Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

32-Bit Microcontroller

FM0+ Family Peripheral Manual
Communication Macro Part

Doc. No. 002-05024 Rev. *C

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

© Cypress Semiconductor Corporation, 2013-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Preface

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

In addition, this manual is defined as separate volume which is extracted the Communication Macro part from the peripheral manual.

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.

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.*
- *Whether a peripheral function is on board or not is dependent on product type. See data sheets for details.*

Sample programs and development environment

Cypress offers sample programs free of charge for using the peripheral functions of the FM0+ family. Cypress also makes available descriptions of the development environment required for this series. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

Microcontroller support information:

https://community.cypress.com/community/MCU

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.*
  Cypress assumes no responsibility for any damage that may occur as a result of using a sample program.

Overall Organization of This Manual

Peripheral Manual Timer Part has 3 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 1-6 : MFS-I2S Interface
CHAPTER 2-1 : CAN Prescaler
CHAPTER 2-2 : CAN Controller
CHAPTER 3-1 : HDMI-CEC/Remote Control Reception
CHAPTER 3-2 : CEC Reception/Remote Reception
CHAPTER 3-3 : CEC Transmission
CHAPTER 4-1 : USB Clock Generation Block overview

CHAPTER 4-2 : USB Clock Generation (A)
CHAPTER 4-3 : USB Clock Generation (B)
CHAPTER 5-1 : USB Device (USB Function)
CHAPTER 5-2 : USB Host
CHAPTER 6 : Smart Card Interface
CHAPTER 7 : I2CSLAVE
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

- FM0+ Family PERIPHERAL MANUAL
  (Called "PERIPHERAL MANUAL" hereafter)
- FM0+ Family PERIPHERAL MANUAL Timer Part
  (Called "Timer Part" hereafter)
- FM0+ Family PERIPHERAL MANUAL Analog Macro Part
  (Called "Analog Macro Part" hereafter)
- FM0+ 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 FM0+ 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-M3 core, see the following documents that can be obtained from http://www.arm.com/.

- Cortex-M0+ Technical Reference Manual
- Armv6-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.

- FM0+ Family FLASH PROGRAMMING MANUAL

Note:

- *The 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 Communication Macro Part.

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 as follows.
  For the descriptions such as "TYPE1-M0+", “TYPE2-M0+” and “TYPE3-M0+”, see the relevant items of the target FM0+ product in the lists below.

Table 1 FM0+ family TYPE1 Product list

TYPE	Flash memory size	
	88 Kbytes	56 Kbytes
TYPE1-M0+	S6E1A12B S6E1A12C	S6E1A11B S6E1A11C

Table 2 FM0+ family TYPE2 Product list

TYPE	Flash memory size	
	304K bytes	560K bytes
TYPE2-M0+	S6E1B84E S6E1B84F S6E1B84G S6E1B34E S6E1B34F S6E1B34G	S6E1B86E S6E1B86F S6E1B86G S6E1B36E S6E1B36F S6E1B36G

Table 3 FM0+ family TYPE3 Product list

TYPE	Flash memory size	
	64K bytes	128K bytes
TYPE3-M0+	S6E1C31B S6E1C31C S6E1C31D S6E1C11B S6E1C11C S6E1C11D	S6E1C32B S6E1C32C S6E1C32D S6E1C12B S6E1C12C S6E1C12D

Table of Contents

Preface **3**
Related Manuals **5**
How to Use This Manual **6**
Table of Contents **8**
CHAPTER1-1: Multi-function Serial Interface **19**
1. Overview of the Multi-Function Serial Interface 20
CHAPTER 1-2: 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 41
   4.1 Baud Rate Settings 42
5. Setting Procedure and Program Flow in Operation Mode 0 (Asynchronous Normal Mode) 47
6. Setting Procedure and Program Flow in Operation Mode 1 (Asynchronous Multiprocessor Mode) 50
7. UART (Asynchronous Serial Interface) Registers 54
   7.1 Serial Control Register (SCR) 55
   7.2 Serial Mode Register (SMR) 58
   7.3 Serial Status Register (SSR) 60
   7.4 Extended Communication Control Register (ESCR) 63
   7.5 Received Data Register/Transmit Data Register (RDR/TDR) 65
   7.6 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 67
   7.7 FIFO Control Register 1 (FCR1) 69
   7.8 FIFO Control Register 0 (FCR0) 71
   7.9 FIFO Byte Register (FBYTE) 75
CHAPTER1-3: CSIO (Clock Synchronous Serial Interface) **77**
1. Overview of CSIO (Clock Synchronous Serial Interface) 78
2. CSIO (Clock Synchronous Serial Interface) Interrupts 79
   2.1 Received Interrupt and Flag Set Timing 81
   2.2 Interrupt and Flag Set Timing when Received FIFO is Used 83
   2.3 Transmit Interrupt and Flag Set Timing 85
   2.4 Interrupt and Flag Set Timing when Transmit FIFO is Used 87
   2.5 Timer Interrupt Occurrence and Flag Setting Timing 88
   2.6 Chip Select Error Occurrence and Flag Setting Timing 89

3. CSIO (Clock Synchronous Serial Interface) Operations 91
3.1 Normal Transfer (I) 91
3.2 Normal Transfer (II) 101
3.3 SPI Transfer (I) 112
3.4 SPI Transfer (II) 123
4. Serial Timer Operation 135
5. Serial Chip Select Operation 141
6. Dedicated Baud Rate Generator 153
6.1 Baud Rate Settings 154
6.2 CSIO (Clock Synchronous Serial Interface) Setup Procedure and Program Flow 158
7. CSIO (Clock Synchronous Serial Interface) Registers 161
7.1 Serial Control Register (SCR) 163
7.2 Serial Mode Register (SMR) 166
7.3 Serial Status Register (SSR) 169
7.4 Extended Communication Control Register (ESCR) 172
7.5 Received Data Register/Transmit Data Register (RDR/TDR) 174
7.6 Serial Support Control Register (SACSR) 176
7.7 Serial Timer Register (STMR) 180
7.8 Serial Timer Comparison Register (STMCR) 181
7.9 Serial Chip Select Control Status Register (SCSCR) 182
7.10 Serial Chip Select Timing Register (SCSTR3-0) 186
7.11 Serial Chip Select Format Register(SCSFR2-0) 189
7.12 Transfer Byte Register (TBYTE3-0) 193
7.13 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 194
7.14 FIFO Control Register 1 (FCR1) 196
7.15 FIFO Control Register 0 (FCR0) 198
7.16 FIFO Byte Register (FBYTE) 202
8. Restrictions on CSIO (Clock Synchronous Serial Interface) 204
CHAPTER1-4: LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1) 205
1. Overview of LIN Interface (Ver. 2.1) (LIN Communication Control Interface Ver. 2.1) 206
2. LIN Interface (Ver. 2.1) Interrupts 207
2.1 Received Interrupt and Flag Set Timing 209
2.2 Interrupt and Flag Set Timing when Received FIFO is Used 211
2.3 Transmit Interrupt and Flag Set Timing 213
2.4 Interrupt and Flag Set Timing when Transmit FIFO is Used 215
3. Dedicated Baud Rate Generator 216
3.1 Baud Rate Settings 217
4. LIN Interface (Ver. 2.1) Operations 222
5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow 236
6. LIN Interface (Ver. 2.1) Registers 241
6.1 Serial Control Register (SCR) 242
6.2 Serial Mode Register (SMR) 245
6.3 Serial Status Register (SSR) 247
6.4 Extended Communication Control Register (ESCR) 250
6.5 Received Data Register/Transmit Data Register (RDR/TDR) 252
6.6 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 254
6.7 FIFO Control Register 1 (FCR1) 256
6.8 FIFO Control Register 0 (FCR0) 258

6.9 FIFO Byte Register (FBYTE) 262
CHAPTER1-5: I2C Interface (I2C Communications Control Interface) 265
1. Overview of I²C Interface (I²C Communications Control Interface) 266
2. I²C Interface Operation 267
2.1 I²C Interface Interrupt 267
2.2 I²C bus Operation 269
2.3 Master Mode 270
2.3.1 Generating Start Condition 270
2.3.2 Slave Address Output 271
2.3.3 Acknowledgement Reception by First Byte Transmission 274
2.3.4 Data Transmission by the Master 282
2.3.5 Data Reception by the Master 304
2.3.6 Arbitration Lost 312
2.3.7 Wait State for Master Mode 312
2.3.8 Issuing Iteration Start Condition when DMA Mode is Enabled (SSR:DMA=1) 313
2.4 Slave Mode 314
2.4.1 Slave Address Match Detection 314
2.4.2 Data Direction Bit 316
2.4.3 Reception in Slave Mode 317
2.4.4 Transmission in Slave Mode 325
2.5 Bus Error 326
2.5.1 Bus Error Occurrence Condition 326
2.5.2 Bus Error Operation 326
3. Dedicated Baud Rate Generator 327
3.1 Selecting the Baud Rate 327
3.2 Calculating the Baud Rate 327
3.3 Reload Values and Baud Rates for Each Bus Clock Frequency 328
3.4 Functions of Reload Counter 328
3.5 Starting Counting 328
4. I²C Communication Operation Flowchart Examples 329
5. I²C Interface Registers 336
5.1 I²C Bus Control Register (IBCR) 337
5.2 Serial Mode Register (SMR) 344
5.3 I²C Bus Status Register (IBSR) 346
5.4 Serial Status Register (SSR) 351
5.5 Received Data Register/Transmit Data Register (RDR/TDR) 356
5.6 Extension I²C Bus Control Register (EIBCR) 358
5.7 7-bit Slave Address Mask Register (ISMK) 360
5.8 7-bit Slave Address Register (ISBA) 362
5.9 Baud Rate Generator Registers 1 and 0 (BGR1 and BGR0) 364
5.10 FIFO Control Register 1 (FCR1) 365
5.11 FIFO Control Register 0 (FCR0) 367
5.12 FIFO Byte Register (FBYTE) 371
CHAPTER 1-6: MFS-I²S (Inter-IC Sound Bus) 373
1. Overview of MFS-I²S 374
2. Configuration of MFS-I²S Interface 375
3. Data Structure 376
4. MFS-I²S interrupts 377

