of a message object *1 is not updated. [Initial value]
1	Indicates that arbitration data (ID + Dir + Xtd + MsgVal) of a message object *1 is updated.

*1: See "4.4 Message objects".

[bit4] Control: Control data update bit

bit	Function
0	Indicates that control data (IFx Message Control Register) of a message object *1 is not updated. [Initial value]
1	Indicates that control data (IFx Message Control Register) of a message object *1 is updated.

*1: See "4.4 Message objects".

[bit3] CIP: Interrupt clear bit

If this bit is set to "0" or "1", it does not affect CAN controller operations.

[bit2] TxRqst/NewDat: Message transmission request bit

bit	Function
0	Indicates that the TxRqst bits of the message object *1 and CAN Transmit Request Register are not changed. [Initial value]
1	Indicates that the TxRqst bits of the message object *1 and CAN Transmit Request Register are set to "1" (transmission requested).

*1: See "4.4 Message objects".

[bit1] Data A: Data0 to Data3 update bit

bit	Function
0	Indicates that Data0 to Data3 of a message object *1 is not updated. [Initial value]
1	Indicates that Data0 to Data3 of a message object *1 is updated.

*1: See "4.4 Message objects".

[bit0] Data B: Data4 to Data7 update bit

bit	Function
0	Indicates that Data4 to Data7 of a message object *1 is not updated. [Initial value]
1	Indicates that Data4 to Data7 of a message object *1 is updated.

*1: See "4.4 Message objects".

Notes:

- *When the TxRqst or NewDat bit of the IFx Command Mask Register is set to "1", the setting of the TxRqst bit in the IFx Message Control Register becomes invalid.*
- *This register is invalid in basic test mode.*

■ When the transfer direction is "reading" (WR/RD="0")

[bit6] Mask: Mask data update bit

bit	Function
0	Indicates that data (ID mask + MDir + MXtd) is not transferred from a message object *1 to IFx Master Register 1 or 2. [Initial value]
1	Indicates that data (ID mask + MDir + MXtd) is transferred from a message object *1 to IFx Master Register 1 or 2.

*1: See "4.4 Message objects".

[bit5] Arb: Arbitration data update bit

bit	Function
0	Indicates that data (ID + Dir + Xtd + MsgVal) is not transferred from a message object *1 to IFx Arbitration Register 1 or 2. [Initial value]
1	Indicates that data (ID + Dir + Xtd + MsgVal) is transferred from a message object *1 to IFx Arbitration Register 1 or 2.

*1: See "4.4 Message objects".

[bit4] Control: Control data update bit

bit	Function
0	Indicates that data is not transferred from a message object *1 to the IFx Message Control Register. [Initial value]
1	Indicates that data is transferred from a message object *1 to the IFx Message Control Register.

*1: See "4.4 Message objects".

[bit3] CIP: Interrupt clear bit

bit	Function
0	Indicates that the IntPnd bits of the message object *1 and CAN Interrupt Pending Register are held. [Initial value]
1	Indicates that the IntPnd bits of the message object *1 and CAN Interrupt Pending Register are cleared to "0".

*1: See "4.4 Message objects".

[bit2] TxRqst/NewDat: Data update bit

bit	Function
0	Indicates that the NewDat bits of the message object *1 and CAN New Data Register are held. [Initial value]
1	Indicates that the NewDat bits of the message object *1 and CAN New Data Register are cleared to "0".

*1: See "4.4 Message objects".

[bit1] Data A: Data0 to Data3 update bit

bit	Function
0	Indicates that data of the message object *1 and CAN Data Register A1 or A2 are held. [Initial value]
1	Indicates that data of the message object *1 and CAN Data Register A1 or A2 are updated.

*1: See "4.4 Message objects".

[bit0] Data B: Data4 to Data7 update bit

bit	Function
0	Indicates that data of the message object *1 and CAN Data Register B1 or B2 are held. [Initial value]
1	Indicates that data of the message object *1 and CAN Data Register B1 or B2 are updated.

*1: See "4.4 Message objects".

Notes:

- *The IntPnd and NewDat bits can be reset to "0" by reading a message object. However, the value before reset by reading is set to the IntPnd and NewDat bits of the IFx Message Control Register.*
- *This register is invalid in basic test mode.*

4.3.3 IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2)

The IFx Mask Registers 1 and 2 are used to write or read message object mask data of the message RAM. The specified mask data is invalid in basic test mode.

For the function of each bit, see "4.4 Message objects".

Register configuration

■ IFx Mask Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MXtd	MDir	Reserved	Msk28 to Msk24				
Attribute	R/W	R/W	R1,W1	R/W				
Initial value	1	1	1	11111				

■ IFx Mask Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Msk23 to Msk16							
Attribute	R/W							
Initial value	0xFF							

■ IFx Mask Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	Msk15 to Msk8							
Attribute	R/W							
Initial value	0xFF							

■ IFx Mask Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	Msk7 to Msk0							
Attribute	R/W							
Initial value	0xFF							

For the explanation of each bit in this register, see "4.4 Message objects".

Read "1" in the reserved bit (bit 13 of IFx Mask Register 2). Set "1" in write mode.

4.3.4 IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2)

The IFx Arbitration Registers 1 and 2 are used to write or read message object arbitration data of the message RAM. This register is invalid in basic test mode.

For the function of each bit, see "4.4 Message objects".

Register configuration

■ IFx Arbitration Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MsgVal	Xtd	Dir	ID28 to ID24				
Attribute	R/W	R/W	R/W	R/W				
Initial value	0	0	0	00000				

■ IFx Arbitration Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	ID23 to ID16							
Attribute	R/W							
Initial value	0x00							

■ IFx Arbitration Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	ID15 to ID8							
Attribute	R/W							
Initial value	0x00							

■ IFx Arbitration Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	ID7 to ID0							
Attribute	R/W							
Initial value	0x00							

For the explanation of each bit in this register, see "4.4 Message objects".

Note:

- *If the MsgVal bit of a message object is cleared to "0" during transmission, the TxOk bit of the CAN Status Register is set to "1" when transmission has been completed. However, the TxRqst bits of the message object and CAN Transmit Request Register are not cleared to "0". Use the Message Interface Register to clear the TxRqst bit to "0".*

4.3.5 IFx Message Control Register (IFxMCTR)

The IFx Message Control Register is used to write or read message object control data of the message RAM. IF1 Message Control Register is invalid in basic test mode. The NewDat and MsgLst bits of the IF2 Message Control Register are used to perform normal operations. The DLC bits indicate the DLC of the received message. The other control bits are invalid ("0").

For the function of each bit, see "4.4 Message objects".

Register configuration

■ IFx Message Control Register (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ IFx Message Control Register (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	EoB	Reserved			DLC3-0			
Attribute	R/W	-			R/W			
Initial value	0	000			0			

For the explanation of each bit in this register, see "4.4 Message objects".

Note:

The values of the TxRqst, NewDat, and IntPnd bits are set as shown below depending on the setting of the WR or RD bit in the IFx Command Mask Register.

- *When the transfer direction is "writing" (IFx Command Mask Register: WR/RD="1")*
- *The TxRqst bit of this register is valid only when the TxRqst or NewDat bit of the IFx Command Mask Register is set to "0".*

- *When the transfer direction is "reading" (IFx Command Mask Register: WR/RD="0")*
- *If the IntPnd bits of the message object and CAN Interrupt Pending Register are reset by setting the CIP bit of the IFx Command Mask Register to "1" and writing data to the IFx Command Request Register, the value of the IntPnd bit that is specified before reset is stored in this register.*
- *If the NewDat bits of the message object and CAN New Data Register are reset by setting the TxRqst or NewDat bit of the IFx Command Mask Register to "1" and writing data to the IFx Command Request Register, the value of the NewDat bit that is specified before reset is stored in this register.*

4.3.6 IFx Data Registers A1, A2, B1, and B2 (IFxDTA1, IFxDTA2, IFxDTB1, and IFxDTB2)

The IFx Data Registers A1, A2, B1, and B2 are used to write or read message object sending or receiving data to or from the message RAM. Those registers are used only to send or receive a data frame, and not to send or receive a remote frame.

Register configuration

	addr+3	addr+2	addr+1	addr+0
IFx Data A Register 1 (Little endian)			Data(1)	Data(0)
IFx Data A Register 2 (Little endian)	Data(3)	Data(2)		
IFx Data B Register 1 (Little endian)			Data(5)	Data(4)
IFx Data B Register 2 (Little endian)	Data(7)	Data(6)		
IFx Data A Register 2 (Big endian)			Data(2)	Data(3)
IFx Data A Register 1 (Big endian)	Data(0)	Data(1)		
IFx Data B Register 2 (Big endian)			Data(6)	Data(7)
IFx Data B Register 1 (Big endian)	Data(4)	Data(5)		

■ IFx Data Register

bit	15 7	14 6	13 5	12 4	11 3	10 2	9 1	8 0
Field	Data							
Attribute	R/W							
Initial value	0x00							

Register functions

- Send message data setting

  The set data is sent in the order of Data(0), Data(1), ..., Data(7), beginning with the MSB (bit 7 or bit 15).

- Received message data

  The received message data is stored in the order of Data(0), Data(1), ..., Data(7), beginning with the MSB (bit 7 or bit 15).

Notes:

- *If the received message data is less than eight bytes in length, undefined data is written to the remaining bytes of the Data Register.*
- *To transfer data to a message object, it is processed every four bytes in the Data A or Data B Register; therefore, it is impossible to update only a part of 4-byte data.*

4.4 Message objects

The message RAM provides 32 message objects. To avoid a confliction when simultaneously accessing the message RAM from the CPU and the CAN controller, the CPU cannot directly access message objects. The message RAM is accessed via the IFx Message Interface Register.

This section explains the configuration and functions of a message object.

Configuration of message object

Message object

UMask	Msk28 to Msk0	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28 to ID0	Xtd	Dir	DLC3 to DLC0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

Note:

- *A message object is not initialized using the Init bit of the CAN Control Register or the hardware reset function. For the hardware reset function, release the hardware reset function, and initialize the message RAM using the CPU or set MsgVal of the message RAM to "0".*

Functions of message object

The ID28 to ID0, Xtd, and Dir bits are used to indicate the ID and message type when sending a message. They are used in the acceptance filter together with the Msk28 to Msk0, MXtd, and MDir bits when receiving a message.

ID, IDE, RTR, DLC, and DATA in a data or remote frame that passed through the acceptance filter are respectively stored in ID28 to ID0, Xtd, Dir, DLC3 to DLC0, and Data7 to Data0 of a message object. Xtd indicates whether the received frame is an extension or standard frame. If Xtd is "1", a 29-bit ID (extension frame) is received. If Xtd is "0", a 11-bit ID (standard frame) is received.

When the received data or remote frame matches one or more message objects, it is stored in the message object with the lowest message number. For details, see Acceptance Filter for Received Messages in "3.3 Message reception".

MsgVal : Valid message bit

bit	Function
0	Message objects are invalid. Disables message sending/receiving.
1	Message objects are valid. Enables message sending/receiving.

Notes:

- *Reset the MsgVal bit of an unused message object to "0" before clearing the Init bit of the CAN Control Register to "0".*
- *Be sure to reset the MsgVal bit of a message object to "0" before changing the value of ID28 to ID0, Xtd, Dir, or DLC3 to DLC0.*
- *If the MsgVal bit of a message object is cleared to "0" during transmission, the TxOk bit of the CAN Status Register is set to "1" when transmission has been completed. However, the TxRqst bits of the message object and CAN Transmit Request Register are not cleared to "0". Use the Message Interface Register to clear the TxRqst bits to "0".*

UMask : Acceptance mask enable bit

bit	Function
0	Does not use Msk28 to Msk0, MXtd, or MDir.
1	Uses Msk28 to Msk0, MXtd, or MDir.

Notes:

- *Change the value of the UMask bit when the Init bit of the CAN Control Register is "1" or the MsgVal bit is "0".*
- *When the Dir bit is "1" and the RmtEn bit is "0", operations vary depending on the setting of the UMask bit.*
- *If the UMask bit is "1", reset the TxRqst bit to "0" when a remote frame has been received through the acceptance filter. The received ID, IDE, RTR, and DLC are stored in a message object, and the NewDat bit is set to "1" while data remains unchanged (data is handled as a data frame).*
- *If the UMask bit is "0", the TxRqst bit is held and a remote frame is ignored even if it has been received.*

ID28 to ID0 : Message ID

	Function
ID28 to ID0	Specifies a 29-bit ID (extension frame).
ID28 to ID18	Specifies a 11-bit ID (standard frame).

Msk28 to Msk0: ID mask

bit	Function
0	Masks the bit that corresponds to the ID of a message object.
1	Does not mask the bit that corresponds to the ID of a message object.

Xtd: Extension ID enable bit

bit	Function
0	Uses the 11-bit ID (standard frame) for message object.
1	Uses the 29-bit ID (extension frame) for message object.

MXtd : Extension ID mask bit

bit	Function
0	Does not compare the set value of the Xtd bit in a message object with that of the IDE bit of a received frame. Determines whether to perform the comparison as the ID of a standard frame or extension frame based on the IDE bit of a received frame.
1	Compares the set value of the Xtd bit in a message object with that of the IDE bit of a received frame.

Note:

- *When a 11-bit ID (standard frame) is set to a message object, the ID of a received data frame is written to ID28 to ID18. Msk28 to Msk18 are used to mask the ID.*

Dir: Message direction bit

bit	Function
0	Indicates the receiving direction. When the TxRqst bit is set to "1", a remote frame is sent. When the TxRqst bit is set to "0", a data frame that passed through the acceptance filter is received.
1	Indicates the transmission direction. When the TxRqst bit is set to "1", a data frame is sent. When the TxRqst is "0" and the RmtEn bit is "1", the CAN controller sets the TxRqst bit to "1" if a data frame that passed through the acceptance filter is received.

MDir : Message direction mask bit

bit	Function
0	Masks the message direction bit (Dir) through the acceptance filter.
1	Does not mask the message direction bit (Dir) through the acceptance filter.

Note:

- *Always set the Mdir bit to "1".*

EoB: End of buffer bit (For details, see "3.4 FIFO buffer function".)

bit	Function
0	Indicates that a message object is used as a FIFO buffer, not the last message.
1	Indicates a single message object or the last message object in the FIFO buffer.

Notes:

- *The EoB bit is used to configure a FIFO buffer for message objects 2 to 32.*
- *When processing a single message object without using a FIFO buffer, be sure to set the EoB bit to "1".*

NewDat: Data update bit

bit	Function
0	Indicates that no valid data resides.
1	Indicates that valid data resides.

MsgLst : Message lost

bit	Function
0	Message lost does not occur.
1	Message lost occurs.

Note:

- *The MsgLst bit is valid only when the Dir bit is "0" (receiving direction).*

RxIE: Receiving interrupt flag enable bit

bit	Function
0	Does not change the value of the IntPnd bit after frame receiving has succeeded.
1	Changes the IntPnd bit to "1" after frame receiving has succeeded.

TxIE: Transmission interrupt flag enable bit

bit	Function
0	Does not change the value of the IntPnd bit after frame transmission has succeeded.
1	Changes the IntPnd bit to "1" after frame transmission has succeeded.

IntPnd: Interrupt pending bit

bit	Function
0	No interrupt factor is detected.
1	An interrupt factor is detected. If other high-priority interrupt is not found, the IntId bit of the CAN Interrupt Register indicates this message object.

RmtEn: Remote enable

bit	Function
0	Does not change the value of the TxRqst bit when a remote frame has been received.
1	Sets the TxRqst bit to "1" when a remote frame is received while the Dir bit is "1".

Note:

When the Dir bit is "1" and the RmtEn bit is "0", operations vary depending on the setting of the UMask bit.

- *If the UMask bit is "1", reset the TxRqst bit to "0" when a remote frame has been received through the acceptance filter. The received ID, IDE, RTR, and DLC are stored in a message object. The NewDat bit is set to "1" while data remains unchanged (data is handled as a data frame).*
- *If the UMask bit is "0", the TxRqst bit is held and a remote frame is ignored even if it has been received.*

TxRqst : Transmission request bit

bit	Function
0	Indicates the sending idle state (neither the sending state nor the sending wait state).
1	Indicates the sending or sending wait state.

DLC3 to DLC0: Data length code

bit	Function
0 to 8	The data frame length is 0 to 8 bytes.
9 to 15	Setting is prohibited. 8-byte length if specified.

Note:

- *The received DLC is stored in the DLC bit if a data frame is received.*

Data 0 to Data7: Data 0 to Data 7

	Function
Data 0	First data byte in CAN data frame
Data 1	2nd data byte in CAN data frame
Data 2	3rd data byte in CAN data frame
Data 3	4th data byte in CAN data frame
Data 4	5th data byte in CAN data frame
Data 5	6th data byte in CAN data frame
Data 6	7th data byte in CAN data frame
Data 7	8th data byte in CAN data frame

Notes:

- *Serial data is output from the MSB (bit 7 or bit 15) to the CAN bus.*
- *If the received message data is less than eight bytes in length, unfixed data is written to the remaining bytes of the Data Register.*
- *To transfer data to a message object, it is processed every four bytes in the Data A or Data B Register; therefore, it is impossible to update only a part of 4-byte data.*

4.5 Message handler registers

Message handler registers are all in read only mode. The TxRqst, NewDat, IntPnd, and MsgVal bits of a message object and the IntId bit indicate the status.

Message handler registers

- CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)
- CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)
- CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)
- CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

4.5.1 CAN Transmit Request Registers 1, 2 (TREQR1, TREQR2)

The CAN Transmit Request Registers indicate the TxRqst bits of all message objects. These registers check which message object transmission request is pending by reading the TxRqst bit.

Register configuration

■ CAN Transmit Request Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	TxRqst32 to TxRqst25							
Attribute	R,WX							
Initial value	0x00							

■ CAN Transmit Request Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	TxRqst24 to TxRqst17							
Attribute	R,WX							
Initial value	0x00							

■ CAN Transmit Request Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	TxRqst16 to TxRqst9							
Attribute	R,WX							
Initial value	0x00							

■ CAN Transmit Request Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	TxRqst8 to TxRqst1							
Attribute	R,WX							
Initial value	0x00							

Register functions

TxRqst32 to TxRqst1: Transmission request bits

bit	Function
0	Indicates the sending idle state (neither the sending state nor the sending wait state).
1	Indicates the sending or sending wait state.

The following shows conditions to set or reset the TxRqst bit.

- Setting conditions
  - Set "1" to the WR/RD bit of the IFx Command Mask Register and "1" to the TxRqst bit, and write data to the IFx Command Request Register to set the TxRqst bit to a specific message object.
  - Set "1" to the WR/RD bit of the IFx Command Mask Register, "0" to the TxRqst bit, and "1" to the Control bit, and "1" to the TxRqst bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to set the TxRqst bit to a specific message object.
  - If the Dir bit is "1" and the RmtEn bit is "1", the TxRqst bit is set by receiving a remote frame that passed through the acceptance filter.

- Resetting conditions
  - Set "1" to the WR/RD bit of the IFx Command Mask Register, "0" to the TxRqst bit, and "1" to the Control, and "0" to the TxRqst bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to reset the TxRqst bit of a specific message object.
  - The TxRqst bit is reset when frame transmission has finished successfully.
  - If the Dir bit is "1", the RmtEn bit is "0", and the UMask bit is "1", the TxRqst bit is reset by receiving a remote frame that passed through the acceptance filter.

Notes:

- *In one of the following conditions, the messages may not be sent until any of the events described below occurs.*

  Conditions : *(1) A message buffer with the lowest priority is used for transmission.*
  (2) The TxRqst bit was previously set to "1", but is set to "0" to abort transmission.
  (3) The TxRqst bit is set to "1" again at the timing of (2).

  Events : *- A valid message flows on the CAN bus.*
  - A transmission request is issued to another message buffer.
  - CAN is initialized by the Init bit.

  If canceling the transmission is required to suit system operations, execute the following steps.

1. *Execute one of the following steps.*
   Do not use a message buffer with the lowest priority as a send message buffer.
   After aborting the transmission, generate any of the above events.

2. *Set the TxRqst bit to "1" again.*

- *If the message objects of ID28 to ID0, DLC3 to DLC0, Xtd, and Data7 to Data0 are changed while the TxRqst bit is "1", message objects before and after the change may be mixed for transmission, or the message objects after the change may not be transmitted. Therefore, be sure to change them while the TxRqst bit is "0".*

4.5.2 CAN New Data Registers 1, 2 (NEWDT1, NEWDT2)

The CAN New Data Registers indicate the NewDat bits of all message objects. These registers check which message object data is updated by reading the NewDat bit.

Register configuration

■ CAN New Data Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	NerwDat32 to NewDat25							
Attribute	R,WX							
Initial value	0x00							

■ CAN New Data Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	NerwDat24 to NewDat17							
Attribute	R,WX							
Initial value	0x00							

■ CAN New Data Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	NerwDat16 to NewDat9							
Attribute	R,WX							
Initial value	0x00							

■ CAN New Data Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	NerwDat8 to NewDat1							
Attribute	R,WX							
Initial value	0x00							

Register functions

NerwDat32 to NewDat1: Data update bit

bit	Function
0	Indicates that no valid data resides.
1	Indicates that valid data resides.

The following shows conditions to set or reset the NewDat bit.

- Setting conditions
  - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "1" to the NewDat bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to set the NewDat bit to a specific message object.
  - The NewDat bit is set by receiving a data frame that passed through the acceptance filter.
  - If the Dir bit is "1", the RmtEn bit is "0", and the UMask bit is "1", the NewDat bit is set by receiving a remote frame that passed through the acceptance filter.

- Resetting conditions
  - Set "0" to the WR/RD bit of the IFx Command Mask Register and "1" to the NewDat bit, and write data to the IFx Command Request Register to reset the NewDat bit of a specific message object.
  - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "0" to the NewDat bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to reset the NewDat bit of a specific message object.
  - The NewDat bit is reset after data has been transferred to the transmission shift register (internal register).

4.5.3 CAN Interrupt Pending Registers 1, 2 (INTPND1, INTPND2)

The CAN Interrupt Pending Registers indicate the IntPnd bits of all message objects. These registers check which message object is pending for interrupt by reading the IntPnd bits.

Register configuration

■ CAN Interrupt Pending Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	IntPnd32 to IntPnd25							
Attribute	R,WX							
Initial value	0x00							

■ CAN Interrupt Pending Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	IntPnd24 to IntPnd17							
Attribute	R,WX							
Initial value	0x00							

■ CAN Interrupt Pending Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	IntPnd16 to IntPnd9							
Attribute	R,WX							
Initial value	0x00							

■ CAN Interrupt Pending Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	IntPnd8 to IntPnd1							
Attribute	R,WX							
Initial value	0x00							

Register functions

IntPnd32 to IntPnd1: Interrupt pending bit

bit	Function
0	No interrupt factor is detected.
1	An interrupt factor is detected.

The following shows conditions to set or reset the IntPnd bit.

- Setting conditions
  - If the TxIE bit is set to "1", the IntPnd bit is set when frame transmission has been completed normally.
  - If the RxIE bit is set to "1", the IntPnd bit is set when a frame that passed through the acceptance filter was received normally.
  - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "1" to the IntPnd bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to set the IntPnd bit of a specific message object.

- Resetting conditions
  - Set "0" to the WR/RD bit of the IFx Command Mask Register and "1" to the CIP bit, and write data to the IFx Command Request Register to reset the IntPnd bit of a specific message object.
  - Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Control bit, and "0" to the IntPnd bit of the IFx Message Control Register. Then write data to the IFx Command Request Register to reset the IntPnd bit of a specific message object.

4.5.4 CAN Message Valid Registers 1, 2 (MSGVAL1, MSGVAL2)

The CAN Message Valid Registers indicate the MsgVal bits of all message objects. These registers check which message object is valid by reading the MsgVal bits.

Register configuration

■ CAN Message Valid Register 2 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MsgVal32 to MsgVal25							
Attribute	R,WX							
Initial value	0x00							

■ CAN Message Valid Register 2 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	MsgVal24 to MsgVal17							
Attribute	R,WX							
Initial value	0x00							

■ CAN Message Valid Register 1 (High-order byte)

bit	15	14	13	12	11	10	9	8
Field	MsgVal16 to MsgVal9							
Attribute	R,WX							
Initial value	0x00							

■ CAN Message Valid Register 1 (Low-order byte)

bit	7	6	5	4	3	2	1	0
Field	MsgVal8 to MsgVal1							
Attribute	R,WX							
Initial value	0x00							

Register functions

MsgVal32 to MsgVal1: Message valid bit

bit	Function
0	Message objects are invalid. Disables message sending/receiving.
1	Message objects are valid. Enables message sending/receiving.

The following shows conditions to set or reset the MsgVal bit.

- Setting conditions

  Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Arb bit, and "1" to the MsgVal bit of the IFx Arbitration Register 2. Then write data to the IFx Command Request Register to set the MsgVal bit of a specific message object.

- Resetting conditions

  Set "1" to the WR/RD bit of the IFx Command Mask Register, and "1" to the Arb bit, and "0" to the MsgVal bit of the IFx Arbitration Register 2. Then write data to the IFx Command Request Register to reset the MsgVal bit of a specific message object.

5. Notes

Table 5-1 and Table 5-2 show input and output signals.

Table 5-1 Table of input and output signals (Input signal)

NO	Signal name	I/O	Polarity	EDGE*1	Functions
1	CAN_CLK	I	-	-	Operation clock
2	CAN_RESET	I	H	ASYNC	Reset. When this signal is "H", initialization is performed.
3	CAN_SELECT	I	H	CAN_CLK↑	Register select signal. When this signal is "H", the register indicated by CAN_ADDR is selected.
4	CAN_WR_B	I	L	CAN_CLK↑	Access direction signal. This indicates read direction when this signal is "H" and CAN_SELECT="H", and indicates write direction when this signal is "L" and CAN_SELECT="H".
5	CAN_WR_SIZE [1:0]	I	-	CAN_CLK↑	Access size. During read, this signal is ignored and 32 bit access is performed. However, CAN_WR_SIZE="11" is disabled. "00" : 8 bit access "01" : 16 bit access "10" : 32 bit access "11" : Setting is prohibited. (32 bit access) When CAN_SELECT="H", this signal is enabled.
6	CAN_ADDR [7:0]	I	-	CAN_CLK↑	Address signal. When CAN_SELECT="H", the register for access is selected by CAN_WR_SIZE and this signal.
7	CAN_DATA_IN [31:0]	I	-	CAN_CLK↑	Writing data input to register.
8	CAN_RX	I	-	ASYNC	CAN receiving data input.

Table 5-2 Table of input and output signals (Output signal)

NO	Signal name	I/O	Polarity	EDGE*1	Initial value	Functions
9	CAN_DATA_OUT [31:0]	O	-	CAN_CLK↑	-	Register data output. When there is no read to register, "L" is returned.
10	CAN_WAIT_B	O	L	CAN_CLK↑	H	Transfer signal. This signal indicates data transferring state between message RAM and interface register. When this signal is "L", access to interface register (IF1/IF2) is disabled.
11	CAN_INT	O	H	CAN_CLK↑	L	Interrupt signal. When this signal is "H", interrupt is requested.
12	CAN_TX	O	-	CAN_CLK↑	H	CAN transmit data output.

*1 : Timing of change is indicated.

CHAPTER 5-3: CAN FD CONTROLLER

This chapter explains the functions and operations of the CAN FD Controller.

1. Overview
2. Configuration
3. Explanation of Operations
4. Setup procedure examples
5. Registers
6. Message RAM

CODE: FIP018CAN-FD-E01.3

1. Overview

The CAN FD Controller performs communication according to ISO11898-1 (CAN specification Rev. 2.0 part A, B) and to the CAN FD specification V1.0.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to a Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 192 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

The CAN FD Controller is accessible by the CPU using a data width of 8/16/32-bits. The CAN FD Controller's clock domain concept allows the separation between the two input clocks, the CAN clock and the Bus clock.

Features of the CAN FD Controller

- Conforms with ISO11898-1 (CAN specification Rev. 2.0 part A, B) and the CAN FD specification V1.0
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- Acceptance filtering
- Two configurable Receive FIFOs
- Separate signalling on reception of High Priority Messages
- Up to 64 dedicated Receive Buffers
- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable loop-back test mode
- Maskable interrupts
- Two clock domains (CAN clock and Bus clock)
- Power-down support
- Debug on CAN support

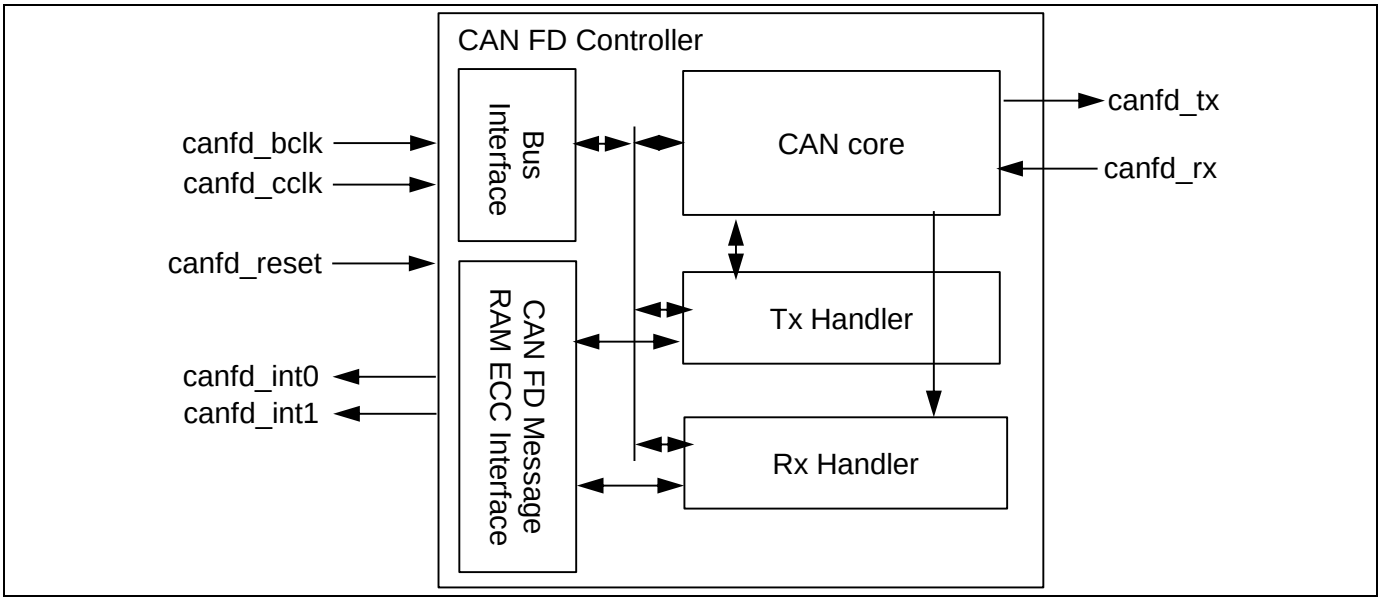
2. Configuration

This section provides configuration related information about the CAN FD controller.

2.1. Block Diagram
2.2. Dual Clock Sources
2.3. Dual Interrupt Lines

2.1 Block Diagram

Figure 2-1 CAN FD controller block diagram



CAN Core
CAN Protocol Controller. Handles all ISO 11898-1 protocol functions. Supports 11-bit and 29-bit identifiers.

CAN FD Message RAM ECC Interface
Interface with the CAN FD Message RAM ECC module. This module contains

- a Message RAM that stores messages to be transmitted, received messages, acceptance filter elements, and Tx Events, and
- ECC (Error Correction Code) logic for the Message RAM.

Bus interface
CPU interface bus with a data width of 8/16/32-bits.

Tx Handler

Controls the message transfer from the Message RAM to the CAN Core. A maximum of 32 Tx Buffers can be configured for transmission. Tx buffers can be used as dedicated Tx Buffers, as Tx FIFO, part of a Tx Queue, or as a combination of them. A Tx Event FIFO stores Tx timestamps together with the corresponding Message ID. Transmit cancellation is also supported.

Rx Handler

Controls the transfer of received messages from the CAN Core to the Message RAM. The Rx Handler supports two Receive FIFOs, each of configurable size, and up to 64 dedicated Rx Buffers for storage of all messages that have passed acceptance filtering. A dedicated Rx Buffer, in contrast to a Receive FIFO, is used to store only messages with a specific identifier. An Rx timestamp is stored together with each message. Up to 128 filters can be defined for 11-bit IDs and up to 64 filters for 29-bit IDs.

2.2 Dual Clock Sources

The two clocks of the CAN FD Controller must be configured to meet the following requirement in order to guarantee proper operation:

Bus clock (canfd_bclk) frequency > CAN clock (canfd_cclk) frequency

2.3 Dual Interrupt Lines

The CAN FD Controller provides two interrupt lines. Interrupts can be routed either to canfd_int0 or to canfd_int1. By default all interrupts are routed to interrupt line canfd_int0. By programming bits Enable Interrupt Line 0 (ILE.EINT0) and Enable Interrupt Line 1 (ILE.EINT1) the interrupt lines can be enabled or disabled separately.

3. Explanation of Operations

This section explains the operations of the CAN FD Controller.

3.1. Operating Modes
3.2. Timestamp Generation
3.3. Timeout Counter
3.4. Rx Handling
3.5. Tx Handling
3.6. FIFO Acknowledge Handling
3.7. Configuring the CAN Bit Timing

3.1 Operating Modes

This section explains the operating modes of the CAN FD Controller.

3.1.1. Software Initialization
3.1.2. Normal Operation
3.1.3. CAN FD Operation
3.1.4. Transceiver Delay Compensation
3.1.5. Restricted Operation Mode
3.1.6. Bus Monitoring Mode
3.1.7. Disabled Automatic Retransmission
3.1.8. Power Down (Sleep Mode)
3.1.9. Test Modes

3.1.1 Software Initialization

■ Setting/Resetting the Initialization bit (CCCR.INIT)

Software initialization is started by setting bit Initialization (CCCR.INIT),

- either by software or by a hardware reset,
- when an uncorrected bit error was detected in the Message RAM,
- or by going Bus_Off.

While CCCR.INIT is set,

- message transfer from and to the CAN bus is stopped,
- the status of the CAN bus output canfd_tx is recessive,
- the protocol error counters are unchanged.

Setting CCCR.INIT does not change any configuration register.

Resetting CCCR.INIT finishes the software initialization. Afterwards the CAN FD Controller synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Notes:

- *In case of setting CCCR.INIT to "1" during Normal Operation, set CCCR.INIT to "1" by issuing a Clock Stop Request via CCCR.CSR = "1". Before resetting CCCR.INIT first reset CCCR.CSR.*
- *When halting transmissions or receptions via CCCR.INIT = "1", cancel any pending transmissions (Tx Buffer Request Pending Register TXBRP ≠ 0x00000000) by setting Tx Buffer Cancellation Request Register TXBCR to 0xFFFF_FFFF.*

■ Access/set/reset properties of registers affected by Configuration Change Enable (CCCR.CCE)

Access to the configuration registers is only enabled when both bits CCCR.INIT and Configuration Change Enable (CCCR.CCE) are set (protected write). CCCR.CCE can only be set/reset while CCCR.INIT = "1". CCCR.CCE is automatically reset when CCCR.INIT is reset.

The following registers are reset when CCCR.CCE is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

In addition

- the Timeout Counter value (TOCV.TOC[15:0]) is preset to the value configured by the Timeout Period (TOCC.TOP[15:0]) when CCCR.CCE is set.
- the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR.CCE = "1".

The following registers are only writeable while CCCR.CCE = "0".

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

Test Mode Enable (CCCR.TEST) and Bus Monitoring Mode (CCCR.MON) can only be set by the CPU while CCCR.INIT = "1" and CCCR.CCE = "1". Both bits may be reset at any time.

Disable Automatic Retransmission (CCCR.DAR) can only be set/reset while CCCR.INIT = "1" and CCCR.CCE = "1".

Note:

- *To set CCCR.INIT and CCCR.CCE to "1", the CPU must follow the procedure below.*
  In case of setting CCCR.CCE to "1" just after hardware reset, it isn't necessary to follow the procedure.

1. *Cancel all pending transmission requests by setting Tx Buffer Cancellation Request Register TXBCR to 0xFFFF_FFFF (the Tx Buffer Request Pending Register TXBRP will be reset to "0")*
2. *Issue a clock stop request by setting Clock Stop Request (CCCR.CSR) to "1"*
3. *Wait till both CCCR.INIT and Clock Stop Acknowledge (CCCR.CSA) are "1"*
4. *First reset CCCR.CSR*
5. *Then reset CCCR.INIT*
6. *Wait till CCCR.INIT is "0"*
7. *Issue a second clock stop request by setting CCCR.CSR to "1"*
8. *Wait till both CCCR.INIT and CCCR.CSA are "1"*
9. *Set CCCR.CCE to "1" and reset CCCR.CSR*

■ Message RAM Initialization

The Message RAM should be zeroized before configuration of the CAN FD Controller in order to prevent Message RAM bit errors when reading uninitialized words, and also to avoid unexpected filter element configurations in the Message RAM.

3.1.2 Normal Operation

Once the CAN FD Controller is initialized and Initialization (CCCR.INIT) is reset to "0", the CAN FD Controller synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC (Data Length Code) are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

3.1.3 CAN FD Operation

There are two variants in the CAN FD frame format,

- The CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes.
- The CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

■ Enabling the CAN operation mode via CAN Mode Enable (CCCR.CME[1:0])

The CAN operation mode is enabled by programming CAN Mode Enable (CCCR.CME[1:0]). The transmission properties of the enabled CAN Mode can be changed via CAN Mode Request (CCCR.CMR[1:0]. See next paragraph for details).

- In case CCCR.CME[1:0] = "00", transmission and reception of CAN frames according to ISO 11898-1 is enabled.
- In case CCCR.CME[1:0] = "01", transmission of long CAN FD frames and reception of long and fast CAN FD frames is enabled.
- With CCCR.CME[1:0] = "10"/"11", transmission and reception of long and fast CAN FD frames is enabled.

CCCR.CME[1:0] can only be changed while Initialization (CCCR.INIT) and Configuration Change Enable (CCCR.CCE) are both set.

■ Changing CAN operation modes via CAN Mode Request (CCCR.CMR[1:0])

When initialization is left (CCCR.INIT set to "0"), a mode change to the CAN FD protocol option has to be requested by writing to CAN Mode Request (CCCR.CMR[1:0]).

A mode change requested by writing to CCCR.CMR[1:0] is allowed only when no transmission requests are pending (Tx Buffer Request Pending Register TXBRP = 0x00000000). If necessary, cancel pending transmission requests before changing the CAN operation mode. When the CAN FD Controller reaches idle phase after such mode change request, CCCR.CMR[1:0] is reset to "00" and the status flags CAN FD Bit Rate Switching (CCCR.FDBS) and CAN FD Operation (CCCR.FDO) are set accordingly. In case the requested CAN operation mode is not enabled, the value written to CCCR.CMR[1:0] is retained until it is overwritten by the next mode change request. Default is CAN operation according to ISO11898-1.

It is not necessary to change the CAN operation mode after system startup. A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting according to ISO11898-1 until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Bus Monitoring mode until programming has completed. Then all nodes switch back to CAN communication according ISO11898-1.
- Wake-up messages in CAN Partial Networking have to be transmitted in CAN format.

■ Interpretation of received frames

When CCCR.CME[1:0] ≠ "00", received CAN FD frames are interpreted according to the CAN FD Specification. The reserved bit in CAN frames with 11-bit identifiers and the first reserved bit in CAN frames with 29-bit identifiers will be decoded as EDL bit. EDL = recessive signifies a CAN FD frame, EDL = dominant signifies a standard CAN frame. In a CAN FD frame, the two bits following EDL, r0 and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by r0 = dominant and BRS = recessive. The coding of r0 = recessive in a CAN FD frame is used to detect Protocol Exception Events (see "3.4.5. Protocol Exception Event").

Reception of CAN frames according to ISO 11898-1 is possible in all CAN operation modes.

■ Format of Transmitted frames

The status bits CAN FD Operation (CCCR.FDO) and CAN FD Bit Rate Switching (CCCR.FDBS) indicate the format of transmitted frames.

- When CCCR.FDO is set, frames will be transmitted in CAN FD format with EDL = recessive.
- When both CCCR.FDO and CCCR.FDBS are set, frames will be transmitted in CAN FD format with bit rate switching and both bits EDL and BRS = recessive.

■ The DLC field

In the CAN FD format, the coding of the DLC differs from the CAN format. The DLC codes 0 to 8 have the same coding as in CAN, the codes 9 to 15, which in CAN all code a data field of 8 bytes, are coded according to Table 3-1 below.

Table 3-1 Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

■ Bit Rate Switching

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal bit timing is used as defined by the Bit Timing & Prescaler Register (BTP). In the following CAN FD data phase, the fast bit timing is used as defined by the Fast Bit Timing & Prescaler Register (FBTP). The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (canfd_cclk).

Example:

With a CAN clock frequency of 20MHz and the shortest configurable bit time of 4 tq, the bit rate in the data phase is 5 Mbit/s.

■ The Error Status Indicator ESI

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter’s error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

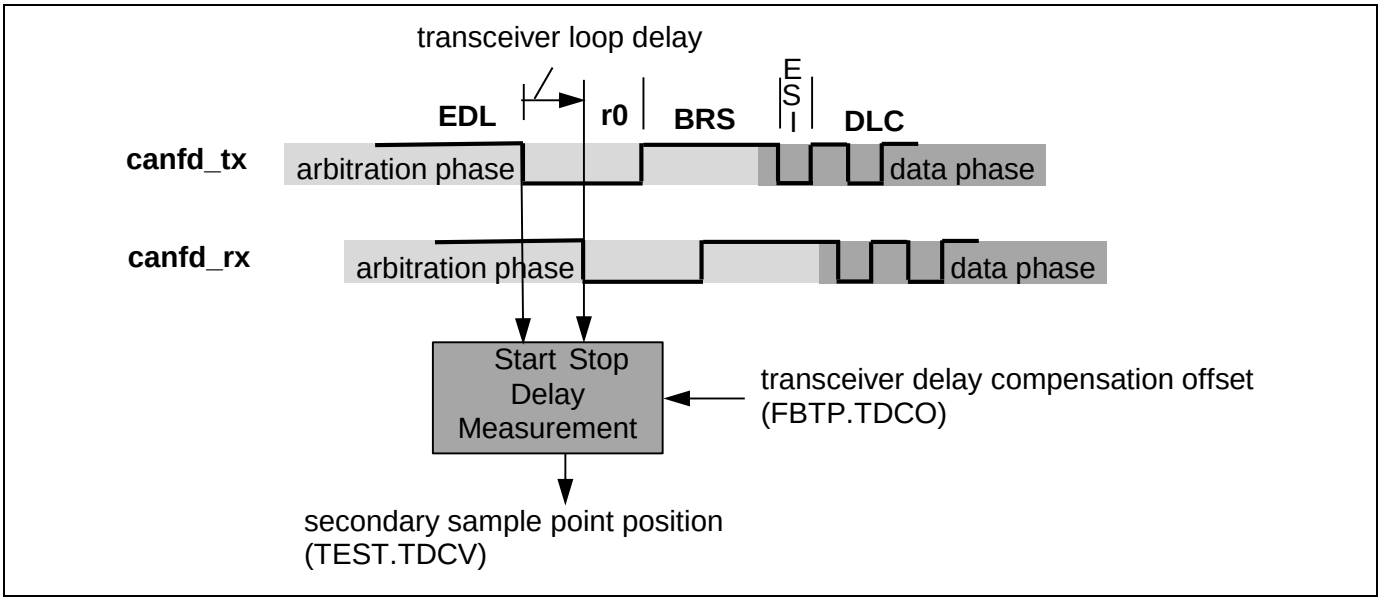
3.1.4 Transceiver Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin canfd_tx the protocol controller receives the transmitted data from its local CAN transceiver via pin canfd_rx. The received data is delayed by the CAN transceiver’s loop delay. In case this delay is greater than the time segment before the sample point of a data phase bit time, a bit error is detected. In order to enable a data phase bit time (fast bit time) that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceiver's loop delay.

3.1.4.1 Description

The CAN FD protocol unit has implemented a delay compensation mechanism to compensate the CAN transceiver's loop delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Figure 3-1 below describes how the transceiver loop delay is measured.

Figure 3-1 Transceiver delay measurement



Within each CAN FD frame, the transmitter measures the delay between the data transmitted at pin canfd_tx and the data received at pin canfd_rx. The measurement is done once, at the falling edge of bit EDL to bit r0. The delay is measured in canfd_cclk periods.

A secondary sample point position is calculated by adding a configurable transceiver delay compensation offset (FBTP.TDCO[4:0]) to the measured transceiver delay. This transceiver delay compensation value (TEST.TDCV[5:0]) is the sum of the measured transceiver delay and the transceiver delay compensation offset. The transceiver delay compensation offset is chosen to adjust the secondary sample point inside the bit time (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of time quanta tq.

To check for bit errors during the data phase, the delayed transmit data is compared against the received data at the secondary sample point. If a bit error is detected at the secondary sample point, the transmitter will react to this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

For the transceiver delay compensation, the sum of the measured delay from canfd_tx to canfd_rx and the configured transceiver delay compensation offset (FBTP.TDCO[4:0]) must meet both of the following boundary conditions:

- has to be less than 3 bit times in the data phase.
- has to be less or equal 63 canfd_cclk periods. In case this sum exceeds 63 canfd_cclk periods, the maximum value of 63 canfd_cclk periods is used for transceiver delay compensation.

The actual delay compensation value is monitored by reading the Transceiver Delay Compensation Value (TEST.TDCV[5:0]).

3.1.4.2 Configuration and Status

Compensation for the transceiver loop delay by the CAN FD Controller is enabled via the Transceiver Delay Compensation bit (FBTP.TDC). The transceiver delay compensation offset is configured via the Transceiver Delay Compensation Offset field (FBTP.TDCO[4:0]). The actual delay compensation value applied by the CAN FD Controller's protocol engine (i.e. the measured delay plus the offset) can be read from the Transceiver Delay Compensation Value field (TEST.TDCV[5:0]).

3.1.5 Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters are not incremented.

The CPU can set the CAN FD Controller into Restricted Operation Mode by setting the Restricted Operation Mode bit (CCCR.ASM). CCCR.ASM can only be set by the CPU when both Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set to "1". CCCR.ASM can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU must follow the procedure below:

10. *Cancel all pending transmission requests by setting Tx Buffer Cancellation Request Register TXBCR to 0xFFFF_FFFF (the Tx Buffer Request Pending Register TXBRP will be reset to "0")*
11. *Issue a clock stop request by setting Clock Stop Request (CCCR.CSR) to "1"*
12. *Wait till both CCCR.INIT and Clock Stop Acknowledge (CCCR.CSA) are "1"*
13. *First reset CCCR.CSR*
14. *Then reset CCCR.INIT*
15. *Wait till CCCR.INIT is "0"*
16. *Issue a second clock stop request by setting CCCR.CSR to "1"*
17. *Wait till both CCCR.INIT and CCCR.CSA are "1"*
18. *Set CCCR.CCE to "1" and reset CCCR.CSR and CCCR.ASM*
19. *Restart the CAN FD Controller by setting CCCR.INIT to "0"*
20. *Wait till CCCR.INIT is "0"*
21. *Reconfigure the CAN Operation Modes CAN FD Operation (CCCR.FDO) and CAN FD Bit Rate Switching (CCCR.FDBS) via CAN Mode Request (CCCR.CMR[1:0])*
22. *Request the transmissions cancelled in the third step*

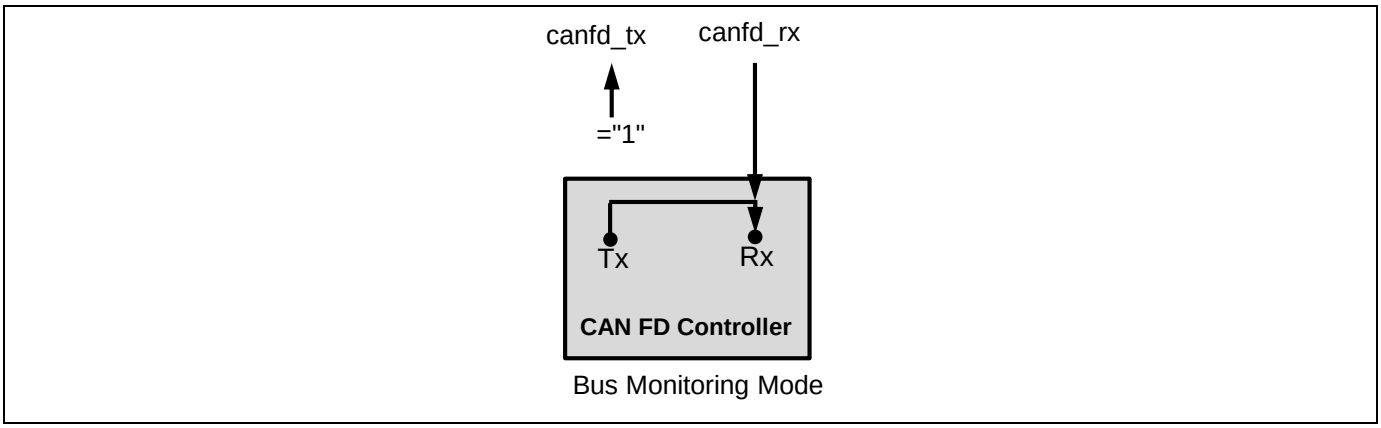
The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

3.1.6 Bus Monitoring Mode

The CAN FD Controller is set in Bus Monitoring Mode by programming the Bus Monitoring Mode bit (CCCR.MON) to one. In Bus Monitoring Mode (see ISO11898-1, 10.12 Bus monitoring), the CAN FD Controller is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN FD Controller is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN FD Controller monitors this dominant bit, although the CAN bus may remain in recessive state. The Tx Buffer Request Pending register (TXBRP) will be cleared when setting Configuration Change Enable CCCR.CCE = "1" upon entering Bus Monitoring Mode via CCCR.MON = "1", and will be held in reset state while it is in this mode.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 3-2 shows the connection of signals canfd_tx and canfd_rx to the CAN FD Controller in Bus Monitoring Mode.

Figure 3-2 Pin Control in Bus Monitoring Mode



3.1.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the CAN FD Controller provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. The automatic retransmission may be disabled via the Disable Automatic Retransmission bit (CCCR.DAR).

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. In this mode the CAN FD Controller will internally reset the corresponding Tx Buffer's Tx Request Pending bit (TXBRP.TRPn) when one of the following conditions occur:

- A successful transmission has occurred. This can be observed as
  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn not set

  or, in case a successful transmission was executed in spite of cancellation,

  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn set
- A transmission has not yet been started at the point of cancellation
- A transmission has been aborted due to lost arbitration or frame transmission disturbed. This can be observed by
  Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOn not set
  Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFn set

Note:

- *Do not use the same Tx Buffer for consecutive DAR transmissions or wait at least for 4 nominal bit times after successful transmission before requesting the next transmission from the same Tx Buffer.*

3.1.8 Power Down (Sleep Mode)

The CAN FD Controller can be set into power down mode via Clock Stop Request (CCCR.CSR).

When all pending transmission requests have completed (when Tx Buffer Request Pending Register TXBRP = 0x00000000), the CAN FD Controller waits until bus idle state is detected. Then the CAN FD Controller sets Initialization (CCCR.INIT) to one to prevent any further CAN transfers. Now the CAN FD Controller acknowledges that it is ready for power down by setting Clock Stop Acknowledge (CCCR.CSA) to one. In this state, before the clocks are switched off, further register accesses can be made. A write access to CCCR.INIT will have no effect. Now the CAN FD Controller clock inputs canfd_bclk and canfd_cclk may be switched off.

To leave power down mode, the application has to turn on the CAN FD Controller clocks before resetting Clock Stop Request (CCCR.CSR). The CAN FD Controller will acknowledge this by resetting CCCR.CSA. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT.

3.1.9 Test Modes

To enable write access to the Test register TEST, Test Mode Enable (CCCR.TEST) has to be set to one. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin canfd_tx by programming Control of Transmit Pin (TEST.TX). These are

- the default function – the serial data output
- drive the CAN Sample Point signal to monitor the CAN FD Controller’s bit timing
- drive constant dominant values
- drive constant recessive values.

The actual value at pin canfd_rx can be read from Receive Pin (TEST.RX). Both functions can be used to check the CAN bus’ physical layer.

Due to the synchronization mechanism between the CAN clock and Bus clock domains, there may be a delay of several Bus clock periods between writing to TEST.TX until the new configuration is visible at output pin canfd_tx. This applies also when reading input pin canfd_rx via TEST.RX.

Note:
Test modes should be used for self test only. The software control for pin canfd_tx interferes with all CAN protocol functions. It is not recommended to use test modes for application.

3.1.9.1 External Loop Back Mode

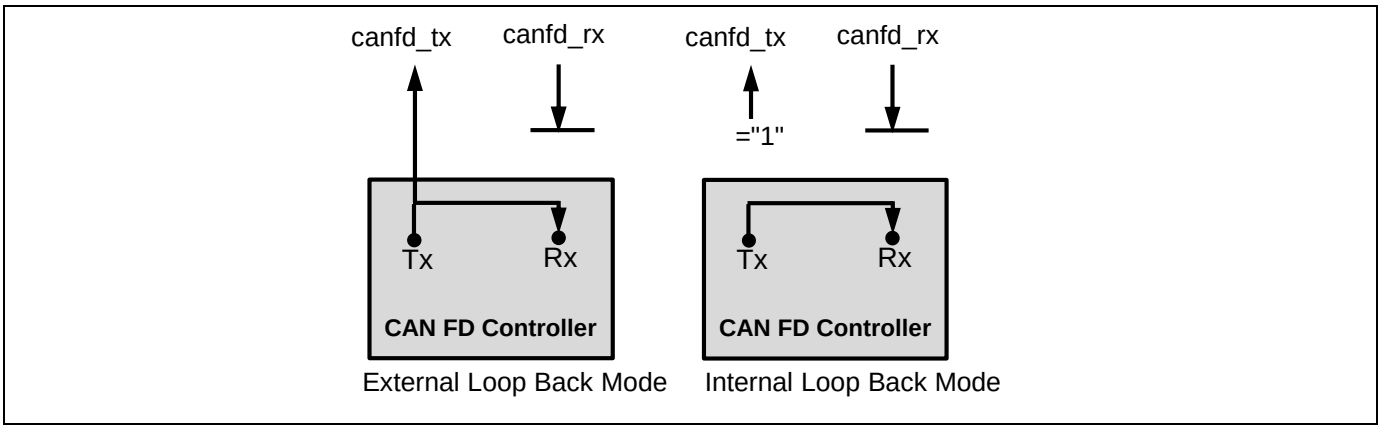
The CAN FD Controller can be set in External Loop Back Mode by programming the Loop Back Mode bit (TEST.LBCK) to one. In External Loop Back Mode, the CAN FD Controller treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a dedicated Rx Buffer or an Rx FIFO. Figure 3-3 shows the connection of signals canfd_tx and canfd_rx to the CAN FD Controller in External Loop Back Mode.

This mode is provided for hardware self test. The CAN FD Controller ignores acknowledge errors in External Loop Back Mode. In this mode the CAN FD Controller performs an internal feedback from its Tx output to its Rx input. The actual value of the canfd_rx input pin is disregarded by the CAN FD Controller. The transmitted messages can be monitored at the canfd_tx pin.

3.1.9.2 Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits Loop Back Mode (TEST.LBCK) and Bus Monitoring Mode (CCCR.MON) to one. This mode can be used for a “Hot Self test”, meaning the CAN FD Controller can be tested without affecting a running CAN system connected to the pins canfd_tx and canfd_rx. In this mode pin canfd_rx is disconnected from the CAN FD Controller and pin canfd_tx is held recessive. Figure 3-3 shows the connection of canfd_tx and canfd_rx to the CAN FD Controller in case of Internal Loop Back Mode.

Figure 3-3 Pin Control in Loop Back Modes



3.2 Timestamp Generation

For timestamp generation the CAN FD Controller supplies a 16-bit wrap-around counter. On start of frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer and FIFO or Tx Event FIFO element (fields RXTS[15:0] and TXTS[15:0] of the respective elements).

For more details, see "5.8.Timestamp Counter Configuration (TSCC)" and "5.9.Timestamp Counter Value (TSCV)".

3.3 Timeout Counter

To signal timeout conditions, the CAN FD Controller supplies a 16-bit Timeout Counter. For Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO triggered operations, it starts down counting when an element is stored, and stops counting when the FIFO is emptied by the CPU. The Timeout Counter may also be triggered by a "0" write to the Initialization bit CCCR.INIT, and count down repeatedly until CCCR.INIT is set back to "1".

The Timeout Counter operates as down-counter and uses the same prescaler controlled by Timestamp Counter Prescaler (TSCC.TCP[3:0]) as the Timestamp Counter. The Timeout Counter is configured via register Timeout Counter Configuration (TOCC). The actual counter value can be read from Timeout Counter (TOCV.TOC[15:0]). The Timeout Counter is operable only when CCCR.INIT = "0". It is stopped when Initialization CCCR.INIT = "1", e.g. when the CAN FD Controller enters Bus_Off state.

The operation mode is selected by the Timeout Select field TOCC.TOS[1:0]. For details, see "5.11.Timeout Counter Value (TOCV) ".

Note:

- *When Timestamp Select TSCC.TSS[1:0] = "01", the clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore, if the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.*

3.4 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the dedicated Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

3.4.1. Acceptance Filtering

3.4.2. Rx FIFOs

3.4.3. Dedicated Rx Buffers

3.4.4. Debug on CAN Support

3.4.5. Protocol Exception Event

3.4.1 Acceptance Filtering

The CAN FD Controller offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to a dedicated Rx Buffer or to Rx FIFO 0, 1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

- Main Features
  - Each filter element can be configured as
    - range filter (from - to)

- filter for one or two specific IDs (dual filter)
- classic bit mask filter
- filter for a single dedicated ID

- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled/disabled individually
- Filters are checked sequentially from filter element 0, execution stops with the first matching filter element

■ Related configuration registers

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

■ Filter Properties

Depending on the configuration of the filter element configuration fields (SFEC[2:0]/EFEC[2:0]) a match triggers one of the following actions:

- Store received frame in Rx FIFO 0 or Rx FIFO 1
- Store received frame in dedicated Rx Buffer
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in Rx FIFO 0 or Rx FIFO 1

■ Description

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching dedicated Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bits to the matching dedicated Rx Buffer or Rx FIFO.

If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected dedicated Rx Buffer or Rx FIFO:

- Dedicated Rx Buffer

  New Data flag NDAT1/2.NDn of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see Last Error Code (PSR.LEC[2:0]) respectively Fast Last Error Code (PSR.FLEC[2:0]).

- Rx FIFO

  Put index of matching Rx FIFO RXFnS.FnPI[5:0] is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC[2:0] respectively PSR.FLEC[2:0].

In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in "3.4.2.2. Rx FIFO Overwrite Mode" have to be considered.

3.4.1.1 Filter Types

■ Range Filter

A range filter matches all received frames with Message IDs in the range defined by the Standard Message ID Filter Element SFID1[10:0]/SFID2[10:0] fields resp. Extended Message ID Filter Element EFID1[28:0]/EFID2[28:0] fields (the range includes the end points SFID1[10:0]/SFID2[10:0] resp. EFID1[28:0]/EFID2[28:0]).

There are two possibilities for the Extended Filter Type of the Extended Message ID Filter Element (EFT[1:0]) when range filtering is used together with extended frames:

- EFT[1:0] = "00":
  The Message ID of received frames is ANDed with the Extended ID AND Mask (XIDAM) before the range filter is applied
- EFT[1:0] = "11":
  The Extended ID AND Mask (XIDAM) is not used for range filtering

■ Filter for specific IDs (Dual Filter)

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SFID1[10:0] = SFID2[10:0] resp. EFID1[28:0] = EFID2[28:0].

■ Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SFID1[10:0]/EFID1[28:0] is used as Message ID filter, while SFID2[10:0]/EFID2[28:0] is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

■ Filter for a single dedicated ID

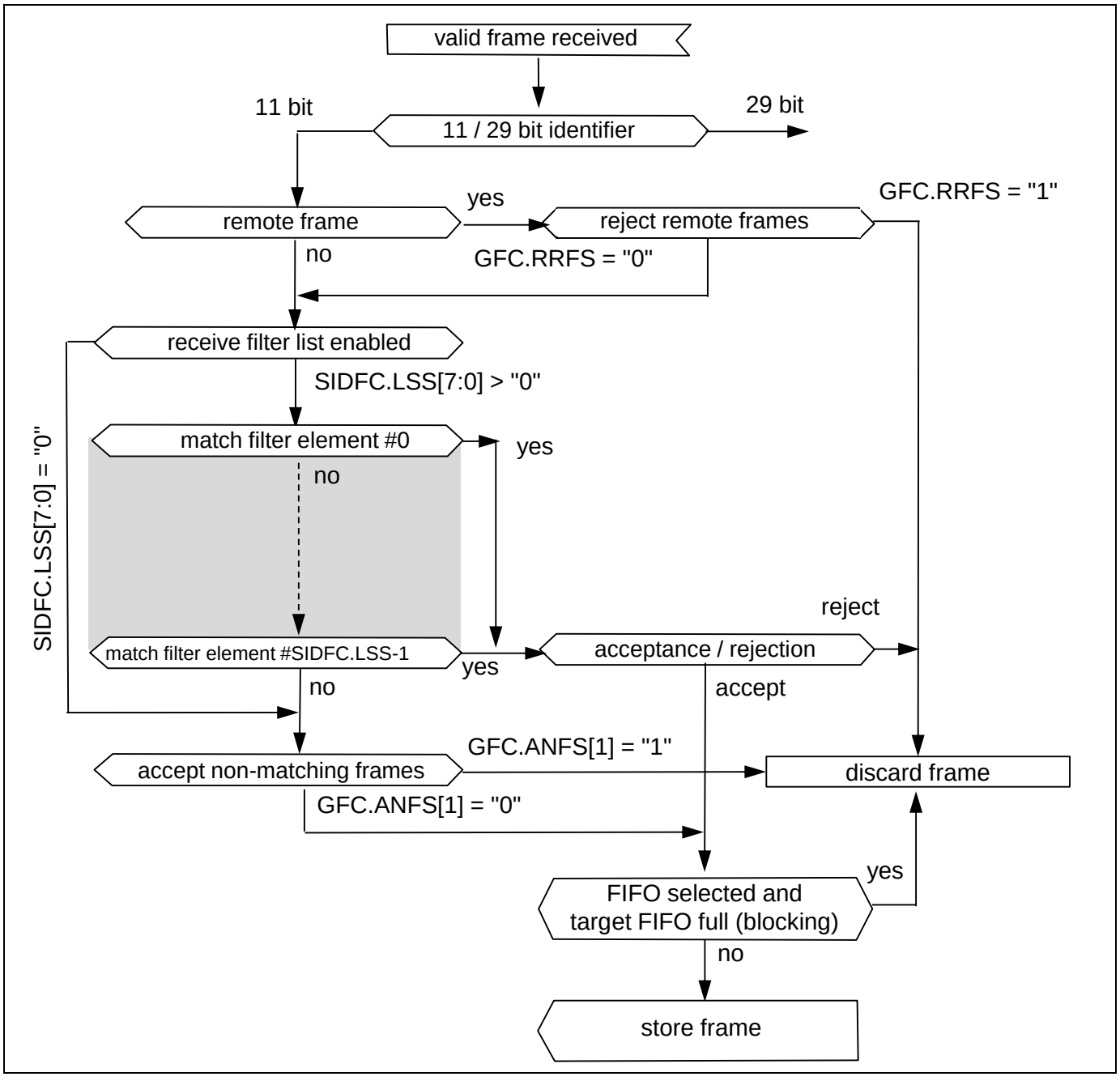
A filter for a single dedicated ID stores received messages in dedicated Rx Buffers if the received ID matches SFID1[10:0]/EFID1[28:0]. SFID2[10:0]/EFID2[28:0] is used to configure the offset to the Rx Buffer Start Address (RXBC.RBSA[15:2]) of where the message is to be stored, and also to define how the message is to be treated (Debug message or Dedicated Rx message).

3.4.1.2 Standard Message ID Filtering

Figure 3-4 below shows the flow for standard Message ID (11-bit Identifier) filtering.

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC, Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 3-4 Standard Message ID Filter Path



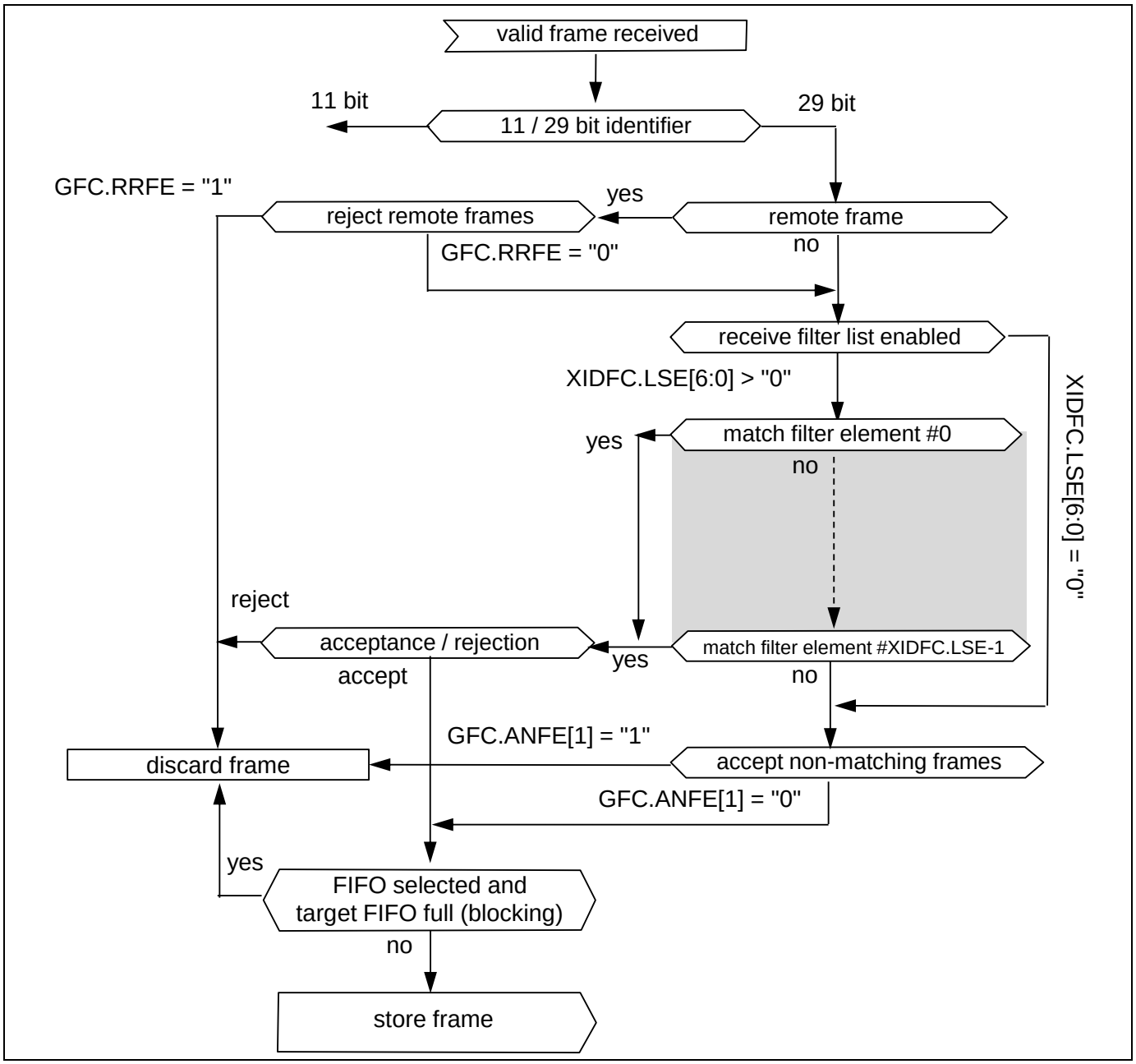
3.4.1.3 Extended Message ID Filtering

Figure 3-5 below shows the flow for extended Message ID (29-bit Identifier) filtering.

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC, Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM.EIDM[28:0] is ANDed with the received identifier before the filter list is executed.

Figure 3-5 Extended Message ID Filter Path



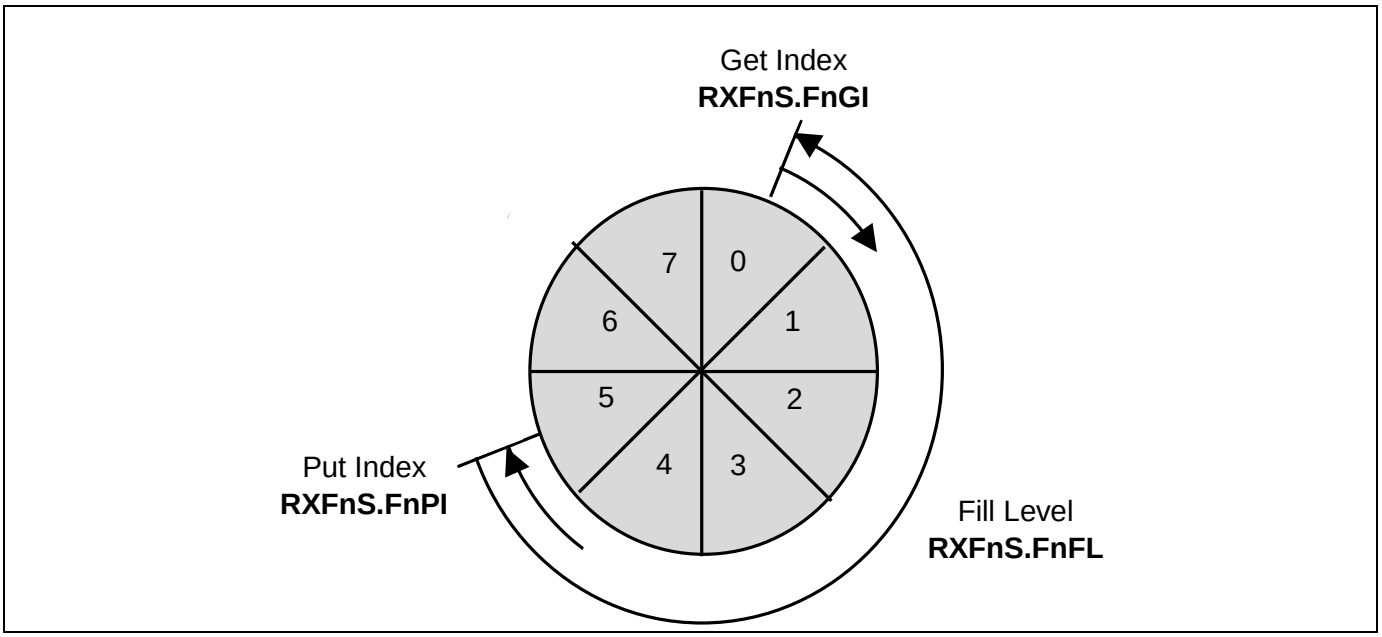
3.4.2 Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via the Rx FIFO 0 and 1 Configuration registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level RXFnS.FnFL[6:0] reaches the Rx FIFO watermark configured by Rx FIFO n Watermark (RXFnC.FnWM[6:0]), interrupt flag Rx FIFO n Watermark Reached (IR.RFnW) is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by Rx FIFO n Full (RXFnS.FnF). In addition interrupt flag Rx FIFO n Full (IR.RFnF) is set.

Figure 3-6 RX FIFO Status



■ Addressing Rx Buffers

An Rx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-2).Therefore when reading from an Rx FIFO, the start address of an Rx Buffer will be

Rx FIFO Get Index (RXFnS.FnGI[5:0]) × Element Size
+corresponding Rx FIFO start address (RXFnC.FnSA[15:2]).

Table 3-2 Rx Buffer/FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

3.4.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by FIFO n Operation Mode RXFnC.FnOM = "0". This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI[5:0] = RXFnS.FnGI[5:0]), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by Rx FIFO n Full (RXFnS.FnF = "1"). In addition interrupt flag Rx FIFO n Full (IR.RFnF) is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by Rx FIFO n Message Lost RXFnS.RFnL = "1". In addition interrupt flag Rx FIFO n Message Lost (IR.RFnL) is set.

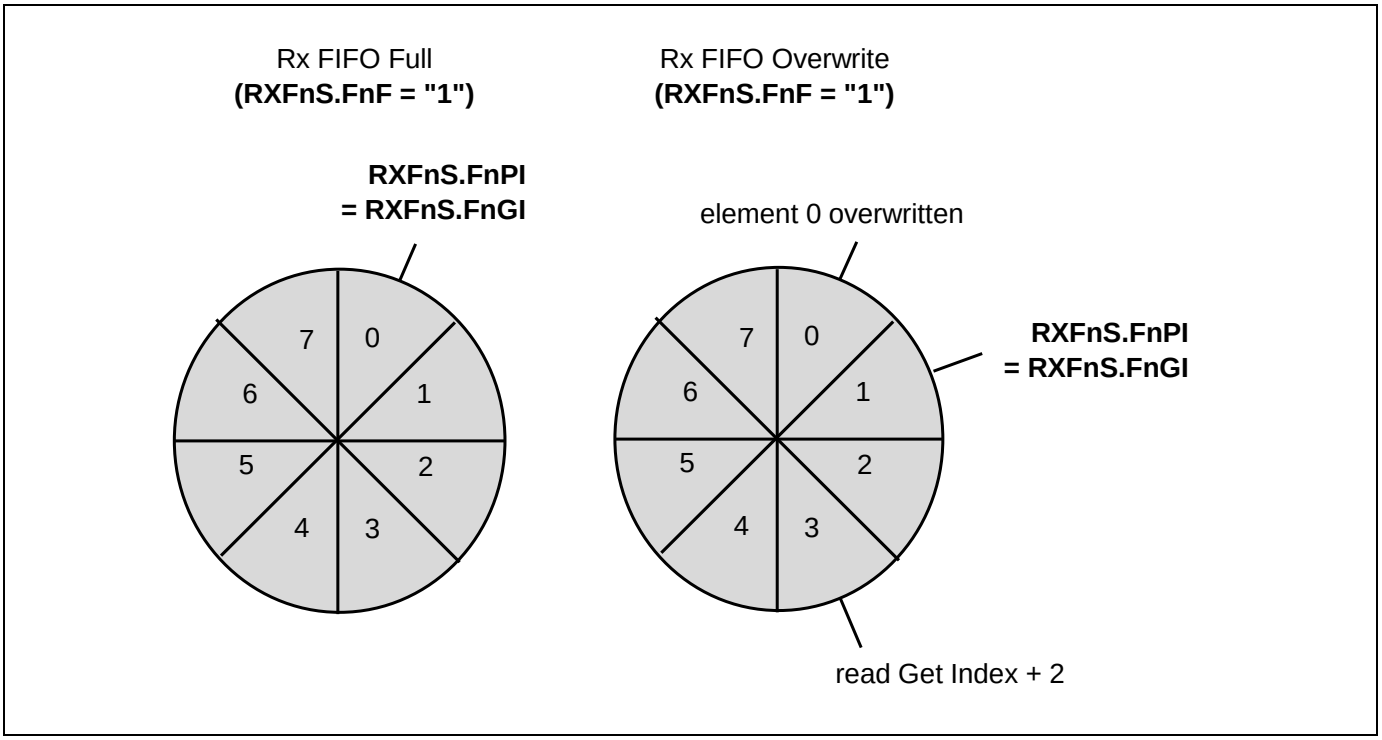
3.4.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by FIFO n Operation Mode RXFnC.FnOM = "1".

When an Rx FIFO full condition (RXFnS.FnPI[5:0] = RXFnS.FnGI[5:0]) is signalled by Rx FIFO n Full RXFnS.FnF = "1", the next message accepted for the Rx FIFO will overwrite the oldest Rx FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. Figure 3-7 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 3-7 RX FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index (RXFnA.FnA[5:0]). This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (Rx FIFO n Full RXFnS.FnF = "0").

3.4.3 Dedicated Rx Buffers

The CAN FD Controller supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via the Rx Buffer Start Address field (RXBC.RBSA[15:2]).

For each dedicated Rx Buffer a Standard or Extended Message ID Filter Element with SFEC[2:0]/EFEC[2:0] = "111" and SFID2[10:9]/EFID2[10:9] = "00" has to be configured.

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR.DRX (Message stored to Dedicated Rx Buffer) in the interrupt register is set.

Table 3-3 Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1, 2 is set. As long as the New Data flag is set, the respective dedicated Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a "1" to the respective bit position.

While a dedicated Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific dedicated Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another dedicated Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

■ Addressing dedicated Rx Buffers

A Dedicated Rx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-2). Therefore the start address of a dedicated Rx Buffer in the Message RAM is calculated by

(Offset defined in Message ID Filter Element SFID2[5:0] resp. EFID2[5:0]) × Element Size + Rx Buffer Start Address (RXBC.RBSA[15:2]).

3.4.4 Debug on CAN Support

For debug handling, three consecutive Rx buffers (e.g. #61, #62, #63) can be used for storage of three specific debug messages A, B, and C. The format is the same as for a dedicated Rx Buffer or an Rx FIFO element.

For filtering of debug messages Standard/Extended Filter Elements with Standard/Extended Filter Configuration SFEC[2:0]/EFEC[2:0] = "111" have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by the lower six bits of the Standard/Extended Message ID Filter Element fields SFID2[5:0]/EFID2[5:0]. The upper two bits SFID2[10:9]/EFID2[10:9] are used to define the Debug Message that is to be stored in the respective Rx Buffer.

When a debug message is stored, neither the respective New Data flag (NDAT1.NDn / NDAT2.NDn) nor the interrupt IR.DRX (Message stored to Dedicated Rx Buffer) are set.

After the three messages have been received in correct order, a DMA transfer is requested. After the DMA transfer has completed, the CAN FD Controller is prepared to receive the next set of debug messages.

Table 3-4 Example Filter Configuration for Debug Messages

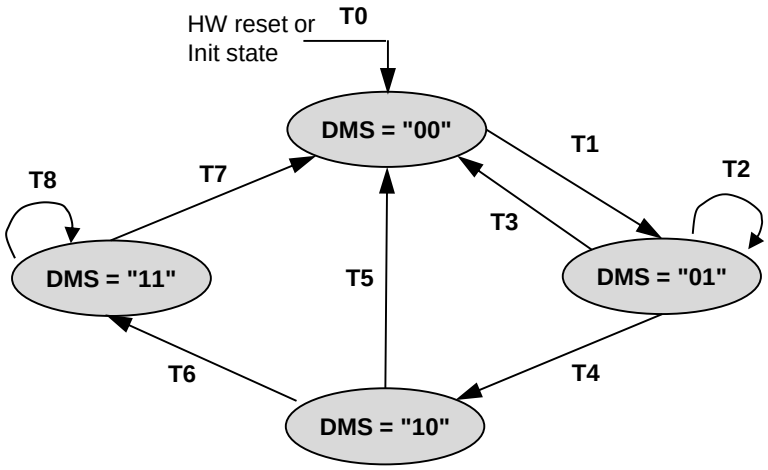
Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID debug message A	01	11 1101
1	ID debug message B	10	11 1110
2	ID debug message C	11	11 1111

3.4.4.1 Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

The status of the debug message handling state machine is signalled via the Debug Message Status field (RXF1S.DMS[1:0]).

Figure 3-8 Debug Message Handling State Machine



T0: reset DMA request, enable reception of debug messages A, B, and C
T1: reception of debug message A
T2: reception of debug message A
T3: reception of debug message C
T4: reception of debug message B
T5: reception of debug messages A, B
T6: reception of debug message C
T7: DMA transfer completed
T8: reception of debug message A, B, C (message rejected)

3.4.5 Protocol Exception Event

When the CAN FD Controller is enabled for CAN FD operation (CCCR.CME ≠ "00") and the CAN FD Controller detects the r0 bit following the EDL bit in a received message to be recessive, it treats this detection as a Protocol Exception Event. As reaction to the Protocol Exception Event,

- the error counters (ECR.REC[6:0] and ECR.TEC[7:0]) are not changed
- hard synchronization is enabled
- the CAN FD Controller will send recessive bits

The CAN FD Controller will also start an internal bit counter that counts the number of recessive bits detected on the CAN bus. If the counter detects eleven consecutive recessive bits, the CAN FD Controller will recover from the Protocol Exception Event and enter an idle state. In case an edge that causes synchronization is detected during the wait period, the CAN FD Controller will reset the internal bit counter and restart counting the number of recessive bits detected.

If the CAN FD Controller

- receives a message directly after the Protocol Exception Event, interrupt flag Message RAM Access Failure (IR.MRAF) will be asserted although the message has been received correctly.
- transmits a message directly after the Protocol Exception Event, IR.MRAF will not be asserted, but that frame will be transmitted with faulty frame format and cause an error frame.

Only the first message after a Protocol Exception Event is affected, all following messages (received or transmitted) will have no problem.

3.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The Tx Buffer element is described in "6.3. Tx Buffer Element".

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

3.5.1. Transmit Pause

3.5.2. Dedicated Tx Buffers

3.5.3. Tx FIFO

3.5.4. Tx Queue

3.5.5. Mixed Dedicated Tx Buffers/Tx FIFO

3.5.6. Mixed Dedicated Tx Buffers/Tx Queue

3.5.7. Transmit Cancellation

3.5.8. Tx Event Handling

3.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU (Electronic Control Unit) sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two nominal bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by the Transmit Pause bit (CCCR.TXP). If the bit is set, the CAN FD Controller will, each time it has successfully transmitted a message, pause for two nominal bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP = "0").

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

3.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

■ Addressing dedicated Tx Buffers

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-5). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by

transmit buffer index (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

Table 3-5 Tx Buffer/FIFO/Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

3.5.3 Tx FIFO

Tx FIFO operation is configured by programming the Tx FIFO/Queue Mode bit (TXBC.TFQM) to "0". Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Tx FIFO Get Index (TXFQS.TFGI[4:0]). After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN FD Controller calculates the number of available (free) Tx FIFO elements TXFQS.TFFL[5:0] (Tx FIFO Free Level) as difference between Tx FIFO Get Index (TXFQS.TFGI[4:0]) and Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]).

■ Adding messages and requesting transmissions

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index (TXFQS.TFQPI[4:0]). An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF = "1") is signalled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a "1" to the Add Request bit (TXBAR.ARn. n is the index of the requested Tx Buffer) related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via the Tx Buffer Add Request register (TXBAR) by requesting all n Tx FIFO Buffers in a single command. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level (TXFQS.TFFL[5:0]). Transmission should not be requested to Tx Buffers outside the Put Index since this will cause the TXFQS.TFFL[5:0] to return a false value.

■ Cancelling transmissions

Transmit Cancellation is intended for Tx Queues and Dedicated Tx Buffers. Cancelling requested Tx FIFO Buffer transmissions are not supported and is not recommended, except for the case where all Tx Buffers within the Tx FIFO are cancelled simultaneously (with a single command).

In the case that a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

■ Addressing Tx Buffers in the Tx FIFO

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 3-5). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by

Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]) (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

3.5.4 Tx Queue

Tx Queue operation is configured by programming the Tx FIFO/Queue Mode bit TXBC.TFQM to "1". Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

■ Adding messages and requesting transmissions

New messages have to be written to the Tx Buffer referenced by the Tx FIFO/Queue Put Index TXFQS.TFQPI[4:0]. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF = "1"), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use the Tx Buffer Request Pending register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

■ Addressing Tx Buffers in the Tx Queue

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 3-5). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by

Tx FIFO/Queue Put Index (TXFQS.TFQPI[4:0]) (0 to 31) × Element Size + Tx Buffers Start Address (TXBC.TBSA[15:2]).

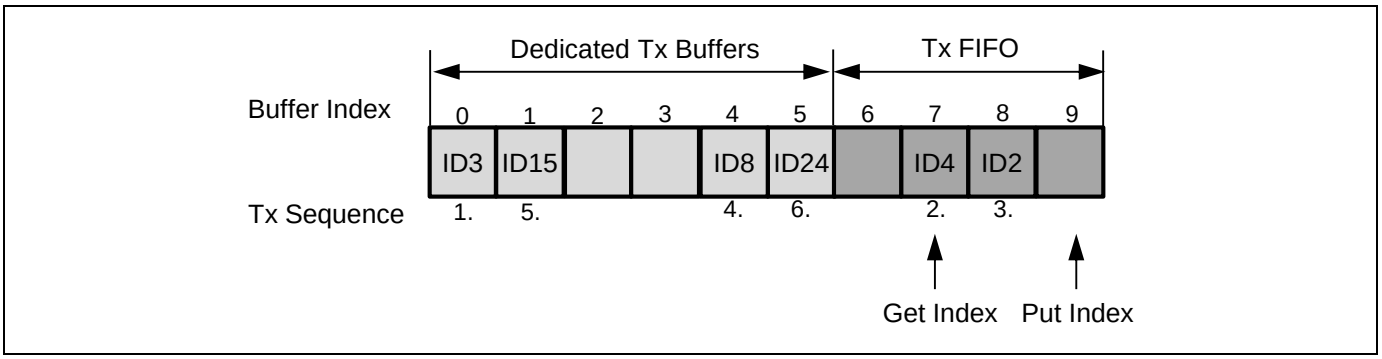
3.5.5 Mixed Dedicated Tx Buffers/Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO.

The number of Dedicated Tx Buffers is configured by TXBC.NDTB[5:0].

The number of Tx Buffers assigned to the Tx FIFO is configured by the Tx FIFO/Queue Size TXBC.TFQS[5:0]. In case TXBC.TFQS[5:0] is programmed to zero, only Dedicated Tx Buffers are used.

Figure 3-9 Example of mixed Configuration Dedicated Tx Buffers/Tx FIFO



Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by the Tx FIFO Get Index TXFQS.TFGI[4:0])
- Buffer with lowest Message ID gets highest priority and is transmitted next

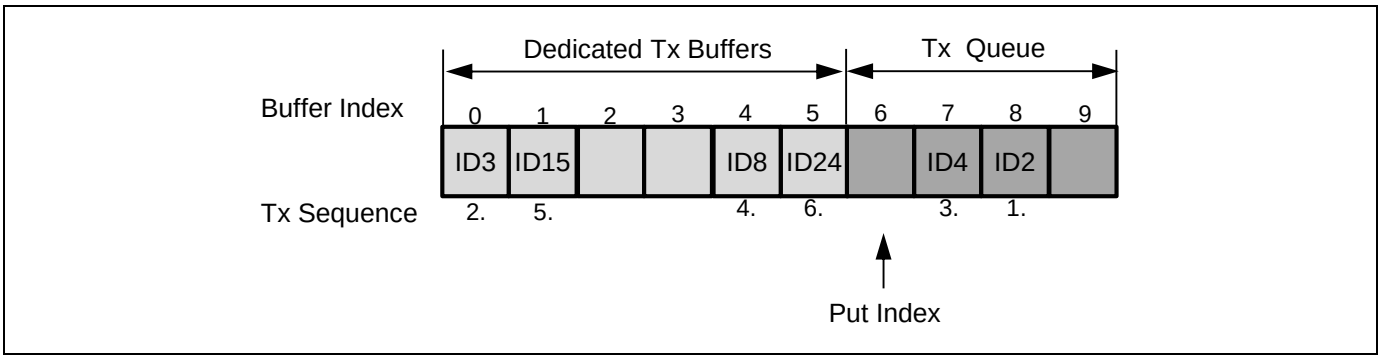
3.5.6 Mixed Dedicated Tx Buffers/Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue.

The number of Dedicated Tx Buffers is configured by TXBC.NDTB[5:0].

The number of Tx Queue Buffers is configured by the Tx FIFO/Queue Size TXBC.TFQS[5:0]. In case TXBC.TFQS[5:0] is programmed to zero, only Dedicated Tx Buffers are used.

Figure 3-10 Example of mixed Configuration Dedicated Tx Buffers/Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

3.5.7 Transmit Cancellation

The CAN FD Controller supports transmit cancellation. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a "1" to the corresponding bit position of the Tx Buffer Cancellation Request register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of the Tx Buffer Cancellation Finished register TXBCF to "1".

A cancellation request resets the corresponding Transmission Request Pending bit (TXBRP.TRPn). In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP.TRPn bit remains set as long as the transmission is in progress.

- If the transmission was successful, the corresponding Transmission Occurred (TXBTO.TOn) and Cancellation Finished (TXBCF.CFn) bits are set.
- If the transmission was not successful (the transmission has been aborted due to lost arbitration, error occurred during frame transmission), it is not repeated and only the corresponding Cancellation Finished bit (TXBCF.CFn) is set.

TXBCF.CFn is also set when the transmission has not yet been started at the point of cancellation,

The cancellation request bits (TXBCR.CRn) are reset directly after the corresponding Transmission Request Pending bit (TXBRP.TRPn) has been reset.

Note:

- *In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority (a greater message ID) than the second message in this node.*

3.5.8 Tx Event Handling

To support Tx event handling the CAN FD Controller has implemented a Tx Event FIFO. After the CAN FD Controller has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer MM[7:0] is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured by Event FIFO Size (TXEFC.EFS[5:0]) to a maximum of 32 elements.

When a Tx Event FIFO full condition is signalled by interrupt flag Tx Event FIFO Full (IR.TEFF), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index TXEFS.EFGI[4:0] has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag Tx Event FIFO Element Lost IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level TXEFS.EFFL[5:0] reaches the Tx Event FIFO watermark configured by TXEFC.EFWM[5:0], interrupt flag Tx Event FIFO Watermark Reached IR.TEFW is set.

■ Addressing Tx Events in the Tx Event FIFO

The start address when reading from the Tx Event FIFO will be

2 × Tx Event FIFO Get Index TXEFS.EFGI[4:0] + Tx Event FIFO start address TXEFC.EFSA[15:2].

3.6 FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO (RXF0S.F0GI[5:0], RXF1S.F1GI[5:0], and TXEFS.EFGI[4:0], respectively) are controlled by writing to the corresponding FIFO Acknowledge Index (RXF0A.F0AI[5:0], RXF1A.F1AI[5:0], and TXEFA.EFAI[4:0], respectively). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

- When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index. (Multiple elements may be read by repeating this operation)
- When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the Rx FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position, alter the Rx FIFO's Fill Level, and also cause some of the older Rx FIFO elements to be lost.

Note:

- *The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN FD Controller does not check for erroneous values.*

3.7 Configuring the CAN Bit Timing

Each CAN node in the CAN network has its own clock generator (usually a quartz oscillator). The time parameter of the bit time can be configured individually for each CAN node. Even if each CAN node's oscillator has a different period, a common bit rate can be generated.

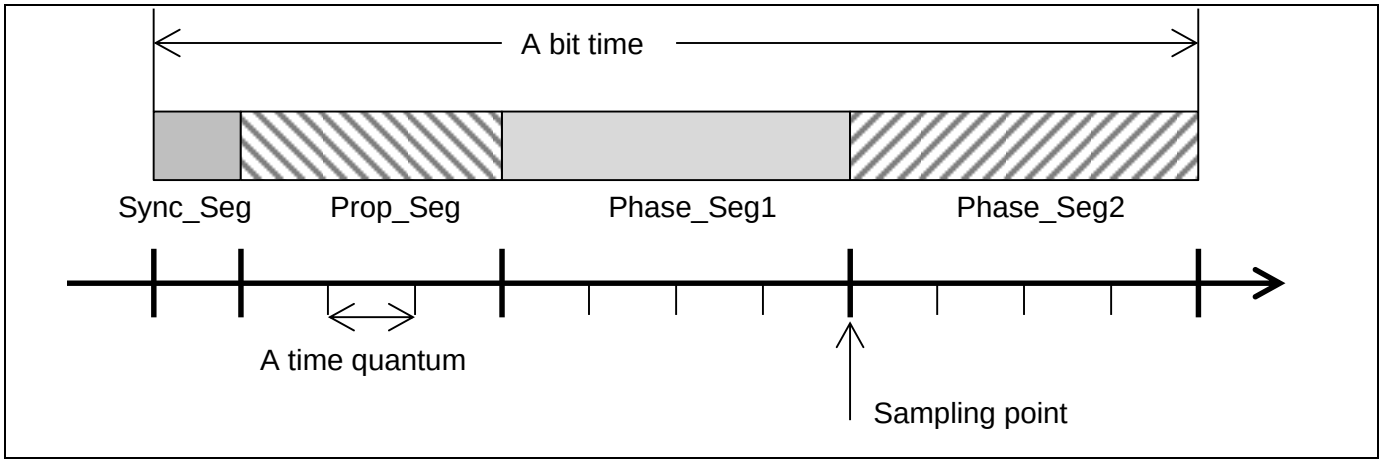
The oscillator frequencies vary slightly because of changes in temperature or voltage, or deterioration of components. As long as the frequencies vary only within the tolerance range of the oscillators, the CAN nodes can compensate for the different bit rates by resynchronizing to the bit stream.

3.7.1 CAN Bit Timing

The CAN FD specification V1.0 defines two bit times - nominal bit time and fast bit time (the CAN FD specification V1.0 calls these nominal bit time and data bit time). The nominal bit time is for the arbitration phase. The fast bit time has an equal or shorter length and can be used to accelerate the data phase (see Bit Rate Switching in "3.1.3.CAN FD Operation" for details).

The basic construction of a bit time is shared with both the nominal and fast bit times. The bit time can be divided into four segments according to the CAN specifications (see Figure 3-11): the synchronization segment (Sync_Seg), the propagation time segment (Prop_Seg), the phase buffer segment 1 (Phase_Seg1), and the phase buffer segment 2 (Phase_Seg2). The sample point, the point of time at which the bus level is read and interpreted as the value of that respective bit, is located at the end of Phase_Seg1.

Figure 3-11 Bit time construction



Each segment consists of a programmable number of time quanta, which is a multiple of the time quantum that is defined by canfd_cclk and a prescaler. The values and prescalers used to define these parameters differ for the nominal and fast bit times, and are configured by BTP (Bit Timing & Prescaler Register) and FBTP (Fast Bit Timing & Prescaler Register) as shown in the table below (Table 3-6).

Table 3-6 Bit time parameters

Parameter	Description
Time quantum tq (nominal) and tqf (fast)	Time quantum. Derived by multiplying the basic unit time quanta (i.e. the canfd_cclk period) with the respective prescaler. The time quantum is configured by the CAN FD Controller as nominal : tq = (BTP.BRP[9:0] +1) × canfd_cclk period fast : tqf = (FBTP.FBRP[4:0] + 1) × canfd_cclk period
Sync_Seg	Sync_Seg is fixed to 1 time quantum as defined by the CAN specifications and is therefore not configurable (inherently built into the CAN FD Controller). nominal : 1 tq fast : 1 tqf
Prop_Seg	Prop_Seg is the part of the bit time that is used to compensate for the physical delay times within the network. The CAN FD Controller configures the sum of Prop_Seg and Phase_Seg1 with a single parameter, i.e. nominal : Prop_Seg + Phase_Seg1 = BTP.TSEG1[5:0] + 1 fast : Prop_Seg + Phase_Seg1 = FBTP.FTSEG1[3:0] + 1
Phase_Seg1	Phase_Seg1 is used to compensate for edge phase errors before the sampling point. Can be lengthened by the resynchronization jump width. The sum of Prop_Seg and Phase_Seg1 is configured by the CAN FD Controller as nominal : BTP.TSEG1[5:0] + 1 fast : FBTP.FTSEG1[3:0] + 1
Phase_Seg2	Phase_Seg2 is used to compensate for edge phase errors after the sampling point. Can be shortened by the resynchronization jump width. Phase_Seg2 is configured by the CAN FD Controller as nominal : BTP.TSEG2[3:0] + 1 fast : FBTP.FTSEG2[2:0] + 1
SJW	Resynchronization Jump Width. Used to adjust the length of Phase_Seg1 and Phase_Seg2. SJW will not be longer than either Phase_Seg1 or Phase_Seg2. SJW is configured by the CAN FD Controller as nominal : BTP.SJW[3:0] + 1 fast : FBTP.FSJW[1:0] + 1

These relations result in the following equations for the nominal and fast bit times:

Nominal Bit time

= [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] × tq
= [1 + (BTP.TSEG1[5:0] + 1) + (BTP.TSEG2[3:0] + 1)] × [(BTP.BRP[9:0] +1) × canfd_cclk period]

and for the fast bit time

= [1 + (FBTP.FTSEG1[3:0] + 1) + (FBTP.FTSEG2[2:0] + 1)] × [(FBTP.FBRP[4:0] +1) × canfd_cclk period]

Note:

- *The Information Processing Time (IPT) of the CAN FD Controller is zero, meaning the data for the next bit is available at the first CAN clock edge after the sample point. Therefore, the IPT does not have to be accounted for when configuring Phase_Seg2, which is the maximum of Phase_Seg1 and the IPT.*

3.7.2 CAN Bit Rates

Since the bit rate is the inverse of the bit time, the nominal bit rate is

1 / [{1 + (BTP.TSEG1[5:0] + 1) + (BTP.TSEG2[3:0] + 1)} × {(BTP.BRP[9:0] +1) × canfd_cclk period}]

and the FD bit rate is

1 / [{1 + (FBTP.FTSEG1[3:0] + 1) + (FBTP.FTSEG2[2:0] + 1)} × {(FBTP.FBRP[4:0] +1) × canfd_cclk period}]

From these formula we can see that the bit rates of the CAN FD Controller will depend on the CAN clock (i.e. canfd_cclk) period, and the range each parameter can be configured to. The tables below lists examples of the configurable bit rates at varying CAN clock frequencies. Empty boxes indicate that the desired bit rate cannot be configured at the specified input CAN clock frequency.

Table 3-7 Example configurations for nominal bit rates

CAN clock frequency	8MHz		10MHz		16MHz		20MHz		32MHz		40MHz	
configuration / nominal bit rate	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1	# of tqs	BTP.BRP + 1
125Kbps	64tq	1	80tq	1	64tq	2	80tq	2	64tq	4	80tq	4
	32tq	2	40tq	2	32tq	4	40tq	4	32tq	8	40tq	8
	16tq	4	20tq	4	16tq	8	20tq	8	16tq	16	20tq	16
	8tq	8	10tq	8	8tq	16	10tq	16	8tq	32	10tq	32
250Kbps	32tq	1	40tq	1	64tq	1	80tq	1	64tq	2	80tq	2
	16tq	2	20tq	2	32tq	2	40tq	2	32tq	4	40tq	4
	8tq	4	10tq	4	16tq	4	20tq	4	16tq	8	20tq	8
					8tq	8	10tq	8	8tq	16	10tq	16
500Kbps	16tq	1	20tq	1	32tq	1	40tq	1	64tq	1	80tq	1
	8tq	2	10tq	2	16tq	2	20tq	2	32tq	2	40tq	2
					8tq	4	10tq	4	16tq	4	20tq	4
									8tq	8	10tq	8
1Mbps	8tq	1	10tq	1	16tq	1	20tq	1	32tq	1	40tq	1
					8tq	2	10tq	2	16tq	2	20tq	2
									8tq	4	10tq	4

Table 3-8 Example configurations for FD bit rates

CAN clock frequency	8MHz		10MHz		16MHz		20MHz		32MHz		40MHz	
configuration / FD bit rate	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1	# of tqfs	FBTP.FBRP+1
500Kbps	16tqf 8tqf	1 2	20tqf 10tqf	1 2	16tqf 8tqf	2 4	20tqf 10tqf	2 4	16tqf 8tqf	4 8	20tqf 10tqf	4 8
1Mbps	8tqf	1	10tqf	1	16tqf 8tqf	1 2	20tqf 10tqf	1 2	16tqf 8tqf	2 4	20tqf 10tqf	2 4
2Mbps					8tqf	1	10tqf	1	16tqf 8tqf	1 2	20tqf 10tqf	1 2
4Mbps									8tqf	1	10tqf	1
5Mbps											8tqf	1

Note:

- *The user must configure the CAN bit timings so they comply with the corresponding CAN standards to ensure proper communication on the CAN bus.*

4. Setup procedure examples

This section provides examples of setup procedures for the CAN FD Controller.

4.1. Configuration of CAN bus

4.2. Configuration of Message RAM

- 4.2.1. Configuration of ID Filter List
- 4.2.2. Configuration of Rx Buffer and Rx FIFO
- 4.2.3. Configuration of Tx Buffer and Tx FIFO/Queue
- 4.2.4. Configuration of ID Filter

4.3. Configuration of Error Monitor

4.4. Configuration of Interrupt

4.5. Control of Communication

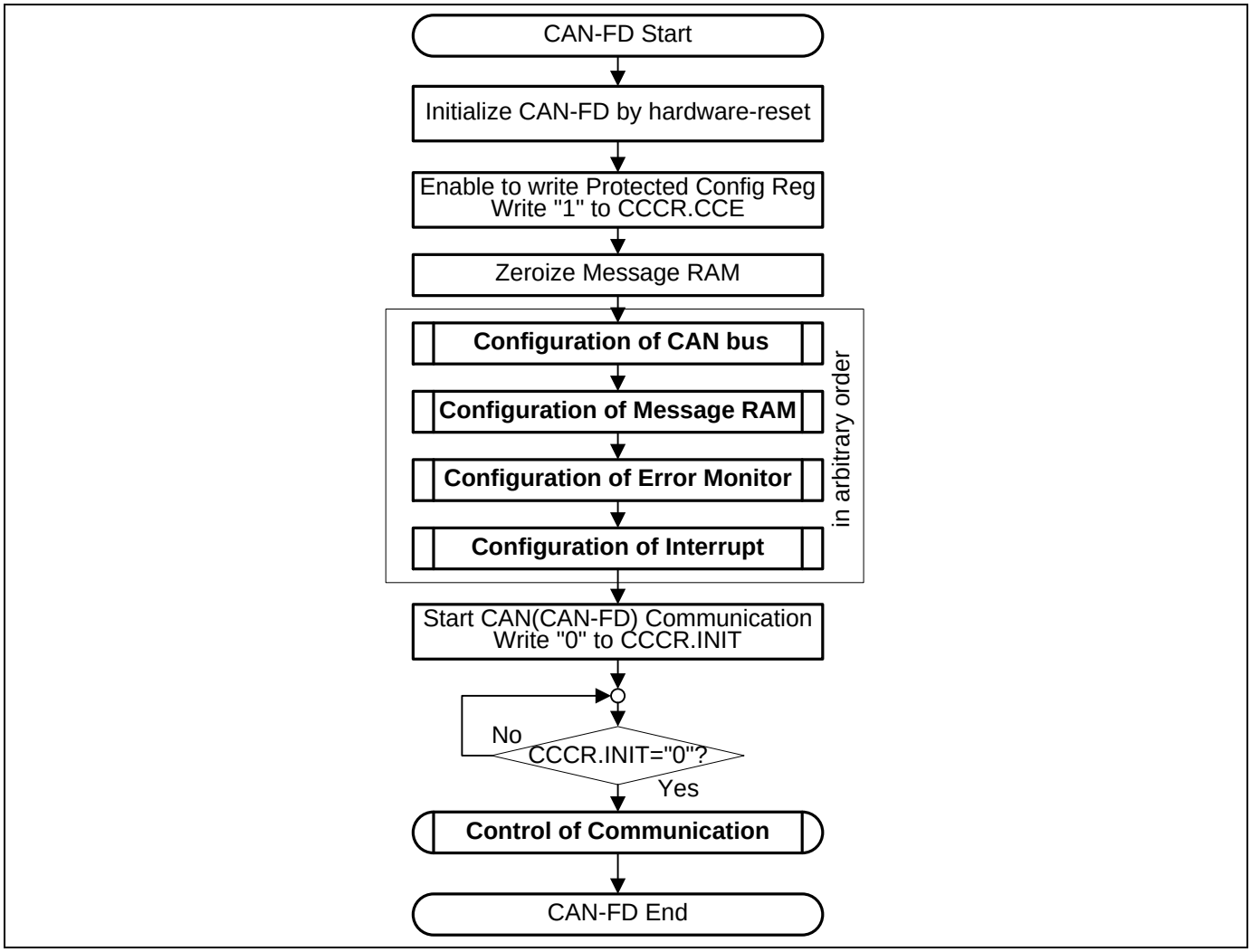
- 4.5.1. CAN Transmit Mode Change
- 4.5.2. Configuration of Transmission Frame

4.6. Interrupt Handling Operation

- 4.6.1. Bus_Off status Handling Operation
- 4.6.2. Message RAM access failure Handling Operation
- 4.6.3. Bit error Handling Operation
- 4.6.4. Tx Event FIFO Handling Operation
- 4.6.5. Dedicated Rx Buffer Handling Operation
- 4.6.6. High priority Message Handling Operation
- 4.6.7. Rx FIFO Handling Operation

Figure 4-1 shows the general flow of a program. The following sections will describe each step in more detail.