4.1 Interrupt and Flag Set Timing When Received FIFO is Used 378
4.2 Interrupt and Flag Set Timing When Transmit FIFO is Used 379
5. MFS-I²S Registers 380
5.1 Serial Control Register (SCR) 381
5.2 Serial Mode Register (SMR) 383
5.3 Serial Status Register (SSR) 385
5.4 Extended Communication Control Register (ESCR) 387
5.5 Received Data Register/Transmit Data Register (RDR/TDR) 388
5.6 FIFO Control Register 1 (FCR1) 390
5.7 FIFO Control Register 0 (FCR0) 392
5.8 FIFO Byte Register (FBYTE) 394
6. MFS-I²S Clock Generator Registers 396
6.1 Control Register (CNTLREG) 397
6.2 MFS-I²S Clock Registers (I2SCLK) 399
6.3 MFS-I2S Status Registers (I2SST) 402
6.4 MFS-I2S Reset Registers (I2SRST) 403
7. MFS-I²S Interface Operation Description 404
7.1 Data transmit operation 404
7.2 Data received operation 405
8. User Precaution 406
8.1 Connection Diagram 406
8.2 MFS-I²S and MSB-Justified Protocol 407
CHAPTER 2-1: CAN Prescaler 409
CHAPTER 2-2: CAN Controller 411
CHAPTER3-1: HDMI-CEC/Remote Control Reception 413
1. Configuration 414
2. Revision 415
3. Usage notes of HDMI-CEC 416
CHAPTER3-2: CEC Reception/ Remote Reception 417
1. Overview 418
2. Configuration 419
3. Operations 420
3.1 SIRCS Mode 420
3.1.1 Operational Flow Chart and Waves of SIRCS Mode 420
3.1.2 Basic Operations of SIRCS Mode 421
3.1.3 Start Bit Detection and Interrupt Output 421
3.1.4 Minimum Pulse Width Violation 422
3.1.5 Device Address Comparison 422
3.1.6 Counter Overflow Detection and Interrupt Output 423
3.2 Operations of NEC/Association for Electric Home Appliances Mode 424
3.2.1 Operational Flow Chart and Waves of NEC/Association for Electric Home Appliances Mode 424
3.2.2 Start Bit Detection 425
3.2.3 Repeat Code Detection 427
3.2.4 Minimum Pulse Width Violation 427
3.2.5 Counter Overflow Detection and Interrupt Output 428
3.3 HDMI-CEC Mode 429
3.3.1 Operational Flow Chart and Waves in HDMI-CEC Mode 429

3.3.2 Start Bit Detection and Interrupt Output 431
3.3.3 Minimum Pulse Width Violation 431
3.3.4 Counter Overflow Detection and Interrupt Output 432
3.3.5 Device Address Comparison 432
3.3.6 Data Bit Width Violation and Error Pulse Automatic Output 433
3.3.7 EOM Detection 434
3.3.8 ACK Detection and Interrupt Output 434
3.4 Noise Filter 435
4. Example of Setting 436
5. Registers 437
5.1 Reception Control Register (RCCR) 438
5.2 Reception Interrupt Control Register (RCST) 440
5.3 Device Address Setting Register 1, 2 (RCADR1, RCADR2) 442
5.4 Start Bit Detection Width Setting Register (RCSHW) 443
5.5 Minimum Pulse Width Setting Register (RCDAHW) 444
5.6 Threshold Value Setting Register (RCDBHW) 445
5.7 Data Save Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL) 446
5.8 Clock Division Setting Register (RCCKD) 447
5.9 Repeat Code Interrupt Control Register (RCRC) 448
5.10 Repeat Code Detection Width Setting Register (RCRHW) 449
5.11 Data Bit Width Violation Interrupt Control Register (RCLE) 450
5.12 Maximum Data Bit Width Setting Register (RCLELW) 452
5.13 Minimum Data Bit Width Setting Register (RCLESW) 453
CHAPTER3-3: CEC Transmission 455
1. Overview of CEC Transmission 456
2. Block Diagram of CEC Transmitting Circuit 457
3. CEC Transmission Interrupts 458
4. CEC Transmission Registers 459
5. CEC Transmission Operations 460
5.1 CEC Transmission Operations 461
5.2 Interrupt Factors and Timing Chart 462
5.3 Arbitration Lost Detection 463
5.4 Signal Free Detection 465
5.5 Filtering 466
5.6 CEC Transmission Operations Flow 467
6. CEC Transmission Register Set 468
6.1 Transmission Control Register (TXCTRL) 469
6.2 Transmission Data Register (TXDATA) 471
6.3 Transmission Status Register (TXSTS) 472
6.4 Signal Free Time Setting Register (SFREE) 474
CHAPTER4-1: USB Clock Generation Block Overview 475
1. Overview and Configuration 476
CHAPTER 4-2: USB Clock Generation (A) 479
1. Overview 480
2. Configuration and Block Diagram 481
3. Explanation of Operation 482
4. Setup Procedure Example 484
5. Register List 485

5.1 USB Clock Control Register (UCCR) 486
5.2 USB-PLL Control Register1 (UPCR1) 487
5.3 USB-PLL Control Register 2 (UPCR2) 488
5.4 USB-PLL Control Register 3 (UPCR3) 489
5.5 USB-PLL Control Register 4 (UPCR4) 490
5.6 USB-PLL Control Register 5 (UPCR5) 491
5.7 USB-PLL Status Register (UP_STR) 492
5.8 USB-PLL Interrupt Factor Enable Register (UPINT_ENR) 493
5.9 USB-PLL Interrupt Factor Status Register (UPINT_STR) 494
5.10 USB-PLL Interrupt Factor Clear Register (UPINT_CLR) 495
5.11 USB (ch.0) Enable Register (USBEN0) 496
5.12 USB (ch.1) Enable Register (USBEN1) 497
6. Usage Precautions 498

CHAPTER4-3: USB Clock Generation(B) 499

1. Overview 500
2. Configuration and Block Diagram 500
3. Explanation of Operation 501
3.1 Selecting the USB Clock 501
3.2 Changing to Standby Mode 501
4. Setup Procedure Example 502
5. Register 503
5.1 Register List 503
5.2 USB Clock Control Register (UCCR) 504
5.3 USB Frequency Divider Register (UPCR6) 505
5.4 USB (ch.0) Enable Register (USBEN0) 506
6. Usage Precautions 507

CHAPTER 5-1: USB Device (USB Function) 509

1. Overview of USB Device (USB Function) 510
1.1 Features of USB Device (USB Function) 510
2. Configuration of USB Device (USB Function) 511
3. Operations of USB Device (USB Function) 513
3.1 USB Device (USB Function) Operation 514
3.2 Detection of Connection and Disconnection 517
3.3 Operation of Each Register in Response to a Command 520
3.4 Suspend Function 523
3.5 Wake-up Function 524
3.6 DMA Transfer Function 526
3.7 NULL Transfer Function 531
3.8 STALL Response/release of Endpoint 0 532
3.9 STALL Response/release of Endpoint 1 to Endpoint 5 534
4. Examples of USB Device (USB Function) Setting Procedures 538
5. USB Device (USB Function) Registers 545
5.1 UDC Control Register (UDCC) 547
5.2 EP0 Control Register (EP0C) 550
5.3 EP1 to EP5 Control Registers (EP1C to EP5C) 552
5.4 Time Stamp Register (TMSP) 555
5.5 UDC Status Register (UDCS) 556
5.6 UDC Interrupt Enable Register (UDCIE) 558

5.7 EP0I Status Register (EP0IS) 560
5.8 EP0O Status Register (EP0OS) 562
5.9 EP1 to EP5 Status Registers (EP1S to EP5S) 564
5.10 EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) 568
CHAPTER 5-2: USB Host 571
1. Overview of USB Host 572
2. USB Host Configuration 573
3. USB Host Operations 574
3.1 Device Connection 575
3.2 USB Bus Resetting 577
3.3 Token Packet 578
3.4 Data Packet 581
3.5 Handshake Packet 582
3.6 Retry Function 583
3.7 SOF Interrupt 584
3.8 Error Status 586
3.9 End of Packet 587
3.10 Suspend and Resume Operations 588
3.11 Device Disconnection 591
4. USB Host Setting Procedure Examples 592
5. USB Host Registers 597
5.1 Host Control Registers 0 and 1 (HCNT0 and HCNT1) 599
5.2 Host Interrupt Register (HIRQ) 604
5.3 Host Error Status Register (HERR) 608
5.4 Host Status Register (HSTATE) 611
5.5 SOF Interrupt Frame Compare Register (HFCOMP) 614
5.6 Retry Timer Setup Register (HRTIMER) 615
5.7 Host Address Register (HADR) 617
5.8 EOF Setup Register (HEOF) 618
5.9 Frame Setup Register (HFRAME) 620
5.10 Host Token Endpoint Register (HTOKEN) 621
CHAPTER6: Smart Card Interface 623
1. Overview of Smart Card Interface 624
2. Smart Card Interface Configuration 625
3. Smart Card Interface Operation 626
3.1 Transmit/Received Data Format 626
3.2 Data Transmission 627
3.3 Data Reception 629
3.4 Baud Rate Configuration 631
3.5 Start Bit Detection 631
3.6 Error Detection 632
3.7 Output Inverse 633
3.8 Port Control 633
4. Smart Card Interface Interrupt 635
4.1 Received Interrupt and Flag Set Timing 636
4.2 Interrupt and Flag Set Timing when Read FIFO is Used 637
4.3 Transmit Interrupt and Flag Set Timing 639
4.4 Interrupt and Flag Set Timing when Write FIFO is Used 639

4.5 Card Event Interrupt and Flag Set Timing 641
4.6 Idle Timer Expired Interrupt and Flag Set Timing 641
5. Smart Card Interface Setting Procedure and Program Flow 643
6. Smart Card Interface Registers 647
6.1 Global Control Register 1 (GLOBALCONTROL1) 648
6.2 Status Register (STATUS) 652
6.3 Port Control Register (PORTCONTROL) 656
6.4 Data Register (DATA) 659
6.5 Card Clock Frequency Register (CARDCLOCK) 660
6.6 Baud Rate Register (BAUDRATE) 661
6.7 Guard Timer Register (GUARDTIMER) 662
6.8 Idle Timer Register (IDLETIMER) 663
6.9 Global Control Register 2 (GLOBALCONTROL2) 664
6.10 FIFO Access Register (DATA_FIFO) 666
6.11 Read FIFO Level Register (FIFO_LEVEL_READ) 667
6.12 Write FIFO Level Register (FIFO_LEVEL_WRITE) 668
6.13 FIFO Mode Register (FIFO_MODE) 669
6.14 Write FIFO Clear Register (FIFO_CLEAR_MSB_WRITE) 671
6.15 Read FIFO Clear Register (FIFO_CLEAR_MSB_READ) 672
6.16 Interrupt Status Register (IRQ_STATUS) 673
CHAPTER7: I2CSLAVE 677
1. Overview 678
2. Configuration 680
3. Operation 681
3.1. Slave Address 681
3.2. Reserved Address 683
3.3. Transmission and Reception of data 685
3.4. Wake-Up 687
3.5. Interrupt Request 687
3.6. Data setup time 688
3.7. Example for the operation steps 689
4. Registers 691
4.1. I2C Bus Slave Status Register (IBSSR) 692
4.2. I2C Bus Slave Control Register (IBSCR) 697
4.3. I2C Bus 7-bit Slave Address Register (IBSADR) 700
4.4. I2C Bus 7-bit Slave Address Mask Register (IBSMSKR) 701
4.5. I2C Bus Slave Data Setup Register (IBSDSTUPR) 702
4.6. I2C Bus Slave Transmission Data Register (IBSTDR) 703
4.7. I2C Bus Slave Reception Data Register (IBSRDR) 703
4.8. I2C Bus Slave Status Clear Register (IBSSCR) 704
4.9. I2C Bus Slave Status Set Register (IBSSSR) 706
APPENDIXES 707
A. Product Type 709
1. Product TYPE List 710
B. Register Map (TYPE1-M0+) 711
1. Register Map 712
1.1 Flash I/F 713
1.2 Unique ID 714