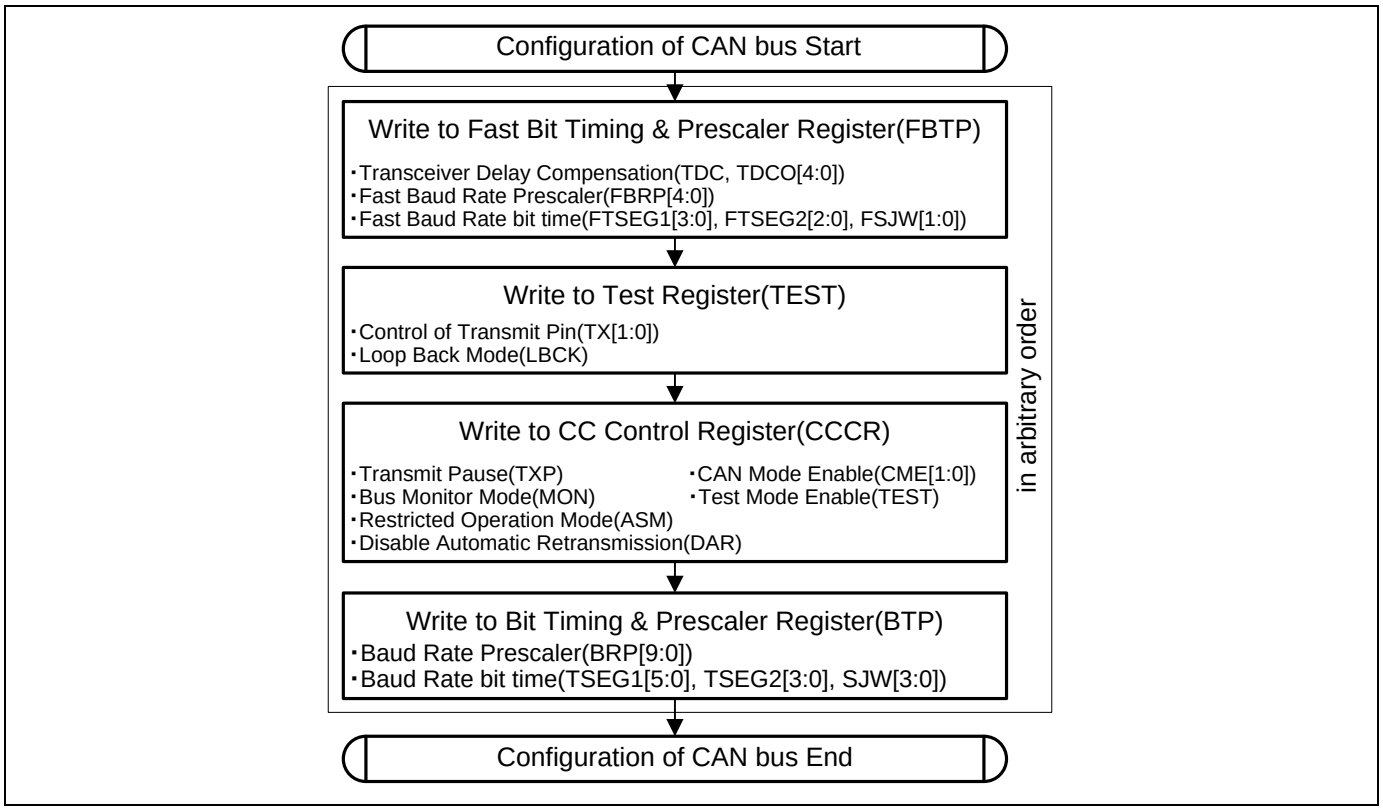
Figure 4-1 General Program Flow



4.1 Configuration of CAN bus

Figure 4-2 lists what to configure in order to set up the CAN bus related functions.

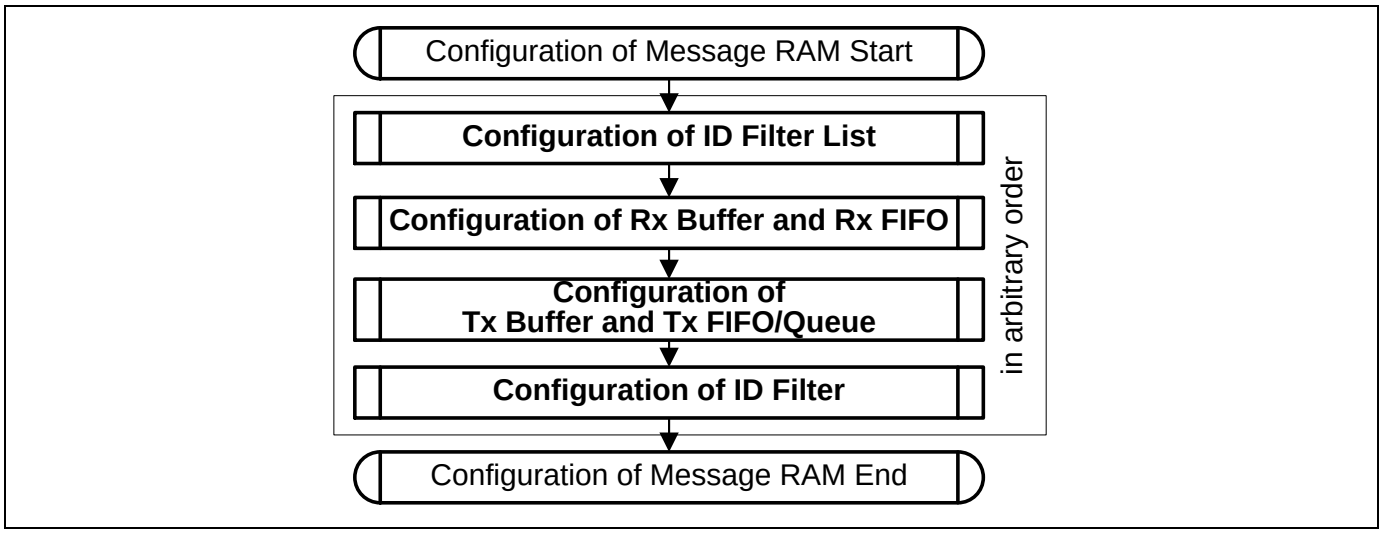
Figure 4-2 Flow diagram "Configuration of CAN bus"



4.2 Configuration of Message RAM

Figure 4-3 shows the overall configuration procedure of the Message RAM.

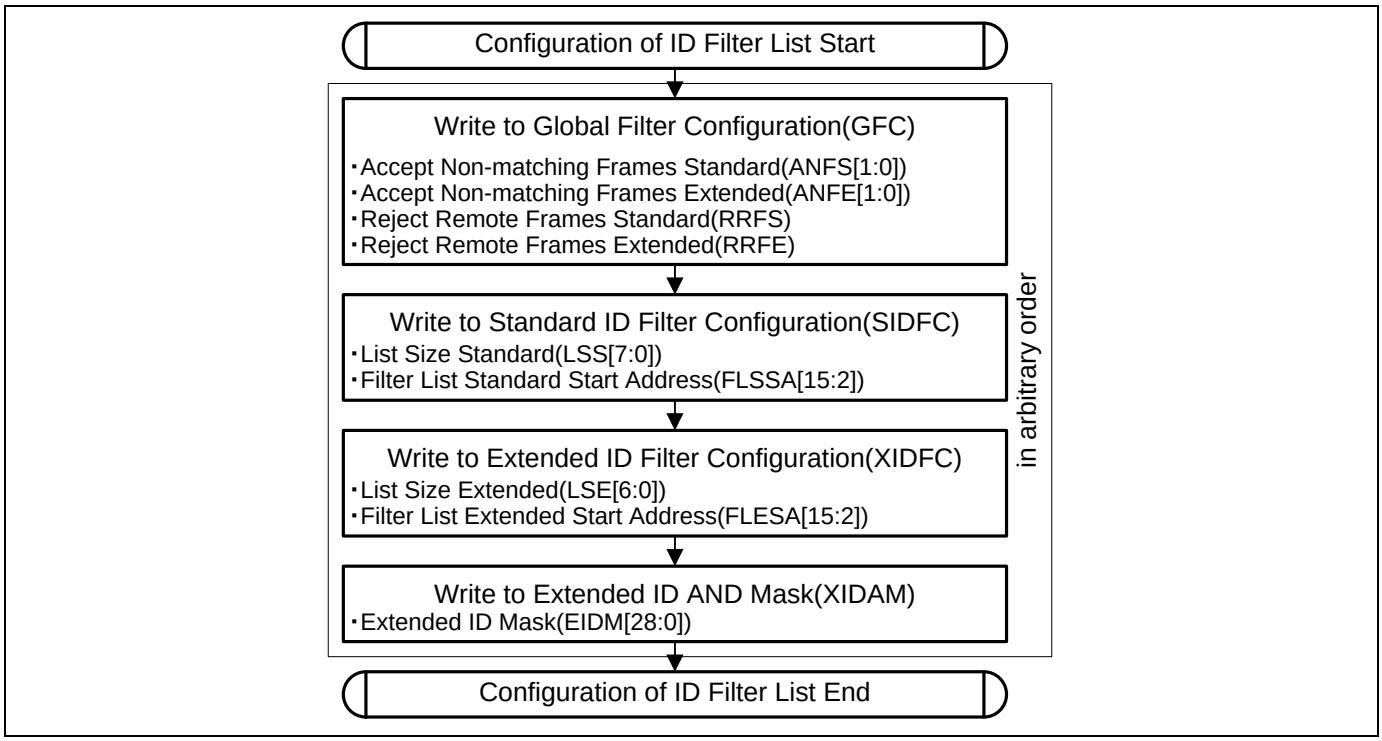
Figure 4-3 Flow diagram "Configuration of Message RAM"



4.2.1 Configuration of ID Filter List

Figure 4-4 lists what to configure in order to set up the ID Filter lists.

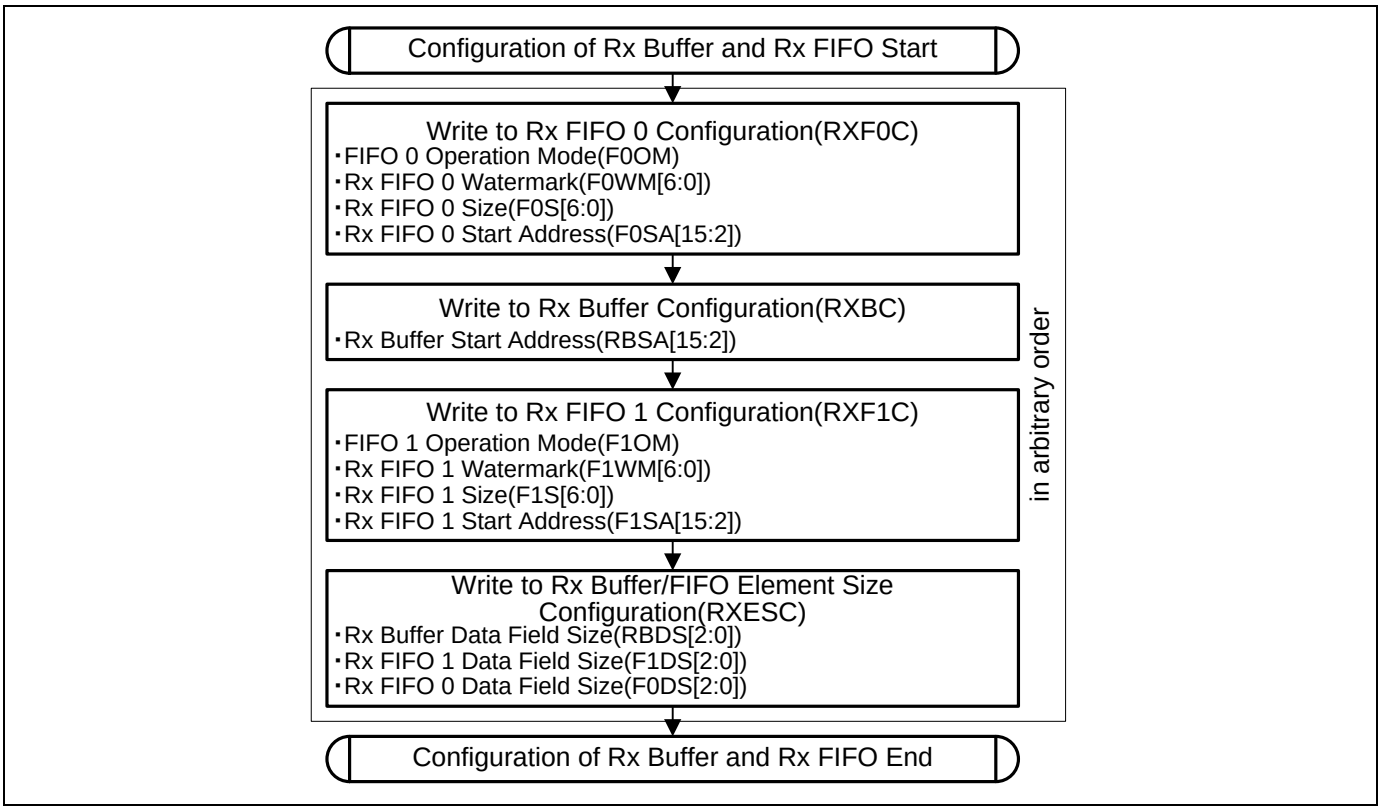
Figure 4-4 Flow diagram "Configuration of ID Filter List"



4.2.2 Configuration of Rx Buffer and Rx FIFO

Figure 4-5 lists what to configure in order to set up the Dedicated Rx Buffers and Rx FIFOs.

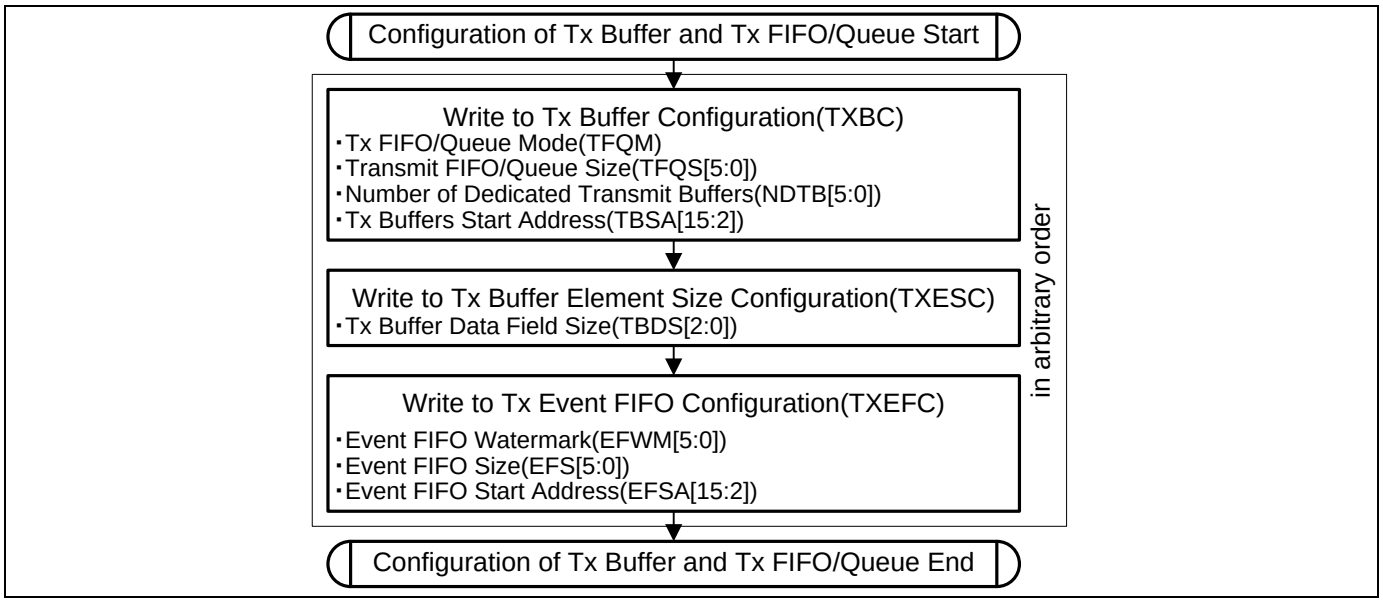
Figure 4-5 Flow diagram "Configuration of Rx Buffer and Rx FIFO"



4.2.3 Configuration of Tx Buffer and Tx FIFO/Queue

Figure 4-6 lists what to configure in order to set up the Tx Buffers and Tx Event FIFO.

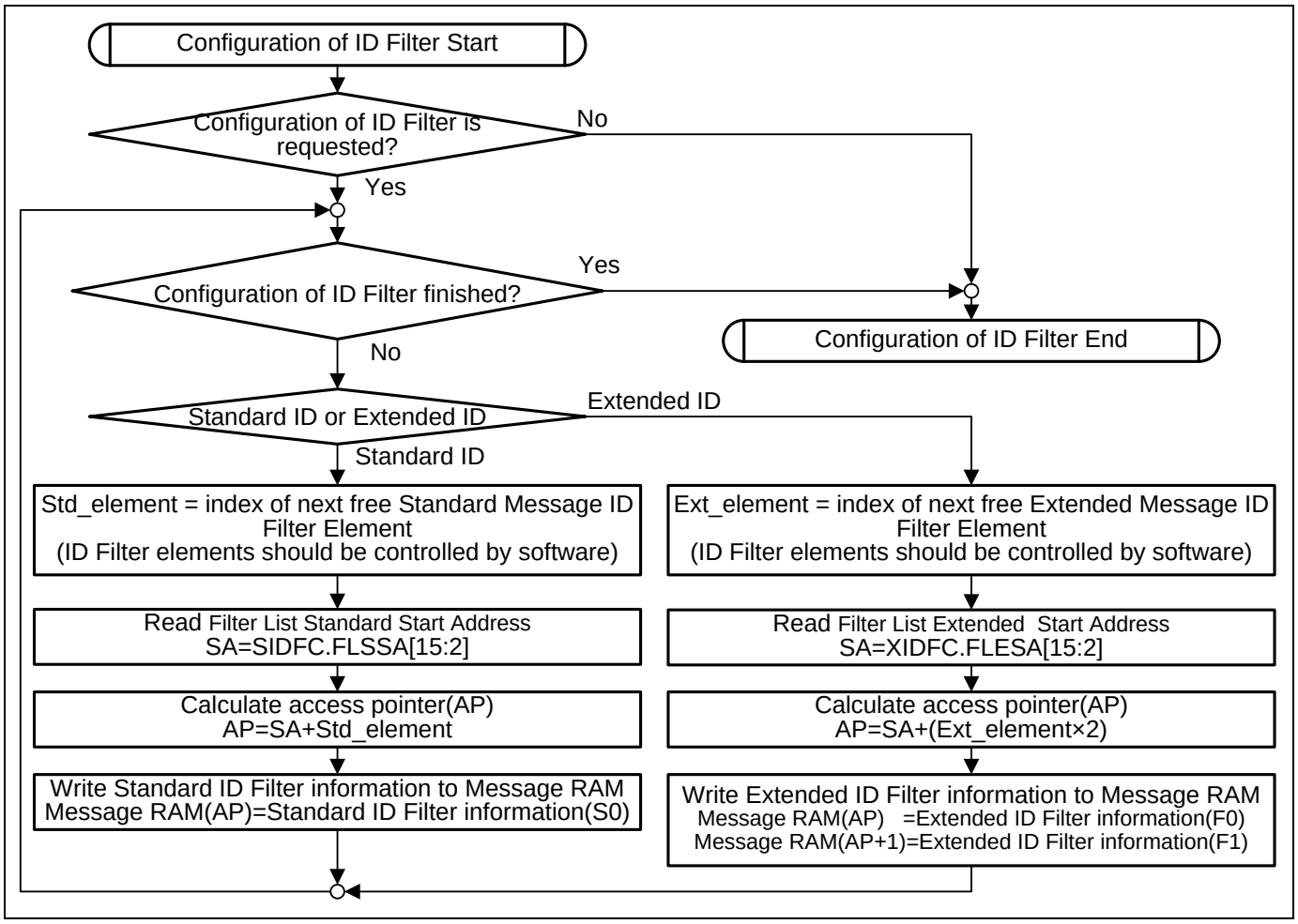
Figure 4-6 Flow diagram "Configuration of Tx Buffer and Tx FIFO/Queue"



4.2.4 Configuration of ID Filter

Figure 4-7 shows the steps to set up an ID Filter.

Figure 4-7 Flow diagram "Configuration of ID Filter"



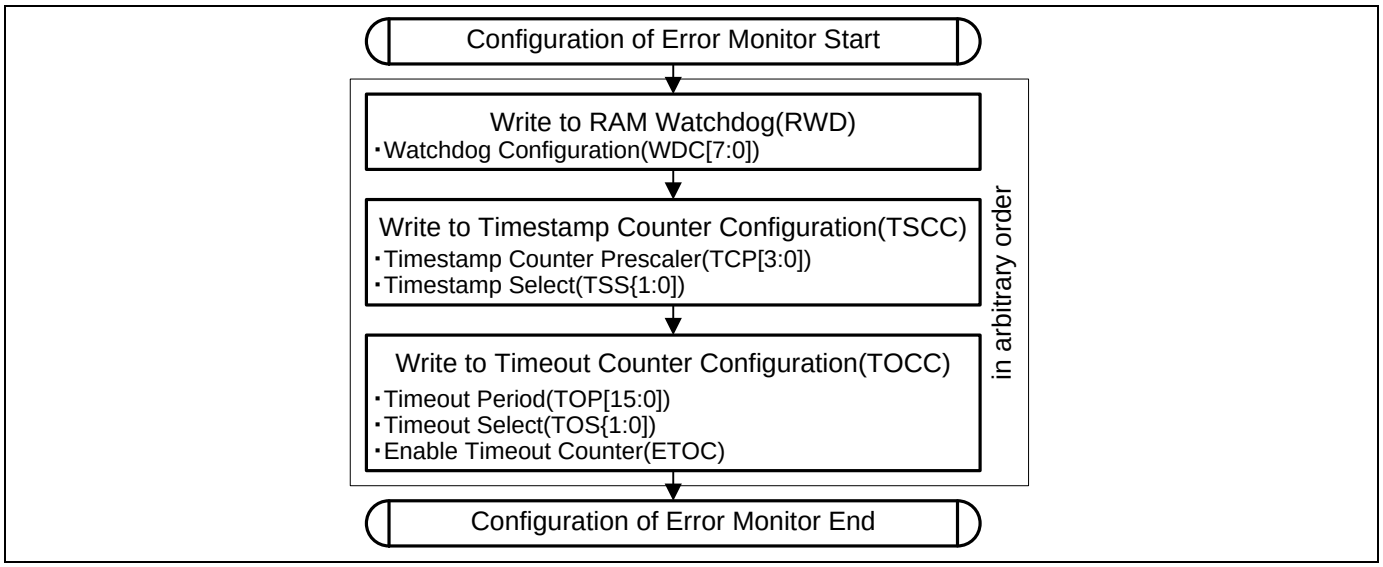
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.3 Configuration of Error Monitor

Figure 4-8 lists the error monitoring functions that can be used.

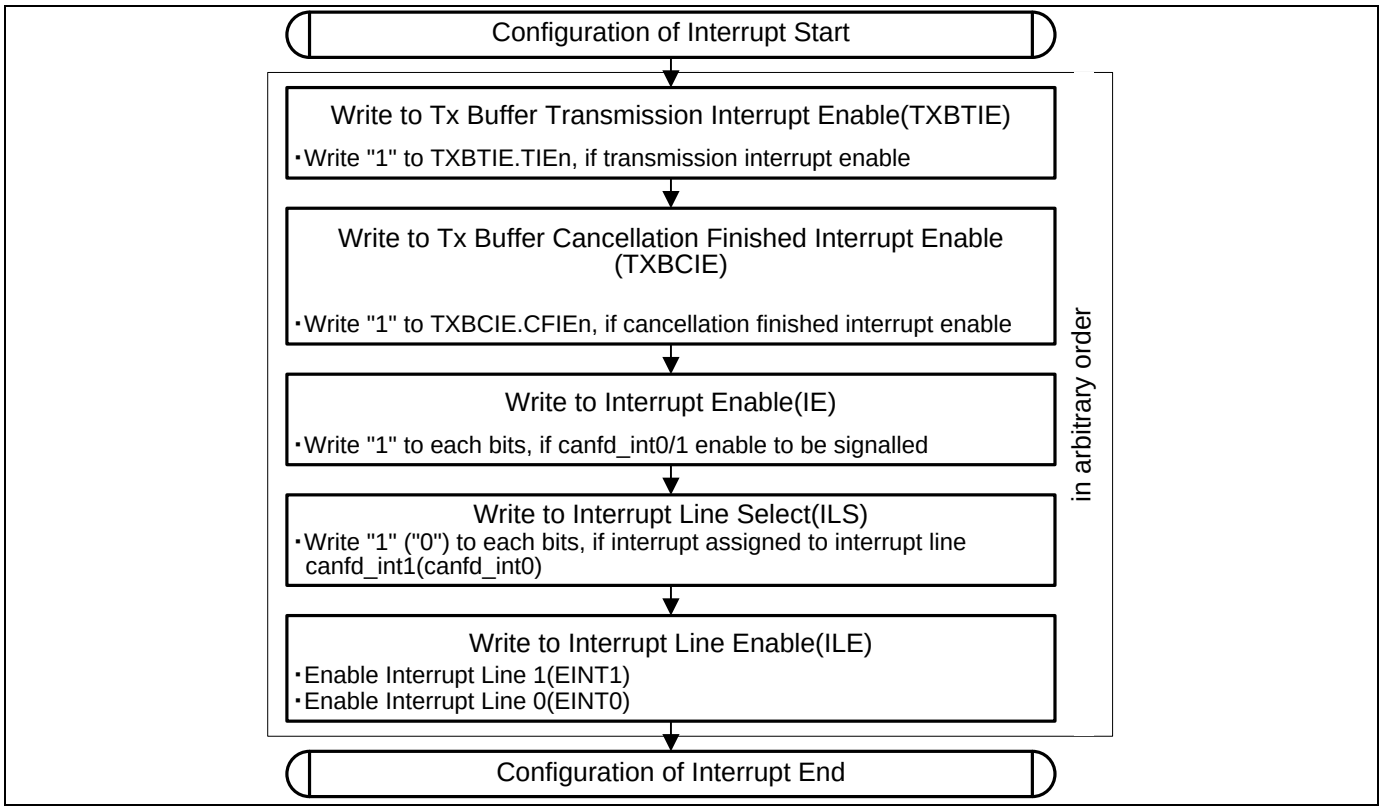
Figure 4-8 Flow diagram "Configuration of Error Monitor"



4.4 Configuration of Interrupt

Figure 4-9 lists what to configure in order to set up the Interrupt related logic.

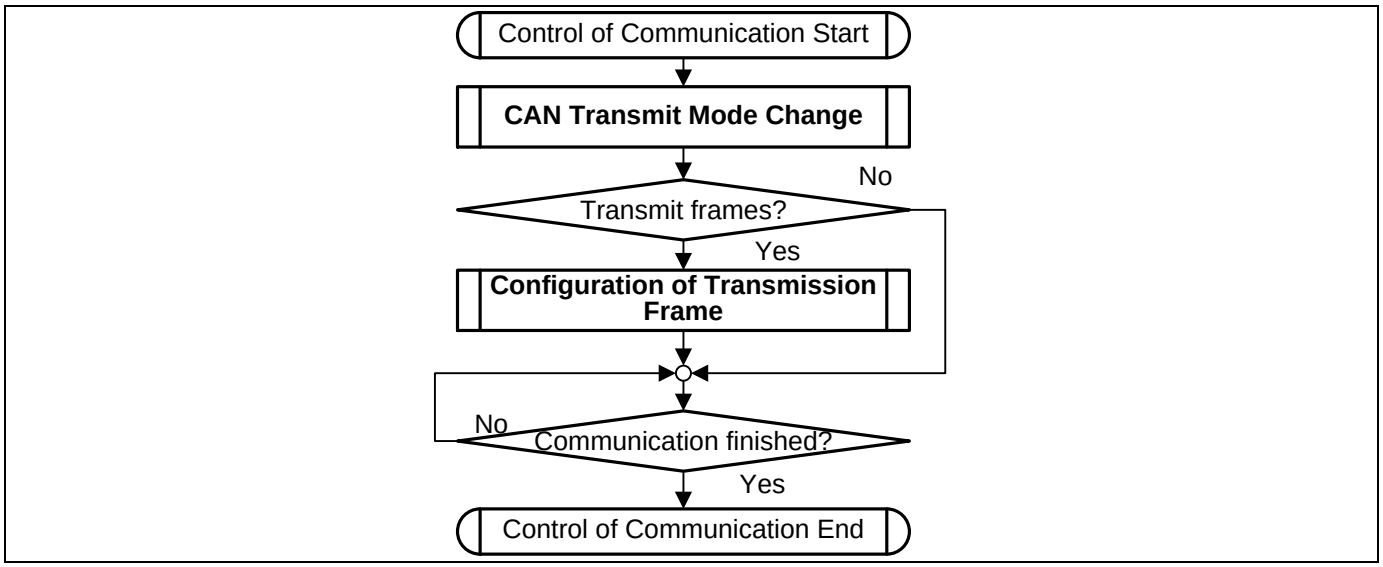
Figure 4-9 Flow diagram "Configuration of Interrupt"



4.5 Control of Communication

Figure 4-10 shows the overall configuration procedure for communication on the CAN bus.

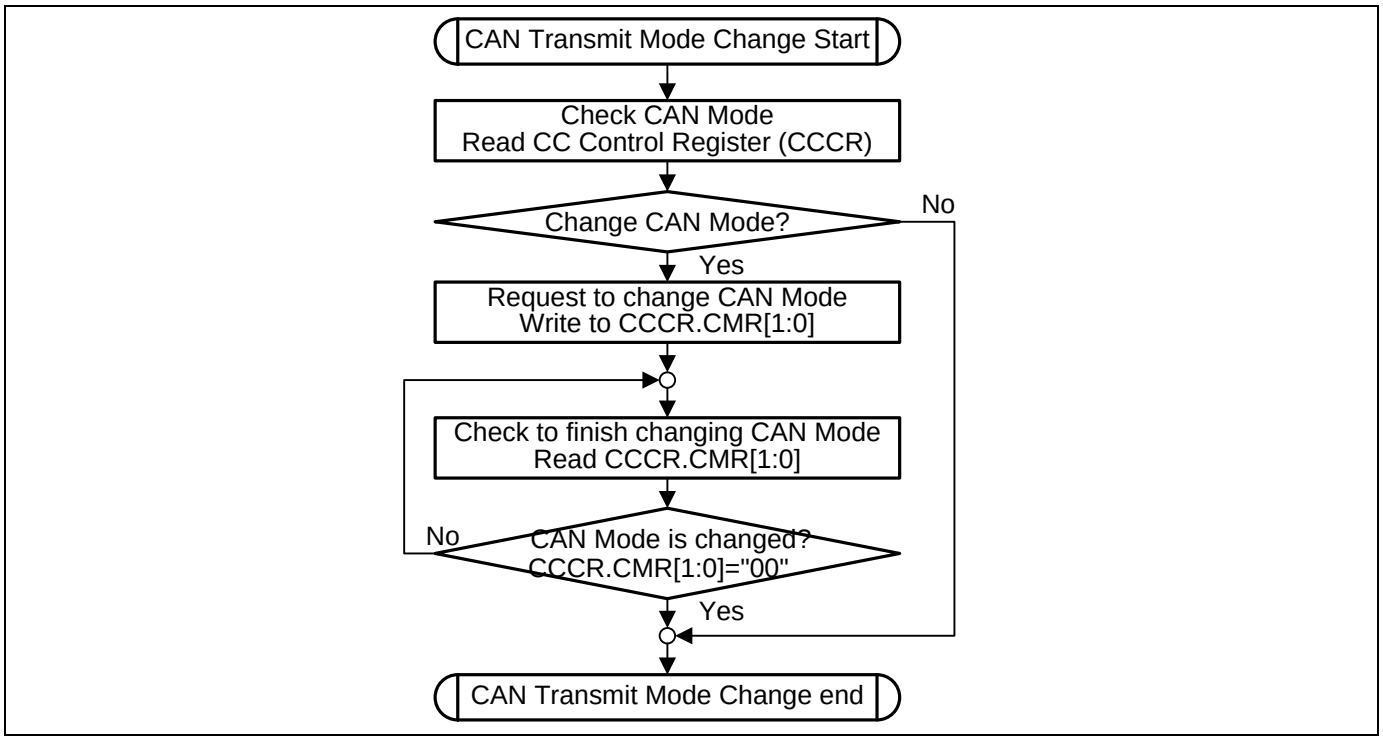
Figure 4-10 Flow diagram "Control of Communication"



4.5.1 CAN Transmit Mode Change

Figure 4-11 shows how the CAN transmission properties CCCR.FDBS (CAN FD Bit Rate Switching) and CCCR.FDO (CAN FD Operation) can be changed for a specific CAN Mode Enable (CCCR.CME[1:0]) setting.

Figure 4-11 Flow diagram "CAN Transmit Mode Change"



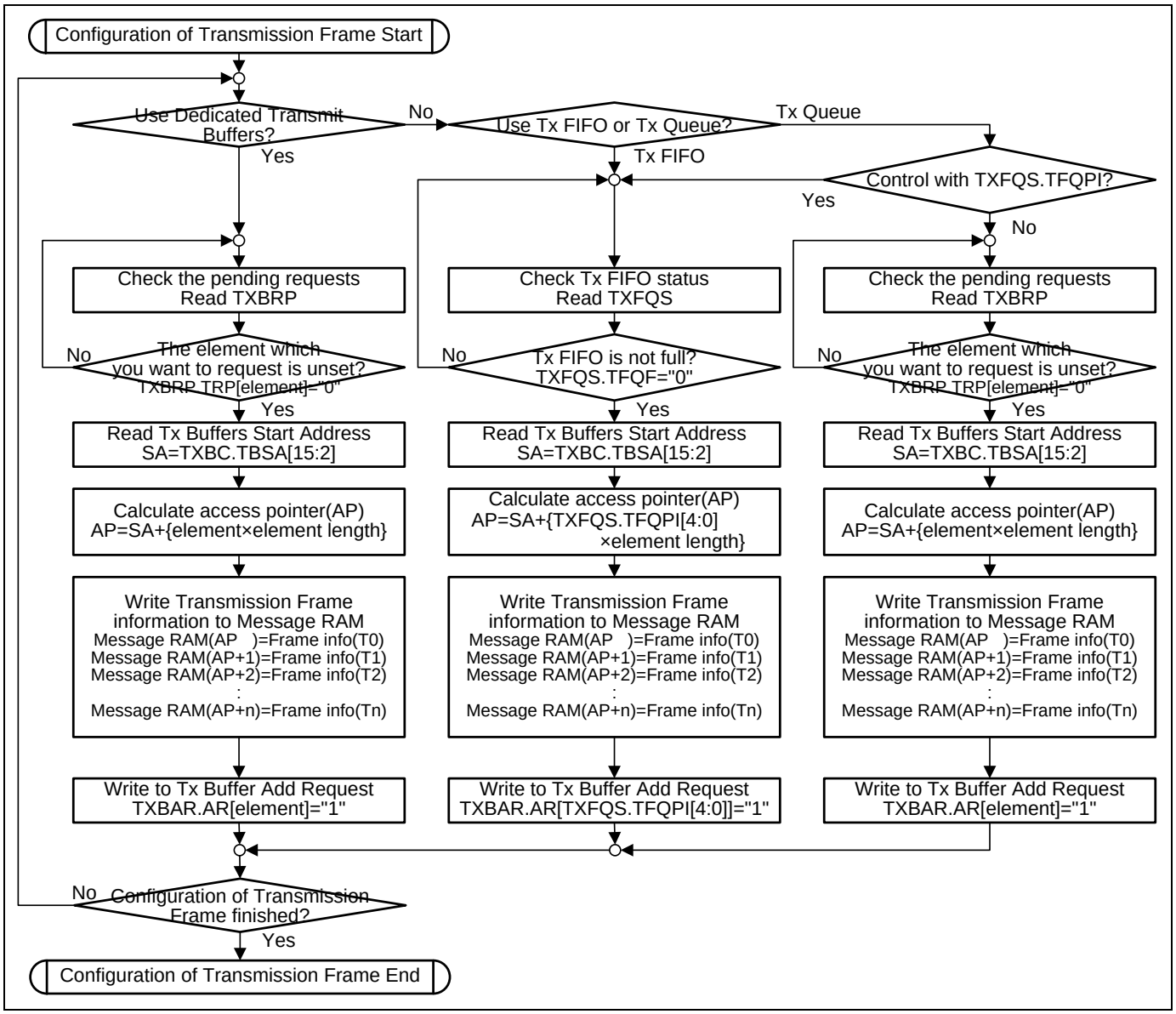
Note:

- *A mode change requested by writing to CCCR.CMR[1:0] is allowed only when no transmission requests are pending. If necessary, cancel pending transmission requests before changing the CAN operation mode.*

4.5.2 Configuration of Transmission Frame

Figure 4-12 shows how to set and transmit frames.

Figure 4-12 Flow diagram "Configuration of Transmission Frame"



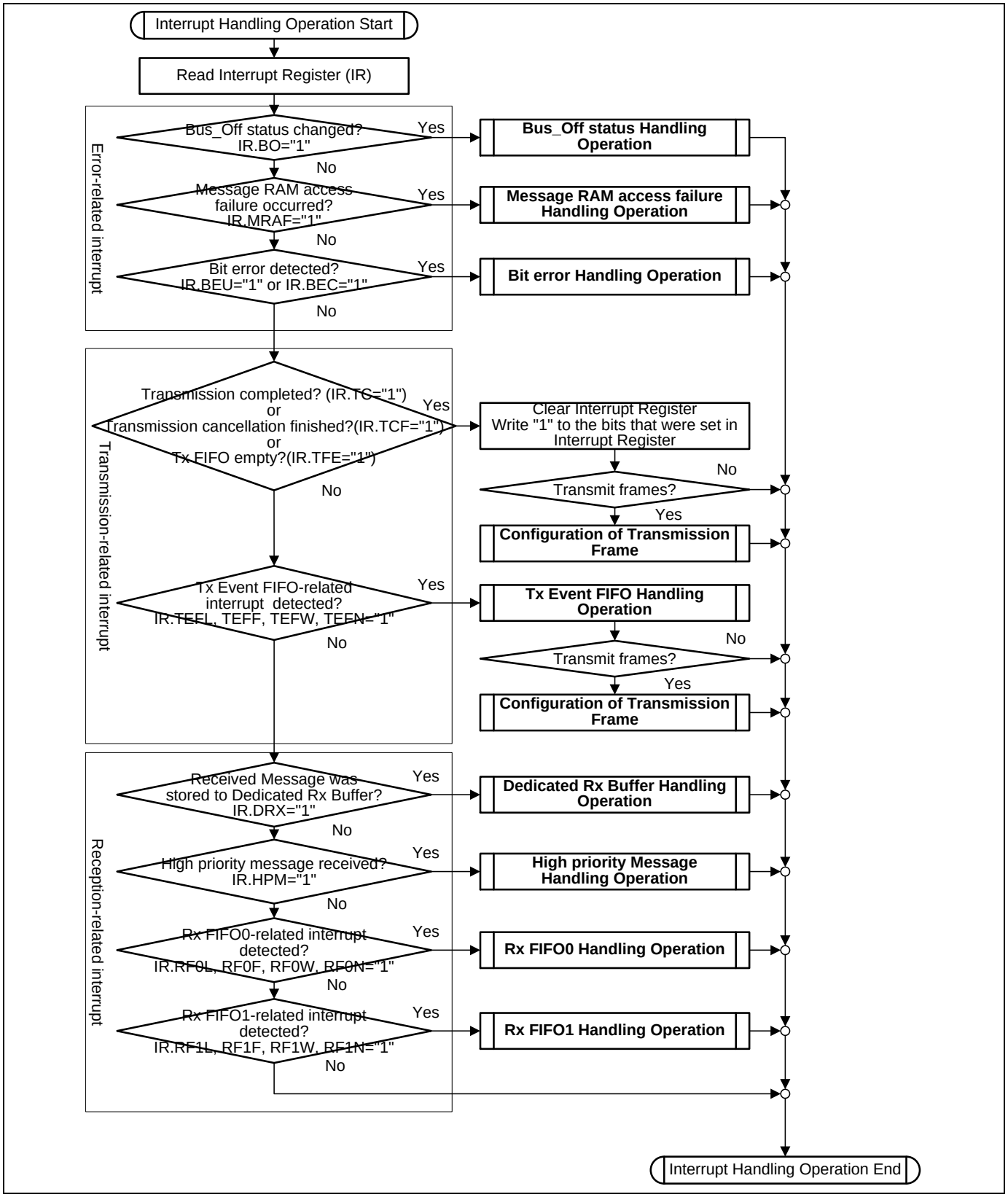
Notes:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*
- *Dedicated Transmit Buffers are normally configured with a single specific identifier. The above flow shows an example where the T0 word of a dedicated Transmit buffer is set only on the first write access to the corresponding Tx Buffer Element.*

4.6 Interrupt Handling Operation

Figure 4-13 shows the overall picture of how interrupts are processed.

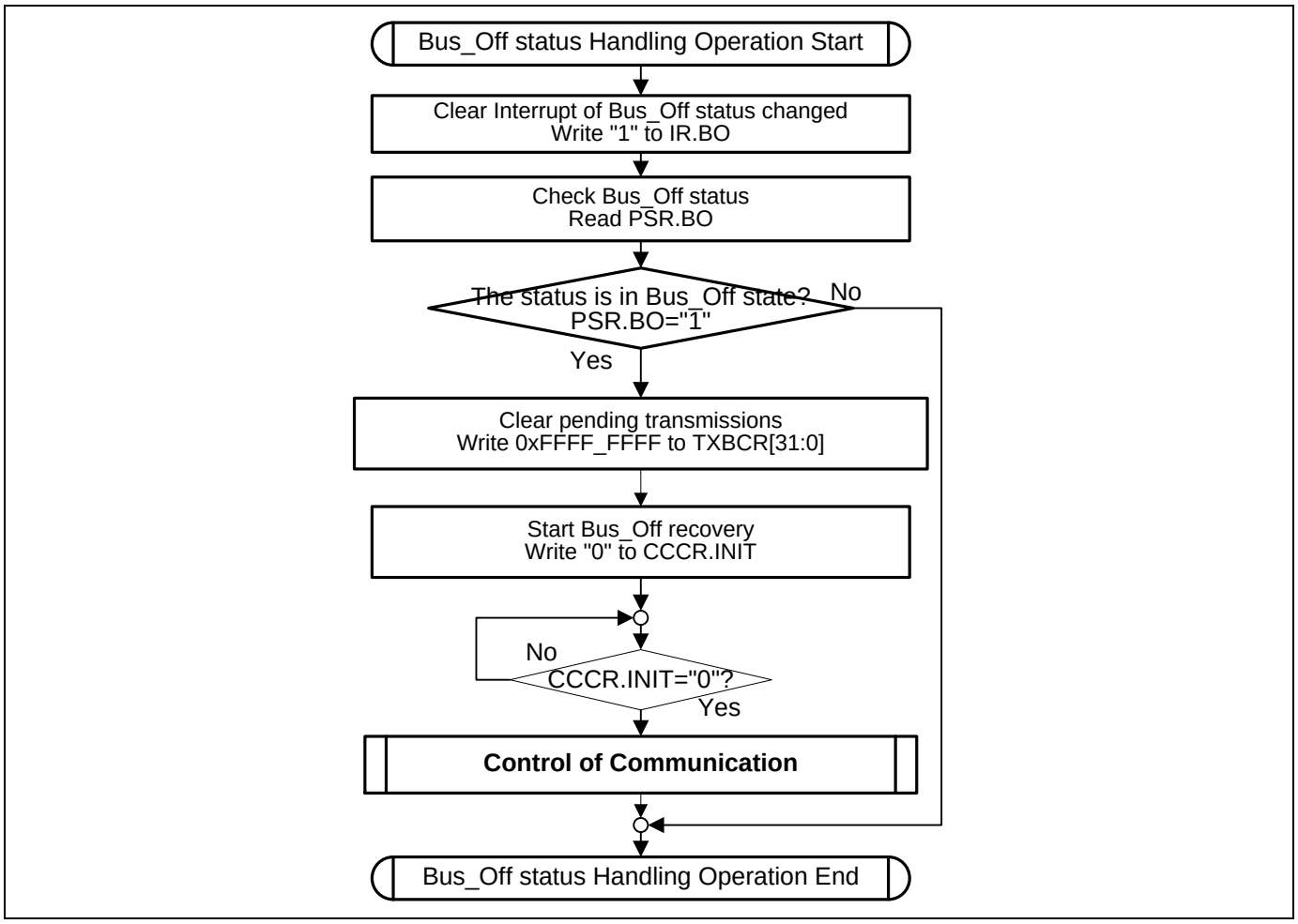
Figure 4-13 Flow diagram "Interrupt Handling Operation"



4.6.1 Bus_Off status Handling Operation

Figure 4-14 shows how to process a "Bus_Off Status" interrupt (IR.BO).

Figure 4-14 Flow diagram "Bus_Off status Handling Operation"



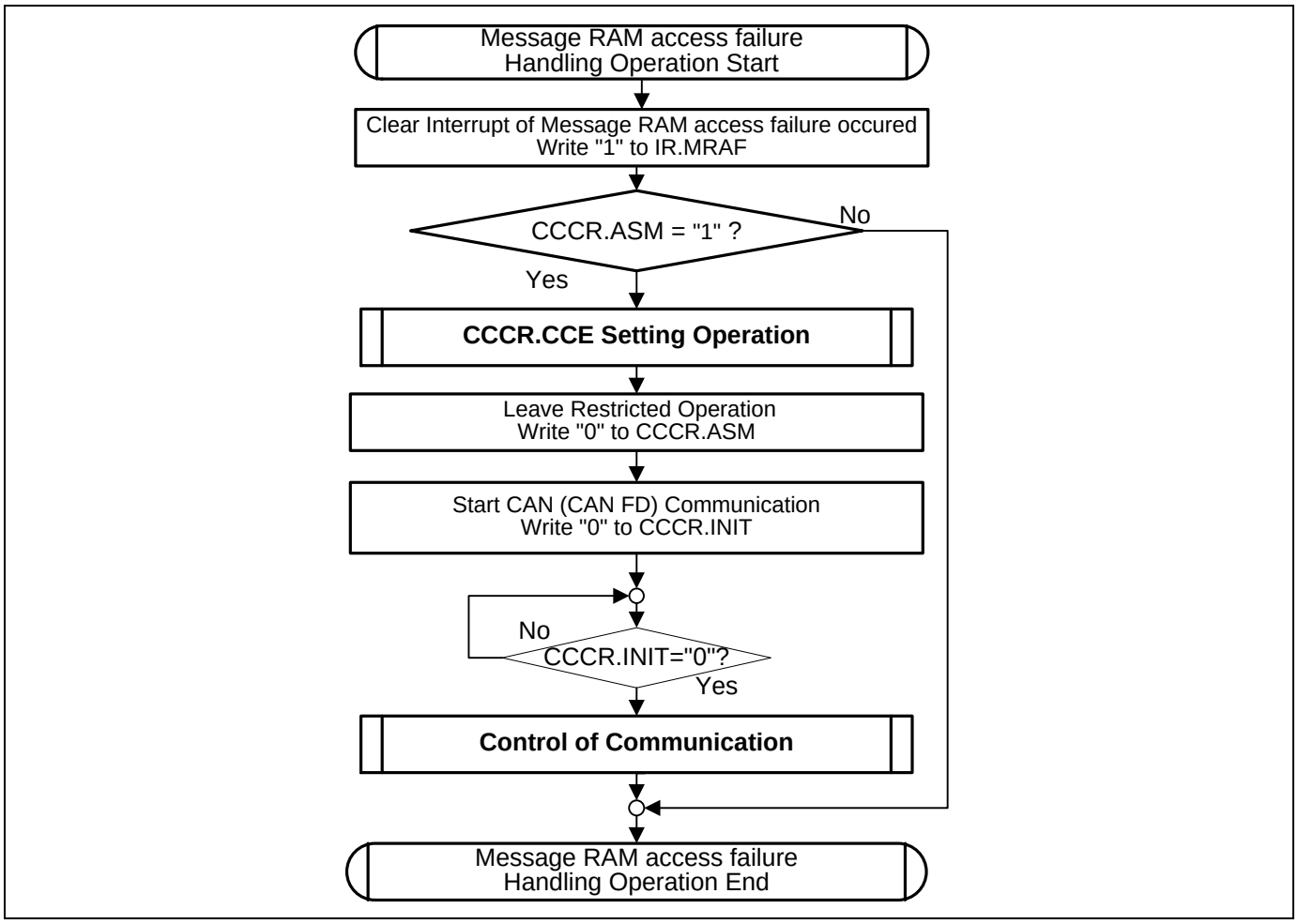
Note:

- *Once the Initialization bit (CCCR.INIT) has been cleared by the CPU, the CAN FD Controller will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation.*

4.6.2 Message RAM access failure Handling Operation

Figure 4-15 shows how to process a "Message RAM Access Failure" interrupt (IR.MRAF).

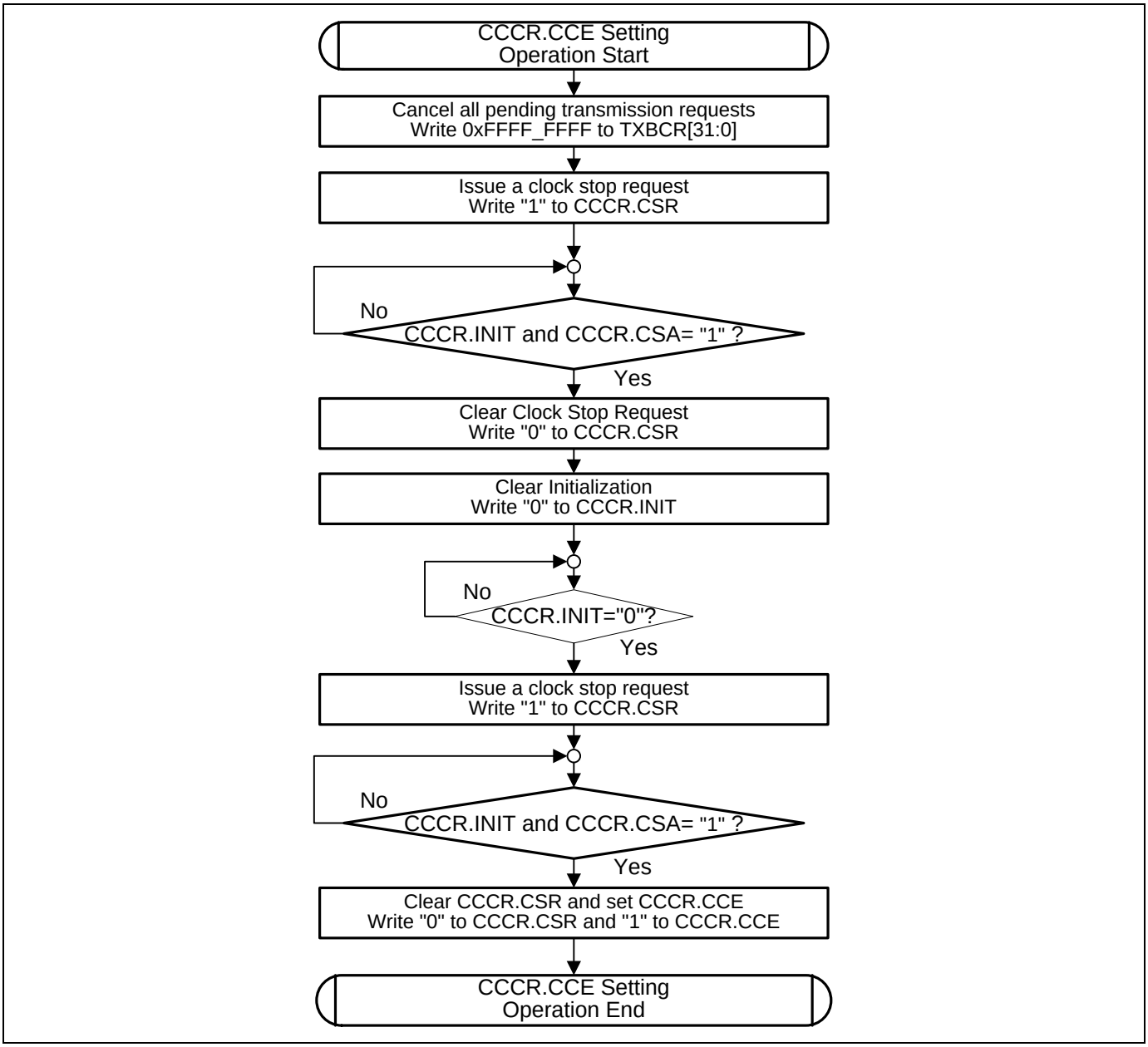
Figure 4-15 Flow diagram "Message RAM access failure Handling Operation"



4.6.2.1 CCCR.CCE Setting Operation

Figure 4-16 shows how to process a "Setting CCCR.CCE".

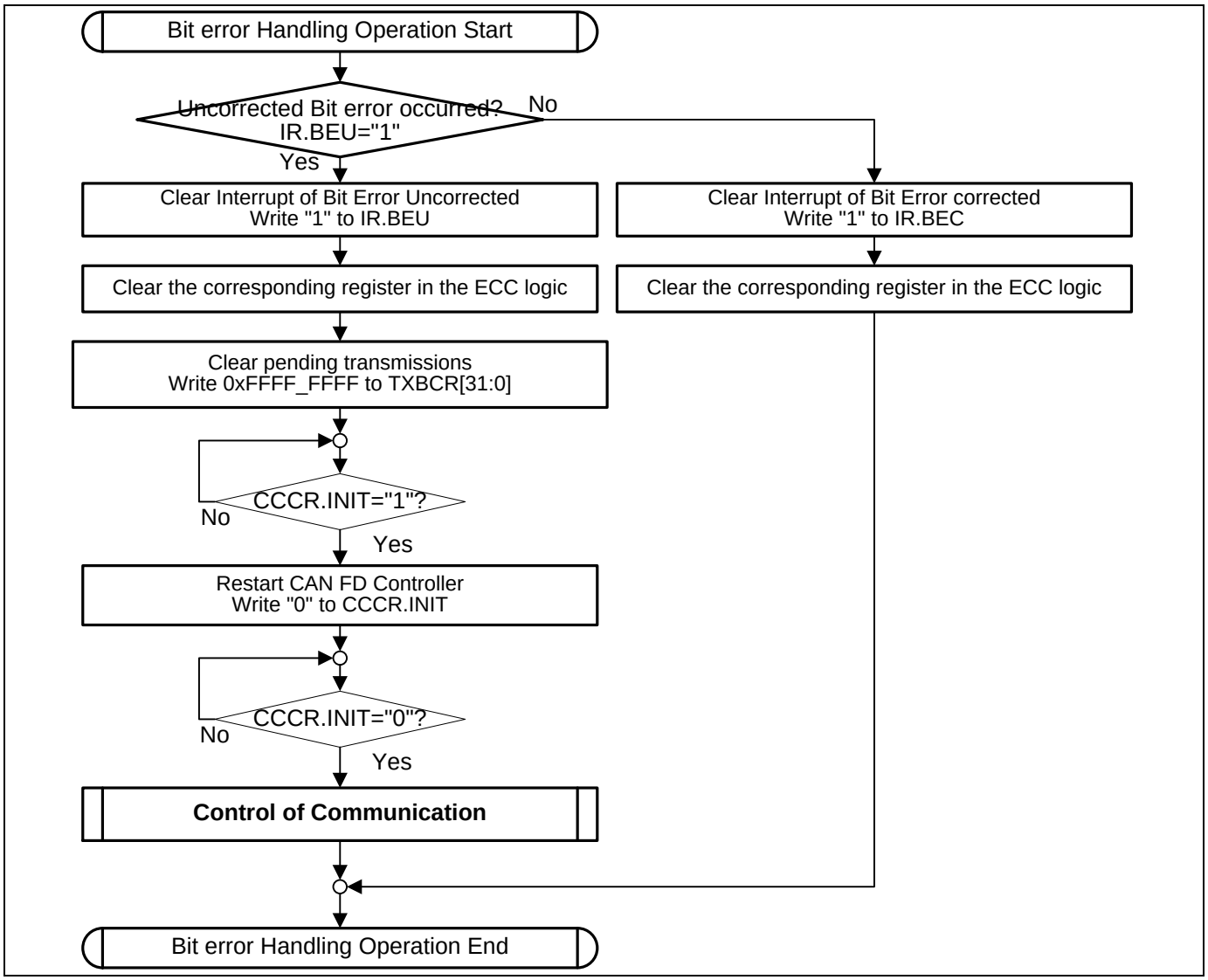
Figure 4-16 Flow diagram "CCCR.CCE Setting Operation"



4.6.3 Bit error Handling Operation

Figure 4-17 shows how to process the "Bit Error" interrupts (IR.BEU, IR.BEC).

Figure 4-17 Flow diagram "Bit error Handling Operation"



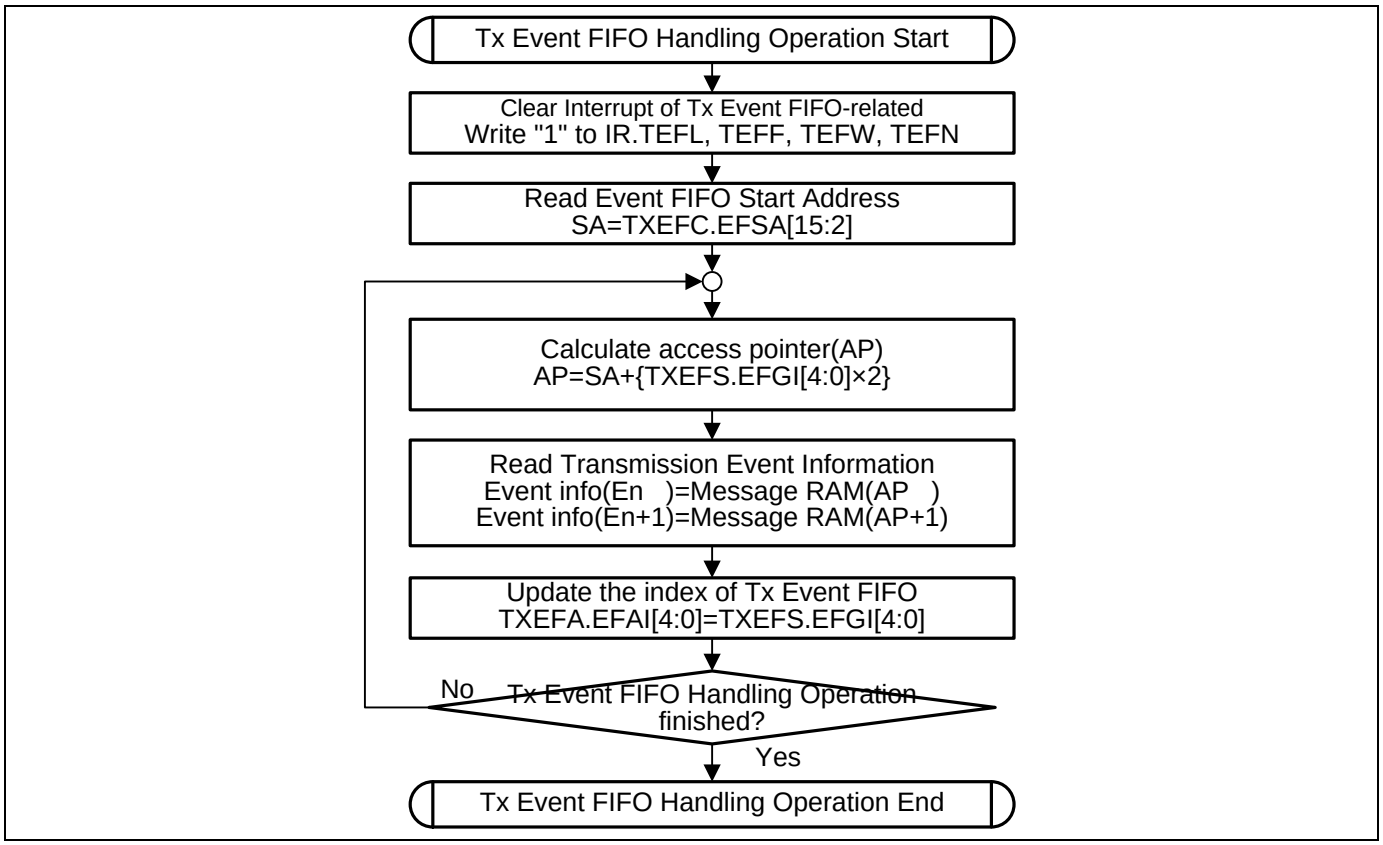
Note:

- *In case Bit Error Uncorrected IR.BEU occurs, due to the synchronization mechanism between the Bus clock and the CAN clock, there may be a delay until Initialization CCCR.INIT set to "1". Therefore the programmer has to assure that CCCR.INIT has been set to "1" by reading CCCR.INIT before resetting CCCR.INIT to "0".*

4.6.4 Tx Event FIFO Handling Operation

Figure 4-18 shows how to process Tx Event FIFO related interrupts.

Figure 4-18 Flow diagram "Tx Event FIFO Handling Operation"



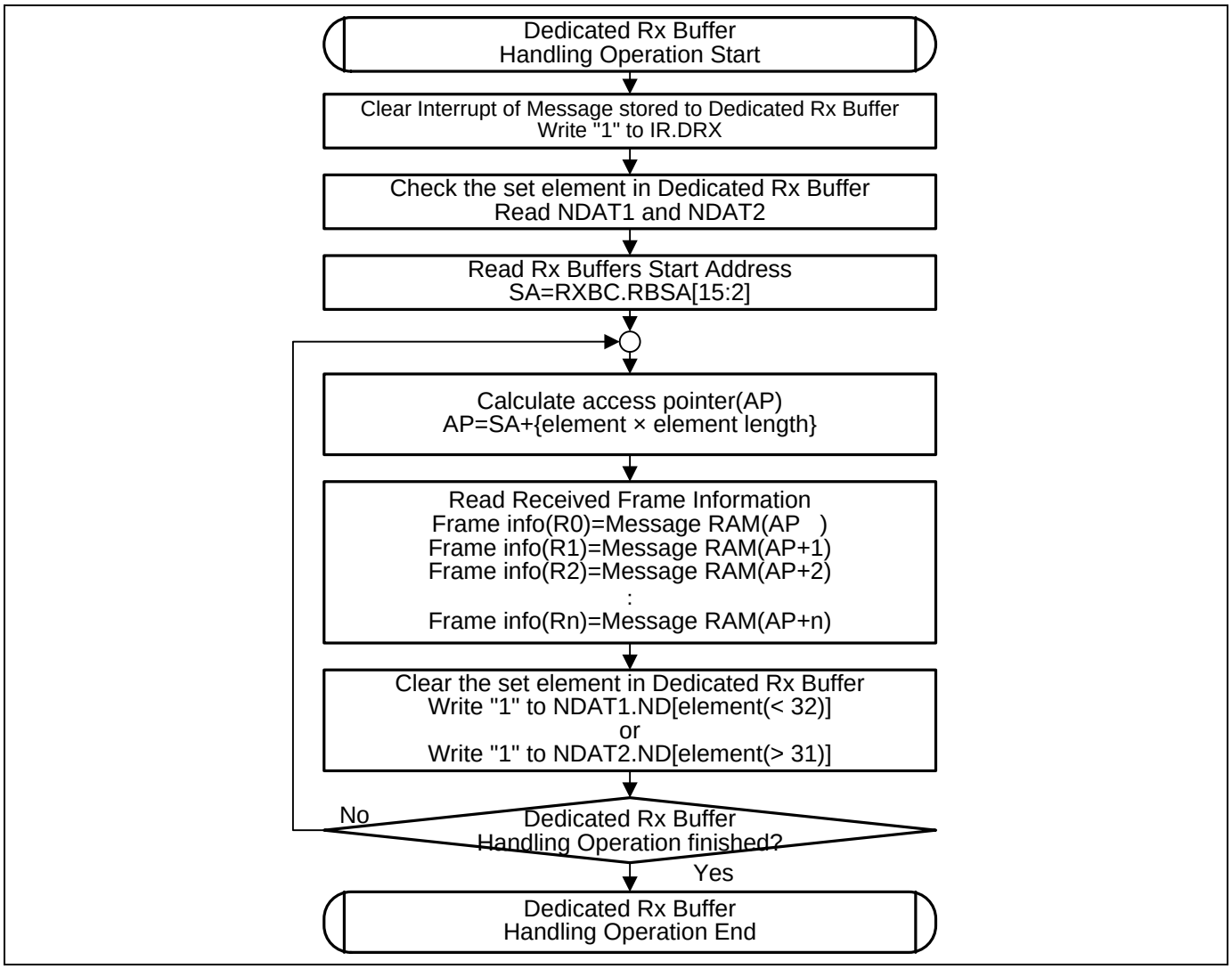
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.5 Dedicated Rx Buffer Handling Operation

Figure 4-19 shows how to process a "Message stored to Dedicated Rx Buffer" interrupt (IR.DRX).

Figure 4-19 Flow diagram "Dedicated Rx Buffer Handling Operation"



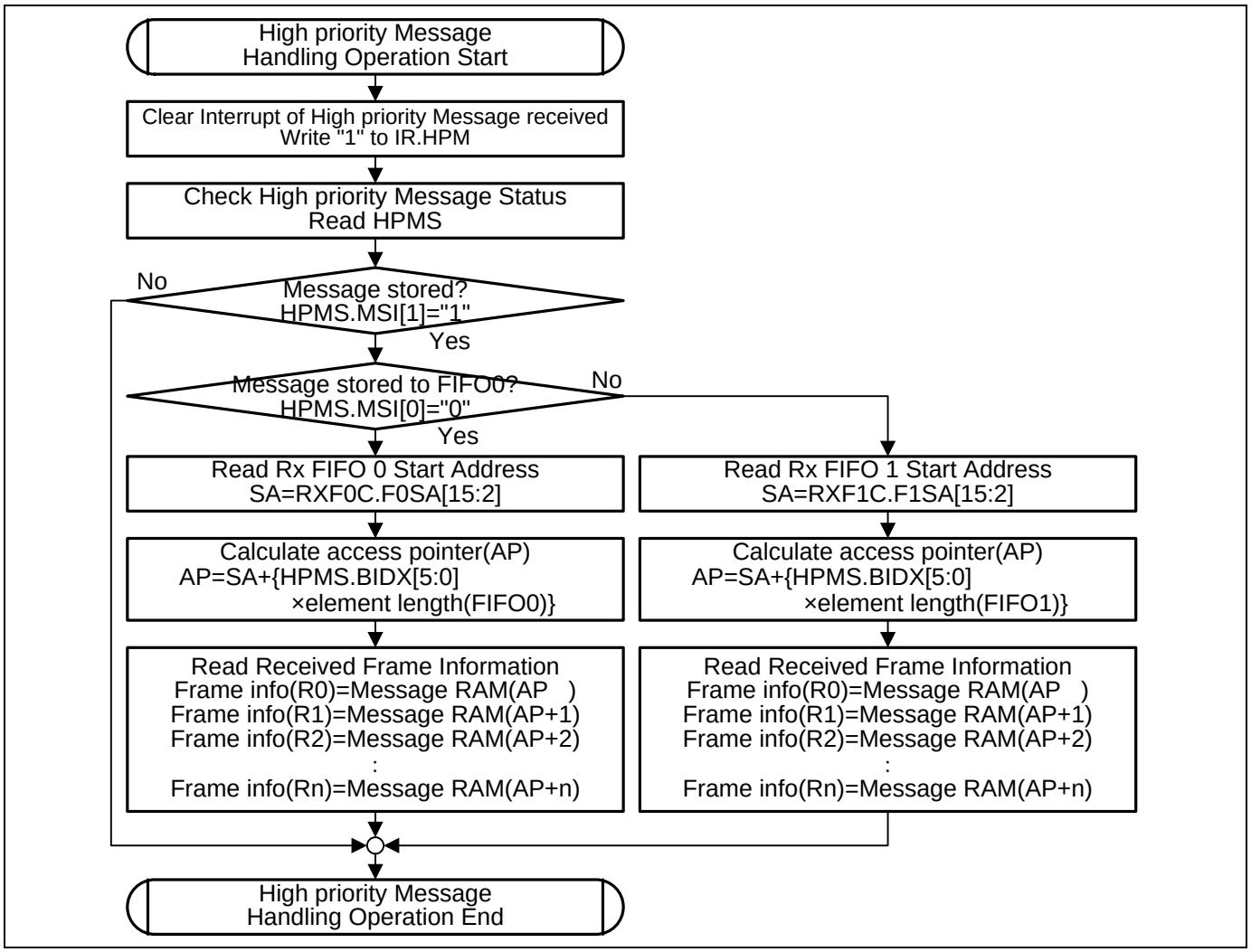
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.6 High priority Message Handling Operation

Figure 4-20 shows how to process a High Priority Message interrupt (IR.HPM).

Figure 4-20 Flow diagram "High priority Message Handling Operation"



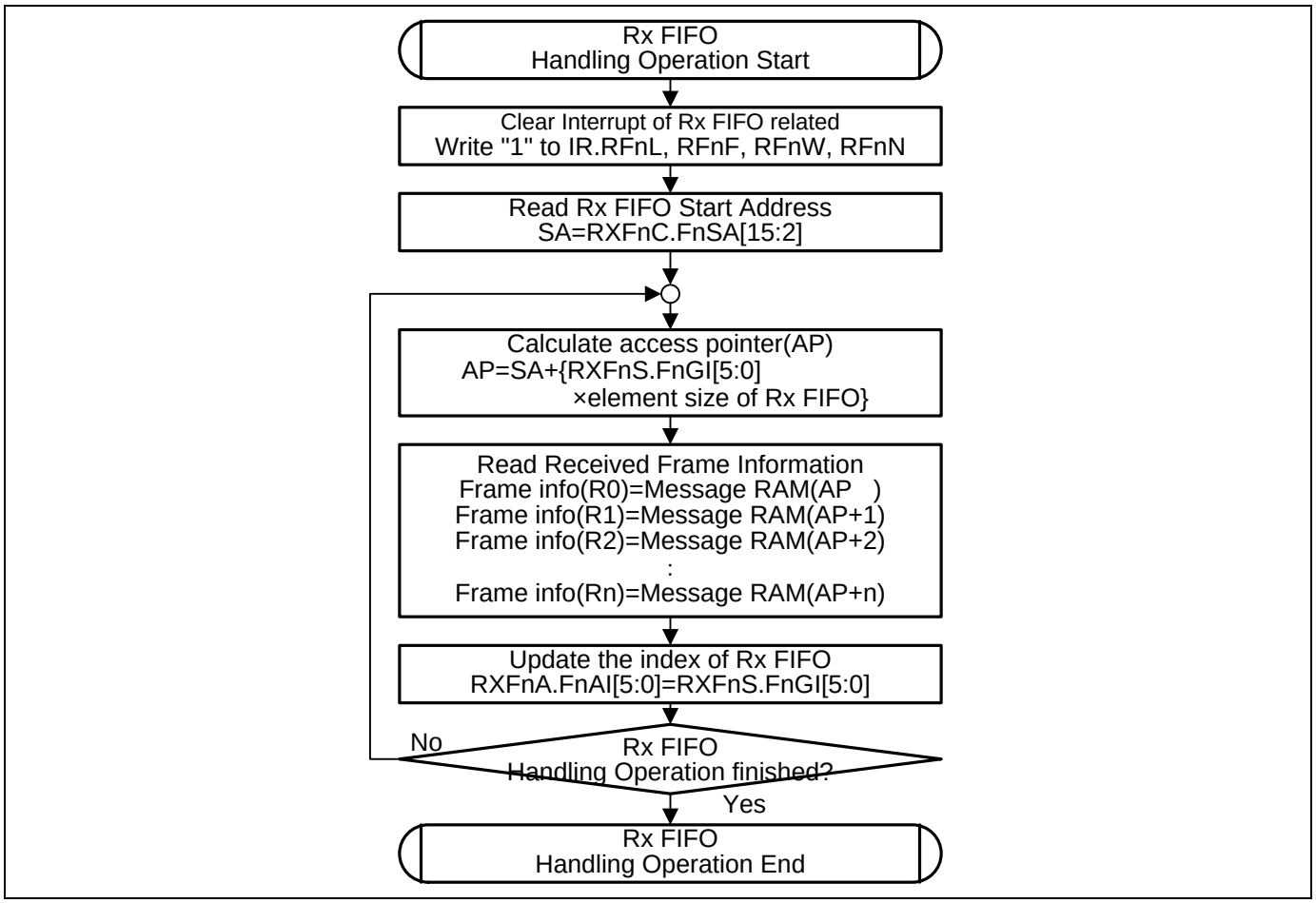
Note:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*

4.6.7 Rx FIFO Handling Operation

Figure 4-21 shows how to process Rx FIFO related interrupts.

Figure 4-21 Flow diagram "Rx FIFO Handling Operation"



Notes:

- *Use 32-bit word address values when calculating the pointers in the flow diagram above.*
- *The "n"s in the register/bit names (e.g. "RXFnS.FnGI") stand for "0" or "1".*
- *If a high priority message is received and processed by the "High priority Message Handling Operation" flow, this flow (i.e. Rx FIFO Handling Operation) may be skipped for that particular message. The following message received by the Rx FIFO may resume with this flow again.*

5. Registers

This section explains the configuration and functions of the registers used for the CAN FD Controller.

- Hardware Reset Description

After hardware reset, the registers of the CAN FD Controller hold the reset values as indicated in each relative register description. Additionally the Bus_Off state is reset and the output canfd_tx is set to recessive. Setting the Initialization bit CCCR.INIT to "1" in the CC Control Register enables software initialization. The CAN FD Controller does not influence the CAN bus until the CPU resets CCCR.INIT to "0".

- Register Map

The CAN FD Controller allocates an address space of 256 bytes for the registers.

All registers are organized as 32-bit registers. The registers are accessible by the CPU using a data width of 8 bit (byte access), 16 bit (half-word access), or 32 bit (word access).

Note:

- *As an exception, the Protocol Status Register (PSR) does not allow byte accessing.*

Write access by the CPU to several of the registers/bits that configure the basic operation of the CAN FD Controller is possible only with Configuration Change Enable CCCR.CCE="1" AND Initialization CCCR.INIT="1".

There is a delay from writing to a command register until the update of the related status register bits due to clock domain crossing.

Table 5-1 List of registers for the CAN FD Controller

Abbreviation	Register name	Reference
CREL	Core Release Register	5.1
ENDN	Endian Register	5.2
FBTP	Fast Bit Timing & Prescaler Register	5.3
TEST	Test Register	5.4
RWD	RAM Watchdog	5.5
CCCR	CC Control Register	5.6
BTP	Bit Timing & Prescaler Register	5.7
TSCC	Timestamp Counter Configuration	5.8
TSCV	Timestamp Counter Value	5.9
TOCC	Timeout Counter Configuration	5.10
TOCV	Timeout Counter Value	5.11
ECR	Error Counter Register	5.12
PSR	Protocol Status Register	5.13
IR	Interrupt Register	5.14
IE	Interrupt Enable	5.15
ILS	Interrupt Line Select	5.16
ILE	Interrupt Line Enable	5.17
GFC	Global Filter Configuration	5.18
SIDFC	Standard ID Filter Configuration	5.19
XIDFC	Extended ID Filter Configuration	5.20
XIDAM	Extended ID AND Mask	5.21
HPMS	High Priority Message Status	5.22
NDAT1	New Data 1	5.23
NDAT2	New Data 2	5.24
RXF0C	Rx FIFO 0 Configuration	5.25
RXF0S	Rx FIFO 0 Status	5.26
RXF0A	Rx FIFO 0 Acknowledge	5.27
RXBC	Rx Buffer Configuration	5.28
RXF1C	Rx FIFO 1 Configuration	5.29
RXF1S	Rx FIFO 1 Status	5.30
RXF1A	Rx FIFO 1 Acknowledge	5.31
RXESC	Rx Buffer/FIFO Element Size Configuration	5.32
TXBC	Tx Buffer Configuration	0
TXFQS	Tx FIFO/Queue Status	5.34
TXESC	Tx Buffer Element Size Configuration	5.35
TXBRP	Tx Buffer Request Pending	5.36
TXBAR	Tx Buffer Add Request	5.37
TXBCR	Tx Buffer Cancellation Request	5.38
TXBTO	Tx Buffer Transmission Occurred	5.39
TXBCF	Tx Buffer Cancellation Finished	5.40
TXBTIE	Tx Buffer Transmission Interrupt Enable	5.41
TXBCIE	Tx Buffer Cancellation Finished Interrupt Enable	5.42
TXEFC	Tx Event FIFO Configuration	5.43
TXEFS	Tx Event FIFO Status	5.44
TXEFA	Tx Event FIFO Acknowledge	5.45

5.1 Core Release Register (CREL)

The Core Release Register displays the revision of the CAN FD Controller.

bit	31	30	29	28	27	26	25	24
Field	REL[3:0]				STEP[3:0]			
Attribute	R				R			
Initial value	0x3				0x0			

bit	23	22	21	20	19	18	17	16
Field	SUBSTEP[3:0]				YEAR[3:0]			
Attribute	R				R			
Initial value	0x1				0x3			

bit	15	14	13	12	11	10	9	8
Field	MON[7:0]							
Attribute	R							
Initial value	0x05							

bit	7	6	5	4	3	2	1	0
Field	DAY[7:0]							
Attribute	R							
Initial value	0x06							

[bit31:28] REL[3:0]: Core Release
One digit, BCD-coded.

[bit27:24] STEP[3:0]: Step of Core Release
One digit, BCD-coded.

[bit23:20] SUBSTEP[3:0]: Sub-step of Core Release
One digit, BCD-coded.

[bit19:16] YEAR[3:0]: Time Stamp Year
One digit, BCD-coded.

[bit15:8] MON[7:0]: Time Stamp Month
Two digits, BCD-coded.

[bit7:0] DAY[7:0]: Time Stamp Day
Two digits, BCD-coded.

Table 5-2 Example for Coding of Revisions

Release	Step	SubStep	Year	Month	Day	Name
3	0	1	3	05	06	Revision 3.0.1, Date 2013/05/06

5.2 Endian Register (ENDN)

The Endian Register can be used to check the endianness of the CAN FD Controller when accessed by the CPU.

bit	31	30	29	28	27	26	25	24
Field	ETV[31:24]							
Attribute	R							
Initial value	0x87							

bit	23	22	21	20	19	18	17	16
Field	ETV[23:16]							
Attribute	R							
Initial value	0x65							

bit	15	14	13	12	11	10	9	8
Field	ETV[15:8]							
Attribute	R							
Initial value	0x43							

bit	7	6	5	4	3	2	1	0
Field	ETV[7:0]							
Attribute	R							
Initial value	0x21							

[bit31:0] ETV[31:0]: Endianness Test Value

The endianness test value is 0x87654321.

5.3 Fast Bit Timing & Prescaler Register (FBTP)

The Fast Bit Timing & Prescaler Register configures the fast bit time and the offset value for Transceiver Delay Compensation.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.1.4.Transceiver Delay Compensation" for a description of the TDCO[4:0] and TDC fields.

See "3.7.Configuring the CAN Bit Timing" for a description of all other fields.

Notes:

- *With a CAN clock (canfd_cclk) of 8 MHz, the reset value of 0x00000A33 configures the CAN FD Controller for a FD bit rate of 500 kBit/s.*
- *The bit rate configured for the CAN FD data phase via the Fast Bit Timing & Prescaler Register (FBTP) must be higher or equal to the bit rate configured for the arbitration phase via the Bit Timing & Prescaler Register (BTP).*

bit	31	30	29	28	27	26	25	24
Field	Reserved			TDCO[4:0]				
Attribute	-			R/W				
Initial value	000			00000				

bit	23	22	21	20	19	18	17	16
Field	TDC	Reserved		FBRP[4:0]				
Attribute	R/W	-		R/W				
Initial value	0	00		00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved				FTSEG1[3:0]			
Attribute	-				R/W			
Initial value	0000				0xA			

bit	7	6	5	4	3	2	1	0
Field	Reserved	FTSEG2[2:0]			Reserved		FSJW[1:0]	
Attribute	-	R/W			-		R/W	
Initial value	0	011			00		11	

[bit31:29] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit28:24] TDCO[4:0] : Transceiver Delay Compensation Offset

TDCO[4:0]	Description
0x00-0x1F	Offset value defining the distance between the measured delay (the delay from canfd_tx to canfd_rx) and the secondary sample point. Valid values are 0 to 31 canfd_cclk periods.

[bit23] TDC: Transceiver Delay Compensation

bit	Description
0	Transceiver Delay Compensation disabled.
1	Transceiver Delay Compensation enabled.

[bit22:21] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit20:16] FBRP[4:0]: Fast Baud Rate Prescaler

FBRP[4:0]	Description
0x00-0x1F	The value by which the oscillator frequency is divided for generating the fast bit time quanta. The fast bit time is built up from a multiple of this quanta. Valid values for the Fast Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit15:12] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit11:8] FTSEG1[3:0]: Fast time segment before sample point

FTSEG1[3:0]	Description
0x1-0xF	Valid values are 1 to 15. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit7] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit6:4] FTSEG2[2:0]: Fast time segment after sample point

FTSEG2[2:0]	Description
0x0-0x7	Valid values are 0 to 7. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit3:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1:0] FSJW[1:0]: Fast (Re) Synchronization Jump Width

FSJW[1:0]	Description
0x0-0x3	Valid values are 0 to 3. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

5.4 Test Register (TEST)

The Test Register monitors the canfd_rx/canfd_tx pins and displays the Transceiver Delay Compensation value. It is also used to enable the Loop Back Modes.

Write access to the Test Register has to be enabled by setting the Test Mode Enable bit CCCR.TEST to "1".

All Test Register functions are set to their reset values when bit CCCR.TEST is reset.

Loop Back Mode and software control of pin canfd_tx are hardware test modes. Programming of TX ≠ "00" may disturb the message transfer on the CAN bus.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved		TDCV[5:0]					
Attribute	-		R					
Initial value	00		000000					

bit	7	6	5	4	3	2	1	0
Field	RX	TX[1:0]		LBCK	Reserved			
Attribute	R	R/W		R/W	-			
Initial value	U	00		0	0000			

U = undefined. The RX bit will reflect the actual level at pin canfd_rx.

[bit31:14] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit13:8] TDCV[5:0]: Transceiver Delay Compensation Value

TDCV[5:0]	Description
0x00-0x3F	Position of the secondary sample point, defined by the sum of the measured delay from canfd_tx to canfd_rx and the Transceiver Delay Compensation Offset (FBTP.TDCO[4:0]). Valid values are 0 to 63 canfd_cclk periods.

[bit7] RX: Receive Pin

Monitors the actual value of pin canfd_rx

bit	Description
0	The CAN bus is dominant (dominant level at pin canfd_rx).
1	The CAN bus is recessive (recessive level at pin canfd_rx).

[bit6:5] TX[1:0]: Control of Transmit Pin

TX[1:0]	Description
00	Reset value. canfd_tx controlled by the CAN Core, updated at the end of the bit time.
01	Sample Point can be monitored at pin canfd_tx.
10	Dominant level at pin canfd_tx.
11	Recessive level at pin canfd_tx.

[bit4] LBCK: Loop Back Mode

bit	Description
0	Reset value, Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

[bit3:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.5 RAM Watchdog (RWD)

The RAM Watchdog monitors the Message RAM to see if it is ready to be accessed or not.

A Message RAM access starts the RAM Watchdog Counter with the value configured by the Watchdog Configuration (RWD.WDC[7:0]). The counter is reloaded with RWD.WDC when a successful access to the Message RAM has been completed. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag Watchdog Interrupt (IR.WDI) is set.

The RAM Watchdog Counter is clocked by the Bus clock (canfd_bclk).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	WDV[7:0]							
Attribute	R							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	WDC[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:8] WDV[7:0]: Watchdog Value

WDV[7:0]	Description
0x00-0xFF	Actual Message RAM Watchdog Counter Value

[bit7:0] WDC[7:0]: Watchdog Configuration

Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.

Write access to this field is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

5.6 CC Control Register (CCCR)

The CC Control Register configures various operating modes of the CAN FD Controller.

For details about setting and resetting of single bits see "3.1.1. Software Initialization".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved	TXP	FDBS	FDO	CMR[1:0]		CME[1:0]	
Attribute	-	R/W	R	R	R/W		R/W	
Initial value	0	0	0	0	00		00	

bit	7	6	5	4	3	2	1	0
Field	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
Attribute	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1

[bit31:15] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit14] TXP: Transmit Pause

If this bit is set, the CAN FD Controller pauses for two nominal bit times before starting the next transmission after it has successfully transmitted a frame.

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	Description
0	Transmit pause disabled.
1	Transmit pause enabled.

[bit13] FDBS: CAN FD Bit Rate Switching

This bit is configured via CCCR.CMR[1:0]. FDBS is reset to "0" when Initialization (CCCR.INIT) is set to "1".

bit	Description
0	This node transmits no frames with bit rate switching.
1	This node transmits all frames (excl. remote frames) with bit rate switching.

[bit12] FDO: CAN FD Operation

This bit is configured via CCCR.CMR[1:0]. FDO is reset to "0" when Initialization (CCCR.INIT) is set to "1".

bit	Description
0	This node transmits all frames in CAN format according to ISO11898-1.
1	This node transmits all frames (excl. remote frames) in CAN FD format.

[bit11:10] CMR[1:0]: CAN Mode Request

A change of the CAN operation mode is requested by writing to this bit field. After change to the requested operation mode the bit field is reset to "00" and the status flags CAN FD Bit Rate Switching (CCCR.FDBS) and CAN FD Operation (CCCR.FDO) are set accordingly.

In case the requested CAN operation mode is not enabled, the value written to CMR[1:0] is retained until it is overwritten by the next mode change request.

In case CAN Mode Enable CCCR.CME[1:0] = "01"/"10"/"11" a change to CAN operation according to ISO 11898-1 is always possible.

Default is CAN operation according to ISO11898-1 (CCCR.FDBS = "0" and CCCR.FDO = "0").

CMR[1:0]	Description
00	Unchanged.
01	Request CAN FD operation.
10	Request CAN FD operation with bit rate switching.
11	Request CAN operation according ISO11898-1.

[bit9:8] CME[1:0]: CAN Mode Enable

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

CME[1:0]	Description
00	CAN operation according to ISO11898-1 enabled.
01	CAN FD operation enabled.
10	CAN FD operation with bit rate switching enabled.
11	CAN FD operation with bit rate switching enabled.

Note:

When CME[1:0] = "00", received frames are strictly interpreted according to ISO11898-1, which leads to the transmission of an error frame when receiving a CAN FD frame.
In case CME[1:0] = "01", transmission of long CAN FD frames and reception of long and fast CAN FD frames is enabled.
With CME[1:0] = "10"/"11", transmission and reception of long and fast CAN FD frames is enabled.

[bit7] TEST: Test Mode Enable

Bit TEST can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

The bit can be reset by the CPU at any time.

bit	Description
0	Normal operation, register TEST holds reset values.
1	Test Mode, write access to register TEST enabled.

[bit6] DAR: Disable Automatic Retransmission

Write access to this bit is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1". For a description of DAR mode, see "3.1.7. Disabled Automatic Retransmission".

bit	Description
0	Automatic retransmission of messages not transmitted successfully enabled.
1	Automatic retransmission disabled.

[bit5] MON: Bus Monitoring Mode

Bit MON can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

The bit can be reset by the CPU at any time.

bit	Description
0	Bus Monitoring Mode is disabled.
1	Bus Monitoring Mode is enabled.

[bit4] CSR: Clock Stop Request

For a description of Power Down, see "3.1.8. Power Down (Sleep Mode)".

bit	Description
0	No clock stop is requested.
1	Clock stop requested. When clock stop is requested, first CCCR.INIT and then CCCR.CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

[bit3] CSA: Clock Stop Acknowledge

For a description of Power Down, see "3.1.8. Power Down (Sleep Mode)".

bit	Description
0	No clock stop acknowledged.
1	CAN FD Controller may be set in power down by stopping canfd_bclk and canfd_cclk.

[bit2] ASM: Restricted Operation Mode

Bit ASM can only be set by the CPU when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1"

The bit can be reset by the CPU at any time, but must follow the procedure as described in "4.6.2. Message RAM access failure Handling Operation".

For a description of the Restricted Operation Mode see "3.1.5. Restricted Operation Mode".

bit	Description
0	Normal CAN operation.
1	Restricted Operation Mode active.

[bit1] CCE: Configuration Change Enable

Write access to this bit is possible only when the Initialization (CCCR.INIT) bit is "1". This bit (CCCR.CCE) is reset to "0" when CCCR.INIT is set to "0".

bit	Description
0	The CPU has no write access to the protected configuration registers.
1	The CPU has write access to the protected configuration registers (while bit Initialization CCCR.INIT = "1").

Note:

- *The CPU must follow the procedure as described in "3.1.1. Software Initialization" to set CCCR.CCE, but it isn't necessary to follow the procedure, in case of setting CCCR.CCE just after hardware reset.*

[bit0] INIT: Initialization

bit	Description
0	Normal Operation.
1	Initialization is started.

Notes:

- *Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to Initialization CCCR.INIT can be read back (the maximum length of the delay is four Bus clocks plus five CAN clocks). Therefore the programmer has to assure that the previous value written to CCCR.INIT has been accepted by reading CCCR.INIT before setting CCCR.INIT to a new value.*
- *In case of setting CCCR.INIT to "1" during Normal Operation, set CCCR.INIT to "1" by issuing a Clock Stop Request via CCCR.CSR = "1". Before resetting CCCR.INIT first reset CCCR.CSR.*

5.7 Bit Timing & Prescaler Register (BTP)

The Bit Timing & Prescaler Register configures the nominal bit time of the CAN FD Controller.

This register is only writable if bits Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) are set.

See "3.7.Configuring the CAN Bit Timing" for a description of the Bit Timing.

Notes:

- *With a CAN clock (canfd_cclk) of 8 MHz, the reset value of 0x00000A33 configures the CAN FD Controller for a bit rate of 500 kBit/s.*
- *In environments that include CAN FD enabled nodes, do not use bit timing configurations where Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000). Such settings will cause the r1 bit (for CAN format frames) or EDL bit (for CAN FD format frames) to be transmitted opposite the protocol defined level.*
  In a CAN only environment, a bit timing configuration where Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000) may be used although this configuration will cause the r1 bit to be transmitted opposite the protocol defined level.

Details
The r1 bit (for CAN format frames) or EDL bit (for CAN FD format frames) will be transmitted opposite the protocol defined level for the following cases (all conditions within each case must be fulfilled).

Case 1:

- *The CAN FD Controller is configured to CAN operation mode according to ISO 11898-1 (CAN FD Bit Rate Switching (CCCR.FDBS) and CAN FD Operation (CCCR.FDO) are both zero)*
- *Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000)*
- *An extended frame format message with the MSB of the Identifier (ID28) set to "1" is transmitted (XTD bit = "1" and ID[28] = "1" in the corresponding Tx Buffer Element)*

Case 2:

- *The CAN FD Controller is configured to CAN FD operation mode (CAN FD Operation (CCCR.FDO) is "1")*
- *Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000)*
- *A Standard Frame format message is transmitted (XTD bit = "0" in the corresponding Tx Buffer Element)*

Case 3:

- *The CAN FD Controller is configured to CAN FD operation mode (CAN FD Operation (CCCR.FDO) is "1")*
- *Time Segment after sample point and the Baud Rate Prescaler are both zero (i.e. BTP.TSEG2[3:0] = 0x0 and BTP.BRP[9:0] = 0x000)*
- *An extended frame format message with the MSB of the Identifier (ID28) set to "0" is transmitted (XTD bit = "1" and ID[28] = "0" in the corresponding Tx Buffer Element)*

bit	31	30	29	28	27	26	25	24
Field	Reserved						BRP[9:8]	
Attribute	-						R/W	
Initial value	000000						00	

bit	23	22	21	20	19	18	17	16
Field	BRP[7:0]							
Attribute	R/W							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved		TSEG1[5:0]					
Attribute	-		R/W					
Initial value	00		001010					

bit	7	6	5	4	3	2	1	0
Field	TSEG2[3:0]				SJW[3:0]			
Attribute	R/W				R/W			
Initial value	0x3				0x3			

[bit31:26] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit25:16] BRP[9:0]: Baud Rate Prescaler

BRP[9:0]	Description
0x000-0x3FF	The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 1023. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

[bit15:14] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit13:8] TSEG1[5:0]: Time segment before sample point

TSEG1[5:0]	Description
0x01-0x3F	Valid values are 1 to 63. The value 0 must not be used. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit7:4] TSEG2[3:0]: Time segment after sample point

TSEG2[3:0]	Description
0x0-0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

[bit3:0] SJW[3:0]: (Re) Synchronization Jump Width

SJW[3:0]	Description
0x0-0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

5.8 Timestamp Counter Configuration (TSCC)

The Timestamp Counter Configuration holds the settings for Timestamp Generation.

Write access to the Timestamp Counter Configuration (TSCC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

For a description of the Timestamp Counter see "3.2. Timestamp Generation".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved				TCP[3:0]			
Attribute	-				R/W			
Initial value	0x0				0x0			

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved						TSS[1:0]	
Attribute	-						R/W	
Initial value	000000						00	

[bit31:20] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit19:16] TCP[3:0]: Timestamp Counter Prescaler

TCP[3:0]	Description
0x0-0xF	Configures the timestamp and timeout counters time unit in multiples of bit times [1 to 16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note:

- *For each timestamp and timeout counter time unit tick, the CAN FD Controller counts the number of bit times that have elapsed with an internal counter, up to the configured TSCC.TCP[3:0] value. This internal counter is not initialized by Initialization (CCCR.INIT), but only by a hardware reset.*

[bit15:2] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit1:0] TSS[1:0]: Timestamp Select

TSS[1:0]	Description
00	Timestamp counter (TSCV.TSC[15:0]) value always 0x0000.
01	Timestamp counter value incremented according to Timestamp Counter Prescaler (TSCC.TCP[3:0]).
10	Timestamp counter value of a counter external to the CAN FD Controller used.
11	Same as "00".

Note:

- *When Timestamp Select TSCC.TSS[1:0] = "01", the clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore, if the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.*

5.9 Timestamp Counter Value (TSCV)

Holds the value of the Timestamp Counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	TSC[15:8]							
Attribute	R/W							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	TSC[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:0] TSC[15:0]: Timestamp Counter

The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx).

- When Timestamp Select TSCC.TSS[1:0] = "01", the Timestamp Counter TSCV.TSC[15:0] is incremented in multiples of bit times [1 to 16] depending on the configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]). A wrap around sets interrupt flag Timestamp Wraparound (IR.TSW). Write access to TSCV resets the counter to zero.
- When Timestamp Select TSCC.TSS[1:0] = "10", TSCV.TSC[15:0] reflects a 16-bit Timestamp Counter value obtained from a counter external to the CAN FD Controller. For this case, a write access has no impact and will not clear the counter value.

Note:

A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by write access to Timestamp Counter Value (TSCV). Such write access will not set IR.TSW to "1".

5.10 Timeout Counter Configuration (TOCC)

The Timeout Counter Configuration holds the settings for the Timeout Counter.

Write access to the Timeout Counter Configuration (TOCC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

For a description of the Timeout Counter see "3.3. Timeout Counter".

bit	31	30	29	28	27	26	25	24
Field	TOP[15:8]							
Attribute	R/W							
Initial value	0xff							

bit	23	22	21	20	19	18	17	16
Field	TOP[7:0]							
Attribute	R/W							
Initial value	0xff							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved					TOS[1:0]		ETOC
Attribute	-					R/W		R/W
Initial value	00000					00		0

[bit31:16] TOP[15:0]: Timeout Period

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

[bit15:3] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit2:1] TOS[1:0]: Timeout Select

When operating in Continuous mode (TOS = "00"), a write to Timeout Counter Value (TOCV) presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]) and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs (TOS = "01"/"10"/"11"), an empty FIFO presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]). Down-counting is started when the first FIFO element is stored.

TOS[1:0]	Description
00	Continuous operation.
01	Timeout controlled by Tx Event FIFO.
10	Timeout controlled by Rx FIFO 0.
11	Timeout controlled by Rx FIFO 1.

[bit0] ETOC: Enable Timeout Counter

bit	Description
0	Timeout Counter disabled.
1	Timeout Counter enabled.

5.11 Timeout Counter Value (TOCV)

Holds the value of the Timeout Counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	TOC[15:8]							
Attribute	R/W							
Initial value	0xff							

bit	7	6	5	4	3	2	1	0
Field	TOC[7:0]							
Attribute	R/W							
Initial value	0xff							

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:0] TOC[15:0]: Timeout Counter

The Timeout Counter is decremented in multiples of bit times [1 to 16] depending on the configuration of Timestamp Counter Prescaler (TSCC.TCP[3:0]). Setting Configuration Change Enable (CCCR.CCE) will preset the Timeout Counter (TOCV.TOC[15:0]) to the Timeout Period value (TOCC.TOP[15:0]).

- When Timeout Select TOCC.TOS[1:0] = "00" (Continuous Mode), the counter starts when Initialization CCCR.INIT is reset. A write to the Timeout Counter Value register TOCV presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]) and continues down-counting. When the counter reaches zero, interrupt flag Timeout Occurred (IR.TOO) is set, and the counter is immediately restarted at Timeout Period (TOCC.TOP[15:0]).
- When Timeout Select TOCC.TOS ≠ "00" (counter is controlled by one of the FIFOs), an empty FIFO presets the counter to the value configured by Timeout Period (TOCC.TOP[15:0]). Down-counting is started when the first FIFO element is stored. In this setting, writing to Timeout Counter Value (TOCV) has no effect. When the counter reaches zero, interrupt flag Timeout Occurred (IR.TOO) is set, and the Timeout Counter is stopped.

5.12 Error Counter Register (ECR)

Holds the values of the Error Counters.

Note:

- *When Restricted Operation Mode (CCCR.ASM) is set, the CAN protocol controller does not increment the Transmit Error Counter (ECR.TEC[7:0]) and Receive Error Counter (ECR.REC[6:0]) when a CAN protocol error is detected, but CAN Error Logging (ECR.CEL[7:0]) is still incremented.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	CEL[7:0]							
Attribute	R							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	RP	REC[6:0]						
Attribute	R	R						
Initial value	0	0000000						

bit	7	6	5	4	3	2	1	0
Field	TEC[7:0]							
Attribute	R							
Initial value	0x00							

[bit31:24] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit23:16] CEL[7:0]: CAN Error Logging

The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL.

The counter stops at 0xFF; the next increment of Transmit Error Counter (ECR.TEC[7:0]) or Receive Error Counter (ECR.REC[6:0]) sets interrupt flag Error Logging Overflow (IR.ELO).

[bit15] RP: Receive Error Passive

bit	Description
0	The Receive Error Counter is below the error passive level of 128.
1	The Receive Error Counter has reached the error passive level of 128.

[bit14:8] REC[6:0]: Receive Error Counter

REC[6:0]	Description
0-127	Actual state of the Receive Error Counter, values between 0 and 127.

[bit7:0] TEC[7:0]: Transmit Error Counter

TEC[7:0]	Description
0-255	Actual state of the Transmit Error Counter, values between 0 and 255.

5.13 Protocol Status Register (PSR)

The Protocol Status Register displays the CAN protocol status of the CAN FD Controller.

Note:

- *The PSR register accepts only 16 bit half-word and 32 bit word accesses. 8 bit byte accesses to this register are prohibited.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0xXX							

bit	15	14	13	12	11	10	9	8
Field	Reserved		REDL	RBRS	RESI	FLEC[2:0]		
Attribute	-		R	R	R	R		
Initial value	XX		0	0	0	111		

bit	7	6	5	4	3	2	1	0
Field	BO	EW	EP	ACT[1:0]		LEC[2:0]		
Attribute	R	R	R	R		R		
Initial value	0	0	0	00		111		

[bit31:14] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit13] REDL: Received a CAN FD Message

This bit is set independent of acceptance filtering. A read access will reset this bit to "0".

bit	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received.
1	Message in CAN FD format with EDL flag set has been received.

Note:

- *Once set to "1", REDL will be held at "1" until it is reset to "0" by the CPU.*

[bit12] RBRS: BRS flag of last received CAN FD Message

This bit is set together with "Received a CAN FD Message" (PSR.REDL), independent of acceptance filtering. A read access will reset this bit to "0".

bit	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set.

[bit11] RESI: ESI flag of last received CAN FD Message

This bit is set together with "Received a CAN FD Message" (PSR.REDL), independent of acceptance filtering. A read access will reset this bit to "0".

bit	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

[bit10:8] FLEC[2:0]: Fast Last Error Code

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for Last Error Code (PSR.LEC[2:0]).

This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

A read access will set this field to 7.

[bit7] BO: Bus_Off Status

bit	Description
0	The CAN FD Controller is not Bus_Off.
1	The CAN FD Controller is in Bus_Off state.

[bit6] EW: Warning Status

bit	Description
0	Both error counters are below the Error_Warning limit of 96.
1	At least one of error counter has reached the Error_Warning limit of 96.

[bit5] EP: Error Passive

bit	Description
0	The CAN FD Controller is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The CAN FD Controller is in the Error_Passive state.

[bit4:3] ACT[1:0]: Activity

Monitors the CAN communication state. This field is reset to "00" when Initialize (CCCR.INIT) is set to "1".

ACT[1:0]	Description
00	Synchronizing - node is synchronizing on CAN communication.
01	Idle - node is neither receiver nor transmitter.
10	Receiver - node is operating as receiver.
11	Transmitter - node is operating as transmitter.

[bit2:0] LEC[2:0]: Last Error Code

The LEC indicates the type of the last error to occur on the CAN bus.

This field will be cleared to 0 when a message has been transferred (reception or transmission) without error.

LEC[2:0]	Description
0	No Error: No error occurred since LEC has been reset by successful reception or transmission.
1	Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	Form Error: A fixed format part of a received frame has the wrong format.
3	AckError: The message transmitted by the CAN FD Controller was not acknowledged by another node.
4	Bit1Error: During the transmission of a message (with the exception of the arbitration field), the CAN FD Controller wanted to send a recessive level, but the monitored bus value was dominant.
5	Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the CAN FD Controller wanted to send a dominant level, but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
7	NoChange: Any read access to the Protocol Status Register (PSR) re-initializes the Last Error Code field (LEC[2:0]) to 7. When the LEC[2:0] shows the value 7, no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

Notes:

- *When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in the Fast Last Error Code field (PSR.FLEC[2:0]) instead of the Last Error Code field (PSR.LEC[2:0]). An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.*
- *The Bus_Off recovery sequence (see CAN Specification Rev. 2.0 or ISO11898-1) cannot be shortened by setting or resetting Initialization (CCCR.INIT). If the CAN FD Controller goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the CAN FD Controller will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Receive and Transmit Error Counters (ECR.REC[6:0] and ECR.TEC[7:0]) will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the Last Error Code field (PSR.LEC[2:0]), enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the Bus_Off recovery sequence. The Receive Error Counter (ECR.REC[6:0]) is used to count these sequences*

5.14 Interrupt Register (IR)

The Interrupt Register holds the flags that are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the CPU clears them. A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

The configuration of Interrupt Enable (IE) controls whether an interrupt is generated.

The configuration of Interrupt Line Select (ILS) controls on which interrupt line (canfd_int0/1) an interrupt is signalled.

Note:

- *The TEFW, RF1W, and RF0W interrupts are asserted only when their respective FIFO levels are equal to their configured watermarks. If at this point the interrupt is cleared, and the respective FIFO continues to be filled, the interrupt will not be re-asserted even if the respective FIFO level is greater than the configured watermark.*

bit	31	30	29	28	27	26	25	24
Field	STE	FOE	ACKE	BE	CRCE	WDI	BO	EW
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31] STE: Stuff Error

bit	Description
0	No Stuff Error detected.
1	More than 5 equal bits in a sequence occurred.

[bit30] FOE: Format Error

bit	Description
0	No Format Error detected.
1	A fixed format part of a received frame has the wrong format.

[bit29] ACKE: Acknowledge Error

bit	Description
0	No Acknowledge Error detected.
1	A transmitted message was not acknowledged by another node.

[bit28] BE: Bit Error

bit	Description
0	No Bit0Error/Bit1Error detected.
1	CAN FD Controller wanted to send a recessive/dominant level, but monitored bus level was dominant/recessive.

[bit27] CRCE: CRC Error

bit	Description
0	No CRC Error detected.
1	Received CRC did not match the calculated CRC.

[bit26] WDI: Watchdog Interrupt

bit	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to incomplete access of Message RAM.

[bit25] BO: Bus_Off Status

bit	Description
0	Bus_Off status unchanged.
1	Bus_Off status changed.

[bit24] EW: Warning Status

bit	Description
0	Error_Warning status unchanged.
1	Error_Warning status changed.

[bit23] EP: Error Passive

bit	Description
0	Error_Passive status unchanged.
1	Error_Passive status changed.

[bit22] ELO: Error Logging Overflow

bit	Description
0	CAN Error Logging Counter (ECR.CEL[7:0]) did not overflow.
1	Overflow of CAN Error Logging Counter (ECR.CEL[7:0]) occurred.

[bit21] BEU: Bit Error Uncorrected

Message RAM bit error during access from CAN FD Controller detected, but could not be corrected by the ECC (Error Correction Code) logic attached to the Message RAM.

An uncorrected Message RAM bit error sets Initialization (CCCR.INIT) to "1". This is done to avoid transmission of corrupted data.

bit	Description
0	No bit error detected when reading from Message RAM.
1	Bit error detected, but could not be corrected.

Note:

- *The corresponding bit in the ECC logic must also be cleared when clearing IR.BEU.*

[bit20] BEC: Bit Error Corrected

Message RAM bit error during access from CAN FD Controller detected and corrected by the ECC logic attached to the Message RAM.

bit	Description
0	No bit error detected when reading from Message RAM.
1	Bit error detected and corrected by ECC logic.

Note:

- *The corresponding bit in the ECC logic must also be cleared when clearing IR.BEC.*

[bit19] DRX: Message stored to Dedicated Rx Buffer

The flag is set whenever a received message has been stored into a dedicated Rx Buffer. This flag is not set by reception of a Debug message.

bit	Description
0	No Rx Buffer updated.
1	At least one received message stored into an Rx Buffer.

[bit18] TOO: Timeout Occurred

bit	Description
0	No timeout.
1	Timeout reached.

[bit17] MRAF: Message RAM Access Failure

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. Such behavior will occur also when a Protocol Exception Event is detected, since this will cause the Rx Handler to halt for that frame (see "3.4.5. Protocol Exception Event").
- completed message reception operation, but was not able to write the message to the Message RAM. In this case message storage is aborted.

In both cases the Rx FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN FD Controller is switched into Restricted Operation Mode (see "3.1.5. Restricted Operation Mode"). To leave Restricted Operation Mode, the CPU has to reset the Restricted Operation Mode bit (CCCR.ASM). See "4.6.2. Message RAM access failure Handling Operation" for a detailed procedure of how to leave the Restricted Operation Mode.

bit	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

[bit16] TSW: Timestamp Wraparound

bit	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

[bit15] TEFL: Tx Event FIFO Element Lost

bit	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost. Also set after write attempt to Tx Event FIFO of size zero.

[bit14] TEFF: Tx Event FIFO Full

bit	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

[bit13] TEFW: Tx Event FIFO Watermark Reached

bit	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

[bit12] TEFN: Tx Event FIFO New Entry

bit	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

[bit11] TFE: Tx FIFO Empty

bit	Description
0	Tx FIFO non-empty.
1	Tx FIFO empty.

Note:

- *The TFE bit will not be asserted for an empty Tx Queue.*

[bit10] TCF: Transmission Cancellation Finished

bit	Description
0	No transmission cancellation finished.
1	Transmission cancellation finished.