1.3 Clock/Reset 715
1.4 HW WDT 717
1.5 SW WDT 718
1.6 Dual Timer 719
1.7 MFT 720
1.8 PPG 724
1.9 Base Timer 728
1.10 IO Selector for Base Timer 729
1.11 QPRC 730
1.12 QPRC NF 731
1.13 A/DC 732
1.14 D/AC 734
1.15 CR Trim 735
1.16 EXTI 736
1.17 INT-Req. READ 737
1.18 GPIO 739
1.19 HDMI-CEC 746
1.20 LVD 747
1.21 DS Mode 748
1.22 MFS 749
1.23 CRC 751
1.24 Watch Counter 752
1.25 RTC 753
1.26 Low-speed CR Prescaler 754
1.27 Peripheral Clock Gating 755
1.28 DMAC 756
1.29 MTB_DWT 758
1.30 Fast GPIO 760
C. Register Map (TYPE2-M0+) 765
1. Register Map 766
1.1 Flash I/F 767
1.2 Unique ID 768
1.3 Clock/Reset 769
1.4 HW WDT 771
1.5 SW_WDT 772
1.6 Dual Timer 773
1.7 MFT 774
1.8 PPG 778
1.9 Base Timer 782
1.10 IO Selector for Base Timer 783
1.11 QPRC 784
1.12 QPRC NF 785
1.13 A/DC 786
1.14 D/AC 788
1.15 CR Trim 789
1.16 EXTI 790
1.17 INT-Req. READ 791
1.18 LCDC 793

1.19 GPIO 794
1.20 HDMI-CEC 801
1.21 LVD 802
1.22 DS Mode 803
1.23 USB Clock 805
1.24 MFS 806
1.25 CRC 808
1.26 Watch Counter 809
1.27 RTC 810
1.28 Low-speed CR Prescaler 814
1.29 Peripheral Clock Gating 815
1.30 Smart Card I/F 816
1.31 MFSI2S 817
1.32 High Resilience 818
1.33 USB 819
1.34 DSTC 821
1.35 MTB_DWT 822
1.36 Fast GPIO 824
D. Register Map (TYPE3-M0+) 829
1. Register Map 830
1.1 Flash I/F 831
1.2 Unique ID 832
1.3 Clock/Reset 833
1.4 HW WDT 835
1.5 SW WDT 836
1.6 Dual Timer 837
1.7 Base Timer 838
1.8 IO Selector for Base Timer 839
1.9 A/DC 840
1.10 CR Trim 842
1.11 EXTI 843
1.12 INT-Req. READ 844
1.13 GPIO 846
1.14 HDMI-CEC 851
1.15 LVD 852
1.16 DS Mode 853
1.17 USB Clock 855
1.18 I2CSLAVE 856
1.19 MFS 857
1.20 CRC 859
1.21 Watch Counter 860
1.22 RTC 861
1.23 Low-speed CR Prescaler 862
1.24 Peripheral Clock Gating 863
1.25 Smart Card I/F 864
1.26 MFSI2S 865
1.27 USB 866
1.28 DSTC 868

1.29 MTB_DWT 869
1.30 Fast GPIO 871
1.31 VIR 873
E. List of Notes 875
1. Notes when High-speed CR Is Used for Master Clock 876
F. Major Changes 877
Revision History 879

CHAPTER1-1: 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_FM0-E03.0

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 and I^2S 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)", "I^2C (I^2C bus interface)", and "I^2S Interface" for details about each interface.*
- *When I^2S interface is used, set CSIO mode.*

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 and I^2S 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

For the maximum FIFO capacity which differ by products, see “Data Sheet” of the product used.

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_FM0-E03.0_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

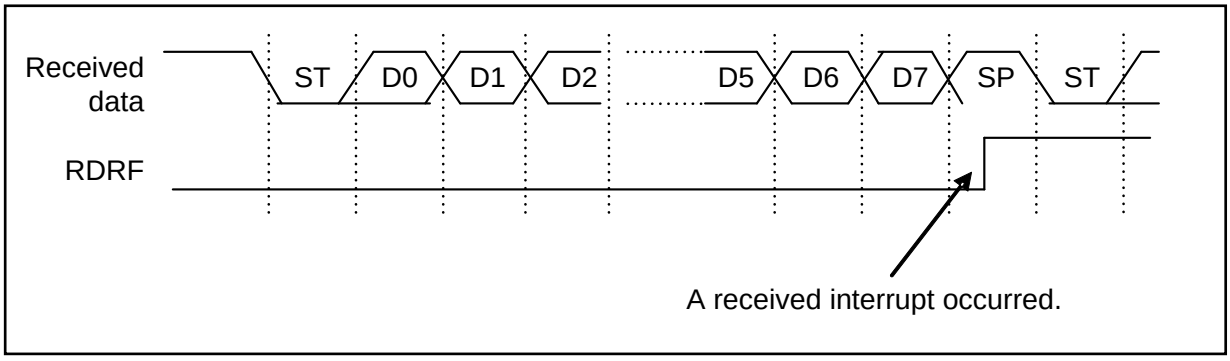
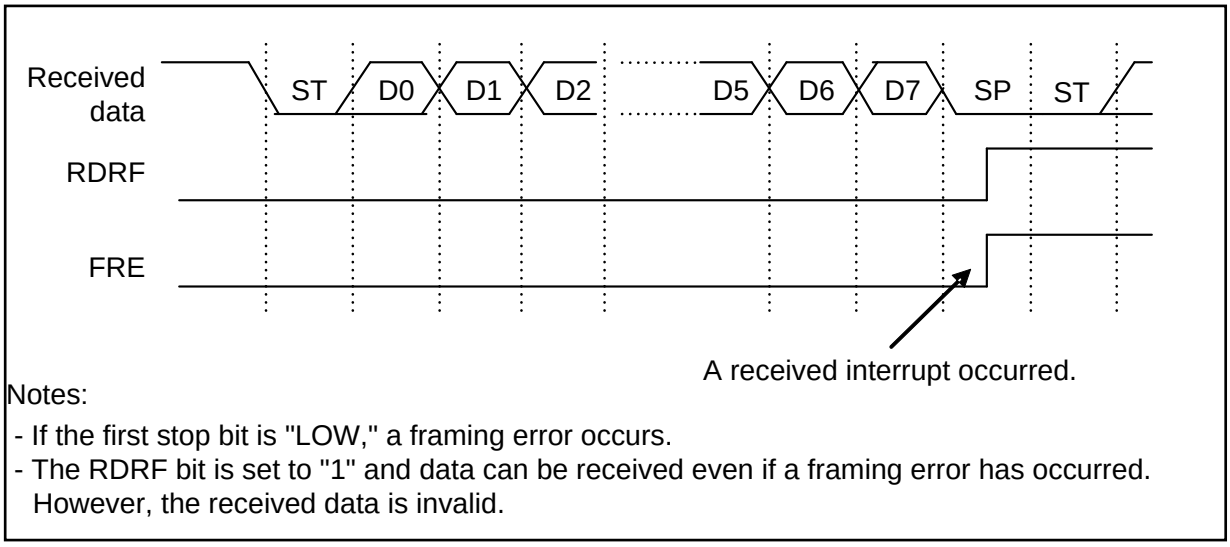


Figure 2-2 FRE (Framing Error) Flag Bit Set Timing

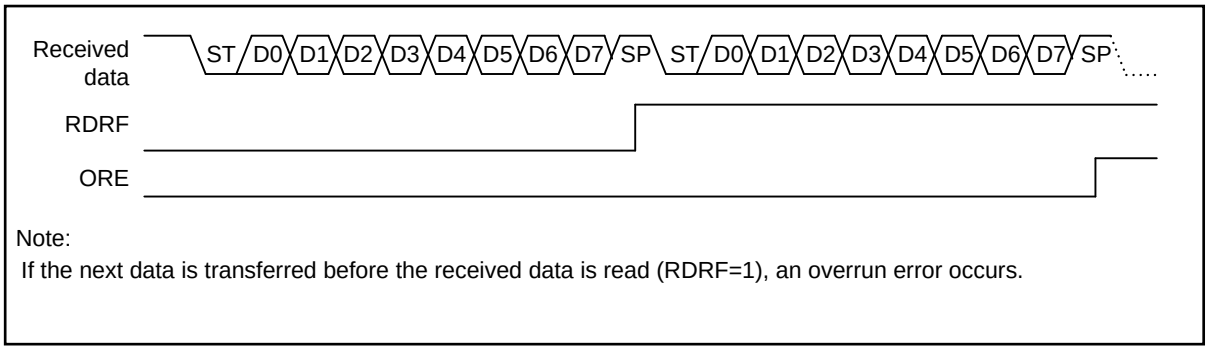


Note:

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

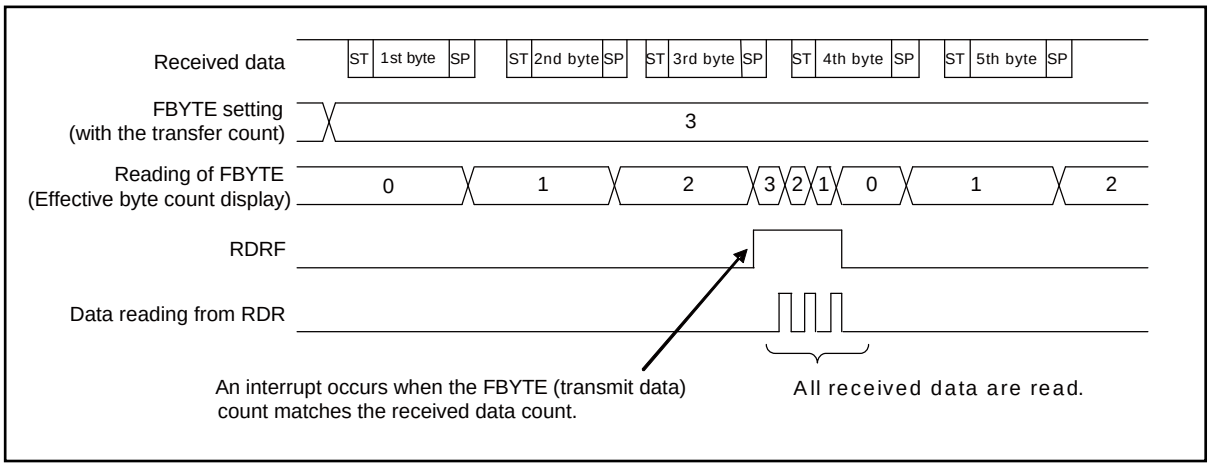
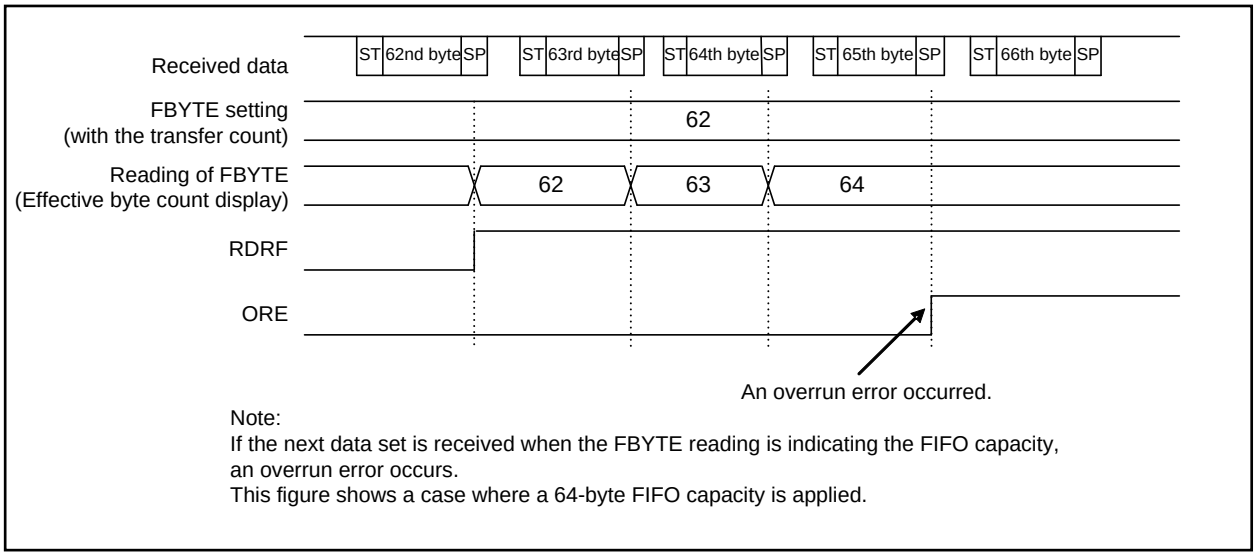


Figure 2-5 ORE (Overrun Error) Flag Bit Set Timing



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

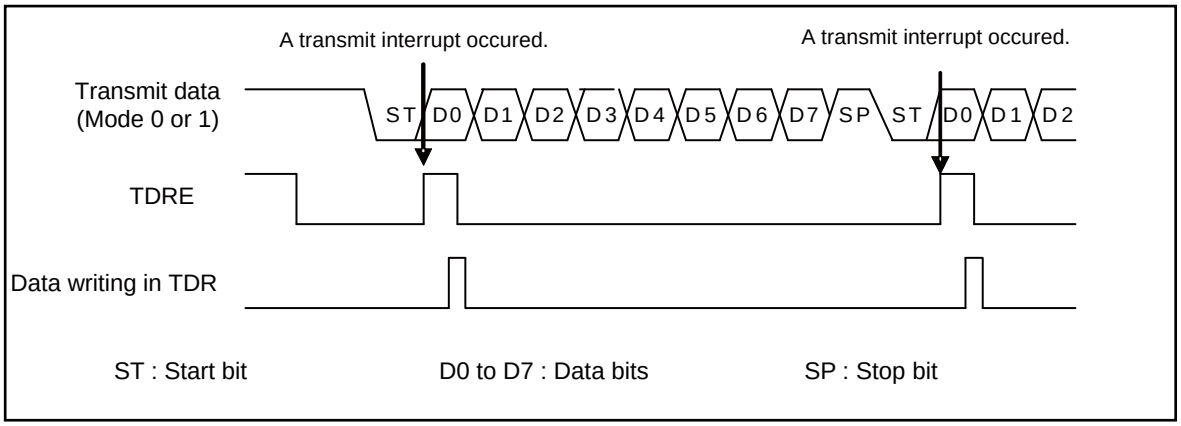
A transmit interrupt occurs when transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE = 1) and transmission starts and when no transmission is performed (SSR:TBI = 1).

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).

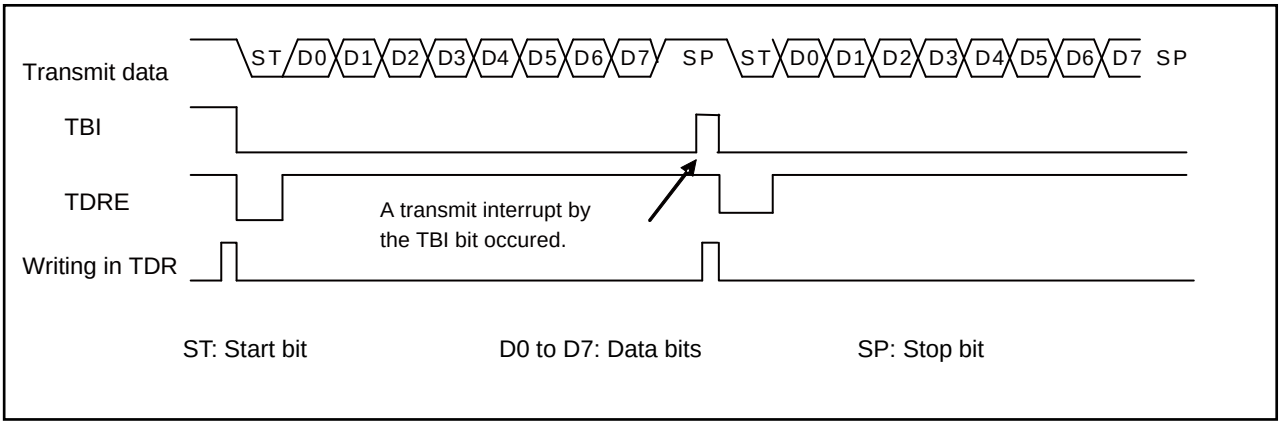
Figure 2-6 Transmit Data Empty Flag (SSR:TDRE) Set Timing



- 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



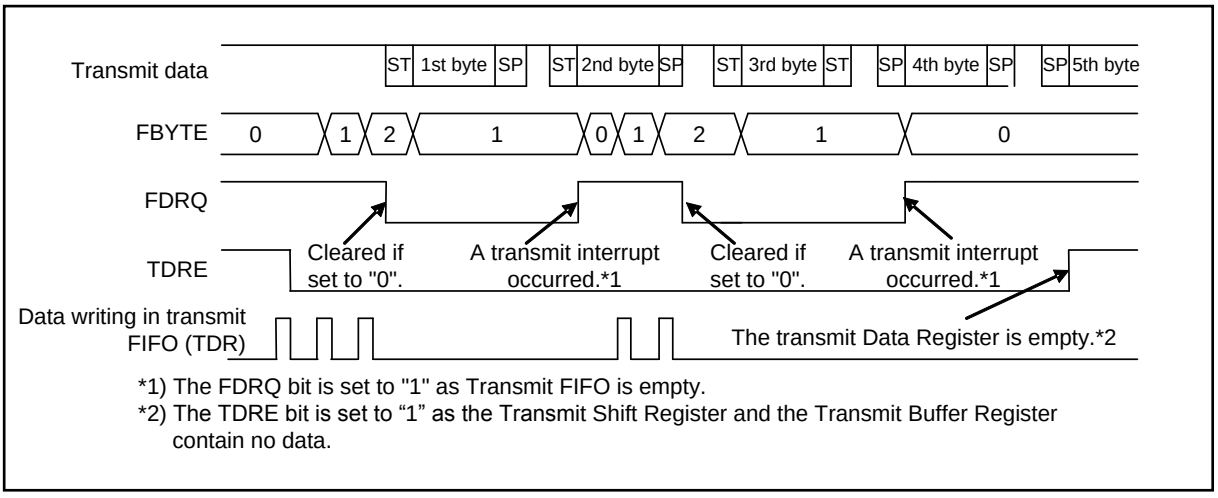
2.4 Interrupt and Flag Set Timing when Transmit FIFO is Used

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

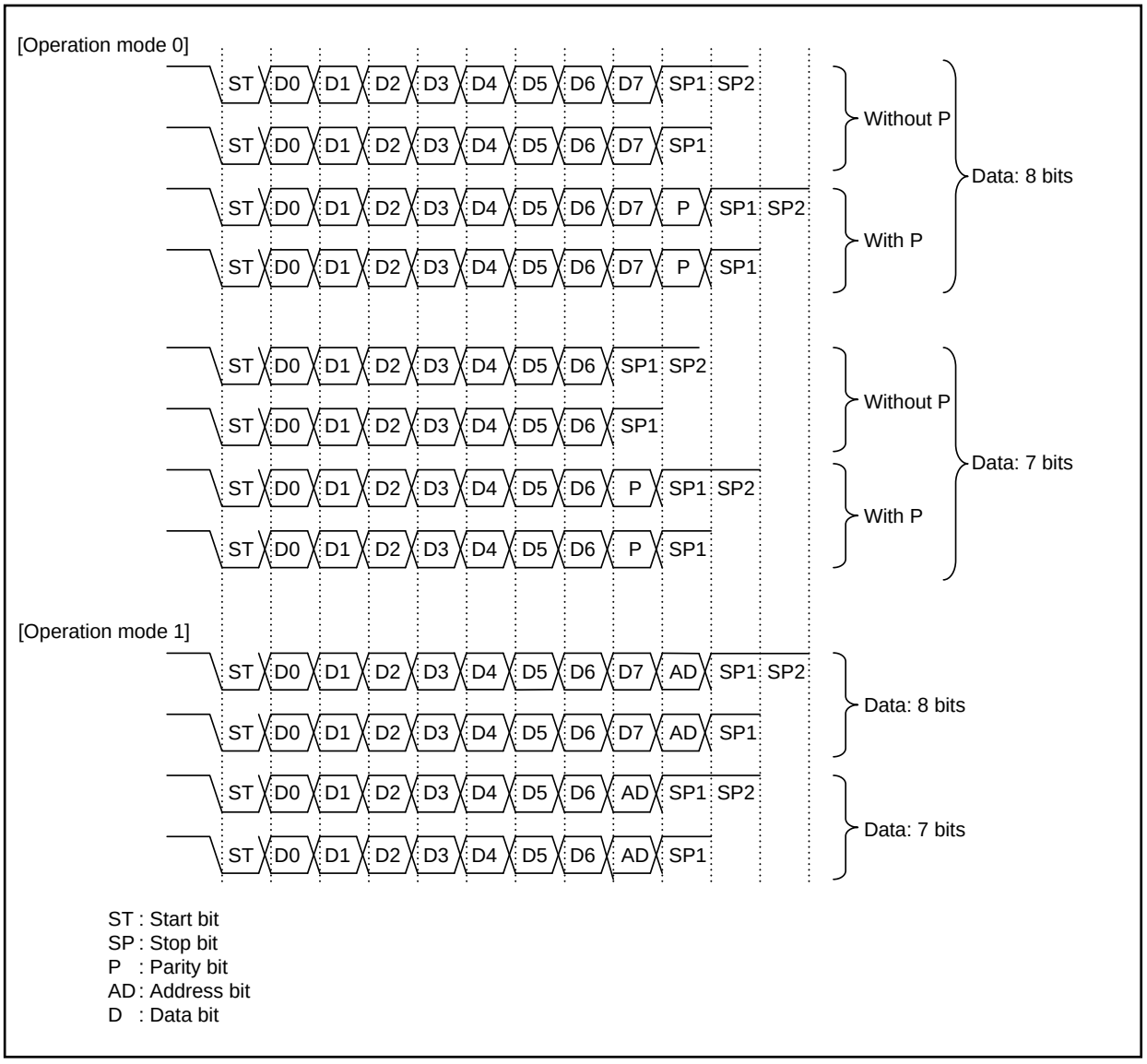
UART operates in bi-directional serial asynchronous communications in mode 0 and master/slave multiprocessor communications in mode 1.