[bit9] TC: Transmission Completed

bit	Description
0	No transmission completed.
1	Transmission completed.

[bit8] HPM: High Priority Message

bit	Description
0	No high priority message received.
1	High priority message received.

[bit7] RF1L: Rx FIFO 1 Message Lost

bit	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. Also set after write attempt to Rx FIFO 1 of size zero.

[bit6] RF1F: Rx FIFO 1 Full

bit	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

[bit5] RF1W: Rx FIFO 1 Watermark Reached

bit	Description
0	Rx FIFO 1 fill level below watermark.
1	Rx FIFO 1 fill level reached watermark.

[bit4] RF1N: Rx FIFO 1 New Message

bit	Description
0	No new message written to Rx FIFO 1.
1	New message written to Rx FIFO 1.

[bit3] RF0L: Rx FIFO 0 Message Lost

bit	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. Also set after write attempt to Rx FIFO 0 of size zero.

[bit2] RF0F: Rx FIFO 0 Full

bit	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

[bit1] RF0W: Rx FIFO 0 Watermark Reached

bit	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

[bit0] RF0N: Rx FIFO 0 New Message

bit	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

5.15 Interrupt Enable (IE)

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register (IR) will be signalled on an interrupt line.

bit	Description
0	Interrupt disabled.
1	Interrupt enabled.

bit	31	30	29	28	27	26	25	24
Field	STEE	FOEE	ACKEE	BEE	CRCEE	WDIE	BOE	EWE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31] STEE: Stuff Error Interrupt Enable

[bit30] FOEE: Format Error Interrupt Enable

[bit29] ACKEE: Acknowledge Error Interrupt Enable

[bit28] BEE: Bit Error Interrupt Enable

[bit27] CRCEE: CRC Error Interrupt Enable

[bit26] WDIE: Watchdog Interrupt Enable

[bit25] BOE: Bus_Off Status Interrupt Enable

[bit24] EWE: Warning Status Interrupt Enable

[bit23] EPE: Error Passive Interrupt Enable

[bit22] ELOE: Error Logging Overflow Interrupt Enable

[bit21] BEUE: Bit Error Uncorrected Interrupt Enable

[bit20] BECE: Bit Error Corrected Interrupt Enable

[bit19] DRXE: Message stored to Dedicated Rx Buffer Interrupt Enable

[bit18] TOOE: Timeout Occurred Interrupt Enable

[bit17] MRAFE: Message RAM Access Failure Interrupt Enable

[bit16] TSWE: Timestamp Wraparound Interrupt Enable

[bit15] TEFLE: Tx Event FIFO Event Lost Interrupt Enable

[bit14] TEFFE: Tx Event FIFO Full Interrupt Enable

[bit13] TEFWE: Tx Event FIFO Watermark Reached Interrupt Enable

[bit12] TEFNE: Tx Event FIFO New Entry Interrupt Enable

[bit11] TFEE: Tx FIFO Empty Interrupt Enable

[bit10] TCFE: Transmission Cancellation Finished Interrupt Enable

[bit9] TCE: Transmission Completed Interrupt Enable

[bit8] HPME: High Priority Message Interrupt Enable

[bit7] RF1LE: Rx FIFO 1 Message Lost Interrupt Enable

[bit6] RF1FE: Rx FIFO 1 Full Interrupt Enable

[bit5] RF1WE: Rx FIFO 1 Watermark Reached Interrupt Enable

[bit4] RF1NE: Rx FIFO 1 New Message Interrupt Enable

[bit3] RF0LE: Rx FIFO 0 Message Lost Interrupt Enable

[bit2] RF0FE: Rx FIFO 0 Full Interrupt Enable

[bit1] RF0WE: Rx FIFO 0 Watermark Reached Interrupt Enable

[bit0] RF0NE: Rx FIFO 0 New Message Interrupt Enable

5.16 Interrupt Line Select (ILS)

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register (IR) to one of the two CAN FD Controller interrupt lines (canfd_int0/1).

bit	Description
0	Interrupt assigned to interrupt line canfd_int0.
1	Interrupt assigned to interrupt line canfd_int1.

bit	31	30	29	28	27	26	25	24
Field	STEL	FOEL	ACKEL	BEL	CRCEL	WDIL	BOL	EWL
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31] STEL: Stuff Error Interrupt Line

[bit30] FOEL: Format Error Interrupt Line

[bit29] ACKEL: Acknowledge Error Interrupt Line

[bit28] BEL: Bit Error Interrupt Line

[bit27] CRCEL: CRC Error Interrupt Line

[bit26] WDIL: Watchdog Interrupt Line

[bit25] BOL: Bus_Off Status Interrupt Line

[bit24] EWL: Warning Status Interrupt Line

[bit23] EPL: Error Passive Interrupt Line

[bit22] ELOL: Error Logging Overflow Interrupt Line

[bit21] BEUL: Bit Error Uncorrected Interrupt Line

[bit20] BECL: Bit Error Corrected Interrupt Line

[bit19] DRXL: Message stored to Dedicated Rx Buffer Interrupt Line

[bit18] TOOL: Timeout Occurred Interrupt Line

[bit17] MRAFL: Message RAM Access Failure Interrupt Line

[bit16] TSWL: Timestamp Wraparound Interrupt Line

[bit15] TEFLL: Tx Event FIFO Event Lost Interrupt Line

[bit14] TEFFL: Tx Event FIFO Full Interrupt Line

[bit13] TEFWL: Tx Event FIFO Watermark Reached Interrupt Line

[bit12] TEFNL: Tx Event FIFO New Entry Interrupt Line

[bit11] TFEL: Tx FIFO Empty Interrupt Line

[bit10] TCFL: Transmission Cancellation Finished Interrupt Line

[bit9] TCL: Transmission Completed Interrupt Line

[bit8] HPML: High Priority Message Interrupt Line

[bit7] RF1LL: Rx FIFO 1 Message Lost Interrupt Line

[bit6] RF1FL: Rx FIFO 1 Full Interrupt Line

[bit5] RF1WL: Rx FIFO 1 Watermark Reached Interrupt Line

[bit4] RF1NL: Rx FIFO 1 New Message Interrupt Line

[bit3] RF0LL: Rx FIFO 0 Message Lost Interrupt Line

[bit2] RF0FL: Rx FIFO 0 Full Interrupt Line

[bit1] RF0WL: Rx FIFO 0 Watermark Reached Interrupt Line

[bit0] RF0NL: Rx FIFO 0 New Message Interrupt Line

5.17 Interrupt Line Enable (ILE)

Interrupt Line Enable can separately enable/disable each of the two interrupt lines to the CPU by the values set to Enable Interrupt Line 0 (ILE.EINT0) and Enable Interrupt Line 1 (ILE.EINT1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved						EINT1	EINT0
Attribute	-						R/W	R/W
Initial value	000000						0	0

[bit31:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1] EINT1: Enable Interrupt Line 1

bit	Description
0	Interrupt line canfd_int1 disabled.
1	Interrupt line canfd_int1 enabled.

[bit0] EINT0: Enable Interrupt Line 0

bit	Description
0	Interrupt line canfd_int0 disabled.
1	Interrupt line canfd_int0 enabled.

5.18 Global Filter Configuration (GFC)

Global settings for Message ID filtering. The Global Filter Configuration (GFC) controls the filter path for standard and extended messages as described in Figure 3-4 and Figure 3-5.

Write access to the Global Filter Configuration (GFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved		ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
Attribute	-		R/W		R/W		R/W	R/W
Initial value	00		00		00		0	0

[bit31:6] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit5:4] ANFS[1:0]: Accept Non-matching Frames Standard

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

ANFS[1:0]	Description
00	Accept in Rx FIFO 0.
01	Accept in Rx FIFO 1.
10	Reject.
11	Reject.

[bit3:2] ANFE[1:0]: Accept Non-matching Frames Extended

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

ANFE[1:0]	Description
00	Accept in Rx FIFO 0.
01	Accept in Rx FIFO 1.
10	Reject.
11	Reject.

[bit1] RRFS: Reject Remote Frames Standard

bit	Description
0	Filter remote frames with 11-bit standard IDs.
1	Reject all remote frames with 11-bit standard IDs.

[bit0] RRFE: Reject Remote Frames Extended

bit	Description
0	Filter remote frames with 29-bit extended IDs.
1	Reject all remote frames with 29-bit extended IDs.

5.19 Standard ID Filter Configuration (SIDFC)

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages as described in Figure 3-4.

Write access to the Standard ID Filter Configuration (SIDFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	LSS[7:0]							
Attribute	R/W							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	FLSSA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	FLSSA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31:24] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit23:16] LSS[7:0]: List Size Standard

LSS[7:0]	Description
0	No standard Message ID filter.
1-128	Number of standard Message ID filter elements.
>128	Values greater than 128 are interpreted as 128.

[bit15:2] FLSSA[15:2]: Filter List Standard Start Address
Start address of standard Message ID filter list (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

5.20 Extended ID Filter Configuration (XIDFC)

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for extended messages as described in Figure 3-5.

Write access to the Extended ID Filter Configuration (XIDFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved	LSE[6:0]						
Attribute	-	R/W						
Initial value	0	0000000						

bit	15	14	13	12	11	10	9	8
Field	FLESA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	FLESA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31:23] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

[bit22:16] LSE[6:0]: List Size Extended

LSE[6:0]	Description
0	No extended Message ID filter.
1-64	Number of extended Message ID filter elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] FLESA[15:2]: Filter List Extended Start Address
Start address of extended Message ID filter list (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits
When writing, always write "0". When reading, "0" is always read.

5.21 Extended ID AND Mask (XIDAM)

The Extended ID AND Mask defines the valid bits of a 29-bit ID for acceptance filtering.

Write access to the Extended ID AND Mask (XIDAM) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved			EIDM[28:24]				
Attribute		-				R/W		
Initial value		000				11111		

bit	23	22	21	20	19	18	17	16
Field	EIDM[23:16]							
Attribute				R/W				
Initial value				0xff				

bit	15	14	13	12	11	10	9	8
Field	EIDM[15:8]							
Attribute				R/W				
Initial value				0xff				

bit	7	6	5	4	3	2	1	0
Field	EIDM[7:0]							
Attribute				R/W				
Initial value				0xff				

[bit31:29] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit28:0] EIDM[28:0]: Extended ID Mask

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame.

With the reset value of all bits set to one the mask is not active.

5.22 High Priority Message Status (HPMS)

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0xXX							

bit	15	14	13	12	11	10	9	8
Field	FLST	FIDX[6:0]						
Attribute	R	R						
Initial value	0	0000000						

bit	7	6	5	4	3	2	1	0
Field	MSI[1:0]		BIDX[5:0]					
Attribute	R		R					
Initial value	00		000000					

[bit31:16] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit15] FLST: Filter List

Indicates the filter list of the matching filter element.

bit	Description
0	Standard Filter List.
1	Extended Filter List.

[bit14:8] FIDX[6:0]: Filter Index

FIDX[6:0]	Description
0-127	Index of matching Rx acceptance filter element. Range is 0 to List Size Standard/Extended minus 1 (i.e. SIDFC.LSS[7:0] - 1 resp. XIDFC.LSE[6:0] - 1).

[bit7:6] MSI[1:0]: Message Storage Indicator

MSI[1:0]	Description
00	No Rx FIFO selected.
01	Rx FIFO message lost.
10	Message stored in Rx FIFO 0.
11	Message stored in Rx FIFO 1.

[bit5:0] BIDX[5:0]: Buffer Index

Index of Rx FIFO element to which the message was stored. Only valid when bit[1] of the Message Storage Indicator MSI[1] = "1".

5.23 New Data 1 (NDAT1)

New Data 1 holds flags that are set when the respective dedicated Rx Buffer receives a frame.

bit	31	30	29	28	27	26	25	24
Field	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] ND[31:0]: New Data

The register holds the New Data flags of dedicated Rx Buffers 0 to 31.

The flags are set when the respective dedicated Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them.

A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

bit	Description
0	Rx Buffer not updated.
1	Rx Buffer updated from new message.

5.24 New Data 2 (NDAT 2)

New Data 2 holds flags that are set when the respective dedicated Rx Buffer receives a frame.

bit	31	30	29	28	27	26	25	24
Field	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] ND[63:32]: New Data

The register holds the New Data flags of dedicated Rx Buffers 32 to 63.

The flags are set when the respective dedicated Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them.

A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. A hard reset will clear the register.

bit	Description
0	Rx Buffer not updated.
1	Rx Buffer updated from new message.

5.25 Rx FIFO 0 Configuration (RXF0C)

Settings for the Rx FIFO 0.

Write access to the Rx FIFO 0 Configuration (RXF0C) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	F0OM	F0WM[6:0]						
Attribute	R/W	R/W						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved	F0S[6:0]						
Attribute	-	R/W						
Initial value	0	0000000						

bit	15	14	13	12	11	10	9	8
Field	F0SA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	F0SA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31] F0OM: FIFO 0 Operation Mode

Rx FIFO 0 can be operated in blocking or in overwrite mode (see "3.4.2. Rx FIFOs").

bit	Description
0	Rx FIFO 0 blocking mode.
1	Rx FIFO 0 overwrite mode.

[bit30:24] F0WM[6:0]: Rx FIFO 0 Watermark

F0WM[6:0]	Description
0	Watermark interrupt disabled.
1-64	Level for Rx FIFO 0 Watermark Reached interrupt (IR.RF0W).
>64	Watermark interrupt disabled.

[bit23] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit22:16] F0S[6:0]: Rx FIFO 0 Size

The Rx FIFO 0 elements are indexed from 0 to F0S[6:0] - 1.

F0S[6:0]	Description
0	No Rx FIFO 0.
1-64	Number of Rx FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] F0SA[15:2]: Rx FIFO 0 Start Address

Start address of Rx FIFO 0 in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.26 Rx FIFO 0 Status (RXF0S)

Status of the Rx FIFO 0.

bit	31	30	29	28	27	26	25	24
Field	Reserved						RF0L	F0F
Attribute	-						R	R
Initial value	XXXXXX						0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved		F0PI[5:0]					
Attribute	-		R					
Initial value	XX		000000					

bit	15	14	13	12	11	10	9	8
Field	Reserved		F0GI[5:0]					
Attribute	-		R					
Initial value	X		000000					

bit	7	6	5	4	3	2	1	0
Field	Reserved	F0FL[6:0]						
Attribute	-	R						
Initial value	X	0000000						

[bit31:26] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit25] RF0L: Rx FIFO 0 Message Lost

This bit is a copy of interrupt flag IR.RF0L (Rx FIFO 0 Message Lost).

When IR.RF0L is reset, this bit is also reset.

bit	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. Also set after write attempt to Rx FIFO 0 of size zero.

Note:

- *Overwriting a message when the FIFO is in overwrite mode (RXF0C.F0OM = "1") will not set this flag.*

[bit24] F0F: Rx FIFO 0 Full

bit	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

[bit23:22] Reserved: Reserved bits
When writing, always write "0". The read value is undefined.

[bit21:16] F0PI[5:0]: Rx FIFO 0 Put Index

F0PI[5:0]	Description
0-63	Rx FIFO 0 write index pointer, range 0 to 63.

[bit15:14] Reserved: Reserved bits
When writing, always write "0". The read value is undefined.

[bit13:8] F0GI[5:0]: Rx FIFO 0 Get Index

F0GI[5:0]	Description
0-63	Rx FIFO 0 read index pointer, range 0 to 63.

[bit7] Reserved: Reserved bit
When writing, always write "0". The read value is undefined.

[bit6:0] F0FL[6:0]: Rx FIFO 0 Fill Level

F0FL[6:0]	Description
0-64	Number of elements stored in Rx FIFO 0, range 0 to 64.

5.27 Rx FIFO 0 Acknowledge (RXF0A)

The Rx FIFO 0 Acknowledge is used to acknowledge that the CPU has read a message or a sequence of messages from the Rx FIFO 0 to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved		F0AI[5:0]					
Attribute	-		R/W					
Initial value	00		000000					

[bit31:6] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit5:0] F0AI[5:0]: Rx FIFO 0 Acknowledge Index

After the CPU has read a message or a sequence of messages from Rx FIFO 0, it has to write the buffer index of the last element read from Rx FIFO 0 to this field (F0AI[5:0]). This will set the Rx FIFO 0 Get Index RXF0S.F0GI[5:0] to F0AI[5:0] + 1 and update the FIFO 0 Fill Level RXF0S.F0FL[6:0].

5.28 Rx Buffer Configuration (RXBC)

Defines the start address of the Rx Buffers section in the Message RAM.

Write access to the Rx Buffer Configuration (RXBC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	RBSA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	RBSA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31:16] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit15:2] RBSA[15:2]: Rx Buffer Start Address

Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).

Also used to reference debug messages A, B, C.

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.29 Rx FIFO 1 Configuration (RXF1C)

Settings for the Rx FIFO 1.

Write access to the Rx FIFO 1 Configuration (RXF1C) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	F1OM	F1WM[6:0]						
Attribute	R/W	R/W						
Initial value	0	0000000						

bit	23	22	21	20	19	18	17	16
Field	Reserved	F1S[6:0]						
Attribute	-	R/W						
Initial value	0	0000000						

bit	15	14	13	12	11	10	9	8
Field	F1SA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	F1SA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31] F1OM: FIFO 1 Operation Mode

Rx FIFO 1 can be operated in blocking or in overwrite mode (see "3.4.2. Rx FIFOs").

bit	Description
0	Rx FIFO 1 blocking mode.
1	Rx FIFO 1 overwrite mode.

[bit30:24] F1WM[6:0]: Rx FIFO 1 Watermark

F1WM[6:0]	Description
0	Watermark interrupt disabled.
1-64	Level for Rx FIFO 1 Watermark Reached interrupt (IR.RF1W).
>64	Watermark interrupt disabled.

[bit23] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit22:16] F1S[6:0]: Rx FIFO 1 Size

The Rx FIFO 1 elements are indexed from 0 to F1S[6:0] - 1.

F1S[6:0]	Description
0	No Rx FIFO 1.
1-64	Number of Rx FIFO 1 elements.
>64	Values greater than 64 are interpreted as 64.

[bit15:2] F1SA[15:2]: Rx FIFO 1 Start Address

Start address of Rx FIFO 1 in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.30 Rx FIFO 1 Status (RXF1S)

Status of the Rx FIFO 1.

bit	31	30	29	28	27	26	25	24
Field	DMS[1:0]		Reserved				RF1L	F1F
Attribute	R		-				R	R
Initial value	00		XXXX				0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved		F1PI[5:0]					
Attribute	-		R					
Initial value	XX		000000					

bit	15	14	13	12	11	10	9	8
Field	Reserved		F1GI[5:0]					
Attribute	-		R					
Initial value	X		000000					

bit	7	6	5	4	3	2	1	0
Field	Reserved	F1FL[6:0]						
Attribute	-	R						
Initial value	X	0000000						

[bit31:30] DMS[1:0]: Debug Message Status

DMS[1:0]	Description
00	Idle state, wait for reception of debug messages, DMA request is cleared.
01	Debug message A received.
10	Debug messages A, B received.
11	Debug messages A, B, C received, DMA request is set.

[bit29:26] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit25] RF1L: Rx FIFO 1 Message Lost

This bit is a copy of interrupt flag IR.RF1L (Rx FIFO 1 Message Lost).

When IR.RF1L is reset, this bit is also reset.

bit	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. Also set after write attempt to Rx FIFO 1 of size zero.

Note:

- *Overwriting a message when the FIFO is in overwrite mode (RXF1C.F1OM = "1") will not set this flag.*

[bit24] F1F: Rx FIFO 1 Full

bit	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

[bit23:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit21:16] F1PI[5:0]: Rx FIFO 1 Put Index

F1PI[5:0]	Description
0-63	Rx FIFO 1 write index pointer, range 0 to 63.

[bit15:14] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit13:8] F1GI[5:0]: Rx FIFO 1 Get Index

F1GI[5:0]	Description
0-63	Rx FIFO 1 read index pointer, range 0 to 63.

[bit7] Reserved: Reserved bit

When writing, always write "0". The read value is undefined.

[bit6:0] F1FL[6:0]: Rx FIFO 1 Fill Level

F1FL[6:0]	Description
0-64	Number of elements stored in Rx FIFO 1, range 0 to 64.

5.31 Rx FIFO 1 Acknowledge (RXF1A)

The Rx FIFO 1 Acknowledge is used to acknowledge that the CPU has read a message or a sequence of messages from the Rx FIFO 1 to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved		F1AI[5:0]					
Attribute	-		R/W					
Initial value	00		000000					

[bit31:6] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit5:0] F1AI[5:0]: Rx FIFO 1 Acknowledge Index

After the CPU has read a message or a sequence of messages from Rx FIFO 1, it has to write the buffer index of the last element read from Rx FIFO 1 to this field (F1AI[5:0]). This will set the Rx FIFO 1 Get Index RXF1S.F1GI[5:0] to F1AI[5:0] + 1 and update the FIFO 1 Fill Level RXF1S.F1FL[6:0].

5.32 Rx Buffer/FIFO Element Size Configuration (RXESC)

Configures the number of data bytes belonging to an Rx Buffer and FIFO element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Write access to the Rx Buffer/FIFO Element Size Configuration (RXESC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by this register (i.e. RXESC) are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved					RBDS[2:0]		
Attribute	-					R/W		
Initial value	00000					000		

bit	7	6	5	4	3	2	1	0
Field	Reserved	F1DS[2:0]			Reserved	F0DS[2:0]		
Attribute	-	R/W			-	R/W		
Initial value	0	000			0	000		

[bit31:11] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit10:8] RBDS[2:0]: Rx Buffer Data Field Size

RBDS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

[bit7] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit6:4] F1DS[2:0]: Rx FIFO 1 Data Field Size

F1DS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

[bit3] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit2:0] F0DS[2:0]: Rx FIFO 0 Data Field Size

F0DS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

5.33 Tx Buffer Configuration (TXBC)

Settings for Tx Buffers stored in the Message RAM.

Write access to the Tx Buffer Configuration (TXBC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Notes:

- *Be aware that the sum of Transmit FIFO/Queue Size (TXBC.TFQS[5:0]) and Number of Dedicated Transmit Buffers (TXBC,NDTB[5:0]) may be not greater than 32. There is no check for erroneous configurations.*
- *The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.*

bit	31	30	29	28	27	26	25	24
Field	Reserved	TFQM	TFQS[5:0]					
Attribute	-	R/W	R/W					
Initial value	0	0	000000					

bit	23	22	21	20	19	18	17	16
Field	Reserved		NDTB[5:0]					
Attribute	-		R/W					
Initial value	00		000000					

bit	15	14	13	12	11	10	9	8
Field	TBSA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	TBSA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31] Reserved: Reserved bit

When writing, always write "0". When reading, "0" is always read.

[bit30] TFQM: Tx FIFO/Queue Mode

bit	Description
0	Tx FIFO operation.
1	Tx Queue operation.

[bit29:24] TFQS[5:0]: Transmit FIFO/Queue Size

TFQS[5:0]	Description
0	No Tx FIFO/Queue.
1-32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

[bit23:22] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit21:16] NDTB[5:0]: Number of Dedicated Transmit Buffers

NDTB[5:0]	Description
0	No Dedicated Tx Buffers.
1-32	Number of Dedicated Tx Buffers.
>32	Values greater than 32 are interpreted as 32.

[bit15:2] TBSA[15:2]: Tx Buffers Start Address

Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.34 Tx FIFO/Queue Status (TXFQS)

The Tx FIFO/Queue status is related to the pending Tx requests listed in the Tx Buffer Request Pending register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

Note:

- *In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices (TXFQS.TFQPI[4:0] resp. TXFQS.TFGI[4:0]) indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.*

Example:

- *For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0xXX							

bit	23	22	21	20	19	18	17	16
Field	Reserved		TFQF	TFQPI[4:0]				
Attribute	-		R	R				
Initial value	XX		0	00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved			TFGI[4:0]				
Attribute	-			R				
Initial value	XXX			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved		TFFL[5:0]					
Attribute	-		R					
Initial value	XX		000000					

[bit31:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit21] TFQF: Tx FIFO/Queue Full

bit	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

[bit20:16] TFQPI[4:0]: Tx FIFO/Queue Put Index

TFQPI[4:0]	Description
0-31	Tx FIFO/Queue write index pointer, range 0 to 31.

[bit15:13] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit12:8] TFGI[4:0]: Tx FIFO Get Index

TFGI[4:0]	Description
0-31	Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (Tx FIFO/Queue Mode TXBC.TFQM = "1").

[bit7:6] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit5:0] TFFL[5:0]: Tx FIFO Free Level

TFFL[5:0]	Description
0-32	Number of consecutive free Tx FIFO elements starting from the Tx FIFO Get Index (TXFQS.TFGI[4:0]), range 0 to 32. Read as zero when Tx Queue operation is configured (Tx FIFO/Queue Mode TXBC.TFQM = "1").

5.35 Tx Buffer Element Size Configuration (TXESC)

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Write access to the Tx Buffer Element Size Configuration (TXESC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

Note:

- *In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS[2:0], the bytes not defined by the Tx Buffer are transmitted as "0xCC" (padding bytes).*

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved					TBDS[2:0]		
Attribute	-					R/W		
Initial value	00000					000		

[bit31:3] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit2:0] TBDS[2:0]: Tx Buffer Data Field Size

TBDS[2:0]	Description
000	8 byte data field.
001	12 byte data field.
010	16 byte data field.
011	20 byte data field.
100	24 byte data field.
101	32 byte data field.
110	48 byte data field.
111	64 byte data field.

5.36 Tx Buffer Request Pending (TXBRP)

Tx Buffer Request Pending holds the status of the transmission requests of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TRP[31:0]: Transmission Request Pending

Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR (Tx Buffer Add Request). The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR (Tx Buffer Cancellation Request).

TXBRP bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields. After a TXBRP bit has been set, a Tx scan (see "3.5. Tx Handling") is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

bit	Description
0	No transmission request pending.
1	Transmission request pending.

Note:

- *TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset.*

5.37 Tx Buffer Add Request (TXBAR)

Tx Buffer Add Request is used to request the transmission of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] AR[31:0]: Add Request

Each Tx Buffer has its own Add Request bit. Writing a "1" will set the corresponding Add Request bit; writing a "0" has no impact. This enables the CPU to set transmission requests for multiple Tx Buffers with one write to TXBAR.

TXBAR bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields.

When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.

bit	Description
0	No transmission request added.
1	Transmission requested added.

Note:

- *If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored.*

5.38 Tx Buffer Cancellation Request (TXBCR)

Used to cancel transmission requests of each corresponding Tx Buffer.

bit	31	30	29	28	27	26	25	24
Field	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CR[31:0]: Cancellation Request

Each Tx Buffer has its own Cancellation Request bit. Writing a "1" will set the corresponding Cancellation Request bit; writing a "0" has no impact. This enables the CPU to set cancellation requests for multiple Tx Buffers with one write to TXBCR.

TXBCR bits are set only for those Tx Buffers that are configured by the TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers) fields. The bits remain set until the corresponding bit of TXBRP (Tx Buffer Request Pending) is reset.

bit	Description
0	No cancellation pending.
1	Cancellation pending.

5.39 Tx Buffer Transmission Occurred (TXBTO)

Displays the status of whether the corresponding Tx Buffer has been transmitted or not.

bit	31	30	29	28	27	26	25	24
Field	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TO[31:0]: Transmission Occurred

Each Tx Buffer has its own Transmission Occurred bit.

The bits are set when the corresponding TXBRP (Tx Buffer Request Pending) bit is cleared after a successful transmission.

The bits are reset when a new transmission is requested by writing a "1" to the corresponding bit of register TXBAR (Tx Buffer Add Request).

bit	Description
0	No transmission occurred.
1	Transmission occurred.

5.40 Tx Buffer Cancellation Finished (TXBCF)

Signals whether the cancellation request of the corresponding Tx Buffer has been successful or not.

bit	31	30	29	28	27	26	25	24
Field	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CF[31:0]: Cancellation Finished

Each Tx Buffer has its own Cancellation Finished bit.

The bits are set when the corresponding TXBRP (Tx Buffer Request Pending) bit is cleared after a cancellation was requested via TXBCR (Tx Buffer Cancellation Request). In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. CF is also set for an unsuccessful transmission when in DAR mode.

The bits are reset when a new transmission is requested by writing a "1" to the corresponding bit of register TXBAR (Tx Buffer Add Request).

bit	Description
0	No transmit buffer cancellation.
1	Transmit buffer cancellation finished.

5.41 Tx Buffer Transmission Interrupt Enable (TXBTIE)

The settings in the Tx Buffer Transmission Interrupt Enable determine which Tx Buffer will assert an interrupt upon transmission.

bit	31	30	29	28	27	26	25	24
Field	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TIE[31:0]: Transmission Interrupt Enable

Each Tx Buffer has its own Transmission Interrupt Enable bit.

bit	Description
0	Transmission interrupt disabled.
1	Transmission interrupt enable.

5.42 Tx Buffer Cancellation Finished Interrupt Enable (TXBCIE)

The settings in the Tx Buffer Cancellation Finished Interrupt Enable determine which Tx Buffer will assert an interrupt upon completion of a transmission cancellation request.

bit	31	30	29	28	27	26	25	24
Field	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] CFIE[31:0]: Cancellation Finished Interrupt Enable

Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

bit	Description
0	Cancellation finished interrupt disabled.
1	Cancellation finished interrupt enabled.

5.43 Tx Event FIFO Configuration (TXEFC)

Settings for the Tx Event FIFO.

Write access to the Tx Event FIFO Configuration (TXEFC) is possible only when the Configuration Change Enable (CCCR.CCE) and Initialization (CCCR.INIT) bits are both "1".

bit	31	30	29	28	27	26	25	24
Field	Reserved		EFWM[5:0]					
Attribute	-		R/W					
Initial value	00		000000					

bit	23	22	21	20	19	18	17	16
Field	Reserved		EFS[5:0]					
Attribute	-		R/W					
Initial value	00		000000					

bit	15	14	13	12	11	10	9	8
Field	EFSA[15:8]							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	EFSA[7:2]						Reserved	
Attribute	R/W						-	
Initial value	000000						00	

[bit31:30] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit29:24] EFWM[5:0]: Event FIFO Watermark

EFWM[5:0]	Description
0	Watermark interrupt disabled.
1-32	Level for Tx Event FIFO Watermark Reached interrupt (IR.TEFW).
>32	Watermark interrupt disabled.

[bit23:22] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit21:16] EFS[5:0]: Event FIFO Size

The Tx Event FIFO elements are indexed from 0 to EFS[5:0] - 1.

EFS[5:0]	Description
0	Tx Event FIFO disabled.
1-32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

[bit15:2] EFSA[15:2]: Event FIFO Start Address

Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 6-1).

[bit1:0] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

5.44 Tx Event FIFO Status (TXEFS)

Status of the Tx Event FIFO.

bit	31	30	29	28	27	26	25	24
Field	Reserved						TEFL	EFF
Attribute	-						R	R
Initial value	XXXXXX						0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved			EFPI[4:0]				
Attribute	-			R				
Initial value	XXX			00000				

bit	15	14	13	12	11	10	9	8
Field	Reserved			EFGI[4:0]				
Attribute	-			R				
Initial value	XXX			00000				

bit	7	6	5	4	3	2	1	0
Field	Reserved		EFFL[5:0]					
Attribute	-		R					
Initial value	XX		000000					

[bit31:26] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit25] TEFL: Tx Event FIFO Element Lost

This bit is a copy of interrupt flag IR.TEFL (Tx Event FIFO Element Lost). When IR.TEFL is reset, this bit is also reset.

bit	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost. Also set after write attempt to Tx Event FIFO of size zero.

[bit24] EFF: Event FIFO Full

bit	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

[bit23:21] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit20:16] EFPI[4:0]: Event FIFO Put Index

EFPI[4:0]	Description
0-31	Tx Event FIFO write index pointer, range 0 to 31.

[bit15:13] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit12:8] EFGI[4:0]: Event FIFO Get Index

EFGI[4:0]	Description
0-31	Tx Event FIFO read index pointer, range 0 to 31.

[bit7:6] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

[bit5:0] EFFL[5:0]: Event FIFO Fill Level

EFFL[5:0]	Description
0-32	Number of elements stored in Tx Event FIFO, range 0 to 32.

5.45 Tx Event FIFO Acknowledge (TXEFA)

The Tx Event FIFO Acknowledge is used to acknowledge that the CPU has read an event from the Tx Event FIFO to indicate to the CAN FD Controller that the corresponding Message RAM area may be released. See "3.6. FIFO Acknowledge Handling" for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved			EFAI[4:0]				
Attribute	-			R/W				
Initial value	000			00000				

[bit31:5] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit4:0] EFAI[4:0]: Event FIFO Acknowledge Index

After the CPU has read an element or a sequence of elements from the Tx Event FIFO, it has to write the index of the last element read from Tx Event FIFO to EFAI[4:0]. This will set the Tx Event FIFO Get Index TXEFS.EFGI[4:0] to EFAI[4:0] + 1 and update the FIFO 0 Fill Level TXEFS.EFFL[5:0].

6. Message RAM

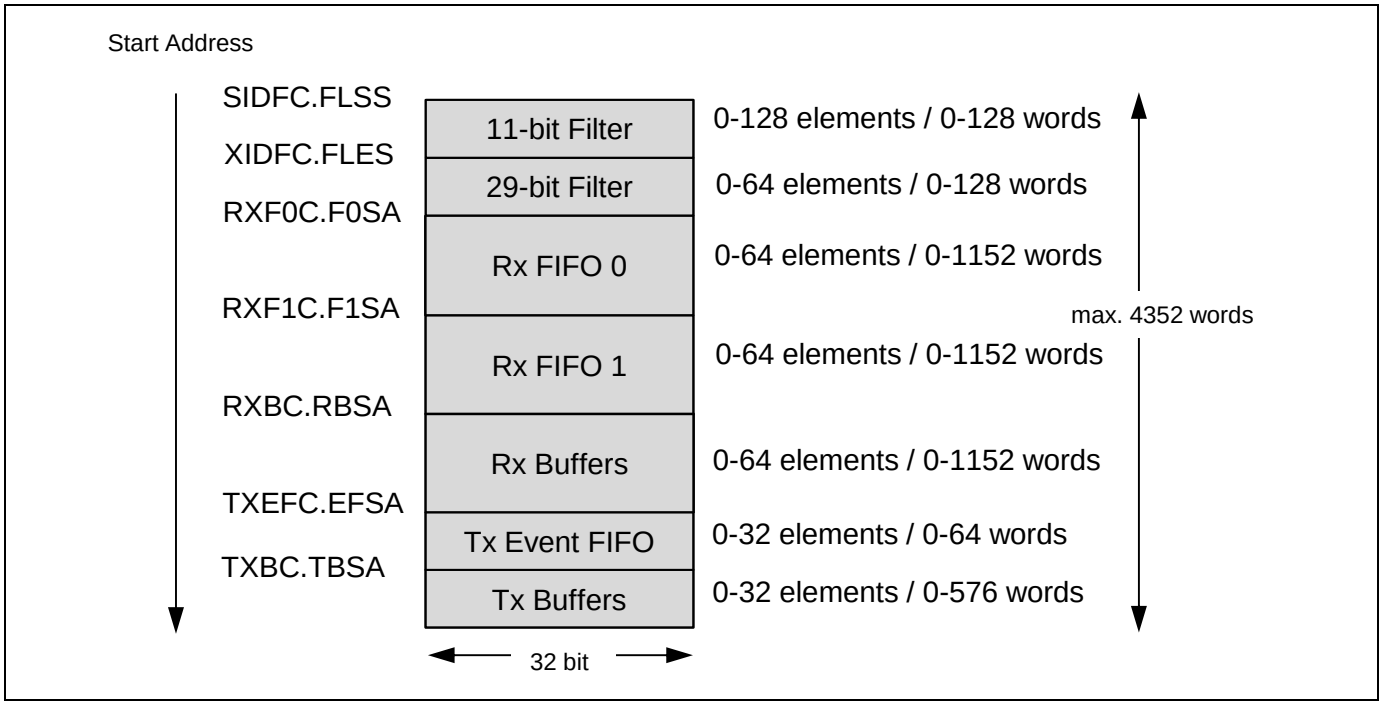
The Message RAM stores Rx/Tx messages and filter configurations.

6.1. Message RAM Configuration
6.2. Rx Buffer and FIFO Element
6.3. Tx Buffer Element
6.4. Tx Event FIFO Element
6.5. Standard Message ID Filter Element
6.6. Extended Message ID Filter Element

6.1 Message RAM Configuration

The Message RAM has a width of 32 bits. The CAN FD Controller can be configured to allocate up to 4352 words in the Message RAM (note that the number of words that can be used will be limited by the size of the actual Message RAM). For the number of words and address, see "CHAPTER: A.Register Map / CAN FD)".It is not necessary to configure each of the sections listed in Figure 6-1, nor is there any restriction with respect to the sequence of the sections.

Figure 6-1 Message RAM Configuration



The CAN FD Controller addresses the Message RAM in 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Notes:

- *The CAN FD Controller does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.*
- *Message RAM accesses by the CPU will take two to four Bus clock cycles.*

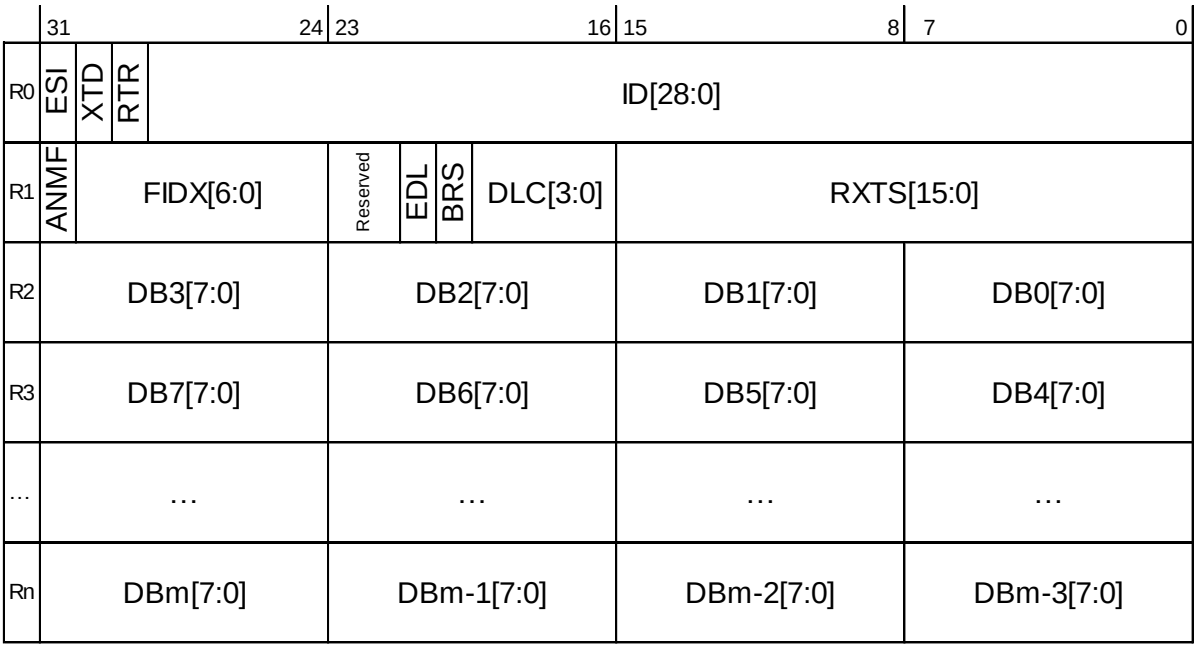
List of Message RAM Elements

Abbreviation	Reference
Rx Buffer and FIFO Element	6.2
Tx Buffer Element	6.3
Tx Event FIFO Element	6.4
Standard Message ID Filter Element	6.5
Extended Message ID Filter Element	6.6

6.2 Rx Buffer and FIFO Element

An Rx Buffer and FIFO Element is a block of 32-bit words that holds the data and status of a received frame that was stored in the Message RAM.

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of an Rx Buffer and FIFO element is shown in the figure below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC (Rx Buffer/FIFO Element Size Configuration).



R0 [bit31] ESI: Error State Indicator

bit	Description
0	Transmitting node is error active.
1	Transmitting node is error passive.

R0 [bit30] XTD: Extended Identifier

Signals to the CPU whether the received frame has a standard or extended identifier.

bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

R0 [bit29] RTR: Remote Transmission Request

Signals to the CPU whether the received frame is a data frame or a remote frame.

bit	Description
0	Received frame is a data frame.
1	Received frame is a remote frame.

Note:

- *There are no remote frames in CAN FD format. In case a CAN FD frame was received (EDL = "1"), bit RTR reflects the state of the reserved bit r1.*

R0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

R1 [bit31] ANMF: Accepted Non-matching Frame

Acceptance of non-matching frames may be enabled via GFC.ANFS[1:0] (Accept Non-matching Frames Standard) and GFC.ANFE[1:0] (Accept Non-matching Frames Extended).

bit	Description
0	Received frame matching filter index FIDX.
1	Received frame did not match any Rx filter element.

R1 [bit30:24] FIDX[6:0]: Filter Index

FIDX[6:0]	Description
0-127	Index of matching Rx acceptance filter element (invalid if ANMF = "1"). Range is 0 to List Size Standard/Extended minus 1 (i.e. SIDFC.LSS - 1 resp. XIDFC.LSE - 1).

R1 [bit23:22] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

R1 [bit21] EDL: Extended Data Length

bit	Description
0	Standard frame format.
1	CAN FD frame format.

R1 [bit20] BRS: Bit Rate Switch

bit	Description
0	Frame received without bit rate switching.
1	Frame received with bit rate switching.

R1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0-8	CAN + CAN FD: received frame has 0-8 data bytes.
9-15	CAN: received frame has 8 data bytes. CAN FD: received frame has 12/16/20/24/32/48/64 data bytes. See Table 3-1 for details.

R1 [bit15:0] RXTS[15:0]: Rx Timestamp

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP[3:0].

R2 [bit31:24]	DB3[7:0] :	Data Byte 3
R2 [bit23:16]	DB2[7:0] :	Data Byte 2
R2 [bit15:8]	DB1[7:0] :	Data Byte 1
R2 [bit7:0]	DB0[7:0] :	Data Byte 0
R3 [bit31:24]	DB7[7:0] :	Data Byte 7
R3 [bit23:16]	DB6[7:0] :	Data Byte 6
R3 [bit15:8]	DB5[7:0] :	Data Byte 5
R3 [bit7:0]	DB4[7:0] :	Data Byte 4
…	…	…
Rn [bit31:24]	DBm[7:0]:	Data Byte m
Rn [bit23:16]	DBm-1[7:0]:	Data Byte m-1
Rn [bit15:8]	DBm-2[7:0]:	Data Byte m-2
Rn [bit7:0]	DBm-3[7:0]:	Data Byte m-3

Notes:

- *Depending on the configuration of the element size (defined by Rx Buffer/FIFO Element Size Configuration (RXESC)), Rn will vary from n = 3 to 17.*
- *m is a function of n, m = (n - 1) × 4 - 1.*
- *The number of valid data bytes are defined by the Data Length Code.*

6.3 Tx Buffer Element

A Tx Buffer Element is a block of 32-bit words stored in the Message RAM that holds data and control information of a frame to be transmitted by the CAN FD Controller.

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO/Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO/Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS[5:0] (Transmit FIFO/Queue Size) and TXBC.NDTB[5:0] (Number of Dedicated Transmit Buffers). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC (Tx Buffer Element Size Configuration).

	31	30	29	28 … 24	23 … 16	15 … 8	7 … 0
T0	Reserved	XTD	RTR	ID[28:0]			
T1	MM[7:0]			EFC	Reserved	DLC[3:0]	Reserved
T2	DB3[7:0]				DB2[7:0]	DB1[7:0]	DB0[7:0]
T3	DB7[7:0]				DB6[7:0]	DB5[7:0]	DB4[7:0]
…	…				…	…	…
Tn	DBm[7:0]				DBm-1[7:0]	DBm-2[7:0]	DBm-3[7:0]

T0 [bit31] Reserved: Reserved bit
When writing, always write "0". The read value is undefined.

T0 [bit30] XTD: Extended Identifier

bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

T0 [bit29] RTR: Remote Transmission Request

bit	Description
0	Transmit data frame.
1	Transmit remote frame.

Note:

- *When RTR = "1", the CAN FD Controller transmits a remote frame according to ISO11898-1, even if the CAN Mode Enable field (CCCR.CME[1:0]) enables the transmission in CAN FD format.*

T0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

T1 [bit31:24] MM[7:0]: Message Marker

Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

T1 [bit23] EFC: Event FIFO Control

bit	Description
0	Don't store Tx events.
1	Store Tx events.

T1 [bit22:20] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

T1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0-8	CAN + CAN FD: transmit frame has 0-8 data bytes.
9-15	CAN: transmit frame has 8 data bytes. CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes. See Table 3-1 for details.

T1 [bit15:0] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

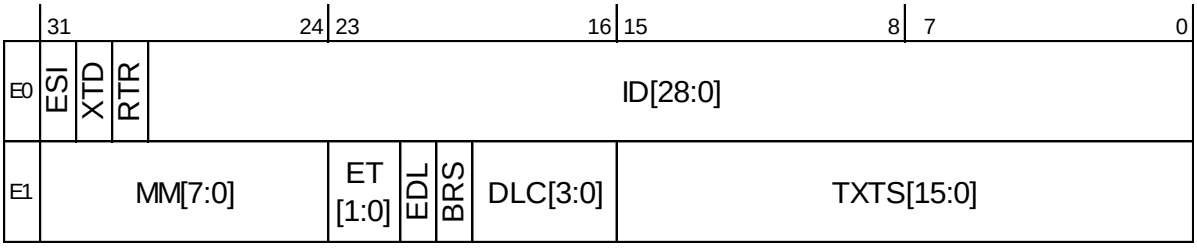
T2 [bit31:24]	DB3[7:0] :	Data Byte 3
T2 [bit23:16]	DB2[7:0] :	Data Byte 2
T2 [bit15:8]	DB1[7:0] :	Data Byte 1
T2 [bit7:0]	DB0[7:0] :	Data Byte 0
T3 [bit31:24]	DB7[7:0] :	Data Byte 7
T3 [bit23:16]	DB6[7:0] :	Data Byte 6
T3 [bit15:8]	DB5[7:0] :	Data Byte 5
T3 [bit7:0]	DB4[7:0] :	Data Byte 4
…	…	…
Tn [bit31:24]	DBm[7:0]:	Data Byte m
Tn [bit23:16]	DBm-1[7:0]:	Data Byte m-1
Tn [bit15:8]	DBm-2[7:0]:	Data Byte m-2
Tn [bit7:0]	DBm-3[7:0]:	Data Byte m-3

Notes:

- *Depending on the configuration of the element size (TXESC), Tn will vary from n = 3 to 17.*
- *m is a function of n: m = (n - 1) × 4 - 1.*

6.4 Tx Event FIFO Element

Each Tx Event FIFO Element stores information about transmitted messages. By reading the Tx Event FIFO the CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS (Tx Event FIFO Status).



E0 [bit31] ESI: Error State Indicator

bit	Description
0	Transmitting node is error active.
1	Transmitting node is error passive.

E0 [bit30] XTD: Extended Identifier

bit	Description
0	11-bit standard identifier.
1	29-bit extended identifier.

E0 [bit29] RTR: Remote Transmission Request

bit	Description
0	Data frame transmitted.
1	Remote frame transmitted.

E0 [bit28:0] ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

E1 [bit31:24] MM[7:0]: Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

E1 [bit23:22] ET[1:0]: Event Type

ET[1:0]	Description
00	Reserved.
01	Tx event.
10	Transmission in spite of cancellation.
11	Reserved.

E1 [bit21] EDL: Extended Data Length

bit	Description
0	Standard frame format.
1	CAN FD frame format (new DLC-coding and CRC).

E1 [bit20] BRS: Bit Rate Switch

bit	Description
0	Frame transmitted without bit rate switching.
1	Frame transmitted with bit rate switching.

E1 [bit19:16] DLC[3:0]: Data Length Code

DLC[3:0]	Description
0-8	CAN + CAN FD: frame with 0-8 data bytes transmitted.
9-15	CAN: frame with 8 data bytes transmitted. CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted. See Table 3-1 for details.

E1 [bit15:0] TXTS[15:0]: Tx Timestamp

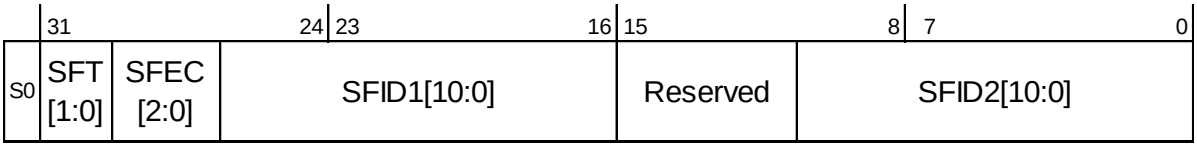
Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler (TSCC.TCP[3:0]).

6.5 Standard Message ID Filter Element

A Standard Message ID Filter Element consists of a single 32-bit word, and can be configured as a range filter, dual filter, classic bit mask filter, or filter for a single dedicated ID, for messages with 11-bit standard IDs.

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is

Filter List Standard Start Address (SIDFC.FLSSA[15:2]) + index of the filter element (0 to 127).



S0 [bit31:30] SFT[1:0]: Standard Filter Type

SFT[1:0]	Description
00	Range filter from SFID1[10:0] to SFID2[10:0] (SFID2[10:0] ≥ received ID ≥ SFID1[10:0]).
01	Dual ID filter for SFID1[10:0] or SFID2[10:0].
10	Classic filter: SFID1[10:0] = filter, SFID2[10:0] = mask. Only those bits of SFID1[10:0] where the corresponding SFID2[10:0] bits are "1" are relevant.
11	Reserved.

Note:

- *The setting SFT[1:0] = "11" is reserved and should not be used. The setting SFT[1:0] = "11" will disable the filter element, but use SFEC[2:0] = "000" instead.*

S0 [bit29:27] SFEC[2:0]: Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached.

If SFEC[2:0] = "100", "101", or "110" a match sets interrupt flag IR.HPM (High Priority Message) and, if enabled, an interrupt is generated. In this case register HPMS (High Priority Message Status) is updated with the status of the priority match.

SFEC[2:0]	Description
000	Disable filter element.
001	Store in Rx FIFO 0 if filter matches.
010	Store in Rx FIFO 1 if filter matches.
011	Reject ID if filter matches.
100	Set priority if filter matches.
101	Set priority and store in Rx FIFO 0 if filter matches.
110	Set priority and store in Rx FIFO 1 if filter matches.
111	Store into dedicated Rx Buffer or as debug message, configuration of SFT[1:0] ignored.

S0 [bit26:16] SFID1[10:0]: Standard Filter ID 1

This bit field has a different meaning depending on the configuration of SFEC[2:0]:

- SFEC[2:0] = "001" to "110"

  Set SFID1[10:0] according to the SFT[1:0] setting.

- SFEC[2:0] = "111"

  SFID1[10:0] defines the ID of a standard dedicated Rx Buffer or debug message to be stored. The received identifiers must match exactly, no masking mechanism is used.

S0 [bit15:11] Reserved: Reserved bits

When writing, always write "0". The read value is undefined.

S0 [bit10:0] SFID2[10:0]: Standard Filter ID 2

This bit field has a different meaning depending on the configuration of SFEC[2:0]:

- SFEC[2:0] = "001" to "110"

  Set SFID2[10:0] according to the SFT[1:0] setting

- SFEC[2:0] = "111"

  Filter for dedicated Rx Buffers or for debug messages

  SFID2[10:9] decides whether the received message is stored into a dedicated Rx Buffer or treated as message A, B, or C of the debug message sequence.

SFID2[10:9]	Description
00	Store message into a dedicated Rx Buffer.
01	Debug Message A.
10	Debug Message B.
11	Debug Message C.

  SFID2[8:6] are reserved bits. When writing, always write "0". The read value is undefined.

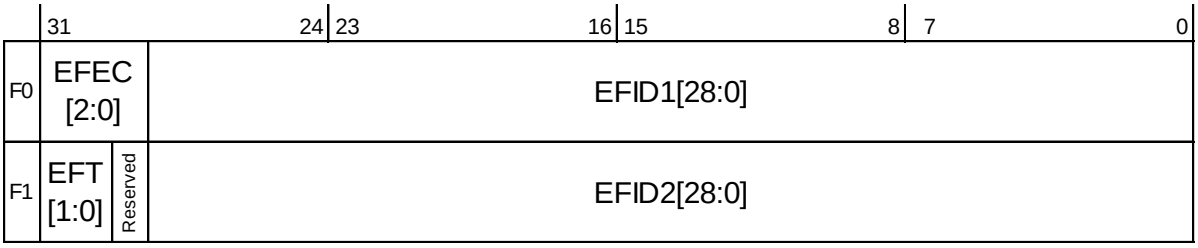
  SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA[15:2] for storage of a matching message.

6.6 Extended Message ID Filter Element

An Extended Message ID Filter Element consists of two 32-bit words, and can be configured as a range filter, dual filter, classic bit mask filter, or filter for a single dedicated ID, for messages with 29-bit extended IDs.

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is

Filter List Extended Start Address (XIDFC.FLESA[15:2]) + 2 × index of the filter element (0 to 63).



F0 [bit31:29] EFEC[2:0]: Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached.

If EFEC[2:0] = "100", "101", or "110" a match sets interrupt flag IR.HPM (High Priority Message) and, if enabled, an interrupt is generated. In this case register HPMS (High Priority Message Status) is updated with the status of the priority match.

EFEC[2:0]	Description
000	Disable filter element.
001	Store in Rx FIFO 0 if filter matches.
010	Store in Rx FIFO 1 if filter matches.
011	Reject ID if filter matches.
100	Set priority if filter matches.
101	Set priority and store in Rx FIFO 0 if filter matches.
110	Set priority and store in Rx FIFO 1 if filter matches.
111	Store into dedicated Rx Buffer or as debug message, configuration of EFT[1:0] ignored.

F0 [bit28:0] EFID1[28:0]: Extended Filter ID 1

This bit field has a different meaning depending on the configuration of EFEC[2:0].

- EFEC[2:0] = "001" to "110"

  Set EFID1[28:0] according to the EFT[1:0] setting.

- EFEC[2:0] = "111"

  EFID1[28:0] defines the ID of an extended dedicated Rx Buffer or debug message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see "3.4.1.3. Extended Message ID Filtering") is used.

F1 [bit31:30] EFT[1:0]: Extended Filter Type

EFT[1:0]	Description
00	Range filter from EFID1[28:0] to EFID2[28:0] (EFID2[28:0] ≥ received ID ANDed with XIDAM ≥ EFID1[28:0]).
01	Dual ID filter Matches when EFID1[28:0] or EFID2[28:0] is equal to received ID ANDed with XIDAM.
10	Classic filter: EFID1[28:0] = filter, EFID2[28:0] = mask. Only those bits of EFID1[28:0] where the corresponding EFID2[28:0] bits are "1" are relevant. Matches when the received ID ANDed with XIDAM is equal to EFID1[28:0] masked by EFID2[28:0].
11	Range filter from EFID1[28:0] to EFID2[28:0] (EFID2[28:0] ≥ EFID1[28:0]), XIDAM mask not applied.

F1 [bit29] Reserved: Reserved bit

When writing, always write "0". The read value is undefined.

F1 [bit28:0] EFID2[28:0]: Extended Filter ID 2

This bit field has a different meaning depending on the configuration of EFEC[2:0]:

- EFEC[2:0] = "001" to "110"

  Set EFID2[28:0] according to the EFT[1:0] setting

- EFEC[2:0] = "111"

  EFID2[28:0] is used to configure this filter for dedicated Rx Buffers or for debug messages

  EFID2[28:11] are reserved bits. When writing, always write "0". The read value is undefined.

  EFID2[10:9] decides whether the received message is stored into a dedicated Rx Buffer or treated as message A, B, or C of the debug message sequence.

EFID2[10:9]	Description
00	Store message into a dedicated Rx Buffer.
01	Debug Message A.
10	Debug Message B.
11	Debug Message C.

  EFID2[8:6] are reserved bits. When writing, always write "0". The read value is undefined.

  EFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA[15:2] for storage of a matching message.

CHAPTER 5-4: CAN FD Message RAM ECC Function

This chapter explains the functions and operations of the CAN FD Message RAM ECC Function.

1. Overview
2. Configuration
3. Interrupts
4. Explanation of Operation
5. Setup procedure examples
6. Registers
7. Precautions

This chapter explains only the block of the CAN FD Message RAM ECC Function.

For the functions and operations of the CAN FD Controller, see CAN FD Controller External Specification.

CODE: FIP018ECC-E01.3

1. Overview

In the CAN FD Message RAM, CAN message objects are stored. The ECC Function of the CAN FD Message RAM enables detection and correction of data errors in the Message RAM.

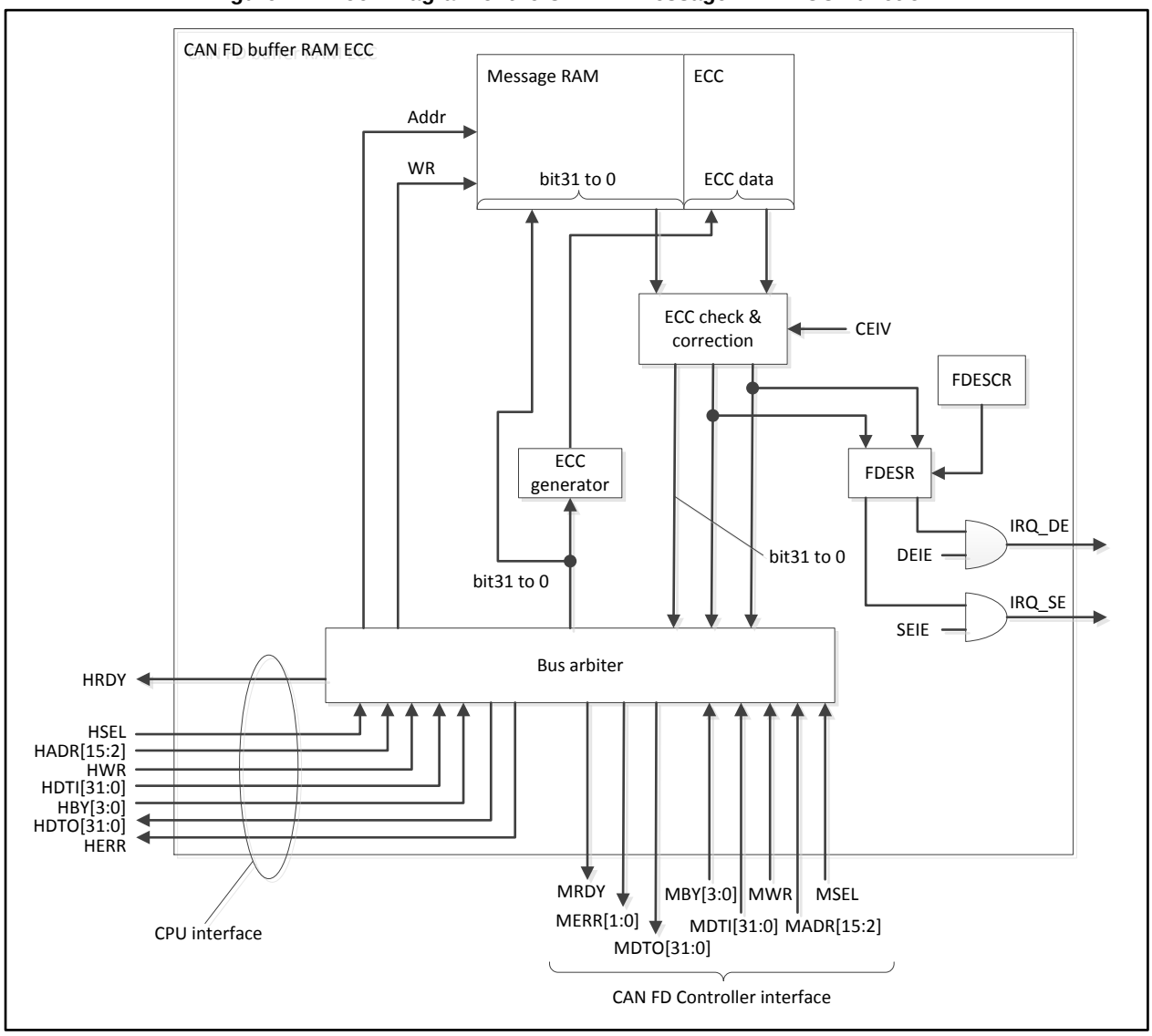
The CAN FD Message RAM ECC Function has the following features.

- 2-bit error detection and 1-bit error correction for 32-bit data in the Message RAM
- Stopping of the CAN FD Controller upon error detection

2. Configuration

Figure 2-1 is a block diagram of the CAN FD Message RAM ECC Function.

Figure 2-1 Block Diagram of the CAN FD Message RAM ECC Function



- Message RAM
  Consists of 32-bit RAM, in which Tx/Rx messages is stored.
- ECC
  Consists of 7 bits, in which the ECC data for bits 31 to 0 of the Message RAM is stored.
- ECC generator
  Generates ECC data from data to be written in the Message RAM.
- ECC check & correction
  Combines Message RAM data and ECC data together to check for a single- or double-bit error. For a single-bit error, this circuit corrects the data.
- Bus arbiter
  Arbitrates access by the CPU and CAN FD Controller, and reads data from or writes data to the Message RAM.
- FDESCR
  CAN FD ECC error status clear register
- FDESR
  CAN FD ECC error status register

3. Interrupts

This section explains the interrupts of the CAN FD Message RAM ECC Function.

Interrupts of the CAN FD Message RAM ECC Function

The Message RAM ECC Function supports the following interrupts.

Table 3-1 Interrupt Controller Bits and Interrupt Factors of the CAN FD Message RAM ECC Function

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Double-bit error interrupt	DEI	FDESR	2-bit data error	FDECR.DEIE	Write "1" to the double-bit error clear bit (FDESCR.DEIC).
Single-bit error interrupt	SEI	FDESR	1-bit data error	FDECR.SEIE	Write "1" to the single-bit error clear bit (FDESCR.SEIC).

4. Explanation of Operation

This section explains the operations and functions of the CAN FD Message RAM ECC.

This section explains the following functions.

- Message RAM ECC generation
- Message RAM ECC check

4.1. Message RAM ECC Generation

This section explains ECC generation of the Message RAM.

ECC generation

The CAN FD Message RAM is made up of 32 bits and an additional 7 bits for ECC. When data is written to the CAN FD Message RAM, ECC data is generated from the write data based on the syndrome, and is written together with the write data.

4.2. Message RAM ECC Check

This section explains the ECC check of the Message RAM.

ECC check

When data is read from the Message RAM, the ECC data is also read. In the ECC check, calculation is performed based on the syndrome to determine whether the data in the Message RAM is correct. If it is determined from the check result that the data in the Message RAM, the following operation is performed.

- In the case where only 1 bit in the data is incorrect (single-bit error):
  The single-bit error occurrence bit (FDESR.SEI) is set and the RAM address is set in the CAN FD ECC single-bit error address register (FDSEAR). The incorrect 1-bit data is corrected and then is read as normal data.
- In the case where 2 bits in the data are incorrect (double-bit error):
  The double-bit error occurrence bit (FDESR.DEI) is set and the RAM address is set in the CAN FD ECC double-bit error address register (FDDEAR). The incorrect data is not corrected and then is read as it is.

If the interrupt enable bit corresponding to each error bit (FDESR.SEI or FDESR.DEI) has been set to "1", an interrupt occurs when the error bit is set to "1".

Stopping of the CAN FD Controller upon error detection

If the CAN FD Controller detects a double-bit error when it reads the Message RAM, it sets the INIT bit in its CC control register (CCCR). Then, the CAN FD Controller stops.

Error response to CPU upon error detection

When error response is enabled (FDECR.CEREN = "1"), if the CPU reads the Message RAM and a double-bit error is detected at this time, an error response is returned to the CPU.

5. Setup procedure examples

This section provides examples of setup procedures for the CAN FD Message RAM ECC Function.

Setup procedures for the CAN FD Message RAM ECC Function

This section explains a setting procedure example for the CAN FD Message RAM ECC Function. For details on the setting of the CAN FD Controller, see the chapter on the CAN FD Controller.

1. Set "1" in the INIT bit and CCE bit in the CC control register (CCCR) of the CAN FD Controller.
   This enables write operation to the control bit of registers.
2. Set the CAN FD Controller registers.
   Make the settings of the CAN FD Controller according to the use conditions.
3. Initialize the Message RAM.
   Initializing the RAM, which is in an undefined state, prevents an ECC error from being detected by mistake.
4. Set "1" to the SEIC bit and DEIC bit in the CAN FD ECC error status clear register (FDESCR).
   This clears the SEI bit and DEI bit in the CAN FD ECC error status register (FDESR).
5. Set each bit in the CAN FD ECC error control register.
   Make the settings of the CAN FD ECC Function according to the use conditions.

Performing the above setting procedure enables the use of the CAN FD Message RAM ECC Function. Since the INIT bit in the CC control register (CCCR) is "1", the CAN FD Controller is in the operation stop state. To enable operation of the CAN FD Controller, set "0" in the INIT bit in the CC control register (CCCR) after performing the above procedure.

6. Registers

This section explains the registers of the CAN FD Message RAM ECC Function.

List of the registers of the Message RAM ECC Function

Table 6-1 List of the Registers of the CAN FD Message RAM ECC Function

Abbreviated Register Name	Register Name	Reference
FDECR	CAN FD ECC error control register	6.1
FDESR	CAN FD ECC error status register	6.2
FDESCR	CAN FD ECC error status clear register	6.3
FDDEAR	CAN FD ECC double-bit error address register	6.4
FDSEAR	CAN FD ECC single-bit error address register	6.5

6.1. CAN FD ECC Error Control Register (FDECR)

The CAN FD ECC error control register (FDECR) is used to set whether to enable the interrupt when single-bit error correction or double-bit error detection occurs during the ECC check. It is also used to set ECC error detection stop and response to the CPU.

CAN FD ECC error control register (FDECR)

bit	7	6	5	4	3	2	1	0
Field	Reserved				CEIV	CEREN	DEIE	SEIE
Attribute	-				R/W	R/W	R/W	R/W
Initial value	0000				0	0	0	0

[bit7:4] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit3] CEIV: ECC check disable bit

bit	Description
0	Enable ECC check. [Initial value]
1	Disable ECC check.

- This bit can be changed when the CCE bit in the CC control register (CCCR) of the CAN FD Controller is set "1".
- When CEIV bit is "0", the ECC check is enabled.
- When CEIV bit is "1", ECC check is disabled.

[bit2] CEREN: ECC error response enable bit

bit	Description
0	Disable error response for double-bit error detection. [Initial value]
1	Enable error response for double-bit error detection.

- This bit can be changed when the CCE bit in the CC control register (CCCR) of the CAN FD Controller is set "1".
- When CEREN bit is "0", even if a double-bit error is detected in the read operation of the Message RAM by the CPU, an error response is not made for this setting.
- When CEREN bit is "1", if a double-bit error is detected in the read operation of the Message RAM by the CPU, an error response is made for this setting.

Note:

- *The CEREN bit does not apply to the single-bit error.*

[bit1] DEIE: Double-bit error factor interrupt enable bit

bit	Description
0	Disable the interrupt caused by the double-bit error (FDESR.DEI). [Initial value]
1	Enable the interrupt caused by the double-bit error (FDESR.DEI).

- When DEIE bit is "0", even if a double-bit error is detected (FDESR.DEI = "1"), the IRQ_DE signal remains "L".
- When DEIE bit is "1", if a double-bit error is detected (FDESR.DEI = "1"), the IRQ_DE signal becomes "H" to request an interrupt.

[bit0] SEIE: Single-bit error factor interrupt enable bit

bit	Description
0	Disable the interrupt caused by the single-bit error (FDESR.SEI). [Initial value]
1	Enable the interrupt caused by the single-bit error (FDESR.SEI).

- When SEIE bit is "0", even if a single-bit error is detected (FDESR.SEI = "1"), the IRQ_SE signal remains "L".
- When SEIE bit is "1", if a single-bit error is detected (FDESR.SEI = "1"), the IRQ_SE signal becomes "H" to request an interrupt.

6.2. CAN FD ECC Error Status Register (FDESR)

The CAN FD ECC error status register (FDESR) displays whether a single-bit error has been corrected in the ECC check and whether a double-bit error has been detected. When any bit in this register becomes "1", it remains "1" unless it is cleared by using CAN FD ECC error status clear register (FDESCR).

CAN FD ECC error status register (FDESR)

bit	7	6	5	4	3	2	1	0
Field	Reserved						DEI	SEI
Attribute	-						R	R
Initial value	000000						0	0

[bit7:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1] DEI: Double-bit error occurrence bit

bit	Description
0	Indicate that a double-bit error has not been detected. [Initial value]
1	Indicate that a double-bit error has been detected.

- Any value written in the DEI bit is ignored.
- When the DEI bit is "1", to clear this bit to "0", write "1" to the DEIC bit in the CAN FD ECC error status clear register (FDESCR).

Notes:

- *When the DEI bit is "1", the CAN FD ECC double-bit error address register (FDDEAR) behaves as follows: If a double-bit error is detected in the Message RAM read operation at an address other than the one retained in this register, the register is not updated, and retains the previous address value.*
- *Even if the DEI bit is "1", a single-bit error can be detected.*
- *If a double-bit error is detected when the CAN FD Controller or CPU reads data from the Message RAM, this bit is set "1".*

[bit0] SEI: Single-bit error occurrence bit

bit	Description
0	Indicate that a single-bit error has not been detected. [Initial value]
1	Indicate that a single-bit error has been detected.

- Any value written in the SEI bit is ignored.
- When the SEI bit is "1", to clear this bit to "0", write "1" to the SEIC bit in the CAN FD ECC error status clear register (FDESCR).

Notes:

- *When the SEI bit is "1", the CAN FD ECC single-bit error address register (FDSEAR) behaves as follows: if a single-bit error is detected in the Message RAM read operation at an address other than the one retained in this register, the register is not updated, and retains the previous address value.*
- *Even if the SEI bit is "1", a double-bit error can be detected.*
- *If a single-bit error is detected when the CAN FD Controller or CPU reads data from the Message RAM, this bit is set "1".*

6.3. CAN FD ECC Error Status Clear Register (FDESCR)

The CAN FD ECC error status clear register (FDESCR) is used to clear bits in the CAN FD ECC error status register.

CAN FD ECC error status clear register (FDESCR)

bit	7	6	5	4	3	2	1	0
Field	Reserved						DEIC	SEIC
Attribute	-						W	W
Initial value	000000						0	0

[bit7:2] Reserved: Reserved bits

When writing, always write "0". When reading, "0" is always read.

[bit1] DEIC: Double-bit error clear bit

bit	Description
	During Write Operation
0	No effect
1	Change the double-bit error occurrence bit (FDESR.DEI) to "0".

"0" is always read.

[bit0] SEIC: Single-bit error clear bit

bit	Description
	During Write Operation
0	No effect
1	Change the single-bit error occurrence bit (FDESR.SEI) to "0".

"0" is always read.

6.4. CAN FD ECC Double-bit Error Address Register (FDDEAR)

The CAN FD ECC double-bit error address register (FDDEAR) indicates the address which a double-bit error has occurred at the Message RAM during the ECC check. This register is valid when the DEI bit in the CAN FD ECC error status register (FDESR) is "1". While the DEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.

CAN FD ECC double-bit error address register (FDDEAR)

bit	15	14	13	12	11	10	9	8
Field	DRA15	DRA14	DRA13	DRA12	DRA11	DRA10	DRA9	DRA8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	DRA7	DRA6	DRA5	DRA4	DRA3	DRA2	DRA1*	DRA0*
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

* : DRA1 and DRA0 are fixed to "0".

[bit15:0] DRA15 to DRA0: Double-bit error Message RAM address bits

bit15:0	Description
0x0000	Indicates that a double-bit error has occurred at Message RAM address 0.
0x0004	Indicates that a double-bit error has occurred at Message RAM address 4.
0x0008	Indicates that a double-bit error has occurred at Message RAM address 8.
...	.....

- Writing access to DRA15 to DRA0 bits is ignored.
- DRA15 to DRA0 bits are valid when the DEI bit in the CAN FD ECC error status register (FDESR) is "1".
- When the DEI bit in the CAN FD ECC error status register (FDESR) changes from "0" to "1", the Message RAM address is set to the CAN FD ECC double-bit error address register (FDDEAR). Then, this value is retained as long as the DEI bit in the CAN FD ECC error status register (FDESR) is "1". Therefore, even if a double-bit error is detected more than once, this register retains the address value for the first detected error until the DEI bit in the CAN FD ECC error status register (FDESR) is cleared to "0".

6.5. CAN FD ECC Single-bit Error Address Register (FDSEAR)

The CAN FD ECC single-bit error address register (FDSEAR) indicates the address which a single-bit error has occurred at the Message RAM during the ECC check. This register is valid when the SEI bit in the CAN FD ECC error status register (FDESR) is "1". While the SEI bit in the CAN FD ECC error status register (FDESR) is "1", the value of this register is retained.

CAN FD ECC single-bit error address register (FDSEAR)

bit	15	14	13	12	11	10	9	8
Field	SRA15	SRA14	SRA13	SRA12	SRA11	SRA10	SRA9	SRA8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SRA7	SRA6	SRA5	SRA4	SRA3	SRA2	SRA1*	SRA0*
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

*: SRA1 and SRA0 are fixed to "0".

[bit15:0] SRA15 to SRA0: Single-bit error Message RAM address bits

bit15:0	Description
0x0000	Indicates that a single-bit error has occurred at Message RAM address 0.
0x0004	Indicates that a single-bit error has occurred at Message RAM address 4.
0x0008	Indicates that a single-bit error has occurred at Message RAM address 8.
...	.....

- Writing access to SRA15 to SRA0 bits is ignored.
- SRA15 to SRA0 bits are valid when the SEI bit in the CAN FD ECC error status register (FDESR) is "1".
- When the SEI bit in the CAN FD ECC error status register (FDESR) changes from "0" to "1", the Message RAM address is set to the CAN FD ECC single-bit error address register (FDSEAR). Then, this value is retained as long as the SEI bit in the CAN FD ECC error status register (FDESR) is "1". Therefore, even if a single-bit error is detected more than once, this register retains the address value for the first detected error until the SEI bit in the CAN FD ECC error status register (FDESR) is cleared to "0".

7. Precautions

This section explains precautions on the use of the CAN FD Message RAM ECC.

Access to the Message RAM

- To access the Message RAM, perform 32-bit access. Using another number of bits as the unit of access may cause an ECC error or the writing of wrong data to the Message RAM because the data in the Message RAM is tested as 32-bit long data.

ECC check

- The data in the Message RAM is undefined after power-on. Therefore, a double- or single-bit error may occur if the Message RAM is read before data is written to the Message RAM area to be used. After power-on, write data to the Message RAM area to be used before reading the Message RAM.
- The following bits can be changed when the configuration change enable bit in the CC control register of the CAN FD Controller (CCCR.CCE bit) is "1".
  - ECC error response enable bit (CEREN) in the CAN FD ECC error control register (FDECR)
  - ECC check disable bit (CEIV) in the CAN FD ECC error control register (FDECR)
- If an ECC error is detected in read access to the Message RAM from the CAN FD Controller, the following bits are set: the error flags regarding the ECC of the CAN FD Controller (IR.BEU and IR.BEC) and the error flags regarding the ECC of the Message RAM (FDESR.DEI and FDESR.SEI).
- If an ECC error is detected in read access to the Message RAM from the CPU, the error flags regarding the ECC of the Message RAM (FDESR.DEI and FDESR.SEI) are set.

CHAPTER 5-5: EXTERNAL TIME STAMP COUNTER FOR CAN FD

This chapter explains the functions and operations of the External Time Stamp Counter for CAN FD.

---

1. Overview
2. Configuration
3. Operations
4. Example of the Operation
5. Register

CODE: FIP023-E01.1

1. Overview

The Time Stamp Counter for CAN FD is the 16-bit counter of external time stamp function.

Time stamp counter

- This counter is a 16-bit counter, and only counts up.
- If the counter is enable, it counts up from the current value.
- If the counter is disable, it holds current value.
- Setting enable/disable the counter is programmable by software.
- The counter counts up to the comparison value.
- When counter value comes to greater than or equal to the comparison value, the counter loads 0 by the next bus clock cycle and continues counting up.
- The limit value of the counter can be programmable by software.
- Time Stamp Counter Data Register (TSCDTR) indicates the count value.

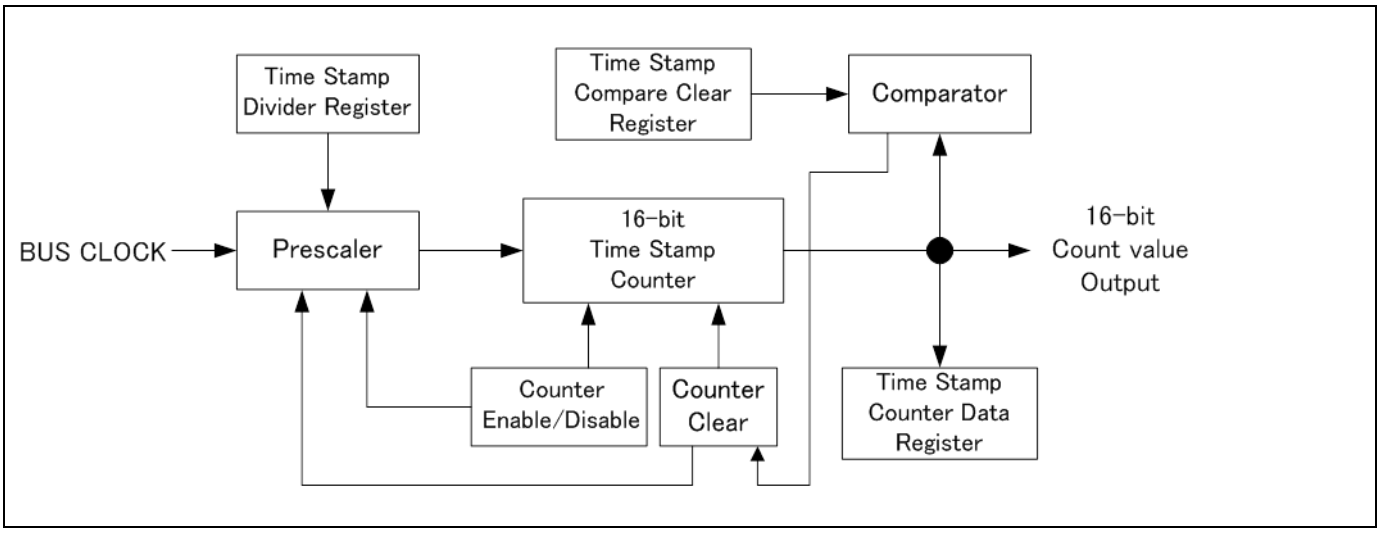
Count up cycle of the time stamp counter

- Enable to select a count up cycle (divided from bus clock ; 1 - 65536 division)
- Enable to set the division ratio any value (i.e. each step)
- Enable to set a count up cycle by software.

2. Configuration

This section explains configuration of the time stamp counter for CAN FD.

Figure 2 1 Block Diagram of Time Stamp Counter for CAN FD



3. Operations

3.1. Operation of Time Stamp Counter for CAN FD

This section explains the operation of Time Stamp Counter for CAN FD.

3.1.1. Count up cycle for Time Stamp Counter

Count up cycle for the Time Stamp Counter can be set for each step between 1 to 65536. Count up cycle is set by the Time Stamp Divider register (TSDIVR).

3.1.2. Time Stamp Counter

After setting the count enable (TSCNTR:CNTEN=1), the count up is started for the current counter value. After setting the counter disable (TSCNTR:CNTEN=0), the count up is stopped and the count value is held.

The counter counts up to the value that is set in the Time Stamp Compare Clear Register (TSCPCLR). When counter value comes to greater than or equal to the value that is set in the Time Stamp Compare Clear Register (TSCPCLR), the counter loads 0 by the next bus clock cycle and continues counting up.

The count value can be seen by reading Time Stamp Counter Data Register (TSCDTR).

Figure 3-1 Operation to enable counting (TSDIVR:CDIV15-CDIV0=0x0003)

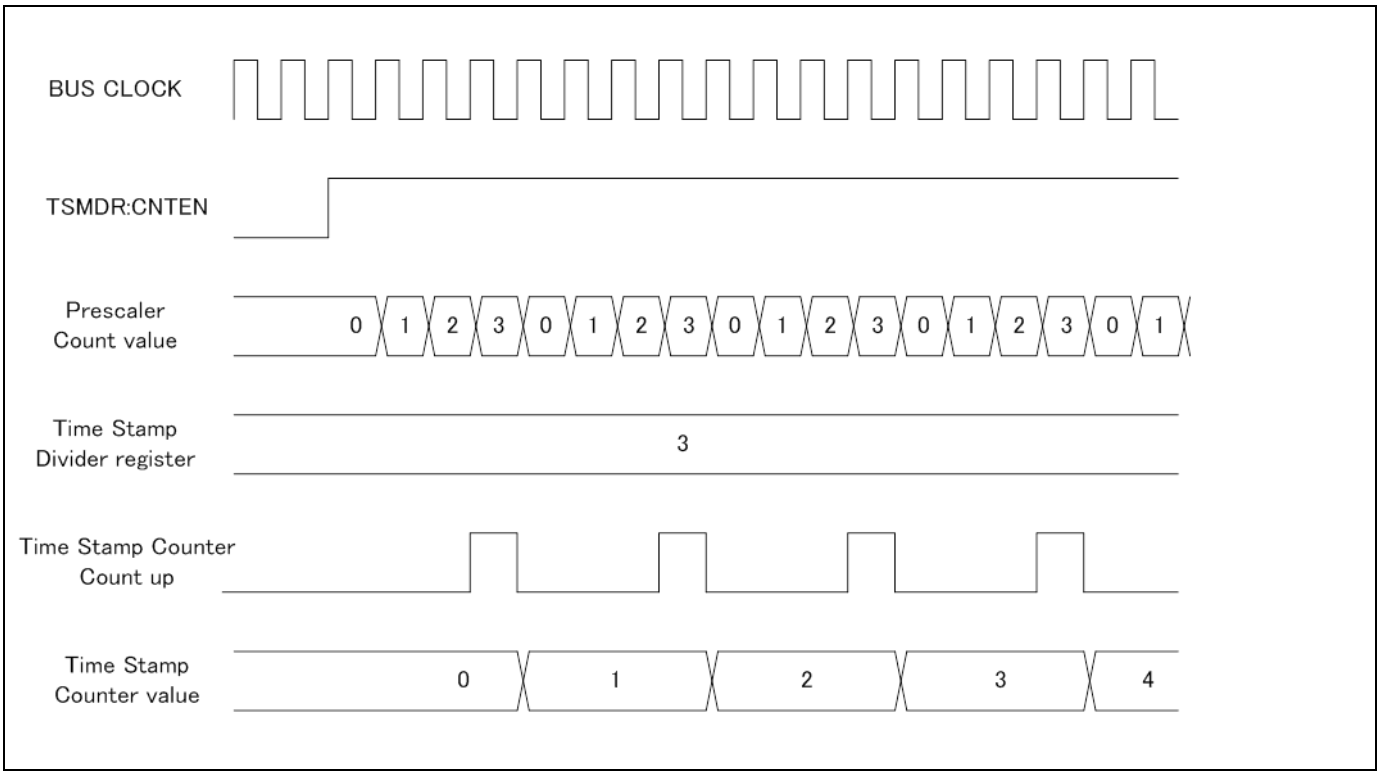


Figure 3-2 Operation to disable counting (TSDIVR:CDIV15-CDIV0=0x0003)

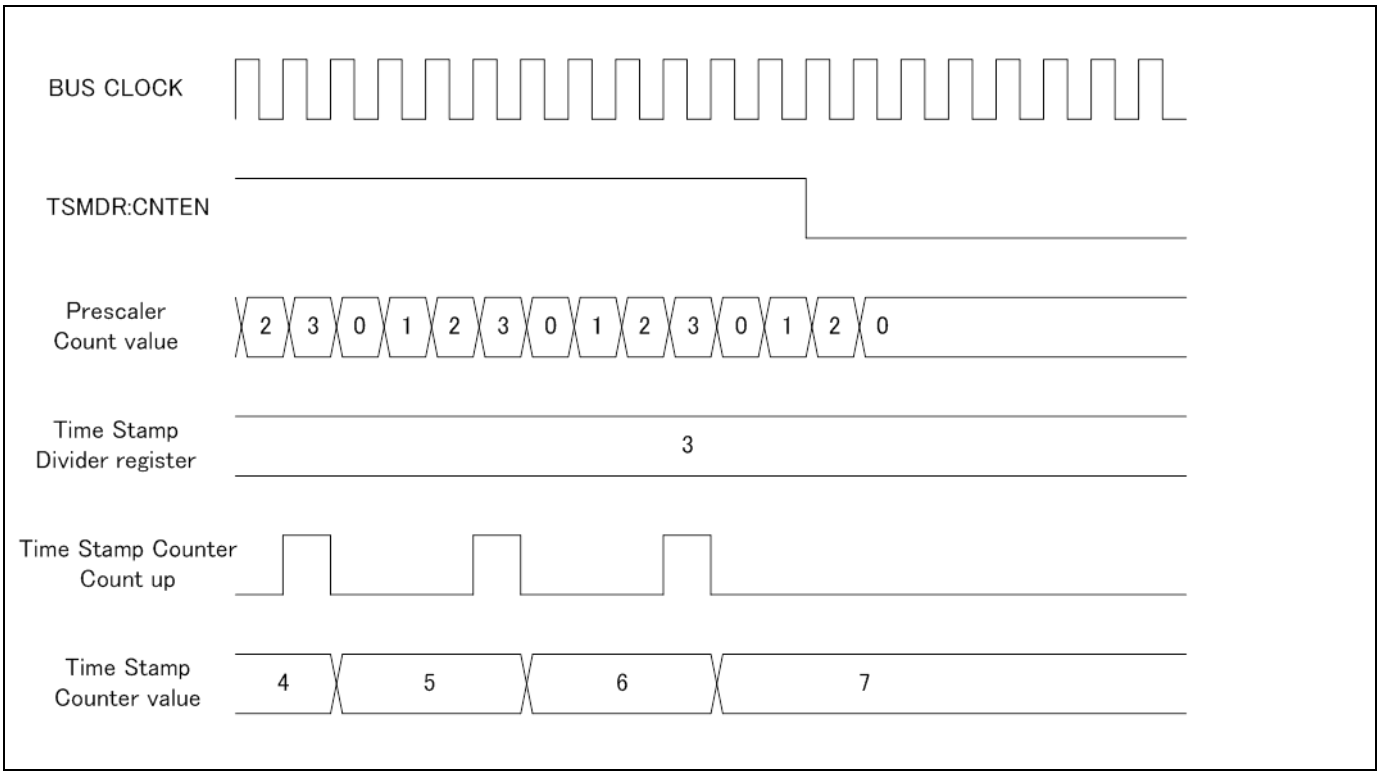
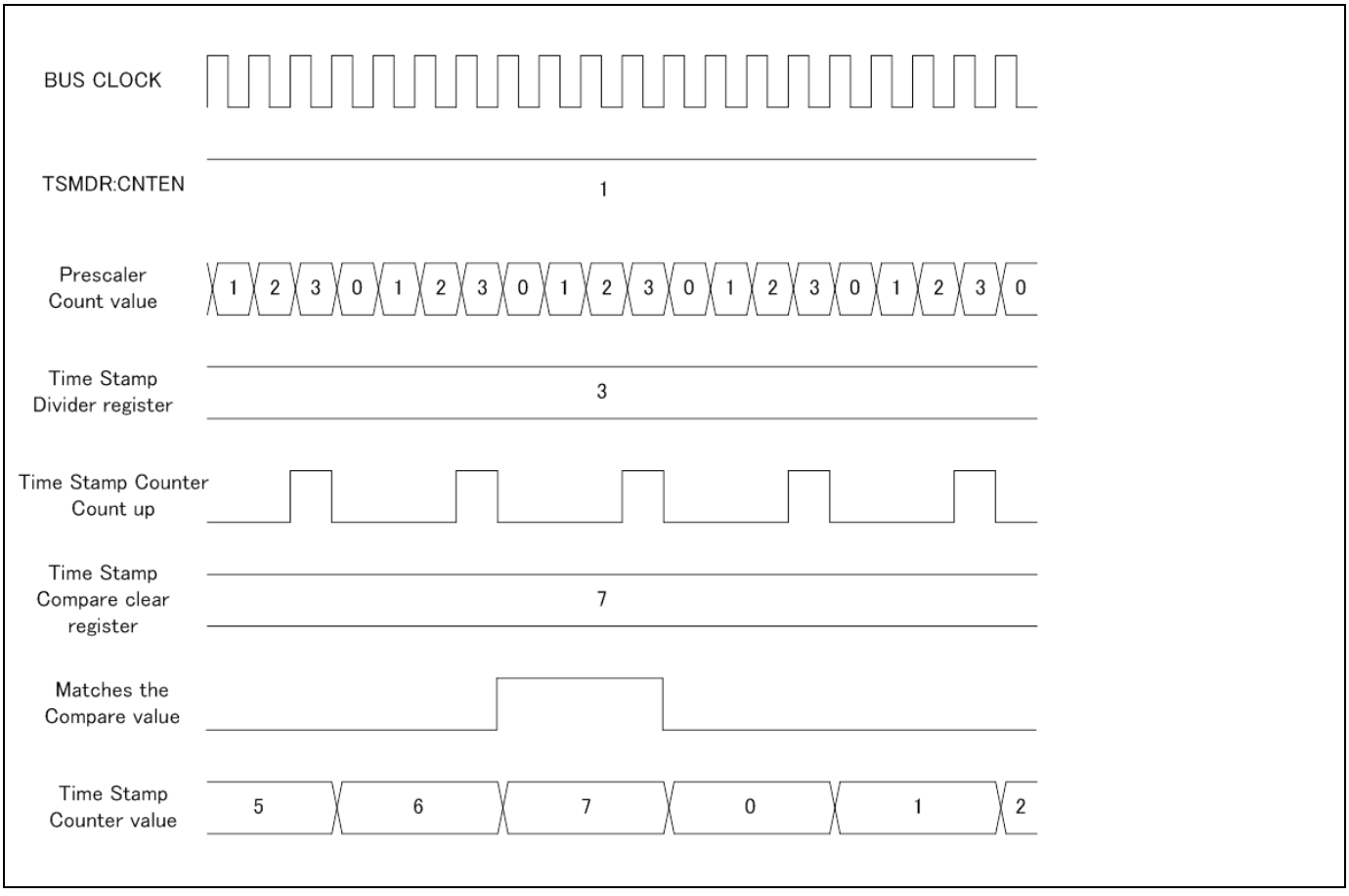


Figure 3-3 Operation to clear the Time Stamp Counter by matching TSCPCLR (TSDIVR:CDIV15-CDIV0=0x0003, TSCPCLR:CMP15-CMP0=0x0007)



3.1.3. Clear Request

By setting the Counter Clear Request of the Time Stamp Control register (TSCNTR), the counters in the macro (Timestamp Counter, Prescaler) are cleared. Time Stamp Mode register (TSMDR), Time Stamp Divider register (TSDIVR), Time Stamp Compare Clear register (TSCPCLR) are not cleared by this request.

When the Counter Clear Request is set with the counter enabled, Time Stamp Counter and Prescaler are cleared, and the counter keeps counting up.

When the Counter Clear Request is set with the counter disabled, Time Stamp Counter and Prescaler are cleared.

Figure 3-4 Setting Counter Clear Request with the counter enabled (TSDIVR:CDIV15-CDIV0=0x0003, TSCPCLR:CMP15-CMP0=0x0007)

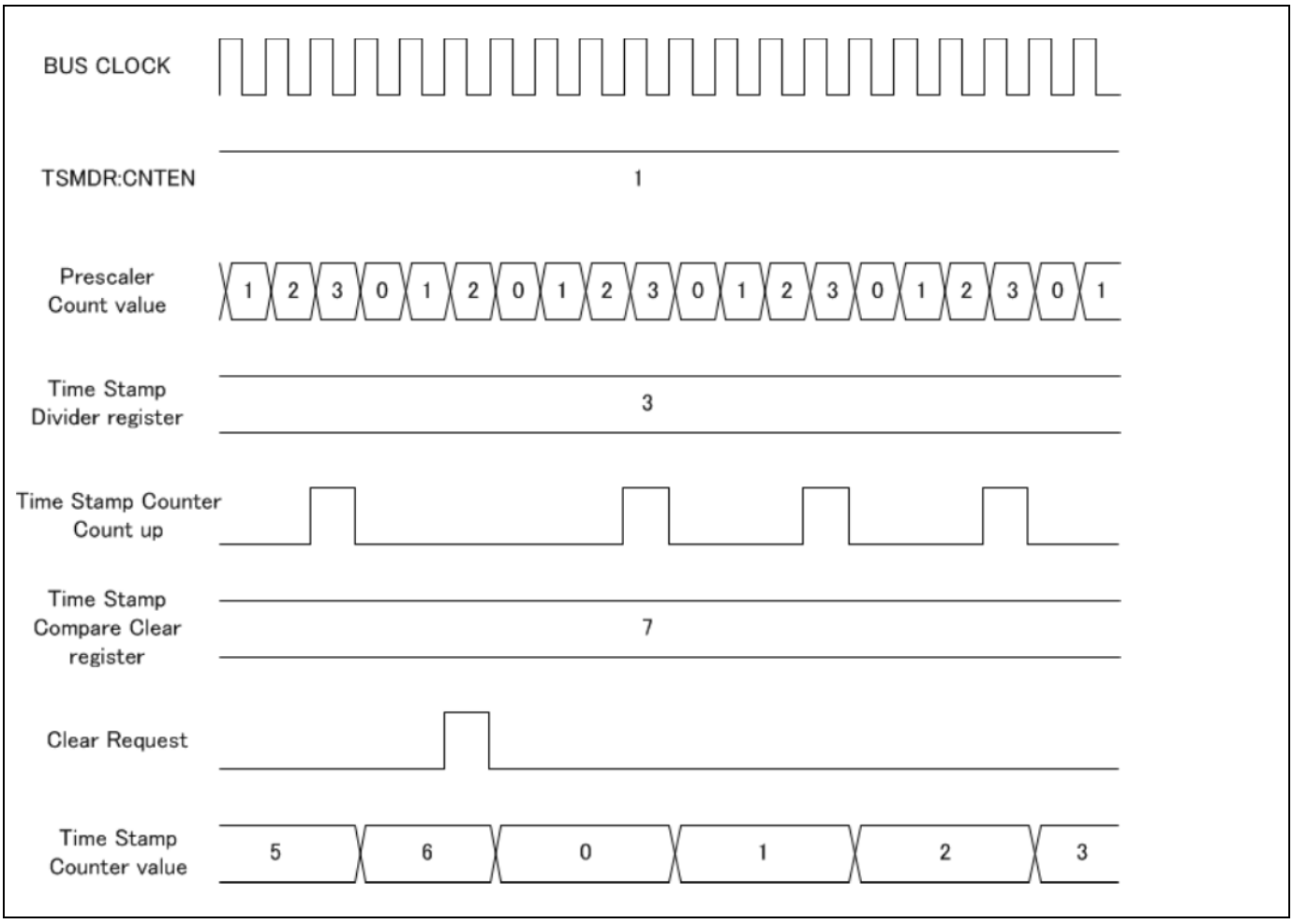
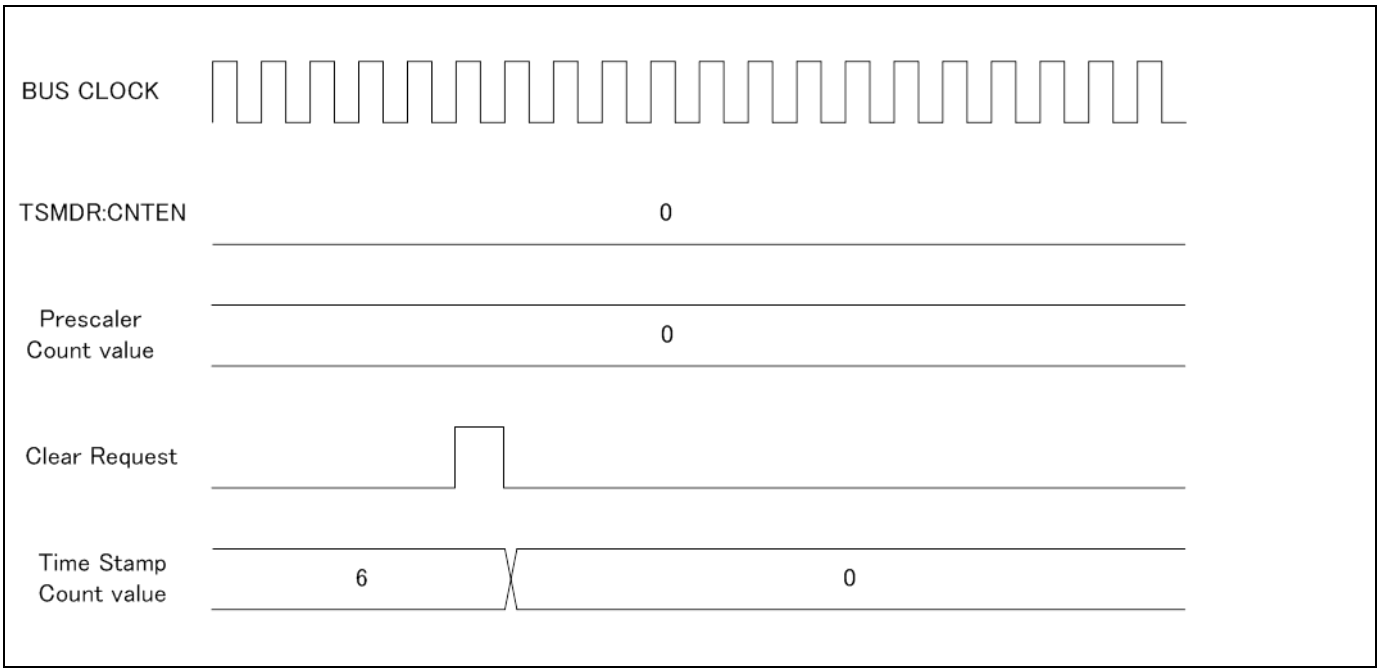


Figure 3-5 Setting Counter Clear Request with the counter disabled

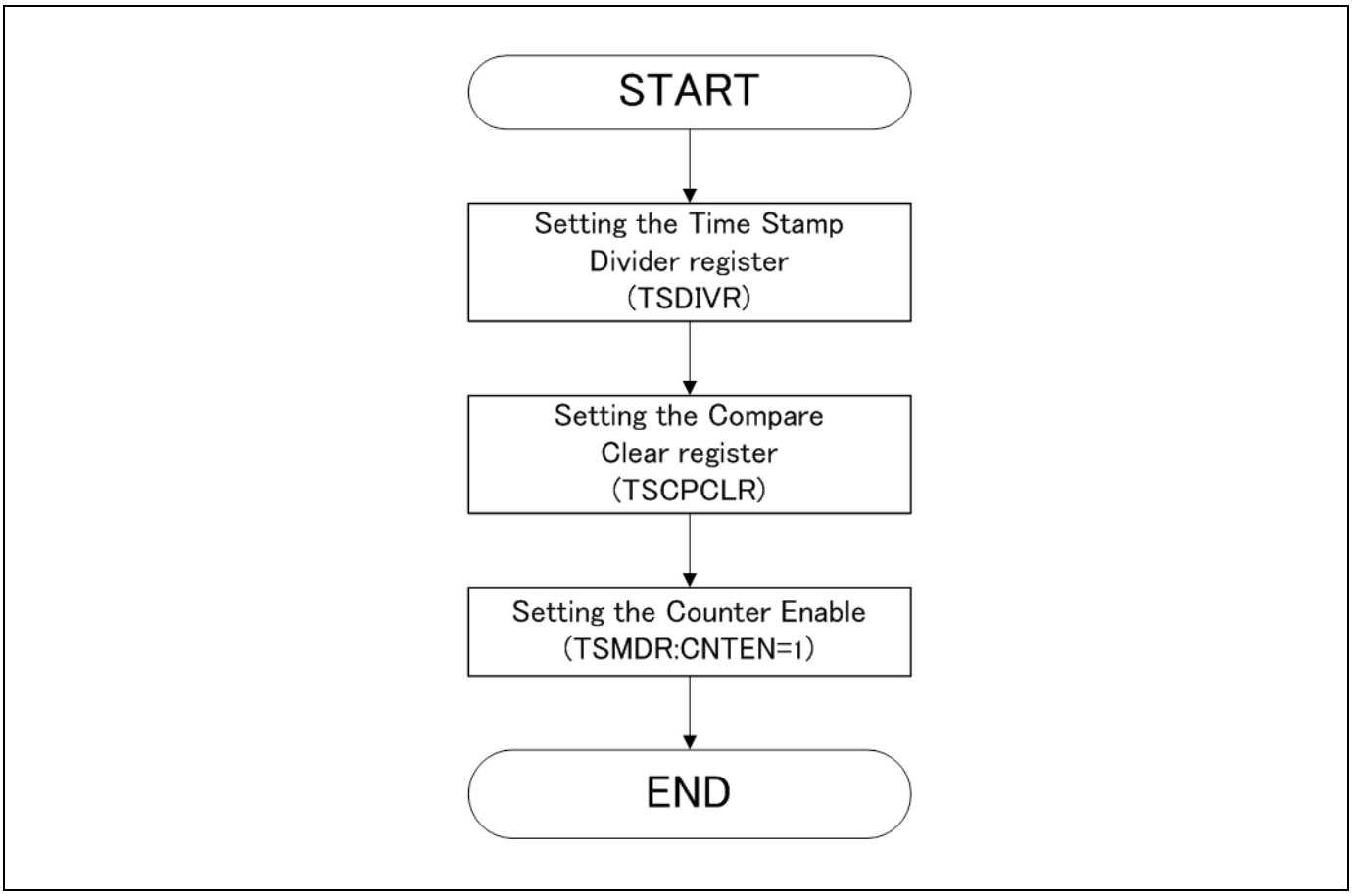


4. Example of the Operation

This section explains the example of operating Time Stamp Counter for CAN FD.

The procedure for setting Time Stamp Counter for CAN FD is shown in Figure 4-1.

Figure 4-1 The setting procedure for the Time Stamp Counter for CAN FD.



5. Register

This section explains the configuration and functions of the registers used for the CAN FD Time Stamp Counter.

- List of CAN FD Time Stamp Counter registers

Table 5-1 List of CAN FD Time Stamp Counter registers

Abbreviation	Register name	See
TSCNTR	Time Stamp Control Register	5.1
TSMDR	Time Stamp Mode Register	5.2
TSDIVR	Time Stamp Divider Register	5.3
TSCDTR	Time Stamp Counter Data Register	5.4
TSCPCLR	Time Stamp Compare Clear Register	5.5

5.1. Time Stamp Control Register (TSCNTR)

The Time Stamp Control Register (TSCNTR) is used to clear the counter in the macro.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							CCLR
Attribute	-							W
Initial value	0000000							0

[bit15:1] Reserved: Reserved bits
Always write "0" to these bits. The read value is "0".

[bit0] CCLR : Counter clear bit
The read value is "0".

bit	Description
	Write
0	No effect
1	Clear the counter(time stamp counter, prescaler).

Notes:

- *Time Stamp Counter Mode Register, Time Stamp Divider Register and Time Stamp Compare Clear Register are not cleared.*

5.2. Time Stamp Mode Register (TSMDR)

The Time Stamp Mode Register (TSMDR) is used to enable/disable the counter.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							CNTEN
Attribute	-							R/W
Initial value	0000000							0

[bit15:1] Reserved: Reserved bits

Always write "0" to these bits. The read value is "0".

[bit0] CNTEN : Counter enable bit

bit	Description
0	Count disabled Stop counting up and keep the counter value.
1	Count enabled Start counting up from the current value.

5.3. Time Stamp Divider Register (TSDIVR)

The Time Stamp Divider Register (TSDIVR) is used to set the clock division ratio for the counter.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	15	14	13	12	11	10	9	8
Field	CDIV15	CDIV14	CDIV13	CDIV12	CDIV11	CDIV10	CDIV9	CDIV8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CDIV7	CDIV6	CDIV5	CDIV4	CDIV3	CDIV2	CDIV1	CDIV0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:16] Reserved: Reserved bits
Always write "0" to these bits. The read value is "0".

[bit15:0] CDIV15 to CDIV0 : Counter clock division ratio setting bit.
These bits set the counter clock division ratio.

The division ratio corresponds to the value added 1 to CDIV15-CDIV0.

Notes:

- *If counter is enabled (TSMDR:CNTEN=1), operating this bit is prohibited*
- *This register must be accessed in 16-bit mode.*

5.4. Time Stamp Counter Data Register (TSCDTR)

The Time Stamp Counter Data Register (TSCDTR) indicates the count value of the Time Stamp Counter.

bit	15	14	13	12	11	10	9	8
Field	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit15:0] CNT15 to CNT0 : Counter value bits

The counter value of the Time Stamp Counter is indicated.

These bits are cleared to "0" by setting Counter Clear bit to "1" (TSCNTR:CCLR=1).

Note:

- *This register must be accessed in 16-bit mode.*

5.5. Time Stamp Compare Clear Register (TSCPCLR)

The Time Stamp Compare Clear Register (TSCPCLR) is used to set the comparison value for clearing the counter.

bit	15	14	13	12	11	10	9	8
Field	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP9	CMP8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	CMP7	CMP6	CMP5	CMP4	CMP3	CMP2	CMP1	CMP0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:0] CMP15 to CMP0 : Compare clear setting bits

These bits are compared with the counter value, and the counter is cleared when the counter value is greater than or equal to these bits.

If these bits are "0x0000", the Time Stamp Counter is always "0".

Notes:

- *If counter is enabled (TSMDR:CNTEN=1), operating this bit is prohibited.*
- *This register must be accessed in 16-bit mode.*

CHAPTER 6-1: HDMI-CEC/Remote Control Reception

HDMI-CEC/remote control reception is explained as follows.

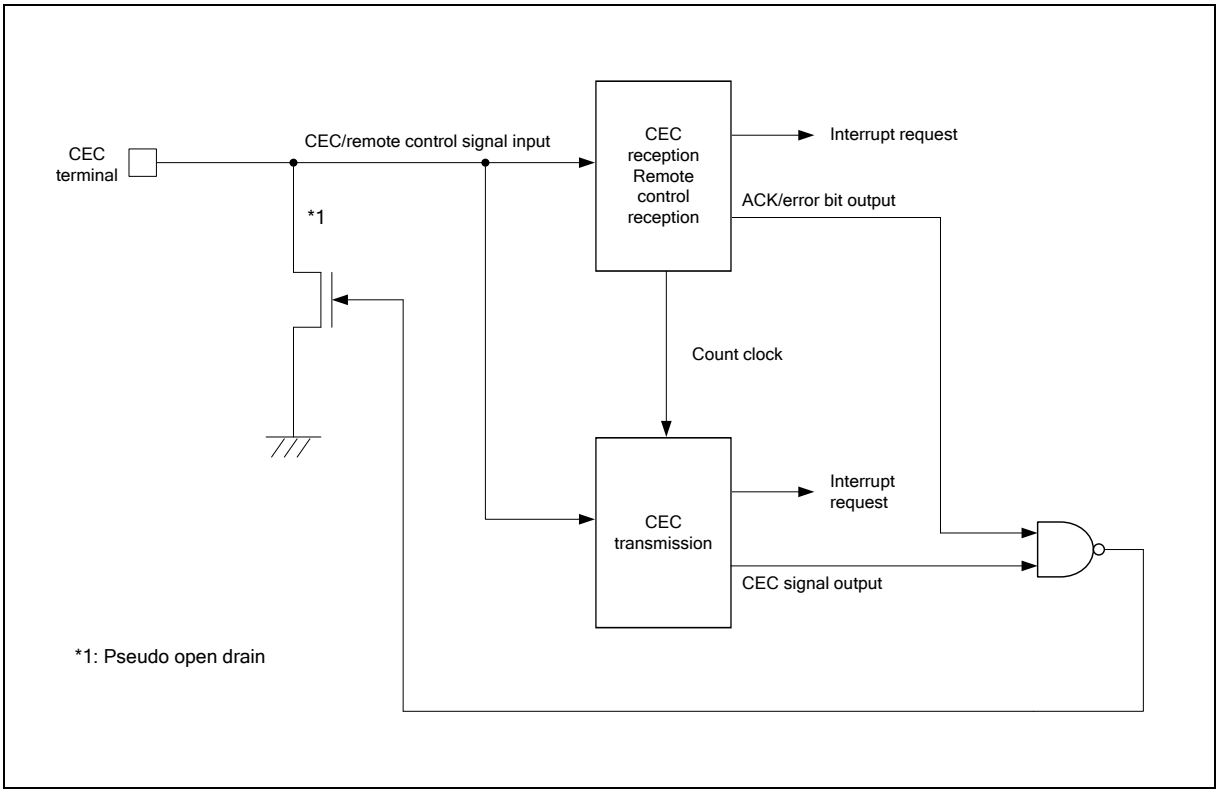
1. Configuration

1. Configuration

Configuration of HDMI-CEC/remote control reception is as follows.

Configuration

Figure 1-1 Configuration of HDMI-CEC/Remote Control Reception



■ CEC Reception/Remote Control Reception

See a separate chapter "CEC Reception/Remote Control Reception".

■ CEC Transmission

See a separate chapter "CEC Transmission".

CHAPTER 6-2: CEC Reception/ Remote Reception

Functions and operations of CEC reception/remote reception are explained as follows.

1. Overview
2. Configuration
3. Operations
4. Example of Setting
5. Registers

CODE: 9BFRCEC-E1.0

1. Overview

CEC reception/remote reception is used for receiving HDMI-CEC signals and infrared remote control signals. The features are as follows.

Features

- Capable of adjusting detection timings for start bit and data bit
- Equipped with noise filter
- Operating modes supporting the following standards can be selected
  - SIRCS
  - NEC/Association for Electric Home Appliances
  - HDMI-CEC

Features of operating modes

■ SIRCS mode

- Start bit detection and interrupt output
- Minimum pulse width violation detection
- Device address comparison
- Counter overflow detection and interrupt output

■ NEC/Association for Electric Home Appliances mode

- Start bit detection and interrupt output
- Repeat code detection and interrupt output
- Minimum pulse width violation detection
- Counter overflow detection and interrupt output

■ HDMI-CEC mode

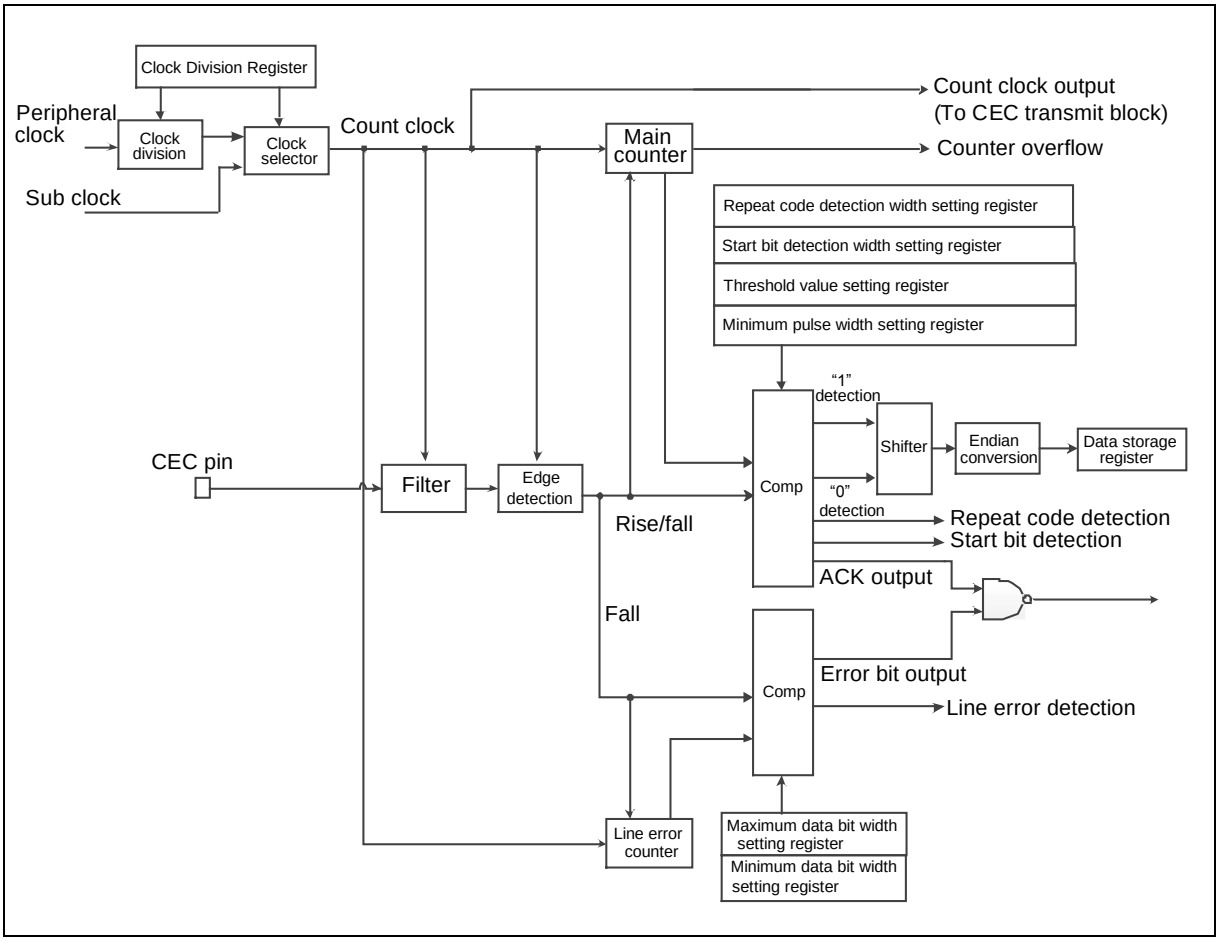
- Start bit detection and interrupt output
- Minimum pulse width violation detection
- Counter overflow detection and interrupt output
- Device address comparison
- Minimum data bit width violation detection and interrupt output (supporting HDMI-CEC line error handling standard)
- Automatic error pulse output (supporting HDMI-CEC line error handling standard)
- Maximum data bit width violation detection and interrupt output
- EOM detection
- ACK detection and interrupt output
- Automatic ACK output

2. Configuration

Block diagram of CEC reception/remote reception is as follows.

Block Diagram

Figure 2-1 CEC Reception/Remote Reception Block Diagram



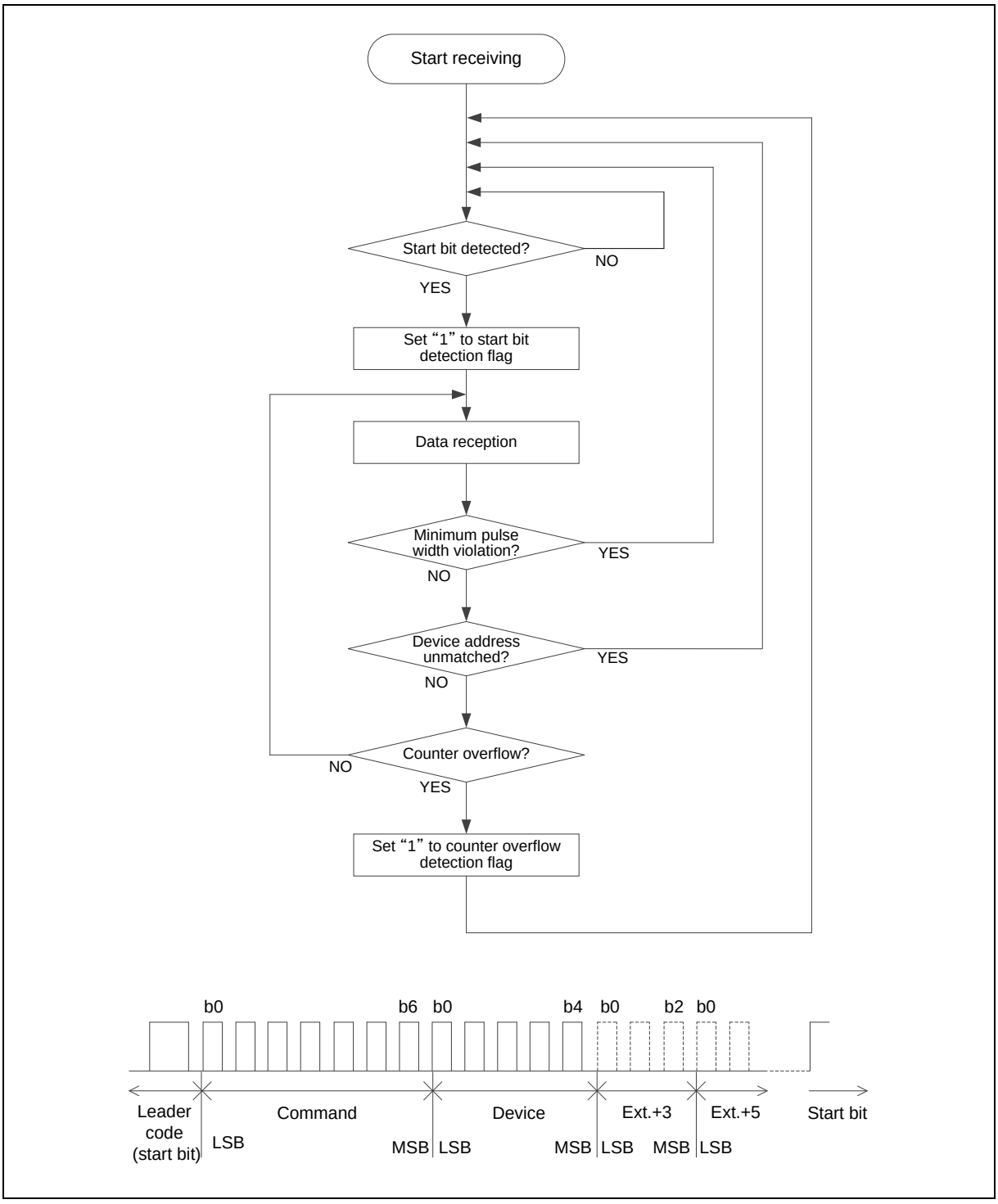
3. Operations

This chapter explains the operations of CEC reception/remote control reception.

3.1 SIRCS mode

3.1.1 Operational flow chart and waves of SIRCS mode

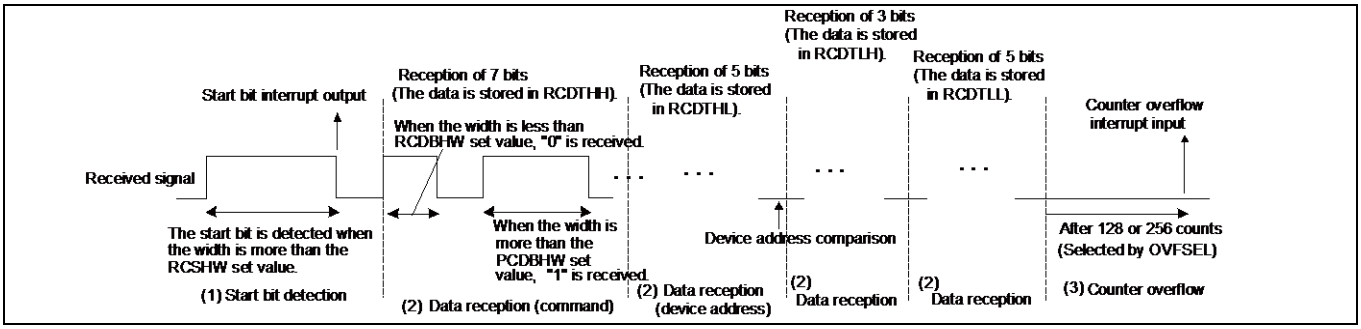
Figure 3-1 Operational Flow Chart and Waves



3.1.2 Basic operations of SIRCS mode

The SIRCS mode counts the width of "High" duration in the received signal with the count clock, and receives the data.

Figure 3-2 Operations of SIRCS Mode



Basic operations

The basic operations are as follows:

(1) If the width of "High" duration more than the set value of RCSHW is input, the start bit is detected and the data receiving state is entered.

(2) Figure 3-2 shows the operation at THSEL=0 (RCCR register). In the operation, "0" is received for the signal less than the RCDBHW set value and "1" is received for the signal more than the RCDBHW set value.
After receiving the 7-bit command, the device address is received for the data reception. 5-bit device address becomes an address match if its address is the same as either of RCADR1 or RCADR2 value. When the address is not matched with the both values, the state returns to the start bit detection waiting state.

(3) For overflowing after data is received, the start bit detection waiting state is resumed.

3.1.3 Start bit detection and interrupt output

Figure 3-3 Start Bit Detection of SIRCS Mode

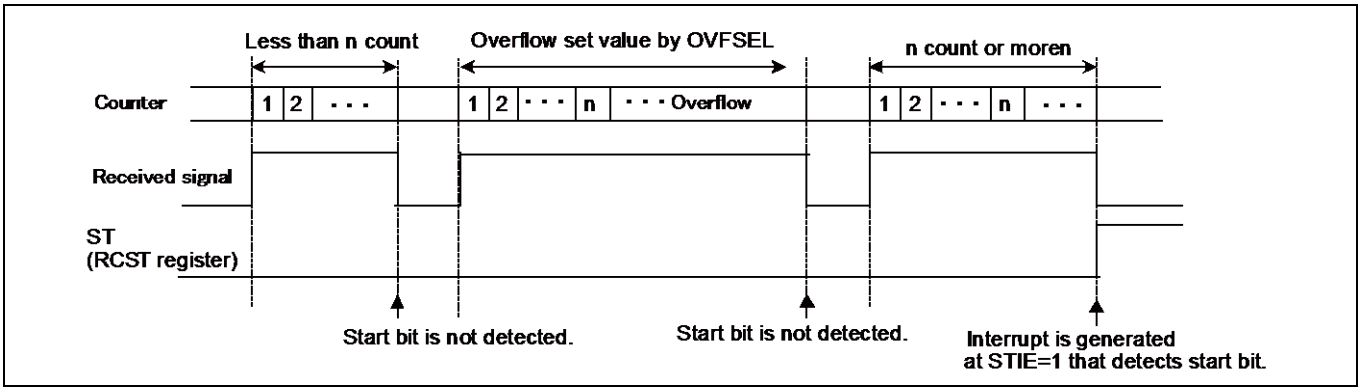


Figure 3-3 explains the start bit detection when RCSHW=n is set.
If the width of "High" duration of "n" or more is input with the start bit detection waited, ST=1 (RCST register) is set by detecting the start bit. Moreover, when STIE=1 (RCST register) is set beforehand, the interrupt is output by detecting the start bit.
Moreover, when the width of "High" duration more than the number of counts specified by OVFSEL (RCST register) setting is input, the overflow occurs and the start bit is not detected.

3.1.4 Minimum pulse width violation

Figure 3-4 Minimum Pulse Width Violation

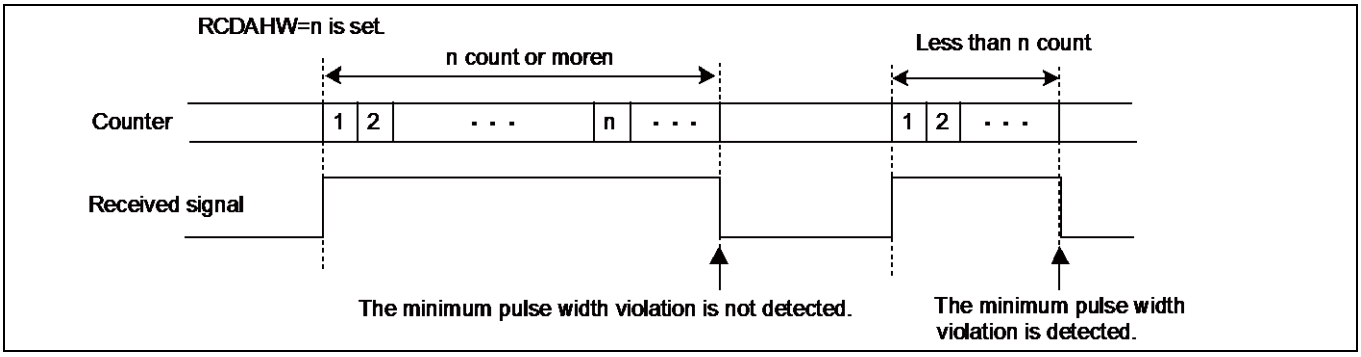


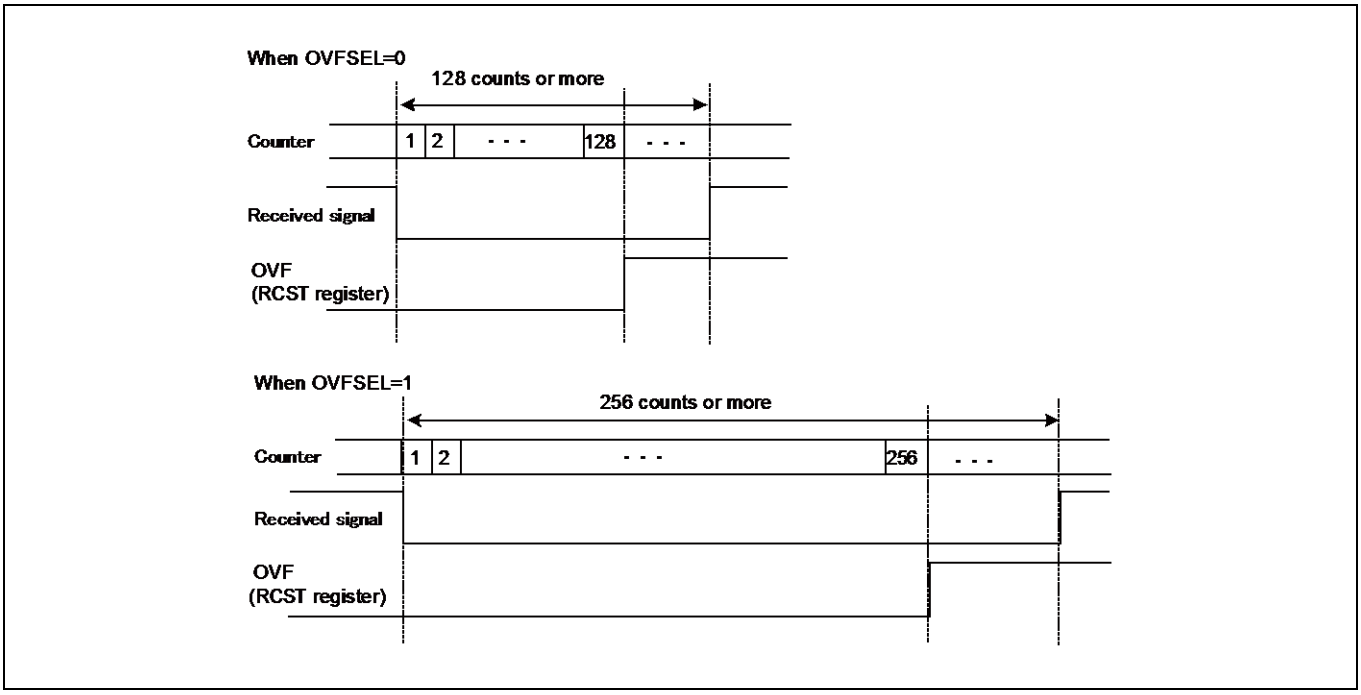
Figure 3-4 explains the minimum pulse width violation when RCDAHW=n is set.
When the signal of less than n is input during the reception operation, the state of the start bit detection waiting is resumed by detecting the minimum pulse width violation.

3.1.5 Device address comparison

In the SIRCS mode, the 5-bit device address is received. For ADRCE=1 (RCCR register), the device address comparison is executed.
The device address becomes an address match if its address is the same as either of RCADR1 or RCADR2 value. When the address is not matched with the both values, the start bit detection waiting state is resumed.

3.1.6 Counter overflow detection and interrupt output

Figure 3-5 Counter Overflow

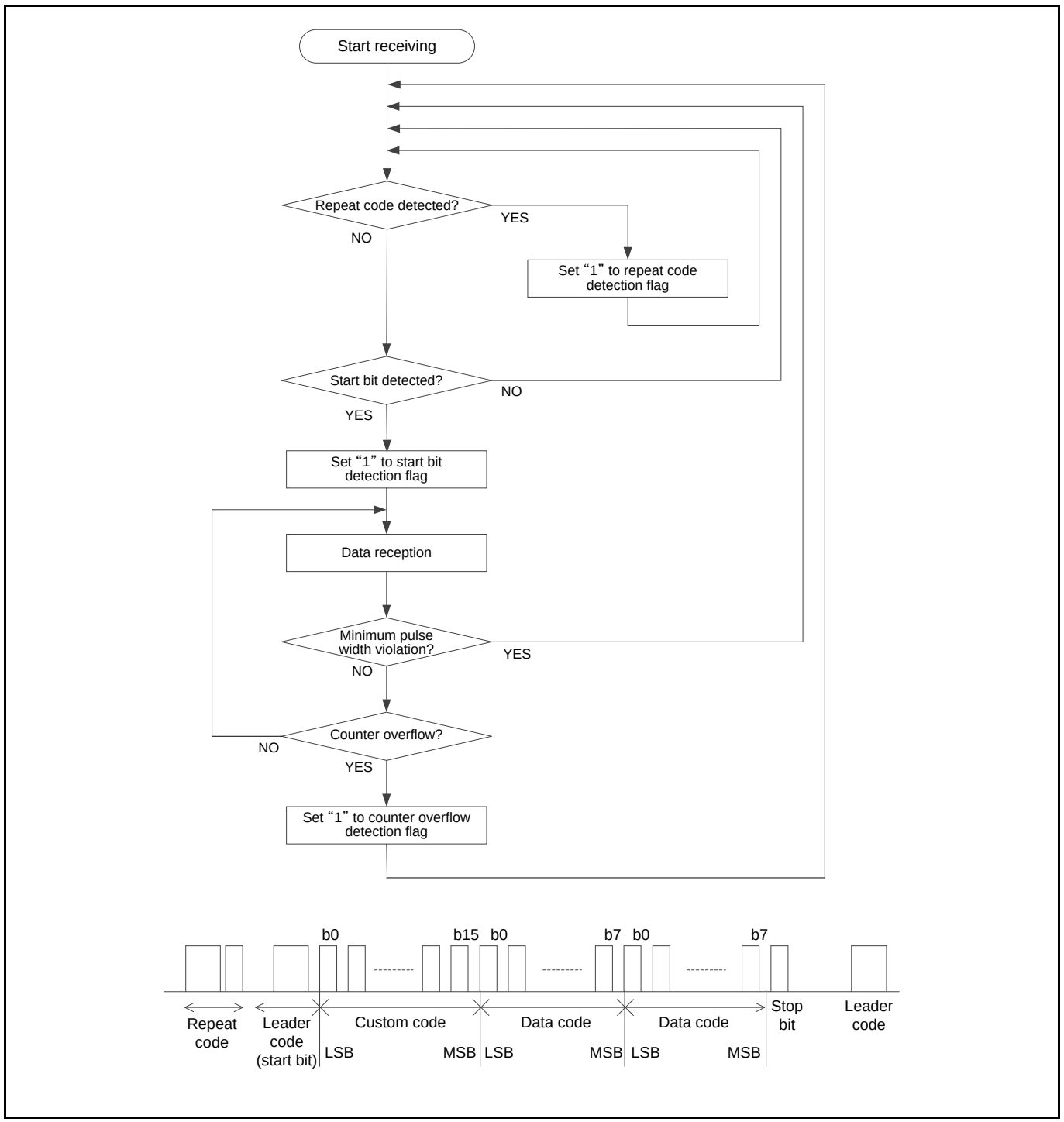


For OVFSEL=0 (RCST register), an overflow occurs and the start bit detection waiting state is resumed when High or Low input continues more than 128 counts. Moreover, for OVFSEL=1, an overflow occurs at 256 counts.
When OVFIE=1 (RCST register) is set beforehand, the interrupt is output after an overflow.

3.2 Operations of NEC/Association for Electric Home Appliances mode

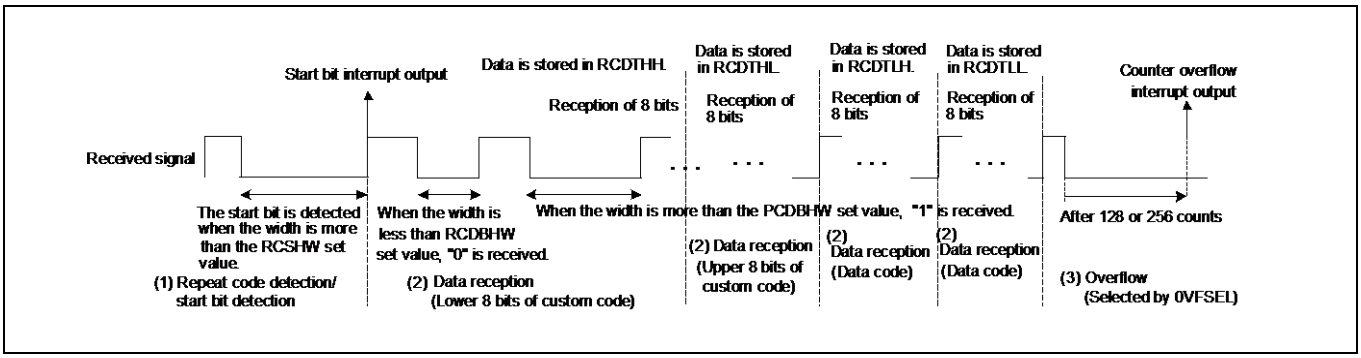
3.2.1 Operational flow chart and waves of NEC/Association for Electric Home Appliances mode

Figure 3-6 Operational flow chart and waves of NEC/ Association for Electric Home Appliances Mode



In NEC/Association for Electric Home Appliances mode, the count clock counts the width of “Low” duration of the received signal and the data is received.

Figure 3-7 Operations of NEC/ Association for Electric Home Appliances mode



Basic Operations

The basic operations are as follows:

1. When the width of "Low" duration of the RCSHW set value or less and the RCRHW set value or more is input, the repeat code is detected. Moreover, if the width of "Low" duration of the RCSHW set value or more is input, the data reception state is entered by detecting the start bit.
2. Figure 3-7 shows the operations for THSEL=0 (RCCR register). In the operations, "0" is received for the signal of less than the RCDBHW set value and "1" is received for the signal of the RCDBHW set value or more.
   In the data reception, the custom code of two bytes and data code of two bytes are received
3. When an overflow occurs after the data reception, the start bit/repeat bit detection waiting state is resumed.

3.2.2 Start bit detection

Figure 3-8 Start bit detection

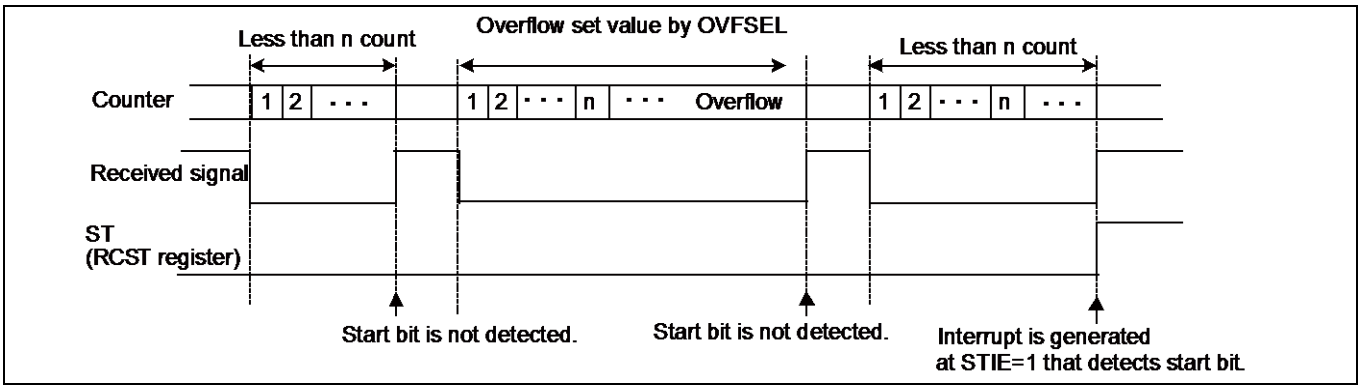


Figure 3-8 explains the start bit detection when "RCSHW=n" is set.
When the width of "Low" duration of n or more is input during the start bit detection waiting, ST=1 (RCST register) is set by detecting the start bit. Moreover, when STIE=1 (RCST register) is set beforehand, the interrupt is output by detecting the start bit.
Moreover, when the width of "Low" duration of the number of counts specified by OVFSEL (RCST register) setting or more is input, an overflow occurs and the start bit is not detected.

3.2.3 Repeat code detection

Figure 3-9 Repeat code detection

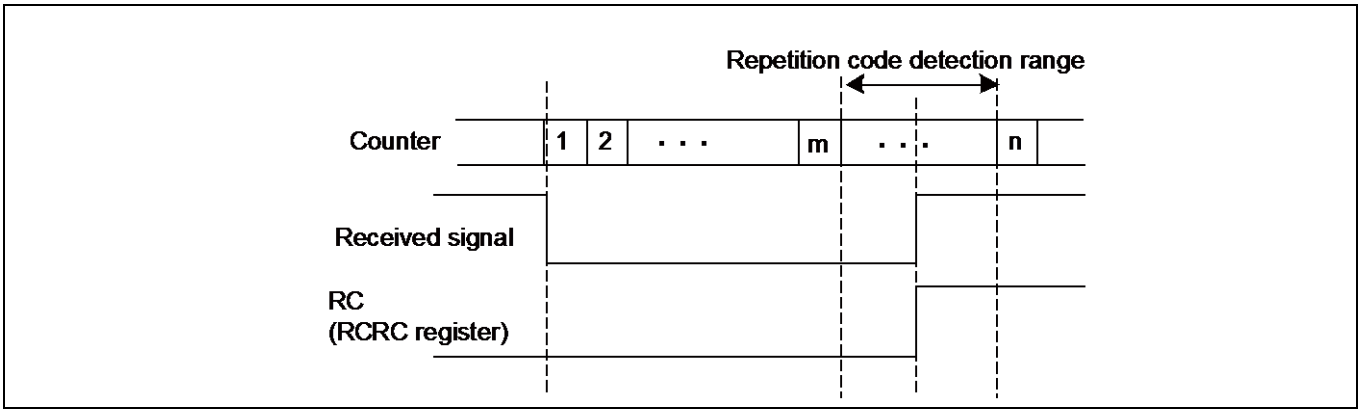


Figure 3-9 explains the start bit detection when RCRHW=m and RCSHW=n are set.
When the "Low" signal of the width of less than n and m or more is input at the reception beginning, RC=1 (RCRC register) is set by detecting the repeat code.
The repeat code is detected only in NEC/Association for Electric Home Appliances mode.

3.2.4 Minimum pulse width violation

Figure 3-10 Minimum pulse width violation

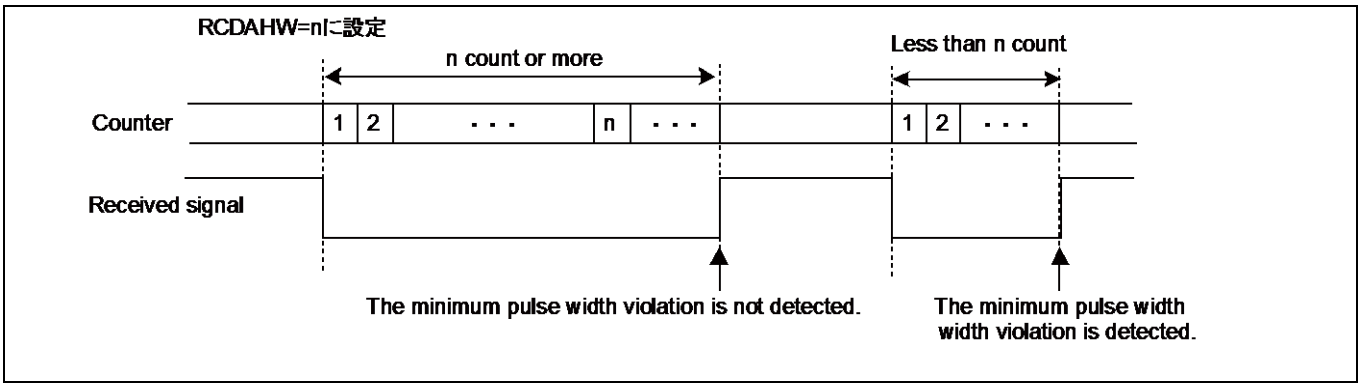
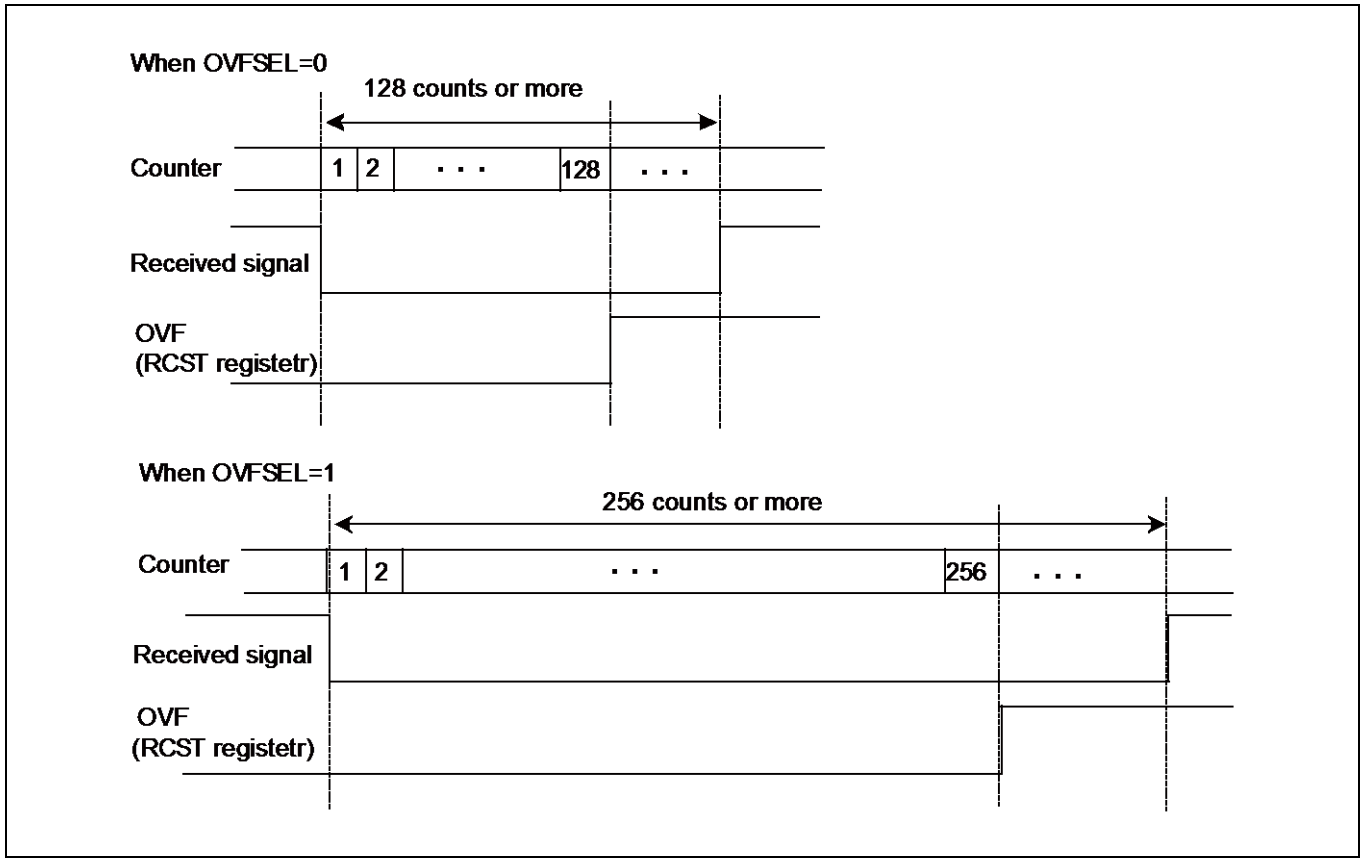


Figure 3-10 explains the minimum pulse width violation when RCDAHW=n is set.
When the width of "Low" duration of less than n is input during the reception operation, the start bit detection waiting state is resumed by detecting the minimum pulse width violation.

3.2.5 Counter overflow detection and interrupt output

Figure 3-11 Counter overflow

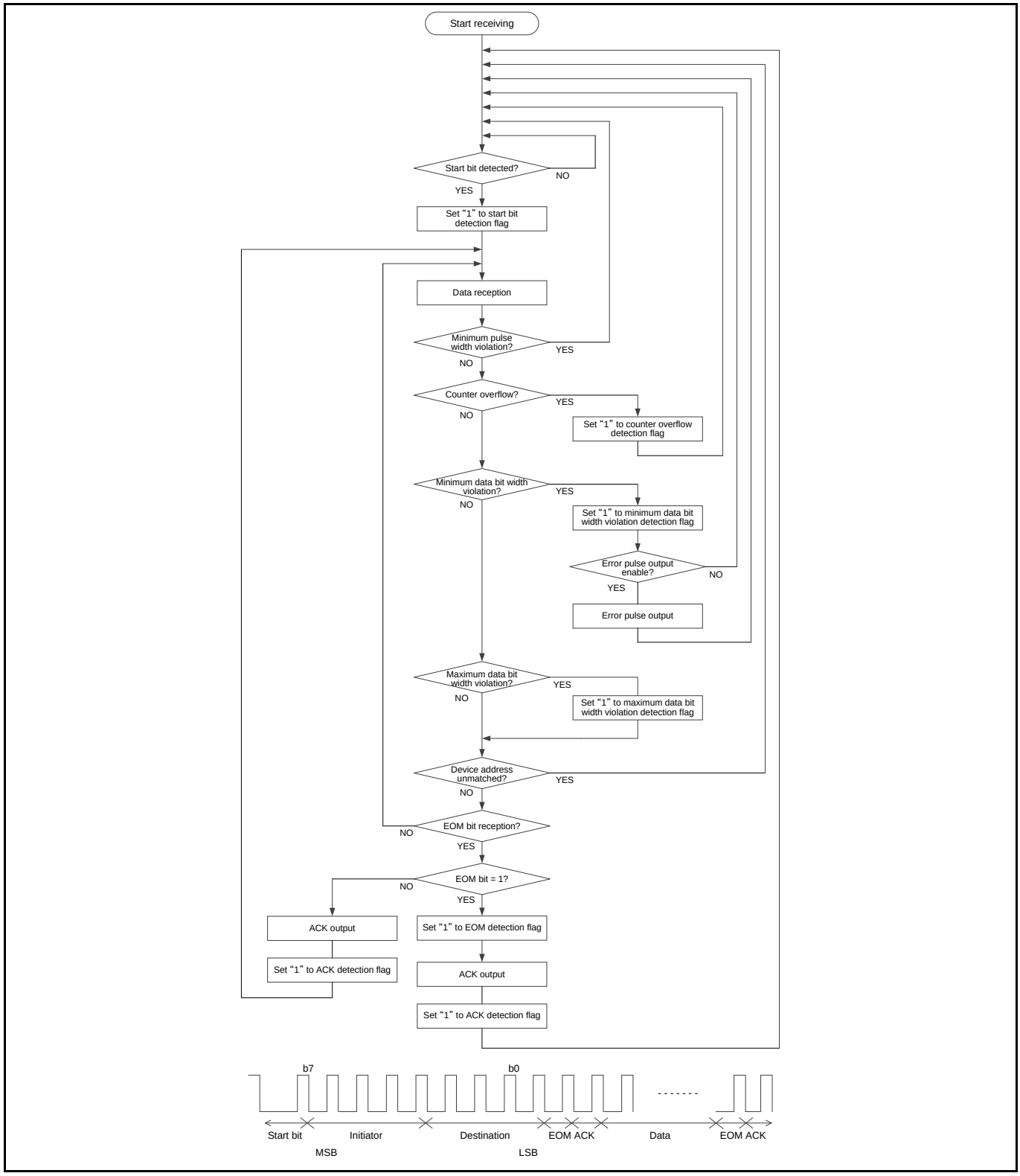


If the "High" or "Low" input of 128 counts or more continues for OVFSEL=0(RCST register), an overflow occurs and the start bit detection waiting state is resumed. Moreover, an overflow occurs with 256 counts of the continuous "High" or "Low" input for OVFSEL=1.
When OVFIE=1 (RCST register) is set beforehand, an overflow occurs and an interrupt is output.

3.3 HDMI-CEC mode

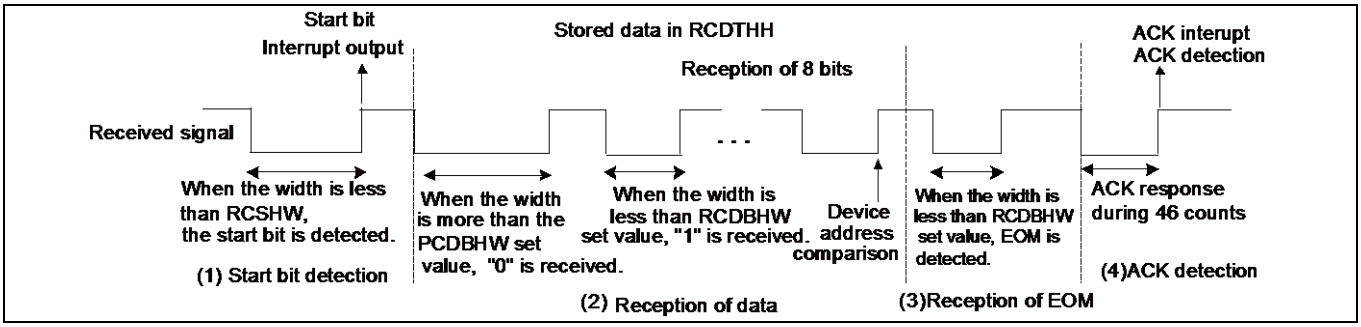
3.3.1 Operational flow chart and waves in HDMI-CEC mode

Figure 3-12 Operational flow chart and waves in HDMI-CEC mode



In the HDMI-CEC mode, the count clock counts the width of "Low" duration of the received signal and the data is received.

Figure 3-13 Operations in HDMI-CEC mode



Basic Operations

The basic operations are as follows:

1. When the width of "Low" duration of less than the RCSHW set value is input, the start bit is detected and the data receiving state is resumed.
2. Figure 3-13 shows the operations at THSEL=1 (RCCR register). For a signal of the RCDBHW set value or more, "0" is received, and for a signal of less than the RCDBHW set value, "1" is received.
   Received data of 8 bits is stored in RCDTHH and the lower 4 bits are compared with the device address. If the destination of 4 bits is the same as either of RCADR1 or RCADR2 value, the address becomes the address match. When the address is not matched with the both values, the start bit detection waiting state is resumed.
3. When EOM is detected after the data reception, EOM=1 (RCST register) is set and the data reception is completed. When EOM is not detected, EOM=0 (RCST register) is held and the data receiving state is resumed to store the received data in RCDTHH again.
4. When "Low" signal is input after the reception of the EOM bit, the ACK signal is output and the start bit detection waiting state is resumed.

3.3.2 Start bit detection and interrupt output

Figure 3-14 Detection of start bit in HDMI-CEC mode

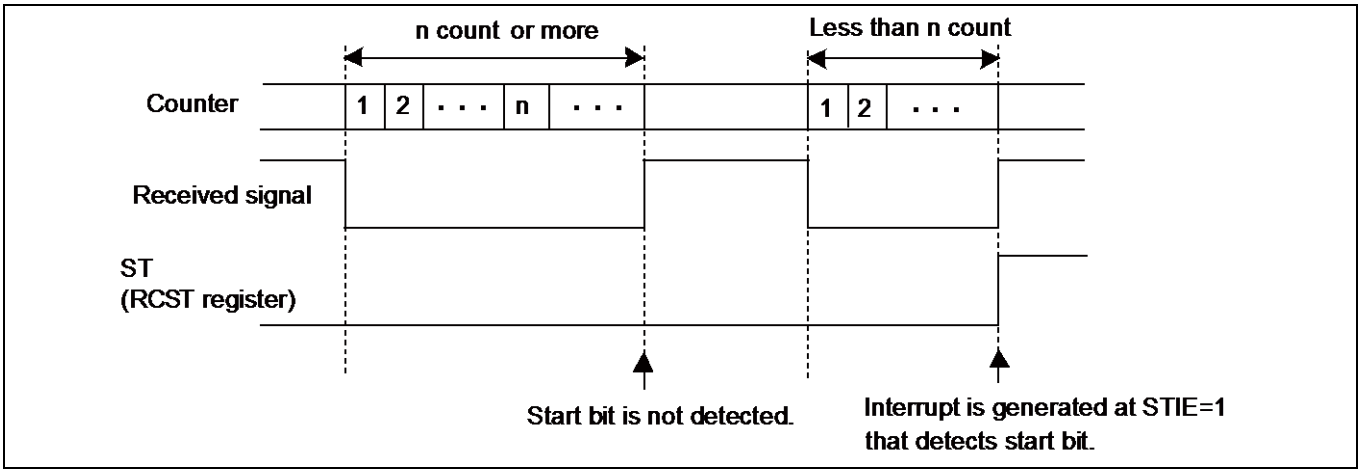


Figure 3-14 shows the start bit detection when "RCSHW=n" is set (the operations for THSEL=1). When the width of "Low" duration of less than n is input with the start bit detection waiting, the start bit is detected and ST=1 (RCST register) is set. Moreover, when STIE=1 (RCST register) is set beforehand, the interrupt is output by detecting the start bit.

3.3.3 Minimum pulse width violation

Figure 3-15 Minimum pulse width violation

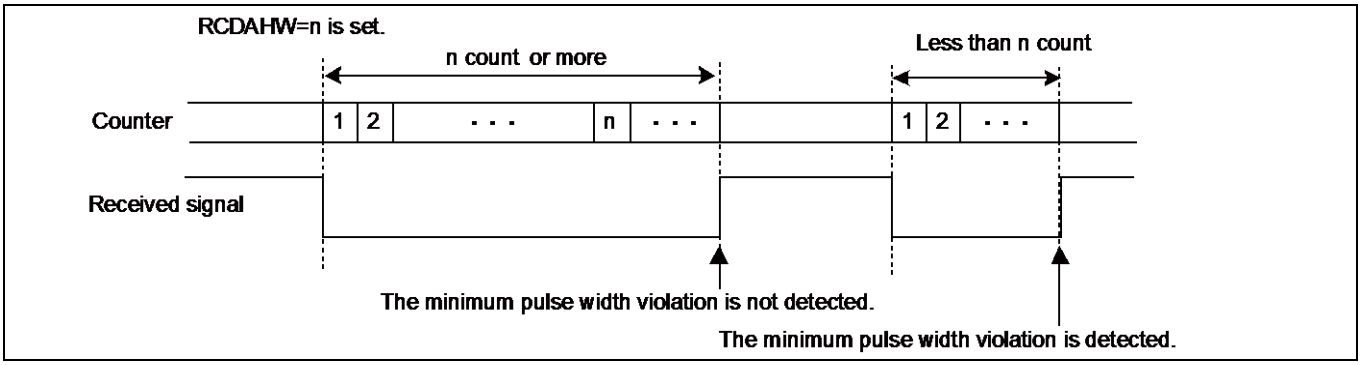
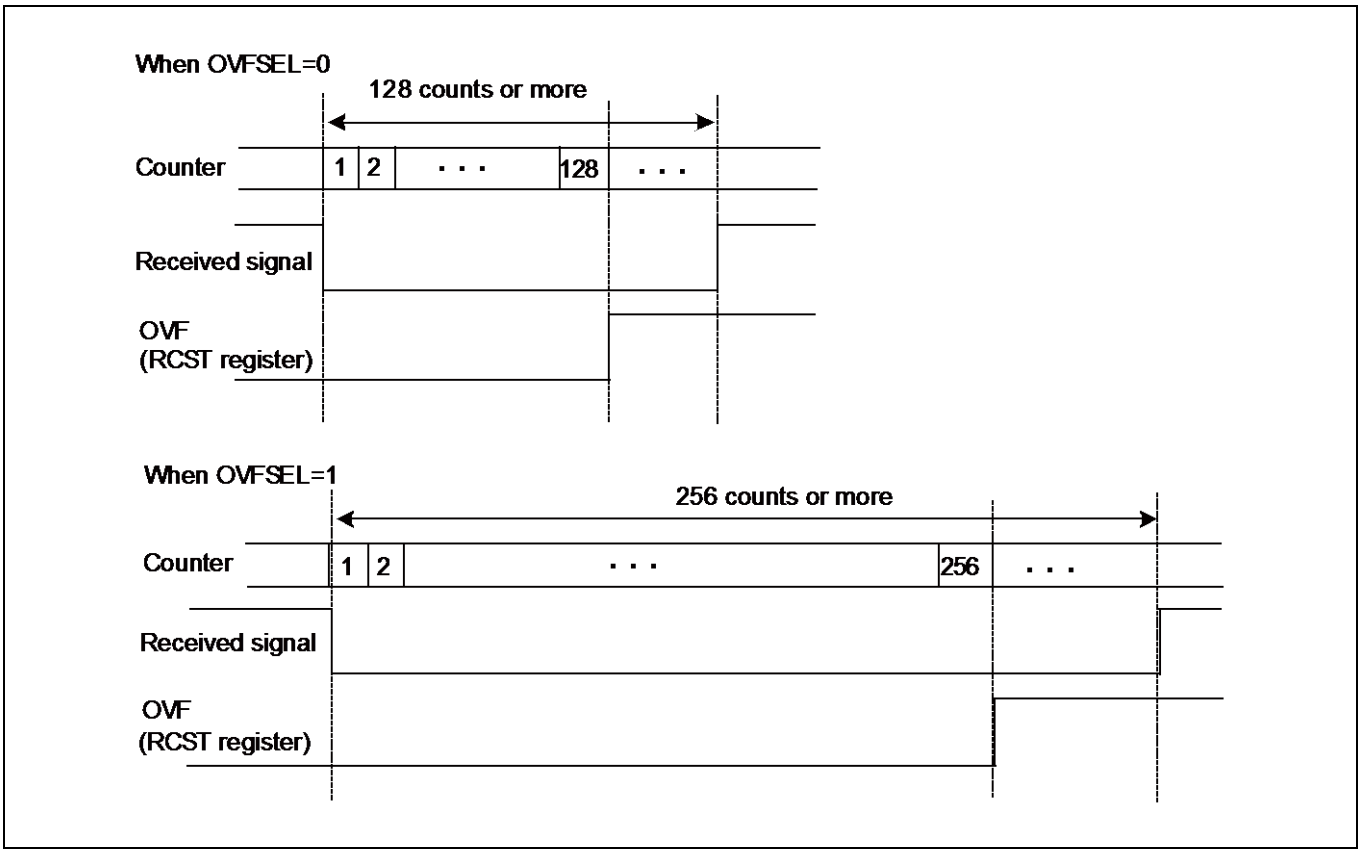


Figure 3-15 shows the minimum pulse width violation when RCDAHW=n is set.
When the signal of less than n is input during the reception operation, the minimum pulse width violation is detected and the start bit detection waiting state is resumed.

3.3.4 Counter overflow detection and interrupt output

Figure 3-16 Counter overflow



If the "High" or "Low" input of 128 counts or more continues for OVFSEL=0(RCST register), an overflow occurs and the start bit detection waiting state is resumed. Moreover, an overflow occurs with 256 counts of the continuous "High" or "Low" input for OVFSEL=1.
When "OVFIE=1 (RCST register) " is set beforehand, an overflow occurs and an interrupt is output.

3.3.5 Device address comparison

In the HDMI-CEC mode, the destination of 4 bits is received. For ADRCE=1 (RCCR register), the device address comparison is executed.
If the destination is the same as either of RCADR1 or RCADR2 value, the address becomes the address match. Moreover, for the broadcast address, an address match is achieved.
When the address is not matched with the both values, the start bit detection waiting state is resumed.

3.3.6 Data bit width violation and error pulse automatic output

Figure 3-17 Minimum data bit width violation

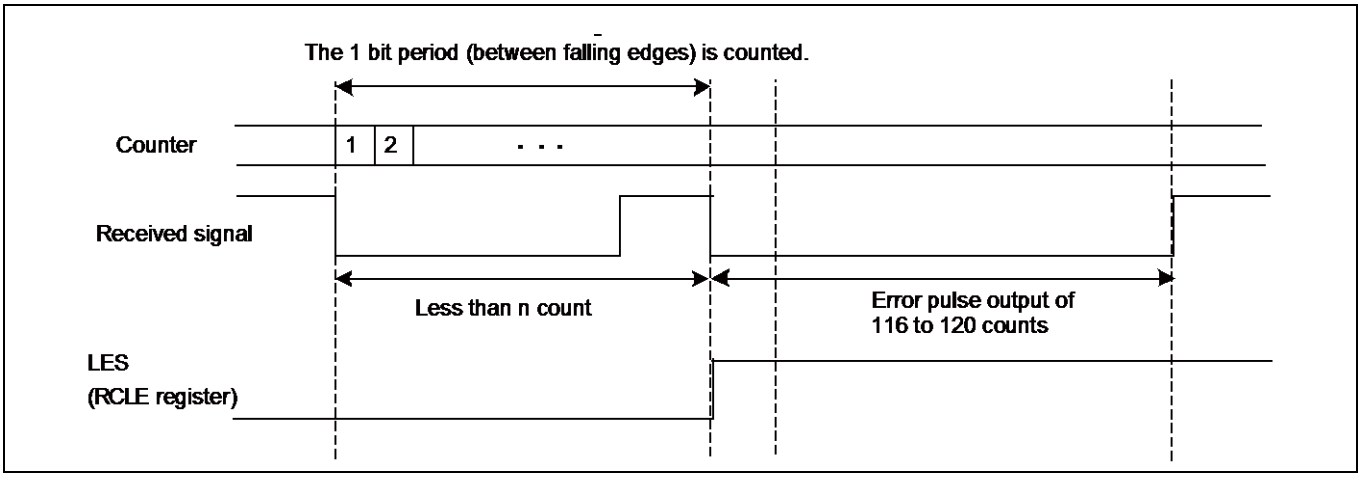


Figure 3-17 explains the minimum data bit width violation when RCLESW=n is set.
At LES=1 (RCLE register), when the 1 bit period (the period between the falling edges) is smaller than the set value of minimum data bit width setting register (RCLESW), the minimum data bit width violation is detected and LES=1 (RCLE register) is set.
When LESIE=1 (RCLE register) is set beforehand, the interrupt is output by detecting the violation of minimum data bit width. Moreover, when EPE=1 (RCLE register) is set, by detecting the violation, the error pulse is output as shown in Figure 3-17.

Figure 3-18 Maximum data bit width violation

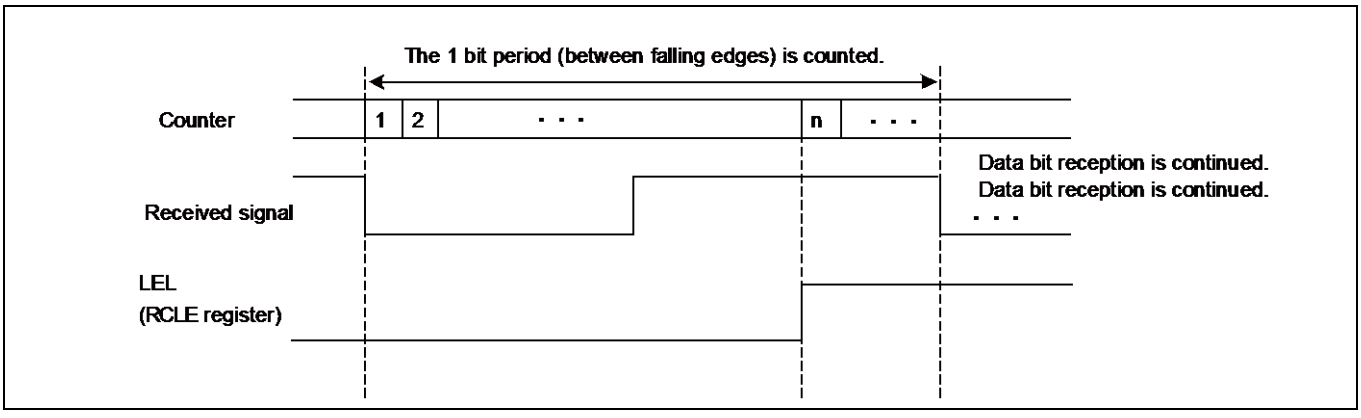


Figure 3-17 explains the minimum data bit width violation when RCLELW=n is set.
For LELE=1 (RCLE register), when the 1 bit period (the period between the falling edges) is more than the set value of maximum data bit width setting register (RCLELW), LEL=1 (RCLE register) is set by detecting the maximum data bit width violation. When LELIE=1 (RCLE register) is set beforehand, the interrupt is output by detecting the maximum data bit width violation.

3.3.7 EOM detection

Figure 3-19 EOM detection

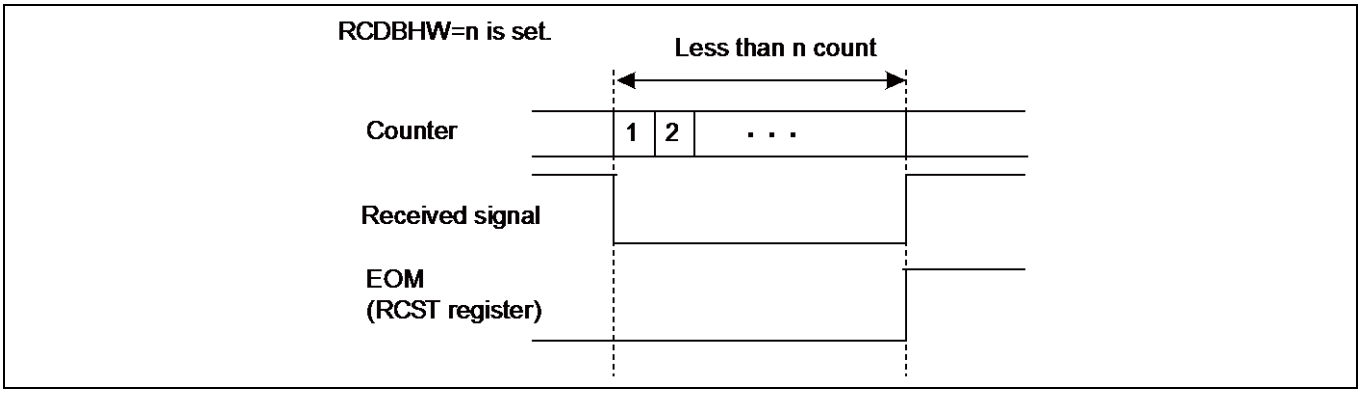
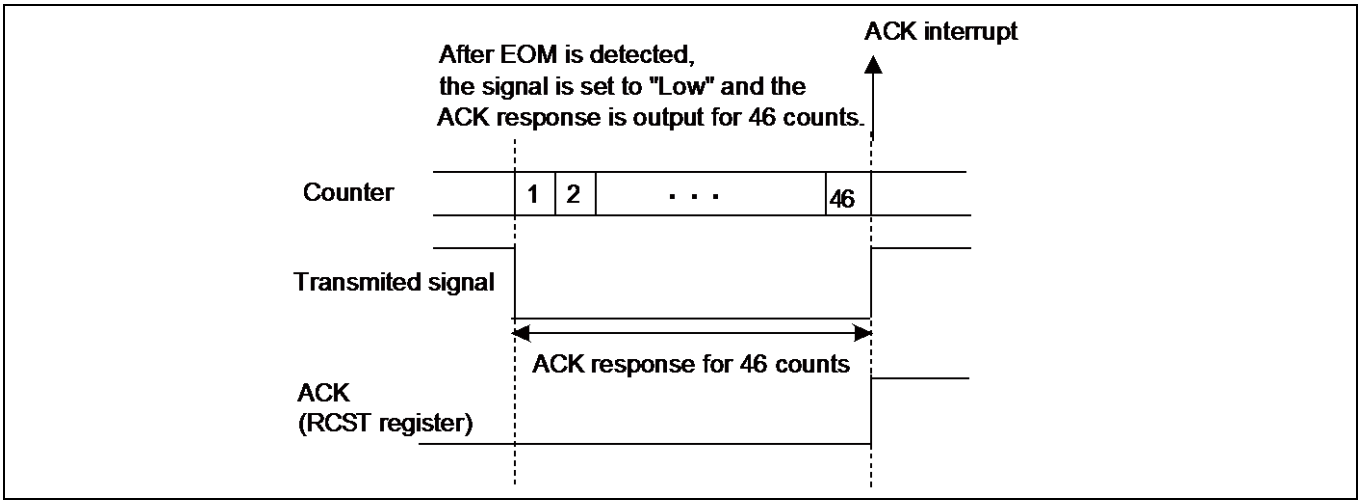


Figure 3-19 shows the operation for THSEL=1 (RCCR register). If the "Low" signal of less than RCDBHW set value is input in EOM bit receiving state, EOM=1 (RCST register) is set by detecting EOM.

3.3.8 ACK detection and interrupt output

Figure 3-20 ACK detection and interrupt output



When "Low" signal is input after EOM detection, "Low" signal is output for 46 counts as ACK response. If the "High" signal is input after "Low" signal is output, ACK=1 (RCST register) is set by detecting the ACK signal. When ACKIE=1 (RCST register) is set beforehand, the interrupt is output by detecting ACK signal.
When address enable bit (ADRCE) of the RCCR register is "1", ACK signal is output only if the address match is detected. For the broadcast address, though it is considered to be the address match, ACK response is not executed.

3.4 Noise filter

When the input of CEC signal changes in the width of less than two clocks of the count clock, the input signal is judged to be a noise and removed.

4. Example of Setting

Example of setting is explained as follows (in case of operating clock at 32.768 kHz).

Table 4-1 Example of setting in remote mode (SIRCS)

Registers	Setting value	Remarks
Reception Control Register	MOD=00, THSEL=0, ADRCE=1	
Reception Interrupt Control Register	ACKIE=0, OVFIE=1	
	OVFSEL=0	3.9 ms
Start Bit Detection Width Setting Register	76	2.3 ms
Minimum Pulse Width Setting Register	17	0.52 ms
Threshold Value Setting Register	37	1.1 ms

Table 4-2 Example of setting in remote mode (NEC)

Registers	Setting value	Remarks
Reception Control Register	MOD=10, THSEL=0	
Reception Interrupt Control Register	ACKIE=0, OVFIE=1	
	OVFSEL=1	7.8 ms
Start Bit Detection Width Setting Register	144	4.4 ms
Minimum Pulse Width Setting Register	15	0.46 ms
Threshold Value Setting Register	52	1.6 ms
Repeat Code Interrupt Control Register	RCIE=1	
Repeat Code Detection Width Setting Register	65	2.0 ms

Table 4-3 Example of setting in HDMI-CEC remote mode

Registers	Setting value	Remarks
Reception Control Register	MOD=11, THSEL=1, ADRCE=1	
Reception Interrupt Control Register	ACKIE=1, OVFIE=1	
	OVFSEL=1	7.8 ms
Start Bit Detection Width Setting Register	114	3.5 ms
Minimum Pulse Width Setting Register	13	0.4 ms
Threshold Value Setting Register	42	1.3 ms
Maximum/Minimum Data Bit Width Violation Control Register	LELIE=1, LESIE=1, LELE=1, LESE=1, EPE=1	
Maximum Data Bit Width Setting Register	91	2.8 ms
Minimum Data Bit Width Setting Register	65	2.0 ms

5. Registers

The list of registers is as follows.

Table 5-1 Registers List

Abbreviated Register Name	Register Name	Reference
RCCR	Reception Control Register	5.1
RCST	Reception Interrupt Control Register	5.2
RCADR1	Device Address Setting Register 1	5.3
RCADR2	Device Address Setting Register 2	5.3
RCSHW	Start Bit Detection Width Setting Register	5.4
RCDAHW	Minimum Pulse Width Setting Register	5.5
RCDBHW	Threshold Value Setting Register	5.6
RCDTHH	Data Save Register HH	5.7
RCDTHL	Data Save Register HL	
RCDTLH	Data Save Register LH	
RCDTLL	Data Save Register LL	
RCCKD	Clock Division Register	5.8
RCRC	Repeat Code Interrupt Control Register	5.9
RCRHW	Repeat Code Detection Width Setting Register	5.10
RCLE	Data Bit Width Violation Interrupt Control Register	5.11
RCLESW	Minimum Data Bit Width Setting Register	5.12
RCLELW	Maximum Data Bit Width Setting Register	5.13

5.1 Reception Control Register (RCCR)

Configuration of Reception Control Register (RCCR) bits is as follows.

bit	7	6	5	4	3	2	1	0
Field	THSEL	Reserved			ADRCE	MOD1	MOD0	EN
Attribute	R/W				R/W	R/W	R/W	R/W
Initial Value	0				0	0	0	0

[bit7] THSEL: Threshold value selection bit

Use RCDAHW and RCDBHW to set a reference for determining "0" or "1".

States	THSEL	
	0	1
W > RCDAHW	"0" data	"1" data
W < RCDBHW		
W > RCDAHW	"1" data	"0" data
W ≥ RCDBHW		

[bit6:4] Reserved: Reserved bits

"0" is always read.
Set "0" for write.

[bit3] ADRCE: Address comparison enable bit

Initial value of this bit is "0" (comparison disabled) and setting this bit to “1” enables comparison between reception address and device address.
An ACK/OVF interrupt will be generated only if the address is matched when comparison is enabled.
In CEC mode, an ACK response will be returned when address match is detected. If the address is an broadcast address, it will be handled as a match but no ACK response will be returned.
In modes other than SIRCS mode or HDMI-CEC mode, set this bit to “0”.

[bit2:1] MOD1, MOD0: Operation mode setting bits

bit2	bit1	Function
0	0	SIRCS mode [Initial value]
0	1	Setting prohibited
1	0	NEC/Association for Electric Home Appliances mode
1	1	HDMI-CEC mode

In modes other than SIRCS mode (MOD1=1), input signals will be inverted internally.
“H” width comparison is applied to "L" width.

[bit0] EN: Operation enable bit

Setting this bit to "1" will start reception operation.
The initial value is "0" (stop).

Note:

- *Do not change the following setting registers and bits while this bit is "1" (operating).*

  THSEL bit, ADRCE bit and MOD bit of RCCR register
  OVFSEL bit of RCST register
  RCSHW, RCDAHW, RCDBHW, RCADR1, RCADR2, and RCCKD registers
  RCRC, RCRHW, RCLE, RCLELW, and RCLESW registers

5.2 Reception Interrupt Control Register (RCST)

Configuration of Reception Interrupt Control Register (RCST) bits is as follows.

bit	7	6	5	4	3	2	1	0
Field	STIE	ACKIE	OVFIE	OVFSEL	ST	ACK	EOM	OVF
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

[bit7] STIE: Start bit interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit6] ACKIE: ACK interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled

This bit is valid only in HDMI-CEC mode.

[bit5] OVFIE: Counter overflow interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled

This interrupt will be generated only if an overflow is detected after a start bit is detected.
No interrupt will be generated without detecting a start bit.

[bit4] OVFSEL: Counter overflow detection condition setting bit

bit	Description
0	An overflow will occur after the counter counted 128 clocks.
1	An overflow will occur after the counter counted 256 clocks.

[bit3] ST: Start bit detection bit

bit	Description
0	Start bit has not been detected
1	Start bit has been detected

Writing "0" will clear this bit.
An interrupt will be generated if a start bit is detected while STIE bit is "1".

[bit2] ACK: ACK detection bit

bit	Description
0	ACK not detected
1	ACK detected

Writing "0" will clear this bit.
An interrupt will be generated if an ACK is detected while ACKIE bit is "1".
An interrupt will be generated only if the address is matched when address comparison is enabled.
This bit is valid only in HDMI-CEC mode.

[bit1] EOM: EOM detection bit

bit	Description
0	EOM not detected
1	EOM detected

Writing "0" will clear this bit.
This bit is valid only in HDMI-CEC mode.

[bit0] OVF: Counter overflow detection bit

bit	Description
0	Counter overflow not detected
1	Counter overflow detected

An interrupt will be generated only if the address is matched when address comparison is enabled.
Writing "0" will clear this bit.
In SIRCS mode, OVF flag will not be set until the second byte is received.

5.3 Device Address Setting Register 1, 2 (RCADR1, RCADR2)

Configuration of Device Address Setting Register 1, 2 (RCADR1, RCADR2) bits is as follows.

bit	7	6	5	4	3	2	1	0
Field	Reserved			RCADR1, 2				
Attribute				R/W				
Initial Value				00000				

[bit7:5] Reserved: Reserved bits

"0" is always read.
Set "0" for write.

[bit4:0] RCADR1, 2: Device address setting bits

Address set in this register will be compared to the received device address or HDMI-CEC destination.
In HDMI-CEC mode, if “0x0F”(broadcast address) is set to this register, ACK response is not given by the an address reception including broadcast address

5.4 Start Bit Detection Width Setting Register (RCSHW)

Configuration of Start Bit Detection Width Setting Register (RCSHW) bits is as follows.

bit	7	6	5	4	3	2	1	0
Field	RCSHW							
Attribute	R/W							
Initial Value	0x00							

This register is used to set a duration of the start bit.
If "H" with a width over the set value is received, it is identified as a start bit.
If the width of received signals is less than the set value, the start bit will not be detected and it once again becomes a state to wait for detecting a start bit.
When OVFSEL=0, the set value must be RCSHW ≤ 127 (equal to or less than a value not to be detected as overflow).

5.5 Minimum Pulse Width Setting Register (RCDAHW)

Configuration of the Minimum Pulse Width Setting Register (RCDAHW) bits is as follows.

bit	7	6	5	4	3	2	1	0
Field	RCDAHW							
Attribute	R/W							
Initial Value	0x00							

[bit7:0] RCDAHW

This is register used to set the minimum pulse width duration.

Values to be set in this register must be: 2 ≤ RCDAHW < RCDBHW.

In CEC mode, it must be RCDAHW < 46 (less than the ACK response pulse width).

If a signal with a width < RCDAHW is received, it will be detected as minimum pulse width violation.

5.6 Threshold Value Setting Register (RCDBHW)

Configuration of the threshold Value Setting Register (RCDBHW) bits is as follows.

bit	7	6	5	4	3	2	1	0
Field	RCDBHW							
Attribute	R/W							
Initial Value	0x00							

[bit7:0] RCDBHW

This is register used to set the threshold value of data reception signal width.

Do not set a value less than RCCDAHW.

Be sure to set a value: RCCDAHW < RCCDBHW < RCSHW.

5.7 Data Save Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

Configuration of the Data Save Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL) bits is as follows.

bit	31	30	29	28	27	26	25	24
Field	RCDTHH							
Attribute	R							
Initial Value	0x00							

bit	23	22	21	20	19	18	17	16
Field	RCDTHL							
Attribute	R							
Initial Value	0x00							

bit	15	14	13	12	11	10	9	8
Field	RCDTLH							
Attribute	R							
Initial Value	0x00							

bit	7	6	5	4	3	2	1	0
Field	RCDTLL							
Attribute	R							
Initial Value	0x00							

This register is used to store received data.
In HDMI-CEC mode, the received data will be stored in the RCDTHH.
In remote control mode, every 8 bits reception will be stored from RCDTHH.
If a counter overflow interrupt is generated, the bits already received by then will be stored from the MSB.
If EN bit of the RCCR register is "0", unknown values will be read from this register.
If signals over 4 bytes are received, the excess will be ignored and not be reflected to the register.

5.8 Clock Division Setting Register (RCCKD)

Configuration of the Clock Division Setting Register (RCCKD) bits is as follows.

bit	15	14	13	12	11	10	9	8
Field	Reserved			CKSEL	CKDIV			
Attribute				R/W	R/W			
Initial Value				0	0000			

bit	7	6	5	4	3	2	1	0
Field	CKDIV							
Attribute	R/W							
Initial Value	0x00							

[bit15:13] Reserved: Reserved bits

"0" is always read.

Set "0" for write.

[bit12] CKSEL: Operating clock selection bit

bit	Description
0	Clock divided from peripheral clock (PCLK) is selected.
1	Sub-clock is selected.

[bit11:0] CKDIV: Operating clock division setting bits

Division ratio becomes CKDIV + 1.

1 division (no division) through 4096 division can be set (no division if CKSEL=1).

5.9 Repeat Code Interrupt Control Register (RCRC)

This register controls repeat code interrupts.

bit	7	6	5	4	3	2	1	0
Field	Reserved			RCIE	Reserved			RC
Attribute				R/W				R/W
Initial Value				0				0

[bit7:5] Reserved: Reserved bits

"0" is always read.

Set "0" for write.

[bit4] RCIE: Repeat Code Interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit3:1] Reserved: Reserved bits

"0" is always read.

Set "0" for write.

[bit0] RC: Repeat code detection flag bit

bit	Description
"0" is read	Repeat code not detected
"1" is read	Repeat code detected
"0" is written	This flag will be cleared
"1" is written	No effect

Note:

- *Repeat code is detected only in NEC/Association for Electric Home Appliances mode.*

5.10 Repeat Code Detection Width Setting Register (RCRHW)

This register is used to set the detection width used for determining a repeat code.

bit	7	0
Field	RCRHW	
Attribute	R/W	
Initial Value	0x00	

[bit7:0] RCRHW: Repeat code detection width setting bits

These bits are used to set the detection width for a repeat code.

If a signal width with RCRHW < "H" width < RCSHW is received while waiting for a start bit or a repeat code, it will be detected as a repeat code.

A value to be set to this register must be RCRHW < RCSHW.

Note:

- *Repeat code is detected only in NEC/Association for Electric Home Appliances mode.*

5.11 Data Bit Width Violation Interrupt Control Register (RCLE)

This register controls maximum/minimum data bit width violation.

bit	7	6	5	4	3	2	1	0
Field	LELIE	LESIE	LELE	LESE	EPE	Reserved	LEL	LES
Attribute	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Initial Value	0	0	0	0	0		0	0

[bit7] LELIE: Maximum data bit width violation interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit6] LESIE: Minimum data bit width violation interrupt enable bit

bit	Description
0	Interrupt disabled
1	Interrupt enabled

[bit5] LELE: Maximum data bit width violation detection enable bit

bit	Description
0	Maximum data bit width violation detection disabled
1	Maximum data bit width violation detection enabled

[bit4] LESE: Minimum data bit width violation detection enable bit

bit	Description
0	Minimum data bit width violation detection disabled
1	Minimum data bit width violation detection enabled

[bit3] EPE: Error pulse output enable bit

bit	Description
0	Output disabled
1	Output enabled

If a minimum data bit width violation is detected when EPE="1", "L" pulses at 116 through 120 cycles will be output.

[bit2] Reserved: Reserved bit

"0" is always read.
Set "0" for write.

[bit1] LEL: Maximum data bit width violation detection flag bit

bit	Description
"0" is read	Maximum data bit width violation has not been detected
"1" is read	Maximum data bit width violation has been detected
"0" is written	This flag will be cleared
"1" is written	No effect on operation

[bit0] LES: Minimum data bit width violation detection flag bit

bit	Description
"0" is read	Minimum data bit width violation has not been detected
"1" is read	Minimum data bit width violation has been detected
"0" is written	This flag will be cleared
"1" is written	No effect on operation

Note:

- *Maximum/minimum data bit width violation is detected only in HDMI-CEC mode.*

5.12 Maximum Data Bit Width Setting Register (RCLELW)

This register is used to set a maximum data bit width.

bit	7	0
Field	RCLELW	
Attribute	R/W	
Initial Value	0x00	

[bit7:0] RCLELW: Maximum data bit width setting bits

These bits are used to set a maximum data bit width.

If a data bit with a width more than RCLELW is received, it will be detected as a maximum data bit width violation.

Note:

- *Maximum data bit width violation is detected only in HDMI-CEC mode.*

5.13 Minimum Data Bit Width Setting Register (RCLESW)

This register is used to set a minimum data bit width.

bit	7	0
Field	RCLESW	
Attribute	R/W	
Initial Value	0x00	

[bit7:0] RCLESW: Minimum data bit width setting bits

These bits are used to set a minimum data bit width.

If a data bit with a width less than RCLESW is received, it will be detected as a minimum data bit width violation.

Note:

- *Minimum data bit width violation is detected only in HDMI-CEC mode.*

CHAPTER 6-3: CEC Transmission

Functions and operations of CEC (Consumer Electronics Control) transmission are as follows.

1. Overview of CEC Transmission
2. Block Diagram of CEC Transmitting Circuit
3. CEC Transmission Interrupts
4. CEC Transmission Registers
5. CEC Transmission Operations
6. CEC Transmission Register Set

CODE: FIP007-E01-01

1. Overview of CEC Transmission

CEC signals standardized by HDMI (High Definition Multimedia Interface) are transmitted. The outline of transmission specification is as follows.

Automatic header transmission
Signal free is recognized to automatically transmit a header block.

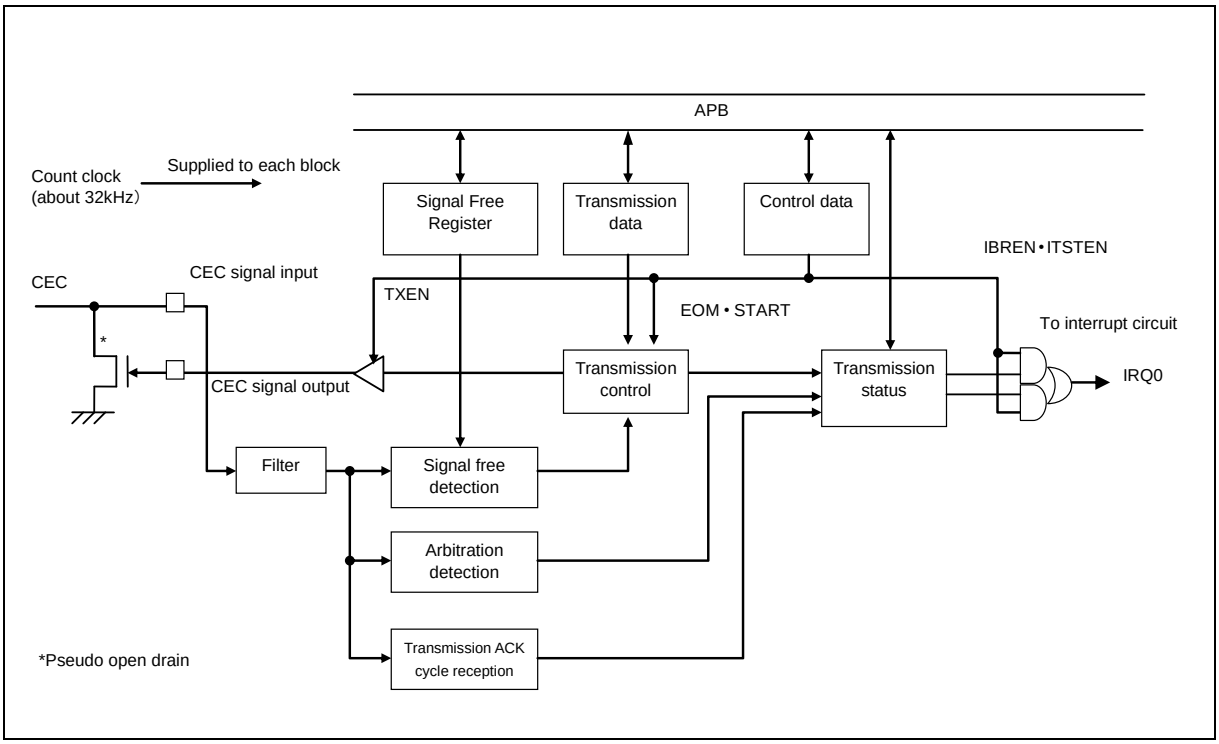
Bus error detection
Arbitration lost is recognized to generate a status interrupt.

Data transmission
- Setting 1 byte data automatically generate START, EOM and ACK to output CEC transmission.
- After 1 block (1 byte data, EMO and ACK) is transmitted, a transmission status interrupt is generated.

2. Block Diagram of CEC Transmitting Circuit

Figure 2-1 shows the block diagram of CEC transmitting circuit.

Figure 2-1 Block Diagram of CEC Transmitting Circuit



3. CEC Transmission Interrupts

A table summarizing interrupt request flags, interrupt enable bits and interrupt factors for CEC transmission is shown as follows.

Interrupt control bits and interrupt factors

Interrupt control bits and interrupt factors are shown in Table 3-1.

Table 3-1 Interrupt Control Bits and Interrupt Factors in each Mode

Transmission status (TXSTS)	Transmission control (TXCTRL)	Interrupt factor	Interrupt factor output signal
Interrupt request flag bit	Interrupt request enable bit		
ITST: bit4	ITSTEN: bit4	Transmission status detected	IRQ0
IBR: bit5	IBREN: bit5	Bus error detected	

4. CEC Transmission Registers

CEC transmission registers are as follows.

CEC Transmission registers

Table 4-1 CEC Transmission Registers

Abbreviated Register Name	Register Name	Reference
TXCTRL	Transmission Control Register	6.1
TXDATA	Transmission Data Register	6.2
TXSTS	Transmission Status Register	6.3
SFREE	Signal Free Time Setting Register	6.4

5. CEC Transmission Operations

Operations of CEC transmission are explained as follows.

5.1. CEC Transmission Operations
5.2. Interrupt Factors and Timing Chart
5.3. Arbitration Lost Detection
5.4. Signal Free Detection
5.5. Filtering
5.6. CEC Transmission Operations Flow

5.1 CEC Transmission Operations

Basic operations for transmission are explained as follows.

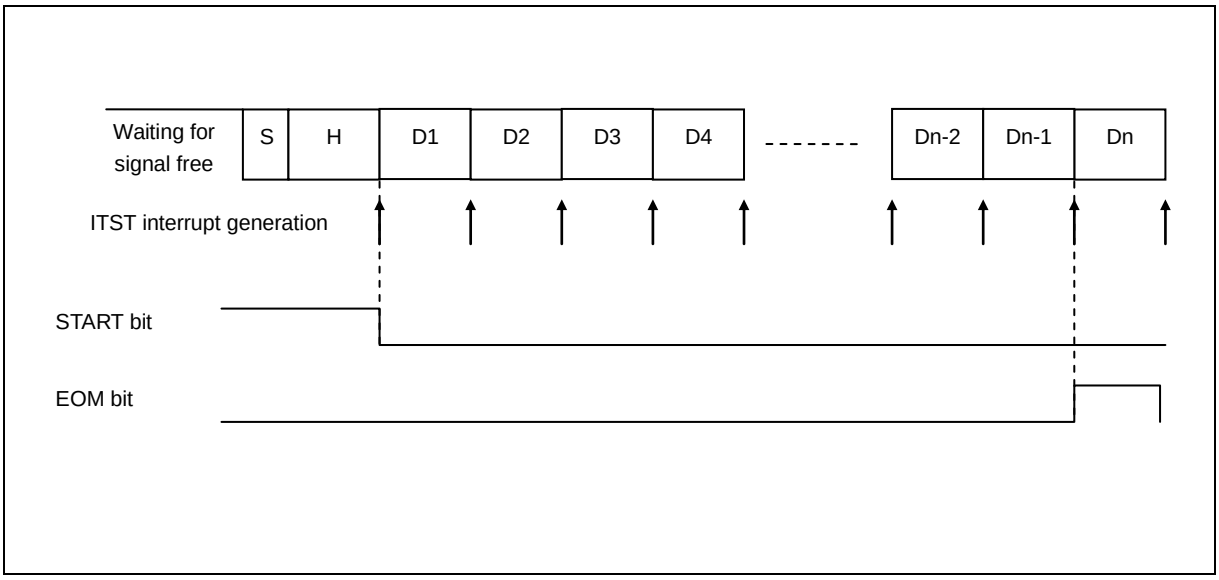
Basic operations

Basic operations are as follows:

- First set count clock for CEC from reception side.
- Next make various transmission setups and write transmitting data to TXDATA register to wait until signal free is detected. When signal free is detected, a start bit will automatically be transmitted.
- After the start bit is transmitted, 1 byte data set in the TXDATA register, data set in the EOM setting bits and ACK bit are automatically transmitted.
- As ITST bit interrupt of TXSTS register will be generated after the ACK bit is automatically transmitted. If the ACK cycle value is correct, make various transmission setups and write transmitting data for next transmission.
- Continue the transmission with the EOM at "1" until the complete transmissions end.

The basic operation timing for CEC transmission is shown in Figure 5-1.

Figure 5-1 Basic Operation timing Chart for CEC transmission



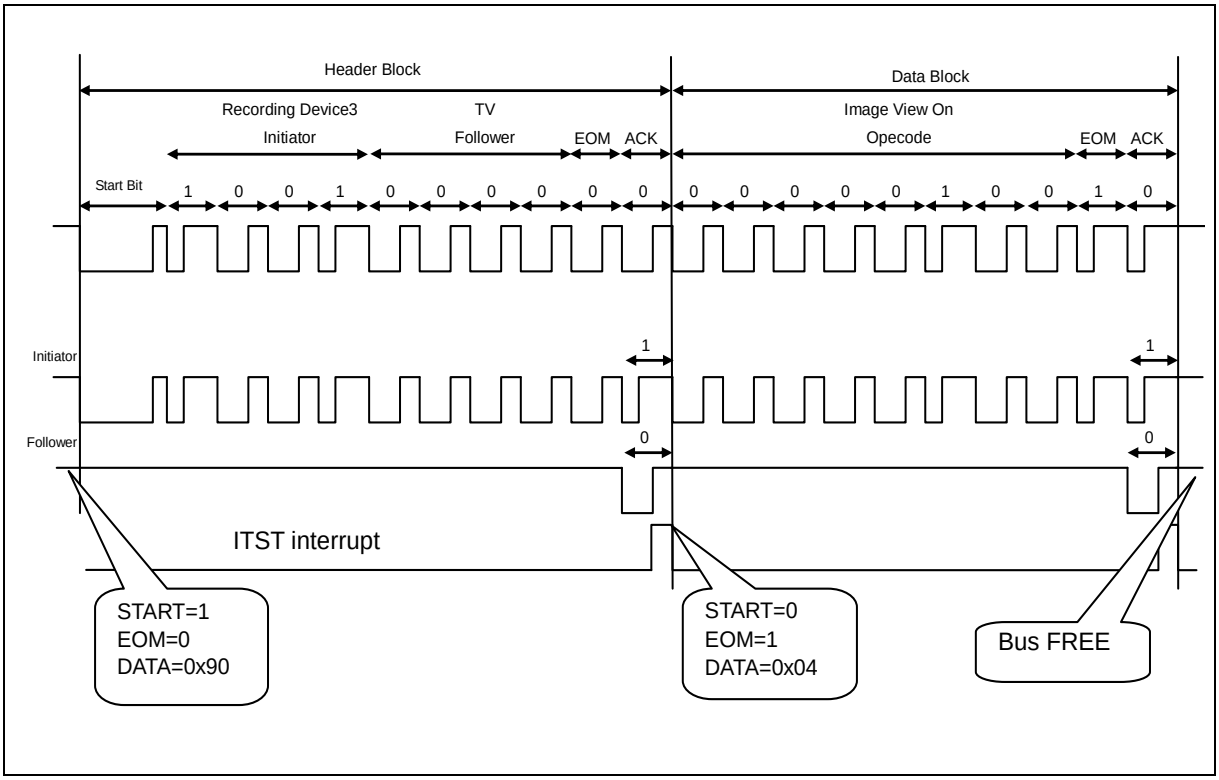
5.2 Interrupt Factors and Timing Chart

Interrupt factors and timing chart are as follows.

Interrupt Factors and Timing Chart

Figure 5-2 shows a transmission for a header block and a single data block in the ITST interrupt factors and timing chart.

Figure 5-2 Interrupt Factors and Timing Chart for CEC Transmission



5.3 Arbitration Lost Detection

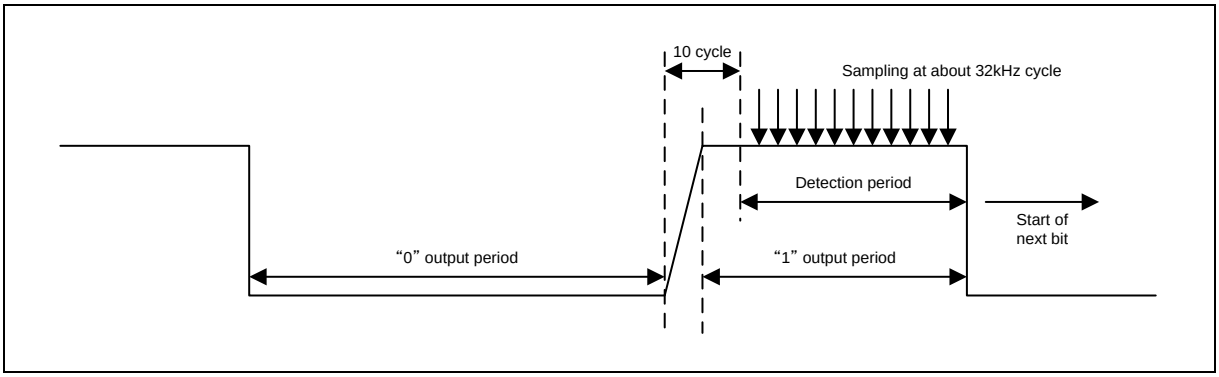
Arbitration lost detection is as follows.

How to detect arbitration lost

Figure 5-3 shows how to detect arbitration lost.

Data on the bus is sampled with about 32 kHz cycle per bit during the following detection period and compared to the transmission output. If any difference is detected, an arbitration lost will be detected. If the arbitration lost is detected, IBR of the TXSTS register becomes "1".

Figure 5-3 Arbitration Lost Detection Period

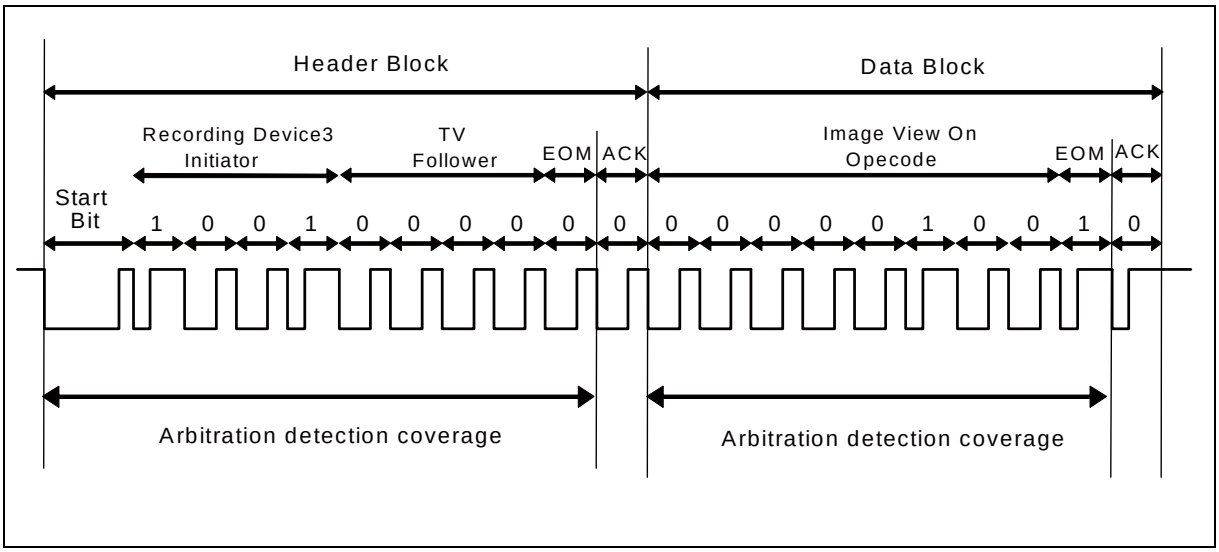


Detection coverage of arbitration lost

Figure 5-4 shows the detection coverage of arbitration lost.

The detection coverage becomes to the EOM during each block transfer excluding ACK cycle.

Figure 5-4 Arbitration Lost Detection Coverage



5.4 Signal Free Detection

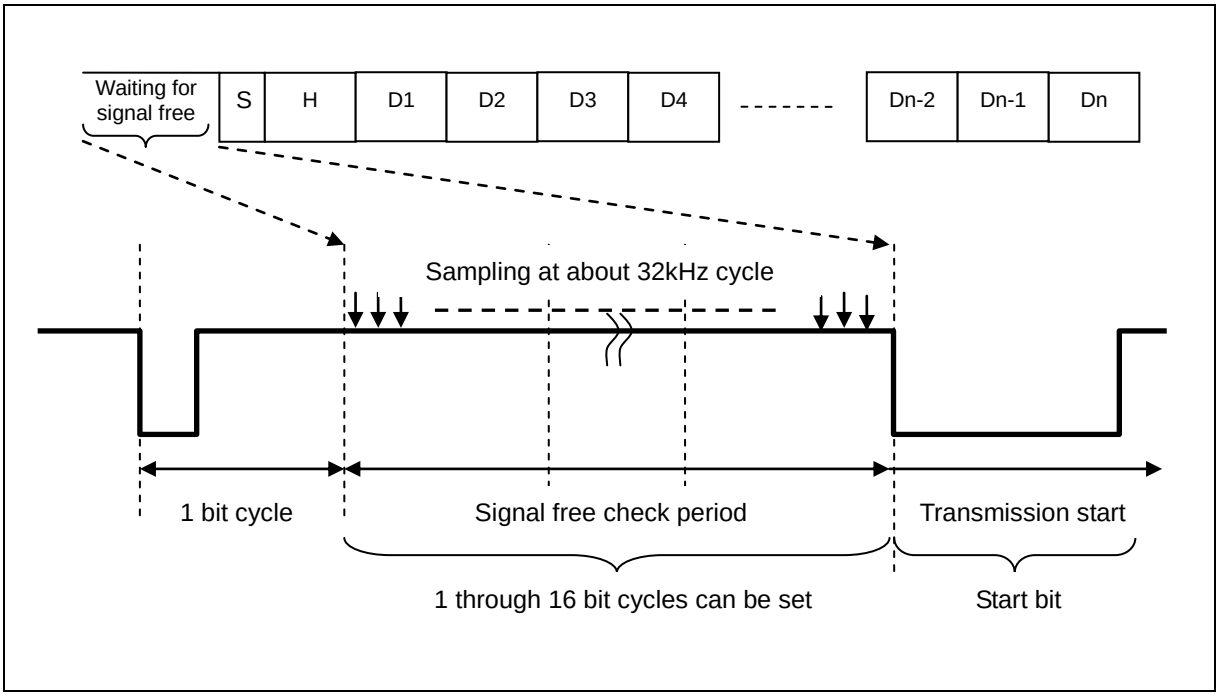
Signal free detection is as follows.

How to detect signal free

Figure 5-5 shows signal free detection.

If no change is found on the CEC bus during the cycles set in the SFREE register after the previous frame end, it becomes signal free detection state.

Figure 5-5 Signal Free Detection



5.5 Filtering

Filtering CEC signal input of transmission side is described as follows.

Filtering CEC signals

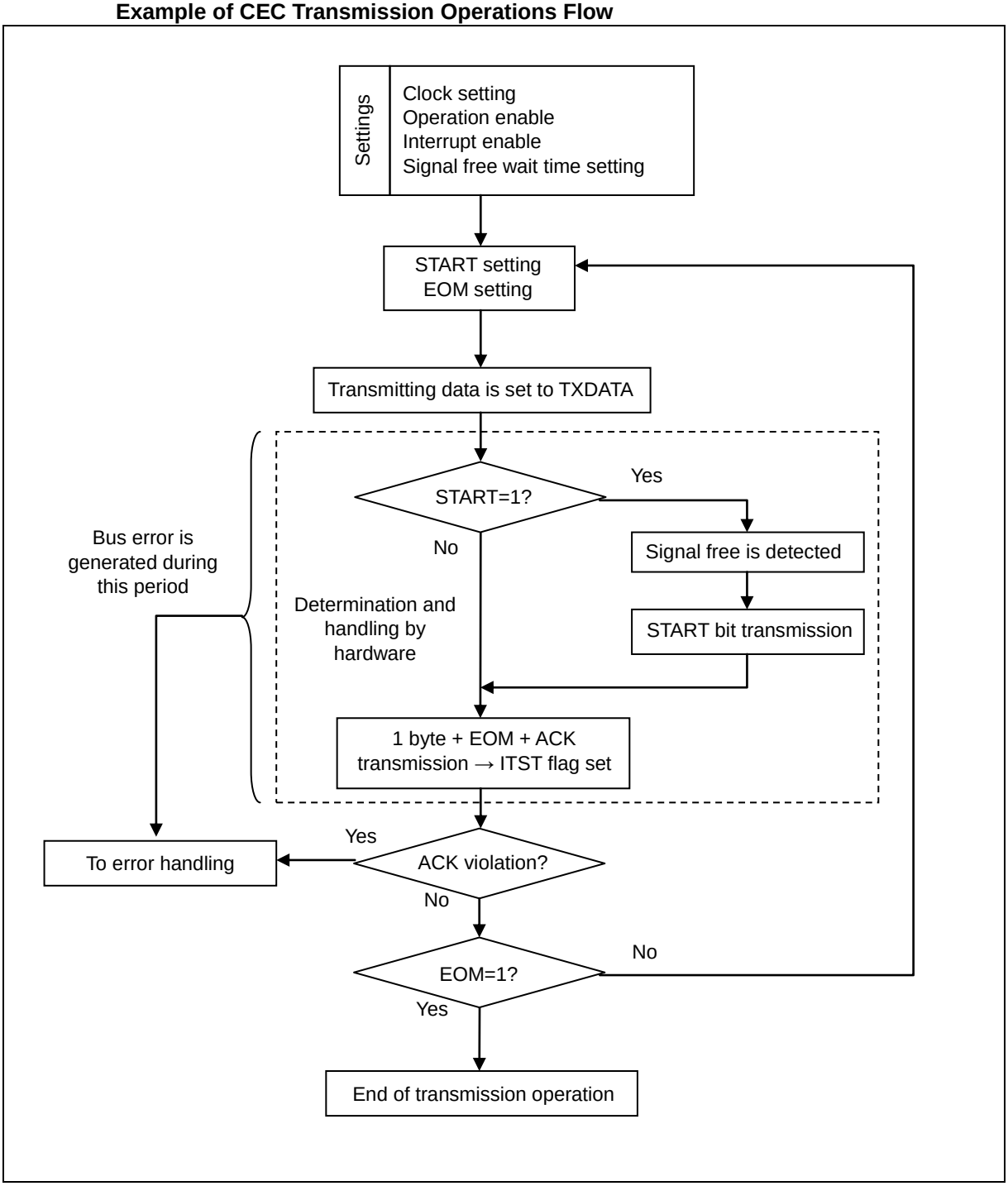
If a CEC signal input is changed within a width less than 2 count clocks, it is determined as noise and the signal will be removed.

An input changed within a width more than 2 count clocks is determined as CEC signal and passes through the filter.

5.6 CEC Transmission Operations Flow

CEC transmission operations flow is described as follows.

Example of CEC Transmission Operations Flow



6. CEC Transmission Register Set

All of CEC transmission registers is explained as follows.

6.1. Transmission Control Register (TXCTRL)

6.2. Transmission Data Register (TXDATA)

6.3. Transmission Status Register (TXSTS)

6.4. Signal Free Time Setting Register (SFREE)

6.1 Transmission Control Register (TXCTRL)

Transmission Control Register (TXCTRL) controls CEC transmission.

bit	7	6	5	4	3	2	1	0
Field	Reserved		IBREN	ITSTEN	EOM	START	Reserved	TXEN
Attribute	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	00		0	0	0	0	0	0

[bit7:6] Reserved: Reserved bits

"0" is read.

Set "0" to these bits for write.

[bit5] IBREN: Bus error detection interrupt enable bit

- This bit controls the interrupt request from bit5, the IBR bit in the TXSTS register.
- When the IBREN bit is enabled and bit5, the IBR bit in the TXSTS register is set, an interrupt request will be generated to the CPU.

bit	Description
0	Disables interrupt request
1	Enables interrupt request

[bit4] ITSTEN: transmission status interrupt enable bit

- This bit controls the interrupt request from bit4, the ITST bit in the TXSTS register.
- When the ITSTEN bit is enabled and bit4, the ITST bit in the TXSTS register is set, an interrupt request will be generated to the CPU.

bit	Description
0	Disables interrupt request
1	Enables interrupt request

[bit3] EOM: EOM setting bit

- This controls EOM transmission bit.
- Combination with the START bit will select block transmission.

bit	Description
0	Outputs EOM0
1	Outputs EOM1

[bit2] START: START setting bit

- This bit sets a header block transmission which adds the START bit to transmitting data.
- Combination with the EOM bit will select block transmission.

bit	Description
0	START bit transmission invalid
1	START bit transmission valid

EOM and START setups make CEC transmission to the following block transmission.

EOM bit	START=1	START=0
0	Header block transmission (beginning of frame)	Data block (with subsequent block)
1	Header block transmission (Polling Message)	Final data block (end of frame)

[bit1] Reserved: Reserved bit

"0" is read.
Set "0" to this bit for write.

[bit0] TXEN: Transmission operation enable bit

- This bit controls CEC transmission operations.
- When the TXEN bit it is changed to disable, automatic clearing for each bit of the status register will occur.

bit	Description
0	CEC transmission operation disabled
1	CEC transmission operation enabled

Note:

- *When "0" is set to the TXEN bit, outputs will immediately be stopped. Incorrect wave form may be output for the CEC signal at that time.*

6.2 Transmission Data Register (TXDATA)

Transmission Data Register (TXDATA) is used to set up transmission data.

bit	7	0
Field	TXDATA[7:0]	
Attribute	R/W	
Initial Value	0x00	

When a value is set to the TXDATA register, one of the following CEC transmissions will be started depending on the condition.

If the following conditions are met, a header block transmission will automatically be started.

- TXEN=1.
- START=1.
- IDLE is detected on the CEC bus during a period set in the SFREE register.

Note:

- *When you set a value to the TXDATA register, if IDLE for a period set in the SFREE register has been detected, a header block transmission will be started immediately after setup to the TXDATA register.*

If the following conditions are met, a data block transmission will immediately be started.

- TXEN=1.
- START=0.

6.3 Transmission Status Register (TXSTS)

Transmission Status Register (TXSTS) is used to indicate transmission statuses.

bit	7	6	5	4	3	2	1	0
Field	Reserved		IBR	ITST	Reserved			ACKSV
Attribute	R/W		R/W	R/W	R/W			R
Initial Value	00		0	0	000			0

[bit7:6] Reserved: Reserved bits

"0" is read.

Set "0" to these bits for write.

[bit5] IBR: Bus error detection interrupt request bit

- When arbitration lost is detected, the IBR bit is set to "1".
- The IBR bit is cleared by writing "0".
- Writing "1" to the IBR bit does not effect to the bit value.
- Read value by read-modify-write operation becomes "1" independent of the bit value.

bit	Description
0	Clears interrupt factor
1	Detects interrupt factor

Notes:

- *When "1" is automatically set to the IBR bit, if it is cleared at the same time by writing "0", the clearing will be ignored and "1" will be set.*
- *Be sure to write "0" while the IBR bit is "1". It may be cleared not knowing it will be automatically set to "1".*
- *If a line error signal is detected, the IBR bit will also be set to "1" as a bus error is detected.*

[bit4] ITST: Transmission status interrupt request bit

- When communication of a status bit at 10 bit in each block transfer is completed, the ITST bit will be set to "1".
- The ITST bit is cleared by writing "0".
- Writing "1" to the ITST bit does not effect the bit value.
- Read value by read-modify-write operation becomes "1" independent of the bit value.

bit	Description
0	Clears interrupt factor
1	Detects interrupt factor

Notes:

- *When "1" is attempted to automatically set to the ITST bit, if it is cleared at the same time by writing "0", the clearing will be ignored and "1" will be set.*
- *Be sure to write "0" while the ITST bit is "1". It may be cleared not knowing it will be automatically set to "1".*

[bit3:1] Reserved: Reserved bits

"0" is read.