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:

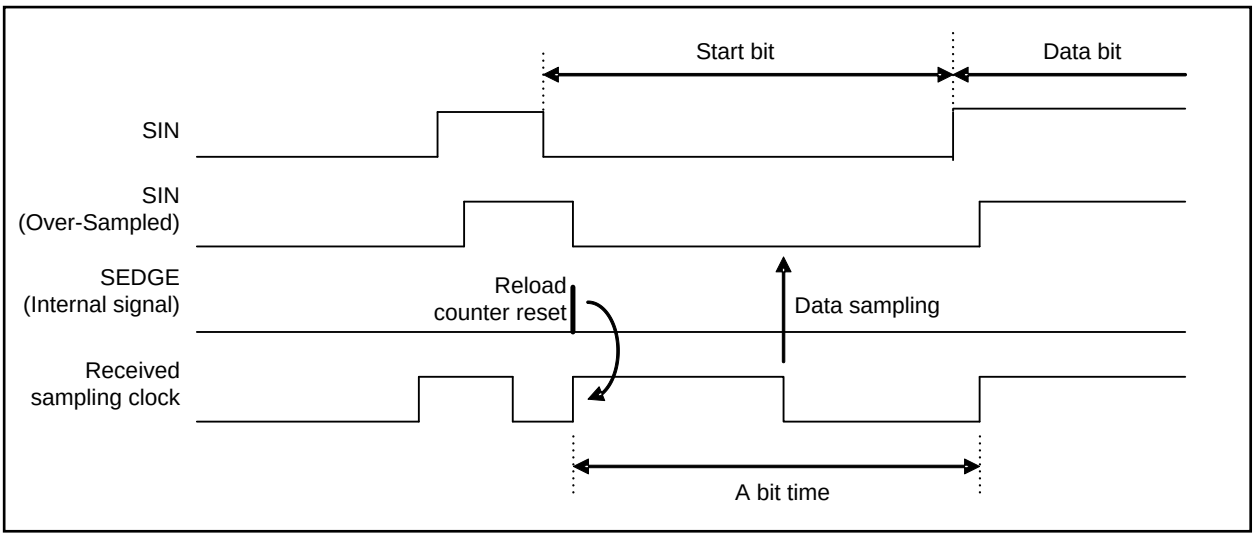
- *As the transmit data empty flag bit (SSR:TDRE) is initially set to 1, a transmit interrupt occurs as soon as transmit interrupts are enabled (SCR:TIE).*
- *As the FIFO transmit data request bit (FCR1:FDRQ) is initially set to 1, a transmit interrupt occurs as soon as FIFO transmit interrupts are enabled (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. The external clock is input from SCK.
- 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.

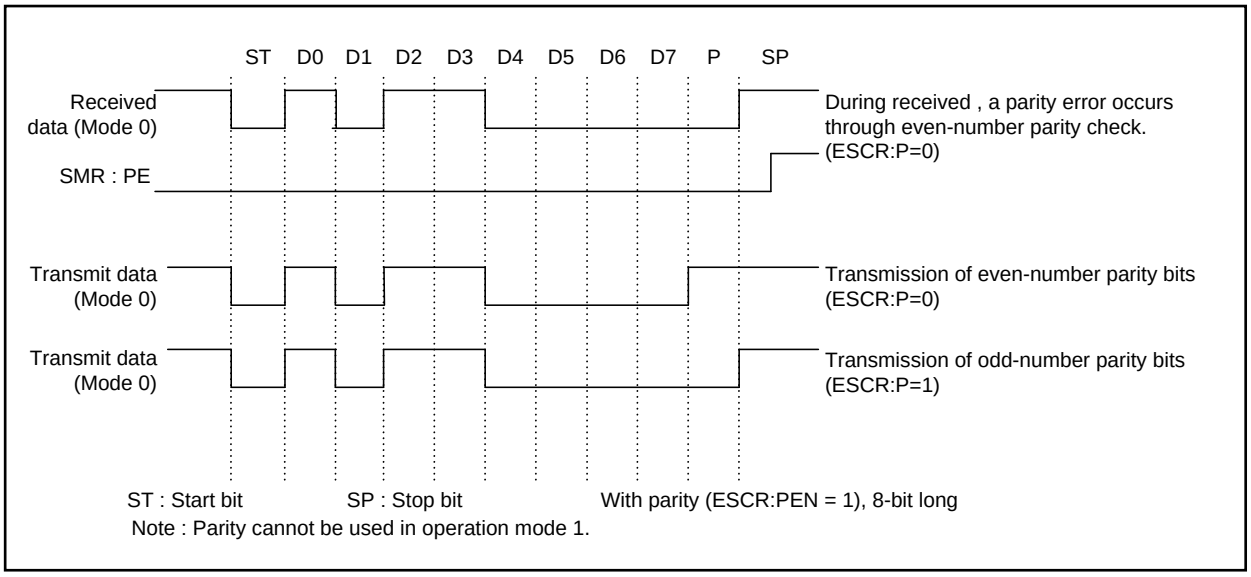


- Stop bit
  - You can select the bit length to be between one and four.
  - The received data full flag bit (SSR:RDRF) is set to 1 upon detection of the first stop bit.
- Error detection
  - In operation mode 0, parity, overrun and framing errors can be detected.
  - In operation mode 1, overrun and 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

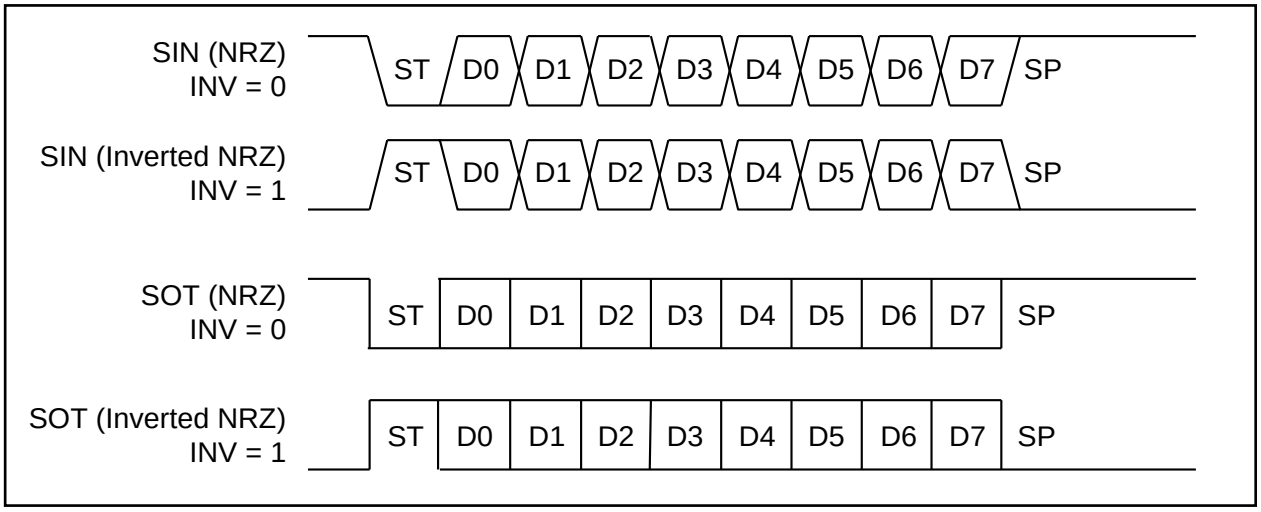


- Data signaling system

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

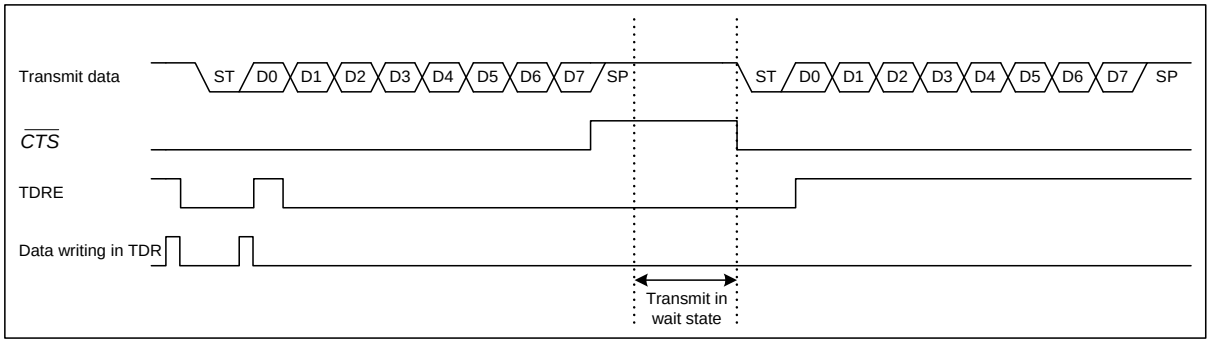
When flow control is enabled (ESCR:FLWEN=1), UART performs hardware flow control.

- During data transmission

  If $\overline{\text{CTS}}$ 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 $\overline{\text{CTS}}$ is set to LOW. To have transmission wait, input HIGH in $\overline{\text{CTS}}$ before the stop bit transmission is completed.

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

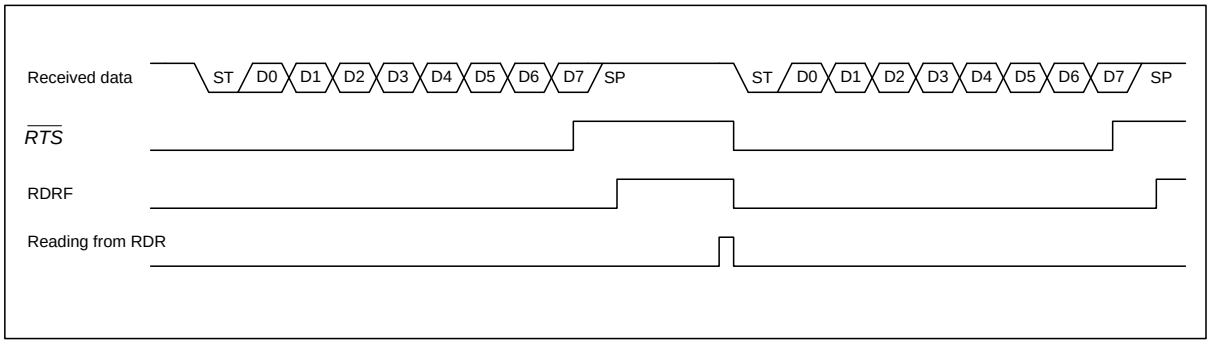
Figure 3-4 Hardware Flow Control During Data Transmission (SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)



- During data reception
- If FIFO is not used

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

Figure 3-5 Hardware Flow Control During Data Reception (with FIFO is Unused.) (SMR:SBL=0, ESCR:ESBL=INV=PEN=L2=L1=L0=0)

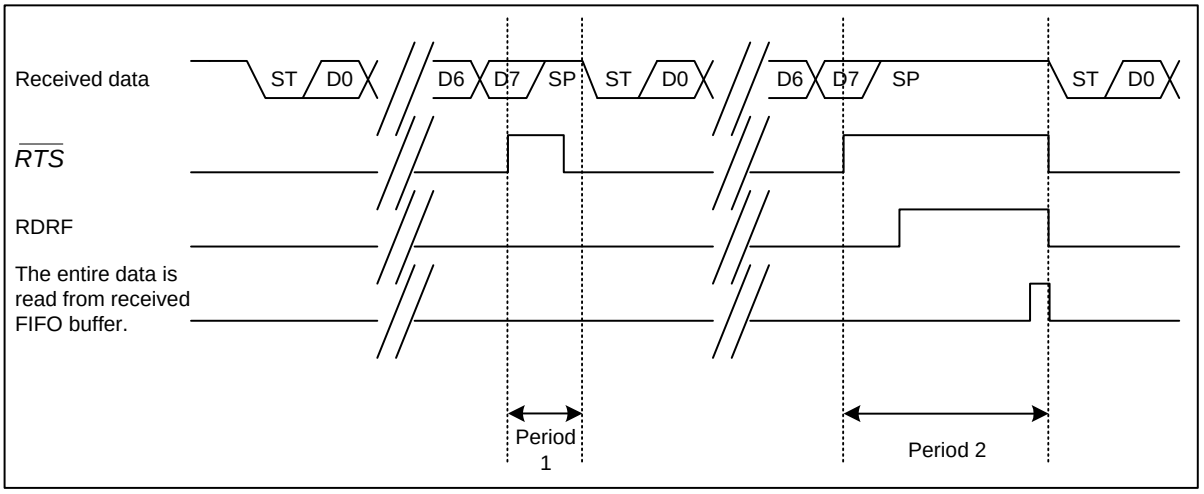


- If FIFO is used

  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. The external clock is input from SCK.
  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 × 100

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

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

Baud 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 (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	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	-	-	-	-	-	-	-	-	-	-

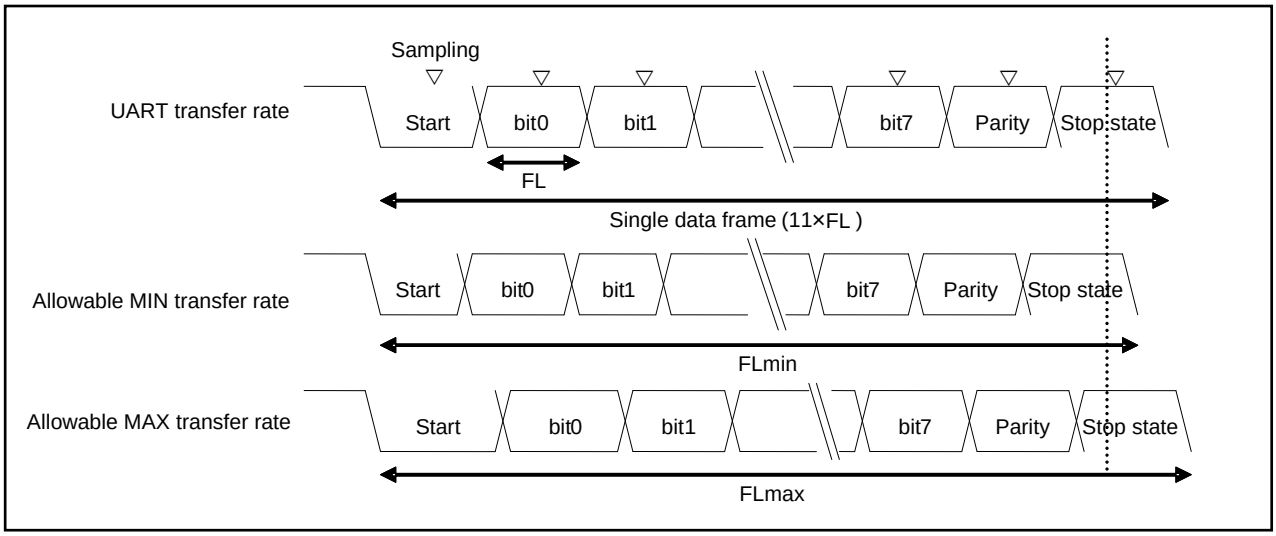
For frequencies not described in Table 4-1 and Table 4-2, calculate them conforming to “4.1 Baud Rate Settings”. (However, for the maximum frequencies which differ by products, see Data Sheet of the product used.

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. The external clock is input from SCK.

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)

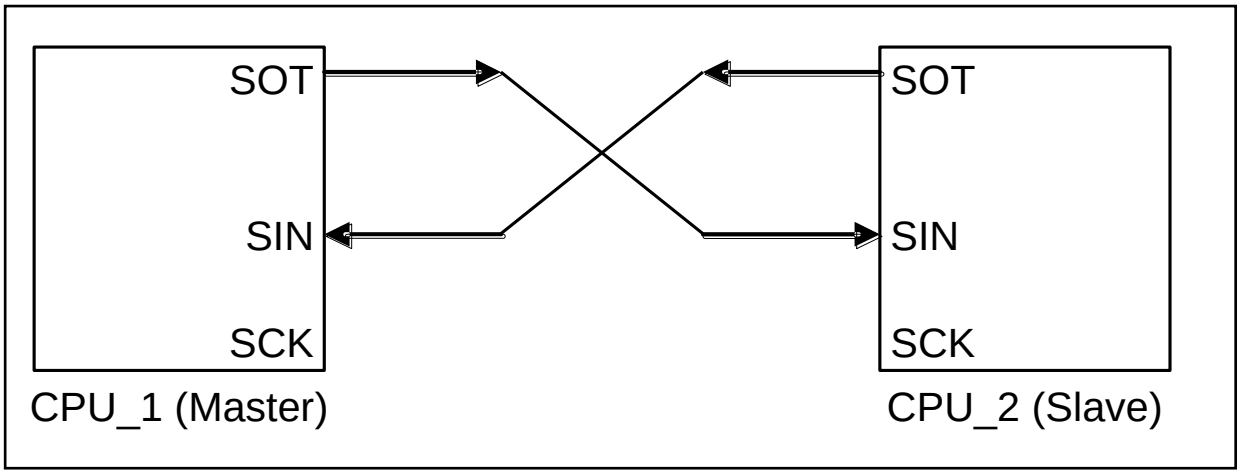
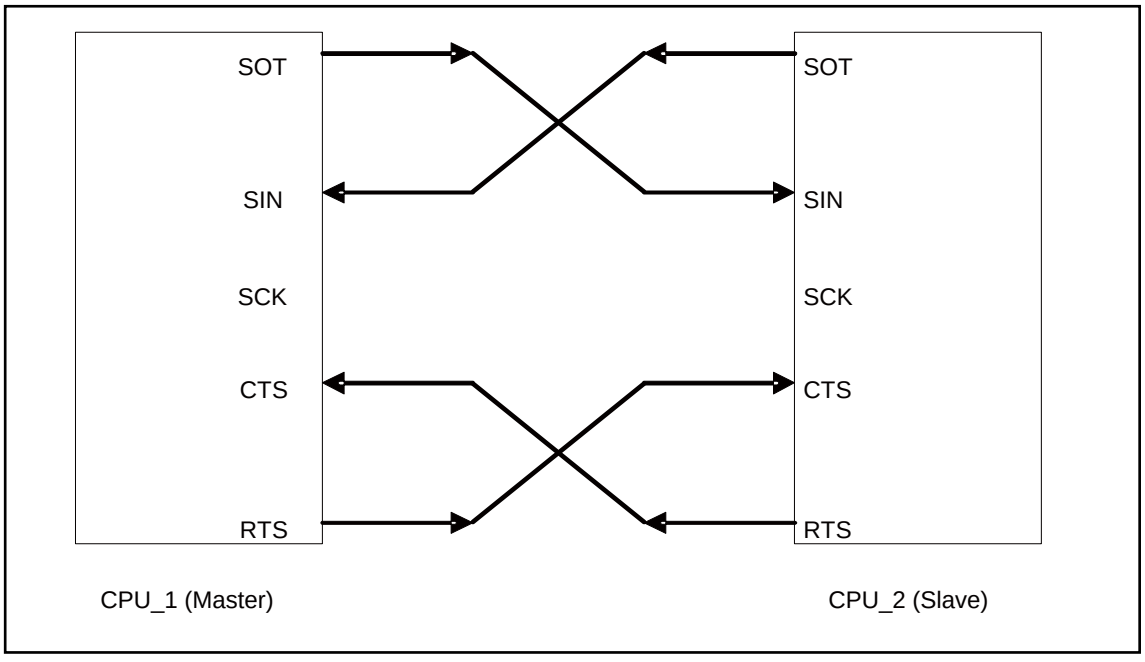