Set "0" to these bits for write.

[bit0] ACKSV: ACK cycle value bit

- This bit indicates received data values in ACK cycle at 10 bit in each block transfer.
- This bit is updated when the ITST bit is changed from "0" to "1".
- Writing "1" to the ACKSV bit does not effect to the bit value.

bit	Description
0	"0" is received in ACK cycle
1	"1" is received in ACK cycle

6.4 Signal Free Time Setting Register (SFREE)

Signal Free Time Setting Register (SFREE) is used to set a signal free time checked before starting transmission.

bit	7	6	5	4	3	2	1	0
Field	Reserved				SFREE[3:0]			
Attribute	R/W				R/W			
Initial Value	0000				0000			

[bit7:4] Reserved: Reserved bits

"0" is read.

Set "0" to these bits for write.

[bit3:0] SFREE[3:0]: Signal free time setting bits

- These bits are used to set a time to check free state on the CEC bus before starting transmission.
- After no communication for bit cycle set on the CEC bus is found, transmission operation will be started.

bit3:0	Description
0000	(Set value + 1) cycle Ex1) 0000: 1bit cycle Ex2) 0111:8bit cycle Ex3) 1000: 9bit cycle Ex3) 1111:16bit cycle
0001	
...	
1110	
1111	

CHAPTER 7-1: I²S Clock Generation

This chapter explains the I²S clock generation.

1. Overview
2. Configuration and Block Diagram
3. Explanation of Operation
4. Setup Procedure Example
5. Register List
6. Usage Precautions

CODE: 9BFI²SETHERPLL_FM4-E01.0

1. Overview

This section provides an overview of the I^{2}S clock generation.

The I^{2}S clock is used by I^{2}S macro for communication.

The I^{2}S clock can be generated by using PLL for I2S(hereinafter I^{2}S-PLL) from 19.2MHz external main clock(hereinafter CLKMO). In addition, output clock from I^{2}S-PLL can be output as external output I2SMCLK(hereinafter I2SMCLKO).

Moreover, external input I2SMCLK(hereinafter I2SMCLKI) can be supplied to I2S macro as I2S clock.

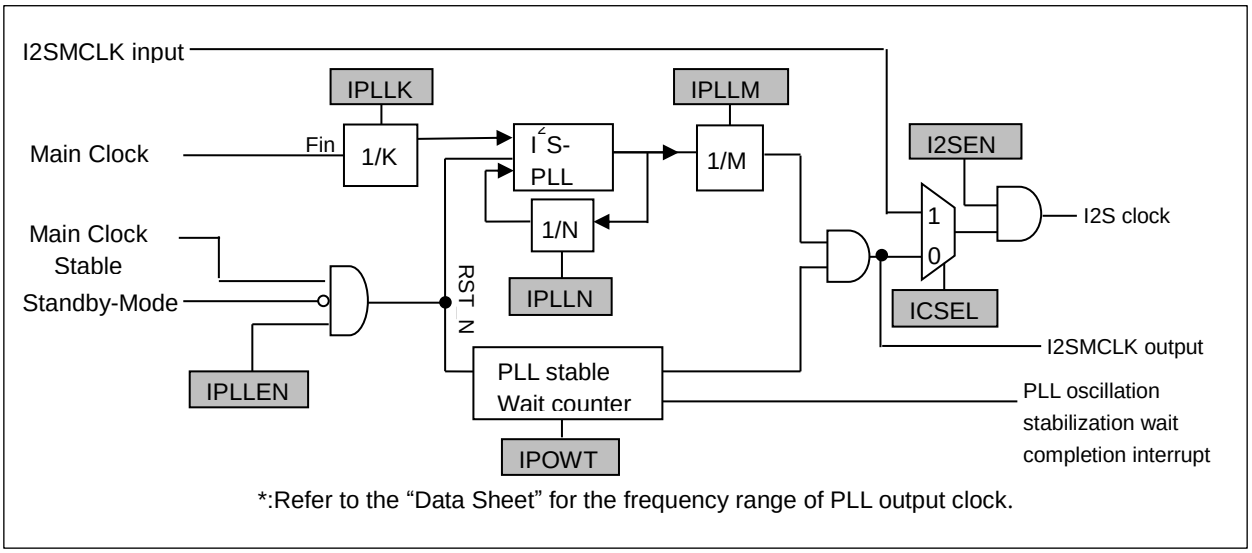
The I^{2}S clock generation unit is responsible for the following functions:

- Enables or stops output of the I^{2}S clock.
- Selects the I^{2}S clock.
- Enables or stops oscillation of I^{2}S-PLL.
- Selects the input clock of I^{2}S-PLL.
- Sets the input clock frequency division of I^{2}S-PLL.
- Sets the output clock multiplication of I^{2}S-PLL.
- Sets the stabilization wait time of I^{2}S-PLL.
- Stops the I^{2}S clock in standby mode.

2. Configuration and Block Diagram

This section explains the configuration and block diagram of the I²S clock generation unit.

Figure 2-1 Block diagram of I²S clock generation unit



I²S-PLL Control Register (IPLLEN)

The control register can enable I²S-PLL oscillation.

I²S-PLL

- Frequency division setting register (IPLLK, IPLLN, IPLLM)

  To generate I²S clock, the settings of K frequency division, N frequency division and M frequency division are required.

  For the specification range of the I²S-PLL input clock frequency, output clock frequency, and multiplier (N division setting value), refer to the PLL use conditions of "PLL input clock frequency", "PLL macro oscillation clock frequency", and "PLL multiplier" in "Data Sheet" of the product used.

- Oscillation stabilization wait time setting (IPOWT)

  Oscillation stabilization wait time for I²S-PLL can be specified.

Output clock

- Output Clock Select bit (ICSEL)

  Can be selected from I2SMCLKI or I²S-PLL output clock.

- PLL Clock Output Enable bit (I2SEN))

  Can set the I²S clock and I2SMCLKO output enable.

Standby mode setting

- The Standby-Mode signal shown in Figure 2-1 turns to be active in the following modes.
  The I²S clock stops in the following standby modes.
  - Stop mode
  - TIMER mode
- The Main Clock stable signal shown in Figure 2-1 is an oscillation stabilization signal for each mode.

3. Explanation of Operation

This section explains the operation of the I^{2}S clock generation unit.

Selecting the I^{2}S clock

The following two types of clocks can be selected for the I^{2}S clock.

■ I2SMCLKI

I2SMCLKI can be used directly as the I^{2}S clock. In this case, set I2SMCLKI frequency required for I^{2}S operation. Enable the output of the I2S clock after setting I2SMCLKI as an input.

■ Selecting the I^{2}S-PLL output clock

The I^{2}S-PLL output clock can be used as the source clock of I^{2}S clock.

Table 3-1 below shows the setting example of the division ratio.

Table 3-1 Example of PLL frequency division ratio settings

Fin[MHz]	K	N	M	PLL clock frequency [MHz]	I2S clock frequency [MHz]
19.2	2	32	25	307.2	12.288
19.2	3	32	25	204.8	8.192
19.2	2	32	75	307.2	4.096
19.2	2	40	34	384	11.294

Changing to standby mode

■ When changing to standby mode

Before changing to standby mode (STOP mode, or TIMER mode), set ICCR:I2SEN bit to "0" to stop the I^{2}S clock supply.

1. Set ICCR:I2SEN=0.
2. Read the ICCR Register to check that I2SEN is set to "0".
3. Changing to standby mode.

When returning from standby mode, set I2SEN bit to "1". The supply starts when the I^{2}S clock oscillation has been stabilized. Take either of the following actions to confirm whether or not the I^{2}S clock oscillation has been stabilized.

a) When I^{2}S-PLL is used

Check that IP_STR:IPRDY is "1", or use the I^{2}S-PLL oscillation stabilization wait interrupt.

b) When I2SMCLKI is used

After setting I2SMCLKI as input, supply the I^{2}S clock.

I^{2}S-PLL oscillation stabilization wait settings

■ Oscillation stabilization wait time for I^{2}S-PLL can be specified

After CLKMO oscillation has been stabilized, the oscillation stabilization wait time for I^{2}S-PLL begins to be counted.

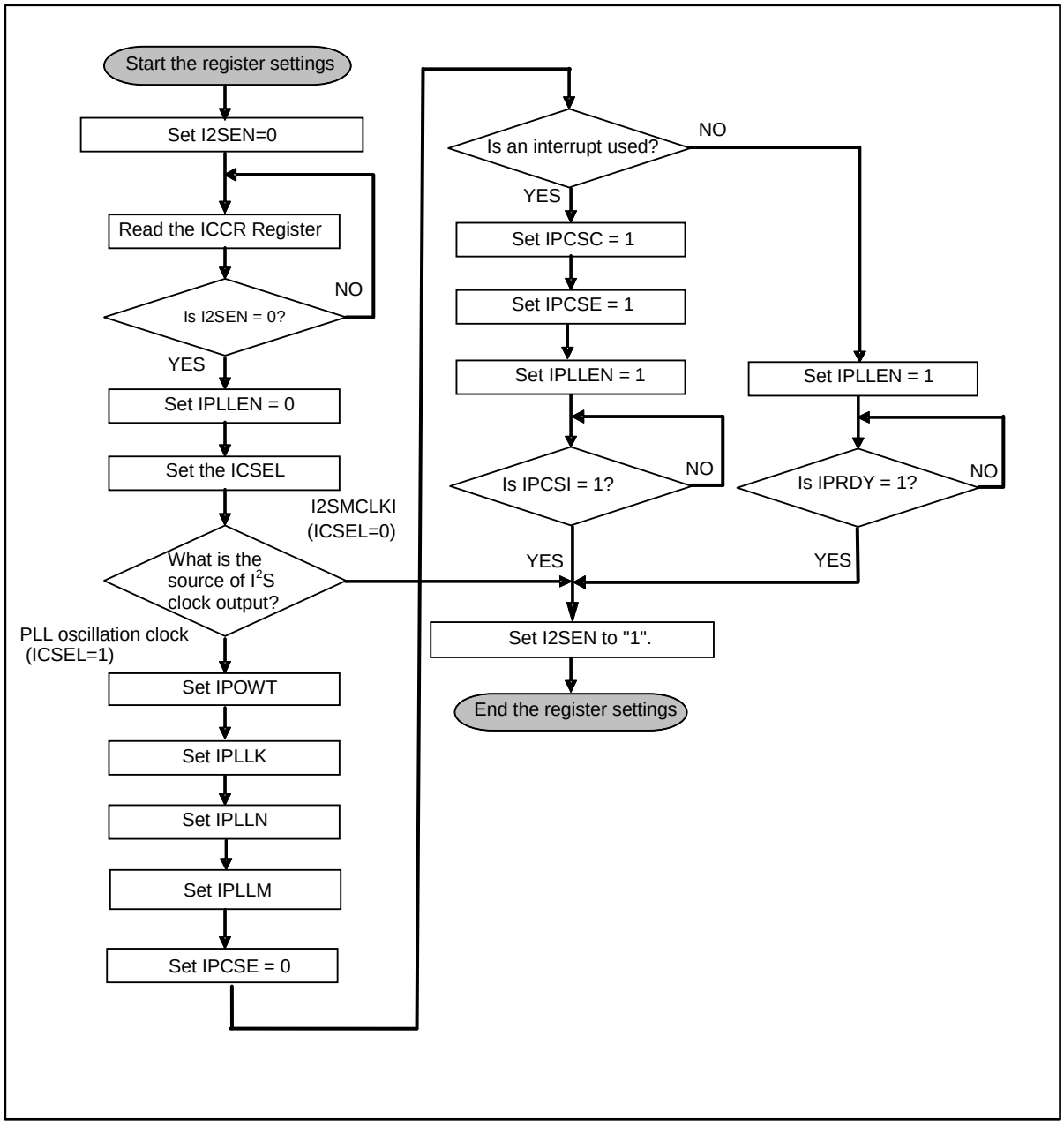
Before enabling the I^{2}S-PLL oscillation, configure the oscillation stabilization wait time for I^{2}S-PLL and the oscillation stabilization complete interrupt. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

4. Setup Procedure Example

This section explains an example of setting up the I^2S clock generation unit.

Figure 4-1 shows an example of setting up the I^2S clock.

Figure 4-1 I^2S clock generation procedure



5. Register List

This section explains the register list of the I^2S clock generation unit.

The register list of the I^2S clock generation unit

Abbreviation	Register name	Reference
ICCR	I^2S Clock Control Register	5.1
IPCR1	I^2S-PLL Control Register 1	5.2
IPCR2	I^2S-PLL Control Register 2	5.3
IPCR3	I^2S-PLL Control Register 3	5.4
IPCR4	I^2S-PLL Control Register 4	5.5
IPCR5	I^2S-PLL Control Register 5	5.6
IP_STR	I^2S-PLL Status Register	5.7
IPINT_ENR	I^2S-PLL Interrupt factor Enable Register	5.8
IPINT_STR	I^2S-PLL Interrupt factor Status Register	5.9
IPINT_CLR	I^2S-PLL Interrupt factor Clear Register	5.10

5.1 I²S Clock Control Register (ICCR)

The ICCR selects the I²S clock and enables/disables the I²S clock output.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						ICSEL	ICEN0
Attribute	-						R/W	R/W
Initial value	-						0	0

Register functions

[bit7:2] Reserved: Reserved bits

"0b000000" is read from these bits.

Set these bits to "0b000000" when writing.

[bit1] ICSEL: I²S clock selection bit

bit	Description
0	The clock divided by M of oscillation output from I2S-PLL macro. [Initial value]
1	I2SMCLKI

[bit0] ICEN: I²S clock output enable bit

bit	Description
0	Disables the I²S clock and I2SMCLKO output [Initial value]
1	Enables the I²S clock and I2SMCLKO clock output

Note:

- *This register is not initialized by software reset.*

5.2 I²S-PLL Control Register1 (IPCR1)

The IPCR1 sets I²S-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							IPLLEN
Attribute	-							R/W
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] IPLLEN: I²S-PLL oscillation enable bit

bit	Description
0	Stops I²S-PLL [Initial value]
1	Enables the I²S-PLL oscillation

Note:

- *This register is not initialized by software reset.*

5.3 I²S-PLL Control Register 2 (IPCR2)

The IPCR2 sets the oscillation stabilization wait time of I²S-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						IPOWT	
Attribute	-						R/W	
Initial value	-						000	

Register functions

[bit7:3] Reserved: Reserved bits

"0b00000" is read from these bits.

Set these bits to "0b00000" when writing.

[bit2:0] IPOWT: I²S-PLL oscillation stabilization wait time setting bits

bit2	bit1	bit0	Description
0	0	0	2^9/Fin : Approx. 26 μs * [Initial value]
0	0	1	2^{10}/Fin : Approx. 53 μs *
0	1	0	2^{11}/Fin : Approx. 106 μs *
0	1	1	2^{12}/Fin : Approx. 213 μs *
1	0	0	2^{13}/Fin : Approx. 426 μs *
1	0	1	2^{14}/Fin : Approx. 853 μs *
1	1	0	2^{15}/Fin : Approx. 1.70 ms *
1	1	1	2^{16}/Fin : Approx. 3.41 ms *

* : When F_{in} = 19.2 MHz

Notes:

- *Fin is the clock (CLKMO) : 19.2MHz.*
- *This register is not initialized by software reset.*
- *Since the oscillation stabilization wait time for PLL macro differs by products, refer to the use conditions of "PLL oscillation stabilization wait time" in "Data Sheet" of the product used.*

5.4 I²S-PLL Control Register 3 (IPCR3)

The IPCR3 sets the frequency division ratio (K) of I²S-PLL macro.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved			IPLLK				
Attribute	-			R/W				
Initial value	-			00001				

Register functions

[bit7:5] Reserved: Reserved bits

"0b000" is read from these bits.

Set these bits to "0b000" when writing.

[bit4:0] IPLLK: Frequency division ratio (K) setting bits of the I²S-PLL clock

bit4:0	Description
00000	Divides the frequency by (IPLLK+1). (Example) IPLLK = "00001" => 1/2 frequency [Initial value]
00001	
•	
•	
11111	

Note:

- *This register is not initialized by software reset.*

5.5 I²S-PLL Control Register 4 (IPCR4)

The IPCR4 Register sets the frequency division ratio (N) of I²S-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved	IPLLN						
Attribute	-	R/W						
Initial value	-	0011111						

Register functions

[bit7] Reserved: Reserved bit

"0b0" is read from this bit.

Set this bit to "0b0" when writing.

[bit6:0] IPLLN: Frequency division ratio (N) setting bits of the I²S-PLL clock

bit6:0	Description
0000000	Setting is prohibited.
•	
0001011	
0001100	Divides the frequency by (IPLLN+1). (Example) IPLLN = "0011111" => 1/32 frequency [Initial value]
•	
•	
1100011	
1100100	Setting is prohibited.
•	
1111111	

Note:

- *This register is not initialized by software reset.*

5.6 I²S-PLL Control Register 5 (IPCR5)

The IPCR5 sets the frequency division ratio (M) of I²S-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved	IPLLM						
Attribute	-	R/W						
Initial value	-	0011000						

Register functions

[bit7] Reserved: Reserved bits

"0b0" is read from these bits.

Set these bits to "0b0" when writing.

[bit6:0] IPLLM: Frequency division ratio (M) setting bits of the I²S-PLL clock

bit6:0	Description
0000000	Setting is prohibited.
0000001	Divides the frequency by (IPLLM+1). (Example) IPLLM = "0011000" => 1/25 frequency [Initial value]
•	
•	
1111111	

Note:

- *This register is not initialized by software reset.*

5.7 I²S-PLL Status Register (IP_STR)

The IP_STR indicates the macro status of I²S-PLL.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							IPRDY
Attribute	-							R
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] IPRDY: I²S-PLL oscillation stabilization bit

bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stabilized state

Note:

- *This register is not initialized by software reset.*

5.8 I²S-PLL Interrupt Factor Enable Register (IPINT_ENR)

The IPINT_ENR enables/disables the I²S-PLL oscillation stabilization wait complete interrupt.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							IPCSE
Attribute	-							R/W
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] IPCSE: I²S-PLL oscillation stabilization wait complete interrupt enable bit

bit	Description
0	Disables the interrupt [Initial value]
1	Enables the interrupt

5.9 I²S-PLL Interrupt Factor Status Register (IPINT_STR)

The IPINT_STR indicates the status of I²S-PLL oscillation stabilization wait interrupts.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							IPCSI
Attribute	-							R
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] IPCSI: I²S-PLL interrupt factor status bit

bit	Description
0	No interrupt has occurred [Initial value]
1	An interrupt has occurred

5.10 I²S-PLL Interrupt Factor Clear Register (IPINT_CLR)

The IPINT_CLR is used to clear the I²S-PLL interrupt factor.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							IPCSC
Attribute	-							W
Initial value	-							0

Register functions

[bit7:1] Reserved: Reserved bits

"0b0000000" is read from these bits.

Set these bits to "0b0000000" when writing.

[bit0] IPCSC: I²S-PLL oscillation stabilization interrupt factor clear bit

bit	Description
0	Disabled [Initial value]
1	Clears the I²S-PLL oscillation stabilization wait interrupt.

Note:

- *Writing "1" to IPCSC bit of this register to clear the IPINT_STR Register.*

6. Usage Precautions

This section explains the precautions for using the clock generation unit.

■ I^2S clock output setting and I^2S clock selection

Do not disable the I^2S clock output (I2SEN = 0) and select the I^2S clock (ICSEL) at the same time.
Be sure to disable the I^2S clock output before selecting the I^2S clock.

■ Setting the frequency division ratio of I^2S-PLL oscillation

When the PLL frequency division ratio is changed after stabilization of PLL oscillation, stop the PLL oscillation once, change the frequency division ratio, and then enable the PLL oscillation again.

■ Selecting I2SMCLKI

By writing "1" to the ICSEL bit, I2SMCLKI is selected as the I^2S clock.
The I2SMCLKI should be selected after . after setting I2SMCLKI as an input.

■ Setting the PLL oscillation stabilization wait time

Set the oscillation stabilization wait time with the PLL Oscillation Stabilization Wait Time Setting Register, and then enable PLL. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

■ Selecting the I^2S-PLL input clock

By writing "0" to the ICSEL bit, the I^2S-PLL oscillation clock is selected as the I^2S clock.

The following Table 6-1 shows relationship among the I^2S clock and ICSEL/IPLLEN.

Table 6-1 I^2S clock and register settings

		ICSEL	IPLLEN
When using I2SMCLKI		1	0
When using the PLL macro oscillation clock	Main oscillation clock input	0	1

■ Standby mode and the I^2S-PLL oscillation stabilization wait counter

If the mode changes to TIMER/STOP mode while waiting for the I^2S-PLL oscillation to stabilize, I^2S-PLL stops and the stabilization wait counter is cleared.

CHAPTER 7-2: I²S (Inter-IC Sound Bus) Interface

This document describes the functionality of the I²S interface, which is a serial audio interface

---

1. Overview of I2S Interface
2. Features of I2S Interface
3. Block Diagram of I2S Interface
4. I2S Interface Operation Description
5. List of I2S Interface Registers
6. Details of I2S Interface Registers
7. Overview of I2S Interface

CODE: 9BF002I2S-J01.0

1. Overview of I²S Interface

The I²S interface can operate as an interface for the transfer of both I²S and other serial PCM (pulse-code modulation) data by specifying the frame format.

2. Features of I²S Interface

The I²S Interface has the following features.

- Master or slave operation can be selected
- Transmit-only, receive-only, and simultaneous transmit/receive operations can be set
- Frames can be set to either a 1-subframe or 2-subframe configuration
- Each subframe can be set with up to 32 channels
- The number of channels in each subframe can be set independently
- The channel length (number of bits per channel) of each subframe can be set independently
- The word length in the channels of each subframe can be set independently
- Each channel in a given subframe can be set as enabled or disabled (*1)
- The word length can be set from 7 bits up to 32 bits
- The frequency of the frame sync signal is programmable
- One frame can be set with up to 3071 bits
- The width of the frame sync signal is programmable (1 bit or 1 channel length)
- The phase of the frame sync signal is programmable (0 bits or 1 bit delay)
- The polarity of the frame sync signal can be set
- The polarity of the serial bit clock is programmable
- The sampling point of the receive data is programmable
- The base clock divider of the serial bit clock in master mode can be selected (internal or external clock)
- The clock division ratio in master mode can be set
  I2SCK frequency=HCLK (or I2SMCLK) frequency/ 2 × CKRT[5:0] (*2)
  The clock division ratio can be set from 0 to 126 at a multiple of 2 (Bypassing the base clock divider when the division ratio is 0).

Allows data transfer to memory by DMA, interrupts, and polling

Note:

- *DMA transfer of I²S interface supports to only hardware transfer of DSTC.*

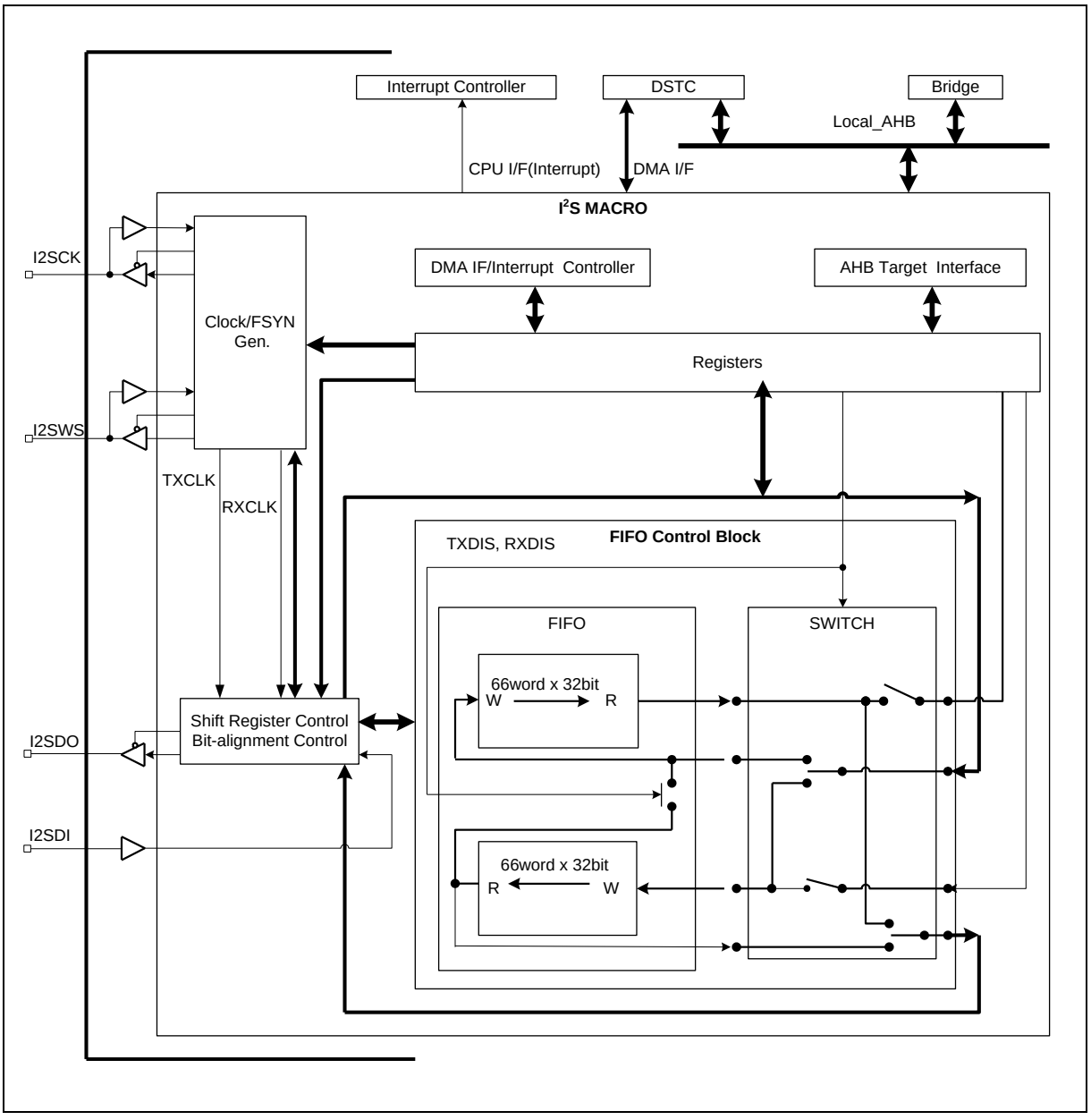
*1 : Data is not transmitted or received for channels that are set to disabled.

*2 : HCLK is the AHB bus clock connected by the I2S interface.

3. Block Diagram of I²S Interface

Figure 4.1-1 shows a block diagram of the I²S interface.

Figure 4.1-1 I²S Interface Block Diagram



4. I²S Interface Operation Description

Overview

I²S macro is a synchronous serial interface that enables specifying of full-duplex and multichannels.
It supports a wide range of frame formats through the use of register settings.
(For details, see the frame configurations in "4.1Clock and Frame Sync Signal".)

I²S can operate in master mode or slave mode. In master mode, the clock (I2SCK pin) and frame sync signal (I2SWS pin) are output to an external slave. In slave mode, the clock (I2SCK pin) and frame sync signal (I2SWS pin) are input from an external master.

In master mode, the I2SCK clock that is output can divide an external clock (I2SMCLK pin) or internal clock (selectable using the registers) for output. The frame sync signal can be generated in free run or burst mode (generated only when transmit data is available). For details, see "4.1Clock and Frame Sync Signal".

Transmit and receive FIFOs are installed internally. The FIFO depth changes based on whether it is in transmit-only mode (132 word × 32 bit configuration transmit FIFO), receive-only mode (132 word × 32 bit configuration receive FIFO), or simultaneous transmit/receive mode (66 word × 32 bit configuration transmit FIFO, 66 word × 32 bit configuration receive FIFO). For details, see "4.3Frame Configuration".

DMA, interrupts, and polling can be used to perform internal transfers between the transmit and receive FIFOs and memory.

4.1 Clock and Frame Sync Signal

Clock

1. The transmit clock TXCLK and receive clock RXCLK are output to the shift register control unit. TXCLK is the clock for the transmit serial shift register control unit, and RXCLK is the clock for the receive serial shift register control unit.
2. TXCLK and RXCLK are generated from I2SCK input when in slave mode (CNTREG: MSMD = 0). The polarity of I2SCK input can be selected by the CNTREG:CPOL bit.
3. In master mode (CNTREG : MSMD = 0), the internal clock (HCLK-AHB bus clock) or external clock (I2SMCLK input) is divided to generate TXCLK and RXCLK. Clock divisions operate when the OPRREG:START bit is set to "1" by the internal CPU. I2SCK output is stopped when the START bit is set to "0". The base clock divider (1: external, 0: internal) is selected by setting the CNTREG:ECKM bit. The polarity of the I2SCK that is output can be reversed by the CNTREG:CPOL bit. For details of the clock division ratio settings, see the explanation of CKRT[5:0] in "6.3Control Register (CNTREG)".

Frame Sync Signal

The frame sync signal generation cycle is determined by the free run mode (CNTREG : FRUN = 1), frame configuration, and OVERHEAD bits. The frame configuration is determined by the number of subframes, number of channels in the subframe, and the length of each channel.(See "4.2Transfer Start, Stop, and Error Operations".)
OVERHEAD bits are inserted as dummy bits after the last channel of the frame and is set when adjusting the frame rate.

In free run mode (FRUN=1) and master mode:
The frame sync signal is output at the frame rate determined by the frame configuration and OVHD.

In free run mode (FRUN=1) and slave mode:
The frame sync signal is input at the frame rate determined by the frame configuration and OVHD. If the signal is not input at the preset frame rate, the STATUS:FERR bit is set to "1".

In burst mode (FRUN=0) and master mode:
The frame sync signal is output only when transmit frame data is included in the transmit FIFO.

In burst mode (FRUN=0) and slave mode:
Frames are transmitted and received whenever a frame sync signal is input from an external source. The STATUS:FERR bit is set to "1" when the next frame sync signal is input even though the frame determined by the register setting for the frame sync signal has not ended.

For details of the frame sync signal settings (phase, polarity, pulse length, output mode, etc.), see "6.3. Control Register (CNTREG)".

4.2 Transfer Start, Stop, and Error Operations

Transmit-only enable mode

■ Transmit/receive settings (TXDIS=0, RXDIS=1)

Operation	Master mode (MSMD=1)	Slave mode (MSMD=0)
Start	Free run mode (FRUN=1): After the START bit is set to "1" and the TXENB bit is set to "1", output of the frame sync signal starts when the transmit FIFO is no longer empty. Then, the frame sync signal is output at the frame rate determined by the register settings. An empty frame is output if the transmit FIFO is empty when the frame sync signal is output. The serial data for an empty frame can be specified as "0" or "1" by using the register settings.	Free run mode (FRUN=1): The frame sync signal is input at the frame rate determined by the register settings. An empty frame is output if the transmit FIFO is empty when the frame sync signal is input while the START bit is "1" and the TXENB bit is "1". The serial data for an empty frame can be specified as "0" or "1" by using the register settings.
	Burst mode (FRUN=0): A frame sync signal is output if the transmit FIFO is not empty when the START bit is "1" and the TXENB bit is "1". The transmit FIFO status is continuously checked at the end of one frame output or when idle, and the frame sync signal is output when the transmit FIFO is not empty.	Burst mode (FRUN=0): One frame is output if the frame sync signal is input while the START bit is "1" and the TXENB bit is "1". An empty frame is output if the transmit FIFO is empty when the frame sync signal is input.
Stop	The transmit FIFO is empty if there is no data transfer to the transmit FIFO of the I2S interface from the internal memory when stopped.	
	When the START bit is held at "1": When TXENB is "1": The frame sync signal is output continuously in free run mode. An empty frame is output once the transmit FIFO becomes empty. The frame sync signal is not output once the transmit FIFO becomes empty when in burst mode. An empty frame bit is output to the serial data bus.	When the START bit is held at "1": When TXENB is "1": Empty frame data is output to the serial bus.
	When TXENB is "0": Because the transmit FIFO will become empty when "0" is written to TXENB, the data in the transmit FIFO is not sent when "0" is written to TXENB. The frame sync signal is output continuously in free run mode, but the serial bus is in the high-impedance state. The frame sync signal is not output in burst mode. The serial data bus is in a high-impedance state.	When TXENB is "0": "0" is written to TXENB. Because the transmit FIFO will become empty, any data in the transmit FIFO is not sent when "0" is written to TXENB. Data writing to the transmit FIFO and transmit frame detection are stopped. The serial data bus is in a high-impedance state.

Operation	Master mode (MSMD=1)	Slave mode (MSMD=0)
Stop	When the START bit is set to "0": "0" is written to the START bit. The transmit FIFO becomes empty. The clock supply to the serial control unit is stopped regardless of the TXENB setting, and the clock is not output even to an external unit. The frame sync signal output is also stopped. The serial data bus is in a high-impedance state.	When the START bit is set to "0": "0" is written to the START bit. The transmit FIFO becomes empty. Writing to the transmit FIFO and detection of the transmit frame sync signal is stopped regardless of the TXENB setting.
Error	If reading to the transmit FIFO occurs while the transmit FIFO is empty, an empty frame bit is output. For details of the setting conditions of the TXUDR0 and TXUDR1 bits of the STATUS register, see the explanation of the TXUDR0 and TXUDR1 bits in "6.10Status Register (STATUS)". TXOVR is set to "1" when writing to the transmit FIFO occurs while the transmit FIFO is full.	If reading to the transmit FIFO occurs while the transmit FIFO is empty, an empty frame bit is output. For details of the setting conditions of the TXUDR0 and TXUDR1 bits of the STATUS register, see the explanation of the TXUDR0 and TXUDR1 bits in "6.10Status Register (STATUS)". However, when START=1 and TXENB=1, TXUDR is not set to "1" for the empty frame that is output first. TXOVR is set to "1" when writing to the transmit FIFO occurs while the transmit FIFO is full. Free run mode: The FERR bit of the STATUS register is set to "1" when the frame sync signal is not input at the predetermined frame rate. Burst mode: The FERR bit of the STATUS register is set to "1" when the next frame sync signal is input even if transmission of a frame has not ended.

Note:

- *TXDIS and RXDIS are setting bits that enable or disable the transmit/receive function of the CNTREG register.*
- *START, TXENB, and RXENB are operation control bits of the OPRREG register.*
- *The empty frame bit is determined by the MSKB bit of the CNTREG register.*

Receive-only enable mode

■ Transmit/receive settings (TXDIS=1, RXDIS=0)

Operation	Master mode (MSMD=1)	Slave mode (MSMD=0)
Start	Free run mode (FRUN=1): After the START bit is set to "1" and the RXENB bit is set to "1", output of the frame sync signal starts when the receive FIFO is no longer full. Then, the frame sync signal is output continuously at the frame rate determined by the register settings.	Free run mode (FRUN=1): The frame sync signal is input at the frame rate determined by the register settings when the START bit is "1" and RXENB = "1". Frames are received whenever the frame sync signal is input.
	Burst mode (FRUN=0): After the START bit is set to "1" and the RXENB bit is set to "1", if the receive FIFO is not full, the frame sync signal is output, and frames are received. The frame sync signal is not output when the receive FIFO is full.	Burst mode (FRUN=0): Frames are received whenever a frame sync signal is input if the START bit is "1" and RXENG="1". The frame sync signal is input at a speed that does not exceed the frame rate in free run mode.
Stop	When stopped, frames are not imported from the serial bus even when the receive FIFO is empty if data transfer to the internal memory from the receive FIFO of the I2S interface is required.	
	When the START bit is held at "1": "0" is written to RXENB. The receive FIFO is empty. The frame sync signal is output continuously in free run mode, but frames are not received. In burst mode, frames are not received, and the frame sync signal is not output.	When the START bit is held at "1": The receive FIFO is emptied by writing "0" to RXENB. The frame sync signal that is input is ignored, and frames are not received.
	When the START bit is set to "0": "0" is written to the START bit. The receive FIFO is emptied. The clock supply to the serial control unit is stopped regardless of the RXENB setting, and the I2SCK to an external unit is also stopped.	When the START bit is set to "0": "0" is written to the START bit. The receive FIFO is emptied. The frame sync signal that is input is ignored, and frames are not received regardless of the RXENB setting.
Error	RXOVR of the STATUS register is set to "1" when writing to the receive FIFO occurs while the receive FIFO is full. RXUDR of the STATUS register is set to "1" when read access to the receive FIFO occurs while the receive FIFO is empty.	RXOVR of the STATUS register is set to "1" when writing to the receive FIFO occurs while the receive FIFO is full. RXUDR of the STATUS register is set to "1" when read access to the receive FIFO occurs while the receive FIFO is empty. Free run mode: The FERR bit of the STATUS register is set to "1" when the frame sync signal is not input at the frame rate determined by the register settings. Burst mode: The FERR bit of the STATUS register is set to "1" when the next frame sync signal is input while receiving another frame.

Note:

- *TXDIS and RXDIS are setting bits that enable or disable the transmit/receive function of the CNTREG register.*
  START, TXENB, and RXENB are operation control bits of the OPRREG register.

Transmit/receive simultaneous transfer enable mode

■ Transmit/receive settings (TXDIS=0, RXDIS=0)

Operation	Master mode (MSMD=1)	Slave mode (MSMD=0)
Start	Free run mode (FRUN=1): When START=1, TXENB=1, and RXENB=0, operation is identical to the transmit-only mode. When START=1, TXENB=0, and RXENB=1, operation is identical to the receive-only mode. When START=1, TXENB=1, and RXENB=1, output of the frame sync signal starts when the transmit FIFO is not empty and the receive FIFO is not full. Then, the frame sync signal is output at the frame rate determined by the register settings. An empty frame is output if the transmit FIFO is empty when the frame sync signal is output. The serial data for an empty frame can be specified as "0" or "1" by using the register settings. Frames are received whenever a frame sync signal is input.	Free run mode (FRUN=1): When START=1, TXENB=1, and RXENB=0, operation is identical to the transmit-only mode. When START=1, TXENB=0, and RXENB=1, operation is identical to the receive-only mode. When START=1, TXENB=1, and RXENB=1, the frame sync signal is input at the frame rate determined by the register settings. An empty frame is output if the transmit FIFO is empty when the frame sync signal is input. The serial data for an empty frame can be specified as "0" or "1" by using the register settings. Frames are received whenever a frame sync signal is input.
	Burst mode (FRUN=0): When START=1, TXENB=1, and RXENB=0, operation is identical to the transmit-only mode. When START=1, TXENB=0, and RXENB=1, operation is identical to the receive-only mode. When START=1, TXENB=1, and RXENB=1, the frame sync signal is output if the transmit FIFO is not empty and the receive FIFO is not full. After that, when the output of a frame is ended or when idle, the status of the receive FIFO is checked, and the frame sync signal is output and frames are transmitted and received if the receive FIFO is not full.	Burst mode (FRUN=0): A frame is transmitted or received whenever a frame sync signal is input if the START bit is "1". An empty frame is output if the transmit FIFO is empty when the frame sync signal is input.

Note:

- *TXDIS and RXDIS are setting bits that enable or disable the transmit/receive function of the CNTREG register.*
  START, TXENB, and RXENB are operation control bits of the OPRREG register.
- *The empty frame bit is determined by the MSKB bit of the CNTREG register.*

■ Transmit/receive settings (TXDIS=0, RXDIS=0)

Operation	Master mode (MSMD=1)	Slave mode (MSMD=0)
Stop	The stop states below are used. Transmit stop The transmit FIFO becomes empty if there is no data transfer to the transmit FIFO of the I2S interface from the internal memory. Receive stop This is the state where there is no need to perform data transfers to the internal memory from the receive FIFO of the I^2S interface.	
	When the START bit is held at "1": The frame sync signal is output continuously in free run mode. The frame sync signal is not output once the transmit FIFO becomes empty when in burst mode. Transmit stop: If held to TXENB=1, once the transmit FIFO becomes empty, an empty frame bit is output continuously. When set to TXENB=0, the transmit FIFO becomes empty, and the transmit serial data bus is set to high impedance. The data in the transmit FIFO is not sent when "0" is written to TXENB. Writing to the transmit FIFO is stopped when TXENB=0. Receive stop: "0" is written to RXENB. The receive FIFO is emptied, and the frame receive operation is stopped.	When the START bit is held at "1": Transmit stop: If held to TXENB=1, once the transmit FIFO becomes empty, an empty frame bit is output continuously. When set to TXENB=0, the transmit FIFO becomes empty, and the transmit serial data bus is set to high impedance. The data in the transmit FIFO is not sent when "0" is written to TXENB. Writing to the transmit FIFO is stopped when TXENB=0. Receive stop: "0" is written to RXENB. The receive FIFO is emptied, and the frame receive operation is stopped.
	When the START bit is set to "0": "0" is written to the START bit. The transmit and receive FIFO are emptied. The clock supply to the internal serial control unit is stopped regardless of the TXENB and RXENB states, and the I2SCK to an external unit is also stopped. The frame sync signal output is also stopped.	When the START bit is set to "0": "0" is written to the START bit. The transmit and receive FIFO are emptied. The transmit/receive operation is stopped regardless of the TXENB and RXENB states.

Operation	Master mode (MSMD=1)	Slave mode (MSMD=0)
Error	If reading to the transmit FIFO occurs while the transmit FIFO is empty, an empty frame bit is output. For details of the setting conditions of the TXUDR0 and TXUDR1 bits of the STATUS register, see the explanation of the TXUDR0 and TXUDR1 bits in "6.10Status Register (STATUS)". TXOVR is set to "1" when writing to the transmit FIFO occurs while the transmit FIFO is full. The RXUDR bit of the STATUS register is set to "1" when read access to the receive FIFO occurs while the receive FIFO is empty. The RXOVR bit of the STATUS register is set to "1" when writing to the receive FIFO occurs while the receive FIFO is full.	If reading to the transmit FIFO occurs while the transmit FIFO is empty, an empty frame bit is output. For details of the setting conditions of the TXUDR0 and TXUDR1 bits of the STATUS register, see the explanation of the TXUDR0 and TXUDR1 bits in "6.10Status Register (STATUS)". TXOVR is set to "1" when writing to the transmit FIFO occurs while the transmit FIFO is full. The RXUDR bit of the STATUS register is set to "1" when read access to the receive FIFO occurs while the receive FIFO is empty. The RXOVR bit of the STATUS register is set to "1" when writing to the receive FIFO occurs while the receive FIFO is full. Free run mode: The FERR bit of the STATUS register is set to "1" when the frame sync signal is not input at the frame rate determined by the register settings. Burst mode: The FERR bit of the STATUS register is set to "1" when the next frame sync signal is input while receiving another frame.

Note:

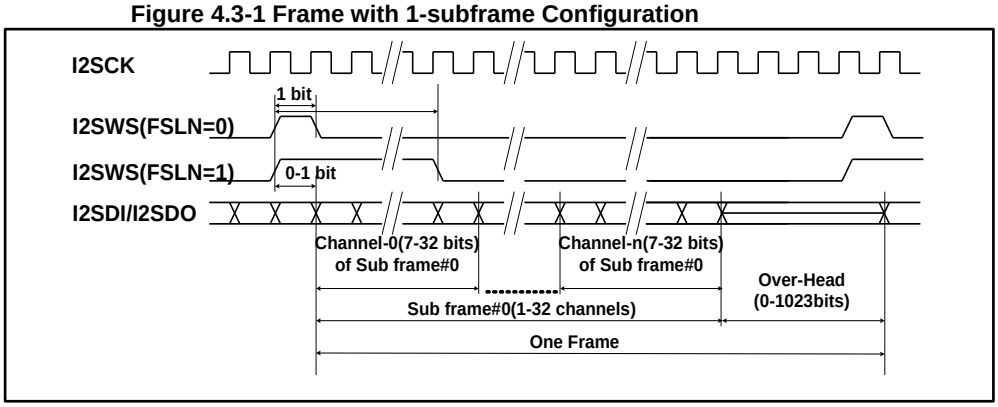
- *TXDIS and RXDIS are setting bits that enable or disable the transmit/receive function of the CNTREG register.*
  START, TXENB, and RXENB are operation control bits of the OPRREG register.
- *The empty frame bit is determined by the MSKB bit of the CNTREG register.*

4.3 Frame Configuration

The I^{2}S interface supports frame formats with multichannel configurations. Frames can be set with a 1-subframe or 2-subframe configuration, and the number of channels and word length for each subframe can be set independently. For I2S setting examples, see "7. Application Notes".

Subframe Configuration

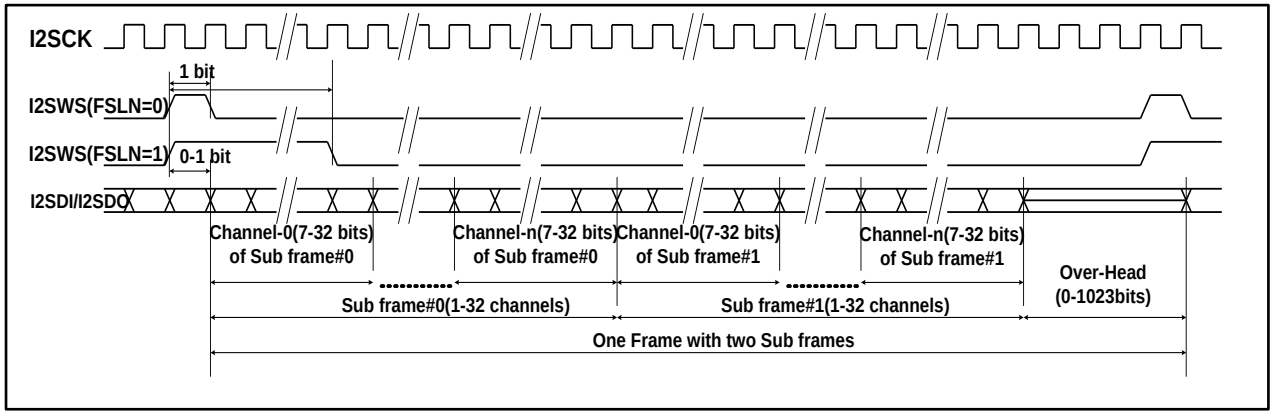
Figure 4.3-1 Frame with 1-subframe Configuration



1. A 1-subframe configuration is set when the SBFN bit of the CNTREG register is set to "0".
2. The number of channels when using a 1-subframe configuration is determined by the S0CHN bit of the MCR0REG register.
   Up to 32 channels can be set.
3. The bit length (word length) of each channel is determined by the S0WDL bit of the MCR0REG register.
4. The subframe channel starts from the number 0. Each channel can be enabled or disabled by the corresponding bit of the MCR1REG register. Data is not transmitted or received for channels that are disabled.
5. The OVHD bit of the CNTREG register can be set to enable insertion of dummy bits after a subframe. The number of insertable bits is from 0 to 1023 bits.
6. The I2SWS polarity is set by the FSPL bit of the CNTREG register.
7. The pulse width of I2SWS is set by the FSLN bit of the CNTREG register, and it can be set to 1 bit or 1 channel length.
8. The timing from the edge where I2SWS is enabled until the first bit of the frame can be set to 0 or 1 bit.
9. When using a 1-subframe configuration, the settings for the S1CHN and S1WDL bits of the MCR0REG register are ignored, and the the setting for the MC2REG register is ignored.

2-subframe Configuration

Figure 4.3-2 Frame with 2-subframe Configuration

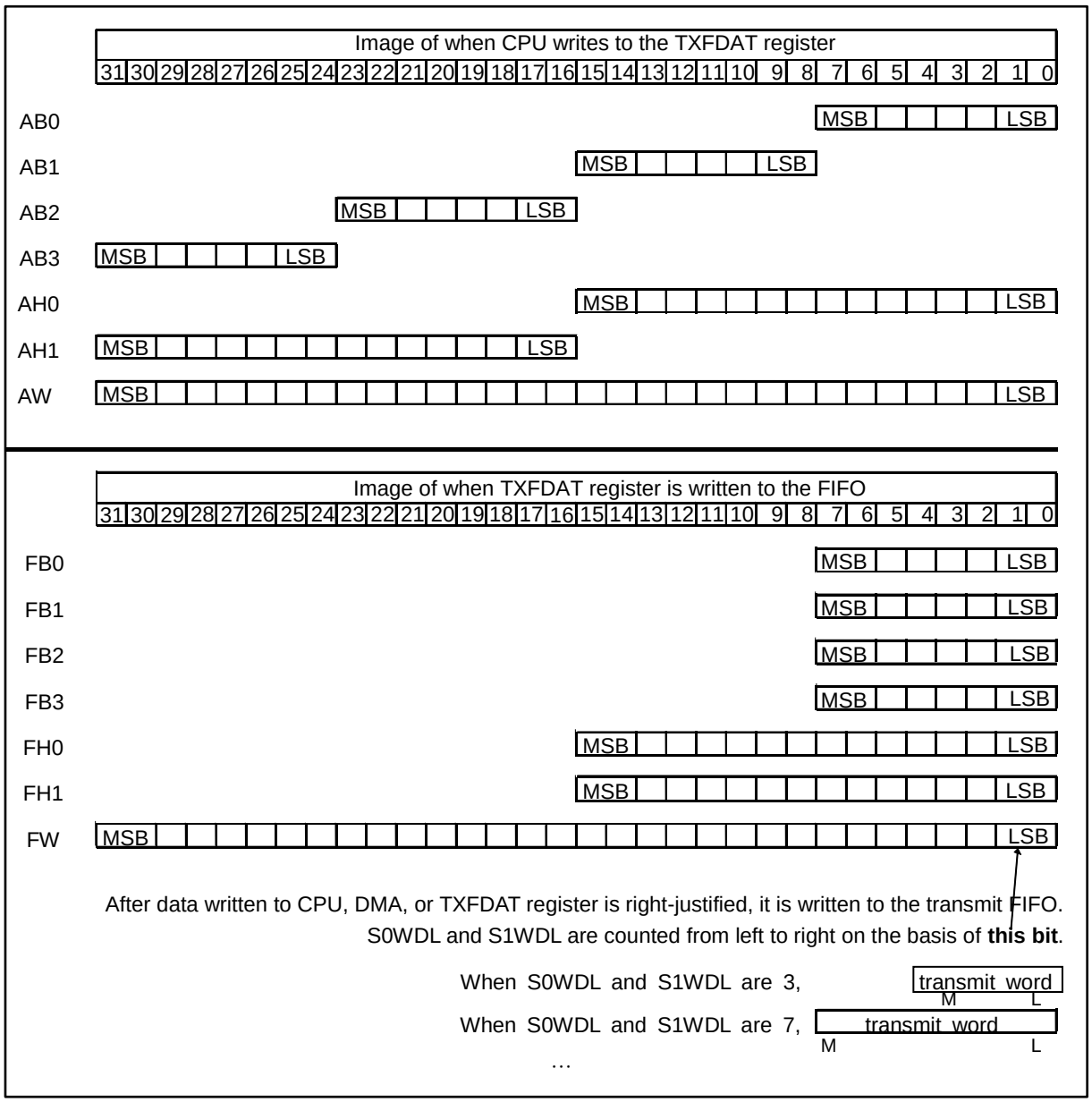


1. A 2-subframe configuration is set when the SBFN bit of the CNTREG register is set to "1".
   The first subframe is number 0, and the next subframe is number 1.
2. The number of channels of subframe 0 is set by the S0CHN bit of the MCR0REG register, and the number of channels of subframe 1 is set by the S1CHN bit of the MCR0REG register. The number of channels for the two subframes can be set independently, and they do not need to be set to the same value. A maximum of 32 channels can be set for each subframe.
3. The channel bit length (word length) of subframe 0 is determined by the S0WDL bit of the MCR0REG register, and the channel bit length (word length) of subframe 1 is determined by the S1WDL bit of the MC0REG register. The channel bit length for each subframe is set independently, and so the channel (word) length for the two subframes do not need to be set to the same value.
4. The channel in the subframe starts from the number 0.
   Each channel of subframe 0 can be enabled or disabled by the corresponding bit of the MCR1REG register, and each channel of subframe 1 can be enabled or disabled by the corresponding bit of the MCR2REG register.
   Data is not transmitted or received for channels that are set to disable.
5. The OVHD bit of the CNTREG register can be set to enable insertion of dummy bits after subframe 1.
   The number of insertable bits is from 0 to 1023 bits.
6. The I2SWS polarity is set by the FSPL bit of the CNTREG register.
7. The pulse width of I2SWS is set by the FSLN bit of the CNTREG register, and it can be set to 1 bit or 1 channel length. When set to 1 channel length, the channel length is determined by the channel length of subframe 0.
8. The timing from the edge where I2SWS is enabled until the first bit of the frame can be set to 0 or 1 bit.

Bit Order Explanation

- Transmit Word Order Operation

Figure 4.3-3 Transmit Word Order



When transmitted by the serial bus, the signal is transmitted from the MSB when the MLSB bit of the CNTREG register is "0" and is transmitted from the LSB when the bit is "1". If the channel length (set to S0CHL and S1CHL) is larger than the word length (set to S0WDL and S1WDL), the bits remaining in the channel are CNTREG:MSKB. Settings where the channel length is smaller than the word length are prohibited.

Note:

- *In Figure 4.3-3, AB0, AB1, AB2, AB3, AH0, AH1, and AW refer to byte 0, byte 1, byte 2, byte 3, half-word 0, half-word 1, and word when performing write access to TXFDAT on the AHB bus. FB0, FB1, FB2, FB3, FH0, FH1, and FW refer to AB0, AB1, AB2, AB3, AH0, AH1, and AW that have been right-justified and then written to the transmit FIFO.*

■ Receive Word Order Operation

Figure 4.3-4 Receive Word Order

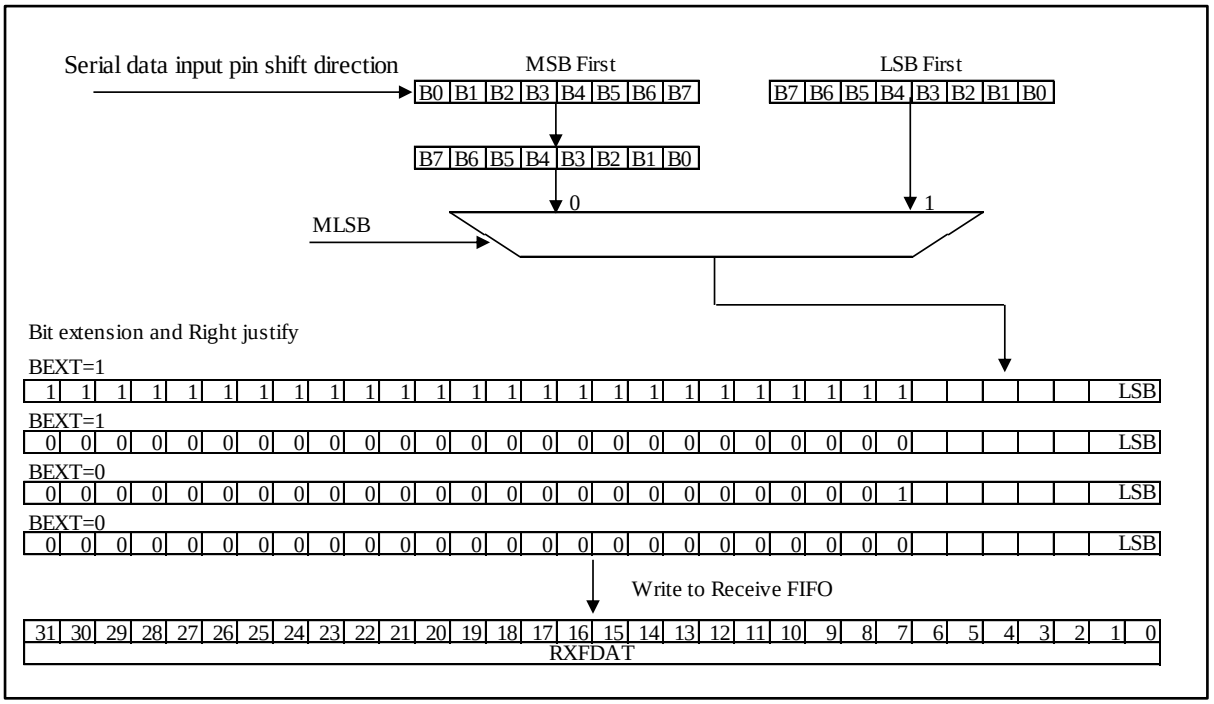


Figure 4.3-4 shows an example when the word length is 8.

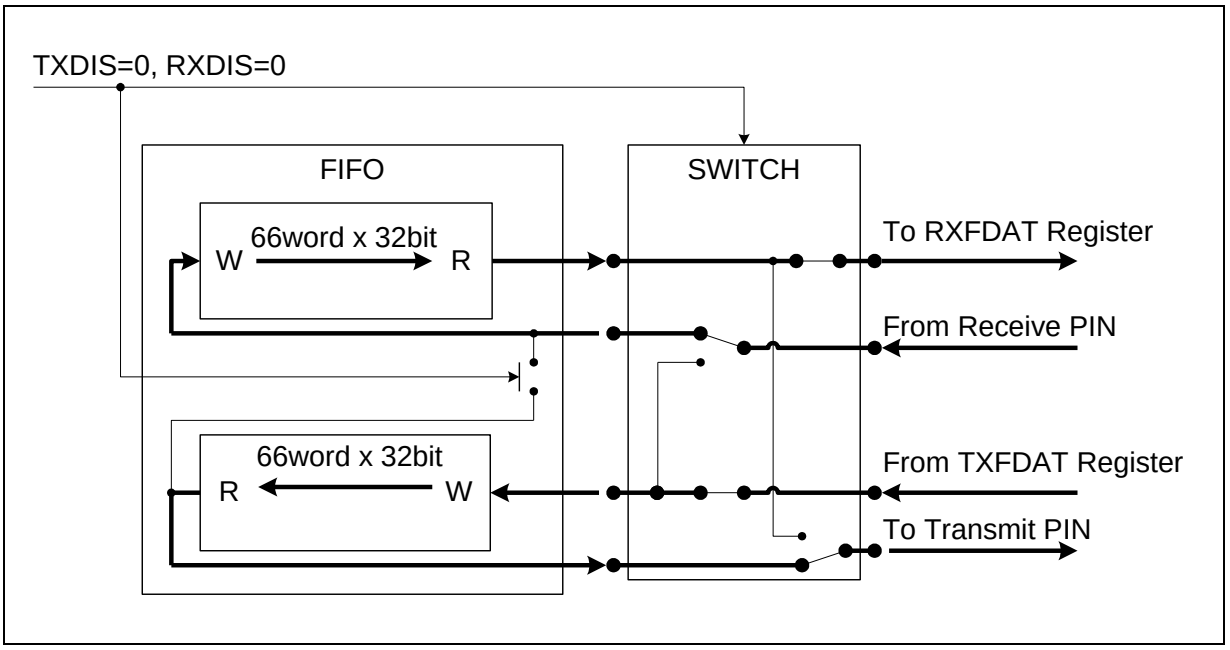
Words received from the serial bus are constantly written to the receive FIFO after right-justification.

As a result, perform read access to RXFDAT from the AHB bus so that byte 0 is read when the word length is 8 or less for a single access, half-word 0 is read when the word length is 8 to 16, and the entire word is read when the word length is 17 to 32.

4.4 FIFO Configuration and Description

Transmit/receive enabled (TXDIS=0, RXDIS=0)

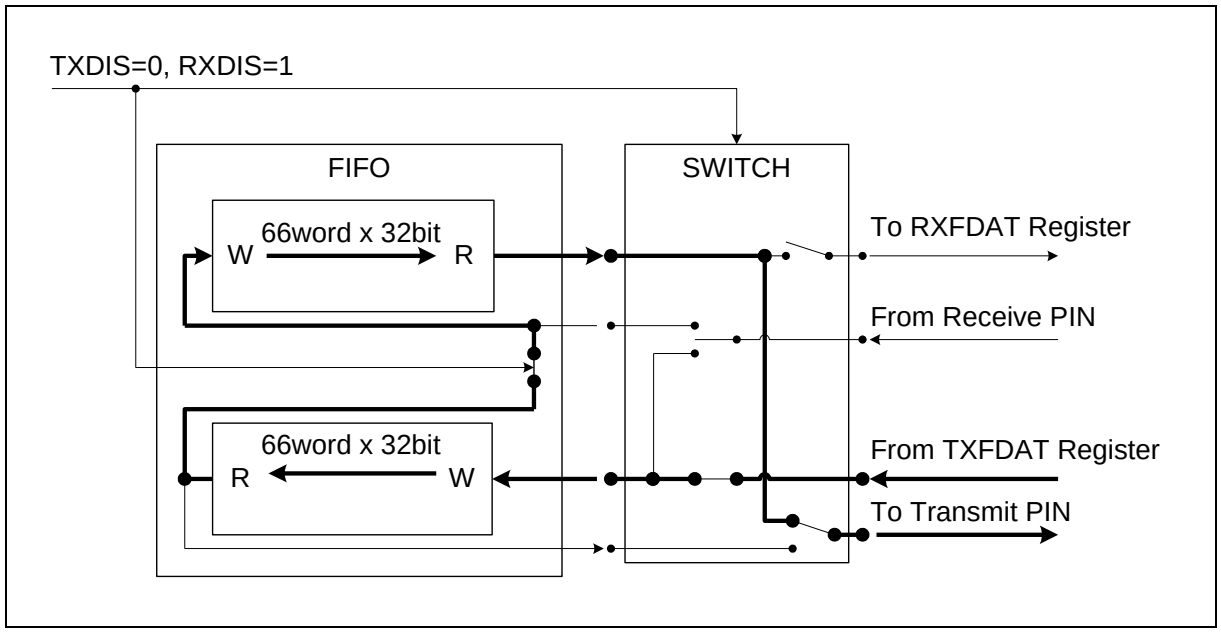
Figure 4.4-1 Transmit/Receive Enabled Mode Data Flow



Transmit/receive enable mode is set by setting the TXDIS bit of the CNTREG register to "0" and the RXDIS bit to "0". This mode operates using a 66 word ×32 bit transmit FIFO or a 66 word ×32 bit receive FIFO.

Transmit-only enabled (TXDIS=0, RXDIS=1)

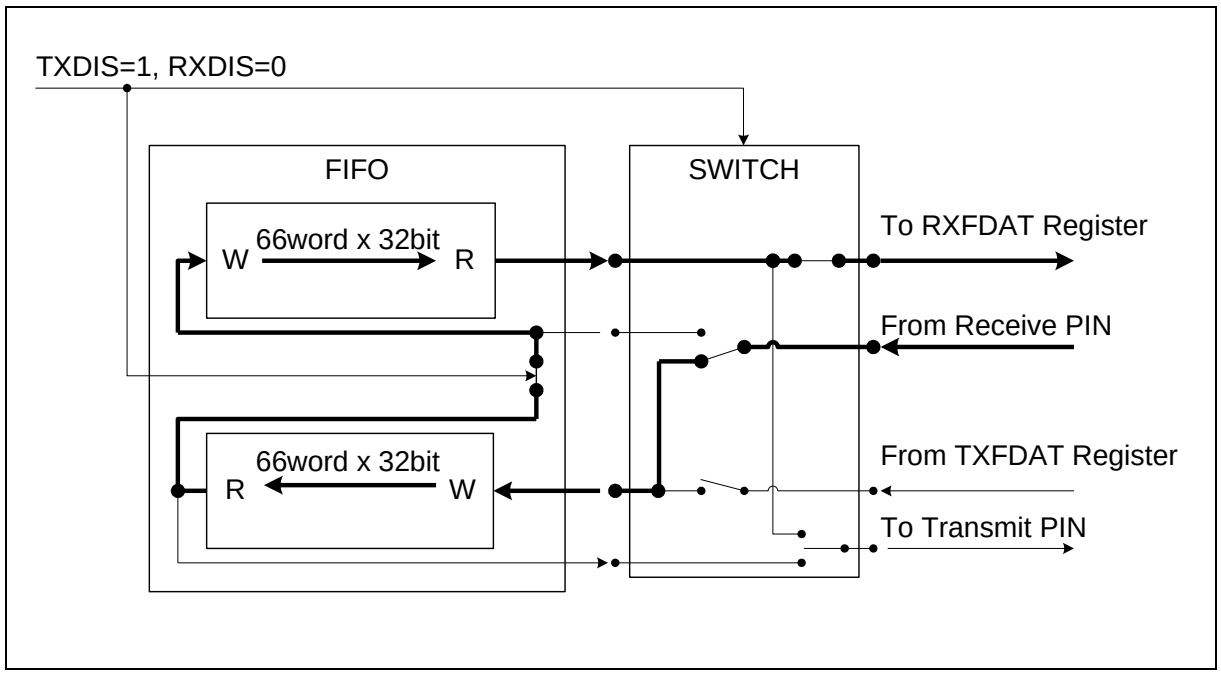
Figure 4.4-2 Data Flow for Transmit-only Enabled Mode



Transmit-only enabled mode is set by setting the TXDIS bit of the CNTREG register to "0" and the RXDIS bit to "1". This mode operates using a 132 word ×32 bit transmit FIFO, and signals are not received.

Receive-only enabled (TXDIS=1, RXDIS=0)

Figure 4.4-3 Data Flow for Receive-only Enabled Mode



Receive-only enabled mode is set by setting the TXDIS bit of the CNTREG register to "1" and the RXDIS bit to "0". This mode operates using a 132 word ×32 bit receive FIFO, and signals are not transmitted.

5. List of I²S Interface Registers

Table 4.4-1 shows the list of registers in I²S.

Table 4.4-1 Register List

Abbreviation	Name	Refer to
RXFDAT	Receive FIFO data register	6.1
TXFDAT	Transmit FIFO data register	6.2
CNTREG	Control register	6.3
MCR0REG	Channel control register 0	6.4
MCR1REG	Channel control register 1	6.5
MCR2REG	Channel control register 2	6.6
OPRREG	Operation control register	6.7
SRST	Soft reset register	6.8
INTCNT	Interrupt control register	6.9
STATUS	Status register	6.10
DMAACT	DMA startup register	6.11
TSTREG	Test register	6.12

6. Details of I^{2}S Interface Registers

This section describes the registers of the I^{2}S interface. All registers of I^{2}S macros support byte (8-bit), half-word (16-bit), and word (32-bit) access. The byte order is little endian.

6.1 Receive FIFO Register (RXFDAT)

The receive FIFO register can hold up to 66 words (transmit/receive enabled mode) or 132 words (receive-only enabled mode).

Bit	31	30	29	28	27	26	25	24
Field	RXDATA							
Attribute	R							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	RXDATA							
Attribute	R							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	RXDATA							
Attribute	R							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	RXDATA							
Attribute	R							
Initial value	0x00							

[bit31:0] RXDATA

Words received from the serial bus are written to the receive FIFO.

- When the frame consists of one subframe
  If the word length set in MCR0REG:S0WDL is smaller than 32 bits (16 bits when CNTREG:RHLL="1"), the upper bits are extended and then written to the receive FIFO.
- When the frame consists of two subframes
  If the word length set in MCR0REG:S0WDL is smaller than 32 bits (16 bits when CNTREG:RHLL="1"), or the word length set in MCR0REG:S1WDL for the received data of subframe 0 is smaller than 32 bits, the upper bits of the received data of subframe 1 are extended and then written to the receive FIFO.
- When CNTREG:BEXT="1"
  Sign extension (extension using MSB of receive word), when this is "0", is accomplished by extended it with "0". When performing read access, the data at the front of the receive FIFO (First In) is read. When performing read access, the data is automatically updated with the next receive FIFO data. Read access can be performed regardless of the shift register operation state. Invalid data is read when STATUS:RXNUM="0".

Writing to RXDATA is ignored.

6.2 Transmit FIFO Register (TXFDAT)

The transmit FIFO register can hold up to 66 words (transmit/receive enabled mode) or 132 words (transmit-only enabled mode).

Bit	31	30	29	28	27	26	25	24
Field	TXDATA							
Attribute	W							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	TXDATA							
Attribute	W							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	TXDATA							
Attribute	W							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	TXDATA							
Attribute	W							
Initial value	0x00							

[bit31:0] TXDATA

As long as the transmit FIFO is not full, the words to be transmitted can be written.
Write access is possible regardless of the shift register operation state.
Words written when the transmit FIFO is full are not written to the transmit FIFO.
Although data to be written uses word, half-word, and byte access, the number of bits actually transmitted is determined by the S0WDL and S1WDL (when the frame has two subframes) of the MCR0REG register.
Data read from TXDATA is invalid data (i.e. the data written last has been right-justified).

6.3 Control Register (CNTREG)

This is the control register.

Bit	31	30	29	28	27	26	25	24
Field	CKRT						OVHD	
Attribute	R/W						R/W	
Initial value	000000						00	

Bit	23	22	21	20	19	18	17	16
Field	OVHD							
Attribute	R/W							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved	MSKB	MSMD	SBFN	RHLL	ECKM	BEXT	FRUN
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	MLSB	TXDIS	RXDIS	SMPL	CPOL	FSPH	FSLN	FSPL
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	0	0	0	0	0

[bit31:26] CKRT

When operating in master mode, this sets the clock division ratio for output.

- When ECKM=0, this divides the frequency of the internal clock (HCLK).
- When ECKM=1, this divides the frequency of the external clock (I2SMCLK external pin).

The frequency division ratio supports even numbers only, and the duty cycle of the clock that is output is 50%.

CKRT[5:0] × 2 is the number of I2SMCLK or HCLK clock cycles contained in one I2SCK cycle.

A setting example is shown below.

■ In external clock mode, when I2SMCLK is 12.288 MHz:

Bit 31:26	Division ratio	I2SCK
0x00	Bypass	12.288 MHz (I2SMCLK is output as-is)
0x01	1/2	6.144 MHz
0x02	1/4	3.072 MHz
0x03	1/6	2.458 MHz
0x04	1/8	1.536 MHz
0x05	1/10	1.228 MHz

■ In internal clock mode, when HCLK is 96 MHz:

Bit 31:26	Division ratio	I2SCK
…	…	…
0x04	1/8	12 MHz
0x05	1/10	9.6 MHz
0x06	1/12	8 MHz
0x07	1/14	6.85 MHz
0x08	1/16	6 MHz
0x09	1/18	5.33 MHz
…	…	…

[bit25:16] OVHD

Following the valid data of the frame, it can insert OVERHEAD bits to enable adjustment of the frame rate.
The OVERHEAD section of the transmit frame is in a high impedance state.
The number of OVERHEAD bits that can be inserted is from 0 to 1023 bits, and are inserted at the end of the frame.
The value set for OVHD is exactly the number of inserted bits.

The relationships below hold between OVHD and the frame sync signal cycle (seconds).

- When the frame consists of one subframe:
  OVHD = Frame sync signal cycle/I2SCK cycle – (S0CHL + 1) × (S0CHN + 1)
- When the frame consists of two subframes:
  OVHD = Frame sync signal cycle/I2SCK cycle – (S0CHL + 1) × (S0CHN + 1) – (S1CHL + 1) × (S1CHN + 1)

[bit15] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit14] MSKB

This sets the serial output data of the invalid transmit frames.

- When set to master mode (MSMD=1), free run mode (FRUN=1), and TXENB=1
  If the transmit FIFO is empty when the frame sync signal is output, MSKB is output in all valid channels of the transmit frame.
- When set to slave mode (MSMD=0) and TXENB=1
  If the transmit FIFO is empty when the frame sync signal is received, MSKB is output in all valid channels of the transmit frame.

If the transmit word length is shorter than the channel length, MSKB is driven for the excess bits in the transmit channel (channel length - word length).

[bit13] MSMD

This sets master or slave mode.

Bit	Description
1	Master mode
0	Slave mode

[bit12] SBFN

This specifies the subframe configuration (number of subframes) of the frame.

Bit	Description
0	The frame has one subframe (subframe 0 only).
1	The frame has two subframes (subframe 0 and subframe 1). The frame starts from subframe number 0.

[bit11] RHLL

This sets the FIFO word configuration to one or two words.
This assumes the use of I2S, MSB-Justified, or other protocols. (See "FIFO Word Configuration by RHLL Bit Settings".)

Bit	Description
0	Handles 32-bit FIFO words as one word.
1	Divides 32-bit FIFO words into an upper 16 bits and lower 16 bits and handles them as two words by the serial bus.

- When transmitting
  Transfer is performed by the serial bus in the order of lower bits→upper bits→lower bits→upper bits.
- When receiving
  Two consecutive words from the serial bus are handled as upper and lower, and then bound to a 32-bit word and written to the receive FIFO.

[bit10] ECKM

In master mode, this selects the base clock divider.

Bit	Description
0	Divides and outputs an internal clock (internal AHB bus clock: HCLK).
1	Divides and outputs an I2S clock generator clock (I2SMCLK pin output) and external clock (I2SMCLK pin input).

[bit9] BEXT

When the receive word length is smaller than the FIFO word length (32 bits when RHLL is "0", and 16 bits when RHLL is "1"), this sets the upper bit (FIFO word length - received word length) extension mode.
(See "Bit Order Explanation" and "FIFO Word Configuration by RHLL Bit Settings".)

Bit	Description
0	Extends using 0.
1	Extends using the sign bit ("1" when word MSB is "1", and "0" when word MSB is "0")

[bit8] FRUN

This sets the output mode of the frame sync signal.

Bit	Description
1	When set to free run mode and OPRREG register START bit is "1", the frame sync signal performs free running at the frame rate that was set. The frame sync signal is not output when the START bit is "0".
0	When set to burst mode and OPRREG register START bit is "1", the frame sync signal is output depending on the TXENB, RXENB, and transmit/receive FIFO status.

[bit7] MLSB

This sets word bit shift order.

Bit	Description
1	Shift starts from LSB of word.
0	Shift starts from MSB of word.

[bit6] TXDIS

This enables or disables the transmit function.

Bit	Description
0	Enables the transmit function.
1	Disables the transmit function.

[bit5] RXDIS

This enables or disables the receive function.

Bit	Description
0	Enables the receive function.
1	Disables the receive function.

[bit4] SMPL

This specifies the point where data is sampled.

Bit	Description
0	Sampling is performed in the middle of the receive data.
1	Sampling is performed at the end of the receive data.

[bit3] CPOL

This specifies the I2SCK polarity where drive sampling of the serial data is performed.

Bit	Description
0	Drives data at the rising edge of I2SCK and is sampled at the falling edge of I2SCK.
1	Drives data at falling edge of I2SCK and is sampled at the rising edge of I2SCK.

[bit2] FSPH

This specifies the phase for I2SWS frame data.

Bit	Description
0	I2SWS is enabled one block before the first bit of the frame data.
1	I2SWS is enabled at the same time as the frame data and first bit.

[bit1] FSLN

This specifies the pulse width of I2SWS.

Bit	Description
0	Pulse width will be one I2SCK cycle (1 bit).
1	Pulse width will be one channel length (1 channel). Setting to "1" is prohibited when the frame length is one channel.

[bit0] FSPL

This sets the polarity of the I2SWS pin.

Bit	Description
0	I2SWS is "1", and the frame sync signal is enabled. This is "0" when idle.
1	I2SWS is "0", and the frame sync signal is enabled. This is "1" when idle.

Note:

- *Overwriting of the CNTREG register is prohibited when the OPRREG:START bit is "1".*

6.4 Channel Control Register 0 (MCR0REG)

This register sets the subframes.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	S1CHN					S1CHL	
Attribute	R	R/W					R/W	
Initial value	0	00000					00	

Bit	23	22	21	20	19	18	17	16
Field	S1CHL			S1WDL				
Attribute	R/W			R/W				
Initial value	000			00000				

Bit	15	14	13	12	11	10	9	8
Field	Reserved	S0CHN					S0CHL	
Attribute	R	R/W					R/W	
Initial value	0	00000					00	

Bit	7	6	5	4	3	2	1	0
Field	S0CHL			S0WDL				
Attribute	R/W			R/W				
Initial value	000			00000				

[bit31] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit30:26] S1CHN

This sets the number of channels for subframe 1.
This is valid only when the frame has a 2-subframe configuration (CNTREG:SBFN="1").
This is invalid when the frame has a 1-subframe configuration (CNTREG:SBFN="0").
Up to 32 channels can be specified.

Set S1CHN to the number of channels - 1.

- Example 1 : When set to S1CHN="00011", subframe 1 has a 4-channel configuration.
- Example 2 : When set to S1CHN="11111", subframe 1 has a 32-channel configuration.

S1WDL is valid only when the frame has a 2-subframe configuration (CNTREG:SBFN="1"), and it is invalid when the frame has a 1-subframe configuration (CNTREG:SBFN="0").

[bit25:21] S1SHL

This sets the channel length (channel bit length) of the channels that make up subframe 1.
The channel length can be set from 7 to 32 bits. Setting of the channel length from 1 to 6 bits is prohibited.

Set S1CHL to the channel length - 1.

- Example 1 : When set to S1CHL="00110", the channel length is 7 bits.
- Example 2 : When set to S1CHL="11111", the channel length is 32 bits.

The channel length can be set to 32 or less regardless of the RHLL value of the CNTREG register.
S1WDL is valid only when the frame has a 2-subframe configuration (CNTREG:SBFN="1"), and it is invalid when the frame has a 1-subframe configuration (CNTREG:SBFN="0").

[bit20:16] S1WDL

This sets the word length (number of bits per word in the channel) of the channels that make up subframe 1.
The word length can be set from 7 to 32 bits. Setting of the word length from 1 to 6 bits is prohibited.
Set S1WDL to the word length - 1.

- Example 1 : When set to S1WDL="00110", the word length is 7 bits.
- Example 2 : When set to S1WDL="11111", the word length is 32 bits.
  - When CNTREG:RHLL="1"
    Set the word length to 16 or less and to the channel length or less that was set in S1CHL.
  - When CNTREG:RHLL="0"
    Set the word length to 32 or less and to the channel length or less that was set in S1CHL.

S1WDL is valid only when the frame has a 2-subframe configuration (CNTREG:SBFN="1"), and it is invalid when the frame has a 1-subframe configuration (CNTREG:SBFN="0").

[bit15] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit14:10] S0CHN

This sets the number of channels for subframe 0.
Up to 32 channels can be specified.
Set S0CHN to the number of channels - 1.

- Example 1: When set to S0CHN="00011", subframe 0 has a 4-channel configuration.
- Example 2: When set to S0CHN="11111", subframe 0 has a 32-channel configuration.

[bit9:6] S0CHL

This sets the channel length (channel bit length) of the channels that make up subframe 0.
The channel length can be set from 4 to 32 bits. Setting of the channel length from 1 to 6 bits is prohibited.
Set S0CHL to the channel length - 1.

- Example 1: When set to S0CHL="00110", the channel length is 7 bits.
- Example 2: When set to S0CHL="11111", the channel length is 32 bits.

The channel length can be set to 32 or less regardless of the RHLL value of the CNTREG register.

[bit4:0] S0WDL

This sets the word length (number of bits per word in the channel) of the channels that make up subframe 0.
The word length can be set from 4 to 32 bits. Setting of the word length from 1 to 6 bits is prohibited.
Set S0WDL to the word length - 1.

- Example 1: When set to S0WDL="00110", the word length is 7 bits.
- Example 2: When set to S0WDL="11111", the word length is 32 bits.
  - When CNTREG:RHLL="1"
    Set the word length to 16 or less and to the channel length or less that was set in S0CHL.
  - When CNTREG:RHLL="0"
    Set the word length to 32 or less and to the channel length or less that was set in S0CHL.

6.5 Channel Control Register 1 (MCR1REG)

This register controls the enable/disable state of the channels in subframe 0.

Bit	31	30	29	28	27	26	25	24
Field	S0CH31	S0CH30	S0CH29	S0CH28	S0CH27	S0CH26	S0CH25	S0CH24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	S0CH23	S0CH22	S0CH21	S0CH20	S0CH19	S0CH18	S0CH17	S0CH16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	S0CH15	S0CH14	S0CH13	S0CH12	S0CH11	S0CH10	S0CH09	S0CH08
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	S0CH07	S0CH06	S0CH05	S0CH04	S0CH03	S0CH02	S0CH01	S0CH00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] S0CH00 to S0CH31

Each name (S0CHxx) of each bit indicates channel xx of subframe 0.

■ Example:

The S0CH00 bit controls channel number 0 of subframe 0.

The S0CH31 bit controls channel number 31 of subframe 0.

Bit	Description
0	Disables the corresponding channel. Transmitting and receiving is not performed for disabled channels.
1	Enables the corresponding channel. Transmitting and receiving is performed for enabled channels.

6.6 Channel Control Register 2 (MCR2REG)

This controls the enable/disable state of the channels in subframe 1.

Bit	31	30	29	28	27	26	25	24
Field	S1CH31	S1CH30	S1CH29	S1CH28	S1CH27	S1CH26	S1CH25	S1CH24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	S1CH23	S1CH22	S1CH21	S1CH20	S1CH19	S1CH18	S1CH17	S1CH16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	S1CH15	S1CH14	S1CH13	S1CH12	S1CH11	S1CH10	S1CH09	S1CH08
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Field	S1CH07	S1CH06	S1CH05	S1CH04	S1CH03	S1CH02	S1CH01	S1CH00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] S1CH00 to S1CH31

Each name (S1CHxx) of each bit indicates channel xx of subframe 1.

This is invalid when the frame has a 1-subframe configuration (CNTREG:SBFN="0").

- Example:

  The S1CH00 bit controls channel number 0 of subframe 1.

  The S1CH31 bit controls channel number 31 of subframe 1.

Bit	Description
0	Disables the corresponding channel. Transmitting and receiving is not performed for disabled channels.
1	Enables the corresponding channel. Transmitting and receiving is performed for enabled channels.

6.7 Operation Control Register (OPRREG)

This register sets operations.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							RXENB
Attribute	R							R/W
Initial value	0000000							0

Bit	23	22	21	20	19	18	17	16
Field	Reserved							TXENB
Attribute	R							R/W
Initial value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved							START
Attribute	R							R/W
Initial value	0000000							0

[bit31:25] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit24] RXENB

This sets the enabled/disabled state of the receive operation.

Bit	Description
0	Disables the receive operation. Writing "0" to this bit makes the receive FIFO empty. When RXENB is "0", data received from the serial receive bus is not written to the receive FIFO. If DMA transfer is in progress, the DMA receive channel is stopped.
1	Enables the receive operation.

[bit23:17] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit16] TXENB

This sets the enabled/disabled state of the transmit operation.

Bit	Description
0	Disables the transmit operation. Writing "0" to this bit makes the receive FIFO empty. When this bit is "0", data written to the DMA or TXFDAT register is not written to the transmit FIFO. If DMA transfer is in progress, the DMA transmit channel is stopped.
1	Enables the transmit operation.

[bit15:1] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit0] START

This enables or disables the I2S interface.

Bit	Description
0	Stops the I2S interface. Writing "0" to this bit makes the internal transmit/receive FIFO empty.
1	Enables the I2S interface.

When START is "1", overwriting of the CNTREG, MCR0REG, MCR1REG, and MCR2REG registers is prohibited.

6.8 Soft Reset Register (SRST)

This is the soft reset register.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved							SRST
Attribute	R							R/W
Initial value	0000000							0

[bit31:1] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit0] SRST

Writing "1" performs a soft reset.

Applying a soft reset will set the STATUS register and the internal state machines to their initial states and make the transmit/receive FIFO empty. Registers other than STATUS, INTCNT, and DMAACT are not affected.

After "1" is written to this bit, a readout value of "0" obtained by read access indicates that the soft reset was completed, and a value of "1" indicates that the soft reset is in progress.

6.9 Interrupt Control Register (INTCNT)

This is the interrupt control register.

Bit	31	30	29	28	27	26	25	24
Field	Reserved	TXUD1M	TBERM	FERRM	TXUD0M	TXOVM	TXFDM	TXFIM
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Field	Reserved		RBERM	RXUDM	RXOVM	EOPM	RXFDM	RXFIM
Attribute	R		R/W	R/W	R/W	R/W	R/W	R/W
Initial value	00		1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Field	Reserved				TFTH			
Attribute	R				R/W			
Initial value	0000				0000			

Bit	7	6	5	4	3	2	1	0
Field	Reserved		RPTMR		RFTH			
Attribute	R		R/W		R/W			
Initial value	00		00		0000			

[bit31] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit30] TXUD1M

This bit masks the transmit FIFO underflow interrupt.

This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by TXUDR1 of the STATUS register.
1	Masks the interrupts to the CPU by TXUDR1 of the STATUS register.

[bit29] TBERM

This bit masks the transmit channel block size error interrupt.

This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by TBERR of the STATUS register.
1	Masks the interrupts to the CPU by TBERR of the STATUS register.

[bit28] FERRM

This bit masks the frame error interrupt mask.

This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by FERR of the STATUS register.
1	Masks the interrupts to the CPU by FERR of the STATUS register.

[bit27] TXUD0M

This bit masks the transmit FIFO underflow interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by TXUDR0 of the STATUS register.
1	Masks the interrupts to the CPU by TXUDR0 of the STATUS register.

[bit26] TXOVM

This bit masks the transmit FIFO overflow interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by TXOVM of the STATUS register.
1	Masks the interrupts to the CPU by TXOVM of the STATUS register.

[bit25] TXFDM

This bit masks the transmit DMA request.
This is set to the initial value by a soft reset.

Bit	Description
0	DMA transfer request is made when the receive data written to the transmit FIFO meets or exceeds the threshold value.
1	DMA transfer request is not made even if the receive data written to the transmit FIFO meets or exceeds the threshold value.

[bit24] TXFIM

This bit masks the transmit FIFO interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by TXFI of the STATUS register.
1	Masks the interrupts to the CPU by TXFI of the STATUS register.

[bit23:22] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit21] RBERM

This bit masks the receive channel block size error interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by RBERR of the STATUS register.
1	Mask the interrupts to the CPU by RBERR of the STATUS register.

[bit20] RBUDM

This bit masks the receive FIFO underflow interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by RXUDR of the STATUS register.
1	Masks the interrupts to the CPU by RXUDR of the STATUS register.

[bit19] RXOVM

This bit masks the receive FIFO overflow interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by RXOVR of the STATUS register.
1	Masks the interrupts to the CPU by RXOVR of the STATUS register.

[bit18] EOPM

This bit masks the interrupts by EOPI of the STATUS register.
This is set to the initial value by the register or a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by EOPI of the STATUS register.
1	Masks the interrupts to the CPU by EOPI of the STATUS register.

[bit17] RXFDM

This bit masks the receive DMA request.
This is set to the initial value by a soft reset.

Bit	Description
0	DMA request is made when the receive data written to the receive FIFO meets or exceeds the threshold value.
1	DMA request is not made even if the receive data written to the receive FIFO meets or exceeds the threshold value.

[bit16] RXFIM

This bit masks the receive FIFO interrupt.
This is set to the initial value by a soft reset.

Bit	Description
0	Does not mask the interrupts to the CPU by RXFI of the STATUS register.
1	Masks the interrupts to the CPU by RXFI of the STATUS register.

[bit15:12] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit11:8] TFTH

This bit sets the transmit FIFO threshold value.
When the transmit FIFO empty space meets or exceeds the threshold value

- An interrupt is issued to the CPU when TXFIM is "0".
- A DMA request is issued to the DSTC when TXFDM is "0".

TFTH is set by the formula below.

TFTH = Transmit FIFO threshold value -1

[bit7:6] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit5:4] RPTMR

This is the bit for setting the packet receive completion timer.
This sets the timeout value of the internal receive completion timer.
The internal receive completion timer continuously counts up when the receive FIFO is not empty and the receive FIFO data count is less than the threshold value. The timer is cleared if the receive FIFO becomes empty or when the threshold value is met or exceeded. When a timeout occurs in the timer, the EOPI bit of the STATUS register is set to "1".
This is set to the initial value by a soft reset.

bit 5:4	Description
00	0 (Timer does not operate.)
01	54000 HCLK cycle
10	108000 HCLK cycle
11	216000 HCLK cycle

[bit3:0] RFTH

This bit sets the receive FIFO threshold value.

When the number of receive words written to the receive FIFO meets or exceeds the threshold value

- An interrupt is issued to the CPU when RXFIM is "0".
- A DMA request is issued to the DSTC when RXFDM is "0".

RFTH is set by the formula below.

RFTH = Receive FIFO threshold value -1

6.10 Status Register (STATUS)

This is the status register.

Bit	31	30	29	28	27	26	25	24
Field	TBERR	RBERR	FERR	TXUDR1	TXUDR0	TXOVR	RXUDR	RXOVR
Attribute	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	Reserved				EOPI	BSY	TXFI	RXFI
Attribute	R				R/W	R	R	R
Initial value	0000				0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	TXNUM							
Attribute	R							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	RXNUM							
Attribute	R							
Initial value	0x00							

[bit31] TBERR

If the block size of the DMA transmit channel is set to a value larger than the transmit FIFO threshold value of the I^2S interface, and operation is performed, this bit is set to "1", and the DMA transmit channel is stopped.

An interrupt is issued to the CPU when this bit is "1" and the INTCNT:TBERM bit is "0".
This is set to the initial value ("0") by a soft reset.

[bit30] RBERR

If the block size of the DMA receive channel is set to a value larger than the receive FIFO threshold value of the I^2S interface, and operation is performed, this bit is set to "1", and the DMA receive channel is stopped.

An interrupt is issued to the CPU when this bit is "1" and the INTCNT:RBERM bit is "0".
This is set to the initial value ("0") by a soft reset.

[bit29] FERR

This indicates that a frame error has occurred.

- When in slave mode (CNTREG:MSMD=0) and free run mode (CNTREG:FRUN=0)
  This is set to "1" if the frame sync signal cannot be received at the frame rate that was set.
- When in slave mode (CNTREG:MSMD=0) and not in free run mode (CNTREG:FRUN=1)
  This is set to "1" if the next frame sync signal is received during a frame transmit/receive operation.

An interrupt is issued to the CPU when this bit is "1" and the INTCNT:FERRM bit is "0".
"1" is written from the CPU, or this is cleared to "0" by a soft reset.

[bit28] TXUDR1

This is set to "1" when the transmit FIFO underflows at the frame start.
"1" is written from the CPU, or this is cleared to "0" by a soft reset.

[bit27] TXUDR0
This is set to "1" when the transmit FIFO underflows during frame transmission (from the second bit word of the frame to the last word of the frame).

"1" is written from the CPU, or this is cleared to the initial value by a soft reset.

[bit26] TXOVR
This is set to "1" when the transmit FIFO overflows.
This indicates that writing of transmit data occurred while the transmit FIFO was full.
This means that transmit data exceeding one word is discarded when this bit is "1".
An interrupt is issued to the CPU when this bit is "1" and the INTCNT:TXOVM bit is "0".
"1" is written from the CPU, or this is cleared to the initial value by a soft reset.

[bit25] RXUDR
This is set to "1" when the receive FIFO underflows.
This means that a read access to the receive FIFO occurred when the receive FIFO was empty.
"1" is written from the CPU, or this is cleared to the initial value by a soft reset.

[bit24] RXOVR
This is set to "1" when the receive FIFO overflows.
This indicates that a receive operation occurred while the receive FIFO was full.
This means that receive data exceeding one word is discarded when this bit is "1".
An interrupt is issued to the CPU when this bit is "1" and the INTCNT:RXOVM bit is "0".
"1" is written from the CPU, or this is cleared to the initial value by a soft reset.

[bit23:20] Reserved: Reserved bit
Writing is ignored. "0" is read as the readout value.

[bit19] EOPI
This is the interrupt flag based on the receive timer. The receive timer is enabled when both of the conditions below are satisfied at the same time.

1. The CNTREG:RXDIS bit is set to "0".
2. The OPRREG:START bit is "1", and RXENB=1 is set.

After reset, the count-up operation is started by the first word receive operation.
This bit is set to "1" if the receive FIFO is not empty when the timer reaches the time that was set by the INTCNT:RPTMR bit. An interrupt is issued to the CPU when this bit is "1" and the INTCNT:EOPM bit is "0".

This is automatically cleared when the receive FIFO data meets or exceeds the threshold value or when the receive FIFO becomes empty.

If the data count for the receive FIFO is less than the threshold value and is not empty, the clear is canceled, and the count-up operation is started again.

This is cleared to "0" even if "1" is written from the CPU. This is set to the initial value by a soft reset.

[bit18] BSY
This indicates the status (busy or idle) of the serial transmit control unit. This bit is not affected by a soft reset.

Bit	Description
1	Serial transmit control unit is busy.
0	Serial transmit control unit is idle.

[bit17] TXFI

This is set to "1" when the transmit FIFO empty slot meets or exceeds the threshold value that was set by the INTCNT:TFTH bit.
An interrupt is issued to the CPU when this bit is "1" and the INTCNT:TXFIM bit is "0".
A DMA request is issued when this bit is "1" and the INTCNT:TXFDM bit is "0".

This is automatically cleared to "0" when the number of empty slots in the receive FIFO is less than the threshold value due to writing of the TXFDAT register from the CPU or DSTC.
This bit is "0" when the OPRRE:START bit is "0" or the OPRREG:TXENB bit is "0". This becomes "0" during the soft reset period if a soft reset is applied when the START bit is "1" and the TXENB bit is "1", but this becomes "1" after the soft reset is ended.

[bit16] RXFI

This is set to "1" when the receive FIFO data count meets or exceeds the threshold value that was set by the INTCNT:RFTH bit.
An interrupt is issued to the CPU when this bit is "1" and the INTCNT:RXFIM bit is "0".
A DMA request is issued when this bit is "1" and the INTCNT:RXFDM bit is "0".

This is automatically cleared to "0" when the data count in the receive FIFO is less than the threshold value due to reading of the RXFDAT register from the CPU or DSTC.
This bit is "0" when the OPRREG:START bit is "0" or the OPRREG:RXENB bit is "0". This is set to "0" by a soft reset.

[bit15:8] TXNUM

This indicates the data count in the transmit FIFO.
This is incremented by write access to the TXFDAT register and is decremented by the transmission of serial words.
A maximum value of 66 can be displayed in transmit/receive simultaneous mode, and a maximum value of 132 can be displayed in transmit-only mode.
This is set to the initial value by a soft reset.

[bit7:0] RXNUM

This indicates the data count in the receive FIFO.
This is incremented when a word is received from the serial bus and is decremented by read access to the RXFDAT register.

A maximum value of 66 can be displayed in simultaneous transmit/receive mode, and a maximum value of 132 can be displayed in receive-only mode.
This is set to the initial value by a soft reset.

6.11 DMA Startup Register (DMAACT)

This is the DMA access register.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							TL1E0
Attribute	R							R/W
Initial value	0000000							0

Bit	23	22	21	20	19	18	17	16
Field	Reserved							TDMACT
Attribute	R							R/W
Initial value	0000000							0

Bit	15	14	13	12	11	10	9	8
Field	Reserved				RL1E0			
Attribute	R				R/W			
Initial value	0000000				0			

Bit	7	6	5	4	3	2	1	0
Field	Reserved							RDMACT
Attribute	R							R/W
Initial value	0000000							0

[bit31:25] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit24] TL1E0

This sets the operation mode of TXDREQ.

Bit	Description
0	Edge-Sensitive mode
1	Level-Sensitive mode

[bit23:17] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit16] TDMACT

This bit is enabled when the same register TL1E0="0" (TXDREQ is operating in Edge-Sensitive mode). This indicates that the transmit channel of the DMA (I2S interface) is enabled and that edge detection of TXDREQ is possible.

After the software starts the transmit channel, notification of the starting of the transmit channel must be sent to the I2S interface by writing "1" to this bit. The I2S interface automatically clears this bit whenever DMA packet transmit transfer is completed. A transmit channel block transfer request is not sent to the DSTC when this bit is "0".

"0" is written from the CPU, or this is cleared by a soft reset.

Bit	Description
0	The DMA transmit channel is stopped, and TXDREQ detection cannot be performed.
1	The DMA transmit channel is running, and TXDREQ detection can be performed.

[bit15:9] Reserved: Reserved bit
Writing is ignored. "0" is read as the readout value.

[bit8] RL1E0
This sets the operation mode of RXDREQ.

Bit	Description
0	Edge-Sensitive mode
1	Level-Sensitive mode

[bit7:1] Reserved: Reserved bit
Writing is ignored. "0" is read as the readout value.

[bit0] RDMACT
This bit is enabled when the same register RL1E0="0" (RXDREQ is operating in Edge-Sensitive mode). This indicates that the receive channel of the DMA (I2S interface) is enabled and that edge detection of RXDREQ is possible.
After the software starts the receive channel, notification of the starting of the receive channel must be sent to the I2S interface by writing "1" to this bit. A receive channel block transfer request is not sent to the DSTC when this bit is "0". The I2S interface automatically clears this bit whenever DMA packet receive transfer is completed.

"0" is written from the CPU, or this is cleared to "0" by a soft reset.

Bit	Description
0	The DMA receive channel is stopped, and RXDREQ detection cannot be performed.
1	The DMA receive channel is running, and RXDREQ detection can be performed.

6.12 Test Register (TSTREG)

This sets the loopback mode.

Bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R							
Initial value	0x00							

Bit	7	6	5	4	3	2	1	0
Field	Reserved							LBMD
Attribute	R							R/W
Initial value	0000000							0

[bit31:1] Reserved: Reserved bit

Writing is ignored. "0" is read as the readout value.

[bit0] LBMD

0: Normal mode

1: Loopback mode

The serial transmit pin is connected internally to the serial receive pin. Serial receive data from external sources is ignored.

Set the following settings when running in loopback mode.

CNTREG : MSMD=1, CNTREG:TXDIS=0, CNTREG:RXDIS=0

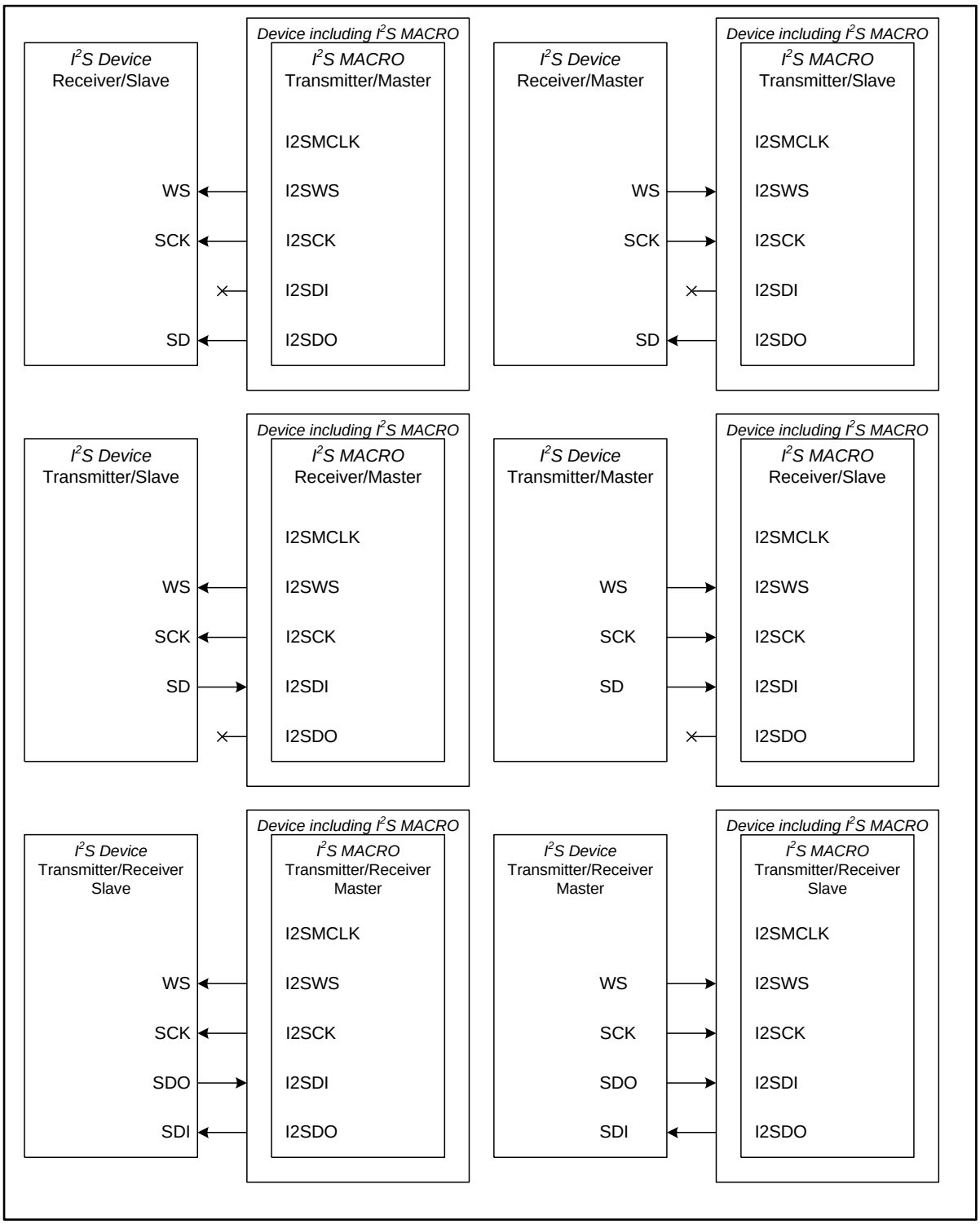
OPRREG : TXENB=0, OPRREG:RXENB=0

7. Application Notes

7.1 I²S and MSB-Justified Protocol

Connection Diagrams

Figure 7.1-1 I2S Connection Examples



I²S, MSB-Justified Protocol

I²S (abbreviation for Inter-Integrated Circuit Sound) is the protocol for digital stereo audio proposed by Philips Semiconductors. SCK and WS are output by the master on the I²S bus. A dedicated controller for the master can be used, and I2S devices for transmitting and receiving can also be used as masters. The master outputs serial data from the MSB of the PCM data. The word select (WS) signal indicates which channel is being used by the PCM data that is being sent. When WS is set to "0", this indicates the left channel, and when it is set to "1", this indicates the right channel. The MSB of the channel data is constantly delayed by one clock from the transition point of WS. There are no specific limitations on the bit length of the channel. The I²S slave counts the word length specified by its own system from the serial data MSB to form one word. The section exceeding the specified word length is ignored, and any sections that are short are filled by "0". Data sampling is always performed on the rising edge of SCK. Serial data output must satisfy the sampling timing of the I²S receive unit. There are no specific limitations on whether signals are output on the rising edge or falling edge of SCK.

Note:

- *I2S is not a protocol for controlling audio codec devices such as by writing and reading registers. As a result, codec devices that support I²S normally provide a separate interface for device control.*

The MSB-Justified protocol is similar to I²S. The WS transition point and serial data MSB occur simultaneously. WS indicates the right channel with "1" and indicates the left channel with "0".

I²S and MSB-Justified Bus signals	
SCK	Continuous serial clock
WS	Word Select
SD	Serial data

Figure 7.1-2 I²S Data Format

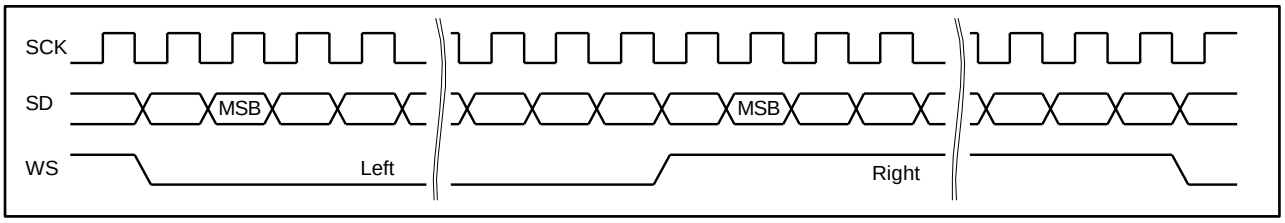
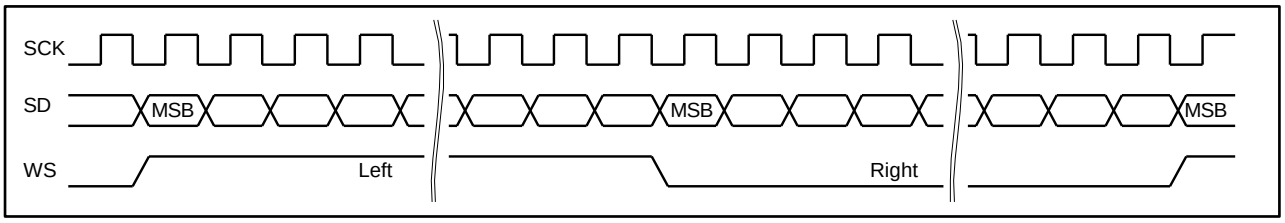


Figure 7.1-3 MSB-Justified Data Format



Initialization Process

Set the settings below to enable operation of the I^2S interface using the I^2S or MSB-Justified protocol.

1. CNTREG register settings
   a. To transmit and receive simultaneously, set TXDIS to "0", and RXDIS to "0".
      The FIFO configuration consists of a transmit and receive FIFO of 66 words each, allowing full duplex communication.
      To transmit only, set TXDIS to "0", and RXDIS to "1".
      The FIFO configuration consists of a transmit FIFO of 132 words.
      To receive only, set TXDIS to "1", and RXDIS to "0".
      The internal FIFO configuration consists of a receive FIFO of 132 words.
   b. This sets the MSMD. (1: Master, 0: Slave (Initial value is 0.))
   c. To operate in master mode, set ECKM. This does not need to be set when operating in slave mode.
   d. When operating in master mode, set the clock division ratio in CKRT.
      This does not need to be set when operating in slave mode.
   e. Set SBFN.
      0: Sets I^2S frames to a 1-subframe configuration, and the number of channels for subframe 0 to two.
      1: Sets I^2S frames to a 2-subframe configuration, and the number of channels for each subframe to one.
   f. Set FSPL.
      Set this to "1" for I^2S protocol, and to "0" for MSB-Justified protocol.
      The first frame after resetting starts from the left channel.
   g. Set FSLN to "1". The width of the frame sync signal will be the length of one word in subframe 0.
   h. Set FSPH.
      Set this to "0" for I2S protocol, and to "1" for MSB-Justified protocol.
   i. Set CPOL.
      Set this to "1" to drive data on the falling edge of SCK, and to "0" to drive data on the falling edge of SCK.
   j. Set RHLL. This will vary depending on how the software arranges the left and right channel audio data in memory. For details, see "FIFO Word Configuration by RHLL Bit Settings".
2. MCR0REG register settings
   a. Set the number of channels.
      When SBFN is set to "0", S0CHN is set to "00001", and S1CHN does not need to be set.
      When SBFN is set to "1", S0CHN and S1CHN are set to "00000".
   b. Set the channel length. The channel length is determined by the frame rate of the application.
      When SBFN is set to "1", S1CHN and S0CHN are set to the same value.
      When SBFN is set to "0", S0CHL only is set, and S1CHL does not need to be set.
   c. Set the word length. This is determined by the bit width of the PCM data to be transmitted or received.
      When SBFN is set to "1", S0WDL and S1WDL are set to the same value.
      When SBFN is set to "0", S0WDL only is set, and S1WDL does not need to be set.
3. MCR1REG register settings
   a. When SBFN is set to "0", S0CH00 and S0CH01 are set to "1", and S0CH02 to S0CH31 are set to "0".
      When SBFN is set to "1", S1CH00 is set to "1", and S0CH01 to S0CH31 are set to "0".
4. MCR2REG register settings
   a. When SBFN is set to "0", this does not need to be set.
      When SBFN is set to "1", S1CH00 is set to "1", and S1CH01 to S1CH31 are set to "0".

5. OPRREG register settings
   a. Set the START bit of the OPRREG register to "1".
   b. Write "1" to the TXENB and RXENB bits of the OPRREG register.
6. INTCNT register settings
   a. Set the threshold value for the receive and transmit FIFO to RFTH and TFTH.
      RFTH and TFTH are set to the same value as DES1:IIN of the DSTC controller.
   b. Set RXFDM and TXFDM.
      Set RXFDM and TXFDM to "0".
   c. Set FERRM, TXUDM, TXOVM, RXUD0M, and RXOVM to "0".
7. DSTC controller register settings
   a. Set DREQENB to "1".
   b. Set the DES2:SA (Source Address) of the receive channel to the address of the RXFDAT register.
      Set the DES3:DA (Destination Address) of the transmit channel to the address of the TXFDAT register.
   c. Set DES0.
   d. Set DES1.
   e. For information about other detailed settings, see the chapter "DSTC".
8. DMAACT register settings
   a. Set the corresponding bit (TL1E0 bit for the transmit channel and RL1E0 bit for the receive channel) to "1".

Note:

- *Every time transfer of a DMA packet (DES1:ORM becomes "0", HWINT becomes "1", and the channel becomes invalid) is completed, set this back to number 8.*
- *When the I2S interface is in master mode and data transfer is not performed, set the START bit of the OPRREG register to "0" (this reduces power consumption).*

FIFO Word Configuration by RHLL Bit Settings

The internal FIFO word configuration will vary based on the RHLL bit set in the CNTREG register.

RHLL="1" (Figure 7.1-4)

One FIFO word corresponds to one I2S frame. The lower 16 bits of FIFO are for the left channel, and the upper 16 bits are for the right channel. The PCM can have a width of up to 16 bits.

When the word length set for the MCR0REG register is larger than 16 bits:

Transmit: Transmit extension bits following the 16-bit channel data until the set word length is reached. (Extend using "0" or the sign bit depending on the mode set for the BEXT bit of the CNTREG register.)

Receive: For words that are received from the serial bus, count 16 bits from the MSB to form the channel data, and ignore the remaining data (received word length - 16 bits).

When the word length set for the MCR0REG register is smaller than 16 bits:

Transmit: Transmit data up to the preset word length counting from the LSB (bit 0) of the 16-bit channel data.

Receive: LSB-justify the word length, and extend the upper bits (16 bits - word length). (Extend using "0" or the sign bit depending on the mode set for the BEXT bit of the CNTREG register.)

RHLL="0" (Figure 7.1-5)

One FIFO word will correspond to one I2S frame channel. The PCM can have a width of up to 32 bits.

When the word length set for the MCR0REG register is smaller than 32 bits:

Transmit: Transmit data up to the preset word length counting from the LSB (bit 0) of the 32-bit FIFO word.

Receive: LSB-justify the word length, and extend the upper bits (32 bits - word length). (Extend using "0" or the sign bit depending on the mode set for the BEXT bit of the CNTREG register.)

Figure 7.1-4 Data Structure when RHLL is "1"

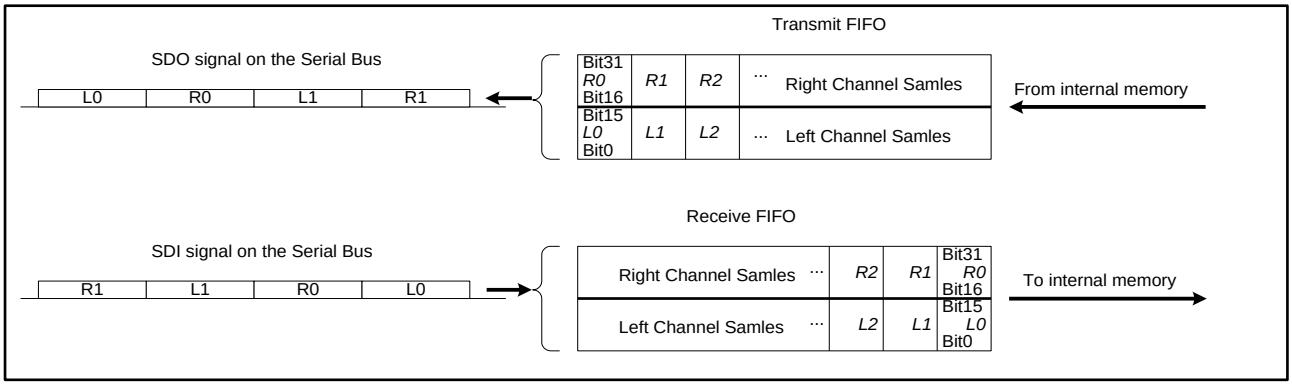
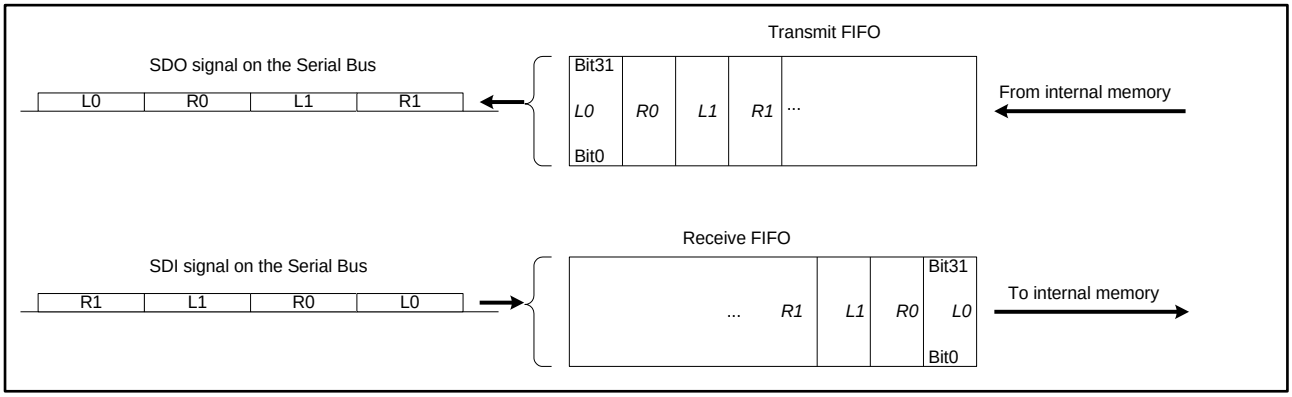


Figure 7.1-5 Data Structure when RHLL is "0"



Error Processing

- Frame Error

When the FERR bit of the STATUS register is set to "1", this indicates that the frame sync signal did not occur at the correct timing.
Set the FERR bit of the STATUS register to "1".

- Receive FIFO Overflow

When the receive FIFO is full, this indicates that there was a write operation to the receive FIFO. This means that the internal process was not completed in time. Words received when the receive FIFO is full are discarded.
Set the RXOVR bit of the STATUS register to "1".

- Receive FIFO Underflow

This means that there was a read access to the receive FIFO when the receive FIFO was empty.

- Transmit FIFO Overflow

This means that there was a write operation to the transmit FIFO from the CPU or DMA when the transmit FIFO was full. This means that the software is not running properly.
Set the TXOVR bit of the STATUS register to "1".

- Transmit FIFO Underflow

This indicates that there was a transmit request when the transmit FIFO was empty.
Set the TXUDR0 bit or TXUDR1 bit of the STATUS register to "1" and output an empty frame bit. (For details of the setting conditions for the TXUDR0 and TXUDR1 bits, see the explanation of bits TXUDR0 and TXUDR1 in "6.10 Status Register (STATUS)".)

CHAPTER 8-1: High-Speed Quad Serial Peripheral Interface Configuration

The chapter explains the configuration of HS_SPI (High-Speed Quad Serial Peripheral Interface).

1. Configuration of HS_SPI

1. Configuration of HS_SPI

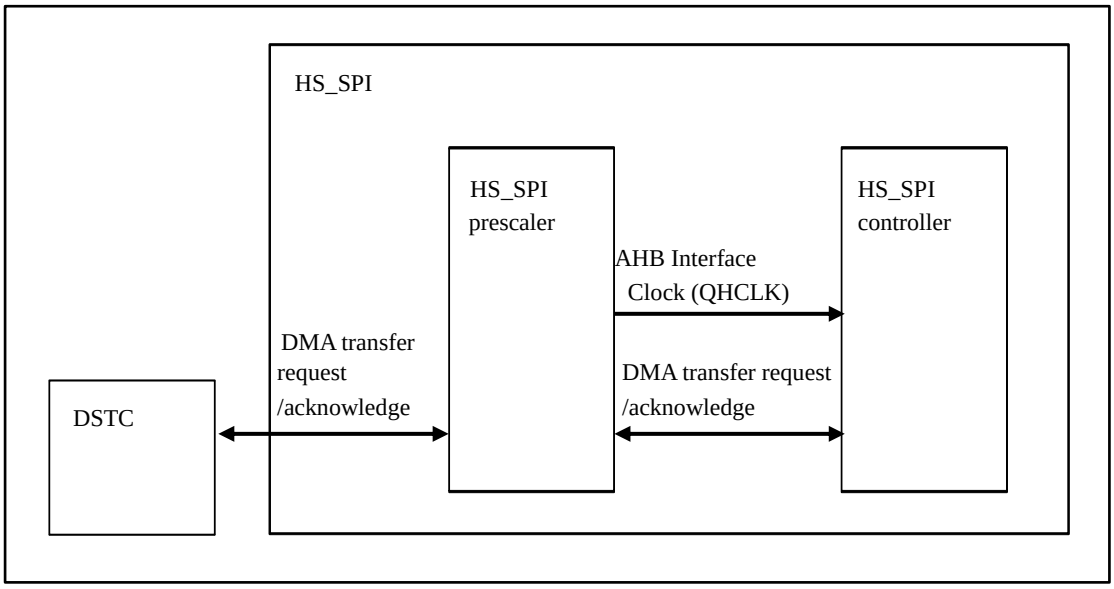
The HS_SPI consists of the HS_SPI controller and the prescaler of it.

The HS_SPI prescaler provides for AHB Interface Clock of HS_SPI controller, and has DMA BRIDGE funciton.

The HS-SPICNT supports various SPI devices (Serial Peripheral devices).

Figure 1-1 shows the configuration of HS_SPI.

Figure 1-1 HS_SPI configuration



CHAPTER 8-2: High-Speed Quad Serial Peripheral Interface Prescaler

This section explains the HS-SPI (High-Speed Quad Serial Peripheral Interface) prescaler.

1. **Overview**
2. **Explanation of Operations**
3. **Register set of the HS-SPI prescaler**

Management Code: 9xFBAD12M3_FM4-J01.0

1. Overview

HS_SPI consists of clock divider, SYNC DOWN BRIDGE, and DMA BRIDGE.

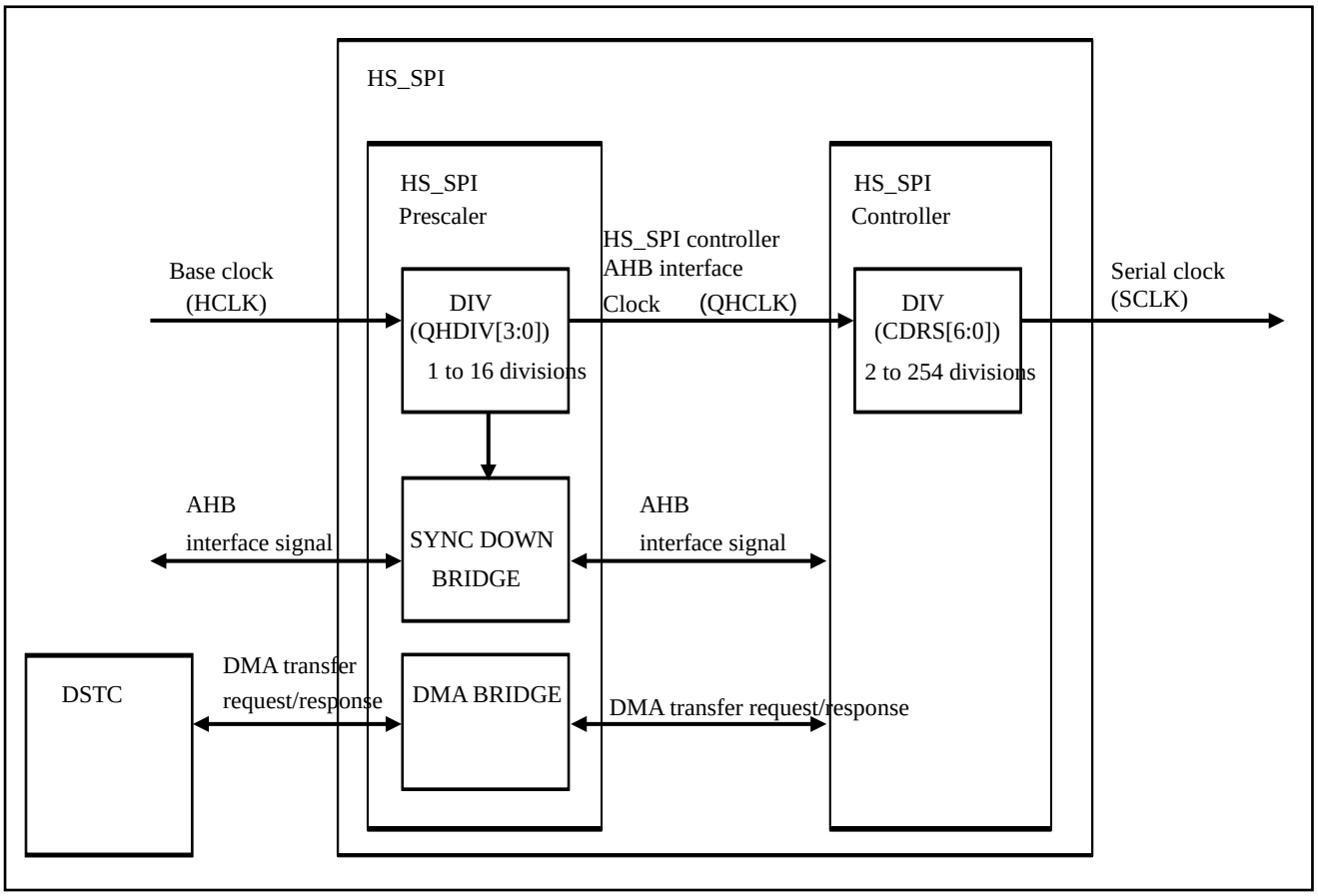
The clock divider divides the base clock (HCLK) by 1 to 16 divisions, and supplies the AHB interface clock (QHCLK) of the HS_SPI controller.

When the clock divider setting is 2 divisions or more, Sync Down Bridge applies Sync Down to the AHB interface signal.

DMA BRIDGE is connected to the DMA interface of the HS_SPI controller and it controls the DMA transfer between DMA controllers.

Figure 1-1 shows the block diagram of the HS-SPI prescaler.

Figure 1-1 Block diagram of the HS-SPI prescaler



2. Explanation of Operations

This section explains the operation of the HS_SPI prescaler.

2.1 Clock setting

The operation frequency of the serial clock (SCLK) of SPI is determined according to the combination of the base clock (HCLK) frequency, the HS_SPI prescaler division setting (QHDIV[3:0]), the HS_SPI controller division setting (CDRS[6:0]).

■ Base clock (HCLK)

Base clock (HCLK) is the clock supplied to HS_SPI.

■ HS_SPI controller AHB interface clock (QHCLK)

HS_SPI controller AHB interface clock (QHCLK) is the system clock of the HS_SPI controller, and the HS_SPI prescaler supplies the base clock (HCLK) divided by 1 to 16 divisions.

Default setting is 16 divisions.

■ Serial clock (SCLK)

Serial clock (SCLK) is the serial clock of SPI.

The HS_SPI controller supplies the AHB interface clock of the HS_SPI controller divided by 2 to 254 divisions.

Note:

- *The default setting of the HS_SPI controller division setting (CDRS[6:0]) is Reserved (no division). Since the HS_SPI controller prohibits the setting of no division, user must change the division setting before enabling the HS_SPI controller operation.*
  For details, refer to the "HS_SPI Controller" chapter.

For the specification range of each clock, refer to the usage conditions of "Datasheet" provided with the product you are using.

Table 2-1 shows a division setting example.

Table 2-1 Division setting example of clock

Base clock (HCLK)	QHDIV[3:0]	HS_SPI controller AHB interface clock (QHCLK)	CDRS[6:0]	Serial clock (SCLK)
200MHz	1	100MHz	1	50MHz
160MHz	1	80MHz	1	40MHz
132MHz	0	132MHz	1	66MHz
80MHz	0	80MHz	1	40MHz
66MHz	0	66MHz	1	33MHz

2.2 DMA BRIDGE control

DMA BRIDGE is connected to the DMA interface of the HS_SPI controller and it controls the DMA transfer between DMA controllers.

The DMA BRIDGE control register (DBCNT) must be set when the HS_SPI controller is used with the direct mode DMA transfer.

Notes:

Be sure to pay attention to the following points when using DMA transfer. (These are described as points for receiving side, but they are also applied to transmitting side.)

- *Set DBCNT:RXDBEN to "1" before setting HSSPIn_DMDMAEN:RXDMAEN to "1" for DMA transfer.*
- *After DMA transfer, set HSSPIn_DMDMAEN:RXDMAEN to "0", then set DBCNT:RXDBEN to "0".*
- *For details of setting procedure example of direct mode DMA transfer, refer to the "HS_SPI Controller" chapter.*
- *Accessing from CPU to RX-FIFO is prohibited when DBCNT:RXDBEN is set to "1" because it will cause inconsistency on the buffer state of DMA BRIDGE. In order to enable an access from CPU to RX-FIFO, stop the DMA transfer, set HSSPIn_DMDMAEN:RXDMAEN to "0" and set DBCNT:RXDBEN to "0".*

3. Register set of the HS-SPI prescaler

This section explains the registers of the HS-SPI prescaler

3.1 Registers

This section provides a list of registers.

List of registers for the HS-SPI prescaler

Abbreviated register name	Register name	See
QDCLKR	Division clock register	3.2
DBCNT	DMA BRIDGE control register	3.3

3.2 Division Clock Register (QDCLKR)

This register sets the division ratio of the HS-SPI system clock.

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	QHDIV[3]	QHDIV[2]	QHDIV[1]	QHDIV[0]
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	1	1	1

[bit7:4] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit3:0] QHDIV[3:0] : HS_SPI clock division ratio setting bits

bit	Description
0000	1 division
0001	2 divisions
0010	3 divisions
…	…
1111	16 divisions [initial value]

This bit sets the division ratio (1 to 16 divisions) of the HS_SPI controller.

This is set to (QHDIV+1) divisions.

When this bit is changed, the read value of this bit is updated after the write value is reflected to the clock divider. Perform accessing the HS_SPI controller area after making sure that the read value becomes the same value as the write value.

Notes:

- *For details of frequency of the HS_SPI controller AHB interface clock which can be set, refer to "Data Sheet" provided with the product you are using.*
- *Accessing the HS_SPI controller area is prohibited between the write access to QHDIV and the update of QHDIV read value to the same value as the write value. If any access is performed, the access is not guaranteed.*

3.3 DMA BRIDGE Control Register (DBCNT)

DMA BRIDGE is connected to the DMA interface of the HS_SPI controller and it controls the DMA transfer between DMA controllers.

This register enables/disables the DMA BRIDGE.

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXDBEN	RXDBEN
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit1] TXDBEN: DMA BRIDGE control bit for transmitting

bit	Description
0	Disables the DMA BRIDGE for transmitting. [Initial value]
1	Enables the DMA BRIDGE for transmitting.

Notes:

- *This register setting can stop the DMA BRIDGE operations, however, it cannot suppress the DMA transfer request or the DMA transfer response.*
- *If TXDBEN is changed during DMA transfer, the DMA transfer is not guaranteed.*
- *Change TXDBEN when HSSPIn_DMDMAEN:TXDMAEN is "0".*
  For details of HSSPIn_DMDMAEN, refer to "HS-SPI Direct Mode DMA Enable Register (HSSPIn_DMDMAEN)".

[bit1] RXDBEN: DMA BRIDGE control bit for receiving

bit	Description
0	Disables the DMA BRIDGE for receiving. [Initial value]
1	Enables the DMA BRIDGE for receiving.

Notes:

- *This register setting can stop the DMA BRIDGE operations, however, it cannot suppress the DMA transfer request or the DMA transfer response.*
- *If RXDBEN is changed during DMA transfer, the DMA transfer is not guaranteed.*
- *Change RXDBEN when HSSPIn_DMDMAEN:RXDMAEN is "0".*
  For details of HSSPIn_DMDMAEN, refer to "HS-SPI Direct Mode DMA Enable Register (HSSPIn_DMDMAEN)".

CHAPTER 8-3: High-Speed Quad Serial Peripheral Interface Controller

This section explains the functions and operations of the HS_SPI (High-Speed Quad Serial Peripheral Interface) controller.

1. **Overview of the HS_SPI controller**
2. **HS_SPI Controller Operations**

1. Overview of the HS_SPI controller

The HS_SPI controller is compatible with various SPI devices (Serial Peripheral Interface devices). This chapter explains the overview of the HS_SPI controller.

1.1 Features of the HS_SPI controller

■ Supports two operation modes (direct mode and command sequencer mode)

However, the command sequencer mode is not compatible with all SPI devices. Therefore, refer to the data sheet provided with the product you are using in order to check if your device is compatible with this mode. For details of direct mode and command sequencer mode, refer to "2.4. Direct mode" and "2.5. Command Sequencer Mode".

■ Compatible with single bit, dual bit and quad bit.

For details, refer to "2.3.2Serial Interface Bit Width" in "2.3Serial Interface".

■ Compatible with up to four slave devices

Note:

- *The maximum number of the slave devices varies depending on the products you are using. Refer to "Data Sheet".*

■ Can set format of serial interface for each slave.

Only for direct mode. For details, refer to "3.3. HS-SPI peripheral communication set register (HSSPIn_PCC0 to 3)".

■ Supports four clocking modes.

For details, refer to "2.3.1Clocking Mode" in "2.3Serial Interface".

■ Realizes the Hold function of serial flash by stopping the serial clock.

The HS_SPI controller realizes this function by stopping the serial clock while some other serial flashes support the Hold function of the serial bus using the HOLD pin.

■ Supports DMA transfer.

For details, refer to "2.1DMA Interface".

2. HS_SPI Controller Operations

This chapter explains the HS_SPI controller operations.

The followings are explained.

2.1. DMA Interface

2.2. Interrupts

2.3. Serial Interface

2.4. Direct mode

2.5. Command Sequencer Mode

2.1 DMA Interface

The HS_SPI has DMA transfer function with hardware start-up. In direct mode, it is used for data write/read with TXFIFO and RXFIFO.

2.1.1 DMA transfer toward TX-FIFO

When all conditions below are met, DMA transfer toward TX-FIFO will be requested.

- DMA for transmitting is enabled (HSSPIn_DMDMAEN.TXDMAEN="1").
- Number of valid data of TX-FIFO equals to the threshold or less (HSSPIn_TXF.TFLETS="1").
- The block counter*1 for transmitting is 0.
  (When DMA transfer is requested to TX-FIFO, (16 - HSSPIn_FIFOCFG.TXFTH) is loaded, and -1 is applied to the block counter for transmitting each time data is written to TX-FIFO. When the HSSPIn_DMDMAEN.TXDMAEN bit is "0", the block counter for transmitting is 0.)
- The DMA write block size violation detection bit is "0" (HSSPIn_FAULTF. DWCBSFS="0").
- The operations of the HS_SPI controller are enabled (HSSPIn_MCTRL.MES="1").
- The HS_SPI controller is set to direct mode (HSSPIn_MCTRL.CSEN="0").
- The HSSPIn_DMTRP.TRP[3:2] bit is set to TX-and-RX(0x0) or TX-Only(0x2).

Notes:

- *The block indicates one access (data transfer).*
- *The block counter for transmitting counts the number of times of data transfer.*

DMA transfer request to TX-FIFO will be disabled by one of the following conditions.

- DMA for transmitting is disabled (HSSPIn_DMDMAEN.TXDMAEN="0").
- The operations of the HS_SPI controller are disabled (HSSPIn_MCTRL.MES="0").
- The HS_SPI controller is set to command sequencer mode (HSSPIn_MCTRL.CSEN="1").
- DMA transfer response for transmitting is returned.

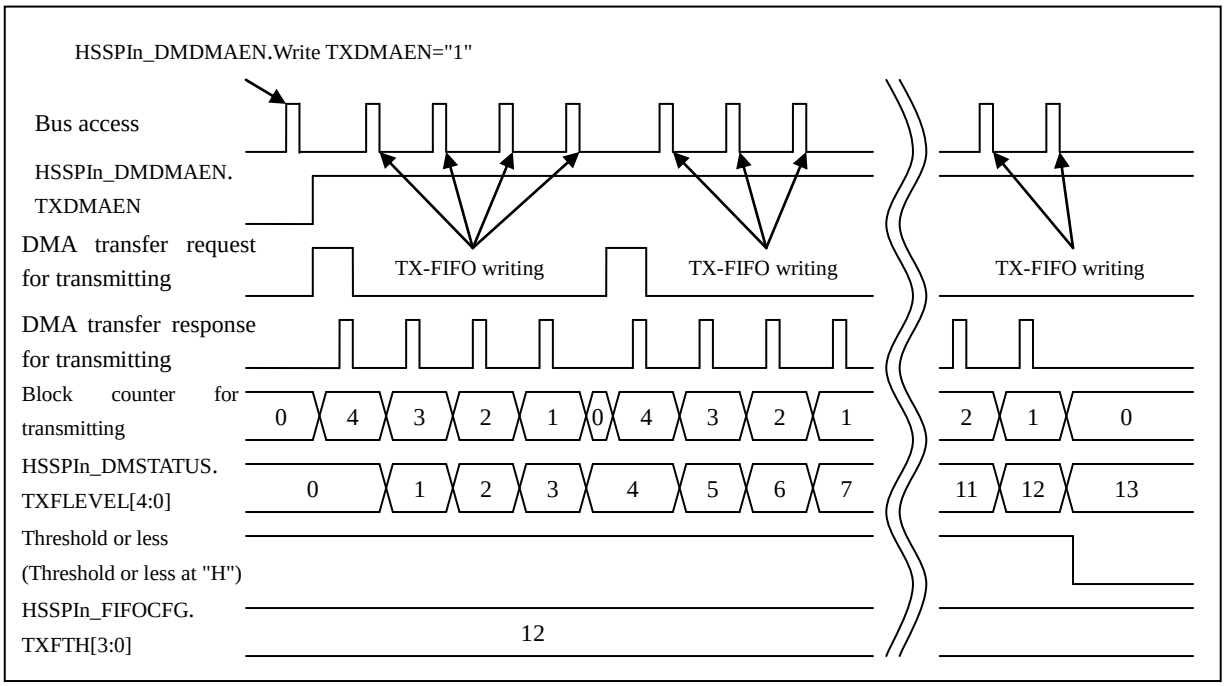
The HS_SPI controller sets the DMA write block size violation detection bit (HSSPIn_FAULTF. DWCBSFS) to "1" and issues a violation interrupt request if the block size setting (16 - HSSPIn_FIFOCFG.TXFTH) and the data transfer count setting (DES1.IIN[15:0]) of DSTC are different for DMA transfer for transmitting.

When all conditions below are met, the DMA write block size violation detection bit (HSSPIn_FAULTF. DWCBSFS) is set to "1".

- Data is written to TX-FIFO when the block counter for transmitting is 0.
- DMA for transmitting is enabled (HSSPIn_DMDMAEN.TXDMAEN="1").
- The operations of the HS_SPI controller are enabled (HSSPIn_MCTRL.MES="1").
- The HS_SPI controller is set to direct mode (HSSPIn_MCTRL.CSEN="0").

The following shows operation example of DMA transfer for TX-FIFO.

Figure 2-1 Operation example of DMA transfer for TX-FIFO (HSSPIn_FIFOCFG.TXFTH=12)



The following explains an operation example of DMA transfer shown above.

- Because the block counter for transmitting is 0 and the value which equals to the threshold or less is "H" when DMA transfer for transmitting is enabled (HSSPIn_DMDMAEN.TXDMAEN="1"), the DMA transfer request for transmitting is set to "H", and (16 – HSSPIn_FIFOCFG.TXFTH) is loaded to the block counter for transmitting.
- Because DMA transfer is requested, DSTC returns DMA transfer response for transmitting. The DMA transfer response for transmitting is changed to "H" and the DMA transfer request for transmitting is changed to "L".
- When DMA transfer is started and the transmission data is written to TX-FIFO, -1 is applied to the block counter for transmitting.
- Because the block length is four, when the transmission data is written to TX-FIFO four times, the block counter for transmitting is changed to 0. Because of the value which equals to the threshold or less is "H" at that time, set the DMA transfer request for transmitting to "H" again, and load (16 - HSSPIn_FIFOCFG.TXFTH) to the block counter for transmitting.
- Repeat this procedure, and if the value which equals to the threshold or less is "L" when the block counter for transmitting becomes 0, the DMA transfer request for transmitting holds the "L" state until the value is changed to "H".

2.1.2 DMA transfer toward RX-FIFO

When all conditions below are met, DMA transfer toward RX-FIFO will be requested.

- DMA for receiving is enabled (HSSPIn_DMDMAEN.RXDMAEN="1").
- Number of valid RX-FIFO data exceeds the threshold.
- The block counter*1 for receiving is 0.

  (When DMA transfer is requested to RX-FIFO, (HSSPIn_FIFOCFG.RXFTH+1) is loaded, and -1 is applied to the block counter for receiving each time data is read from RX-FIFO. When the HSSPIn_DMDMAEN.RXDMAEN bit is "0", the block counter for receiving is 0.)
- The DMA read block size violation detection bit is "0" (HSSPIn_FAULTF. DRCBSFS="0").
- The operations of the HS_SPI controller are enabled (HSSPIn_MCTRL.MES="1").
- The HS_SPI controller is set to direct mode (HSSPIn_MCTRL.CSEN="0").
- The HSSPIn_DMTRP.TRP[3:2] bit is set to TX-and-RX(0x0) or RX-Only(0x1).

Notes:

- *The block indicates one access (data transfer).*
- *The block counter for receiving counts the number of times of data transfer.*

DMA transfer request to RX-FIFO will be disabled by one of the following conditions.

- DMA for receiving is disabled (HSSPIn_DMDMAEN.RXDMAEN="0").
- The operations of the HS_SPI controller are disabled (HSSPIn_MCTRL.MES="0").
- The HS_SPI controller is set to command sequencer mode (HSSPIn_MCTRL.CSEN="1").
- DMA transfer response for receiving is returned.

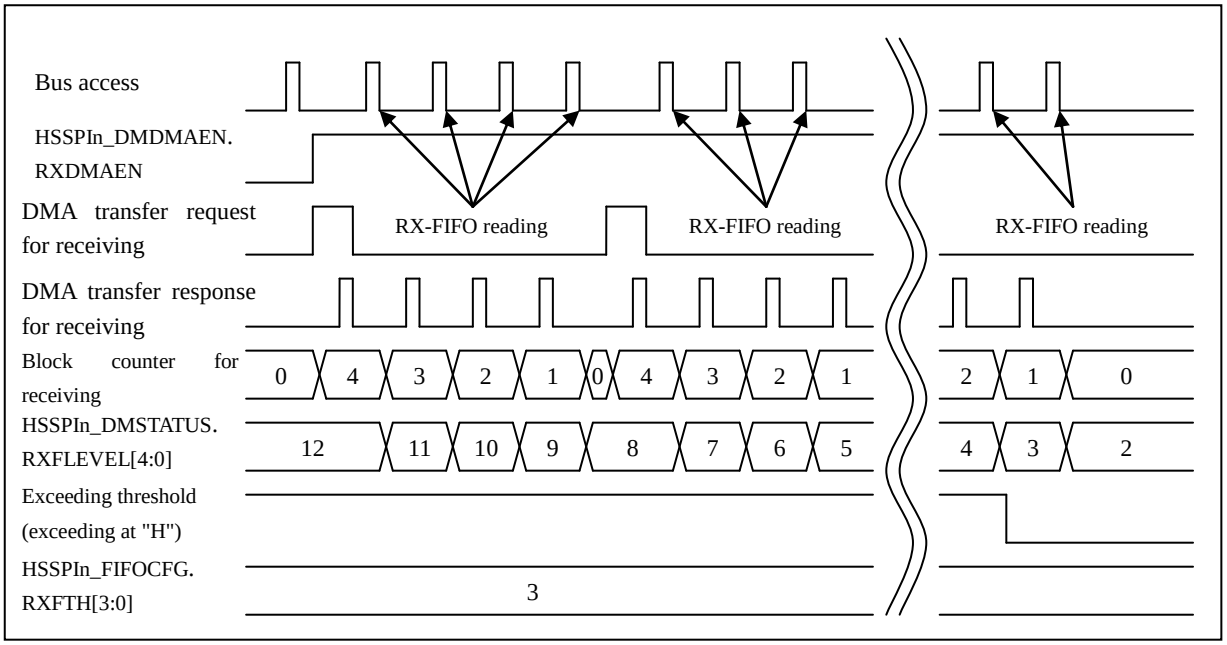
The HS_SPI controller sets the DMA read block size violation detection bit (HSSPIn_FAULTF. DRCBSFS) to "1" and issues a violation interrupt request if the block size setting (HSSPIn_FIFOCFG.RXFTH+1) and the data transfer count setting (DES1.IIN[15:0]) of DSTC are different for DMA transfer for receiving.

When all conditions below are met, the DMA read block size violation detection bit (HSSPIn_FAULTF. DRCBSFS) is set to "1".

- Data is read from RX-FIFO when the block counter for receiving is 0.
- DMA for receiving is enabled (HSSPIn_DMDMAEN.RXDMAEN="1").
- The operations of the HS_SPI controller are enabled (HSSPIn_MCTRL.MES="1").
- The HS_SPI controller is set to direct mode (HSSPIn_MCTRL.CSEN="0").

The following shows operation example of DMA transfer for RX-FIFO.

Figure 2-2 Operation example of DMA transfer for RX-FIFO (HSSPIn_FIFOCFG.RXFTH=3)



The following explains an operation example of DMA transfer shown above.

- Because the block counter for receiving is 0 and the value which exceeds the threshold is "H" when DMA transfer for receiving is enabled (HSSPIn_DMDMAEN.RXDMAEN="1"), the DMA transfer request for receiving is set to "H", and (HSSPIn_FIFOCFG.RXFTH+1) is loaded to the block counter for receiving.
- Because DMA transfer is requested, DSTC returns DMA transfer response for receiving. The DMA transfer response for receiving is changed to "H" and the DMA transfer request for receiving is changed to "L".
- When DMA transfer is started and the receive data is read from RX-FIFO, -1 is applied to the block counter for receiving.
- Because the block length is four, when the receive data is read from RX-FIFO four times, the block counter for receiving is changed to 0. Because of the value which exceeds the threshold is "H" at that time, set the DMA transfer request for receiving to "H" again, and load (HSSPIn_FIFOCFG.RXFTH+1) to the block counter for receiving.
- Repeat this procedure, and if the value which exceeds the threshold is "L" when the block counter for receiving becomes 0, the DMA transfer request for receiving holds the "L" state until the value is changed to "H".

2.2 Interrupts

The HS_SPI controller supports the transmission interrupt requests, receive interrupt requests and violation interrupt requests.

2.2.1 Transmission interrupt requests

When one of the following transmission interrupt factors is "1" and the corresponding transmission interrupt enable bit is "1", a transmission interrupt request is generated.

Table 2-1 Transmission interrupt factors and transmission interrupt enable

Transmission interrupt factor		Transmission interrupt enable		Function
Register name	Bit name	Register name	Bit name	
HSSPIn_TXF	TSSRS="1"	HSSPIn_TXE	TSSRE="1"	This is the interrupt generated by canceling the slave selection. When the slave selection is canceled, the HSSPIn_TXF.TSSRS bit is set to "1". When "1" is written to the HSSPIn_TXC.TSSRC bit, the HSSPIn_TXF.TSSRS bit is set to "0".
	TFMTS="1"		TFMTE="1"	This is the interrupt generated by exceeding the TX-FIFO threshold. When the valid TX-FIFO data exceeds the threshold, the HSSPIn_TXF.TFMTS bit is set to "1". When "1" is written to the HSSPIn_TXC.TFMTC bit, the HSSPIn_TXF.TFMTS bit is set to "0".
	TFLETS="1"		TFLETE="1"	This is the interrupt generated by falling below the TX-FIFO threshold. When the valid TX-FIFO data falls below the threshold, the HSSPIn_TXF.TFLETS bit is set to "1". When "1" is written to the HSSPIn_TXC.TFLETC bit, the HSSPIn_TXF.TFLETS bit is set to "0".
	TFUS="1"		TFUE="1"	This is the interrupt generated by the TX-FIFO underrun. When TX-FIFO is read while TX-FIFO is empty, the HSSPIn_TXF.TFUS bit is set to "1". When "1" is written to the HSSPIn_TXC.TFUC bit, the interrupt factor is set to "0".
	TFOS="1"		TFOE="1"	This is the interrupt generated by the TX-FIFO overrun. When TX-FIFO is written while TX-FIFO is full, the HSSPIn_TXF.TFOS bit is set to "1". When "1" is written to the HSSPIn_TXC.TFOC bit, the HSSPIn_TXF.TFOS bit is set to "0".
	TFES="1"		TFEE="1"	This is the interrupt generated when TX-FIFO and the transmission shifter are empty. When TX-FIFO is empty while the transmission shifter is empty, the HSSPIn_TXF.TFES bit is set to "1". When "1" is written to the HSSPIn_TXC.TFEC bit, the HSSPIn_TXF.TFES bit is set to "0".
	TFFS="1"		TFFE="1"	This is the interrupt generated when TX_FIFO is full. When TX_FIFO is full, the HSSPIn_TXF.TFFS bit is set to "1". When "1" is written to the HSSPIn_TXC.TFFC bit, the HSSPn_TXF.TFFS bit is set to "0".

2.2.2 Receive interrupt requests

When one of the following receive interrupt factors is "1" and the corresponding receive interrupt enable bit is "1", a receive interrupt request is generated.

Table 2-2 Receive interrupt factors and receive interrupt enable

Receive interrupt factor		Receive interrupt enable		Function
Register name	Bit name	Register name	Bit name	
HSSPIn_RXF	RSSRS="1"	HSSPIn_RXE	RSSRE="1"	This is the interrupt generated by canceling the slave selection. When the slave selection is canceled, the HSSPIn_RXF.RSSRS bit is set to "1". When "1" is written to the HSSPIn_RXC.RSSRC bit, the HSSPIn_RXF.RSSRS bit is set to "0".
	RFMTS="1"		RFMTE="1"	This is the interrupt generated by exceeding the RX-FIFO threshold. When the valid RX-FIFO data exceeds the threshold, the HSSPIn_RXF.RFMTS bit is set to "1". When "1" is written to the HSSPIn_RXC.RFMTC bit, the HSSPIn_RXF.RFMTS bit is set to "0".
	RFLETS="1"		RFLETE="1"	This is the interrupt generated by falling below the RX-FIFO threshold. When the valid RX-FIFO data falls below the threshold, the HSSPIn_RXF.RFLETS bit is set to "1". When "1" is written to the HSSPIn_RXC.RFLETC bit, the HSSPIn_RXF.RFLETS bit is set to "0".
	RFUS="1"		RFUE="1"	This is the interrupt generated by the RX-FIFO underrun. When RX-FIFO is read while RX-FIFO is empty, the HSSPIn_RXF.RFUS bit is set to "1". When "1" is written to the HSSPIn_RXC.RFUC bit, the interrupt factor is set to "0".
	RFOS="1"		RFOE="1"	This is the interrupt generated by the RX-FIFO overrun. When RX-FIFO is written while RX-FIFO is full, the HSSPIn_RXF.RFOS bit is set to "1". When "1" is written to the HSSPIn_RXC.RFOC bit, the HSSPIn_RXF.RFOS bit is set to "0".
	RFES="1"		RFEE="1"	This is the interrupt generated when RX-FIFO is empty. When RX-FIFO is empty, the HSSPIn_RXF.RFES bit is set to "1". When "1" is written to the HSSPIn_RXC.RFEC bit, the HSSPIn_RXF.RFES bit is set to "0".
	RFFS="1"		RFFE="1"	This is the interrupt generated when RX_FIFO is full. When RX_FIFO is full, the HSSPIn_RXF.RFFS bit is set to "1". When "1" is written to the HSSPIn_RXC.RFFC bit, the HSSPn_RXF.RFFS bit is set to "0".

2.2.3 Violation detection interrupt requests

Because of the following violation interrupt factors, a violation interrupt request is generated. The HS_SPI cannot prohibit this interrupt.

Table 2-3 Violation detection interrupt factors

Violation interrupt factor		Function
Register name	Bit name	
HSSPIn_FAULTF	DRCBSFS="1"	This is the interrupt generated by the DMA read block length violation. When the receive block length of DMA transfer does not match the DSTC block length, the HSSPIn_FAULTF.DRCBSFS bit is set to "1". When "1" is written to the HSSPIn_FAULTC.DRCBSFC bit, the HSSPIn_FAULTF.DRCBSFS bit is set to "0".
	DWCBSFS="1"	This is the interrupt generated by the DMA write block length violation. When the transmission block length of DMA transfer does not match the DSTC block length, the HSSPIn_FAULTF.DWCBSFS bit is set to "1". When "1" is written to the HSSPIn_FAULTC.DWCBSFC bit, the HSSPIn_FAULTF.DWCBSFS bit is set to "0".
	WAFS="1"	This is the interrupt generated by the write access violation. In command sequencer mode, the HSSPIn_FAULTF.WAFS bit is set to "1" when the HSSPIn_CSCFG.SRAM bit is set to "0" (write prohibited) and write access is attempted from the HS_SPI controller to serial devices. When "1" is written to the HSSPIn_FAULTC.WAFC bit, the HSSPIn_FAULTF.WAFS bit is set to "0".
	UMAFS="1"	This is the interrupt generated by the memory access violation. When an access is attempted to the memory area which is not allocated, the HSSPIn_FAULTF.UMAFS bit is set to "1". When "1" is written to the HSSPIn_FAULTC.UMAFC bit, the HSSPIn_FAULTF.UMAFS bit is set to "0".

2.3 Serial Interface

The HS_SPI controller supports four modes as the clocking modes.

The HS_SPI controller supports master only.

2.3.1 Clocking Mode

Four clocking modes can be set by setting each bit of HSSPIn_PCC0 to 3.ACES, CPOL or CPHA. Table 2-4 shows the definitions of four clocking modes.

Table 2-4 Clocking mode

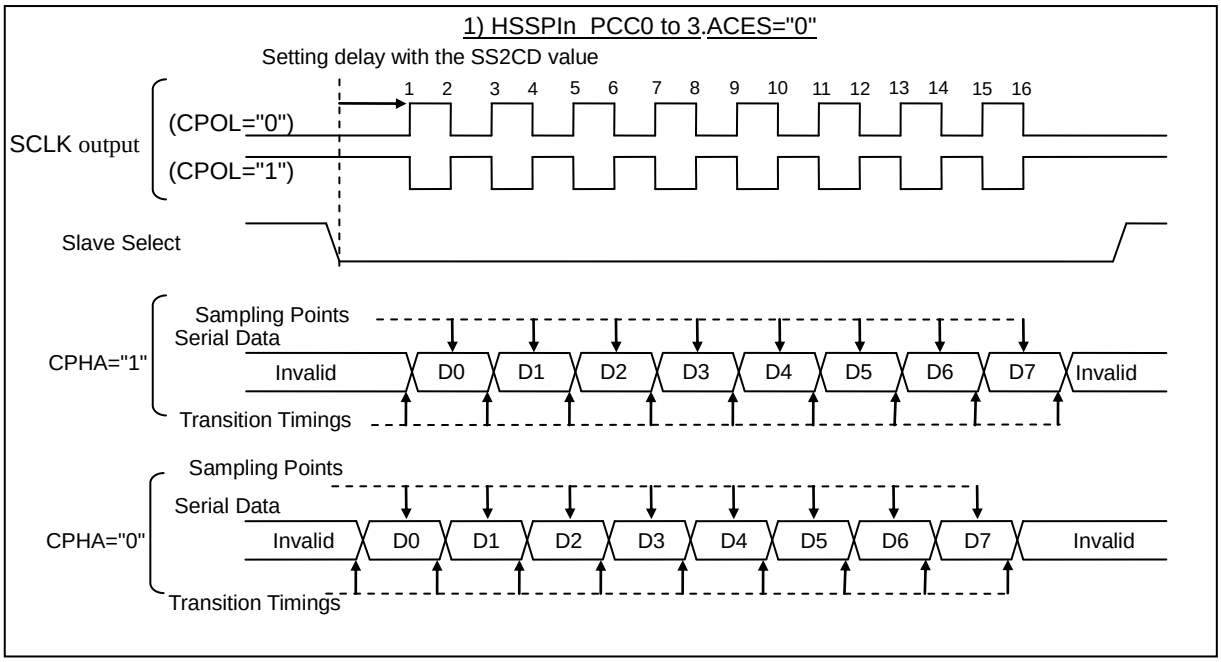
Mode	HSSPIn_PCC0 to 3 register			Description
	ACES bit	CPOL bit	CPHA bit	
Mode 0	0	0	0	The output data changes at the half cycle before the first rising edge of the serial clock, and then it changes at the falling edge of this serial clock. The input data is captured at the rising edge of the same serial clock.
Mode 1		0	1	The output data changes at the rising edge of the serial clock. The input data is captured at the falling edge of the same serial clock.
Mode 2		1	0	The output data changes at the half cycle before the first falling edge of the serial clock, and then it changes at the rising edge of this serial clock. The input data is captured at the falling edge of the same serial clock.
Mode 3		1	1	The output data changes at the falling edge of the serial clock. The input data is captured at the rising edge of the same serial clock.
Any settings other than the settings above				Not compatible

Figure 2-3 shows the relationship among the settings for each control bit of ACES/CPOL/CPHA and serial data/serial clock.

Note:

- *In command sequencer mode, use the HSSPIn_PCC0 to 3 registers with same settings.*

Figure 2-3 Operation wave form of serial interface clock



2.3.2 Serial Interface Bit Width

The HS_SPI controller is compatible with the following serial interface bit width.

- Single bit
- Dual bit
- Quad bit

For examples of operation wave form, refer to "Figure 2-4".

1) Single bit

The single bit is a communication protocol of full-duplex line. When the HS_SPI is set to use the single bit, the serial data is received from SDATA[1] with one bit width, and it is transmitted from SDATA[0] at the same time.

2) Dual bit

In dual mode, the 2-bit width data line (SDATA[1:0]) with the half-duplex line is used. Data transmission/receive operation is performed exclusively.

3) Quad bit

In quad mode, the 4-bit width data line (SDATA[3:0]) with the half-duplex line is used. Data transmission/receive operation is performed exclusively.

2.3.3 Shift direction

The HSSPIn_PCC0 to 3.SDIR bits of the HS_SPI controller determine the shift direction of the shift register used for data transfer.

When the HSSPIn_PCC0 to 3.SDIR bits are "0" (MSB First), the highest bit is transferred at first and the lowest bit is transferred at last, in the order from upper bits to lower bits. When displaying the bits from the highest bit to lowest bit on a line from from left to right, the shift direction is left direction.

When the HSSPIn_PCC0 to 3.SDIR bits are "1" (LSB First), the shift direction is right direction with the bit line in same manner.

Figure 2-4 and Figure 2-5 show operating examples of the shift register for the single bit/dual bit/quad bit and transmission FIFO (TX-FIFO)/receive FIFO(RX-FIFO).

Figure 2-4 Shift direction (when settings are CPOL="0", CPHA="0", SDIR="0", and FWIDTH="00")

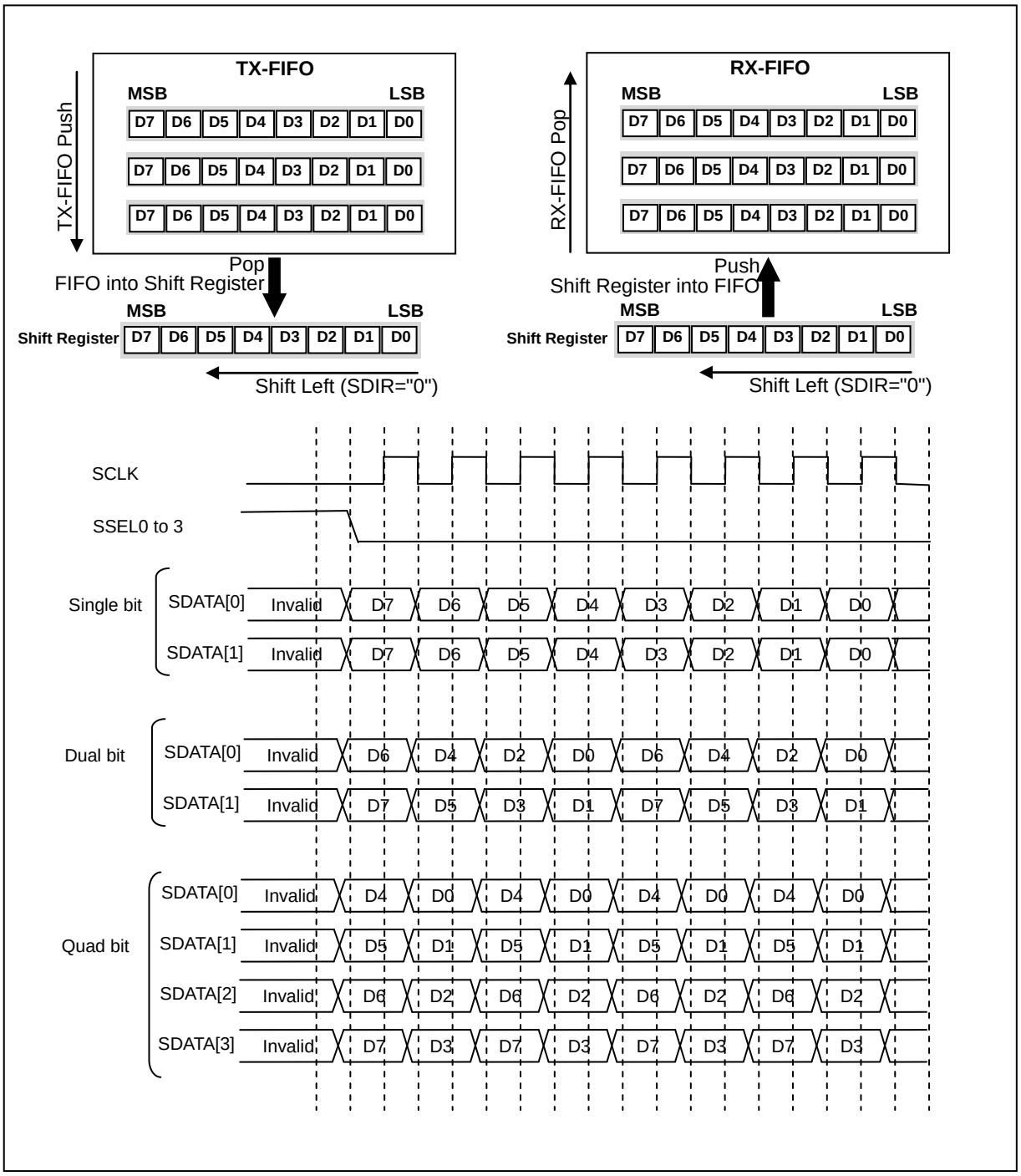
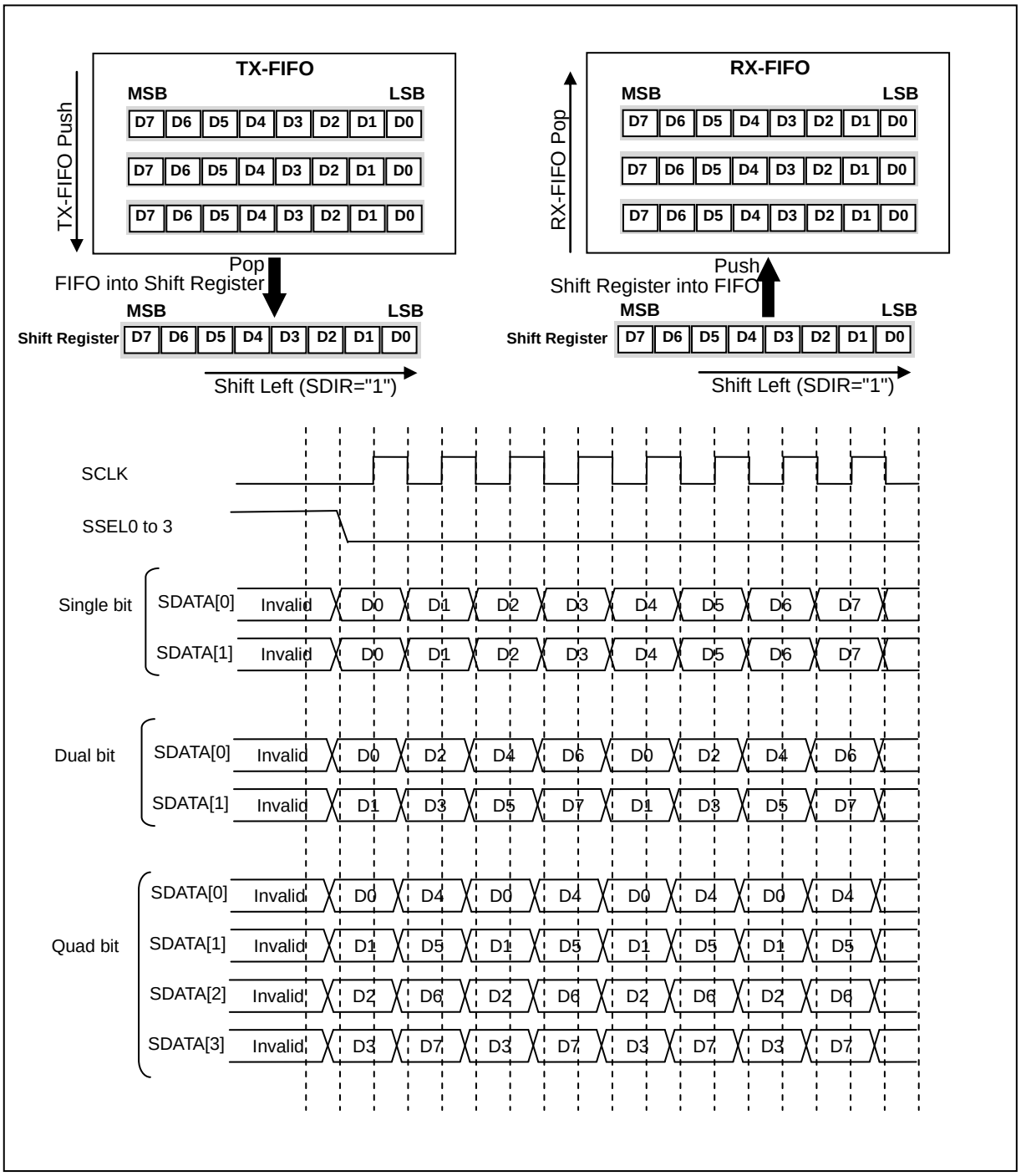


Figure 2-5 Shift direction (when settings are CPOL="0", CPHA="0", SDIR="1", and FWIDTH="00")



2.3.4 Endian of serial interface

The PCC HSSPIn serial interface is 0 to 3. By SENDIAN endianness of the transmit and receive data.

When the HSSPIn_PCC0 to 3.SENDIAN is "0", serial transfer is performed from the upper bytes (refer to Figure 2-6).

When the HSSPIn_PCC0 to 3.SENDIAN is "1", serial transfer is performed from the lower bytes (refer to Figure 2-7).

Figure 2-6 Endian of serial interface (SENDIAN="0")

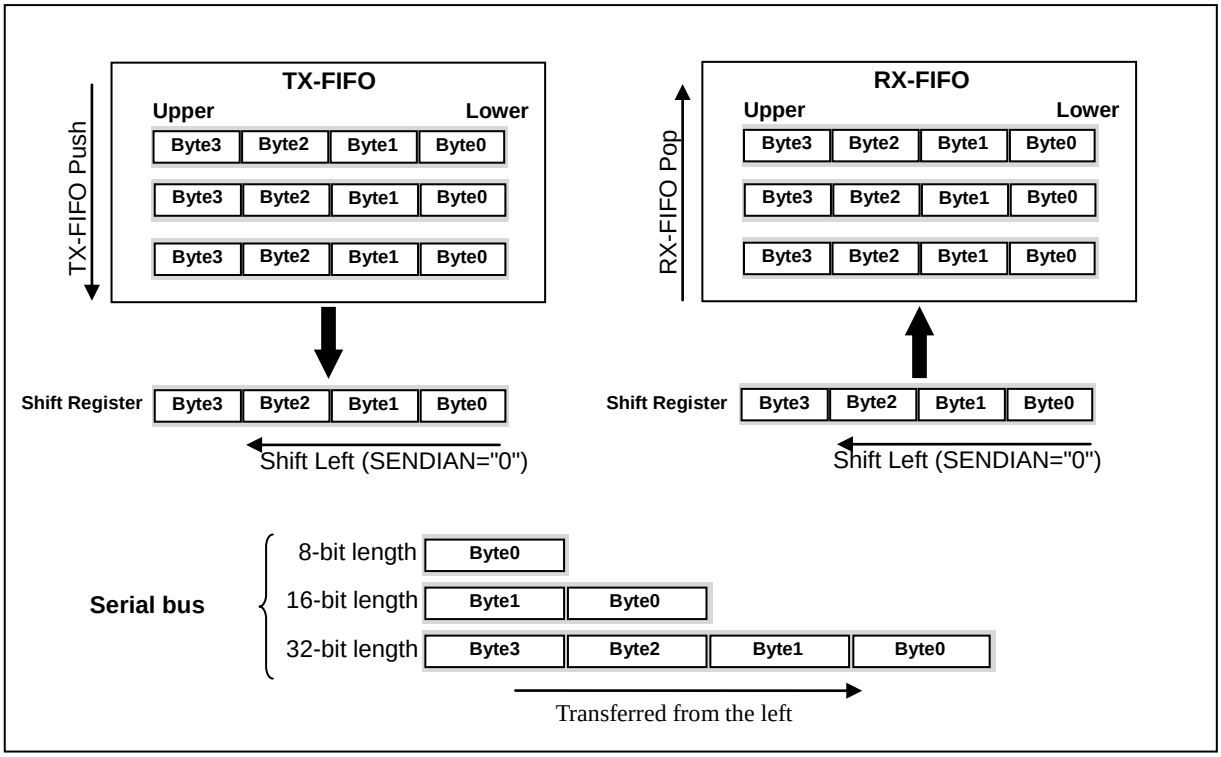
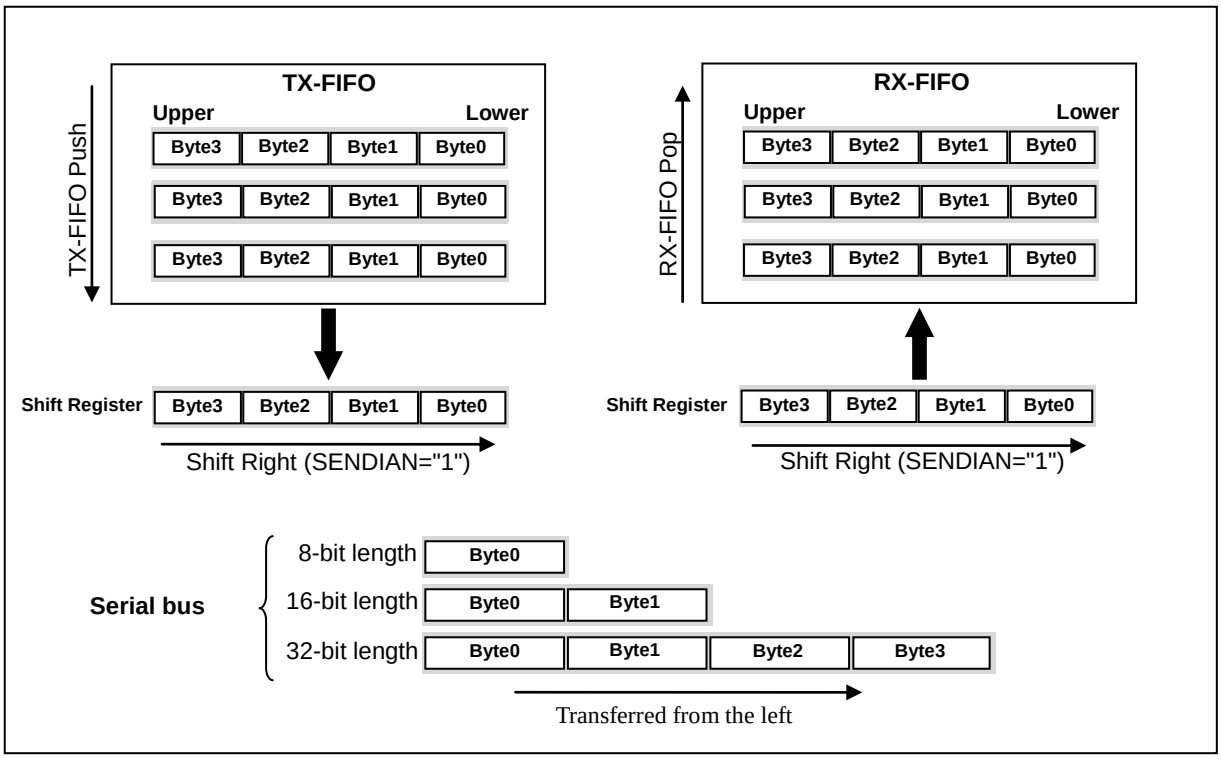


Figure 2-7 Endian of serial interface (SENDIAN="1")



2.3.5 Deselect time

The deselect time is the time between the deassertion of slave selection and the next assertion (refer to Figure 2-8).

The deselect time is set according to the settings of the HSSPIn_PCC0 to 3.RDDSEL[1:0] bits or the HSSPIn_PCC0 to 3.WRDSEL[3:0] bits. The HSSPIn_PCC0 to 3.WRDSEL[3:0] bits are used under the following conditions, and the HSSPIn_PCC0 to 3.RDDSEL[1:0] bits are used in the other cases.

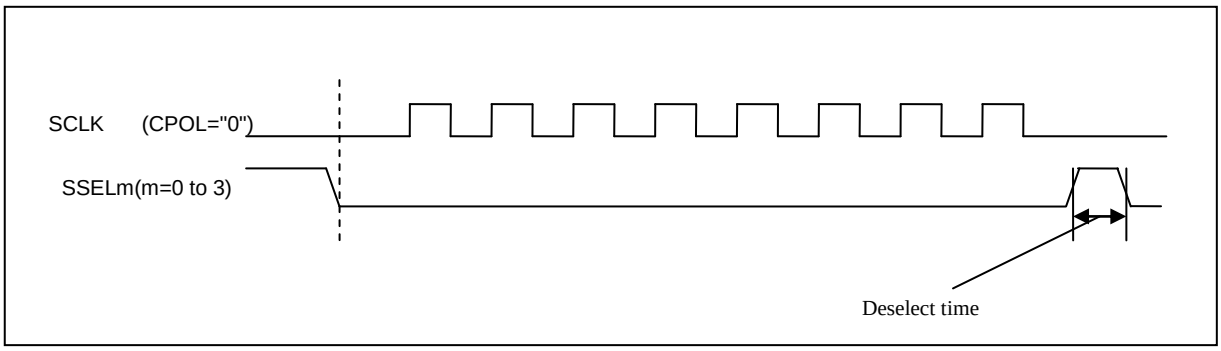
- Usage conditions of the HSSPIn_PCC0 to 3.WRDSEL[3:0] bit
  - When the DEC bit of the register RDCSDC0 to 7 or RDCSDATA[7:0] is rewritten
  - When the DEC bit of the register WRCSDC0 to 7 or WRCSDATA[7:0] is rewritten
  - After using the RDCSDC0 to 7 registers, the WRCSDC0 to 7 registers are used

    (when write access is performed after read access)
  - After using the WRCSDC0 to 7 registers, the RDCSDC0 to 7 registers are used

    (when read access is performed after write access)
  - When using the WRCSDC0 to 7 registers (write access)
  - When the deselect is occurred by the idle timer

The control of the deselect time is enabled only in command sequencer mode.

Figure 2-8 Deselect time

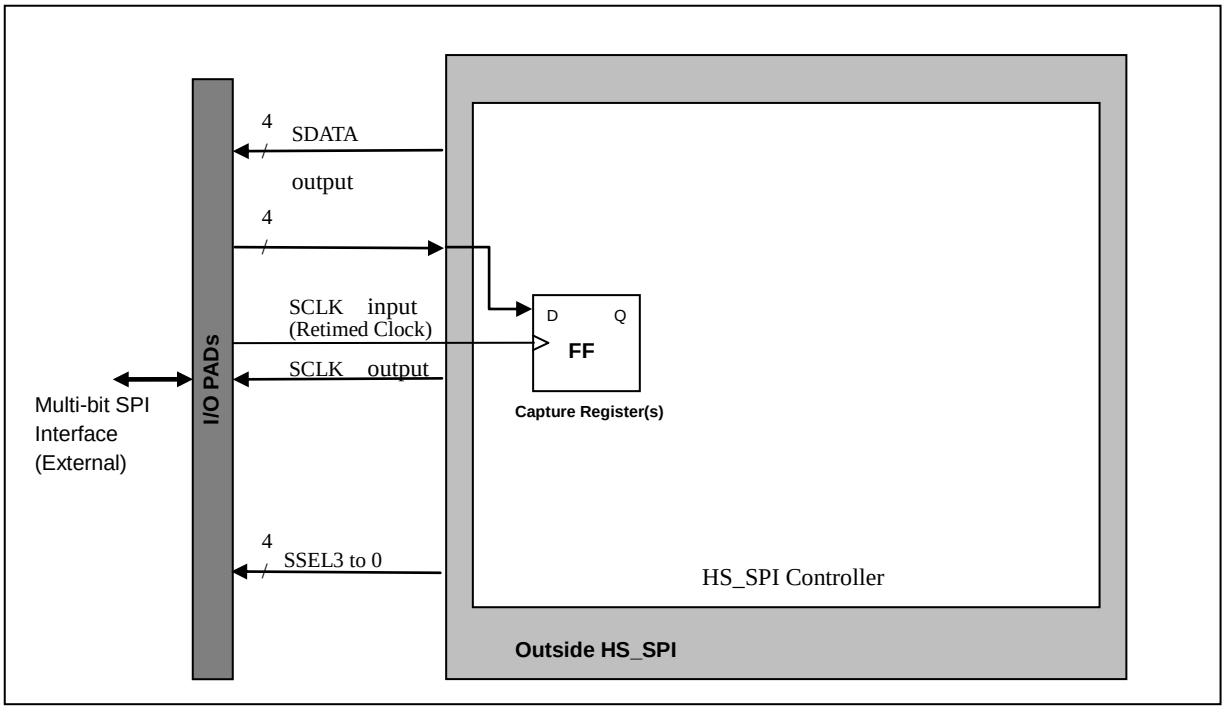


2.3.6 Timing-corrected clock

When operating a serial FLASH memory at high speed clock frequency, the AC characteristics, such as the setup time, are very difficult for the HS_SPI controller Therefore, the timing correction of clock is supported. This timing correction is enabled by setting "1" to the HSSPIn_PCC0 to 3.RTM bits.

Figure 2-9 shows how to correct the clock timings in the HS_SPI controller.

Figure 2-9 Timing-corrected clock in HS_SPI



In clock timing correction mode (HSSPIn_PCC0 to 3.RTM bit = "1"), Flip-flop which captures the external serial data captures the data at the clock (SCLK input) returned to the HS_SPI controller.

Such clock is called "timing-corrected clock". This methods provides the appropriate timing to capture input data which is difficult to be captured in cycle.

2.4 Direct mode

The HS_SPI controller supports direct mode and command sequencer mode.

Direct mode is for accessing serial interface via FIFO. By setting "0" for the HSSPIn_MCTRL.CSEN bit, direct mode is activated.

2.4.1 FIFO

The HS_SPI controller has the transmission FIFO (TX-FIFO) and reception FIFO (RX-FIFO), and each FIFO has 16 stages while each stage consists of 32-bit data width. For each FIFO, writing and reading must be performed in accordance with the FIFO bit width (HSSPIn_FIFOCFG.FWIDTH). An error will be returned if an access is performed with settings different from the FIFO bit width (HSSPIn_FIFOCFG.FWIDTH).

If the FIFO bit width is 24-bit (HSSPIn_FIFOCFG.FWIDTH=2), access with 32-bit.

The FIFO bit width (HSSPIn_FIFOCFG.FWIDTH) is also used as the data length of serial data. For example, 16-bit is specified for the FIFO bit width (HSSPIn_FIFOCFG.FWIDTH), the data for transmission/reception of serial data will be 16-bit length. Although transmission/reception byte count is set to the HSSPIn_DMBCC.BCC bit and transmission/reception is performed for transfer bytes in byte counter mode (HSSPIn_DMCFG.SSDC="1"), transmission/reception is performed in accordance of the data length setting and the last transmission/reception is performed for the remaining data length if the transfer byte count is not the multiple of the data length.

2.4.2 AHB interface clock (QHCLK) and serial clock

In accordance of the relationship between the frequencies of the AHB interface (QHCLK) and serial clock, set the HSSPIn_PCC0 to 3.SAFESYNC bit by following the conditions below.

Table 2-5 AHB slave interface

Bit length *1	Bit width	Conditions when the SAFESYNC bit must be set to "1"
8-bit	Single bit	No condition *2 (The SAFESYNC bit can be set to either "0" or "1".)
	Dual bit	When QHCLK is selected as the serial operation clock source Fsclk = Fqhclk / 2
	Quad bit	SCLK less than 5 divisions of QHCLK
16-bit	Single bit	No condition *2 (The SAFESYNC bit can be set to either "0" or "1".)
	Dual bit	
	Quad bit	When QHCLK is selected as the serial operation clock source Fsclk = Fqhclk / 2
24-bit	Single bit	No condition *2 (The SAFESYNC bit can be set to either "0" or "1".)
	Dual bit	
	Quad bit	
32-bit	Single bit	No condition *2 (The SAFESYNC bit can be set to either "0" or "1".)
	Dual bit	
	Quad bit	

(Fsclk: Serial clock frequency, Fqhclk: AHB interface clock frequency)

*1 : In byte counter mode, the bit length is determined according to the FIFO bit width (HSSPIn_FIFOCFG.FWIDTH) setting and the HSSPIn_DMBCC.BCC bit setting (refer to "2) Byte counter mode" in "2.4.3 Transfer control mode")

*2 : When "1" is set to the SAFESYNC bit, transfer speed will be reduced in order to have data intervals. To increase the transfer speed, it is recommended to set "0" to the SAFESYNC bit.

2.4.3 Transfer control mode

In direct mode, the HS_SPI controller supports two control modes for controlling transmission/reception stop of serial data. Those two control modes are as follows.

1) Software control mode (HSSPIn_DMCFG.SSDC="0")
2) Byte counter mode (HSSPIn_DMCFG.SSDC="1")

This section explains each control mode.

1) Software control mode (HSSPIn_DMCFG.SSDC="0")

In software control mode, when the HS_SPI controller operation is enabled (HSSPIn_MCTRL.MEN="1", HSSPIn_MCTRL.MES="1"), setting "1" to the HSSPIn_DMSTART.START bit starts transmission/reception. However, in TX-and-RX mode or TX-Only mode, transmission/reception will not be started until the transmission data is written.
When "1" is set to the HSSPIn_DMSTOP.STOP bit, transmission/reception will be stopped, slave select will be negated and the interrupt flag; the HSSPIn_TXF.TSSRS bit or the HSSPIn_RXF.RSSRS bit, will be "1" (refer to Table 2-6). However, when "1" is set to the HSSPIn_DMSTOP.STOP bit while reception FIFO is Full, slave select will be asserted state if reception data read is not performed and the interrupt flag; the HSSPIn_TXF.TSSRS bit
or the HSSPIn_RXF.RSSRS bit will not be "1".

2) Byte counter mode (HSSPIn_DMCFG.SSDC="1")

In byte counter mode, when the HS_SPI controller operation is enabled (HSSPIn_MCTRL.MEN="1", HSSPIn_MCTRL.MES="1"), setting "1" to the HSSPIn_DMSTART.START bit starts transmission/reception. However, in TX-and-RX mode or TX-Only mode, transmission/reception will not be started until the transmission data is written.
If transmission/reception is performed for bytes set for the HSSPIn_DMBCC.BCC bit, transmission/reception will be automatically stopped, slave select signal will be negated and the interrupt flag; the HSSPIn_TXF.TSSRS bit or the HSSPIn_RXF.RSSRS bit, will be "1" (refer to Table 2-6). However, when transmission/reception is performed for bytes set for the HSSPIn_DMBCC.BCC bit while reception FIFO is Full, slave select signal will be asserted state if reception data read is not performed and the interrupt flag; the HSSPIn_TXF.TSSRS bit or the HSSPIn_RXF.RSSRS bit will not be "1".
The remaining amount of transmission/reception is shown in the HSSPIn_DMBCS.BCS bit.

The FIFO bit width (HSSPIn_FIFOCFG.FWIDTH) is also used as the data length of serial data. In byte counter mode, if the HSSPIn_DMBCC.BCC bit setting is not multiple of the data length of serial data, the last data transmission/reception is performed for the remaining data length.
For example, when HSSPIn_DMBCC.BCC=9 is set and the data length of serial data is 32-bit, transmission/reception of 8 bytes will be performed with 32-bit length. Transmission/reception of the remaining 1 byte is performed with 8-bit length. Therefore, the setting conditions of the 8-bit data length is required for the HSSPIn_PCC0 to 3.SAFESYNC bits .

2.4.4 Operation mode

The HS_SPI controller supports three operation modes in direct mode. Those three operation modes are as follows.

1) TX-and-RX mode (HSSPIn_DMTRP.TRP[3:2]="00")
2) RX-Only mode (HSSPIn_DMTRP.TRP[3:2]="01")
3) TX-Only mode (HSSPIn_DMTRP.TRP[3:2]="10")

The following explains each operation.

1) TX-and-RX mode (HSSPIn_DMTRP.TRP[3:2]="00")

In TX-and-RX mode, transmission/reception of serial data will be performed at the same time.
This can be used only for single bit.

2) RX-Only mode (HSSPIn_DMTRP.TRP[3:2]="01")

In RX-Only mode, only reception of serial data will be performed.
This can be used all for single bit, dual bit and quad bit.

3) TX-Only mode (HSSPIn_DMTRP[3:2]="10")

In TX-Only mode, only transmission of serial data will be performed.
This can be used all for single bit, dual bit and quad bit.

Conditions for transmission/reception start, stop and hold (when slave select signal is asserted state while serial clock is stopped state) in each operation mode for direct mode are shown below. When transmission/reception is stopped, the HSSPIn_TXF.TSSRS bit or the HSSPIn_RXF.RSSRS bit will be "1".

Table 2-6 Conditions for transmission/reception start, stop and hold in each operation mode for direct mode

<table>
<tr><th>Operation mode</th><th>Transfer control mode</th><th>Conditions for transmission/reception start</th><th>Transmission/reception stop</th><th>Transmission/reception hold</th></tr>
<tr><td rowspan="2">TX-and-RX</td><td>Controlled by software</td><td rowspan="4">Writing transmission data while setting "1" to the HSSPIn_DMSTART.START bit</td><td>All transmission data transmitted while writing "1" to the HSSPIn_DMSTOP.STOP bit*1</td><td rowspan="2">Transmission FIFO (TX-FIFO) is Empty or reception FIFO (RX-FIFO) is Full</td></tr>
<tr><td>Byte counter</td><td>Transmission/reception is performed for setting of the HSSPIn_DMBCC.BCC bit*1</td></tr>
<tr><td rowspan="2">TX-Only</td><td>Controlled by software</td><td>All transmission data transmitted while writing "1" to the HSSPIn_DMSTOP.STOP bit</td><td rowspan="2">Transmission FIFO (TX-FIFO) is Empty</td></tr>
<tr><td>Byte counter</td><td>Transmission is performed for setting of the HSSPIn_DMBCC.BCC bit</td></tr>
<tr><td rowspan="2">RX-Only</td><td>Controlled by software</td><td rowspan="2">Writing "1" to the HSSPIn_DMSTART bit</td><td>Reception data transferred to reception FIFO while writing "1" to the HSSPIn_DMSTOP.STOP bit</td><td rowspan="2">Reception FIFO (RX-FIFO) is Full</td></tr>
<tr><td>Byte counter</td><td>Reception is performed for setting of the HSSPIn_DMBCC.BCC bit*1</td></tr>
</table>

*1: If reception FIFO is Full and data exists in reception shifter, operation does not stop unless reading the reception data from reception FIFO.

2.4.5 Tri-state control of data output

When "1" is set to the HSSPIn_FIFOCFG.TXCTRL bit and transmission data is written to transmission FIFO (TX-FIFO), the HS_SPI controller combines the lowest bit of the written data with the HSSPIn_FIFOCFG.TXCTRL bit during write and performs tri-state control of data output. Table 2-7 shows those combinations and tri-state control. Even if the HSSPIn_FIFOCFG.TXCTRL bit is changed during transmission data write, the combination of data which has been already written at the time and the HSSPIn_FIFOCFG.TXCTRL bit is retained. Therefore, the HSSPIn_FIFOCFG.TXCTRL can be changed.

Table 2-7 Tri-state of data output

TXCTRL	TX-FIFO data bit [0]	Description
0	-	The serial data output will not be Hi-Z state during data output period (will be drive state).
1	0	The serial data output will be Hi-Z state for 1 byte data. The data corresponding to TX-FIFO will not be transmitted.
1	1	Data transmission will be performed as follows. 1) Bit [7:4] of the data corresponding to TX-FIFO is transmitted. The bit transmission order of this data is determined by the setting of the HSSPIn_PCC0 to 3.SDIR bits. 2) Then the SDATA output will be Hi-Z state for 4-bit data.

Note:

- *When "1" is set to the HSSPIn_FIFOCFG.TXCTRL bit, regardless of the FIFO bit width, the written data is handled as 8-bit.*

2.5 Command Sequencer Mode

The HS_SPI controller supports direct mode and command sequencer mode.

In command sequencer mode, when the serial memory accesses the allocated system area, the access will be automatically converted to read/write operation to serial memory by the HS_SPI controller. By setting "1" for the HSSPIn_MCTRL.CSEN bit, command sequencer mode is activated.

Slave devices can be controlled with four slave select, however, the same value must be set all for the HSSPIn_PCC0 to 3 registers (slave devices of the same characteristic can be used).

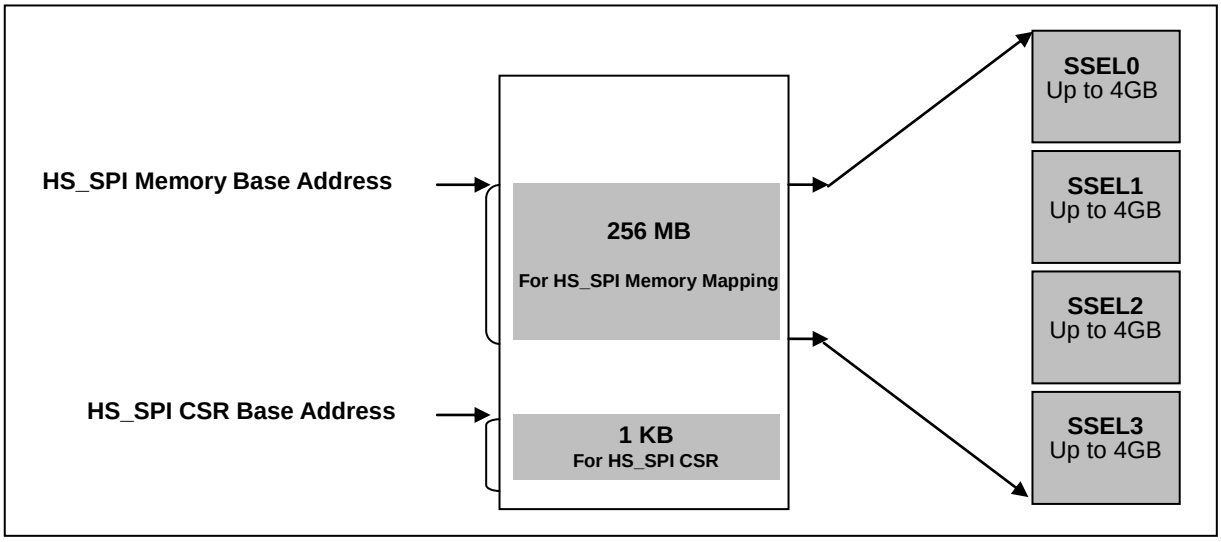
2.5.1 Allocation on the memory

In command sequencer mode, up to four serial memory can be allocated to system space. To these serial memory, four signals of slave select output from the SPI HS controller are connected. The connected memory units (up to 4 units) must have the same characteristics (because the same setting must be applied to all the HSSPIn_PCC0 to 3 registers).

In command sequencer mode, these memory units are allocated to the memory area of 256 Mbytes within the system space. By using the address extension feature, each slave select can have a memory space of up to 4 G bytes theoretically. The extension address combines the upper part of the address extension register (HSSPIn_CSAEXT[18:0]) and the lower part of the AHB address value, and generates 32-bit of slave address.

The address section of 256 M bytes in the system is virtually allocated to memory space of 16 G bytes. Figure 2-10 shows this concept.

Figure 2-10 Memory device allocation for each slave select

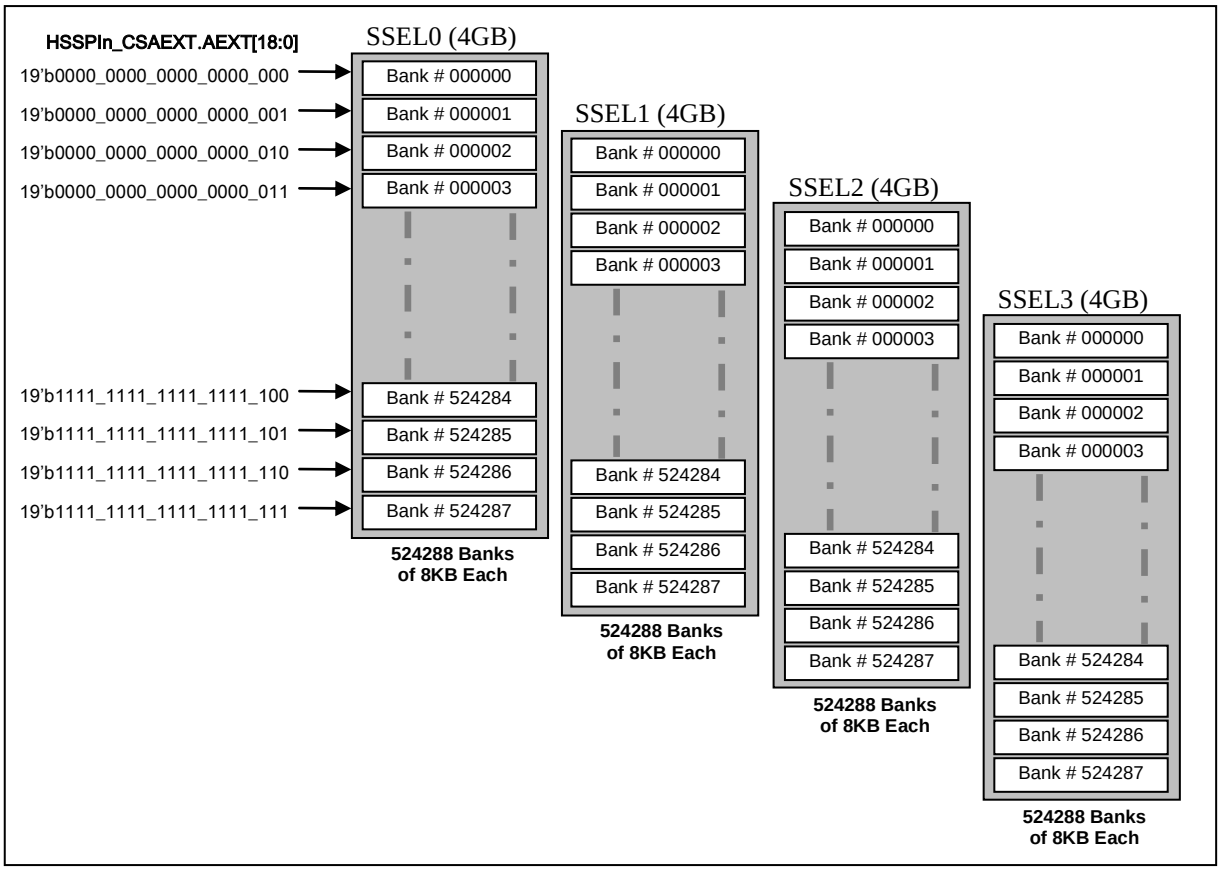


2.5.2 32-bit memory address

Up to 4 G bytes can be allocated to each slave select lines for the section of 256 M bytes in the system space. At that time, the address extension feature is applied. Each serial memory configures memory bank which can be seen on the system space. The size of each bank is set by HSSPIn_CSCFG.MSEL. The range of each memory bank is switched by the HSSPIn_CSAEXT register and accessing to other area in the serial memory can be possible by the switching. In this way, accessing individual memory space up to 4 G bytes will be possible.

Figure 2-11 shows this concept. In the figure, the memory space of 4 G bytes is divided into 524,288 banks, and HSSPIn_CSCFG.MSEL is set to "0000" at that time.

Figure 2-11 Address space allocation of 4 G bytes using bank switching



The lower bit of the AHB address which accesses the HS_SPI controller is used as the address offset value within the selected memory bank. Memory Bank is selected by the address extension register HSSPIn_CSAEXT.AEXT bit.

HSSPIn_CSCFG.MSEL which specifies the size of each bank, the address extension register and the AHB address are combined and 32-bit address space is generated on serial interface. Table 2-8 shows this combination.

Table 2-8 Allocation to address within system space

HSSPIn_CSCFG.MSEL[3:0]	Each slave select memory bank size	AHB address used for slave select	Bit range of the HSSPIn_CSAEXT register used for memory bank select	Bit range of the AHB address used for address specification within memory bank
0000	8K bytes	SSEL0 to SSEL3 is selected depending on AHB address [14:13] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:0]	AHB address [12:0]
0001	16K bytes	SSEL0 to SSEL3 is selected depending on AHB address [15:14] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:1]	AHB address [13:0]
0010	32K bytes	SSEL0 to SSEL3 is selected depending on AHB address [16:15] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:2]	AHB address [14:0]
0011	64K bytes	SSEL0 to SSEL3 is selected depending on AHB address [17:16] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:3]	AHB address [15:0]
0100	128K bytes	SSEL0 to SSEL3 is selected depending on AHB address [18:17] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:4]	AHB address [16:0]
0101	256K bytes	SSEL0 to SSEL3 is selected depending on AHB address [19:18] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:5]	AHB address [17:0]
0110	512K bytes	SSEL0 to SSEL3 is selected depending on AHB address [20:19] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:6]	AHB address [18:0]
0111	1M bytes	SSEL0 to SSEL3 is selected depending on AHB address [21:20] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:7]	AHB address [19:0]
1000	2M bytes	SSEL0 to SSEL3 is selected depending on AHB address [22:21] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:8]	AHB address [20:0]
1001	4M bytes	SSEL0 to SSEL3 is selected depending on AHB address [23:22] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:9]	AHB address [21:0]
1010	8M bytes	SSEL0 to SSEL3 is selected depending on AHB address [24:23] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:10]	AHB address [22:0]
1011	16M bytes	SSEL0 to SSEL3 is selected depending on AHB address [25:24] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:11]	AHB address [23:0]
1100	32M bytes	SSEL0 to SSEL3 is selected depending on AHB address [26:25] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:12]	AHB address [24:0]
1101	64M bytes	SSEL0 to SSEL3 is selected depending on AHB address [27:26] 0:SSEL0, 1:SSEL1, 2:SSEL2, 3:SSEL3	AEXT[18:13]	AHB address [25:0]
1110	128M bytes	SSEL0 or SSEL1 is selected depending on AHB address [27] 0:SSEL0, 1:SSEL1	AEXT[18:14]	AHB address [26:0]
1111	256M bytes	SSEL0	AEXT[18:15]	AHB address [27:0]

The right 2 columns of Table 2-8 indicate the combination of the HSSPIn_CSAEXT.AEXT and the AHB address which is used as the address accessing serial memory. The address output to serial memory is the combination of the HSSPIn_CSAEXT register for the upper address and the AHB address for the lower address. For example, when HSSPIn_CSCFG.MSEL[3:0]="0000", the 32-bit address of the upper address AEXT[18:0] and the lower AHB address [12:0] is used as serial memory address by specifying address in command sequence.

Slave select of serial interface is selected according to 1 to 2 bits of the AHB address or regardless of conditions as shown in Table 2-8. For example, the AHB address [12:0] is output to serial interface when HSSPIn_CSCFG.MSEL[3:0]="0000", slave select 0 is selected when the AHB address [14:13]=0, slave select 1 is selected when the AHB address [14:13]=1,
slave select 2 is selected when the AHB address [14:13]=2, and slave select 3 is selected when the AHB address [14:13]=3.

2.5.3 AHB interface clock (QHCLK) and serial clock

In accordance of the relationship between the frequencies of the AHB interface (QHCLK) and serial clock, set the HSSPIn_PCC0 to 3.SAFESYNC bit by following the conditions below.

Table 2-9 Setting conditions of the SAFESYNC bit in command sequencer mode

Bit length	Bit width	Conditions when the SAFESYNC bit must be set to "1"	
		When the AHB interface clock (QHCLK) and serial operation clock source are asynchronous (HSSPIn_MCTRL.SYNCON="0")	When the AHB interface clock (QHCLK) and serial operation clock source are synchronous (HSSPIn_MCTRL.SYNCON="1")
8-bit	Single bit	No condition *1 (The SAFESYNC bit can be set to either "0" or "1".)	The SAFESYNC bit is "0"
	Dual bit	SCLK less than 5 divisions of QHCLK	
	Quad bit	SCLK less than 5 divisions of QHCLK or HSSPIn_PCC0 to 3.CDRS=0x0	
16-bit	Single bit	No condition *1 (The SAFESYNC bit can be set to either "0" or "1".)	
	Dual bit		
	Quad bit	SCLK less than 5 divisions of QHCLK	
32-bit	Single bit	No condition *1 (The SAFESYNC bit can be set to either "0" or "1".)	
	Dual bit		
	Quad bit		

(Fsclk: Serial clock frequency, Fqhclk: AHB interface clock frequency)

*1 : When "1" is set to the SAFESYNC bit, transfer speed will be reduced in order to have data intervals. To increase the transfer speed, it is recommended to set "0" to the SAFESYNC bit.

2.5.4 Command sequence

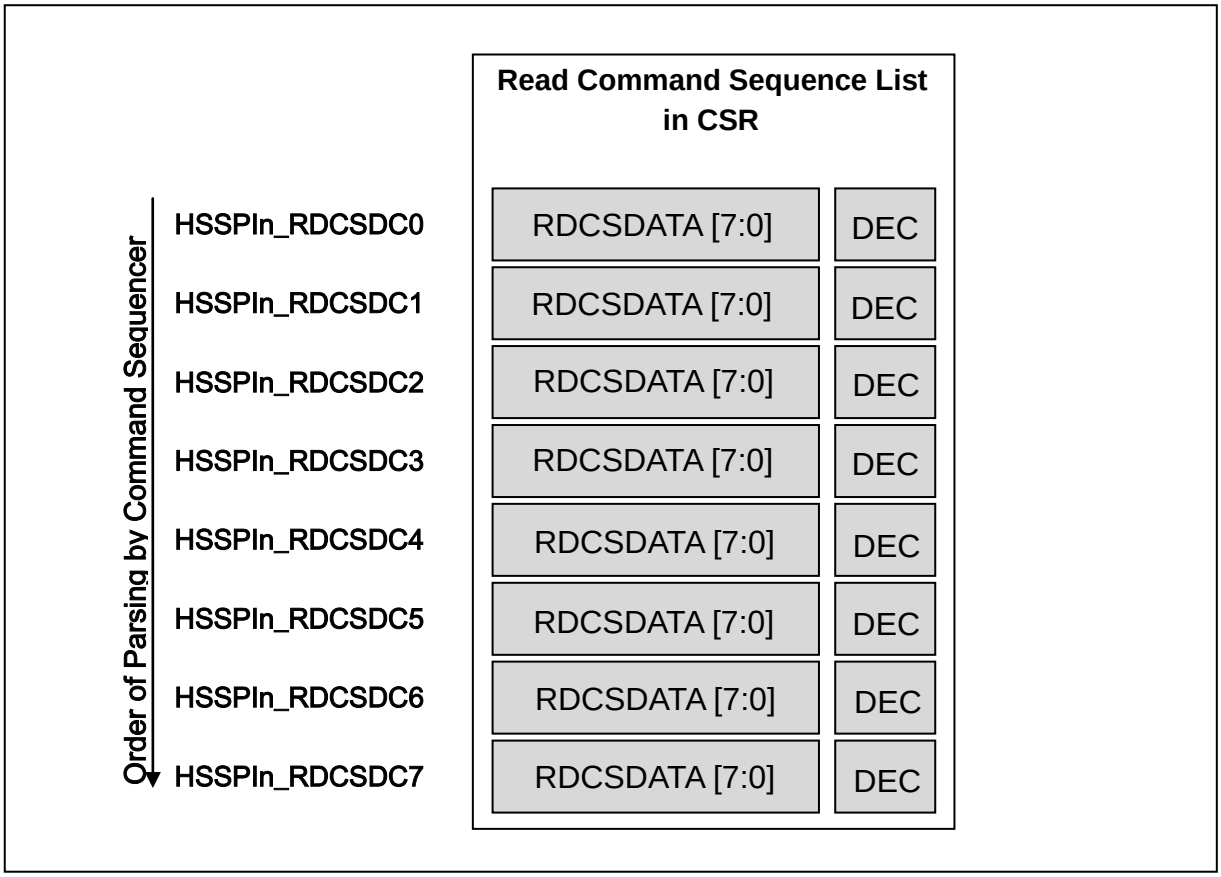
The commands of serial memory system are different depending on companies; however, the HS_SPI controller provides register for setting command sequence to be used for general-purpose. Two kinds of registers are provided for this purpose; HSSPIn_RDCSDC0 to 7 for read commands, and HSSPIn_WRCSDC0 to 7 for write commands.

Serial memory command phase flow (the instruction phase, the address phase, and the data phase) is generated in command sequencer mode of the HS_SPI controller. When read access to system space is performed while each phase is set to HSSPIn_RDCSDC0 to HSSPIn_RDCSDC7 and HSSPIn_WRCSDC0 to HSSPIn_WRCSDC7, the read access to serial memory is performed according to the setting in order from HSSPIn_RDCSDC0. In addition, write access is also possible if "1" is set to the HSSPIn_CSCFG.SRAM bit, and when write access to system space at this time, the write access to serial memory is performed according to the setting in order from HSSPIn_WRCSDC0.

2.5.4.1 Read command sequence

When the order of the command phase is set to 8 registers (HSSPIn_RDCSDC0 to HSSPIn_RDCSDC7) and read access to system space is performed, the read command sequence is performed according to the entry in order from the HSSPIn_RDCSDC0 register. This is shown in Figure 2-12.

Figure 2-12 Sequence list of read commands



The DEC bit of each register indicates whether or not to decode the data type. Table 2-10 shows the relationship between the DEC bit and the bit field (RDCSDATA[2:0]) of data type. When the DEC bit is "0", RDCSDATA[7:0] will be transmitted as is.

Table 2-10 SPI interface

DEC	RDCSDATA[2:0]	Description
0	-	RDCSDATA[7:0] is sent as is.
1	000	Address [7:0] is sent to access serial memory.
1	001	Address [15:8] is sent to access serial memory.
1	010	Address [23:16] is sent to access serial memory.
1	011	Address [31:24] is sent to access serial memory.
1	100	SDATA will be Hi-Z state for 1 byte data.
1	101	4 bits in 1 byte data are transmitted and SDATA will be Hi-Z state for the remaining 4 bits. The operations are as follows. 1) RDCSDATA[7:4] is sent as is. The bit transmission order of this data is determined by the setting of HSSPIn_PCC0 to 3.SDIR. 2) Then SDATA will be Hi-Z state for 4-bit data.
1	111	This indicates the end of list.

In command sequencer mode, one of the following conditions are met, data read from serial memory is performed.

- The end of the list (RDCSDATA[2:0]=7, DEC="1") is detected.
- Reading the HSSPIn_RDCSDC7 register is completed.

Data is read as serial data from serial memory, and the HS_SPI controller receives the data and outputs as serial memory data to AHB.

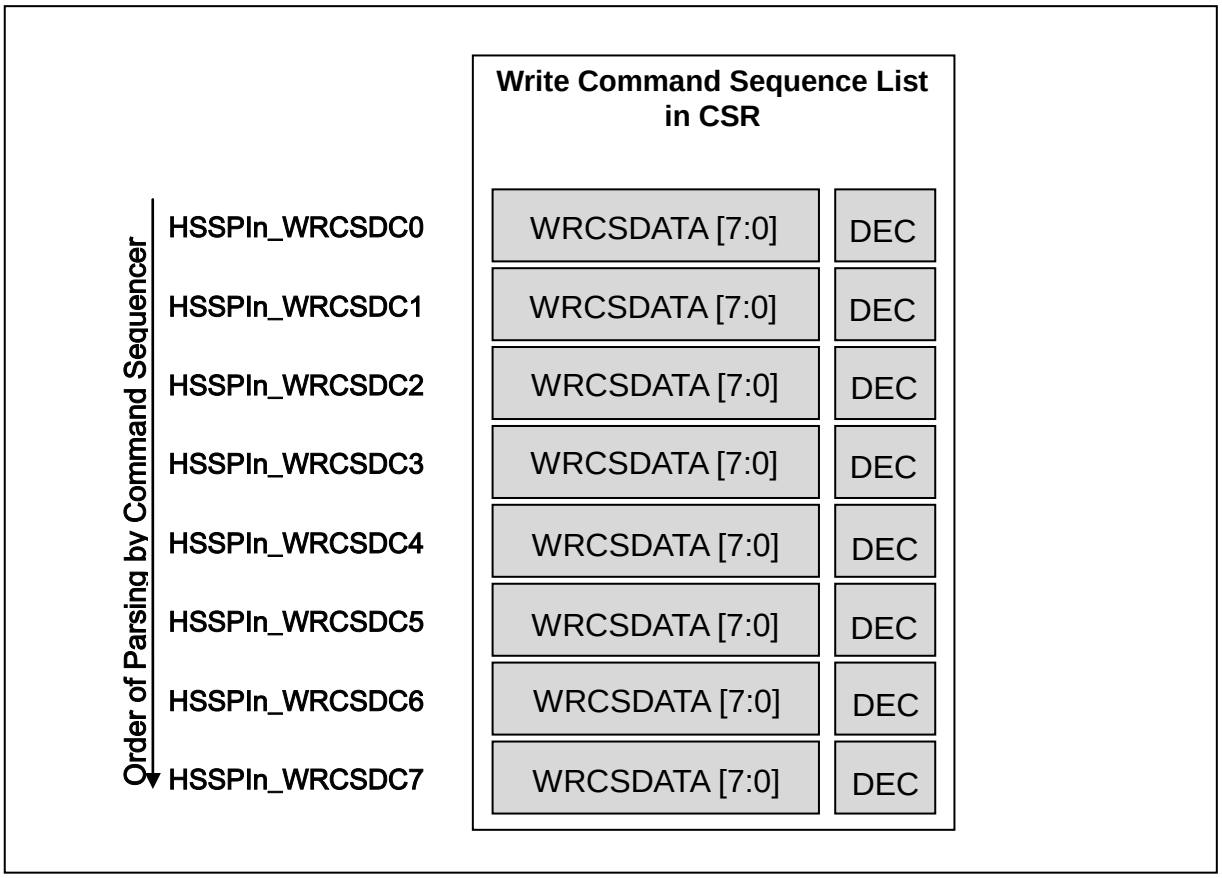
Notes:

- *When reading the HSSPIn_RDCSDC7 register is completed and data reading is started, the TRP[1:0] bit and the CONT bit of the HSSPIn_RDCSDC7 register is used.*
- *Setting the "end of list" to the HSSPIn_RDCSDC0 register is prohibit.*

2.5.4.2 Write command sequence

When the order of the command phase is set to 8 registers (HSSPIn_WRCSDC0 to HSSPIn_WRCSDC7) and write access to system space is performed, the write command sequence is performed according to the entry in order from the HSSPIn_WRCSDC0 register. However, memory write operation is performed only when the HSSPIn_CSCFG.SRAM bit is set to "1". This is shown in Figure 2-13.

Figure 2-13 Sequence list of write commands



The DEC bit of each register indicates whether or not to decode the data type. Table 2-11 shows the relationship between the DEC bit and the bit field (WRCSDATA[2:0]) of data type. When the DEC bit is "0", WRCSDATA[7:0] will be transmitted as is.

Table 2-11 Write command sequence list

DEC	WRCSDATA[2:0]	Description
0	-	WRCSDATA[7:0] is sent as is.
1	000	Address [7:0] is sent to access serial memory.
1	001	Address [15:8] is sent to access serial memory.
1	010	Address [23:16] is sent to access serial memory.
1	011	Address [31:24] is sent to access serial memory.
1	100	SDATA will be Hi-Z state for 1 byte data.
1	101	4 bits in 1 byte data are transmitted and SDATA will be Hi-Z state for the remaining 4 bits. The operations are as follows. 1) WRCSDATA[7:4] is sent as is. The bit transmission order of this data is determined by the setting of HSSPIn_PCC0 to 3.SDIR. 2) Then SDATA will be Hi-Z state for 4-bit data.
1	111	This indicates the end of list.

In command sequencer mode, one of the following conditions are met, data write to serial memory is performed.

- The end of the list (WRCSDATA[2:0]=7, DEC="1") is detected.
- Reading the HSSPIn_WRCSDC7 register is completed.

Data is written and output as serial data to serial bus.

Note:

- *When reading the HSSPIn_WRCSDC7 register is completed and data writing is started, the TRP[1:0] bit and the CONT bit of the HSSPIn_WRCSDC7 register is used.*

2.5.4.3 Continuous instructions

When "1" is set to the CONT bit of the HSSPIn_RDCSDC0 register, the first read operation processes all information set by HSSPIn_RDCSDC0; however, the following operations will omit the processing for the HSSPIn_RDCSDC0 to 7 registers whose CONT bit is set to "1".

In order to switch instructions from continuous instruction to instructions other than continuous instructions, perform the processing below.

<For reading>

1. Rewrite the toggle code within the HSSPIn_RDCSDC0 to 7 registers to release and perform reading.
2. Then set "0" for all CONT bits of the HSSPIn_RDCSDC0 register.
3. Then perform reading.

By this processing, returning from continuous instructions will be possible.