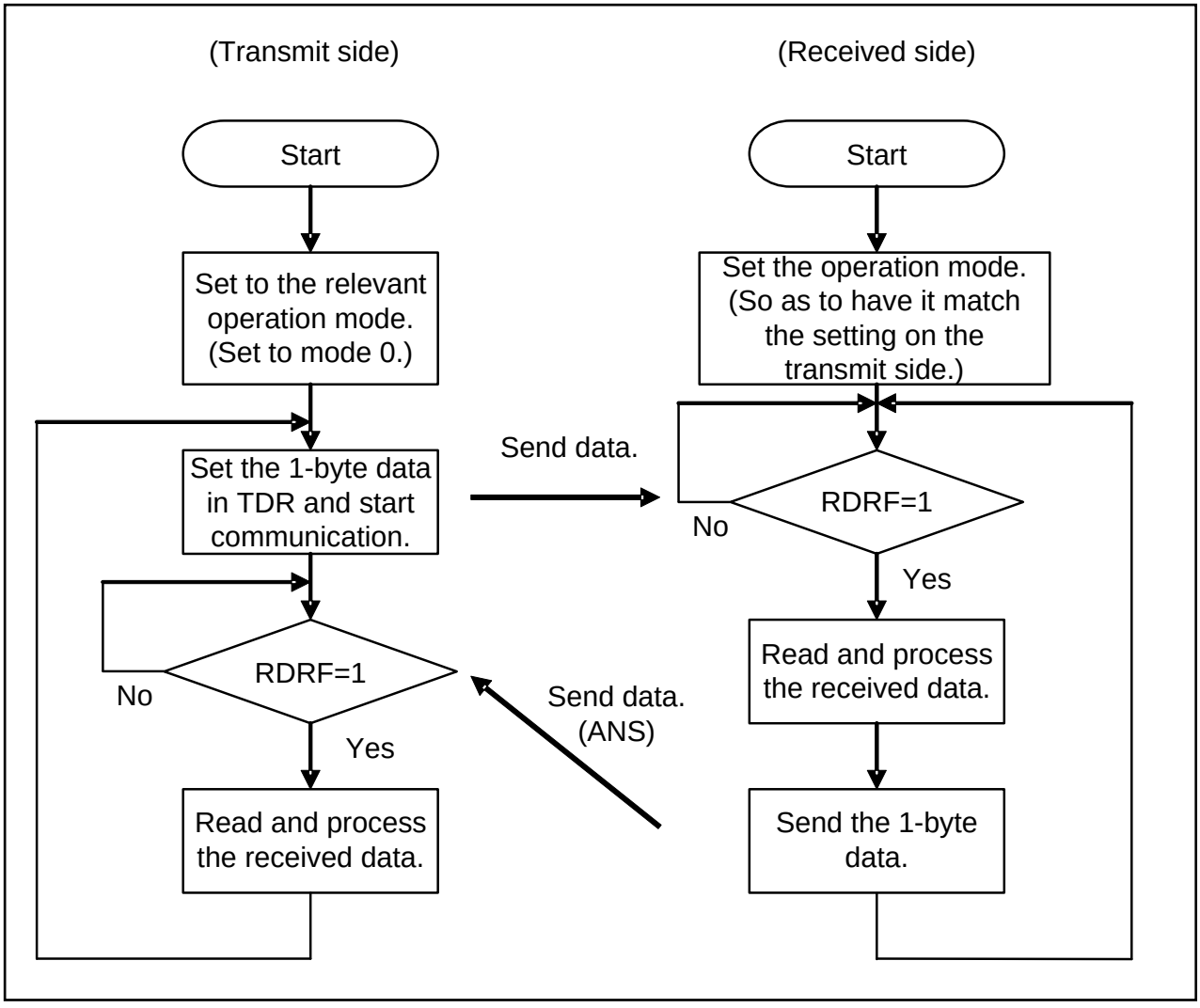
Figure 5-2 A Connection Example of Bi-directional Communications in UART Operation Mode 0 (with Flow Control)



Flowcharts

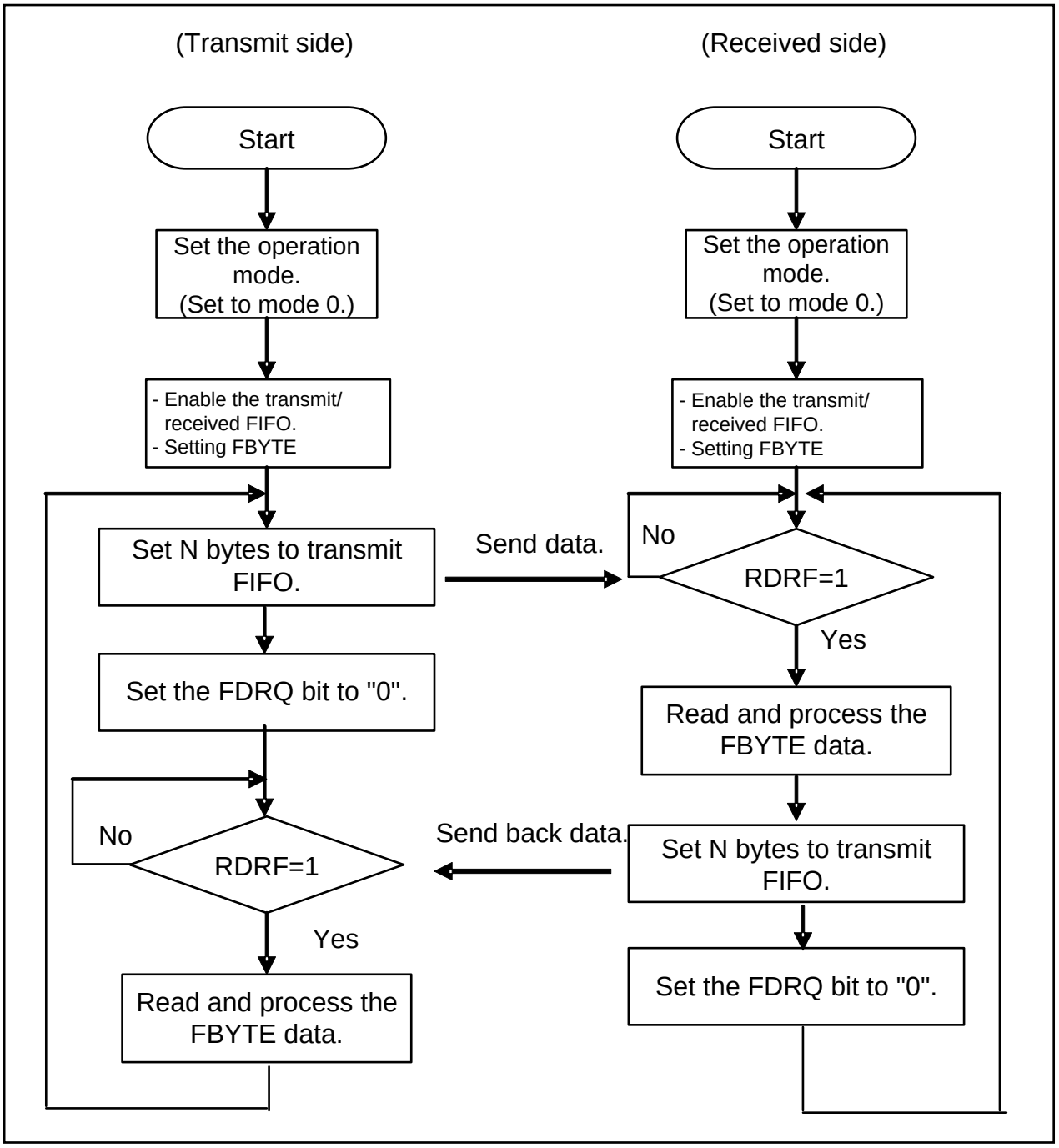
- If FIFO is not used

Figure 5-3 An Example of Bi-directional Communication Flowchart (if FIFO is not Used)



■ If FIFO is used

Figure 5-4 An Example of Bi-directional Communication Flowchart (if FIFO is Used)



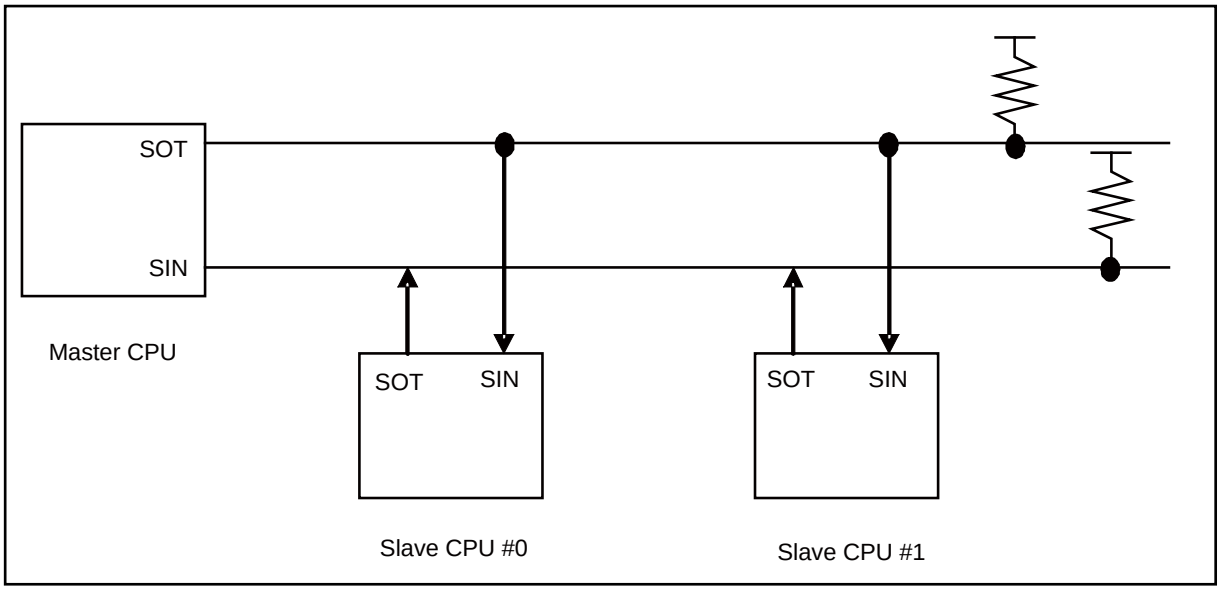
6. Setting Procedure and Program Flow in Operation Mode 1 (Asynchronous Multiprocessor Mode)