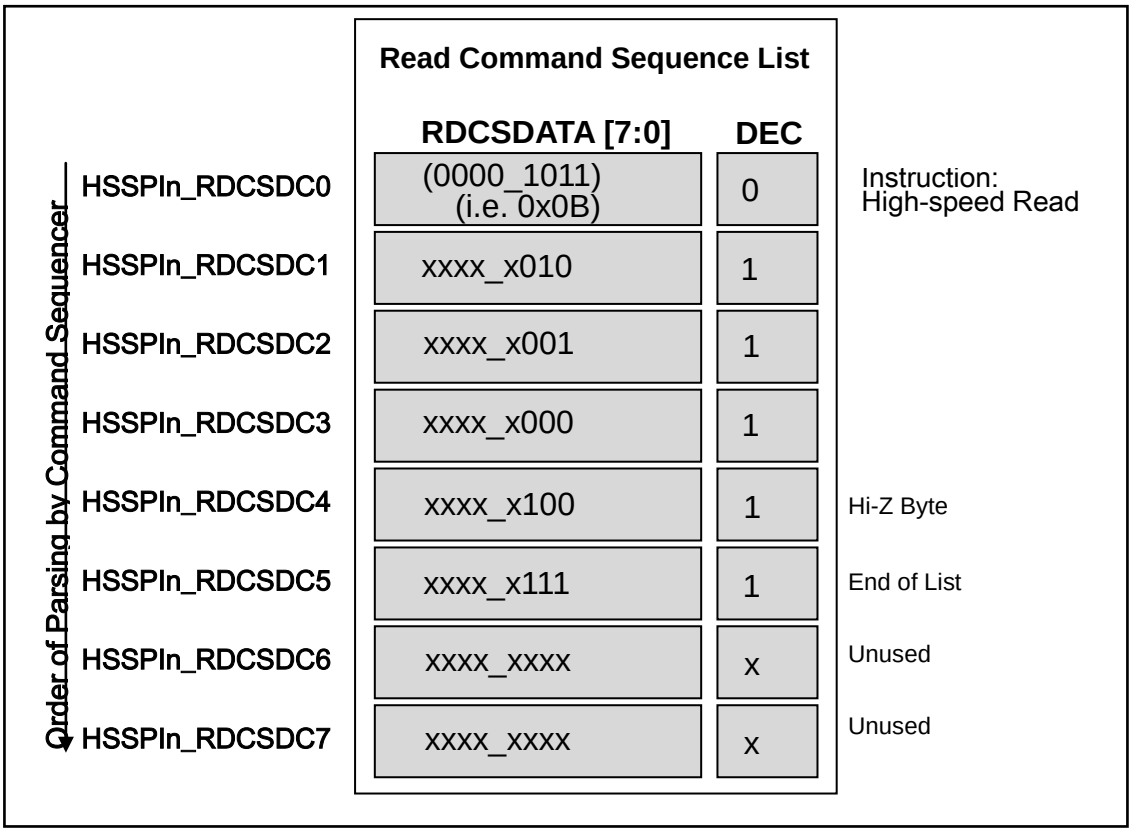
Notes:

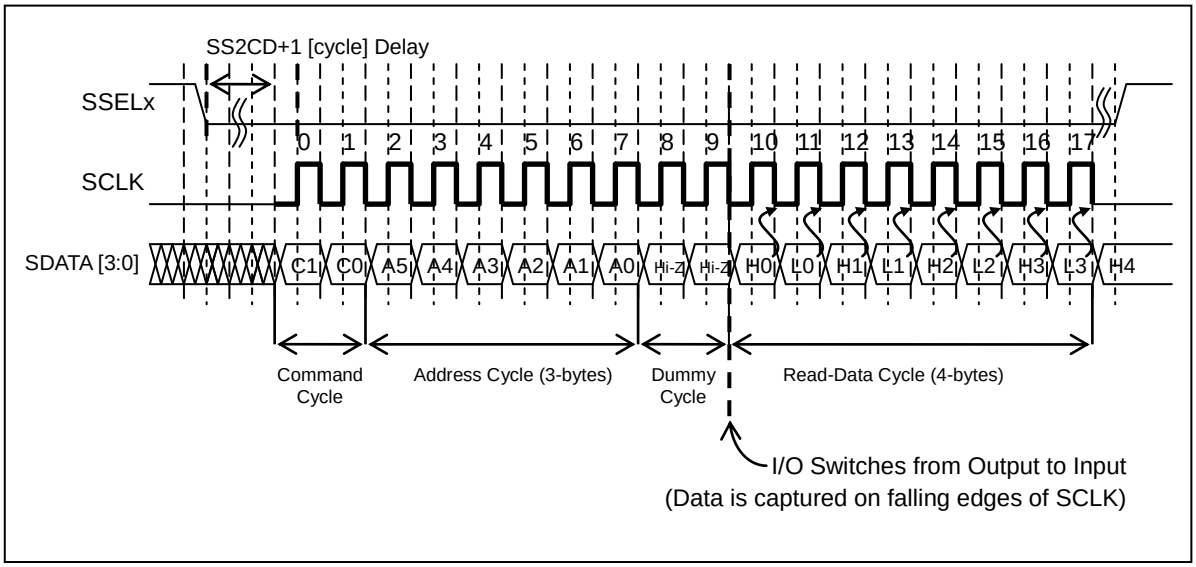
- *Among the CONT bits of the HSSPIn_RDCSDC0 to 7 registers, setting "1" to the CONT bits of the HSSPIn_RDCSDC1 to 7 registers is prohibited.*
- *Set "0" to the CONT bits of the HSSPIn_WRCSDC0 to 7 registers..*

2.5.4.4 Operation example in command sequencer mode

Figure 2-14 shows an operation example of read command sequence and the timing (clocking mode is mode 0) in command sequencer mode. For timing, when read command sequence is set to the HSSPIn_RDCSDC0 to 5 registers, command sequencer execute the command starting from the HSSPIn_RDCSDC0.

Figure 2-14 Operation example of read command sequence (mode 0)





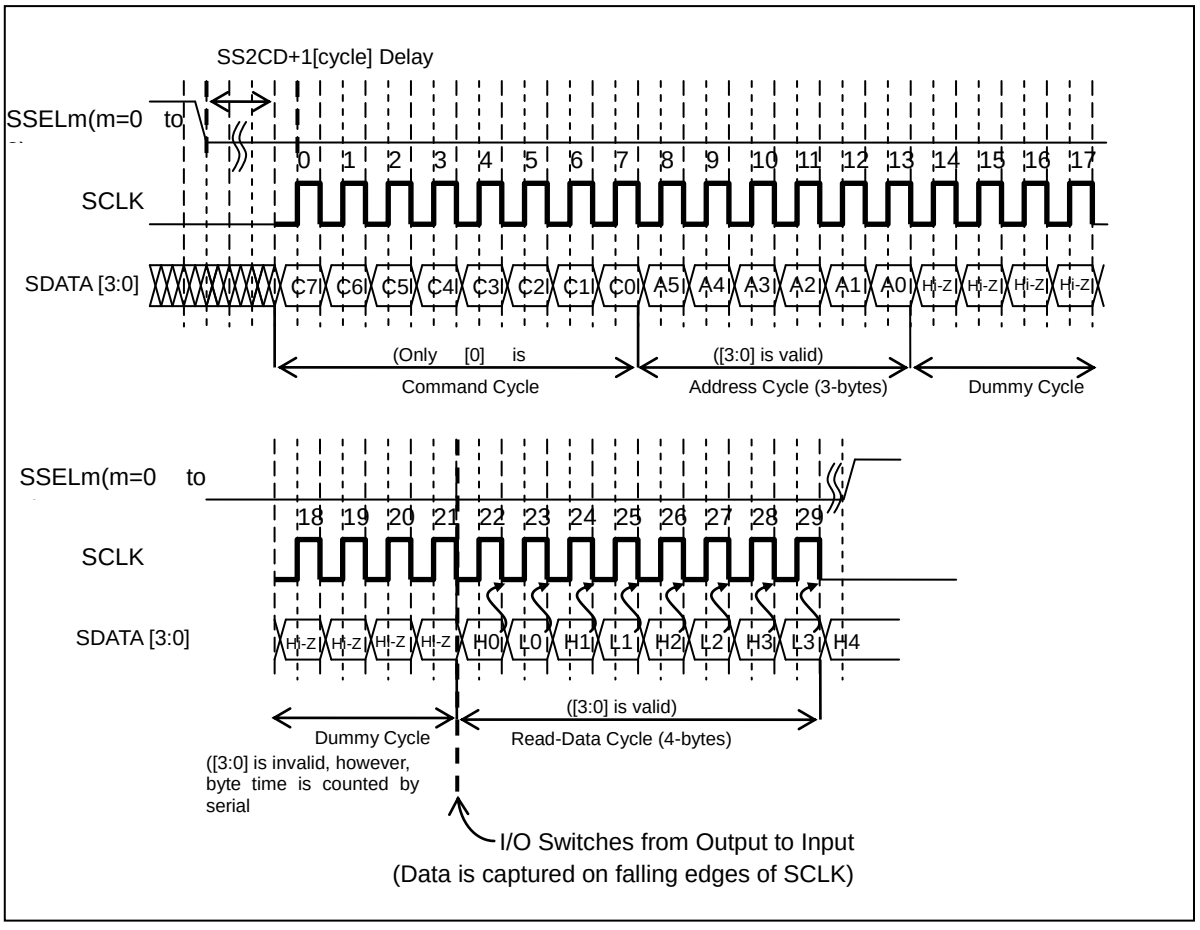
In addition, by the control bit TRP in the list, the bit width of the serial interface for each transmission byte can be controlled to single/dual/quad.

Add the TRP[1:0] bit value setting to the register setting example above.

- HSSPIn_RDCSDC0.TRP=3 (single) : Instruction High-speed Read
- HSSPIn_RDCSDC1.TRP=2 (quad) : Memory Address[23:16]
- HSSPIn_RDCSDC2.TRP=2 (quad) : Memory Address[15:8]
- HSSPIn_RDCSDC3.TRP=2 (quad) : Memory Address[7:0]
- HSSPIn_RDCSDC4.TRP=3 (single) : Hi-Z Byte (Hi-Z of 1byte data)
- HSSPIn_RDCSDC5.TRP=0 (in accordance to MBM[1:0]) : End of List (bit width control is invalid)
- HSSPIn_RDCSDC6.TRP=0 (in accordance to MBM[1:0]) : Not used (bit width control is invalid)
- HSSPIn_RDCSDC7.TRP=0 (in accordance to MBM[1:0]) : Not used (bit width control is invalid)

Operations in this case are shown below.

Figure 2-15 Serial bit width specification by list



2.5.4.5 Timeout of AHB idle state

After accessing to serial memory in command sequencer mode, the HS_SPI controller continues asserting slave select. If the following AHB transaction is detected during the period set by the HSSPIn_CSITIME.ITIME bit, the HS_SPI controller performs only data read for serial memory and does not issue new command or address. By this, the access time can be decreased.

- Address is consecutive from the previous AHB address space.
- Command type of write/read is same as that of the previous one.

If one of the following AHB transaction is detected, even before the idle timer reaches the setting time, the HS_SPI controller negates slave select (and completes the current transfer at the same time), then issues new transaction.

- The following AHB access in idle period is performed to the non-consecutive memory space.
- Command type of write/read is different from that of the previous one.
- The mode is changed to direct mode (write "0" to HSSPIn_MCTRL.CSEN).

If the AHB access to consecutive memory space is not generated in idle period, the idle timer reaches the setting time, and the HS_SPI controller negates slave select and completes the data transfer.

The unit of time set to the HSSPIn_CSITIME.ITIME bit is cycle count of the AHB interface clock (QHCLK).

3. HS_SPI controller register

The HS_SPI controller is equipped with various registers which set those operations.

This chapter explains the HS_SPI controller registers.

3.1 Registers

This section provides a list of registers.

Table 3-1 HS_SPI controller register list

Abbreviated register name	Register name	Reference
HSSPIn_MCTRL	HS-SPI control register	3.2
HSSPIn_PCC0	HS-SPI peripheral communication set register 0	3.3
HSSPIn_PCC1	HS-SPI peripheral communication set register 1	
HSSPIn_PCC2	HS-SPI peripheral communication set register 2	
HSSPIn_PCC3	HS-SPI peripheral communication set register 3	
HSSPIn_TXF	HS-SPI transmission interrupt factor register	3.4
HSSPIn_TXE	HS-SPI transmit interrupt enable register	3.5
HSSPIn_TXC	HS-SPI transmission interrupt clear register	3.6
HSSPIn_RXF	HS-SPI reception interrupt factor register	3.7
HSSPIn_RXE	HS-SPI reception interrupt enable register	3.8
HSSPIn_RXC	HS-SPI reception interrupt clear register	3.9
HSSPIn_FAULTF	HS-SPI violation interrupt factor register	3.10
HSSPIn_FAULTC	HS-SPI violation interrupt clear register	3.11
HSSPIn_DMCFG	HS-SPI direct mode set register	3.12
HSSPIn_DMDMAEN	HS-SPI direct mode DMA enable register	3.13
HSSPIn_DMSTART	HS-SPI direct mode transfer start control register	3.14
HSSPIn_DMSTOP	HS-SPI direct mode transfer stop control register	3.15
HSSPIn_DMPSEL	HS-SPI direct mode slave select register	3.16
HSSPIn_DMTRP	HS-SPI direct mode transfer protocol set register	3.17
HSSPIn_DMBCC	HS-SPI direct mode transfer byte count set register	3.18
HSSPIn_DMBCS	HS-SPI direct mode transfer count remaining register	3.19
HSSPIn_DMSTATUS	HS-SPI direct mode status register	3.20
HSSPIn_TXFIFO0	HS-SPI TX-FIFO register 0	3.21
HSSPIn_TXFIFO1	HS-SPI TX-FIFO register 1	
HSSPIn_TXFIFO2	HS-SPI TX-FIFO register 2	
HSSPIn_TXFIFO3	HS-SPI TX-FIFO register 3	
HSSPIn_TXFIFO4	HS-SPI TX-FIFO register 4	
HSSPIn_TXFIFO5	HS-SPI TX-FIFO register 5	
HSSPIn_TXFIFO6	HS-SPI TX-FIFO register 6	
HSSPIn_TXFIFO7	HS-SPI TX-FIFO register 7	
HSSPIn_TXFIFO8	HS-SPI TX-FIFO register 8	
HSSPIn_TXFIFO9	HS-SPI TX-FIFO register 9	
HSSPIn_TXFIFO10	HS-SPI TX-FIFO register 10	
HSSPIn_TXFIFO11	HS-SPI TX-FIFO register 11	
HSSPIn_TXFIFO12	HS-SPI TX-FIFO register 12	
HSSPIn_TXFIFO13	HS-SPI TX-FIFO register 13	
HSSPIn_TXFIFO14	HS-SPI TX-FIFO register 14	
HSSPIn_TXFIFO15	HS-SPI TX-FIFO register 15	

Abbreviated register name	Register name	Reference
HSSPIn_RXFIFO0	HS-SPI RX-FIFO register 0	3.22
HSSPIn_RXFIFO1	HS-SPI RX-FIFO register 1	
HSSPIn_RXFIFO2	HS-SPI RX-FIFO register 2	
HSSPIn_RXFIFO3	HS-SPI RX-FIFO register 3	
HSSPIn_RXFIFO4	HS-SPI RX-FIFO register 4	
HSSPIn_RXFIFO5	HS-SPI RX-FIFO register 5	
HSSPIn_RXFIFO6	HS-SPI RX-FIFO register 6	
HSSPIn_RXFIFO7	HS-SPI RX-FIFO register 7	3.22
HSSPIn_RXFIFO8	HS-SPI RX-FIFO register 8	
HSSPIn_RXFIFO9	HS-SPI RX-FIFO register 9	
HSSPIn_RXFIFO10	HS-SPI RX-FIFO register 10	
HSSPIn_RXFIFO11	HS-SPI RX-FIFO register 11	
HSSPIn_RXFIFO12	HS-SPI RX-FIFO register 12	
HSSPIn_RXFIFO13	HS-SPI RX-FIFO register 13	
HSSPIn_RXFIFO14	HS-SPI RX-FIFO register 14	
HSSPIn_RXFIFO15	HS-SPI RX-FIFO register 15	
HSSPIn_FIFOCFG	HS-SPI FIFO set register	3.23
HSSPIn_CSCFG	HS-SPI command sequencer set register	3.24
HSSPIn_CSITIME	HS-SPI command sequencer idle timer set register	3.25
HSSPIn_CSAEXT	HS-SPI command sequencer address extend register	3.26
HSSPIn_RDCSDC0	HS-SPI read command sequence data/control register 0	3.27
HSSPIn_RDCSDC1	HS-SPI read command sequence data/control register 1	
HSSPIn_RDCSDC2	HS-SPI read command sequence data/control register 2	
HSSPIn_RDCSDC3	HS-SPI read command sequence data/control register 3	
HSSPIn_RDCSDC4	HS-SPI read command sequence data/control register 4	
HSSPIn_RDCSDC5	HS-SPI read command sequence data/control register 5	
HSSPIn_RDCSDC6	HS-SPI read command sequence data/control register 6	
HSSPIn_RDCSDC7	HS-SPI read command sequence data/control register 7	
HSSPIn_WDCSDC0	HS-SPI write command sequence data/control register 0	3.28
HSSPIn_WDCSDC1	HS-SPI write command sequence data/control register 1	
HSSPIn_WDCSDC2	HS-SPI write command sequence data/control register 2	
HSSPIn_WDCSDC3	HS-SPI write command sequence data/control register 3	
HSSPIn_WDCSDC4	HS-SPI write command sequence data/control register 4	
HSSPIn_WDCSDC5	HS-SPI write command sequence data/control register 5	
HSSPIn_WDCSDC6	HS-SPI write command sequence data/control register 6	
HSSPIn_WDCSDC7	HS-SPI write command sequence data/control register 7	
HSSPIn_MID	HS-SPI module identification register	3.29

3.1.1 Total control register list

Abbreviated register name	Register name	Reference
HSSPIn_MCTRL	HS-SPI control register	3.2

3.1.2 Serial interface control register list

Abbreviated register name	Register name	Reference
HSSPIn_PCC0	HS-SPI peripheral communication set register 0	3.3
HSSPIn_PCC1	HS-SPI peripheral communication set register 1	
HSSPIn_PCC2	HS-SPI peripheral communication set register 2	
HSSPIn_PCC3	HS-SPI peripheral communication set register 3	

3.1.3 Fault register list

Abbreviated register name	Register name	Reference
HSSPIn_FAULTF	HS-SPI violation interrupt factor register	3.10
HSSPIn_FAULTC	HS-SPI violation interrupt clear register	3.11

3.1.4 DMA control register list

Abbreviated register name	Register name	Reference
HSSPIn_DMDMAEN	HS-SPI direct mode DMA enable register	3.13

3.1.5 FIFO control register list (common to TX/RX)

Abbreviated register name	Register name	Reference
HSSPIn_FIFOCFG	HS-SPI FIFO set register	3.23

3.1.6 TX-FIFO control register list

Abbreviated register name	Register name	Reference
HSSPIn_TXF	HS-SPI transmission interrupt factor register	3.4
HSSPIn_TXE	HS-SPI transmit interrupt enable register	3.5
HSSPIn_TXC	HS-SPI transmission interrupt clear register	3.6
HSSPIn_TXFIFO0	HS-SPI TX-FIFO register 0	3.21
HSSPIn_TXFIFO1	HS-SPI TX-FIFO register 1	
HSSPIn_TXFIFO2	HS-SPI TX-FIFO register 2	
HSSPIn_TXFIFO3	HS-SPI TX-FIFO register 3	
HSSPIn_TXFIFO4	HS-SPI TX-FIFO register 4	
HSSPIn_TXFIFO5	HS-SPI TX-FIFO register 5	
HSSPIn_TXFIFO6	HS-SPI TX-FIFO register 6	
HSSPIn_TXFIFO7	HS-SPI TX-FIFO register 7	
HSSPIn_TXFIFO8	HS-SPI TX-FIFO register 8	
HSSPIn_TXFIFO9	HS-SPI TX-FIFO register 9	
HSSPIn_TXFIFO10	HS-SPI TX-FIFO register 10	
HSSPIn_TXFIFO11	HS-SPI TX-FIFO register 11	
HSSPIn_TXFIFO12	HS-SPI TX-FIFO register 12	
HSSPIn_TXFIFO13	HS-SPI TX-FIFO register 13	
HSSPIn_TXFIFO14	HS-SPI TX-FIFO register 14	
HSSPIn_TXFIFO15	HS-SPI TX-FIFO register 15	

3.1.7 RX-FIFO control register list

Abbreviated register name	Register name	Reference
HSSPIn_RXF	HS-SPI reception interrupt factor register	3.7
HSSPIn_RXE	HS-SPI reception interrupt enable register	3.8
HSSPIn_RXC	HS-SPI reception interrupt clear register	3.9
HSSPIn_RXFIFO0	HS-SPI RX-FIFO register 0	3.22
HSSPIn_RXFIFO1	HS-SPI RX-FIFO register 1	
HSSPIn_RXFIFO2	HS-SPI RX-FIFO register 2	
HSSPIn_RXFIFO3	HS-SPI RX-FIFO register 3	
HSSPIn_RXFIFO4	HS-SPI RX-FIFO register 4	
HSSPIn_RXFIFO5	HS-SPI RX-FIFO register 5	
HSSPIn_RXFIFO6	HS-SPI RX-FIFO register 6	
HSSPIn_RXFIFO7	HS-SPI RX-FIFO register 7	
HSSPIn_RXFIFO8	HS-SPI RX-FIFO register 8	
HSSPIn_RXFIFO9	HS-SPI RX-FIFO register 9	
HSSPIn_RXFIFO10	HS-SPI RX-FIFO register 10	
HSSPIn_RXFIFO11	HS-SPI RX-FIFO register 11	
HSSPIn_RXFIFO12	HS-SPI RX-FIFO register 12	
HSSPIn_RXFIFO13	HS-SPI RX-FIFO register 13	
HSSPIn_RXFIFO14	HS-SPI RX-FIFO register 14	
HSSPIn_RXFIFO15	HS-SPI RX-FIFO register 15	

3.1.8 Direct mode control register list

Direct mode control register list

Abbreviated register name	Register name	Reference
HSSPIn_DMCFG	HS-SPI direct mode set register	3.12
HSSPIn_DMSTART	HS-SPI direct mode transfer start control register	3.14
HSSPIn_DMSTOP	HS-SPI direct mode transfer stop control register	3.15
HSSPIn_DMPSEL	HS-SPI direct mode slave select register	3.16
HSSPIn_DMTRP	HS-SPI direct mode transfer protocol set register	3.17
HSSPIn_DMBCC	HS-SPI direct mode transfer byte count set register	3.18
HSSPIn_DMBCS	HS-SPI direct mode transfer count remaining register	3.19
HSSPIn_DMSTATUS	HS-SPI direct mode status register	3.20

3.1.9 Command sequencer control register list (common to write/read)

Abbreviated register name	Register name	Reference
HSSPIn_CSCFG	HS-SPI command sequencer set register	3.24
HSSPIn_CSITIME	HS-SPI command sequencer idle timer set register	3.25
HSSPIn_CSAEXT	HS-SPI command sequencer address extend register	3.26

3.1.10 Command sequence set register list (write command)

Abbreviated register name	Register name	Reference
HSSPIn_WDCSDC0	HS-SPI write command sequence data/control register 0	3.28
HSSPIn_WDCSDC1	HS-SPI write command sequence data/control register 1	
HSSPIn_WDCSDC2	HS-SPI write command sequence data/control register 2	
HSSPIn_WDCSDC3	HS-SPI write command sequence data/control register 3	
HSSPIn_WDCSDC4	HS-SPI write command sequence data/control register 4	
HSSPIn_WDCSDC5	HS-SPI write command sequence data/control register 5	
HSSPIn_WDCSDC6	HS-SPI write command sequence data/control register 6	
HSSPIn_WDCSDC7	HS-SPI write command sequence data/control register 7	

3.1.11 Command sequence set register list (read command)

Abbreviated register name	Register name	Reference
HSSPIn_RDCSDC0	HS-SPI read command sequence data/control register 0	3.27
HSSPIn_RDCSDC1	HS-SPI read command sequence data/control register 1	
HSSPIn_RDCSDC2	HS-SPI read command sequence data/control register 2	
HSSPIn_RDCSDC3	HS-SPI read command sequence data/control register 3	
HSSPIn_RDCSDC4	HS-SPI read command sequence data/control register 4	
HSSPIn_RDCSDC5	HS-SPI read command sequence data/control register 5	
HSSPIn_RDCSDC6	HS-SPI read command sequence data/control register 6	
HSSPIn_RDCSDC7	HS-SPI read command sequence data/control register 7	

3.1.12 Module identification register list

Abbreviated register name	Register name	Reference
HSSPIn_MID	HS-SPI module identification register	3.29

3.2 HS-SPI control register (HSSPIn_MCTRL)

This register controls the HS_SPI controller.

This register can be used to switch enable/disable for operations of this controller.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	SYNCON	MES	Reserved	Reserved	CSEN	MEN
Attribute	R	R	R/W	R	R	R0	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:6] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit5] SYNCON : "Synchronizer ON" synchronization circuit operation bit

bit	Description
0	Asynchronous mode Set this when QHCLK and serial operation clock are asynchronous. [Initial value]
1	Synchronous mode Set this when QHCLK and serial operation clock are synchronous.

Notes:

- *This bit is enabled only in command sequencer mode. "0" is read in direct mode.*
- *If changing this bit, perform changing when the MES bit is "0".*

[bit4] MES : "Module Enable Status" module enable state bit

bit	Description
0	This indicates the module operation is completely stopped. [Initial value]
1	The module operation is enabled.

[bit3:2] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit1] CSEN : "Command Sequencer Enable" command sequencer mode enable bit

bit	Description
0	Direct mode is enabled. [Initial value]
1	Command sequencer mode is enabled.

Notes:

- *To switch to direct mode while the HS_SPI controller operates (MEN="1", MES="1") in command sequencer mode,*
  perform the procedure below.
  1) Change to direct mode (CSEN="0").
  2) Wait for detection that slave select is deselected (HSSPIn_TXF.TSSRS="1" or HSSPIn_RXF.RSSRS="1").
  3) Disable the operations of the HS_SPI controller (set to MEN="0").
  4) Wait until the HS_SPI controller becomes prohibited state (MES="0").
  5) Set each register in direct mode.
  6) Enable the operations of the HS_SPI controller (set to MEN="1").
  7) Wait until the HS_SPI controller becomes operating state (MES="1").
  When MES="1" after the procedure above is performed, direct mode can be used.
- *To switch to command sequencer mode while the HS_SPI controller operates (MEN="1", MES="1") in direct mode,*
  perform the procedure below.
  1) Disable the operations of the HS_SPI controller (set to MEN="0").
  2) Wait for detection that slave select is deselected (HSSPIn_TXF.TSSRS="1" or HSSPIn_RXF.RSSRS="1").
  3) Wait until the HS_SPI controller operation becomes prohibited state (MES="0").
  4) Change to command sequencer mode (set to CSEN="1").
  5) Set each register in command sequencer mode.
  6) Enable the operations of the HS_SPI controller (set to MEN="1").
  7) Wait until the HS_SPI controller becomes operating state (MES="1").
  When MES="1" after the procedure above is performed, command sequencer mode can be used.

[bit0] MEN : "Module Enable" module enable bit

bit	Description
0	The HS_SPI controller is disabled. [Initial value]
1	This controller is enabled. After setting each register of the HS_SPI controller, set "1" to this bit.

3.3 HS-SPI peripheral communication set register (HSSPIn_PCC0 to 3)

This register set performs various settings related to serial communication of slave select 0 to 3. The software uses these registers to set according to the specifications of serial devices corresponding to slave select (0 to 3). In command sequencer mode, use the register set with same settings.

These registers (0 to 3) have same bit field configurations. This chapter explains the HSSPIn_PCC0 register for details.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	RDDSEL [1]	RDDSEL [0]	WRDSEL [3]	WRDSEL [2]	WRDSEL [1]	WRDSEL [0]	SAFE SYNC
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	CDRS [6]	CDRS [5]	CDRS [4]	CDRS [3]	CDRS [2]	CDRS [1]	CDRS [0]	SENDIAN
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SDIR	SS2CD [1]	SS2CD [0]	SSPOL	RTM	ACES	CPOL	CPHA
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0)	0	0	0	0	0

[bit31:23] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit22:21] RDDSEL[1:0] : "Read Deselect Time" read deselect time set bits

bit	Description
0x0 to 0x3	Used as the deselect time. 0:1 serial clock time 1:2 serial clock time 2:3 serial clock time 3:4 serial clock time (initial value)

This bit is used in the deselect time of slave select during reading.

This bit is enabled only in command sequencer mode.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit20:17] WRDSEL[3:0] : "Write/different command Deselect Time" write or different command deselect time set bits

bit	Description
0x0-0xF	Used as the deselect time. 0:1 serial clock time 1:2 serial clock time 2:3 serial clock time 3:4 serial clock time 4:5 serial clock time 5:6 serial clock time 6:7 serial clock time 7:8 serial clock time 8:9 serial clock time 9:10 serial clock time 10:11 serial clock time 11:12 serial clock time 12:13 serial clock time 13:14 serial clock time 14:15 serial clock time 15:16 serial clock time (initial value)

This bit is used in the deselect time of slave select during writing or when a different command is executed.

This bit is enabled only in command sequencer mode.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit16] SAFESYNC : "Safe Synchronization" safe synchronization bit

bit	Description
0	The HS_SPI controller operates normally. In the period when serial transfer is performed, for safe re-synchronization, the preset delay will not be added.
1	The preset delay in which the serial communication can be safely re-synchronized will be added. [Initial value]

For setting of this bit, refer to "2.4 Direct mode" and "2.5 Command Sequencer Mode".

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit15:9] CDRS[6:0] : "Clock Division Ratio Select " clock division ratio set bits

bit	Description
0x00	Reserved [initial value]
0x01 to 0x7F	The followings are meanings in decimal. 1: 2 divisions. 2: 4 divisions. 3: 6 divisions. ... 127: 254 divisions. The clock frequency is divided by two times of CDRS setting value. When the clock frequency before division is Fi and the clock frequency after division is Fo, the relationship is expressed as the following formula. Fo = Fi / (2 x CDRS)

Set 0x00 to CDRS and serial communication is prohibited. Set to 0x01 or more.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit8] SENDIAN : "SPI Endian" endian set bit

bit	Description
0	It will be big endian. [Initial value] The byte data on serial interface will be sorted from the upper side to the lower side of word data.
1	It will be little endian. The byte data on serial interface will be sorted from the lower side to the upper side of word data.

This bit sets the order of byte data within 1 word data. For details, refer to "2.3.4 Endian of serial interface" in "2.3 Serial Interface".

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit7] SDIR : "Shift Direction " shift direction set bit

bit	Description
0	The byte order is set from the highest bit to the lowest bit. [Initial value]
1	The byte order is set from the lowest bit to the highest bit.

This bit determines the order for bit transfer within field (shift direction). For details of operations, refer to "2.3.3 Shift direction" in "2.3 Serial Interface".

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit6:5] SS2CD[1:0] : "Slave-Select to Clock Delay "

Delay time set bit from slave select to clock start

Delays from fluctuation point of slave select to the fluctuation point of SCLK in SCLK cycle unit.

When HSSPIn_PCC0 to 3.CPHA are "0", delay from slave select assert start to the first fluctuation point of SCLK will be "SS2CD + 1.5" cycle in SCLK.

When HSSPIn_PCC0 to 3.CPHA are "1", delay from slave select assert start to the first fluctuation point of SCLK will be "SS2CD + 1.0" cycle in SCLK.

Figure 2-3 shows the operations of SS2CD.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit4] SSPOL : "Slave Select Polarity" slave select polarity set bit

bit	Description
0	It will be negative logic (activated with "L"). [Initial value]
1	It will be positive logic (activated with "H").

This bit sets the polarity of slave select.

Figure 2-3 shows the operations when SSPOL="0". When SSPOL="1", the polarity of "Slave Select" in the figure will be reversed.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit3] RTM : "Use retimed clock for Capturing the data" timing compensation set bit

bit	Description
0	Capture the reception serial data without using the timing compensated clock. [Initial value]
1	Capture the reception serial data by using the timing compensated clock.

This bit sets whether or not to use the timing compensated serial clock, and "1" is set if the margin of the setup hold time of reception serial data is insufficient.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit2] ACES : "Active Clock Edges are Same" serial data transmission/reception timing set bit

bit	Description
0	Different clock edges are used for data transmission and reception (one is rising and another is falling). [Initial value]
1	Reserved

This bit sets whether or not to use common fluctuation point (rising/falling) of synchronized clock both for data transmission and reception. Figure 2-3 shows the operations of the ACES bit.

Note:

- *Setting "1" to this bit is prohibited.*

[bit1] CPOL : "Clock Polarity" serial clock polarity set bit

bit	Description
0	"L" level is specified. [Initial value]
1	"H" level is specified.

This bit specifies the level of SCLK in the period when serial transfer is not performed.

Figure 2-3 shows the operations of CPOL.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

[bit0] CPHA : "Clock Phase" clock phase set bit

bit	Description
0	Captured at odd-numbered edge. [Initial value]
1	Captured at even-numbered edge.

This bit selects serial clock edge when capturing reception data. Because the clock edge used can be set both for rising/falling, if counting both edges (starting at the first one) to name them odd-numbered or even-numbered, the description above can be applied.

Figure 2-3 shows the operations of CPHA.

Note:

- *If changing this bit, perform changing when the HS_SPI controller operation is disabled (HSSPIn_MCTRL.MEN="0", MES="0").*

3.4 HS-SPI transmission interrupt factor register (HSSPIn_TXF)

This register indicates the state of the transmission interrupt flags. These flag bits can operate in direct mode.

The interrupt enable bit is prepared for each interrupt flag, and by enabling the interrupt enable bit, interrupts can be generated.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	TSSRS	TFMTS	TFLETS	TFUS	TFOS	TFES	TFFS
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit6] TSSRS : "Slave Select Released Interrupt" slave select cancellation detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the slave select is released by master.

When the HSSPIn_TXE.TSSRE bit is "1" and this bit is "1", a transmit interrupt occurs. When "1" is written to the HSSPIn_TXC.TSSRC bit, this bit is set to "0".

[bit5] TFMTS : "TX-FIFO Fill Level is More than Threshold Interrupt"

TX-FIFO threshold exceeding detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the TX-FIFO level (number of valid data) has exceeded its threshold. Conditions where this bit is set to "1" are as follows.

- HSSPIn_DMSTATUS.TXFLEVEL ＞ HSSPIn_FIFOCFG.TXFTH

This bit is set to "1" while the condition above is met.

When the HSSPIn_TXE.TFMTE bit is "1" and this bit is "1", a transmit interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- HSSPIn_DMSTATUS.TXFLEVEL ≦ HSSPIn_FIFOCFG.TXFTH
- Writing "1" to the HSSPIn_TXC.TFMTC bit

[bit4] TFLETS : "TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

TX-FIFO threshold falling detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the TX-FIFO level (number of valid data) has fallen below its threshold. Conditions where this bit is set to "1" are as follows.

- HSSPIn_DMSTATUS.TXFLEVEL ≦ HSSPIn_FIFOCFG.TXFTH

This bit is set to "1" while the condition above is met.

When the HSSPIn_TXE.TFLETE bit is "1" and this bit is "1", a transmit interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- HSSPIn_DMSTATUS.TXFLEVEL > HSSPIn_FIFOCFG.TXFTH
- Writing "1" to the HSSPIn_TXC.TFLETS bit

[bit3] TFUS : "TX-FIFO Underrun Interrupt" TX-FIFO underrun detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the TX-FIFO underrun is caused.

When TX-FIFO is Empty, if data transfer from TX-FIFO to transmission shift register is attempted, this bit will be "1".

When HSSPIn_TXE.TFUE is "1" and this bit is "1", a transmit interrupt occurs.

When "1" is written to the HSSPIn_TXC.TFUC bit, this bit is set to "0".

[bit2] TFOS : "TX-FIFO Overrun Interrupt" TX-FIFO overrun detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the TX-FIFO overrun is caused.

When TX-FIFO is Full, if transmission data is written to TX-FIFO by the HS_SPI controller, this bit is set to "1".

When the HSSPIn_TXE.TFOE bit is "1" and this bit is "1", a transmit interrupt occurs.

When "1" is written to the HSSPIn_TXC.TFOC bit, this bit is set to "0".

Note:

- *If this flag is set to "1", the data will not be sent successfully. In addition, other flag also do not operate correctly. Clear TX-FIFO by writing "1" to the HSSPIn_FIFOCFG.TXFLSH bit.*

[bit1] TFES : "TX-FIFO and Shift Register are Empty Interrupt"

TX-FIFO and shift register empty detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates TX-FIFO and transmission shift register are Empty. This bit is set to "1" while TX-FIFO and transmission shift register are Empty.

When the HSSPIn_TXE.TFEE bit is "1" and this bit is "1", a transmit interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- TX-FIFO or transmission shift register has a valid data.
- Writing "1" to the HSSPIn_TXC.TFEC bit

[bit0] TFFS : "TX-FIFO Full Interrupt" TX-FIFO Full detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the TX-FIFO has been Full. "1" is set when TX-FIFO is Full. This bit is set to "1" while TX-FIFO is Full.

When the HSSPIn_TXE.TFFE bit is "1" and this bit is "1", a transmit interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- TX-FIFO is not Full state.
- Writing "1" to the HSSPIn_TXC.TFFC bit

3.5 HS-SPI transmission interrupt enable register (HSSPIn_TXE)

This register set whether or not to generate transmission interrupt by each bit of the HSSPIn_TXF register.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	TSSRE	TFMTE	TFLETE	TFUE	TFOE	TFEE	TFFE
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit6] TSSRE : "Slave Select Released Interrupt Enable"

slave select cancellation detection enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TSSRS bit.

[bit5] TFMTE : "TX-FIFO Fill Level is More than Threshold Interrupt Enable"

TX-FIFO threshold exceeding detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TFMTS bit.

[bit4] TFLETE : "TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt Enable"

TX-FIFO threshold falling detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TFLETS bit.

[bit3] TFUE : "TX-FIFO Underrun Interrupt Enable"

TX-FIFO underrun detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TFUS bit.

[bit2] TFOE : "TX-FIFO Overrun Interrupt Enable"

TX-FIFO overrun detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TFOS bit.

[bit1] TFEE : "TX-FIFO and Shift Register are Empty Interrupt Enable"

TX-FIFO and shift register empty detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TFES bit.

[bit0] TFFE : "TX-FIFO Full Interrupt Enable" TX-FIFO Full detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the transmission interrupt by the HSSPIn_TXF.TFFS bit.

3.6 HS-SPI transmission interrupt clear register (HSSPIn_TXC)

This register clears the state of the HSSPIn_TXF register to "0".

By writing "1" to this register bit, the related interrupt factor flags within the HSSPIn_TXF register can be cleared.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	TSSRC	TFMTC	TFLETC	TFUC	TFOC	TFEC	TFFC
Attribute	R	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit6] TSSRC : "Slave Select Released Interrupt Clear" slave select cancellation detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TSSRS.

[bit5] TFMTC : "TX-FIFO Fill Level is More than Threshold Interrupt Clear"

TX-FIFO threshold exceeding detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TFMTS.

[bit4] TFLETC : "TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

TX-FIFO threshold falling detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TFLETS.

[bit3] TFUC : "TX-FIFO Underrun Interrupt Clear" TX-FIFO underrun detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TFUS.

[bit2] TFOC : "TX-FIFO Overrun Interrupt Clear" TX-FIFO overrun detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TFOS.

[bit1] TFEC : "TX-FIFO and Shift Register are Empty Interrupt Clear"

TX-FIFO and shift register empty detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TFES.

[bit0] TFFC : "TX-FIFO Full Interrupt Clear" TX-FIFO Full detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_TXF.TFFS.

3.7 HS-SPI reception interrupt factor register (HSSPIn_RXF)

This register indicates the state of the reception interrupt flags. These flag bits can operate in direct mode.

The interrupt enable bit is prepared for each interrupt flag, and by enabling the interrupt enable bit, interrupts can be generated.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	RSSRS	RFMTS	RFLETS	RFUS	RFOS	RFES	RFFS
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit6] RSSRS : "Slave Select Released Interrupt"

slave select cancellation detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the slave select is released by master.

When the HSSPIn_RXE.RSSRE bit is "1" and this bit is "1", a reception interrupt occurs.

When "1" is written to the HSSPIn_RXC.RSSRC bit, this bit is set to "0".

[bit5] RFMTS : "RX-FIFO Fill Level is More than Threshold Interrupt"

RX-FIFO threshold exceeding detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the RX-FIFO level (number of valid data) has exceeded its threshold. Conditions where this bit is set to "1" are as follows.

- HSSPIn_DMSTATUS.RXFLEVEL ＞ HSSPIn_FIFOCFG.RXFTH

This bit is set to "1" while the condition above is met.

When the HSSPIn_RXE.RFMTE bit is "1" and this bit is "1", a reception interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- HSSPIn_DMSTATUS.RXFLEVEL ≦ HSSPIn_FIFOCFG.RXFTH
- Writing "1" to the HSSPIn_RXC.RFMTC bit

[bit4] RFLETS : "RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

RX-FIFO threshold falling detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the RX-FIFO level (number of valid data) has fallen below its threshold. Conditions where this bit is set to "1" are as follows.

- HSSPIn_DMSTATUS.RXFLEVEL ≦ HSSPIn_FIFOCFG.RXFTH

This bit is set to "1" while the condition above is met.

When the HSSPIn_RXE.RFLETE bit is "1" and this bit is "1", a reception interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- HSSPIn_DMSTATUS.RXFLEVEL > HSSPIn_FIFOCFG.RXFTH
- Writing "1" to the HSSPIn_RXC.RFLETS bit

[bit3] RFUS : "RX-FIFO Underrun Interrupt" RX-FIFO underrun detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the RX-FIFO underrun is caused.

When RX-FIFO is Empty, if data is read from RX-FIFO, this bit will be "1".

When HSSPIn_RXE.RFUE is "1" and this bit is "1", a reception interrupt occurs.

When "1" is written to the HSSPIn_RXC.RFUC bit, this bit is set to "0".

Note:

- *If this flag is set to "1", clear RX-FIFO by writing "1" to the HSSPIn_FIFOCFG.RXFLSH bit.*

[bit2] RFOS : "RX-FIFO Overrun Interrupt" RX-FIFO overrun detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the RX-FIFO overrun is caused.

When RX-FIFO is Full, if reception data is attempt to write to RX-FIFO by the HS_SPI controller, this bit is set to "1". If overrun occurs, the data already written to RX-FIFO will not be overwritten and will be protected, and the data where overrun occurs will be discarded.

The HS_SPI can receive up to 17 words, including RX-FIFO and reception shift register. If receiving data is performed for even 1 bit, it is considered as overrun and RFOS is set to "1". When the HSSPIn_RXE.RFOE bit is "1" and this bit is "1", a reception interrupt occurs.

When "1" is written to the HSSPIn_RXC.RFOC bit, this bit is set to "0".

[bit1] RFES : "RX-FIFO is Empty Interrupt" RX-FIFO Empty detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the RX-FIFO has been Empty. This bit is set to "1" while RX-FIFO is Empty.

When the HSSPIn_RXE.RFEE bit is "1" and this bit is "1", a reception interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- RX-FIFO has valid data.
- Writing "1" to the HSSPIn_RXC.RFEC bit

[bit0] RFFS : "RX-FIFO Full Interrupt" RX-FIFO Full detection bit

bit	Description
0	The interrupt factors were not detected. [Initial value]
1	The interrupt factors were detected.

This bit is the interrupt factor flag which indicates the RX-FIFO has been Full. "1" is set when RX-FIFO is Full. This bit is set to "1" while RX-FIFO is Full.

When the HSSPIn_RXE.RFFE bit is "1" and this bit is "1", a reception interrupt occurs.

This bit will be cleared to "0" when all conditions below are met.

- RX-FIFO is not Full state.
- Writing "1" to the HSSPIn_RXC.RFFC bit

3.8 HS-SPI reception interrupt enable register (HSSPIn_RXE)

This register set whether or not to generate reception interrupt by each bit of the HSSPIn_RXF register.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	RSSRE	RFMTE	RFLETE	RFUE	RFOE	RFEE	RFFE
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit6] RSSRE : "Slave Select Released Interrupt Enable"

slave select cancellation detection enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RSSRS bit.

[bit5] RFMTE : "RX-FIFO Fill Level is More than Threshold Interrupt Enable"

RX-FIFO threshold exceeding detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RFMTS bit.

[bit4] RFLETE : "RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt Enable"

RX-FIFO threshold falling detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RFLETS bit.

[bit3] TFUE : "TX-FIFO Underrun Interrupt Enable"

RX-FIFO underrun detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RFUS bit.

[bit2] RFOE : "RX-FIFO Overrun Interrupt Enable"

RX-FIFO overrun detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RFOS bit.

[bit1] RFEE : "RX-FIFO and Shift Register are Empty Interrupt Enable"

RX-FIFO and shift register empty detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RFES bit.

[bit0] RFFE : "RX-FIFO Full Interrupt Enable" RX-FIFO Full detection interrupt enable bit

bit	Description
0	Interrupts are disabled. [Initial value]
1	Interrupts are enabled.

This bit enables the reception interrupt by the HSSPIn_RXF.RFFS bit.

3.9 HS-SPI reception interrupt clear register (HSSPIn_RXC)

This register clears the state of the HSSPIn_RXF register to "0".

By writing "1" to this register bit, the related interrupt factor flags within the HSSPIn_RXF register can be cleared.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	RSSRC	RFMTC	RFLETC	RFUC	RFOC	RFEC	RFFC
Attribute	R	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

[bit31:7] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit6] RSSRC : "Slave Select Released Interrupt Clear" slave select cancellation detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RSSRS.

[bit5] RFMTC : "RX-FIFO Fill Level is More than Threshold Interrupt Clear"

RX-FIFO threshold exceeding detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RFMTS.

[bit4] RFLETC : "RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

RX-FIFO threshold falling detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RFLETS.

[bit3] RFUC : "RX-FIFO Underrun Interrupt Clear" RX-FIFO underrun detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RFUS.

[bit2] RFOC : "RX-FIFO Overrun Interrupt Clear" RX-FIFO overrun detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RFOS.

[bit1] RFEC : "RX-FIFO and Shift Register are Empty Interrupt Clear"

RX-FIFO and shift register empty detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RFES.

[bit0] RFFC : "RX-FIFO Full Interrupt Clear" RX-FIFO Full detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_RXF.RFFS.

3.10 HS-SPI fault interrupt factor register (HSSPIn_FAULTF)

This register indicates the state of the violation interrupt flags. The violation interrupt cannot be masked with the HS_SPI controller.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	DRCBSFS	DWCBSFS	PVFS	WAFS	UMAFS
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:5] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit4] DRCBSFS : "DMA Read Channel Block Size Fault"

DMA read block size violation detection bit

bit	Description
0	There is no violation of the DMA block size. [Initial value]
1	There is a violation of the DMA block size.

This bit is the interrupt factor flag which indicates a violation of the block size in DMA read channel occurs in direct mode.

When the reception block counter of the HS_SPI controller is 0 and valid read access to RX-FIFO is performed, this bit will be "1".

This bit enables violation detection when HSSPIn_DMDMAEN.RXDMAEN is "1".

When "1" is written to the HSSPIn_FAULTC.DRCBSFC bit, this bit is set to "0".

Interrupt factors by this bit cannot be masked.

[bit3] DWCBSFS : "DMA WriteChannel Block Size Fault"

DMA write block size violation detection bit

bit	Description
0	There is no violation of the DMA block size. [Initial value]
1	There is a violation of the DMA block size.

This bit is the interrupt factor flag which indicates a violation of the block size in DMA write channel occurs in direct mode.

When the transmission block counter of the HS_SPI controller is 0 and valid write access to TX-FIFO is performed, this bit will be "1".

This bit enables violation detection when HSSPIn_DMDMAEN.TXDMAEN is "1".

When "1" is written to the HSSPIn_FAULTC.DWCBSFC bit, this bit is set to "0".

Interrupt factors by this bit cannot be masked.

[bit2] PVFS : "Protection Violation Fault" protection violation detection bit

bit	Description
0	There is no protection violation. [Initial value]
1	There is a protection violation.

This bit is the interrupt factor flag which indicates a protection violation occurs, and it is set when any one of the following conditions is met.

- Read access to the register area where no register exists

  (the lower 10-bit of the AHB address within register area is read to 0x100 to 0x3FF).

- Invalid access to the register.

  The access to HSSPIn_TXFIFO0 to 15 and HSSPIn_RXFIFO0 to 15 registers does not match the FIFO width (HSSPIn_FIFOCFG.FWIDTH). However, when FWIDTH="10" (24-bit setting), word access is valid.

- Write access to the register area which is write protected.

  (writing to the AHB address [9:0]=0x100 to 0x3FF within the register area, or writing to the area reserved in byte unit. For example, writing half words or bytes to the upper 16-bit of the HSSPIn_FIFOCFG register.)

This bit enables violation detection when HSSPIn_DMDMAEN.TXDMAEN is "1".

When "1" is written to the HSSPIn_FAULTC.PVFC bit, this bit is set to "0".

Interrupt factors by this bit cannot be masked.

[bit1] WAFS : "Write Access Fault" write access violation detection bit

bit	Description
0	There is no write access violation. [Initial value]
1	There is a write access violation.

This bit is the interrupt factor flag which indicates a write access violation occurs.

In command sequencer mode, when HSSPIn_CSCFG.SRAM is set to "0" and write access to the area where the external serial memory in system memory is allocated is performed, this bit will be "1".

When "1" is written to the HSSPIn_FAULTC.WAFC bit, this bit is set to "0".

Interrupt factors by this bit cannot be masked.

[bit0] UMAFS : "Unmapped Memory Access Fault" memory access violation detection bit

bit	Description
0	There is no UMA violation. [Initial value]
1	There is a UMA violation.

This bit is the interrupt factor flag which indicates an access violation to the memory area which is not allocated.

This bit will be "1" by one of the following conditions.

- When, in direct mode (HSSPIn_MCTRL.CSEN="0"), the AHB master accesses 256 M byte memory space starting with HS_SPI Base Address.
- When, in command sequencer mode, (HSSPIn_MCTRL.CSEN="1"), the AHB master accesses the external serial memory area which is not enabled The HSSPIn_CSCFG.SSEL0EN to SSEL3EN bits indicate the state which is not enabled.
- When, in command sequencer mode, (HSSPIn_MCTRL.CSEN="1"), the AHB master accesses the area where the external serial memory is not allocated The HSSPIn_CFGFG.MSEL bit indicates the allocation.
- When the HS_SPI controller is disabled (HSSPIn_MCTRL.MEN="0"), the AHB master accesses the area where the external serial memory of the system memory space is allocated

When "1" is written to the HSSPIn_FAULTC.UMAFC bit, this bit is set to "0".

Interrupt factors by this bit cannot be masked.

3.11 HS-SPI fault interrupt clear register (HSSPIn_FAULTC)

This register clears the state of the HSSPIn_FAULTF register to "0".

By writing "1" to this register bit, the related interrupt factor flags within the HSSPIn_FAULTF register can be cleared.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	DRCBSFC	DWCBSFC	PVFC	WAFC	UMAFC
Attribute	R	R	R	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

[bit31:5] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit4] DRCBSFC : "DMA Read Channel Block Size Fault Interrupt Clear"

DMA read block size violation detection bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_FATLTF.DRCBSFS.

[bit3] DWCBSFC : "DMA Write Channel Block Size Fault Interrupt Clear"

DMA write block size violation detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_FATLTF.PVFS.

[bit2] PVFC : "Protection Violation Fault Interrupt Clear" protection violation detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_FATLTF.PVFS.

[bit1] WAFC : "Write Access Fault Interrupt Clear" write access violation detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_FATLTF.WAFS.

[bit0] UMAFC : "Unmapped Memory Access Fault Interrupt Clear"

Memory access violation detection clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	Writing "1" clears the interrupt flag.

This bit clears HSSPIn_FATLTF.UMAFS.

3.12 HS-SPI direct mode set register (HSSPIn_DMCFG)

This register sets the following operations of the HS_SPI controller.

- Whether to control the stop operation of transmission/reception by software control or byte counter mode
- Whether or not to perform transfer start with the iMSTART pin

This register is enabled when the HS_SPI controller is in direct mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SSDC	Reserved
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	1

[bit7:2] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit1] SSDC : "Slave Select Deassertion Control" slave select deassertion set bit

bit	Description
0	It is controlled by the software. Setting "1" to the HSSPIn_DMSTOP.STOP bit stops transmission/reception. [Initial value]
1	This is byte counter mode. When the HSSPIn_DMBCS.BCS bit becomes 0, transmission/reception will be stopped.

When the conditions of this bit are met, transmission/reception stops and negates slave select.

[bit0] Reserved: Reserved bit

Reserved bits are read as "1", and must be set to "1" when writing.

3.13 HS-SPI direct mode DMA enable register (HSSPIn_DMDMAEN)

This register enables/disables the DMA service requests.

In addition, this register is enabled only in direct mode.

bit	15	6	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXDMAEN	RXDMAEN
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7:2] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit1] TXDMAEN : "TXDMA Enable" transmission DMA enable bit

bit	Description
0	The transmission DMA is disabled. [Initial value]
1	The transmission DMA is enabled.

[bit0] RXDMAEN : "RXDMA Enable" reception DMA enable bit

bit	Description
0	The reception DMA is disabled. [Initial value]
1	The reception DMA is enabled.

Notes:

- *This register sets to enable/disable the DMA transfer requests from the HS-SPI controller. When using the DMA transfer requests, use this by always setting the DREQENB register of DSTC to "1". If it is set "0", the transfer request will not received by DSTC and the transfer processing will not be performed correctly. For details, refer to the "DSTC" chapter in "Peripheral Manual".*
- *Set DES0.ACK[1:0] of DSTC to ACK=01. If it is set to ACK=00, the DMA transfer will not be performed correctly. For details, refer to the "DSTC" chapter in "Peripheral Manual".*
- *Before setting this register to "1", set the DMA BRIDGE control register (DBCNT) to "1". In addition, after setting this register to "0", set the DMA BRIDGE control register (DBCNT) to "0". For details, refer to the "HS_SPI Prescaler" chapter.*

3.14 HS-SPI direct mode transfer start control register (HSSPIn_DMSTART)

This register sets the start of serial transfer.

This register is enabled only in direct mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	START
Attribute	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7:1] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit0] START : Transfer start bit

bit	Description
0	Writing "0" is invalid. Indicates that the start of transfer is completed. [Initial value]
1	If "1" is written, the transfer start in direct mode is set.

After "1" is written to this bit and serial transfer is started, this bit will automatically return to "0". Although this bit can be set during serial transfer, the operations are as follows.

- Transmission operation:

  When the previous serial transfer is completed (the TSSRS bit is set) and the transmission data is written to TX-FIFO, the serial transmission operation is started. After transfer is started, this bit will be cleared automatically, as same as the above case.

- Reception operation:

  When the previous serial transfer is completed (the RSSRS bit is set), the next serial reception operation is started. After transfer is started, this bit will be cleared automatically, as same as the above case.

In TX-and-RX mode, transmission operations take priority.

3.15 HS-SPI direct mode transfer stop control register (HSSPIn_DMSTOP)

When the HS_SPI controller is in direct mode and software control mode (HSSPIn_DMCFG.SSDC="0"), this register performs to stop transmission/reception.

If the STOP bit of this register is set, clearing the STOP bit is prohibited until the current serial transfer is stopped.

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	STOP
Attribute	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:9] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit8] STOP : Transfer stop bit

bit	Description
0	The transfer stop is not set. [Initial value]
1	The transfer stop is set.

This bit is enabled only when setting to software control (HSSPIn_DMCFG.SSDC="0"). Under conditions other than those, writing to this bit is ignored.

When "1" is written to this bit, slave select operations vary depending on the following conditions.

- TX-Only mode:

  This bit is set to "1", and transmission operation will be finished when all data within TX-FIFO is transmitted.

- RX-Only mode:

  This bit is set to "1", and reception operation will be stopped (SCLK output operation is stopped and SSELm(m=0 to 3) will be negated) when reception operation which is working to the reception shift register is completed. For example, if the FIFO bit width is 32-bit (HSSPIn_FIFOCFG.FWIDTH[1:0]="11"), at the time 32-bit data is received within the reception shift register, the data is transferred to RX-FIFO and the operation is stopped.
  If RX-FIFO is Full and in hold state (if slave select is asserted while serial clock is stopped), reading data of RX-FIFO will stop operations.

- TX-and-RX mode:

  This bit is set to "1", and transmission/reception operation will be stopped when all data within TX-FIFO is transmitted or RX-FIFO is Full.. When RX-FIFO is Full and serial interface is in hold state, reading reception data of RX-FIFO will stop transmission/reception.

Notes:

- *When "1" is written to this bit, until HSSPIn_RXF.RSSRS="1" is set, writing "0" to this bit is prohibited.*
- *This bit cannot be "0" automatically.*

3.16 HS-SPI direct mode slave select register (HSSPIn_DMPSEL)

This register lets the HS_SPI controller select one of four slave select signal to perform serial transfer.

This register is enabled only in direct mode.

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PSEL[1]	PSEL[0]
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit23:18] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit17:16] PSEL : "Peripheral Select" slave select bit

bit	Description
00	Slave 0 (SSEL0) is selected. [Initial value]
01	Slave 1 (SSEL1) is selected.
10	Slave 2 (SSEL2) is selected.
11	Slave 3 (SSEL3) is selected.

This bit selects one of four slave selections; SSEL0 to SSEL3, and enables it.

Note:

- *The selectable slave devices vary depending on the products you are using. Refer to "Data Sheet".*

3.17 HS-SPI direct mode transfer protocol set register (HSSPIn_DMTRP)

This register sets the protocol using for serial transfer.

This register is enabled only in direct mode.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	TRP[3]	TRP[2]	TRP[1]	TRP[0]
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:28] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit27:26] TRP : "Transfer Protocol" transfer protocol set bit (transfer direction)

bit	Description
00	The mode is TX-and-RX mode. In this case, set TRP[1:0] to "00". [Initial value]
01	The mode is RX-Only mode.
10	The mode is TX-Only mode.
11	Reserved (same operation as "00").

[bit25:24] TRP : "Transfer Protocol" transfer protocol set bit (transfer bit width)

bit	Description
00	Single bit. [Initial value]
01	Dual-bit (2-bit width). In this case, set TRP[3:2] to "01" or "10".
10	Quad-bit mode (4-bit width). In this case, set TRP[3:2] to "01" or "10".
11	Reserved (same operation as "00").

Note:

- *Even if you want to only perform transmission operations in TX-and-RX mode, you need to read reception data from RX-FIFO because the reception data is piled up in RX-FIFO.*

3.18 HS-SPI direct mode transfer byte count set register (HSSPIn_DMBCC)

This register sets the amount of data transfer in bytes for serial transfer. When the mode is byte counter mode (HSSPIn_DMCFG.SSDC="1"), this register is used.

This register is enabled only in direct mode.

bit	15	14	13	12	11	10	9	8
Field	BCC[15]	BCC[14]	BCC[13]	BCC[12]	BCC[11]	BCC[10]	BCC[9]	BCC[8]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	BCC[7]	BCC[6]	BCC[5]	BCC[4]	BCC[3]	BCC[2]	BCC[1]	BCC[0]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:0] BCC : "Byte Count Control" transfer byte count set value

This bit sets the amount of data in bytes for serial transfer. Before starting transmission/reception (HSSPIn_DMSTART.START="0"), set the amount of data in bytes with this bit.

This bit is captured by the HSSPIn_DMBCS.BCS bit when serial transfer starts, and each time transfer is performed to TX-FIFO/RX-FIFO and the shift register, "FWIDTH[1:0]+1" is reduced. When the HSSPIn_DMBCS.BCS bit becomes 0, the HS_SPI controller stops serial transfer, negates slave select and stops transmission/reception.

If the BCC bit is not multiple of the bit value set by the HSSPIn_FIFOCFG.FWIDTH, data will be transmitted/received in bytes specified by the HSSPIn_FIFOCFG.FWIDTH bit, and the last data will be transmitted/received for the remaining bytes. Regardless of the remaining bytes, access to FIFO needs to be performed in byte width specified by the HSSPIn_FIFOCFG.FWIDTH bit. For example, when BCC=9 and HSSPIn_FIFOCFG.FWIDTH=3, for the data up to 8 bytes, transmission/reception is performed by accessing FIFO with 32-bit length. FIFO access for the remaining 1 byte is performed in 32-bit length, and the lower 8 bits will be handled as valid data.

Note:

- *When using byte counter mode, set the BCC bit to "1" or more.*

3.19 HS-SPI direct mode transfer count remaining register (HSSPIn_DMBCS)

This is a read-only register to indicate the amount of transmission/reception currently remaining. This register is enabled when all the following conditions are met.

- Direct mode
- Byte counter mode for flow control (HSSPIn_DMCFG.SSDC="1")

bit	31	30	29	28	27	26	25	24
Field	BCS[15]	BCS[14]	BCS[13]	BCS[12]	BCS[11]	BCS[10]	BCS[9]	BCS[8]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	BCS[7]	BCS[6]	BCS[5]	BCS[4]	BCS[3]	BCS[2]	BCS[1]	BCS[0]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:16] BCS : "Byte Count Status" transfer remaining byte count

This is a read-only bit to indicate the data amount (in bytes) of transmission/reception currently remaining. This bit is enabled when all the following conditions are met.

- The HS_SPI controller is set to direct mode.
- HSSPIn_DMCFG.SSDC is set to "1" (byte counter mode).

When the HSSPIn_DMBCC.BCC bit is set and serial transfer is started, the HSSPIn_DMBCC.BCC bit is passed as the initial value of BCS, then each time data transfer is performed between TX-FIFO/RX-FIFO and the shift register, "FWIDTH[1:0]+1" is reduced from BCS. When this register becomes 0, transmission/reception will be stopped.

3.20 HS-SPI direct mode status register (HSSPIn_DMSTATUS)

This register includes status bits which indicate whether or not transmission/reception is performed and which indicate the valid number of data in current transmission/reception FIFO.

This register is enabled only in direct mode.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	TXFLEVEL[4]	TXFLEVEL[3]	TXFLEVEL[2]	TXFLEVEL[1]	TXFLEVEL[0]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	RXFLEVEL[4]	RXFLEVEL[3]	RXFLEVEL[2]	RXFLEVEL[1]	RXFLEVEL[0]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXACTIVE	RXACTIVE
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:21] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit20:16] TXFLEVEL[4:0] : "Current fill level of TX-FIFO" data remaining display bit within TX-FIFO

This bit indicates the valid data count of TX-FIFO. The first level of FIFO shows the data count indicated by "FWIDTH+1".

[bit15:13] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit12:8] RXFLEVEL[4:0] : "Current fill level of RX-FIFO" data remaining display bit within RX-FIFO

This bit indicates the valid data count of RX-FIFO. The first level of FIFO shows the data count indicated by "FWIDTH+1".

[bit1] TXACTIVE : transmission state bit

bit	Description
0	Serial transmission is not being performed. [Initial value]
1	Serial transmission is being performed.

This bit indicates the serial data transmission operation is in progress. This set "1" to the HSSPIn_DMSTART.START bit, this is set on serial interface when transmission is started, and this is cleared when transmission of the last bit of the transmission data is completed.

[bit0] RXACTIVE : reception state bit

bit	Description
0	Serial reception is not being performed. [Initial value]
1	Serial reception is being performed.

This bit indicates the serial data reception operation is in progress. This set "1" to the HSSPIn_DMSTART.START bit, this is set on serial interface when reception is started, and this is cleared when reception of the last bit of the data is completed.

3.21 HS-SPI TX-FIFO register (HSSPIn_TXFIFO0 to 15)

This register set is used for writing transmission data to TX-FIFO. 16 TX-FIFO data write registers are consecutively placed on the register map. All these 16 registers have the same functions. Such configuration is provided because the AHB protocol is not compatible with burst access to same address. This chapter explains HSSPIn_TXFIFO0, one of 16 registers, as an example.

Access this registers with same bus width as the FIFO width (HSSPIn_FIFOCFG.FWIDTH). If the bus width is mismatched for access, a protection violation occurs.

bit	31	30	29	28	27	26	25	24
Field	TXDATA [31]	TXDATA [30]	TXDATA [29]	TXDATA [28]	TXDATA [27]	TXDATA [26]	TXDATA [25]	TXDATA [24]
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TXDATA [23]	TXDATA [22]	TXDATA [21]	TXDATA [20]	TXDATA [19]	TXDATA [18]	TXDATA [17]	TXDATA [16]
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TXDATA [15]	TXDATA [14]	TXDATA [13]	TXDATA [12]	TXDATA [11]	TXDATA [10]	TXDATA [9]	TXDATA [8]
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TXDATA [7]	TXDATA [6]	TXDATA [5]	TXDATA [4]	TXDATA [3]	TXDATA [2]	TXDATA [1]	TXDATA [0]
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TXDATA[31:0] : "TX-FIFO Register 0" TX-FIFO0 write data

This bit field writes transmission data to TX-FIFO. This write operation does not depend on the order of HSSPIn_TXFIFO0 to HSSPIn_TXFIFO15 of this register.

Before writing to TXDATA[31:0], specify whether or not to output data with Hi-Z to the HSSPIn_FIFOCFG.TXCTRL bit, then write to TXDATA[31:0]. For details, refer to "2.4.5 Tri-state control of data output" in "2.4 Direct mode".

Write access to TXDATA[31:0] has to be performed in accordance of specification of the FIFO width (FWIDTH[1:0]). If this is ignored when writing, the write operation is invalid, an error is returned, and a violation interrupt (caused by PVFS) is generated.

FWIDTH[1:0]	Accessing TXDATA[31:0]
00	8-bit access
01	16-bit access
10	32-bit access
11	32-bit access

3.22 HS-SPI RX-FIFO register (HSSPIn_RXFIFO0 to 15)

This register set is used for reading reception data from RX-FIFO. 16 RX-FIFO data read registers are consecutively placed on the register map. All these 16 registers have the same functions. Such configuration is provided because the AHB protocol is not compatible with burst access to same address. This chapter explains HSSPIn_RXFIFO0, one of 16 registers, as an example.

Access this registers with same bus width as the FIFO width (HSSPIn_FIFOCFG.FWIDTH). If the bus width is mismatched for access, a protection violation occurs.

bit	31	30	29	28	27	26	25	24
Field	RXDATA [31]	RXDATA [30]	RXDATA [29]	RXDATA [28]	RXDATA [27]	RXDATA [26]	RXDATA [25]	RXDATA [24]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	RXDATA [23]	RXDATA [22]	RXDATA [21]	RXDATA [20]	RXDATA [19]	RXDATA [18]	RXDATA [17]	RXDATA [16]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	RXDATA [15]	RXDATA [14]	RXDATA [13]	RXDATA [12]	RXDATA [11]	RXDATA [10]	RXDATA [9]	RXDATA [8]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RXDATA [7]	RXDATA [6]	RXDATA [5]	RXDATA [4]	RXDATA [3]	RXDATA [2]	RXDATA [1]	RXDATA [0]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] RXDATA[31:0] : "RX-FIFO Register 0" RX-FIFO0 read data

This bit field reads reception data from RX-FIFO. This read operation does not depend on the order of HSSPIn_RXFIFO0 to HSSPIn_RXFIFO15 of this register.

Read access from RXDATA[31:0] has to be performed in accordance of specification of the FIFO width (FWIDTH[1:0]). If this is ignored when reading, the read operation is invalid, an error is returned, and a violation interrupt (caused by PVFS) is generated.

FWIDTH[1:0]	Accessing RXDATA[31:0] (data width on AHB)
00	8-bit access
01	16-bit access
10	32-bit access
11	32-bit access

3.23 HS-SPI FIFO set register (HSSPIn_FIFOCFG)

This register sets the operations of TX-FIFO and RX-FIFO. FIFO threshold value and FIFO data bit width can be set. In addition, TXFLSH and RXFLSH bits can make transmission/reception FIFO empty.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	TXFLSH	RXFLSH	TXCTRL	FWIDTH [1]	FWIDTH [0]
Attribute	R	R	R	W	W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TXFTH [3]	TXFTH [2]	TXFTH [1]	TXFTH [0]	RXFTH [3]	RXFTH [2]	RXFTH [1]	RXFTH [0]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	1	0	1	1	1

[bit31:13] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit12] TXFLSH : "TX-FIFO Flush" TX-FIFO clear bit

bit	Description
0	Writing "0" is invalid. The read value is always "0". [Initial value]
1	TX-FIFO is cleared by writing "1" and becomes Empty state.

This bit initializes the internal state of TX-FIFO. Note that the data is not cleared.

[bit11] RXFLSH : "RX-FIFO Flush" RX-FIFO clear bit

bit	Description
0	The write data to TX-FIFO (HSSPIn_TXFIFO0 to 15) will be transmitted as transmission data. [Initial value]
1	Control transmission as shown below in accordance of the lowest bit of valid bit range of the write data to TX-FIFO (HSSPIn_TXFIFO0 to 15). 0: The serial data output will be Hi-Z for 1 byte data. The data corresponding to TX-FIFO will not be transmitted. 1: The lower 8-bit data within TX-FIFO is used. Within this 8-bit data, the upper 4-bit data is transmitted, and, during the 4-bit time, the SDATA output will be Hi-Z.

After setting "1" or "0" to this bit, write data to the HSSPIn_TXFIFO0 to 15 registers. The written data will be transmitted with control by this bit. The relationship between the data and this bit is maintained within the HS_SPI controller, changing this bit during operation does not cause a problem.

Note:

- *When "1" is set to the TXCTRL bit, regardless of the FIFO bit width, the written data is handled as 8-bit. LSB of valid bit range set with the FWIDTH value corresponds "TX-FIFO data bit [0]" above.*

[bit10] TXCTRL : "TX-FIFO Control" TX-FIFO transmission data control bit

bit	Description
0	The write data to TX-FIFO (HSSPIn_TXFIFO0 to 15) will be transmitted as data as is. [Initial value]
1	The write data to TX-FIFO (HSSPIn_TXFIFO0 to 15) is controlled for transmission as shown below in accordance of the lowest bit in valid bit range. 0: The serial data output will be Hi-Z for 1 byte data. The data corresponding to TX-FIFO will not be transmitted. 1: The lower 8-bit data within TX-FIFO is used. Within this 8-bit data, the upper 4-bit data is transmitted, and, during the 4-bit time, the SDATA output will be Hi-Z.

After setting "1" or "0" to this bit, write data to the HSSPIn_TXFIFO0 to 15 registers. The written data will be transmitted with control by this bit. The relationship between the data and this bit is maintained within the HS_SPI controller, changing this bit during operation does not cause a problem.

Note:

- *When "1" is set to the TXCTRL bit, regardless of the FIFO bit width, the written data is handled as 8-bit. LSB of valid bit range set with the FWIDTH value corresponds "TX-FIFO data bit [0]" above.*

[bit9:8] FWIDTH[1:0] : "FIFO Width" FIFO bit width set value

bit	Description
00	TX-FIFO, RX-FIFO and transmission/reception data length are all 8-bit width. [Initial value]
01	TX-FIFO, RX-FIFO and transmission/reception data length are all 16-bit width.
10	TX-FIFO, RX-FIFO and transmission/reception data length are all 24-bit width.
11	TX-FIFO, RX-FIFO and transmission/reception data length are all 32-bit width.

This bit indicates the bit width of FIFO.

[bit7:4] TXFTH[3:0] : "TX-FIFO Threshold Level" TX-FIFO threshold

This bit indicates a threshold of TX-FIFO.

[bit3:0] RXFTH[3:0] : "RX-FIFO Threshold Level" RX-FIFO threshold

This bit indicates a threshold of RX-FIFO.

3.24 HS-SPI command sequencer set register (HSSPIn_CSCFG)

This register sets transfer protocol type, enable/disable for memory write, memory device volume, etc. which are used in command sequencer mode. Before enabling command sequencer mode, set this register.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	MSEL [3]	MSEL [2]	MSEL [1]	MSEL [0]
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	SSEL3EN	SSEL2EN	SSEL1EN	SSEL0EN
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	MBM [1]	MBM [0]	SRAM
Attribute	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:20] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit19:16] MSEL[3:0] : "Memory Device Selection bits" memory device select bit

Set this bit in accordance of each memory bank area of memory devices you are using. For details, refer to "2.5.2 32-bit memory address" in "2.5 Command Sequencer Mode".

[bit15:12] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit11] SSEL3EN : "Slave Select 3 Enable" slave select 3 enable bit

bit	Description
0	Disables access to the area where slave select 3 is allocated. [Initial value]
1	Enables access to the area where slave select 3 is allocated.

This bit enables slave select 3.

[bit10] SSEL2EN : "Slave Select 2 Enable" slave select 2 enable bit

bit	Description
0	Disables access to the area where slave select 2 is allocated. [Initial value]
1	Enables access to the area where slave select 2 is allocated.

This control bit indicates that slave select 2 becomes enable.

[bit9] SSEL1EN : "Slave Select 1 Enable" slave select 1 enable bit

bit	Description
0	Disables access to the area where slave select 1 is allocated. [Initial value]
1	Enables access to the area where slave select 1 is allocated.

This control bit indicates that slave select 1 becomes enable.

[bit8] SSEL0EN : "Slave Select 0 Enable" slave select 0 enable bit

bit	Description
0	Disables access to the area where slave select 0 is allocated. [Initial value]
1	Enables access to the area where slave select 0 is allocated.

This control bit indicates that slave select 0 becomes enable.

[bit7:3] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

–

[bit2:1] MBM[1:0] : "Multi Bit Mode" SPI data width set bit

bit	Description
00	Serial interface will be single bit. Received from SDATA[0] and transmitted from SDATA[1]. [Initial value]
01	Serial interface will be dual bit. Transmission/reception is performed through SDATA[1:0].
10	Serial interface will be quad bit. Transmission/reception is performed through SDATA[3:0].
11	Cannot be set because this is reserved.

This bit sets the bit width of serial interface. It depends on the setting of the HSSPIn_RDCSDC0 to 7.TRP bits and the HSSPIn_WRCSDC0 to 7.TRP bits. This bit is enabled in command sequencer mode.

[bit0] SRAM : "Serial SRAM /Serial Flash Memory Type Select"

"readable/writable" / "read only" select bit

bit	Description
0	Write access is disabled. [Initial value]
1	Write access is enabled.

Set "1" to this bit to perform writing to memory in command sequencer mode.

3.25 HS-SPI command sequencer idle timer set register (HSSPIn_CSITIME)

This register sets timer measuring idle time of the AHB interface. When this timer reaches the cycle count of this register value, transaction of serial interface will finish.
Before enabling command sequencer mode, set this register. This register is enabled in command sequencer mode.

bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ITIME [15]	ITIME [14]	ITIME [13]	ITIME [12]	ITIME [11]	ITIME [10]	ITIME [9]	ITIME [8]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	ITIME [7]	ITIME [6]	ITIME [5]	ITIME [4]	ITIME [3]	ITIME [2]	ITIME [1]	ITIME [0]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

[bit31:16] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit15:0] ITIME[15:0] : "Idle Timer" idle timer set value

This bit sets idle time of the AHB interface. This is enabled in command sequencer mode.

3.26 HS-SPI command sequencer address extend register (HSSPIn_CSAEXT)

This register extends the space allocated on memory in command sequencer mode.

By this, access to serial memory up to 16 G bytes can be virtually performed. If not using address extension function, set all bits of this register to 0.
This register is enabled in command sequencer mode.

bit	31	30	29	28	27	26	25	24
Field	AEXT [18]	AEXT [17]	AEXT [16]	AEXT [15]	AEXT [14]	AEXT [13]	AEXT [12]	AEXT [11]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	AEXT [10]	AEXT [9]	AEXT [8]	AEXT [7]	AEXT [6]	AEXT [5]	AEXT [4]	AEXT [3]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	AEXT [2]	AEXT [1]	AEXT [0]	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R0	R0	R0	R0	R0
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial value	0	0	0	0	0	0	0	0

[bit31:13] AEXT[18:0] : " Address Extension Bits" address extension bit

This bit extends address on memory in command sequencer mode. The value set here is used by command sequencer as the bit range [31:13] of memory address. At this time, according to the MSEL setting, [31:13] to [31:28] of the memory address are used.

The address generated by each slave select is the combination of this bit value and the AHB address value.

For details, refer to "2.5.2 32-bit memory address" in "2.5 Command Sequencer Mode".

If not using address extension, set 0x00000000 to this bit.

[bit12:0] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

3.27 HS-SPI read command sequence data/control register (HSSPIn_RDCSDC0 to 7)

This register set is configured of 8 sets of data/control registers, and this sets serial transfer phase generated from command sequencer during memory read operations. This register set is enabled in command sequencer mode.

This chapter explains the HSSPIn_RDCSDC0 register as an example because these 8 registers have the same structure.

bit	15	14	13	12	11	10	9	8
Field	RDCS DATA[7]	RDCS DATA[6]	RDCS DATA[5]	RDCS DATA[4]	RDCS DATA[3]	RDCS DATA[2]	RDCS DATA[1]	RDCS DATA[0]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	CONT	TRP[1]	TRP[0]	DEC
Attribute	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:8] RDCSDATA[7:0] : "Command Sequencer Data /Control Byte for Memory-Read transactions" read command sequencer data/control set value

This bit is the register set for writing data and control code with command sequence. DEC is the bit controlling transmission for transmitting RDCSDATA[7:0] as is or transmitting it with decoding. For details, refer to "2.5.4.1 Read command sequence" in "2.5 Command Sequencer Mode".

Note:

- *Setting the "end of list (DEC="1" & RDCSDATA[2:0]="111")" to the HSSPIn_RDCSDC0 register is prohibit.*

[bit7:4] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit3] CONT : "Continuous" Continuous command specification bit

bit	Description
0	The list is not omitted. [Initial value]
1	The list is omitted.

This bit controls for omitting the second and subsequent command transmission. For details of whether or not to be able to omit the second and subsequent command, refer to the command system in data sheet for the serial memory you are using. For details, refer to "2.5.4.3 Continuous instructions" in "2.5 Command Sequencer Mode".

Note:

- *For details of processing for quitting list omission, refer to "2.5.4.3 Continuous instructions" in "2.5 Command Sequencer Mode".*
- *Among the CONT bits of the HSSPIn_RDCSDC0 to 7 registers, setting "1" to the CONT bits of the HSSPIn_RDCSDC1 to 7 registers is prohibited.*

[bit2:1] TRP : "Transfer Protocol" serial interface width control bit

bit	Description
00	The bit width of serial interface follows the setting of HSSPIn_CSCFG.MBM[1:0]. [Initial value]
01	Dual bit
10	Quad bit
11	Single bit

This bit sets the bit width of serial interface.

[bit0] DEC : "Decode" decode control bit

bit	Description
0	RDCSDATA[7:0] will be transmitted as 1 byte data as is. [Initial value]
1	Decodes RDCSDATA[2:0] as the control code.

This bit is a control bit to distinguish whether RDCSDATA[7:0] is data or a control code. For details, refer to "2.5.4.1 Read command sequence" in "2.5 Command Sequencer Mode".

3.28 HS-SPI write command sequence data/control register (HSSPIn_WRCSDC0 to 7)

This register set is configured of 8 sets of data/control registers, and this sets serial transfer phase generated from command sequencer during memory write operations. This register set is enabled in command sequencer mode.

This chapter explains the HSSPIn_WRCSDC0 register as an example because these 8 registers have the same structure.

bit	15	14	13	12	11	10	9	8
Field	WRCS DATA[7]	WRCS DATA[6]	WRCS DATA[5]	WRCS DATA[4]	WRCS DATA[3]	WRCS DATA[2]	WRCS DATA[1]	WRCS DATA[0]
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	CONT	TRP[1]	TRP[0]	DEC
Attribute	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:8] WRCSDATA[7:0] : "Command Sequencer Data /Control Byte for Memory-Write transactions" write command sequencer data/control set value

This bit is the register set for writing data and control code given to command sequence. DEC is the bit controlling transmission for transmitting WRCSDATA[7:0] as is or transmitting it with decoding. For details, refer to "2.5.4.2 Write command sequence" in "2.5 Command Sequencer Mode".

[bit7:4] Reserved: Reserved bits

Reserved bits are read as "0", and must be set to "0" when writing.

[bit3] CONT : "Continuous" Continuous command specification bit

bit	Description
0	The list is not omitted. [Initial value]
1	The list is omitted.

Note:

- *Set "0" to the CONT bits of the WRCSDC0 to 7 registers..*

[bit2:1] TRP : "Transfer Protocol" serial interface width control bit

bit	Description
00	The bit width of serial interface follows the setting of HSSPIn_CSCFG.MBM[1:0]. [Initial value]
01	Dual bit
10	Quad bit
11	Single bit

This bit sets the bit width of serial interface.

Note:

- *If the end of list is not found among WRCSDC0 to 7, the TRP setting value of WRCSDC7 is used.*

[bit0] DEC : "Decode" decode control bit

bit	Description
0	WRCSDATA[7:0] will be transmitted as 1 byte data as is. [Initial value]
1	Decodes WRCSDATA[2:0] as the control code.

This bit distinguishes whether WRCSDATA[7:0] is data or a control code. For details, refer to "2.5.4.2 Write command sequence" in "2.5 Command Sequencer Mode".

3.29 HS-SPI module identification register (HSSPIn_MID)

This register indicates the controller-specific identification number . This identification number indicates the version of the HS_SPI controller.

bit	31	30	29	28	27	26	25	24
Field	MID [31]	MID [30]	MID [29]	MID [28]	MID [27]	MID [26]	MID [25]	MID [24]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	MID [23]	MID [22]	MID [21]	MID [20]	MID [19]	MID [18]	MID [17]	MID [16]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	MID [15]	MID [14]	MID [13]	MID [12]	MID [11]	MID [10]	MID [9]	MID [8]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	1	1	0

bit	7	6	5	4	3	2	1	0
Field	MID [7]	MID [6]	MID [5]	MID [4]	MID [3]	MID [2]	MID [1]	MID [0]
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	1	1	0	0	0	0

[bit31:0] MID[31:0] : "Module ID" module identification information bit

This bit field is read-only and indicates the unique module identification number of the HS_SPI controller.

This unique identification number is used for indicating the version of the HS_SPI controller.

4. Precautions for using the HS_SPI controller

This chapter describes precautions when using the HS_SPI controller.

4.1 General precautions for usage

This section describes general precautions for usage. Precautions for operations are explained respectively in the descriptions of operations and the descriptions of registers

■ Timing for changing registers

When the HS_SPI controller operations are disabled (HSSPIn_MCTRL.MEN="0", HSSPIn_MCTRL.MES="0"), change the following register or bit. Changing is disabled under other conditions.

- HSSPIn_MCTRL register (when MEN="1" is set, other bits can be changed simultaneously)
- HSSPIn_PCC0 to 3 registers (for all bits)
- HSSPIn_DMCFG.SSDC bit

In direct mode, before "1" is set to the HSSPIn_DMSTART.START bit
, change the following register or bit.

- HSSPIn_DMCFG.MSTARTEN bit
- HSSPIn_DMPSEL register (for all bits)

The other registers cannot be changed while non-FIFO serial interface is running (serial clock is running).

■ Operation mode change or bit width change in direct mode

In direct mode, to change operation mode (TX-Only, RX-Only, TX-and-RX) or serial interface bit width (single bit, dual bit, quad bit) while transmission/reception is operating, perform changing when serial interface is in hold state (slave select is asserted and serial clock is stopped).

■ Changing from direct mode to command sequencer mode

To change from direct mode to command sequencer mode, perform the procedure below when serial interface is in hold state (slave select is asserted and serial clock is stopped).

4. Set "0" to the HSSPIn_MCTRL.MEN bit.
5. Wait until the HSSPIn_TXF.TSSRS bit or the HSSPIn_RXF.RSSRS bit becomes "1".
6. Wait until the HSSPIn_MCTRL.MES bit becomes "0".

When the HSSPIn_MCTRL.MES bit becomes "0", the various setting of command sequencer mode can be set.

■ Changing from command sequencer mode to direct mode

To change from command sequencer mode to direct mode, perform the procedure below.

7. Set "0" to the HSSPIn_MCTRL.CSEN bit.
8. Wait until the HSSPIn_TXF.TSSRS bit or the HSSPIn_RXF.RSSRS bit becomes "1".
9. Set "0" to the HSSPIn_MCTRL.MEN bit.
10. Wait until the HSSPIn_MCTRL.MES bit becomes "0".

When the HSSPIn_MCTRL.MES bit becomes "0", the various setting of direct mode can be set.

■ FIFO of the HS_SPI controller

11. In direct mode, perform processing so that FIFO of the HS_SPI controller does not cause overrun or underrun. If overrun or underrun is caused, because the FIFO level does not have meaning, clear FIFO at that time.
12. Before starting serial transfer, clear FIFO by using HSSPIn_FIFOCFG.RXFLSH and TXFLSH.

5. Setting procedure example of the HS_SPI controller

This chapter describes a setting procedure example of the HS_SPI controller for serial memory.

It also describes the setting example of DSTC during DMA transfer in direct mode.

5.1 Direct mode, command sequencer mode

This section explains examples of procedure for data read/write for serial memory in direct mode and command sequencer mode.

5.1.1 Initial settings

1. HS_SPI controller initial setting

1) Set the format division value of serial interface.

Set all bits of the HSSPIn_PCC0 to 3 registers.

2) Set operation permission and clock selection for the HS_SPI controller.

Set HSSPIn_MCTRL.SYNCON, CDSS,CSEN, and MEN="1". Set "0" to SYNCON in direct mode.

Set "0" (direct) or "1"(command sequencer) to CSEN.

3) Wait until operations are enabled.

Wait until the HSSPIn_MCTRL.MES bit becomes "1". When the HSSPIn_MCTRL.MES bit becomes "1", the initial setting is completed.

5.1.2 Data read for serial memory (direct mode: data transfer by interrupt)

2. Initializing TX-FIFO/RX-FIFO and setting FIFO bit width, transmission threshold and reception threshold
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", FWIDTH="00", TXFTH="threshold", RXFTH="threshold".
3. Transmission control settings (byte counter mode).
    1) Set "1" to the HSSPIn_DMCFG.SSDC bit.
    2) Set transfer byte count to the HSSPIn_DMBCC register.
4. Enable setting for used slave select
    1) Set slave select to the HSSPIn_DMPSEL.PSEL bit.
5. Operation mode (TX-Only) and bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width".
6. Serial operation start
    1) Set HSSPIn_DMSTART="1".
7. Go to 7-1 when serial memory command and serial bit width of address is same, and go to 7-2 when they are different.

7-1. Command and address write to serial memory
    1) Write command and address to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit. Jump to 8.

7-2. Command and address write to serial memory
    1) Write command to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.
    4) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width" (address).
    5) Write address to TX-FIFOm(m=0 to 15).
    6) Set HSSPIn_TXE.TFEC="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.

8. Go to 8-1 when inserting Hi-Z and go to 9 when not inserting.

8-1. Hi-Z output to serial bus
    1) Set HSSPIn_FIFOCFG.TXCTRL="1".
    2) Write data to TXFIFOm(m=0 to 15) for Hi-Z output.
    3) Set HSSPIn_TXE. TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.

9. Disabling interrupt
    1) Set "0" to all bits of the HSSPIn_TXE register.
10. FIFO initialization, FIFO threshold and bit width setting
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", TXCTRL="0", FWIDTH="bit width", TXFTH="threshold value", RXFTH="threshold value".
11. Reception interrupt flag clear
    1) Set "1" to all bits of the HSSPIn_RXC register.
12. Operation mode (RX-Only) and serial bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="01"(RX-Only) and TRP[1:0]="bit width".
13. Reception interrupt enable setting
    1) Set HSSPIn_RXE.RSSRE="1" and RFMTE="1". Wait for interrupt by the HSSPIn_RXF.RSSRS or RFMTS bit.
14. Go to 14-1 when HSSPIn_RXF.RSSRS="0", and go to 14-2 when it is "1".

14-1. Processing for HSSPIn_RXF.RFMTS="1"
    1) Read data for the threshold setting (RXFTH+1) from RXFIFO.
    2) Set HSSPIn_RXC.RFMTC="1". Jump to 13.

14-2. Processing for HSSPIn_RXF.RSSRS="1"
    1) Check HSSPIn_DMSTATUS.RXFLEVEL.
    2) Read the reception data for HSSPIn_DMSTATUS.RXFLEVEL from RXFIFO.
    3) Set HSSPIn_RXC.RSSRC="1" and RFMTC="1".

15. Completing reading from serial memory
    1) Set "0" to all bits of the HSSPIn_RXE register.

5.1.3 Data read for serial memory (direct mode: DMA transfer)

2. Initializing TX-FIFO/RX-FIFO and setting FIFO bit width, transmission threshold and reception threshold
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", FWIDTH="00", TXFTH="threshold", RXFTH="threshold".
3. Transmission control settings (byte counter mode).
    1) Set "1" to the HSSPIn_DMCFG.SSDC bit.
    2) Set transfer byte count to the HSSPIn_DMBCC register.
4. Enable setting for used slave select
    2) Set slave select to the HSSPIn_DMPSEL.PSEL bit.
5. Operation mode (TX-Only) and bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width".
6. Serial operation start
    1) Set HSSPIn_DMSTART="1".
7. Go to 7-1 when serial memory command and serial bit width of address is same, and go to 7-2 when they are different.

7-1. Command and address write to serial memory
    1) Write command and address to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit. Jump to 8.

7-2. Command and address write to serial memory
    1) Write command to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.
    4) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width" (address).
    5) Write address to TX-FIFOm(m=0 to 15).
    6) Set HSSPIn_TXE.TFEC="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.

8. Go to 8-1 when inserting Hi-Z and go to 9 when not inserting.

8-1. Hi-Z output to serial bus
    1) Set HSSPIn_FIFOCFG.TXCTRL="1".
    2) Write data to TXFIFOm(m=0 to 15) for Hi-Z output.

9. Disabling interrupt
    1) Set "0" to all bits of the HSSPIn_TXE register.
10. FIFO initialization, FIFO threshold and bit width setting
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", TXCTRL="0", FWIDTH="bit width", TXFTH="threshold value", RXFTH="threshold value".
11. Reception interrupt flag clear
    1) Set "1" to all bits of the HSSPIn_RXC register.
12. Reception DMA setting
    1) DSTC setting (Refer to 5.2.1 Setting during reception DMA transfer for details.)
    2) Set DBCNT.RXDBEN="1" for the DMA BRIDGE control register.
    3) Set HSSPIn_DMDMAEN.RXDMAEN="1".
13. Operation mode (RX-Only) and serial bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="01"(RX-Only) and TRP[1:0]="bit width".
14. Reception interrupt enable setting
    1) Set HSSPIn_RXE.RSSRE="1". Wait for interrupt by the HSSPIn_RXF.RSSRS or RSSRS bit.
15. Processing for HSSPIn_RXF.RSSRS="1"
    1) Set HSSPIn_RXC.RSSRC="1".
    2) Set HSSPIn_DMDMAEN.RXDMAEN="0".
    3) Set DBCNT.RXDBEN="0" for the DMA BRIDGE control register.
    4) Perform processing for DSTC. (mask clear by writing '1' to DQMSKCLR[n] of DSTC)
16. Completing reading from serial memory
    1) Set "0" to all bits of the HSSPIn_RXE register.

5.1.4 Data write to serial memory (direct mode: data transfer by interrupt)

2. Initializing TX-FIFO/RX-FIFO and setting FIFO bit width, transmission threshold and reception threshold
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", FWIDTH="00", TXFTH="threshold", RXFTH="threshold".
3. Transmission control settings (byte counter mode).
    1) Set "1" to the HSSPIn_DMCFG.SSDC bit.
    2) Set transfer byte count to the HSSPIn_DMBCC register.
4. Enable setting for used slave select
    2) Set slave select to the HSSPIn_DMPSEL.PSEL bit.
5. Operation mode (TX-Only) and bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width".
6. Serial operation start
    1) Set HSSPIn_DMSTART="1".
7. Go to 7-1 when serial memory command and serial bit width of address is same, and go to 7-2 when they are different.

7-1. Command and address write to serial memory
    1) Write command and address to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit. Jump to 8.

7-2. Command and address write to serial memory
    1) Write command to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.
    4) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width" (address).
    5) Write address to TX-FIFOm(m=0 to 15).
    6) Set HSSPIn_TXE.TFEC="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.

8. Operation mode (TX-Only) and serial bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="10"(TX-Only) and TRP[1:0]="bit width".
9. Writing to serial memory
    1) Write data to TX-FIFO
10. Transmission interrupt enable setting
    1) Set HSSPIn_TXE.TSSRE="1" and TFEE="1". Wait for interrupt by the HSSPIn_TXF.TSSRS or TFES bit.
11. Go to 11-1 when HSSPIn_TXF.TSSRS="0", and go to 12 when it is "1".

11-1. Processing for HSSPIn_TXF.TFES="1"
    1) Check if HSSPIn_DMBCS.BCS≠0. When HSSPIn_DMBCS.BCS=0, set HSSPIn_TXE.TFEE="0" and wait until HSSPIn_TXF.TSSRS="1" is set and then jump to 12.
    2) Write data to TX-FIFO. Jump to 10.

12. Processing for HSSPIn_TXF.TSSRS="1"
    1) Set HSSPIn_TXC.TSSRC="1".
13. Completing writing to serial memory
    1) Set "0" to all bits of the HSSPIn_TXE register.

5.1.5 Data write to serial memory (direct mode: DMA transfer)

2. Initializing TX-FIFO/RX-FIFO and setting FIFO bit width, transmission threshold and reception threshold
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", FWIDTH="00", TXFTH="threshold", RXFTH="threshold".
3. Transmission control settings (byte counter mode).
    1) Set "1" to the HSSPIn_DMCFG.SSDC bit.
    2) Set transfer byte count to the HSSPIn_DMBCC register.
4. Enable setting for used slave select
    2) Set slave select to the HSSPIn_DMPSEL.PSEL bit.
5. Operation mode (TX-Only) and bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width".
6. Serial operation start
    1) Set HSSPIn_DMSTART="1".
7. Go to 7-1 when serial memory command and serial bit width of address is same, and go to 7-2 when they are different.

7-1. Command and address write to serial memory
    1) Write command and address to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit. Jump to 8.

7-2. Command and address write to serial memory
    1) Write command to TX-FIFOm(m=0 to 15).
    2) Set "1" to all bits of the HSSPIn_TXC register (except reserved bit).
    3) Set HSSPIn_TXE.TFEE="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.
    4) Set HSSPIn_DMTRP.TRP[3:2]="10" and TRP[1:0]="bit width" (address).
    5) Write address to TX-FIFOm(m=0 to 15).
    6) Set HSSPIn_TXE.TFEC="1". Wait for interrupt by the HSSPIn_TXF.TFES bit.

8. FIFO initialization, FIFO threshold and bit width setting
    1) Set HSSPIn_FIFOCFG.TXFLSH="1", RXFLSH="1", TXCTRL="0", FWIDTH="bit width", TXFTH="threshold value", RXFTH="threshold value".
9. Operation mode (TX-Only) and serial bit width setting
    1) Set HSSPIn_DMTRP.TRP[3:2]="10"(TX-Only) and TRP[1:0]="bit width".
10. Transmission DMA setting
    1) Set DSTC. (Refer to 5.2.2 Setting during transmission DMA transfer for details.)
    2) Set DBCNT.TXDBEN="1" for the DMA BRIDGE control register.
    3) Set HSSPIn_DMDMAEN.TXDMAEN="1".
11. Transmission interrupt enable setting
    1) Set HSSPIn_TXE.TSSRE="1". Wait for interrupt by the HSSPIn_TXF.TSSRS bit.
12. Processing for HSSPIn_TXF.TSSRS="1"
    1) Set HSSPIn_TXC.TSSRC="1".
    2) Set HSSPIn_DMDMAEN.TXDMAEN="0".
    3) Set DBCNT.TXDBEN="0" for the DMA BRIDGE control register.
    4) Perform processing for DSTC. (mask clear by writing '1' to DQMSKCLR[n] of DSTC)
13. Completing writing to serial memory
    1) Set "0" to all bits of the HSSPIn_TXE register.

5.1.6 Data read for serial memory (command sequencer mode)

2. Setting for used slave select and memory size
    1) Set HSSPIn_CSCFG.MSEL, SSEL0EN to 3EN, SPICHG="0", and MBM, SRAM="0".
3. Extension address setting
    1) Set extension address to the HSSPIn_CSAEXT.AEXT bit.
4. ITIMER setting
    1) Set timer value to the HSSPIn_CSITIME.ITIME bit.
5. Command sequence setting
    1) Set read sequence of serial memory to the HSSPIn_RDCSDC0 to 7 registers.
    2) Set write sequence of serial memory to the HSSPIn_WRCSDC0 to 7 registers.
6. Access to system space (read access to serial memory)
    1) During read access when iHSEL_MEM="H", access serial memory in accordance of setting.

5.1.7 Data write to serial memory (command sequencer mode)

2. Setting for used slave select and memory size
    1) Set HSSPIn_CSCFG.MSEL, SSEL0EN to 3EN, SPICHG="0", and MBM, SRAM="1".
3. Extension address setting
    1) Set extension address to the HSSPIn_CSAEXT.AEXT bit.
4. ITIMER setting
    1) Set timer value to the HSSPIn_CSITIME.ITIME bit.
5. Command sequence setting
    1) Set read sequence of serial memory to the HSSPIn_RDCSDC0 to 7 registers.
    2) Set write sequence of serial memory to the HSSPIn_WRCSDC0 to 7 registers.
6. Access to system space (write access to serial memory)
    1) During write access when iHSEL_MEM="H", access serial memory in accordance of setting.

5.2 Setting during DMA transfer by DSTC

This section describes setting examples and precautions for the HS_SPI controller and DSTC during DMA transfer in direct mode.

5.2.1 Setting during reception DMA transfer

To use reception DMA transfer, the settings of the items shown below have to be matched each other for the HS_SPI controller and DSTC.

Parameter	HS_SPI controller setting	DSTC setting	Remarks
Read size from RX-FIFO	HSSPIn_FIFOCFG.FWIDTH[1:0]	DES0.TW[1:0]	If a setting does not match, a violation interrupt will be generated by the protection violation detection bit (HSSPIn_FAULTC.PVFC). FWIDTH[1:0] has the setting of 24-bit width (10) and 32-bit width (11), it will not be FWIDTH[1:0]=TM[1:0] necessarily.
Read count from RX-FIFO	Block size setting (HSSPIn_FIFOCFG.RXFTH+1)	DES1.IIN[15:0]	If a setting does not match, a violation interrupt will be generated by the DMA read block size violation detection bit (HSSPIn_FAULTF. DRCBSFS).
Transfer byte count	HSSPIn_DMBCC.BCC[15:0]	IIN[15:0]× ORM[15:0]× Access size (+ command count)	Set the following value to HSSPIn_DMBCC[15:0] to prevent unmatching count of the data to be transferred. IIN × ORM × access size (byte) In addition, if command write is required, set the value to which the number of commands required for command write is added to HSSPIn_DMBCC[15:0].

This section describes a setting example of DSTC when performing reception DMA transfer below.

By the settings below, DSTC reads 512 bytes data from RX-FIFO, and write the data to consecutive address area.

Descriptor	bit	Value	Remarks
DES0	DV[1:0]	01	-
	ST[1:0]	00	-
	MODE	0	Set MODE to "0".
	ORL[2:0]	000	-
	TW[1:0]	10	Always set HSSPIn_FIFOCFG.FWIDTH[1:0] to "11".
	SAC[2:0]	001	By the Inner Reload function, after 16 times of reading, DSTC specifies the address of HSSPIn_RXFIFO0 again.
	DAC[2:0]	000	The target of writing is consecutive address area.
	CHRS[5:0]	000000	Transfer completion is notified by the interrupt (HSSPIn_RXF.RSSRS) from the HS_SPI controller; therefore, HWINT interrupt from DSTC is not required to be used.
	DMSET	1	Always set DMSET to "1". After transfer completion, perform mask clear by writing "1" to DQMSKCLR[n].
	CHLK	0	-
	ACK[1:0]	01	Always set ACK[1:0] to "01".
DES1	IIN[15:0]	0x0010	Always set HSSPIn_FIFOCFG.RXFTH[3:0] to "1111".
	ORM[15:0]	0x0008	IIN=16 times × ORM=8 times × 4 bytes (1 word) = 512 bytes
DES2	SA[31:0]	0xD000_0090	Always specify the address of HSSPIn_RXFIFO0.
DES3	DA[31:0]	Arbitrary address	Specify an arbitrary address to store reception data.

5.2.2 Setting during transmission DMA transfer

To use transmission DMA transfer, the settings of the items shown below have to be matched each other for the HS_SPI controller and DSTC.

Parameter	HS_SPI controller setting	DSTC setting	Operations in case of violation
TX-FIFO write size	HSSPIn_FIFOCFG.FWIDTH[1:0]	DES0.TW[1:0]	If a setting does not match, a violation interrupt will be generated by the protection violation detection bit (HSSPIn_FAULTC.PVFC). FWIDTH[1:0] has the setting of 24-bit width (10) and 32-bit width (11), it will not be FWIDTH[1:0]=TM[1:0] necessarily.
TX-FIFO write count	Block size setting (16-HSSPIn_FIFOCFG.TXFTH)	DES1.IIN[15:0]	If a setting does not match, a violation interrupt will be generated by the DMA write block size violation detection bit (HSSPIn_FAULTF. DWCBSFS).
Transfer byte count	HSSPIn_DMBCC.BCC[15:0]	IIN[15:0]× ORM[15:0]× Access size (+ command count)	Set the following value to HSSPIn_DMBCC[15:0] to prevent unmatching count of the data to be transferred. IIN × ORM × access size (byte) In addition, if command write is required, set the value to which the number of commands required for command write is added to HSSPIn_DMBCC[15:0].

This section describes a setting example of DSTC when performing transmission DMA transfer below.

By the settings below, DSTC reads 512 bytes data in consecutive address area, and write the data to TX-FIFO.

Descriptor	bit	Value	Remarks
DES0	DV[1:0]	01	-
	ST[1:0]	00	-
	MODE	0	Set MODE to "0".
	ORL[2:0]	000	-
	TW[1:0]	10	Always set HSSPIn_FIFOCFG.FWIDTH[1:0] to "11".
	SAC[2:0]	000	The source of reading is consecutive address area.
	DAC[2:0]	001	By the Inner Reload function, after 16 times of writing, DSTC specifies the address of HSSPIn_TXFIFO0 again.
	CHRS[5:0]	000000	Transfer completion is notified by the interrupt (HSSPIn_TXF.TSSRS) from the HS_SPI controller; therefore, HWINT interrupt from DSTC is not required to be used.
	DMSET	1	Always set DMSET to "1". After transfer completion, perform mask clear by writing "1" to DQMSKCLR[n].
	CHLK	0	-
	ACK[1:0]	01	Always set ACK[1:0] to "01".
DES1	IIN[15:0]	0x0010	Always set HSSPIn_FIFOCFG.TXFTH[3:0] to "0000".
	ORM[15:0]	0x0008	IIN=16 times × ORM=8 times × 1 word (4 bytes) = 512 bytes
DES2	SA[31:0]	0xD000_0050	Always specify the address of HSSPIn_TXFIFO0.
DES3	DA[31:0]	Arbitrary address	Specify an arbitrary address to store reception data.

APPENDIXES

This chapter shows the register map, list of notes, limitations and product type list.

A. Register Map

B. List of Notes

CODE: 9BFAPPENDIXES-E03.0

A. Register Map

This chapter shows the register map.

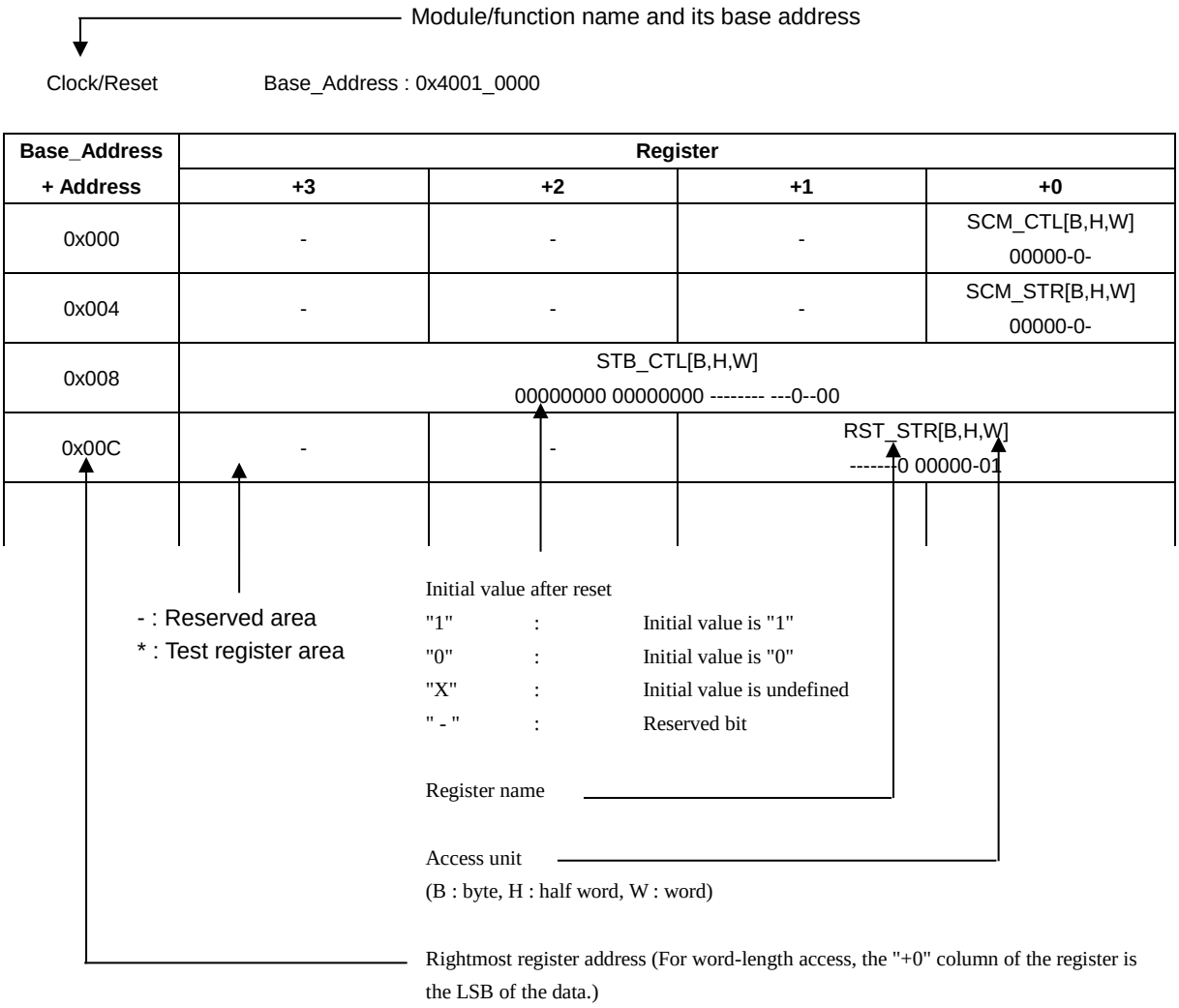
1. **Register Map**
   - 1.1. FLASH_IF
   - 1.2. Unique ID
   - 1.3. ECC Capture Address
   - 1.4. Clock/Reset
   - 1.5. HW WDT
   - 1.6. SW WDT
   - 1.7. Dual_Timer
   - 1.8. MFT
   - 1.9. PPG
   - 1.10. Base Timer
   - 1.11. IO Selector for Base Timer
   - 1.12. QPRC
   - 1.13. QPRC NF
   - 1.14. A/DC
   - 1.15. CR Trim
   - 1.16. EXTI
   - 1.17. INT-Req. READ
   - 1.18. D/AC
   - 1.19. HDMI-CEC
   - 1.20. GPIO
   - 1.21. LVD
   - 1.22. DS_Mode
   - 1.23. USB Clock
   - 1.24. CAN_Prescaler
   - 1.25. MFS
   - 1.26. CRC
   - 1.27. Watch Counter
   - 1.28. RTC
   - 1.29. Low-speed CR Prescaler
   - 1.30. Peripheral Clock Gating
   - 1.31. I2S Prescaler
   - 1.32. EXT-Bus I/F
   - 1.33. USB
   - 1.34. DMAC
   - 1.35. DSTC
   - 1.36. CAN
   - 1.37. Ethernet-MAC
   - 1.38. Ethernet-Control
   - 1.39. I2S
   - 1.40. SD-Card

1.41. CAN FD
1.42. Programmable-CRC
1.43. WorkFlash_IF
1.44. Hi-Speed Quad SPI controller

1. Register Map

Register map is shown on the table every module/function.

[How to read the each table]



Notes:

- *The register table is represented in the little-endian.*
- *When performing a data access, the addresses should be as below according to the access size.*
- *Word access : Address should be multiples of 4 (least significant 2 bits should be "0x00")*
- *Half word access : Address should be multiples of 2 (least significant bit should be "0x0")*
- *Byte access : -*
- *Do not access the test register area.*
- *Do not access the area that is not written in the register table.*

1.1 FLASH_IF

1.1.1 TYPE1-M4, TYPE2-M4 products

FLASH_IF Base_Address : 0x4000_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	FASZR[B,H,W]			
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C	*			
0x010	FSYNDN[B,H,W]			
0x014	FBFCR[B,H,W]			
0x018 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x1FC	-	-	-	-

Note:

- *For details of Flash I/F registers, see "FLASH PROGRAMMING MANUAL" of the product used.*

1.1.2 TYPE3-M4 product

FLASH_IF Base_Address : 0x4000_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	FASZR[B,H,W]			
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C	*			
0x010	FSYNDN[B,H,W]			
0x014	FBFCR[B,H,W]			
0x018 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C	-	-	-	-
0x030	DFCTRLR[W]			
0x034 - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x10C	-	-	-	-
0x110	FGPDM1[B,H,W]			
0x114	FGPDM2[B,H,W]			
0x118	FGPDM3[B,H,W]			
0x11C	FGPDM4[B,H,W]			
0x120 - 0x1FC	-	-	-	-

Note:

- *For details of Flash I/F registers, see "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 - 0xDFC	-	-	-	-

1.3 ECC Capture Address

ECC Capture Address Base_Address : 0x4000_0300

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	FERRAD[W]			
	-------- -XXXXXXX XXXXXXXX XXXXXXXX			
0x004 - 0xFFC	-	-	-	-

1.4 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-0-
0x004	-	-	-	SCM_STR[W]
				00000-0-
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 0000--01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	APBC2_PSR[W]
				1--0--00
0x020	-	-	-	SWC_PSR[W]
				------00
0x024 – 0x027	-	-	-	-
0x028	-	-	-	TTC_PSR[W]
				------00
0x02C – 0x02F	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x050	-	-	FCSWD_CTL[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----
0x058	-	-	-	*
0x05C - 0x05F	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0–000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C – 0xFFC	-	-	-	-

1.5 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[W]
				-------0
0x014	*			
0x018 – 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 – 0xFFC	-	-	-	-

1.6 SW WDT

SW WDT Base_Address : 0x4001_2000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	-	-	-	WdogRIS[W]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C – 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			
0xC04 - 0xDFC	-	-	-	-
0xF00 - 0xF04	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.7 Dual_Timer

Dual_Timer Base_Address : 0x4001_5000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.8 MFT

1.8.1 TYPE1-M4, TYPE2-M4 products

MFT unit0 Base_Address : 0x4002_0000

MFT unit1 Base_Address : 0x4002_1000

MFT unit2 Base_Address : 0x4002_2000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	OCCP0[H,W]		-	-
	00000000 00000000			
0x104	OCCP1[H,W]		-	-
	00000000 00000000			
0x108	OCCP2[H,W]		-	-
	00000000 00000000			
0x10C	OCCP3[H,W]		-	-
	00000000 00000000			
0x110	OCCP4[H,W]		-	-
	00000000 00000000			
0x114	OCCP5[H,W]		-	-
	00000000 00000000			
0x118	-	OCSD10[B,H,W]	OCSB10[B,H,W]	OCSA10[B,H,W]
		00000000	00000000	00000000
0x11C	-	OCSD32[B,H, W]	OCSB32[B,H, W]	OCSA32[B,H,W]
		00000000	00000000	00000000
0x120	-	OCSD54[B,H,W]	OCSB54[B,HW]	OCSA54[B,H,W]
		00000000	00000000	00000000
0x124	-	-	OCSC[B,H,W]	-
			--000000	
0x128	-	-	OCSE0[B,H,W]	
			00000000 00000000	
0x12C	OCSE1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[B,H,W]	
			00000000 00000000	
0x134	OCSE3[B,H,W]			
	00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[B,H,W]	
			00000000 00000000	
0x13C	OCSE5[B,H,W]			
	00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W]		-	-
	11111111 11111111			
0x144	TCDT0[H,W]		-	-
	00000000 00000000			
0x148	TCSC0[H,W]		TCSA0[B,H,W]	
	00000000 00000000		00000000 01000000	
0x14C	TCCP1[H,W]		-	-
	11111111 11111111			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x150	TCDT1[H,W]		-	-
	00000000 00000000			
0x154	TCSC1[H,W]		TCSA1[B,H,W]	
	00000000 00000000		00000000 01000000	
0x158	TCCP2[H,W]		-	-
	11111111 11111111			
0x15C	TCDT2[H,W]		-	-
	00000000 00000000			
0x160	TCSC2[H,W]		TCSA2[B,H,W]	
	00000000 00000000		00000000 01000000	
0x164	TCAL[W]			
	00000000 00000000 11111111 11111111 *1			
	-	-	-	- *2
	*1 MFT unit0			
	*2 MFT unit1,unit2			
0x168	-	OCFS54[B,H,W]	OCFS32[B,H,W]	OCFS10[B,H,W]
		00000000	00000000	00000000
0x16C	-	-	ICFS32[B,H,W]	ICFS10[B,H,W]
			00000000	00000000
0x170	-	ACFS54[B,H,W]	ACFS32[B,H,W]	ACFS10[B,H,W]
		00000000	00000000	00000000
0x174	ICCP0[H,W]		-	-
	00000000 00000000			
0x178	ICCP1[H,W]		-	-
	00000000 00000000			
0x17C	ICCP2[H,W]		-	-
	00000000 00000000			
0x180	ICCP3[H,W]		-	-
	00000000 00000000			
0x184	-	-	ICSB10[B,H,W]	ICSA10[B,H,W]
			------00	00000000
0x188			ICSB32[B,H,W]	ICSA32[B,H,W]
			------00	00000000
0x18C	WFTF10[H,W]		-	-
	00000000 00000000			
0x190	WFTB10[H,W]		WFTA10[H,W]	
	00000000 00000000		00000000 00000000	
0x194	WFTF32[H,W]		-	-
	00000000 00000000			
0x198	WFTB32[H,W]		WFTA32[H,W]	
	00000000 00000000		00000000 00000000	
0x19C	WFTF54[H,W]		-	-
	00000000 00000000			
0x1A0	WFTB54[H,W]		WFTA54[H,W]	
	00000000 00000000		00000000 00000000	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x1A4	-	-	WFSA10[B,H,W]	
			--000000 000000	
0x1A8	-	-	WFSA32[B,H,W]	
			--000000 000000	
0x1AC	-	-	WFSA54[B,H,W]	
			--000000 000000	
0x1B0	-	-	WFIR[H,W]	
			00000000 00000000	
0x1B4	-	-	NZCL[H,W]	
			00000000 00000000	
0x1B8	ACMP0[H,W]		-	-
	00000000 00000000			
0x1BC	ACMP1[H,W]		-	-
	00000000 00000000			
0x1C0	ACMP2[H,W]		-	-
	00000000 00000000			
0x1C4	ACMP3[H,W]		-	-
	00000000 00000000			
0x1C8	ACMP4[H,W]		-	-
	00000000 00000000			
0x1CC	ACMP5[H,W]		-	-
	00000000 00000000			
0x1D0	-	-	ACSA[B,H,W]	
			00000000 00000000	
0x1D4	-	-	ACSD0[B,H,W]	ACSC0[B,H,W]
			00000000	00000000
0x1D8	-	-	ACSD1[B,H,W]	ACSC1[B,H,W]
			00000000	00000000
0x1DC	-	-	ACSD2[B,H,W]	ACSC2[B,H,W]
			00000000	00000000
0x1E0	-	-	ACSD3[B,H,W]	ACSC3[B,H,W]
			00000000	00000000
0x1E4	-	-	ACSD4[B,H,W]	ACSC4[B,H,W]
			00000000	00000000
0x1E8	-	-	ACSD5[B,H,W]	ACSC5[B,H,W]
			00000000	00000000
0x1EC-0xFFC	-	-	-	-

1.8.2 TYPE3-M4 product

MFT unit0 **Base_Address : 0x4002_0000**

MFT unit1 **Base_Address : 0x4002_1000**

MFT unit2 **Base_Address : 0x4002_2000**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	OCCP0[H,W] 00000000 00000000		-	-
0x104	OCCP1[H,W] 00000000 00000000		-	-
0x108	OCCP2[H,W] 00000000 00000000		-	-
0x10C	OCCP3[H,W] 00000000 00000000		-	-
0x110	OCCP4[H,W] 00000000 00000000		-	-
0x114	OCCP5[H,W] 00000000 00000000		-	-
0x118	OCSD10[B,H,W] --000000 00000000		OCSB10[B,H,W] 00000000	OCSA10[B,H,W] 00000000
0x11C	OCSD32[B,H,W] --000000 00000000		OCSB32[B,H,W] 00000000	OCSA32[B,H,W] 00000000
0x120	OCSD54[B,H,W] --000000 00000000		OCSB54[B,H,W] 00000000	OCSA54[B,H,W] 00000000
0x124	-	-	OCSC[B,H,W] --000000	-
0x128	-	-	OCSE0[B,H,W] 00000000 00000000	
0x12C	OCSE1[B,H,W] 00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[B,H,W] 00000000 00000000	
0x134	OCSE3[B,H,W] 00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[B,H,W] 00000000 00000000	
0x13C	OCSE5[B,H,W] 00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W] 11111111 11111111		-	-
0x144	TCDT0[H,W] 00000000 00000000		-	-
0x148	TCSC0[H,W] 00000000 00000000		TCSA0[B,H,W] 00000000 01000000	
0x14C	TCCP1[H,W] 11111111 11111111		-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x150	TCDT1[H,W] 00000000 00000000		-	-
0x154	TCSC1[H,W] 00000000 00000000		TCSA1[B,H,W] 00000000 01000000	
0x158	TCCP2[H,W] 11111111 11111111		-	-
0x15C	TCDT2[H,W] 00000000 00000000		-	-
0x160	TCSC2[H,W] 00000000 00000000		TCSA2[B,H,W] 00000000 01000000	
0x164	TCAL[W] 00000000 00000000 11111111 11111111 *1			
	-	-	-	- *2
	*1 MFT unit0 *2 MFT unit1,unit2			
0x168	-	OCFS54[B,H,W] 00000000	OCFS32[B,H,W] 00000000	OCFS10[B,H,W] 00000000
0x16C	-	-	ICFS32[B,H,W] 00000000	ICFS10[B,H,W] 00000000
0x170	-	ACFS54[B,H,W] 00000000	ACFS32[B,H,W] 00000000	ACFS10[B,H,W] 00000000
0x174	ICCP0[H,W] 00000000 00000000		-	-
0x178	ICCP1[H,W] 00000000 00000000		-	-
0x17C	ICCP2[H,W] 00000000 00000000		-	-
0x180	ICCP3[H,W] 00000000 00000000		-	-
0x184	-	-	ICSB10[B,H,W] ------00	ICSA10[B,H,W] 00000000
0x188			ICSB32[B,H,W] ------00	ICSA32[B,H,W] 00000000
0x18C	WFTF10[H,W] 00000000 00000000		-	-
0x190	WFTB10[H,W] 00000000 00000000		WFTA10[H,W] 00000000 00000000	
0x194	WFTF32[H,W] 00000000 00000000		-	-
0x198	WFTB32[H,W] 00000000 00000000		WFTA32[H,W] 00000000 00000000	
0x19C	WFTF54[H,W] 00000000 00000000		-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x1A0	WFTB54[H,W] 00000000 00000000		WFTA54[H,W] 00000000 00000000	
0x1A4	-	-	WFSA10[B,H,W] --000000 000000	
0x1A8	-	-	WFSA32[B,H,W] --000000 000000	
0x1AC	-	-	WFSA54[B,H,W] --000000 000000	
0x1B0	-	-	WFIR[H,W] 00000000 00000000	
0x1B4	-	-	NZCL[H,W] 00000000 00000000	
0x1B8	ACMP0[H,W] 00000000 00000000		-	-
0x1BC	ACMP1[H,W] 00000000 00000000		-	-
0x1C0	ACMP2[H,W] 00000000 00000000		-	-
0x1C4	ACMP3[H,W] 00000000 00000000		-	-
0x1C8	ACMP4[H,W] 00000000 00000000		-	-
0x1CC	ACMP5[H,W] 00000000 00000000		-	-
0x1D0	-	-	ACSA[B,H,W] 00000000 00000000	
0x1D4	-	ACMC0[B,H,W] 00--0000	ACSD0[B,H,W] 00000000	ACSC0[B,H,W] 00000000
0x1D8	-	ACMC1[B,H,W] 00--0000	ACSD1[B,H,W] 00000000	ACSC1[B,H,W] 00000000
0x1DC	-	ACMC2[B,H,W] 00--0000	ACSD2[B,H,W] 00000000	ACSC2[B,H,W] 00000000
0x1E0	-	ACMC3[B,H,W] 00--0000	ACSD3[B,H,W] 00000000	ACSC3[B,H,W] 00000000
0x1E4	-	ACMC4[B,H,W] 00--0000	ACSD4[B,H,W] 00000000	ACSC4[B,H,W] 00000000
0x1E8	-	ACMC5[B,H,W] 00--0000	ACSD5[B,H,W] 00000000	ACSC5[B,H,W] 00000000
0x1EC	-	-	-	TCSD[B,H,W] ------00
0x1F0-0xFFC	-	-	-	-

1.9 PPG

PPG Base_Address : 0x4002_4000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	TTCR0 [B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0 [B,H,W] 00000000	-
0x00C	-	-	-	COMP2 [B,H,W] 00000000
0x010	-	-	COMP4 [B,H,W] 00000000	-
0x014	-	-	-	COMP6 [B,H,W] 00000000
0x018 - 0x01C	-	-	-	-
0x020	-	-	TTCR1 [B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1 [B,H,W] 00000000	-
0x02C	-	-	-	COMP3 [B,H,W] 00000000
0x030	-	-	COMP5 [B,H,W] 00000000	-
0x034	-	-	-	COMP7 [B,H,W] 00000000
0x038 - 0x03C	-	-	-	-
0x040	-	-	TTCR2 [B,H,W] 11110000	-
0x044	-	-	-	*
0x048	-	-	COMP8 [B,H,W] 00000000	-
0x04C	-	-	-	COMP10 [B,H,W] 00000000
0x050	-	-	COMP12 [B,H,W] 00000000	-
0x054	-	-	-	COMP14 [B,H,W] 00000000
0x058 - 0x0FC	-	-	-	-
0x100	-	-	TRG0 [B,H,W] 00000000 00000000	
0x104	-	-	REVC0 [B,H,W] 00000000 00000000	
0x108 - 0x13C	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x140	-	-	TRG1 [B,H,W]	
			-------- 00000000	
0x144	-	-	REVC1 [B,H,W]	
			-------- 00000000	
0x148 - 0x1FC	-	-	-	-
0x200	-	-	PPGC0 [B,H,W]	PPGC1 [B,H,W]
			00000000	00000000
0x204	-	-	PPGC2 [B,H,W]	PPGC3 [B,H,W]
			00000000	00000000
0x208	-	-	PRLH0 [B,H,W]	PRLL0 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x20C	-	-	PRLH1 [B,H,W]	PRLL1 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x210	-	-	PRLH2 [B,H,W]	PRLL2 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x214	-	-	PRLH3 [B,H,W]	PRLL3 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x218	-	-	-	GATEC0 [B,H,W]
				--00---00
0x21C - 0x23C	-	-	-	-
0x240	-	-	PPGC4 [B,H,W]	PPGC5 [B,H,W]
			00000000	00000000
0x244	-	-	PPGC6 [B,H,W]	PPGC7 [B,H,W]
			00000000	00000000
0x248	-	-	PRLH4 [B,H,W]	PRLL4 [B.H.W]
			XXXXXXXX	XXXXXXXX
0x24C	-	-	PRLH5 [B,H,W]	PRLL5 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x250	-	-	PRLH6 [B,H,W]	PRLL6 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x254	-	-	PRLH7 [B,H,W]	PRLL7 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x258	-	-	-	GATEC4 [B,H,W]
				------00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8 [B,H,W]	PPGC9 [B,H,W]
			00000000	00000000
0x284	-	-	PPGC10 [B,H,W]	PPGC11 [B,H,W]
			00000000	00000000
0x288	-	-	PRLH8 [B,H,W]	PRLL8 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x28C	-	-	PRLH9 [B,H,W]	PRLL9 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x290	-	-	PRLH10 [B,H,W]	PRLL10 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x294	-	-	PRLH11 [B,H,W]	PRLL11 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x298	-	-	-	GATEC8 [B,H,W]
				--00--00

0x29C - 0x2BC	-	-	-	-
0x2C0	-	-	PPGC12 [B,H,W]	PPGC13 [B,H,W]
			00000000	00000000
0x2C4	-	-	PPGC14 [B,H,W]	PPGC15 [B,H,W]
			00000000	00000000
0x2C8	-	-	PRLH12 [B,H,W]	PRLL12 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x2CC	-	-	PRLH13 [B,H,W]	PRLL13 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D0	-	-	PRLH14 [B,H,W]	PRLL14 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D4	-	-	PRLH15 [B,H,W]	PRLL15 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D8	-	-	-	GATEC12 [B,H,W]
				------00
0x2DC - 0x2FC	-	-	-	-
0x300	-	-	PPGC16 [B,H,W]	PPGC17 [B,H,W]
			00000000	00000000
0x304	-	-	PPGC18 [B,H,W]	PPGC19 [B,H,W]
			00000000	00000000
0x308	-	-	PRLH16 [B,H,W]	PRLL16 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x30C	-	-	PRLH17 [B,H,W]	PRLL17 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x310	-	-	PRLH18 [B,H,W]	PRLL18 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x314	-	-	PRLH19 [B,H,W]	PRLL19 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x318	-	-	-	GATEC16 [B,H,W]
				--00---00
0x31C - 0x33C	-	-	-	-
0x340	-	-	PPGC20 [B,H,W]	PPGC21 [B,H,W]
			00000000	00000000
0x344	-	-	PPGC22 [B,H,W]	PPGC23 [B,H,W]
			00000000	00000000
0x348	-	-	PRLH20 [B,H,W]	PRLL20 [B.H.W]
			XXXXXXXX	XXXXXXXX
0x34C	-	-	PRLH21 [B,H,W]	PRLL21 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x350	-	-	PRLH22 [B,H,W]	PRLL22 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x354	-	-	PRLH23 [B,H,W]	PRLL23 [B,H,W]
			XXXXXXXX	XXXXXXXX
0x358	-	-	-	GATEC20 [B,H,W]
				------00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	-
0x384 - 0xFFC	-	-	-	-

1.10 Base Timer

Base Timer ch.0 Base Address : 0x4002_5000

Base Timer ch.1 Base Address : 0x4002_5040

Base Timer ch.2 Base Address : 0x4002_5080

Base Timer ch.3 Base Address : 0x4002_50C0

Base Timer ch.4 Base Address : 0x4002_5200

Base Timer ch.5 Base Address : 0x4002_5240

Base Timer ch.6 Base Address : 0x4002_5280

Base Timer ch.7 Base Address : 0x4002_52C0

Base Timer ch.8 Base Address : 0x4002_5400

Base Timer ch.9 Base Address : 0x4002_5440

Base Timer ch.10 Base Address : 0x4002_5480

Base Timer ch.11 Base Address : 0x4002_54C0

Base Timer ch.12 Base Address : 0x4002_5600

Base Timer ch.13 Base Address : 0x4002_5640

Base Timer ch.14 Base Address : 0x4002_5680

Base Timer ch.15 Base Address : 0x4002_56C0

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL [H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF [H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR [H,W]	
			00000000 00000000	
0x00C	-	-	TMCR [B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2 [B,H,W]	STC [B,H,W]
			0------0	0000-000
0x014 - 0x03C	-	-	-	-

1.11 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) **Base Address : 0x4002_5100**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	BTSEL0123 [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.4-ch.7(Base Timer)Base Address : 0x4002_5300

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	BTSEL4567 [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.8-ch.11(Base Timer) **Base Address : 0x4002_5500**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	BTSEL89AB [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.12-ch.15(Base Timer) **Base Address : 0x4002_5700**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	BTSELCDEF [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simulation Startup(Base Timer) **Base Address : 0x4002_5F00**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR [B,H,W]	
			XXXXXXXX XXXXXXXX	

1.12 QPRC

1.12.1 TYPE1-M4, TYPE2-M4 products

QPRC ch.0 **Base Address : 0x4002_6000**

QPRC ch.1 **Base Address : 0x4002_6040**

QPRC ch.2 **Base Address : 0x4002_6080**

QPRC ch.3 **Base Address : 0x4002_60C0**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	QPCR [H,W]	
			00000000 00000000	
0x0004	-	-	QRCR [H,W]	
			00000000 00000000	
0x0008	-	-	QPCCR [H,W]	
			00000000 00000000	
0x000C	-	-	QPRCR [H,W]	
			00000000 00000000	
0x0010	-	-	QMPR [H,W]	
			11111111 11111111	
0x0014	-	-	QICRH [B,H,W]	QICRL [B,H,W]
			--000000	00000000
0x0018	-	-	QCRH [B,H,W]	QCRL [B,H,W]
			00000000	00000000
0x001C	-	-	QECR [B,H,W]	
			-------- -----000	
0x0020 - 0x003B	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12.2 TYPE3-M4 product

QPRC ch.0 **Base Address : 0x4002_6000**

QPRC ch.1 **Base Address : 0x4002_6040**

QPRC ch.2 **Base Address : 0x4002_6080**

QPRC ch.3 **Base Address : 0x4002_60C0**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	QPCR [H,W] 00000000 00000000	
0x0004	-	-	QRCR [H,W] 00000000 00000000	
0x0008	-	-	QPCCR [H,W] 00000000 00000000	
0x000C	-	-	QPRCR [H,W] 00000000 00000000	
0x0010	-	-	QMPR [H,W] 11111111 11111111	
0x0014	-	-	QICRH [B,H,W] --000000	QICRL [B,H,W] 00000000
0x0018	-	-	QCRH [B,H,W] 00000000	QCRL [B,H,W] 00000000
0x001C	-	-	QECR [B,H,W] -------- ----0000	
0x0020 - 0x003B	-	-	-	-
0x003C	QPCRR[B,H,W] 00000000 00000000		QRCRR[B,H,W] 00000000 00000000	

1.13 QPRC NF

QPRC ch.0 NF **Base Address : 0x4002_6100**

QPRC ch.1 NF **Base Address : 0x4002_6110**

QPRC ch.2 NF **Base Address : 0x4002_6120**

QPRC ch.3 NF **Base Address : 0x4002_6130**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	-	-	-	NFCTLA[B,H,W]
				--00-000
0x0004	-	-	-	NFCTLB[B,H,W]
				--00-000
0x0008	-	-	-	NFCTLZ[B,H,W]
				--00-000
0x000C	-	-	-	-

1.14 A/DC

12bit A/DC unit0 Base_Address : 0x4002_7000

12bit A/DC unit1 Base_Address : 0x4002_7100

12bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			10000000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	-	-	-	WCMRCOT[B,H,W]
				00000000
0x048	-	-	-	WCMRCIF[B,H,W]
				00000000
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			00000000	00100000
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00000000		00000000 00000000	
0x040 - 0x0FC	-	-	-	-

1.15 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W]
				-----001
0x004	-	-	MCR_FTRM[B,H,W]	
			------01 11101111	
0x008	-	-	-	MCR_TTRM[B,H,W]
				---10000
0x00C	MCR_RLR[W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[R/W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[R/W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	-	NMIRR[B,H,W]
				-------0
0x018	-	-	-	NMICL[B,H,W]
				-------1
0x01C	-	-	-	-
0x020 - 0x0FC	-	-	-	-

1.17 INT-Req. READ

1.17.1 TYPE1-M4, TYPE2-M4 products

INT-Req. READ Base_Address : 0x4003_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	DRQSEL[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004 – 0x00C	-			
0x010	-	-	-	ODDPKS[B]
				---00000
0x014	-	-	-	-
0x018	-	*	-	*
0x01C – 0x10C	-	-	-	-
0x110	IRQ003SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x114	IRQ004SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x118	IRQ005SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x11C	IRQ006SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x120	IRQ007SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x124	IRQ008SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x128	IRQ009SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x12C	IRQ010SEL[B,H,W]			
	-------- 00000000 -------- 00000000			
0x130 – 0x1FC	-	-	-	-
0x200	EXC02MON[B,H,W]			
	-------- -------- -------- ------00			
0x204	IRQ000MON[B,H,W]			
	-------- -------- -------- -------0			
0x208	IRQ001MON[B,H,W]			
	-------- -------- -------- -------0			
0x20C	IRQ002MON[B,H,W]			
	-------- -------- -------- -------0			
0x210	IRQ003MON[B,H,W]			
	-------- -------- -------- 00000000			
0x214	IRQ004MON[B,H,W]			
	-------- -------- -------- 00000000			
0x218	IRQ005MON[B,H,W]			
	-------- -------- -------- 00000000			

Base_Address + Address	Register +3	+2	+1	+0
0x21C	IRQ006MON[B,H,W]			
	-------- -------- -------- 00000000			
0x220	IRQ007MON[B,H,W]			
	-------- -------- -------- 00000000			
0x224	IRQ008MON[B,H,W]			
	-------- -------- -------- 00000000			
0x228	IRQ009MON[B,H,W]			
	-------- -------- -------- 00000000			
0x22C	IRQ010MON[B,H,W]			
	-------- -------- -------- 00000000			
0x230	IRQ011MON[B,H,W]			
	-------- -------- -------- -------0			
0x234	IRQ012MON[B,H,W]			
	-------- -------- -------- -------0			
0x238	IRQ013MON[B,H,W]			
	-------- -------- -------- -------0			
0x23C	IRQ014MON[B,H,W]			
	-------- -------- -------- -------0			
0x240	IRQ015MON[B,H,W]			
	-------- -------- -------- -------0			
0x244	IRQ016MON[B,H,W]			
	-------- -------- -------- -------0			
0x248	IRQ017MON[B,H,W]			
	-------- -------- -------- -------0			
0x24C	IRQ018MON[B,H,W]			
	-------- -------- -------- -------0			
0x250	IRQ019MON[B,H,W]			
	-------- -------- -------- --000000			
0x254	IRQ020MON[B,H,W]			
	-------- -------- -------- --000000			
0x258	IRQ021MON[B,H,W]			
	-------- -------- -------- ----0000			
0x25C	IRQ022MON[B,H,W]			
	-------- -------- -------- ----0000			
0x260	IRQ023MON[B,H,W]			
	-------- -------- -------- ----0000			
0x264	IRQ024MON[B,H,W]			
	-------- -------- -------- -----000			
0x268	IRQ025MON[B,H,W]			
	-------- -------- -------- -----000			
0x26C	IRQ026MON[B,H,W]			
	-------- -------- -------- ----0000			
0x270	IRQ027MON[B,H,W]			
	-------- -------- -------- --000000			
0x274	IRQ028MON[B,H,W]			
	-------- -------- -------- -----000			
0x278	IRQ029MON[B,H,W]			
	-------- -------- -------- -----000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x27C	IRQ030MON[B,H,W]			
	-------- -------- -------- ----0000			
0x280	IRQ031MON[B,H,W]			
	-------- -------- -------- --000000			
0x284	IRQ032MON[B,H,W]			
	-------- -------- -------- -----000			
0x288	IRQ033MON[B,H,W]			
	-------- -------- -------- -----000			
0x28C	IRQ034MON[B,H,W]			
	-------- -------- -------- ---00000			
0x290	IRQ035MON[B,H,W]			
	-------- -------- -------- --000000			
0x294	IRQ036MON[B,H,W]			
	-------- -------- -------- -----000			
0x298	IRQ037MON[B,H,W]			
	-------- -------- -------- -----000			
0x29C	IRQ038MON[B,H,W]			
	-------- -------- -------- -----000			
0x2A0	IRQ039MON[B,H,W]			
	-------- -------- -------- ------00			
0x2A4	IRQ040MON[B,H,W]			
	-------- -------- -------- ------00			
0x2A8	IRQ041MON[B,H,W]			
	-------- -------- -------- ------00			
0x2AC	IRQ042MON[B,H,W]			
	-------- -------- -------- ------00			
0x2B0	IRQ043MON[B,H,W]			
	-------- -------- -------- ------00			
0x2B4	IRQ044MON[B,H,W]			
	-------- -------- -------- ------00			
0x2B8	IRQ045MON[B,H,W]			
	-------- -------- -------- ------00			
0x2BC	IRQ046MON[B,H,W]			
	-------- -------- -------- ------00			
0x2C0	IRQ047MON[B,H,W]			
	-------- -------- -------- ------00			
0x2C4	IRQ048MON[B,H,W]			
	-------- -------- -------- -------0			
0x2C8	IRQ049MON[B,H,W]			
	-------- -------- -------- -------0			
0x2CC	IRQ050MON[B,H,W]			
	-------- -------- -------- -------0			
0x2D0	IRQ051MON[B,H,W]			
	-------- -------- -------- -------0			
0x2D4	IRQ052MON[B,H,W]			
	-------- -------- -------- -------0			
0x2D8	IRQ053MON[B,H,W]			
	-------- -------- -------- -------0			

Base_Address + Address	Register +3	+2	+1	+0
0x2DC	IRQ054MON[B,H,W]			
	-------- -------- -------- -------0			
0x2E0	IRQ055MON[B,H,W]			
	-------- -------- -------- -------0			
0x2E4	IRQ056MON[B,H,W]			
	-------- -------- -------- -------0			
0x2E8	IRQ057MON[B,H,W]			
	-------- -------- -------- -------0			
0x2EC	IRQ058MON[B,H,W]			
	-------- -------- -------- -------0			
0x2F0	IRQ059MON[B,H,W]			
	-------- -------- -------- ----0000			
0x2F4	IRQ060MON[B,H,W]			
	-------- -------- -------- -------0			
0x2F8	IRQ061MON[B,H,W]			
	-------- -------- -------- ------00			
0x2FC	IRQ062MON[B,H,W]			
	-------- -------- -------- -------0			
0x300	IRQ063MON[B,H,W]			
	-------- -------- -------- ------00			
0x304	IRQ064MON[B,H,W]			
	-------- -------- -------- -------0			
0x308	IRQ065MON[B,H,W]			
	-------- -------- -------- ------00			
0x30C	IRQ066MON[B,H,W]			
	-------- -------- -------- -------0			
0x310	IRQ067MON[B,H,W]			
	-------- -------- -------- ------00			
0x314	IRQ068MON[B,H,W]			
	-------- -------- -------- -------0			
0x318	IRQ069MON[B,H,W]			
	-------- -------- -------- ------00			
0x31C	IRQ070MON[B,H,W]			
	-------- -------- -------- -------0			
0x320	IRQ071MON[B,H,W]			
	-------- -------- -------- ------00			
0x324	IRQ072MON[B,H,W]			
	-------- -------- -------- -------0			
0x328	IRQ073MON[B,H,W]			
	-------- -------- -------- ------00			
0x32C	IRQ074MON[B,H,W]			
	-------- -------- -------- -------0			
0x330	IRQ075MON[B,H,W]			
	-------- -------- -------- ------00			
0x334	IRQ076MON[B,H,W]			
	-------- -------- -------- ---00000			
0x338	IRQ077MON[B,H,W]			
	-------- -------- -------- ---00000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x33C	IRQ078MON[B,H,W]			
	-------- -------- -------- ---00000			
0x340	IRQ079MON[B,H,W]			
	-------- -------- -------- --000000			
0x344	IRQ080MON[B,H,W]			
	-------- -------- -------- -------0			
0x348	IRQ081MON[B,H,W]			
	-------- -------- -------- -------0			
0x34C	IRQ082MON[B,H,W]			
	-------- -------- -------- -----000			
0x350	IRQ083MON[B,H,W]			
	-------- -------- -------- -------0			
0x354	IRQ084MON[B,H,W]			
	-------- -------- -------- -------0			
0x358	IRQ085MON[B,H,W]			
	-------- -------- -------- -------0			
0x35C	IRQ086MON[B,H,W]			
	-------- -------- -------- -------0			
0x360	IRQ087MON[B,H,W]			
	-------- -------- -------- -------0			
0x364	IRQ088MON[B,H,W]			
	-------- -------- -------- -------0			
0x368	IRQ089MON[B,H,W]			
	-------- -------- -------- -------0			
0x36C	IRQ090MON[B,H,W]			
	-------- -------- -------- -------0			
0x370	IRQ091MON[B,H,W]			
	-------- -------- -------- ------00			
0x374	IRQ092MON[B,H,W]			
	-------- -------- -------- ----0000			
0x378	IRQ093MON[B,H,W]			
	-------- -------- -------- ----0000			
0x37C	IRQ094MON[B,H,W]			
	-------- -------- -------- ----0000			
0x380	IRQ095MON[B,H,W]			
	-------- -------- -------- ----0000			
0x384	IRQ096MON[B,H,W]			
	-------- -------- -------- --000000			
0x388	IRQ097MON[B,H,W]			
	-------- -------- -------- --000000			
0x38C	IRQ098MON[B,H,W]			
	-------- -------- -------- ------00			
0x390	IRQ099MON[B,H,W]			
	-------- -------- -------- ------00			
0x394	IRQ100MON[B,H,W]			
	-------- -------- -------- ------00			
0x398	IRQ101MON[B,H,W]			
	-------- -------- -------- ------00			

Base_Address + Address	Register +3	+2	+1	+0
0x39C	IRQ102MON[B,H,W]			
	-------- -------- -------- ------00			
0x3A0	IRQ103MON[B,H,W]			
	-------- -------- -------- -------0			
0x3A4	IRQ104MON[B,H,W]			
	-------- -------- -------- ------00			
0x3A8	IRQ105MON[B,H,W]			
	-------- -------- -------- -------0			
0x3AC	IRQ106MON[B,H,W]			
	-------- -------- -------- ------00			
0x3B0	IRQ107MON[B,H,W]			
	-------- -------- -------- -------0			
0x3B4	IRQ108MON[B,H,W]			
	-------- -------- -------- ------00			
0x3B8	IRQ109MON[B,H,W]			
	-------- -------- -------- -------0			
0x3BC	IRQ110MON[B,H,W]			
	-------- -------- -------- ------00			
0x3C0	IRQ111MON[B,H,W]			
	-------- -------- -------- ---00000			
0x3C4	-	-	-	-
0x3C8	IRQ113MON[B,H,W]			
	-------- -------- -------- ---00000			
0x3CC	IRQ114MON[B,H,W]			
	-------- -------- -------- --000000			
0x3D0 – 0x3D8	-	-	-	-
0x3DC	IRQ118MON[B,H,W]			
	-------- -------- -------- ------00			
0x3E0	IRQ119MON[B,H,W]			
	-------- -------- -------- -------0			
0x3E4	IRQ120MON[B,H,W]			
	-------- -------- -------- -------0			
0x3E8	IRQ121MON[B,H,W]			
	-------- -------- -------- ------00			
0x3EC	IRQ122MON[B,H,W]			
	-------- -------- -------- -------0			
0x3F0	IRQ123MON[B,H,W]			
	-------- -------- -------- ------00			
0x3F4	IRQ124MON[B,H,W]			
	-------- -------- -------- -------0			
0x3F8	IRQ125MON[B,H,W]			
	-------- -------- -------- ------00			
0x3FC	IRQ126MON[B,H,W]			
	-------- -------- -------- -------0			
0x400	IRQ127MON[B,H,W]			
	-------- -------- -------- ------00			
0x404 – 0xFFC	-	-	-	-

1.17.2 TYPE3-M4 product

INT-Req. READ Base_Address : 0x4003_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	DRQSEL[B,H,W] 00000000 00000000 00000000 00000000			
0x004 – 0x00C	-			
0x010	-	-	-	ODDPKS[B] ---00000
0x014	-	-	-	ODDPKS1[B] --00000
0x018	-	*	-	*
0x01C – 0x10C	-	-	-	-
0x110	IRQ003SEL[B,H,W] -------- 00000000 -------- 00000000			
0x114	IRQ004SEL[B,H,W] -------- 00000000 -------- 00000000			
0x118	IRQ005SEL[B,H,W] -------- 00000000 -------- 00000000			
0x11C	IRQ006SEL[B,H,W] -------- 00000000 -------- 00000000			
0x120	IRQ007SEL[B,H,W] -------- 00000000 -------- 00000000			
0x124	IRQ008SEL[B,H,W] -------- 00000000 -------- 00000000			
0x128	IRQ009SEL[B,H,W] -------- 00000000 -------- 00000000			
0x12C	IRQ010SEL[B,H,W] -------- 00000000 -------- 00000000			
0x130 – 0x1FC	-	-	-	-
0x200	EXC02MON[B,H,W] -------- -------- -------- ------00			
0x204	IRQ000MON[B,H,W] -------- -------- -------- -------0			
0x208	IRQ001MON[B,H,W] -------- -------- -------- -------0			
0x20C	IRQ002MON[B,H,W] -------- -------- -------- -------0			
0x210	IRQ003MON[B,H,W] -------- -------- -------- 00000000			
0x214	IRQ004MON[B,H,W] -------- -------- -------- 00000000			
0x218	IRQ005MON[B,H,W] -------- -------- -------- 00000000			
0x21C	IRQ006MON[B,H,W] -------- -------- -------- 00000000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x220	IRQ007MON[B,H,W] -------- -------- -------- 00000000			
0x224	IRQ008MON[B,H,W] -------- -------- -------- 00000000			
0x228	IRQ009MON[B,H,W] -------- -------- -------- 00000000			
0x22C	IRQ010MON[B,H,W] -------- -------- -------- 00000000			
0x230	IRQ011MON[B,H,W] -------- -------- -------- -------0			
0x234	IRQ012MON[B,H,W] -------- -------- -------- -------0			
0x238	IRQ013MON[B,H,W] -------- -------- -------- -------0			
0x23C	IRQ014MON[B,H,W] -------- -------- -------- -------0			
0x240	IRQ015MON[B,H,W] -------- -------- -------- -------0			
0x244	IRQ016MON[B,H,W] -------- -------- -------- -------0			
0x248	IRQ017MON[B,H,W] -------- -------- -------- -------0			
0x24C	IRQ018MON[B,H,W] -------- -------- -------- -------0			
0x250	IRQ019MON[B,H,W] -------- -------- -------- --000000			
0x254	IRQ020MON[B,H,W] -------- -------- -------- --000000			
0x258	IRQ021MON[B,H,W] -------- -------- -------- ----0000			
0x25C	IRQ022MON[B,H,W] -------- -------- -------- ----0000			
0x260	IRQ023MON[B,H,W] -------- -------- -------- ----0000			
0x264	IRQ024MON[B,H,W] -------- -------- -------- -----000			
0x268	IRQ025MON[B,H,W] -------- -------- -------- -----000			
0x26C	IRQ026MON[B,H,W] -------- -------- -------- ----0000			
0x270	IRQ027MON[B,H,W] -------- -------- -------- --000000			
0x274	IRQ028MON[B,H,W] -------- -------- -------- -----000			
0x278	IRQ029MON[B,H,W] -------- -------- -------- -----000			
0x27C	IRQ030MON[B,H,W] -------- -------- -------- ----0000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x280	IRQ031MON[B,H,W] -------- -------- -------- --000000			
0x284	IRQ032MON[B,H,W] -------- -------- -------- -----000			
0x288	IRQ033MON[B,H,W] -------- -------- -------- -----000			
0x28C	IRQ034MON[B,H,W] -------- -------- -------- ---00000			
0x290	IRQ035MON[B,H,W] -------- -------- -------- --000000			
0x294	IRQ036MON[B,H,W] -------- -------- -------- -----000			
0x298	IRQ037MON[B,H,W] -------- -------- -------- -----000			
0x29C	IRQ038MON[B,H,W] -------- -------- -------- -----000			
0x2A0	IRQ039MON[B,H,W] -------- -------- -------- ------00			
0x2A4	IRQ040MON[B,H,W] -------- -------- -------- ------00			
0x2A8	IRQ041MON[B,H,W] -------- -------- -------- ------00			
0x2AC	IRQ042MON[B,H,W] -------- -------- -------- ------00			
0x2B0	IRQ043MON[B,H,W] -------- -------- -------- ------00			
0x2B4	IRQ044MON[B,H,W] -------- -------- -------- ------00			
0x2B8	IRQ045MON[B,H,W] -------- -------- -------- ------00			
0x2BC	IRQ046MON[B,H,W] -------- -------- -------- ------00			
0x2C0	IRQ047MON[B,H,W] -------- -------- -------- ------00			
0x2C4	IRQ048MON[B,H,W] -------- -------- -------- -------0			
0x2C8	IRQ049MON[B,H,W] -------- -------- -------- -------0			
0x2CC	IRQ050MON[B,H,W] -------- -------- -------- -------0			
0x2D0	IRQ051MON[B,H,W] -------- -------- -------- -------0			
0x2D4	IRQ052MON[B,H,W] -------- -------- -------- -------0			
0x2D8	IRQ053MON[B,H,W] -------- -------- -------- -------0			
0x2DC	IRQ054MON[B,H,W] -------- -------- -------- -------0			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2E0	IRQ055MON[B,H,W] -------- -------- -------- -------0			
0x2E4	IRQ056MON[B,H,W] -------- -------- -------- -------0			
0x2E8	IRQ057MON[B,H,W] -------- -------- -------- -------0			
0x2EC	IRQ058MON[B,H,W] -------- -------- -------- -------0			
0x2F0	IRQ059MON[B,H,W] -------- -------- -------- ---00000			
0x2F4	IRQ060MON[B,H,W] -------- -------- -------- -------0			
0x2F8	IRQ061MON[B,H,W] -------- -------- -------- ------00			
0x2FC	IRQ062MON[B,H,W] -------- -------- -------- -------0			
0x300	IRQ063MON[B,H,W] -------- -------- -------- ------00			
0x304	IRQ064MON[B,H,W] -------- -------- -------- -------0			
0x308	IRQ065MON[B,H,W] -------- -------- -------- ------00			
0x30C	IRQ066MON[B,H,W] -------- -------- -------- -------0			
0x310	IRQ067MON[B,H,W] -------- -------- -------- ------00			
0x314	IRQ068MON[B,H,W] -------- -------- -------- -------0			
0x318	IRQ069MON[B,H,W] -------- -------- -------- ------00			
0x31C	IRQ070MON[B,H,W] -------- -------- -------- -------0			
0x320	IRQ071MON[B,H,W] -------- -------- -------- ------00			
0x324	IRQ072MON[B,H,W] -------- -------- -------- -------0			
0x328	IRQ073MON[B,H,W] -------- -------- -------- ------00			
0x32C	IRQ074MON[B,H,W] -------- -------- -------- -------0			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x330	IRQ075MON[B,H,W] -------- -------- -------- ------00			
0x334	IRQ076MON[B,H,W] -------- -------- -------- ---00000			
0x338	IRQ077MON[B,H,W] -------- -------- -------- ---00000			
0x33C	IRQ078MON[B,H,W] -------- -------- -------- ---00000			
0x340	IRQ079MON[B,H,W] -------- -------- -------- --000000			
0x344	IRQ080MON[B,H,W] -------- -------- -------- -------0			
0x348	IRQ081MON[B,H,W] -------- -------- -------- ---00000			
0x34C	IRQ082MON[B,H,W] -------- -------- -------- -----000			
0x350	IRQ083MON[B,H,W] -------- -------- -------- -------0			
0x354	IRQ084MON[B,H,W] -------- -------- -------- -------0			
0x358	IRQ085MON[B,H,W] -------- -------- -------- -------0			
0x35C	IRQ086MON[B,H,W] -------- -------- -------- -------0			
0x360	IRQ087MON[B,H,W] -------- -------- -------- -------0			
0x364	IRQ088MON[B,H,W] -------- -------- -------- -------0			
0x368	IRQ089MON[B,H,W] -------- -------- -------- -------0			
0x36C	IRQ090MON[B,H,W] -------- -------- -------- -------0			
0x370	IRQ091MON[B,H,W] -------- -------- -------- ------00			
0x374	IRQ092MON[B,H,W] -------- -------- -------- ----0000			
0x378	IRQ093MON[B,H,W] -------- -------- -------- ----0000			
0x37C	IRQ094MON[B,H,W] -------- -------- -------- ----0000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x380	IRQ095MON[B,H,W] -------- -------- -------- ----0000			
0x384	IRQ096MON[B,H,W] -------- -------- -------- --000000			
0x388	IRQ097MON[B,H,W] -------- -------- -------- --000000			
0x38C	IRQ098MON[B,H,W] -------- -------- -------- ------00			
0x390	IRQ099MON[B,H,W] -------- -------- -------- ------00			
0x394	IRQ100MON[B,H,W] -------- -------- -------- ------00			
0x398	IRQ101MON[B,H,W] -------- -------- -------- ------00			
0x39C	IRQ102MON[B,H,W] -------- -------- -------- ------00			
0x3A0	IRQ103MON[B,H,W] -------- -------- -------- -------0			
0x3A4	IRQ104MON[B,H,W] -------- -------- -------- ------00			
0x3A8	IRQ105MON[B,H,W] -------- -------- -------- -------0			
0x3AC	IRQ106MON[B,H,W] -------- -------- -------- ------00			
0x3B0	IRQ107MON[B,H,W] -------- -------- -------- -------0			
0x3B4	IRQ108MON[B,H,W] -------- -------- -------- ------00			
0x3B8	IRQ109MON[B,H,W] -------- -------- -------- -------0			
0x3BC	IRQ110MON[B,H,W] -------- -------- -------- ------00			
0x3C0	IRQ111MON[B,H,W] -------- -------- -------- ---00000			
0x3C4	IRQ112MON[B,H,W] -------- -------- -------- --000000			
0x3C8	IRQ113MON[B,H,W] -------- -------- -------- --000000			
0x3CC	IRQ114MON[B,H,W] -------- -------- -------- -0000000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x3D0	IRQ115MON[B,H,W] -------- -------- -------- -----000			
0x3D4	IRQ116MON[B,H,W] -------- -------- -------- --------			
0x3D8	IRQ117MON[B,H,W] -------- -------- -------- ------00			
0x3DC	IRQ118MON[B,H,W] -------- -------- -------- ------00			
0x3E0	IRQ119MON[B,H,W] -------- -------- -------- -------0			
0x3E4	IRQ120MON[B,H,W] -------- -------- -------- -------0			
0x3E8	IRQ121MON[B,H,W] -------- -------- -------- ------00			
0x3EC	IRQ122MON[B,H,W] -------- -------- -------- -------0			
0x3F0	IRQ123MON[B,H,W] -------- -------- -------- ------00			
0x3F4	IRQ124MON[B,H,W] -------- -------- -------- -------0			
0x3F8	IRQ125MON[B,H,W] -------- -------- -------- ------00			
0x3FC	IRQ126MON[B,H,W] -------- -------- -------- -------0			
0x400	IRQ127MON[B,H,W] -------- -------- -------- ------00			
0x404 – 0xFFC	-	-	-	-

1.18 D/AC

12bit D/AC unit0 Base_Address : 0x4003_3000

12bit D/AC unit1 Base_Address : 0x4003_3008

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	DACR[B,H,W]
				--00--00
0x004	-	-	DADR[H,W]	
			----XXXX XXXXXXXX	
0x010 – 0xFFC	-	-	-	-

1.19 HDMI-CEC

HDMI-CEC/Remote Control Receiver ch.0 Base_Address : 0x4003_4000

HDMI-CEC/Remote Control Receiver ch.1 Base_Address : 0x4003_4100

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	TXCTRL[B,H,W] --0000-0
0x004	-	-	-	TXDATA[B,H,W] 00000000
0x008	-	-	-	TXSTS[B,H,W] --00---0
0x00C	-	-	-	SFREE[B,H,W] ----0000
0x010 – 0x03C	-	-	-	-
0x040	-	-	RCCR[B,H,W] 0---0000	RCST[B,H,W] 00000000
0x044	-	-	RCSHW[B,H,W] 00000000	RCDAHW[B,H,W] 00000000
0x048	-	-	RCDBHW[B,H,W] 00000000	-
0x04C	-	-	RCADR1[B,H,W] ---00000	RCADR2[B,H,W] ---00000
0x050	-	-	RCDTHH[B,H,W] 00000000	RCDTHL[B,H,W] 00000000
0x054	-	-	RCDTLH[B,H,W] 00000000	RCDTLL[B,H,W] 00000000
0x058	-	-	RCCKD[B,H,W] ---00000 00000000	
0x05C	-	-	RCRC[B,H,W] ---0---0	RCRHW[B,H,W] 00000000
0x060	-	-	RCLE[B,H,W] 00000-00	-
0x064	-	-	RCLELW[B,H,W] 00000000	RCLESW[B,H,W] 00000000
0x068 – 0x0FC	-	-	-	-

1.20 GPIO

1.20.1 TYPE1-M4, TYPE2-M4 products

GPIO Base_Address : 0x4006_F000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0001 1111			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x040 - 0x0FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0001 1111			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-			
0x124	PCR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x140 - 0x1FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x21C	DDR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224	DDR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x228	DDRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x22C	DDRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x230	DDRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x234	DDRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C	DDRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x240 - 0x2FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x32C	PDIRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x340 - 0x3FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x43C	PDORF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x440 - 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	1111 1111 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- --00 01--			
0x584 - 0x5FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x600	EPFR00[B,H,W]			
	---- --00 ---- --11 --0- --0- 0000 --00			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x608	EPFR02[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W]			
	--00 0000 --00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x628	EPFR10[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x62C	EPFR11[B,H,W]			
	---- --00 0000 0000 0000 0000 0000 0000			
0x630	EPFR12[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x634	EPFR13[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x638	EPFR14[B,H,W]			
	--00 0000 0000 00-- ---- ---- --00 0000			
0x63C	EPFR15[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	--00 0000 0000 0000 0000 0000 0000 0000			
0x644	EPFR17[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x648	EPFR18[B,H,W]			
	--00 0000 0000 0000 00-- --00 0000 ----			
0x64C	EPFR19[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- ----			
0x650	EPFR20[B,H,W]			
	---- ---0 0000 0000 0000 0000 0000 0000			
0x654 – 0x6FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x700	PZR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C	PZR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x720	PZR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x740 - 0xEFC	-	-	-	-
0xF00 – 0xF04	*			
0xF08 – 0xFDC	-	-	-	-
0xFE0	*			
0xFE4 - 0xFFC	-	-	-	-

1.20.2 TYPE3-M4 product

GPIO Base_Address : 0x4006_F000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	PFR0[B,H,W] ---- ---- ---- ---- 0000 0000 0001 1111			
0x004	PFR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x040 - 0x0FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	PCR0[B,H,W] ---- ---- ---- ---- 0000 0000 0001 1111			
0x104	PCR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x108	PCR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-			
0x124	PCR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x140 - 0x1FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x200	DDR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x21C	DDR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x220	DDR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x224	DDR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x228	DDRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x22C	DDRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x230	DDRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x234	DDRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x238	DDRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x23C	DDRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x240 - 0x2FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x300	PDIR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x32C	PDIRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x340 - 0x3FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x400	PDOR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x43C	PDORF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x440 - 0x4FC	-	-	-	-
0x500	ADE[B,H,W] 1111 1111 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W] ---- ---- ---- ---- ---- ---- --00 01--			
0x584 - 0x5FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x600	EPFR00[B,H,W] ---- 0000 ---- --11 --0- --0- 0000 –00			
0x604	EPFR01[B,H,W] 0000 0000 0000 0000 ---0 0000 0000 0000			
0x608	EPFR02[B,H,W] 0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W] 0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W] --00 0000 --00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W] 0000 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x628	EPFR10[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x62C	EPFR11[B,H,W] ---- --00 0000 0000 0000 0000 0000 0000			
0x630	EPFR12[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x634	EPFR13[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x638	EPFR14[B,H,W] --00 0000 0000 00-- ---- ---- --00 0000			
0x63C	EPFR15[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x640	EPFR16[B,H,W] --00 0000 0000 0000 0000 0000 0000 0000			
0x644	EPFR17[B,H,W] ---- 0000 0000 0000 0000 0000 0000 ----			
0x648	EPFR18[B,H,W] --00 0000 0000 0000 00-- --00 0000 0000			
0x64C	EPFR19[B,H,W] ---- ---- ---- ---- ---- ---- ---- ----			
0x650	EPFR20[B,H,W] ---- ---0 0000 0000 0000 0000 0000 0000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x654	EPFR21[B,H,W] ---- ---- ---- ---- ---- ---- ---- ----			
0x658	EPFR22[B,H,W] ---- ---- ---- ---- ---- ---- ---- ----			
0x65C	EPFR23[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x660	EPFR24[B,H,W] ---- ---- ---- ---- ---- 0000 0000 0000			
0x664	EPFR25[B,H,W] ---- ---- ---- ---- ---- ---- ---- 0000			
0x668	EPFR26[B,H,W] ---- ---- ---- --00 0000 0000 0000 0000			
0x66C – 0x6FC	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x700	PZR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x71C	PZR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x720	PZR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x740	PDSR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x744	PDSR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x748	PDSR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x74C	PDSR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x750	PDSR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x754	PDSR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x758	PDSR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x75C	PDSR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x760	PDSR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x764	PDSR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x768	PDSRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x76C	PDSRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x770	PDSRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x774	PDSRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x778	PDSRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x77C	PDSRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x780 - 0xEFC	-	-	-	-
0xF00 – 0xF04	*			
0xF08 – 0xFDC	-	-	-	-
0xFE0	*			
0xFE4 - 0xFFC	-	-	-	-

1.21 LVD

LVD Base_Address : 0x4003_5000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	LVD_CTL[B,H,W]
				000111--
0x004	-	-	-	LVD_STR[B,H,W]
				0-------
0x008	-	-	-	LVD_CLR[B,H,W]
				1-------
0x00C	LVD_RLR[W]			
	00000000 00000000 00000000 00000001			
0x010	-	-	-	LVD_STR2 [B,H,W]
				0------
0x014 - 0x0FC	-	-	-	-

1.22 DS_Mode

DS_Mode Base_Address : 0x4003_5100

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	*
0x004	-	-	-	RCK_CTL[B,H,W]
				------01
0x008 - 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL[B,H,W]
				-------0
0x704	-	-	-	WRFSR[B,H,W]
				------00
0x708	-	-	WIFSR[B,H,W]	
			------00 00000000	
0x70C	-	-	WIER[B,H,W]	
			------00 00000-00	
0x710	-	-	-	WILVR[B,H,W]
				---00000
0x714	-	-	-	DSRAMR[B,H,W]
				------00
0x718 - 0x7FC	-	-	-	-
0x800	BUR04[B,H,W]	BUR03[B,H,W]	BUR02[B,H,W]	BUR01[B,H,W]
	00000000	00000000	00000000	00000000
0x804	BUR08[B,H,W]	BUR07[B,H,W]	BUR06[B,H,W]	BUR05[B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012[B,H,W]	BUR11[B,H,W]	BUR10[B,H,W]	BUR09[B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16[B,H,W]	BUR15[B,H,W]	BUR14[B,H,W]	BUR13[B,H,W]
	00000000	00000000	00000000	00000000
0x810 - 0xEFC	-	-	-	-

1.23 USB Clock

USB Clock **Base_Address : 0x4003_6000**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	UCCR[B,H,W]
				-0000000
0x004	-	-	-	UPCR1[B,H,W]
				------00
0x008	-	-	-	UPCR2[B,H,W]
				-----000
0x00C	-	-	-	UPCR3[B,H,W]
				---00000
0x010	-	-	-	UPCR4[B,H,W]
				-0111011
0x014	-	-	-	UP_STR[B,H,W]
				-------0
0x018	-	-	-	UPINT_ENR[B,H,W]
				-------0
0x01C	-	-	-	UPINT_CLR[B,H,W]
				-------0
0x020	-	-	-	UPINT_STR[B,H,W]
				-------0
0x024	-	-	-	UPCR5[B,H,W]
				----0100
0x028	-	-	-	UPCR6[B,H,W]
				----0010
0x02C	-	-	-	UPCR7[B,H,W]
				-------0
0x030	-	-	-	USBEN0[B,H,W]
				-------0
0x034	-	-	-	USBEN1[B,H,W]
				-------0
0x038 - 0x0FC	-	-	-	-

1.24 CAN_Prescaler

CAN_Prescaler Base_Address : 0x4003_7000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	CANPRE[B,H,W]
				----1011
0x004 - 0xFFC	-	-	-	-

1.25 MFS

MFS ch.0 Base_Address : 0x4003_8000

MFS ch.1 Base_Address : 0x4003_8100

MFS ch.2 Base_Address : 0x4003_8200

MFS ch.3 Base_Address : 0x4003_8300

MFS ch.4 Base_Address : 0x4003_8400

MFS ch.5 Base_Address : 0x4003_8500

MFS ch.6 Base_Address : 0x4003_8600

MFS ch.7 Base_Address : 0x4003_8700

MFS ch.8 Base_Address : 0x4003_8800

MFS ch.9 Base_Address : 0x4003_8900

MFS ch.10Base_Address : 0x4003_8A00

MFS ch.11Base_Address : 0x4003_8B00

MFS ch.12Base_Address : 0x4003_8C00

MFS ch.13Base_Address : 0x4003_8D00

MFS ch.14Base_Address : 0x4003_8E00

MFS ch.15Base_Address : 0x4003_8F00

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	SCR / IBCR[B,H,W]	SMR[B,H,W]
			0--0000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR / IBSR[B,H,W]
			0-00011	0000000
0x008	-	-	RDR/TDR[H,W]	
			00000000 00000000	

Base_Address + Address	Register			
	+3	+3	+3	+3
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/	SCSTR0/
			EIBCR[B,H,W]	NFCR[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR1[B,H,W]	SACSR0[B,H,W]
			00000000	00000000
0x028	-	-	STMR1[B,H,W]	STMR0[B,H,W]
			00000000	00000000
0x02C	-	-	STMCR1[B,H,W]	STMCR0[B,H,W]
			00000000	00000000
0x030	-	-	SCSCR1[B,H,W]	SCSCR0[B,H,W]
			00000000	00100000
0x034	-	-	SCSFR1[B,H,W]	SCSFR0[B,H,W]
			10000000	10000000
0x038	-	-	-	SCSFR2[B,H,W]
				10000000
0x03C	-	-	TBYTE1[B,H,W] 00000000	TBYTE0[B,H,W] 00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x0144 - 0x1FC	-	-	-	-

1.26 CRC

CRC Base_Address : 0x4003_9000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.27 Watch Counter

Watch Counter Base_Address : 0x4003_A000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00--0000	--000000	--000000
0x004 - 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 -------0	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 - 0xFFC	-	-	-	-

1.28 RTC

RTC Base_Address : 0x4003_B000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	-	-	-	WTCR10[B,H,W]
				00000000
0x104	-	-	-	WTCR11[B,H,W]
				---00000
0x108	-	-	-	WTCR12[B,H,W]
				00000000
0x10C	-	-	-	WTCR13[B,H,W]
				00000000
0x110	-	-	-	WTCR20[B,H,W]
				--000000
0x114	-	-	-	WTCR21[B,H,W]
				-----000
0x118	-	-	-	*
0x11C	-	-	-	WTSR[B,H,W]
				-0000000
0x120	-	-	-	WTMIR[B,H,W]
				-0000000
0x124	-	-	-	WTHR[B,H,W]
				--000000
0x128	-	-	-	WTDR[B,H,W]
				--000000
0x12C	-	-	-	WTDW[B,H,W]
				-----000
0x130	-	-	-	WTMOR[B,H,W]
				---00000
0x134	-	-	-	WTYR[B,H,W]
				00000000
0x138	-	-	-	ALMIR[B,H,W]
				-0000000
0x13C	-	-	-	ALHR[B,H,W]
				--000000
0x140	-	-	-	ALDR[B,H,W]
				--000000
0x144	-	-	-	ALMOR[B,H,W]
				---00000
0x148	-	-	-	ALYR[B,H,W]
				00000000
0x14C	-	-	-	WTTR0[B,H,W]
				00000000
0x150	-	-	-	WTTR1[B,H,W]
				00000000
0x154	-	-	-	WTTR2[B,H,W]
				------00

Base_Address + Address	Register			
	+3	+2	+1	+0
0x158	-	-	-	WTCAL0[B,H,W]
				00000000
0x15C	-	-	-	WTCAL1[B,H,W]
				------00
0x160	-	-	-	WTCALEN[B,H,W]
				-------0
0x164	-	-	-	WTDIV[B,H,W]
				----0000
0x168	-	-	-	WTDIVEN[B,H,W]
				------00
0x16C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x170	-	-	-	WTCOSEL[B,H,W]
				-------0
0x174	-	-	-	VB_DIVCLK[B,H,W]
				00000111
0x178	-	-	-	WTOSCCNT[B,H,W]
				------01
0x17C	-	-	-	CCS[B,H,W]
				00001000
0x180	-	-	-	CCB[B,H,W]
				00010000
0x184	-	-	-	*
0x188	-	-	-	BOOST[B,H,W]
				------11
0x18C	-	-	-	EWKUP[B,H,W]
				-------0
0x190	-	-	-	VDET[B,H,W]
				00------
0x194	-	-	-	*
0x198	-	-	-	HIBRST[B,H,W]
				-------0
0x19C	-	-	-	VBPFR[B,H,W]
				--011100
0x1A0	-	-	-	VBPCR[B,H,W]
				----0000
0x1A4	-	-	-	VBDDR[B,H,W]
				----XXXX
0x1A8	-	-	-	VBDIR[B,H,W]
				----0000
0x1AC	-	-	-	VBDOR[B,H,W]
				----1111
0x0B0	-	-	-	VBPZR[B,H,W]
				------11
0x1B4-1FF	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x200	BREG03[B,H,W]	BREG02[B,H,W]	BREG01[B,H,W]	BREG00[B,H,W]
	00000000	00000000	00000000	00000000
0x204	BREG07[B,H,W]	BREG06[B,H,W]	BREG05[B,H,W]	BREG04[B,H,W]
	00000000	00000000	00000000	00000000
0x208	BREG0B[B,H,W]	BREG0A[B,H,W]	BREG09[B,H,W]	BREG08[B,H,W]
	00000000	00000000	00000000	00000000
0x20C	BREG0F[B,H,W]	BREG0E[B,H,W]	BREG0D[B,H,W]	BREG0C[B,H,W]
	00000000	00000000	00000000	00000000
0x210	BREG13[B,H,W]	BREG12[B,H,W]	BREG11[B,H,W]	BREG10[B,H,W]
	00000000	00000000	00000000	00000000
0x214	BREG17[B,H,W]	BREG16[B,H,W]	BREG15[B,H,W]	BREG14[B,H,W]
	00000000	00000000	00000000	00000000
0x218	BREG1B[B,H,W]	BREG1A[B,H,W]	BREG19[B,H,W]	BREG18[B,H,W]
	00000000	00000000	00000000	00000000
0x21C	BREG1F[B,H,W]	BREG1E[B,H,W]	BREG1D[B,H,W]	BREG1C[B,H,W]
	00000000	00000000	00000000	00000000
0x220	BREG23[B,H,W]	BREG22[B,H,W]	BREG21[B,H,W]	BREG20[B,H,W]
	00000000	00000000	00000000	00000000
0x224	BREG27[B,H,W]	BREG26[B,H,W]	BREG25[B,H,W]	BREG24[B,H,W]
	00000000	00000000	00000000	00000000
0x228	BREG2B[B,H,W]	BREG2A[B,H,W]	BREG29[B,H,W]	BREG28[B,H,W]
	00000000	00000000	00000000	00000000
0x22C	BREG2F[B,H,W]	BREG2E[B,H,W]	BREG2D[B,H,W]	BREG2C[B,H,W]
	00000000	00000000	00000000	00000000
0x230	BREG33[B,H,W]	BREG32[B,H,W]	BREG31[B,H,W]	BREG30[B,H,W]
	00000000	00000000	00000000	00000000
0x234	BREG37[B,H,W]	BREG36[B,H,W]	BREG35[B,H,W]	BREG34[B,H,W]
	00000000	00000000	00000000	00000000
0x238	BREG3B[B,H,W]	BREG3A[B,H,W]	BREG39[B,H,W]	BREG38[B,H,W]
	00000000	00000000	00000000	00000000
0x23C	BREG3F[B,H,W]	BREG3E[B,H,W]	BREG3D[B,H,W]	BREG3C[B,H,W]
	00000000	00000000	00000000	00000000
0x240	BREG43[B,H,W]	BREG42[B,H,W]	BREG41[B,H,W]	BREG40[B,H,W]
	00000000	00000000	00000000	00000000
0x244	BREG47[B,H,W]	BREG46[B,H,W]	BREG45[B,H,W]	BREG44[B,H,W]
	00000000	00000000	00000000	00000000
0x248	BREG4B[B,H,W]	BREG4A[B,H,W]	BREG49[B,H,W]	BREG48[B,H,W]
	00000000	00000000	00000000	00000000
0x24C	BREG4F[B,H,W]	BREG4E[B,H,W]	BREG4D[B,H,W]	BREG4C[B,H,W]
	00000000	00000000	00000000	00000000
0x250	BREG53[B,H,W]	BREG52[B,H,W]	BREG51[B,H,W]	BREG50[B,H,W]
	00000000	00000000	00000000	00000000
0x254	BREG57[B,H,W]	BREG56[B,H,W]	BREG55[B,H,W]	BREG54[B,H,W]
	00000000	00000000	00000000	00000000
0x258	BREG5B[B,H,W]	BREG5A[B,H,W]	BREG59[B,H,W]	BREG58[B,H,W]
	00000000	00000000	00000000	00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x25C	BREG5F[B,H,W]	BREG5E[B,H,W]	BREG5D[B,H,W]	BREG5C[B,H,W]
	00000000	00000000	00000000	00000000
0x260	BREG63[B,H,W]	BREG62[B,H,W]	BREG61[B,H,W]	BREG60[B,H,W]
	00000000	00000000	00000000	00000000
0x264	BREG67[B,H,W]	BREG66[B,H,W]	BREG65[B,H,W]	BREG64[B,H,W]
	00000000	00000000	00000000	00000000
0x268	BREG6B[B,H,W]	BREG6A[B,H,W]	BREG69[B,H,W]	BREG68[B,H,W]
	00000000	00000000	00000000	00000000
0x26C	BREG6F[B,H,W]	BREG6E[B,H,W]	BREG6D[B,H,W]	BREG6C[B,H,W]
	00000000	00000000	00000000	00000000
0x270	BREG73[B,H,W]	BREG72[B,H,W]	BREG71[B,H,W]	BREG70[B,H,W]
	00000000	00000000	00000000	00000000
0x274	BREG77[B,H,W]	BREG76[B,H,W]	BREG75[B,H,W]	BREG74[B,H,W]
	00000000	00000000	00000000	00000000
0x278	BREG7B[B,H,W]	BREG7A[B,H,W]	BREG79[B,H,W]	BREG78[B,H,W]
	00000000	00000000	00000000	00000000
0x27C	BREG7F[B,H,W]	BREG7E[B,H,W]	BREG7D[B,H,W]	BREG7C[B,H,W]
	00000000	00000000	00000000	00000000
0x280-0xFFC	-	-	-	-

1.29 Low-speed CR Prescaler

Low-speed CR Prescaler Base_Address : 0x4003_C000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W],
				--000000
0x004 – 0x0FC	-	-	-	-

1.30 Peripheral Clock Gating

1.30.1 TYPE-M1, TYPE2-M4 products

Peripheral Clock Gating **Base_Address : 0x4003_C100**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1-1-1 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	-----0-0 ----0000 00000000 00000000			
0x008 – 0x00F	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000 ----0000 ----0000			
0x018 – 0x01F	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -------- -------0 --**--00 Products with CAN : *="1" Products without CAN : *="0"			
0x024	MRST2[B,H,W]			
	-------- -------- -------0 --00--00			
0x028 – 0x67C	-	-	-	-

1.30.2 TYPE3-M4 product

Peripheral Clock Gating **Base_Address : 0x4003_C100**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	CKEN0[B,H,W] ---1-1-1 ----1111 11111111 11111111			
0x004	MRST0[B,H,W] -----0-0 ----0000 00000000 00000000			
0x008 – 0x00F	-	-	-	-
0x010	CKEN1[B,H,W] -------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W] -------- ----0000 ----0000 ----0000			
0x018 – 0x01F	-	-	-	-
0x020	CKEN2[B,H,W] ---0--11 ---1--00 -------0 -***--00 Products with : *="1" Products without CAN : *="0"			
0x024	MRST2[B,H,W] ---0--00 ---0--00 -------0 -000--00			
0x028 – 0x67C	-	-	-	-

1.31 I2S Prescaler

I2S_Prescaler **Base_Address : 0x4003_D000**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	ICCR[B,H,W] -------- -------- -------- ------00			
0x004	IPCR1[B,H,W] -------- -------- -------- -------0			
0x008	IPCR2[B,H,W] -------- -------- -------- -----000			
0x00C	IPCR3[B,H,W] -------- -------- -------- ---00001			
0x010	IPCR4[B,H,W] -------- -------- -------- -0011111			
0x014	IP_STR[B,H,W] -------- -------- -------- -------0			
0x018	IPINT_ENR[B,H,W] -------- -------- -------- -------0			
0x01C	IPINT_CLR[B,H,W] -------- -------- -------- -------0			
0x020	IPINT_STR[B,H,W] -------- -------- -------- -------0			
0x024	IPCR5[B,H,W] -------- -------- -------- -0011000			
0x028 – 0xFFC	-	-	-	-

1.32 EXT-Bus I/F

1.32.1 TYPE1-M4 product

EXT-Bus I/F Base_Address : 0x4003_F000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	MODE0[W]			
	-------- -------- --000-00 00000000			
0x0004	MODE1[W]			
	-------- -------- --000-00 00000000			
0x0008	MODE2[W]			
	-------- -------- --000-00 00000000			
0x000C	MODE3[W]			
	-------- -------- --000-00 00000000			
0x0010	MODE4[W]			
	-------- -------- --000-00 00000001			
0x0014	MODE5[W]			
	-------- -------- --000-00 00000000			
0x0018	MODE6[W]			
	-------- -------- --000-00 00000000			
0x001C	MODE7[W]			
	-------- -------- --000-00 00000000			
0x0020	TIM0[W]			
	00000101 01011111 11110000 00001111			
0x0024	TIM1[W]			
	00000101 01011111 11110000 00001111			
0x0028	TIM2[W]			
	00000101 01011111 11110000 00001111			
0x002C	TIM3[W]			
	00000101 01011111 11110000 00001111			
0x0030	TIM4[W]			
	00000101 01011111 11110000 00001111			
0x0034	TIM5[W]			
	00000101 01011111 11110000 00001111			
0x0038	TIM6[W]			
	00000101 01011111 11110000 00001111			
0x003C	TIM7[W]			
	00000101 01011111 11110000 00001111			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0040	AREA0[W]			
	-------- -0001111 -------- 00000000			
0x0044	AREA1[W]			
	-------- -0001111 -------- 00010000			
0x0048	AREA2[W]			
	-------- -0001111 -------- 00100000			
0x004C	AREA3[W]			
	-------- -0001111 -------- 00110000			
0x0050	AREA4[W]			
	-------- -0001111 -------- 01000000			
0x0054	AREA5[W]			
	-------- -0001111 -------- 01010000			
0x0058	AREA6[W]			
	-------- -0001111 -------- 01100000			
0x005C	AREA7[W]			
	-------- -0001111 -------- 01110000			
0x0060	ATIM0[W]			
	-------- -------- ----0100 01011111			
0x0064	ATIM1[W]			
	-------- -------- ----0100 01011111			
0x0068	ATIM2[W]			
	-------- -------- ----0100 01011111			
0x006C	ATIM3[W]			
	-------- -------- ----0100 01011111			
0x0070	ATIM4[W]			
	-------- -------- ----0100 01011111			
0x0074	ATIM5[W]			
	-------- -------- ----0100 01011111			
0x0078	ATIM6[W]			
	-------- -------- ----0100 01011111			
0x007C	ATIM7[W]			
	-------- -------- ----0100 01011111			
0x0080 - 0x00FC	-	-	-	-
0x0100	SDMODE			
	-------- -------0 00010011 --00-000			
0x0104	REFTIM			
	-------0 00000000 0000000000110011			
0x0108	PWRDWN			
	-------- -------- 00000000 00000000			
0x010C	SDTIM			
	------00 01000010 00010001 0100--01			
0x0110	SDCMD			
	0------- ---00000 00000000 00000000			
0x0114 - 0x01FC	-	-	-	-

Base_Address + Address	Register			
	+3	+ 2	+1	+ 0
0x0200	MEMCERR			
	-------- -------- -------- ----0000			
0x0204 – 0x02FC	-	-	-	-
0x0300	DCLKR[W]			
	-------- -------- -------- ---01111			
0x0304	EST			
	-------- -------- -------- -------0			
0x0308	WEAD			
	00000000 00000000 000000000 00000000			
0x030C	ESCLR			
	-------- -------- -------- -------1			
0x0310	AMODE			
	-------- -------- -------- -------1			
0x031C - 0x0EFC	-	-	-	-
0x0F00 – 0x0F14	*	*	*	*
0x0F18 – 0x0FFC	-	-	-	-

1.32.2 TYPE3-M4 product

EXT-Bus I/F Base_Address : 0x4003_F000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	MODE0[W] -------- -------- --000-00 00000000			
0x0004	MODE1[W] -------- -------- --000-00 00000000			
0x0008	MODE2[W] -------- -------- --000-00 00000000			
0x000C	MODE3[W] -------- -------- --000-00 00000000			
0x0010	MODE4[W] -------- -------- --000-00 00000001			
0x0014	MODE5[W] -------- -------- --000-00 00000000			
0x0018	MODE6[W] -------- -------- --000-00 00000000			
0x001C	MODE7[W] -------- -------- --000-00 00000000			
0x0020	TIM0[W] 00000101 01011111 11110000 00001111			
0x0024	TIM1[W] 00000101 01011111 11110000 00001111			
0x0028	TIM2[W] 00000101 01011111 11110000 00001111			
0x002C	TIM3[W] 00000101 01011111 11110000 00001111			
0x0030	TIM4[W] 00000101 01011111 11110000 00001111			
0x0034	TIM5[W] 00000101 01011111 11110000 00001111			
0x0038	TIM6[W] 00000101 01011111 11110000 00001111			
0x003C	TIM7[W] 00000101 01011111 11110000 00001111			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0040	AREA0[W] -------- -0001111 -------- 00000000			
0x0044	AREA1[W] -------- -0001111 -------- 00010000			
0x0048	AREA2[W] -------- -0001111 -------- 00100000			
0x004C	AREA3[W] -------- -0001111 -------- 00110000			
0x0050	AREA4[W] -------- -0001111 -------- 01000000			
0x0054	AREA5[W] -------- -0001111 -------- 01010000			
0x0058	AREA6[W] -------- -0001111 -------- 01100000			
0x005C	AREA7[W] -------- -0001111 -------- 01110000			
0x0060	ATIM0[W] -------- -------- ----0100 01011111			
0x0064	ATIM1[W] -------- -------- ----0100 01011111			
0x0068	ATIM2[W] -------- -------- ----0100 01011111			
0x006C	ATIM3[W] -------- -------- ----0100 01011111			
0x0070	ATIM4[W] -------- -------- ----0100 01011111			
0x0074	ATIM5[W] -------- -------- ----0100 01011111			
0x0078	ATIM6[W] -------- -------- ----0100 01011111			
0x007C	ATIM7[W] -------- -------- ----0100 01011111			
0x0080 - 0x00FC	-	-	-	-
0x0100	SDMODE -------- -------0 00010011 --00-000			
0x0104	REFTIM -------0 00000000 0000000000110011			
0x0108	PWRDWN -------- -------- 00000000 00000000			
0x010C	SDTIM 0-----00 01000010 00010001 0100--01			
0x0110	SDCMD 0------- ---00000 00000000 00000000			
0x0114 - 0x01FC	-	-	-	-

Base_Address + Address	Register			
	+3	**+ 2**	**+1**	**+ 0**
0x0200	MEMCERR -------- -------- -------- ----0000			
0x0204 – 0x02FC	-	-	-	-
0x0300	DCLKR[W] -------- -------- -------- ---01111			
0x0304	EST -------- -------- -------- -------0			
0x0308	WEAD 00000000 00000000 000000000 00000000			
0x030C	ESCLR -------- -------- -------- -------1			
0x0310	AMODE -------- -------- -------- -------1			
0x031C - 0x0EFC	-	-	-	-
0x0F00 – 0x0F14	*	*	*	*
0x0F18 – 0x0FFC	-	-	-	-

1.33 USB

USB ch.0 Base_Address : 0x4004_0000

USB ch.1 Base_Address : 0x4005_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W]	HCNT0[B,H,W]
			-----001	00000000
0x2104	-	-	HERR[B,H,W]	HIRQ[B,H,W]
			00000011	0-000000
0x2108	-	-	HFCOMP[B,H,W]	HSTATE[B,H,W]
			00000000	--010010
0x210C	-	-	HRTIMER(1/0)[B,H,W]	
			00000000 00000000	
0x2110	-	-	HADR[B,H,W]	HRTIMER(2)[B,H,W]
			-0000000	------00
0x2114	-	-	HEOF(1/0)[B,H,W]	
			--000000 00000000	
0x2118	-	-	HFRAME(1/0)[B,H,W]	
			-----000 00000000	
0x211C	-	-	-	HTOKEN[B,H,W]
				00000000
0x2120	-	-	UDCC[B,H,W]	
			-------- 10100-00	
0x2124	-	-	EP0C[H,W]	
			------0- -1000000	
0x2128	-	-	EP1C[H,W]	
			01100001 00000000	
0x212C	-	-	EP2C[H,W]	
			0110000- -1000000	
0x2130	-	-	EP3C[H,W]	
			0110000- -1000000	
0x2134	-	-	EP4C[H,W]	
			0110000- -1000000	
0x2138	-	-	EP5C[H,W]	
			0110000- -1000000	
0x213C	-	-	TMSP[H,W]	
			-----000 00000000	
0x2140	-	-	UDCIE[B,H,W]	UDCS[B,H,W]
			--000000	--000000
0x2144	-	-	EP0IS[H,W]	
			10---1-- --------	
0x2148	-	-	EP0OS[H,W]	
			100--00- -XXXXXXX	
0x214C	-	-	EP1S[H,W]	
			100-000X XXXXXXXX	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2150	-	-	EP2S[H,W]	
			100-000- -XXXXXXX	
0x2154	-	-	EP3S[H,W]	
			100-000- -XXXXXXX	
0x2158	-	-	EP4S[H,W]	
			100-000- -XXXXXXX	
0x215C	-	-	EP5S[H,W]	
			100-000- -XXXXXXX	
0x2160	-	-	EP0DTH[B,H,W]	EP0DTL[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W]	EP1DTL[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W]	EP2DTL[B,H,W]
			XXXXXXXX	XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W]	EP3DTL[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W]	EP4DTL[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W]	EP5DTL[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2178 - 0x217C	-	-	-	-

1.34 DMAC

DMAC Base_Address : 0x4006_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	DMACR[B,H,W]			
	00-00000 -------- -------- --------			
0x0010	DMACA0[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0014	DMACB0[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0018	DMACSA0[B,H,W]			
	00000000 00000000 00000000 00000000			
0x001C	DMACDA0[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0020	DMACA1[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0024	DMACB1[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0028	DMACSA1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x002C	DMACDA1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0030	DMACA2[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0034	DMACB2[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0038	DMACSA2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x003C	DMACDA2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0040	DMACA3[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0044	DMACB3[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0048	DMACSA3[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004C	DMACDA3[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0050	DMACA4[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0054	DMACB4[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0058	DMACSA4[B,H,W]			
	00000000 00000000 00000000 00000000			
0x005C	DMACDA4[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0060	DMACA5[B,H,W]			
	00000000 0---0000 00000000 00000000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0064	DMACB5[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0068	DMACSA5[B,H,W]			
	00000000 00000000 00000000 00000000			
0x006C	DMACDA5[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0070	DMACA6[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0074	DMACB6[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0078	DMACSA6[B,H,W]			
	00000000 00000000 00000000 00000000			
0x007C	DMACDA6[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0080	DMACA7[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0084	DMACB7[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0088	DMACSA7[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008C	DMACDA7[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0090 - 0x00FC	-	-	-	-

1.35 DSTC

DSTC Base_Address : 0x4006_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	DESTP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0004	HWDESP[B,H,W]			
	00XXXXXX XXXXXX00 00000000 00000000			
0x0008	SWTR[H]		CFG[B]	CMD[B]
	00000000 00000000		01000000	00000001
0x000C	MONERS[B,H,W]			
	00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x0010	DREQENB[31:0] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0014	DREQENB[63:32] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0018	DREQENB[95:64] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x001C	DREQENB[127:96] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0020	DREQENB[159:128] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0024	DREQENB[191:160] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0028	DREQENB[223:192] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x002C	DREQENB[255:224] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0030	HWINT[31:0] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0034	HWINT[63:32] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0038	HWINT[95:64] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x003C	HWINT[127:96] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0040	HWINT[159:128] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0044	HWINT[191:160] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0048	HWINT[223:192] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x004C	HWINT[255:224] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0050	HWINTCLR[31:0] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0054	HWINTCLR[63:32] [B,H,W]			
	00000000 00000000 00000000 00000000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0058	HWINTCLR[95:64] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x005C	HWINTCLR[127:96] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x060	HWINTCLR[159:128] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x064	HWINTCLR[191:160] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x068	HWINTCLR[223:192] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x06C	HWINTCLR[255:224] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x070	DQMSK[31:0] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x078	DQMSK[95:64] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x07C	DQMSK[127:96] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x080	DQMSK[159:128] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x084	DQMSK[191:160] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x088	DQMSK[223:192] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x08C	DQMSK[255:224] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x090	DQMSKCLR[31:0] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x098	DQMSKCLR[95:64] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x09C	DQMSKCLR[127:96] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0A0	DQMSKCLR[159:128] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0A4	DQMSKCLR[191:160] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0A8	DQMSKCLR[223:192] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x0AC	DQMSKCLR[255:224] [B,H,W]			
	00000000 00000000 00000000 00000000			
0x00B0 - 0x0FFC	-	-	-	-

1.36 CAN

CAN ch.0 Base_Address : 0x4006_2000

CAN ch.1 Base_Address : 0x4006_3000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000	STATR[B,H,W]		CTRLR[B,H,W]	
	-------- 00000000		-------- 000-0001	
0x0004	BTR[B,H,W]		ERRCNT[B,H,W]	
	-0100011 00000001		00000000 00000000	
0x0008	TESTR[B,H,W]		INTR[B,H,W]	
	-------- X00000--		00000000 00000000	
0x000C	-	-	BRPER[B,H,W]	
			-------- ----0000	
0x0010	IF1CMSK[B,H,W]		IF1CREQ[B,H,W]	
	-------- 00000000		0------- 00000001	
0x0014	IF1MSK2[B,H,W]		IF1MSK1[B,H,W]	
	11-11111 11111111		11111111 11111111	
0x0018	IF1ARB2[B,H,W]		IF1ARB1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x001C	-	-	IF1MCTR[B,H,W]	
			00000000 0---0000	
0x0020	IF1DTA2[B,H,W]		IF1DTA1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0024	IF1DTB2[B,H,W]		IF1DTB1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0028 - 0x002F	-	-	-	-
0x0030	IF1DTA1[B,H,W]		IF1DTA2[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0034	IF1DTB1[B,H,W]		IF1DTB2[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0038 - 0x003C	-	-	-	-
0x0040	IF2CMSK[B,H,W]		IF2CREQ[B,H,W]	
	-------- 00000000		0------- 00000001	
0x0044	IF2MSK2[B,H,W]		IF2MSK1[B,H,W]	
	11-11111 11111111		11111111 11111111	
0x0048	IF2ARB2[B,H,W]		IF2ARB1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x004C	-	-	IF2MCTR[B,H,W]	
			00000000 0---0000	
0x0050	IF2DTA2[B,H,W]		IF2DTA1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0054	IF2DTB2[B,H,W]		IF2DTB1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0058 - 0x005C	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0060	IF2DTA1[B,H,W]		IF2DTA2[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0064	IF2DTB1[B,H,W]		IF2DTB2[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0068 - 0x007C	-	-	-	-
0x0080	TREQR2[B,H,W]		TREQR1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0084 - 0x008F	-	-	-	-
0x0090	NEWDT2[B,H,W]		NEWDT1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x0094 - 0x009F	-	-	-	-
0x00A0	INTPND2[B,H,W]		INTPND1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x00A4 - 0x00AF	-	-	-	-
0x00B0	MSGVAL2[B,H,W]		MSGVAL1[B,H,W]	
	00000000 00000000		00000000 00000000	
0x00B4 - 0x0FFC	-	-	-	-

1.37 Ethernet-MAC

Ethernet-MAC Base_Address : 0x4006_4000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0000 – 0x1FFC	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

Note:

- *For the register details of Ethernet-MAC block, refer to the "Ethernet part".*

1.38 Ethernet-Control

Ethernet-Control Base_Address : 0x4006_6000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000 - 0xFFC	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

Note:

- *For the register details of Ethernet-Control block, refer to the "Ethernet part".*

1.39 I2S

I2S Base_Address : 0x4006_C000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	RXFDAT[B,H,W] 00000000 00000000 00000000 00000000			
0x004	TXFDAT[B,H,W] 00000000 00000000 00000000 00000000			
0x008	CNTREG[B,H,W] 00000000 00000000 00000000 00000000			
0x00C	MCR0REG[B,H,W] -0000000 00000000 -0000000 00000000			
0x010	MCR1REG[B,H,W] 00000000 00000000 00000000 00000000			
0x014	MCR2REG[B,H,W] 00000000 00000000 00000000 00000000			
0x018	OPRREG[B,H,W] -------0 -------0 -------- -------0			
0x01C	SRST[B,H,W] -------- -------- -------- -------0			
0x020	INTCNT[B,H,W] -1111111 --111111 ----0000 --000000			
0x024	STATUS[B,H,W] 00000000 ----0000 00000000 00000000			
0x028	DMAACT[B,H,W] -------0 -------0 -------0 -------0			
0x02C	TSTREG[B,H,W] -------- -------- -------- -------0			
0x030 - 0xFFC	-	-	-	-

1.40 SD-Card

SD-Card Base_Address : 0x4006_E000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 – 0xFFC	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

Note:

- *For the register details of SD-Card block, refer to the "CHAPTER:SD Card Interface".*

1.41 CAN FD

CAN FD Base_Address : 0x4007_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	CREL[B,H,W] 00110000 00010011 00000101 0000110			
0x004	ENDN[B,H,W] 10000111 01100101 01000011 00100001			
0x008	-	-	-	-
0x00C	FBTP[B,H,W] ---00000 0--00000 ----1010 -011--11			
0x010	TEST[B,H,W] -------- -------- --000000 X000----			
0x014	RWD[B,H,W] -------- -------- 00000000 00000000			
0x018	CCCR[B,H,W] -------- -------- -0000000 00000001			
0x01C	BTP[B,H,W] ------00 00000000 --001010 00110011			
0x020	TSCC[B,H,W] -------- ----0000 -------- ------00			
0x024	TSCV[B,H,W] -------- -------- 00000000 00000000			
0x028	TOCC[B,H,W] 11111111 11111111 -------- -----000			
0x02C	TOCV[B,H,W] -------- -------- 11111111 11111111			
0x030 - 0x03C	-	-	-	-
0x040	ECR[B,H,W] -------- 00000000 00000000 00000000			
0x044	PSR[B,H,W] -------- -------- --000111 00000111			
0x048 - 0x04C	-	-	-	-
0x050	IR[B,H,W] 00000000 00000000 00000000 00000000			
0x054	IE[B,H,W] 00000000 00000000 00000000 00000000			
0x058	ILS[B,H,W] 00000000 00000000 00000000 00000000			
0x05C	ILE[B,H,W] -------- -------- -------- ------00			
0x060 - 0x07C	-	-	-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x080	GFC[B,H,W] -------- -------- -------- --000000			
0x084	SIDFC[B,H,W] -------- 00000000 00000000 000000--			
0x088	XIDFC[B,H,W] -------- -0000000 00000000 000000--			
0x08C	-	-	-	-
0x090	XIDAM[B,H,W] ---11111 11111111 11111111 11111111			
0x094	HPMS[B,H,W] -------- -------- 00000000 00000000			
0x098	NDAT1[B,H,W] 00000000 00000000 00000000 00000000			
0x09C	NDAT2[B,H,W] 00000000 00000000 00000000 00000000			
0x0A0	RXF0C[B,H,W] 00000000 -0000000 00000000 000000--			
0x0A4	RXF0S[B,H,W] ------00 --000000 --000000 -0000000			
0x0A8	RXF0A[B,H,W] -------- -------- -------- --000000			
0x0AC	RXBC[B,H,W] -------- -------- 00000000 000000--			
0x0B0	RXF1C[B,H,W] 00000000 -0000000 00000000 000000--			
0x0B4	RXF1S[B,H,W] 00----00 --000000 --000000 -0000000			
0x0B8	RXF1A[B,H,W] -------- -------- -------- --000000			
0x0BC	RXESC[B,H,W] -------- -------- -----000 -000-000			
0x0C0	TXBC[B,H,W] -0000000 --000000 00000000 000000--			
0x0C4	TXFQS[B,H,W] -------- --000000 ---00000 –000000			
0x0C8	TXESC[B,H,W] -------- -------- -------- -----000			
0x0CC	TXBRP[B,H,W] 00000000 00000000 00000000 00000000			
0x0D0	TXBAR[B,H,W] 00000000 00000000 00000000 00000000			
0x0D4	TXBCR[B,H,W] 00000000 00000000 00000000 00000000			
0x0D8	TXBTO[B,H,W] 00000000 00000000 00000000 00000000			
0x0DC	TXBCF[B,H,W] 00000000 00000000 00000000 00000000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0E0	TXBTIE[B,H,W] 00000000 00000000 00000000 00000000			
0x0E4	TXBCIE[B,H,W] 00000000 00000000 00000000 00000000			
0x0E8 - 0x0EC	-	-	-	-
0x0F0	TXEFC[B,H,W] --000000 --000000 00000000 000000--			
0x0F4	TXEFS[B,H,W] ------00 ---00000 ---00000 --000000			
0x0F8	TXEFA[B,H,W] -------- -------- -------- ---00000			
0x0FC - 0x1FC	-	-	-	-
0x200	FDSEAR[B,H,W] 00000000 00000000		FDESR[B,H,W] ------00	FDECR[B,H,W] ----0000
0x204	FDDEAR[B,H,W] 00000000 00000000		FDESCR[B,H,W] ------00	-
0x208	FDFECR[B,H,W] 0------- -----000 00000000 00000000			
0x20C	-	-	-	-
0x210	TSMDR[B,H,W] -------- -------0		TSCNTR[B,H,W] -------- -------0	
0x214	TSDIVR[B,H,W] -------- -------- 00000000 00000000			
0x218	TSCPCLR[B,H,W] 00000000 00000000		TSCDTR[B,H,W] 00000000 00000000	
0x21C - 0xFFC	-	-	-	-

CAN FD Message RAM

Base_Address + Address	Message RAM			
	+3	+2	+1	+0
0x8000 - 0xBFFC	**Rx Buffer and FIFO Element [W]** **Tx Buffer Element [W]** **Tx Event FIFO Element [W]** **Standard Message ID Filter Element [W]** **Extended Message ID Filter Element [W]**			

Note:

- *For the register details of CAN FD Message RAM block, refer to the "CHAPTER:CAN FD Controller".*

1.42 Programmable-CRC

Programmable-CRC Base_Address : 0x4008_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	CRCn_PORY[B,H,W] 00000100 11000001 00011101 10110111			
0x004	CRCn_SEED[B,H,W] 11111111 11111111 11111111 11111111			
0x008	CRCn_FXOR[B,H,W] 11111111 11111111 11111111 11111111			
0x00C	CRCn_CFG[B,H,W] 00000000 11100000 00000000 00000000			
0x010	CRCn_WR[B,H,W] 00000000 00000000 00000000 00000000			
0x014	CRCn_RD[B,H,W] 00000000 00000000 00000000 00000000			
0x018 - 0xFFC	-	-	-	-

1.43 WorkFlash_IF

WorkFlash_IF Base_Address : 0x200E_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WFASZR[B,H,W]			
0x004	WFRWTR[B,H,W]			
0x008	WFSTR[B,H,W]			
0x00C - 0xFFF	-	-	-	-

Note:

- *For the register details of Workflash IF block, refer to the "FLASH PROGRAMMING MANUAL" of the product used.*

1.44 Hi-Speed Quad SPI controller

Hi-Speed Quad SPI controller **Base_Address : 0xD000_0000**

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	HSSPIn_MCTRL[B,H,W] -------- -------- -------- --000-00			
0x004	HSSPIn_PCC0[B,H,W] -------- -1111111 00000000 00000000			
0x008	HSSPIn_PCC1[B,H,W] -------- -1111111 00000000 00000000			
0x00C	HSSPIn_PCC2[B,H,W] -------- -1111111 00000000 00000000			
0x010	HSSPIn_PCC3[B,H,W] -------- -1111111 00000000 00000000			
0x014	HSSPIn_TXF[B,H,W] -------- -------- -------- -0000000			
0x018	HSSPIn_TXE[B,H,W] -------- -------- -------- -0000000			
0x01C	HSSPIn_TXC[B,H,W] -------- -------- -------- -0000000			
0x020	HSSPIn_RXF[B,H,W] -------- -------- -------- -0000000			
0x024	HSSPIn_RXE[B,H,W] -------- -------- -------- -0000000			
0x028	HSSPIn_RXC[B,H,W] -------- -------- -------- -0000000			
0x02C	HSSPIn_FAULTF[B,H,W] -------- -------- -------- ---00000			
0x030	HSSPIn_FAULTC[B,H,W] -------- -------- -------- ---00000			
0x034	-	-	HSSPIn_DMDMAEN [B,H,W] ------00	HSSPIn_DMCFG [B,H,W] -----001
0x038	HSSPIn_DMTRP [B,H,W] ----0000	HSSPIn_DMPSEL [B,H,W] ------00	HSSPIn_DMSTOP [B,H,W] -------0	HSSPIn_DMSTART [B,H,W] -------0
0x03C	HSSPIn_DMBCS[B,H,W] 00000000 00000000		HSSPIn_DMBCC[B,H,W] 00000000 00000000	
0x040	HSSPIn_DMSTATUS[B,H,W] -------- ---00000 ---00000 ------00			
0x044	-	-	-	-
0x048	-	-	-	-
0x04C	HSSPIn_FIFOCFG[B,H,W] --------_--------_---00000_01110111			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x050	HSSPIn_TXFIFO0[B,H,W] 00000000 00000000 00000000 00000000			
0x054	HSSPIn_TXFIFO1[B,H,W] 00000000 00000000 00000000 00000000			
0x058	HSSPIn_TXFIFO2[B,H,W] 00000000 00000000 00000000 00000000			
0x05C	HSSPIn_TXFIFO3[B,H,W] 00000000 00000000 00000000 00000000			
0x060	HSSPIn_TXFIFO4[B,H,W] 00000000 00000000 00000000 00000000			
0x064	HSSPIn_TXFIFO5[B,H,W] 00000000 00000000 00000000 00000000			
0x068	HSSPIn_TXFIFO6[B,H,W] 00000000 00000000 00000000 00000000			
0x06C	HSSPIn_TXFIFO7[B,H,W] 00000000 00000000 00000000 00000000			
0x070	HSSPIn_TXFIFO8[B,H,W] 00000000 00000000 00000000 00000000			
0x074	HSSPIn_TXFIFO9[B,H,W] 00000000 00000000 00000000 00000000			
0x078	HSSPIn_TXFIFO10[B,H,W] 00000000 00000000 00000000 00000000			
0x07C	HSSPIn_TXFIFO11[B,H,W] 00000000 00000000 00000000 00000000			
0x080	HSSPIn_TXFIFO12[B,H,W] 00000000 00000000 00000000 00000000			
0x084	HSSPIn_TXFIFO13[B,H,W] 00000000 00000000 00000000 00000000			
0x088	HSSPIn_TXFIFO14[B,H,W] 00000000 00000000 00000000 00000000			
0x08C	HSSPIn_TXFIFO15[B,H,W] 00000000 00000000 00000000 00000000			
0x090	HSSPIn_RXFIFO0[B,H,W] 00000000 00000000 00000000 00000000			
0x094	HSSPIn_RXFIFO1[B,H,W] 00000000 00000000 00000000 00000000			
0x098	HSSPIn_RXFIFO2[B,H,W] 00000000 00000000 00000000 00000000			
0x09C	HSSPIn_RXFIFO3[B,H,W] 00000000 00000000 00000000 00000000			
0x0A0	HSSPIn_RXFIFO4[B,H,W] 00000000 00000000 00000000 00000000			
0x0A4	HSSPIn_RXFIFO5[B,H,W] 00000000 00000000 00000000 00000000			
0x0A8	HSSPIn_RXFIFO6[B,H,W] 00000000 00000000 00000000 00000000			
0x0AC	HSSPIn_RXFIFO7[B,H,W] 00000000 00000000 00000000 00000000			

Base_Address + Address	Register			
	+3	+2	+1	+0
0x0B0	HSSPIn_RXFIFO8[B,H,W] 00000000 00000000 00000000 00000000			
0x0B4	HSSPIn_RXFIFO9[B,H,W] 00000000 00000000 00000000 00000000			
0x0B8	HSSPIn_RXFIFO10[B,H,W] 00000000 00000000 00000000 00000000			
0x0BC	HSSPIn_RXFIFO11[B,H,W] 00000000 00000000 00000000 00000000			
0x0C0	HSSPIn_RXFIFO12[B,H,W] 00000000 00000000 00000000 00000000			
0x0C4	HSSPIn_RXFIFO13[B,H,W] 00000000 00000000 00000000 00000000			
0x0C8	HSSPIn_RXFIFO14[B,H,W] 00000000 00000000 00000000 00000000			
0x0CC	HSSPIn_RXFIFO15[B,H,W] 00000000 00000000 00000000 00000000			
0x0D0	HSSPIn_CSCFG[B,H,W] -------- ----0000 ----0000 --000000			
0x0D4	HSSPIn_CSITIME[B,H,W] -------- -------- 11111111 11111111			
0x0D8	HSSPIn_CSAEXT[B,H,W] 00000000 00000000 000----- --------			
0x0DC	HSSPIn_RDCSDC1[B,H,W] 00000000 ----0000		HSSPIn_RDCSDC0[B,H,W] 00000000 ----0000	
0x0E0	HSSPIn_RDCSDC3[B,H,W] 00000000 ----0000		HSSPIn_RDCSDC2[B,H,W] 00000000 ----0000	
0x0E4	HSSPIn_RDCSDC5[B,H,W] 00000000 ----0000		HSSPIn_RDCSDC4[B,H,W] 00000000 ----0000	
0x0E8	HSSPIn_RDCSDC7[B,H,W] 00000000 ----0000		HSSPIn_RDCSDC6[B,H,W] 00000000 ----0000	
0x0EC	HSSPIn_WRCSDC1[B,H,W] 00000000 ----0000		HSSPIn_WRCSDC0[B,H,W] 00000000 ----0000	
0x0F0	HSSPIn_WRCSDC3[B,H,W] 00000000 ----0000		HSSPIn_WRCSDC2[B,H,W] 00000000 ----0000	
0x0F4	HSSPIn_WRCSDC5[B,H,W] 00000000 ----0000		HSSPIn_WRCSDC4[B,H,W] 00000000 ----0000	
0x0F8	HSSPIn_WRCSDC7[B,H,W] 00000000 ----0000		HSSPIn_WRCSDC6[B,H,W] 00000000 ----0000	
0x0FC	HSSPIn_MID[B,H,W] 00000000 00000000 00000110 00110000			
0x100 - 0x3FC	-	-	-	-
0x400	-	-	-	QDCLKR[B,H,W] ----1111
0x404	-	-	-	DBCNT[B,H,W] ------00
0x408 - 0xFFC	-	-	-	-

B. List of Notes

This section explains notes for each function.

1. Notes when high-speed CR is used for the master clock

CODE: 9BPRECAUTION-E01.3

1. Notes when high-speed CR is used for the master clock

This section explains notes when the high-speed CR is used for the master clock.

The frequency of the high-speed CR varies depending on the temperature and/or the power supply voltage. The following table shows notes on each function macro when the high-speed CR is used for the master clock.

Furthermore, pay attention to notes when the high-speed CR is used as an input clock of the PLL and the master clock is selected for PLL.

■ Notes on Each Macro

Macro	Function/mode	Notes
Internal Bus Clock	HCLK/FCLK/PCLK0/ PCLK1/PCLK2/ TPIUCLK	When the frequency of the high-speed CR is the maximum value, the setting of the internal operating clock frequency shall not exceed the upper limit specified in the "data sheet" for the product that you are using.
Timer	Multi-function Timer Base Timer Watch Timer Dual Timer Watch Dog Timer Quadrature	The frequency variation of the high-speed CR should be considered for the timer count value of each macro.
A/D Converter	Sampling Time Compare Time	Considering the frequency variation of the high-speed CR, the sampling time and the compare time of the A/D converter shall satisfy the specification specified in the "data sheet" for the product that you are using.
USB	-	As the frequency accuracy does not meet the required specification, these macros cannot be used when the high-speed CR is used for the master clock.
Ethernet-MAC		
CAN		
CAN-FD		
I^2S		
Multi Function Serial Interface	UART	Even if the frequency of the high-speed CR is the minimum or the maximum value, the baud rate error should be considered. The baud rate error shall not exceed the limit.
	CSIO	The frequency variation of the high-speed CR should be considered for the communication of each macro.
	I2C	
	LIN	As the required frequency accuracy cannot be met, this function cannot be used as master. As a slave, the specified baud rate has more error at the maximum/minimum frequency of high-speed clock. So, if the error limit of the baud rate is exceeded, this function cannot be used.
Debug Interface	Serial Wire	As the frequency variation of the high-speed CR, the SWV(Serial Wire View) may not be used.
External Bus Interface	Clock Output	When the external bus clock output is used, the frequency variation of the high-speed CR should be considered for devices to be connected.
Hi-Speed Quad SPI	-	The frequency variation of the high-speed CR should be considered for devices to be connected.
SD card Interface	-	The frequency variation of the high-speed CR should be considered for devices to be connected.

MAJOR CHANGES

Page	Section	Changes
Revision 1.0		
-	-	Initial release
Revision 2.0		
7	The target product in this manual	Added TYPE1-M4, TYPE2-M4, TYPE3-M4 products.
24	CHAPTER1-2:UART (Asynchronous Serial Interface) overview of UART	Added Explanation of *3.
41	4.1 Baud rate settings	Added in Table 4-2, 100MHz.
134 -140	CHAPTER1-3: CSIO (Clock Synchronous Serial Interface) Interface) 5.Serial Chip Select Operation	Added the explanation corresponded to TYPE3-M4 product.
144	6.1 Baud rate settings	Added in Table 6-2,100MHz.
200	CHAPTER1-4: LIN Interface (Ver.2.1)(LIN Communication Control Interface Ver.2.1) 3.1 Baud rate settings	Added in Table 3-2,100MHz.
290	CHAPTER1-5: I2C Interface (I2C Communications Control Interface) 3.Dedicated Baud Rate Generator	Added in Table 3-1, 100MHz.
288	4.I2C communication operation flowchart examples	Corrected Figure 4-2.
392	CHAPTER3-1:USB Function 3.3 Operation of each register in response to a command	Added note.
472 - 478, 488	CHAPTER3-2: USB Host	Added the explanation about each register.
489	CHAPTER4:Ethernet	Added new Chapter
494	CHAPTER5-1:CAN Prescaler 2.1 CAN Prescaler Register (CANPRE)	Added in note, attention matter of TYPE3-M4 product.
557 - 702	CHAPTER5-3: CAN FD CONTROLLER	Added new Chapter
703 - 720	CHAPTER5-4: CAN FD Message RAM ECC Function	Added new Chapter
721 - 734	CHAPTER5-5: EXTERNAL TIME STAMP COUNTER FOR CAN FD	Added new Chapter
735 - 736	CHAPTER6-1: HDMI-CEC/Remote Control Reception	Added new Chapter
737 - 770	CHAPTER6-2: CEC Reception/ Remote Reception	Added new Chapter
771 - 790	CHAPTER6-3: CEC Transmission	Added new Chapter

Page	Section	Changes
791 - 808	CHAPTER7-1: I2S Clock Generation	Added new Chapter
809 - 856	CHAPTER7-2: I2S (Inter-IC Sound Bus) Interface	Added new Chapter
857 - 858	CHAPTER8-1: High-Speed Quad Serial Peripheral Interface Configuration	Added new Chapter
859 - 866	CHAPTER8-2: High-Speed Quad Serial Peripheral Interface Prescaler	Added new Chapter
867 - 966	CHAPTER8-3: High-Speed Quad Serial Peripheral Interface Controller	Added new Chapter
-	-	Company name and layout design change

MN709-00004-2v0-E

Spansion・Controller Manual

32-BIT MICROCONTROLLER
FM4 Family
PERIPHERAL MANUAL
Communication Macro Part

September 2014 Rev. 2.0

Published : Spansion Inc.
Edited : Communications Dept.

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.

Copyright © 2013-2014 Spansion. All rights reserved. Spansion®, the Spansion logo, MirrorBit®, MirrorBit® Eclipse™, ORNAND™, Easy DesignSim™, Traveo™ and combinations thereof, are trademarks and registered trademarks of Spansion LLC in the United States and other countries. Other names used are for informational purposes only and may be trademarks of their respective owners.