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

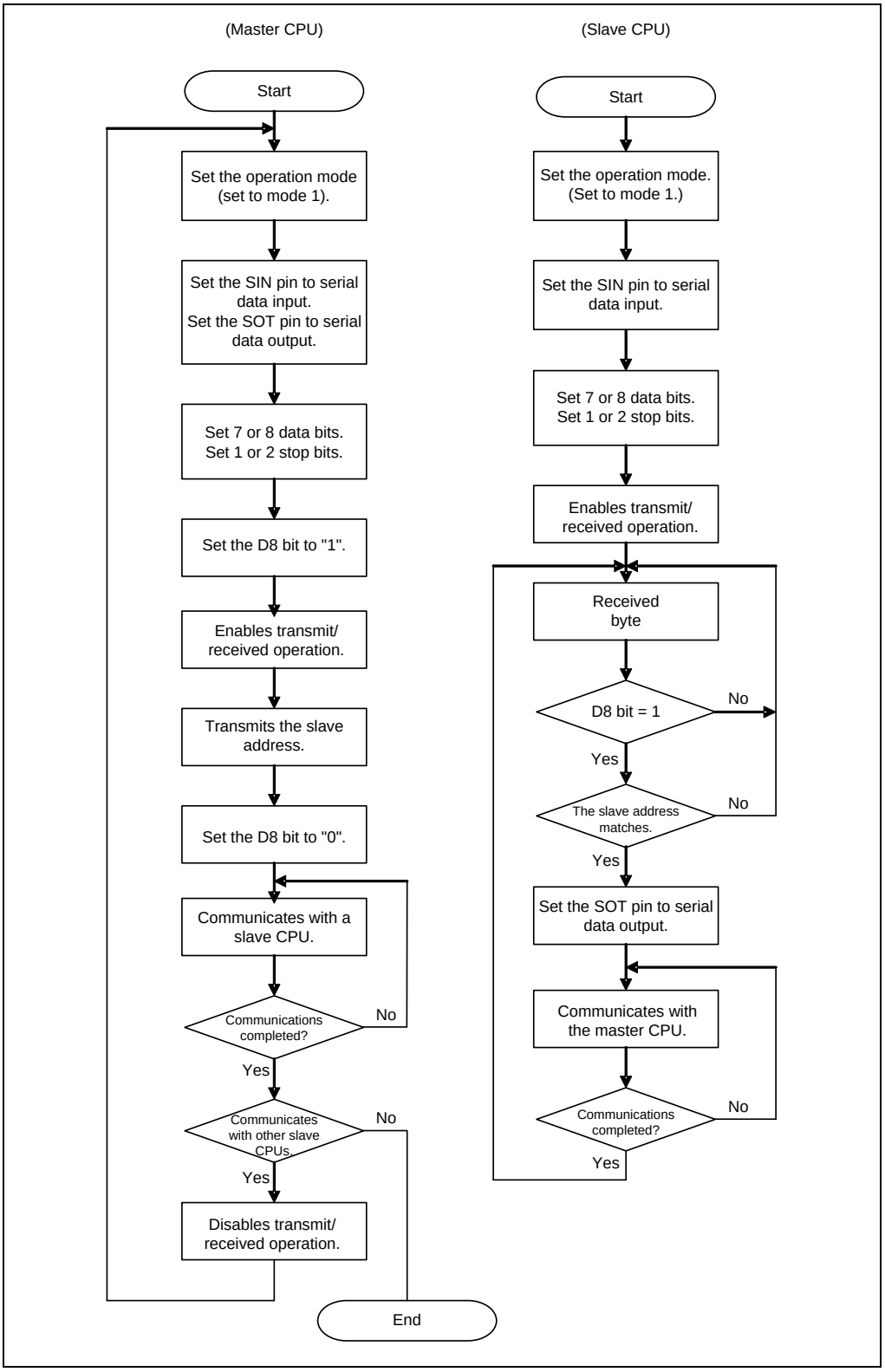
Communications start when the master CPU transmits address data. Address data is a data set whose D8 bit is 1, and used for selecting a slave CPU to communicate with. Each slave CPU judges the address as programmed, and communicates with the master CPU if that address matches the assigned address.

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

Flowcharts

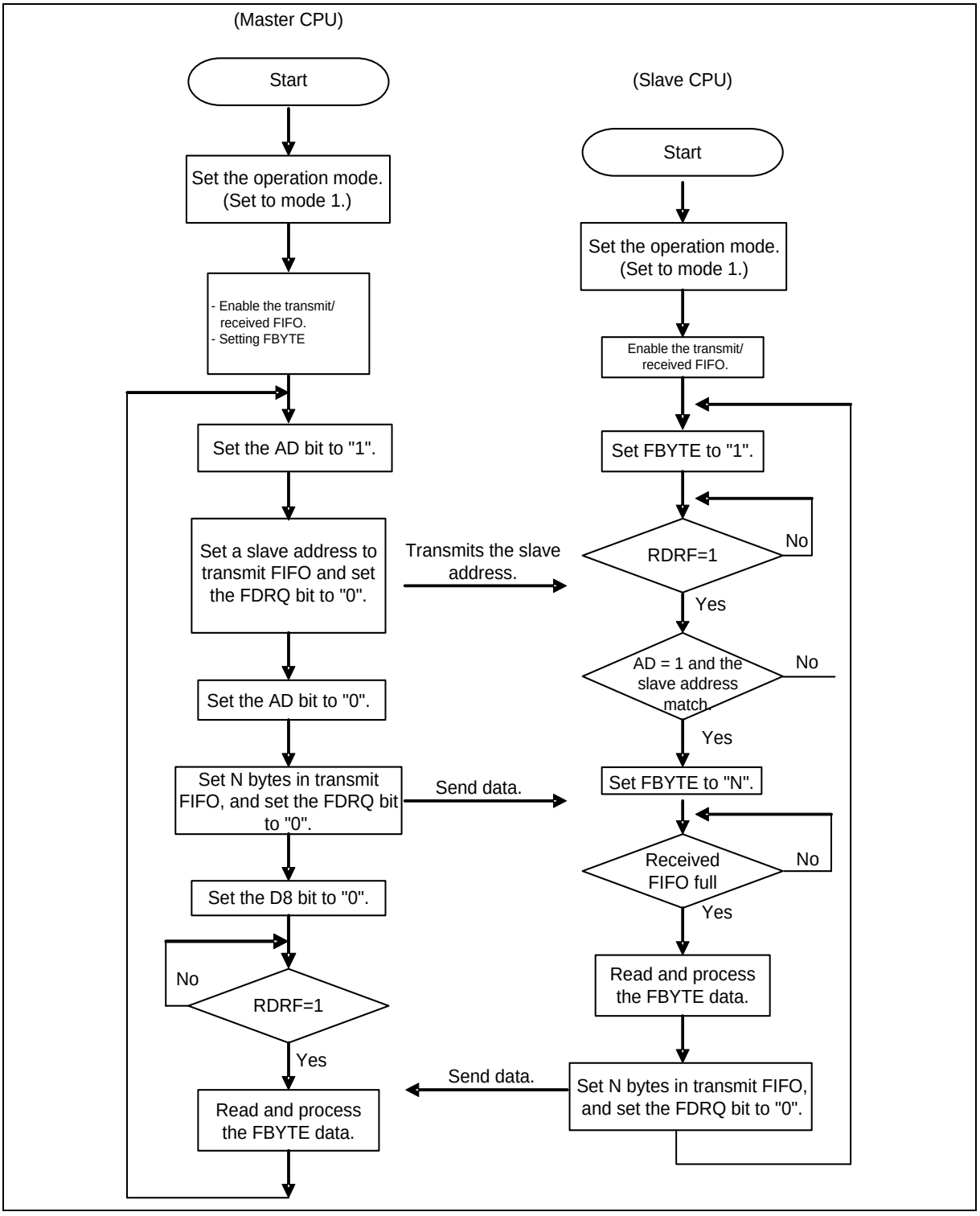
■ If FIFO is not used

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 operation.	"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. The external clock is input from SCK.

[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	Entire 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.*

CHAPTER1-3: 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)
3. CSIO (Clock Synchronous Serial Interface)
4. Serial Timer Operation
5. Serial Chip Select Operation
6. Dedicated Baud Rate
7. CSIO (Clock Synchronous Serial Interface)
8. Restrictions on CSIO (Clock Synchronous Serial Interface)

CODE: 9BFCSIO_FM0-E03.0_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) *1installed.

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	− 4 channels control (single control, Round-Robin 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.

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Operation to Clear Interrupt Request Flag
	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).
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

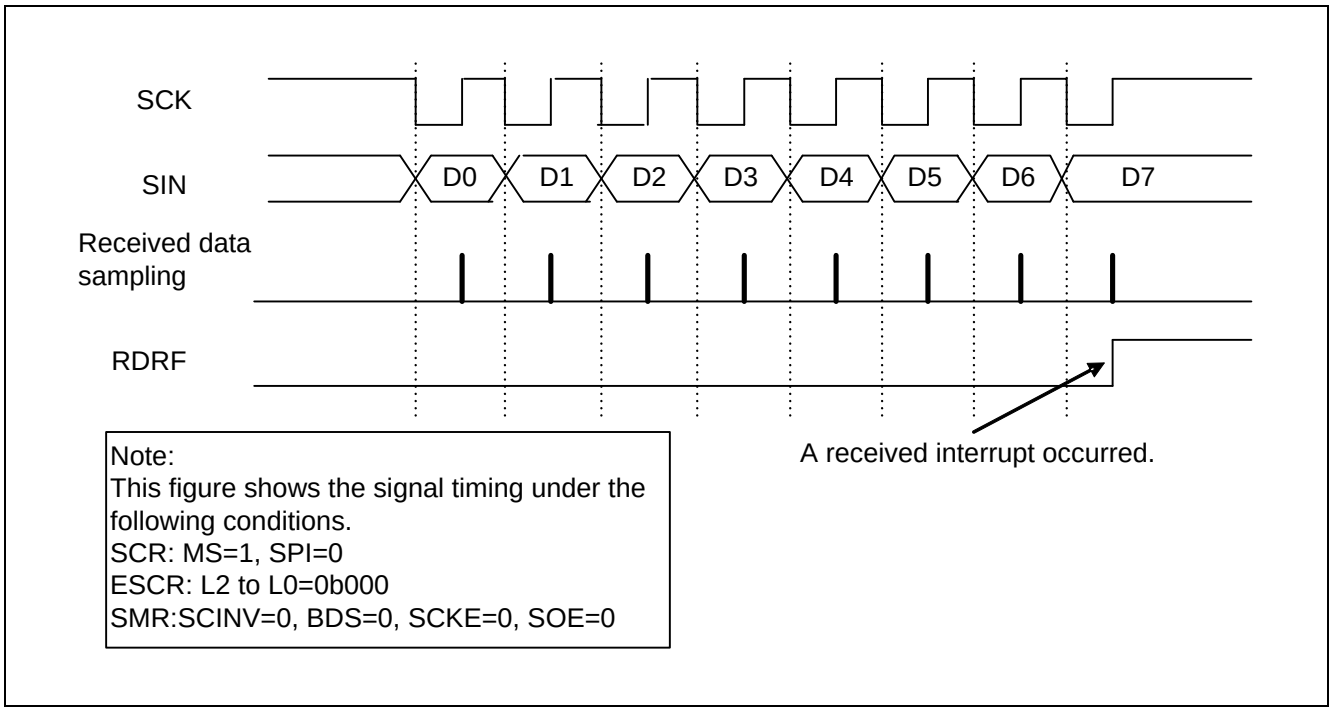
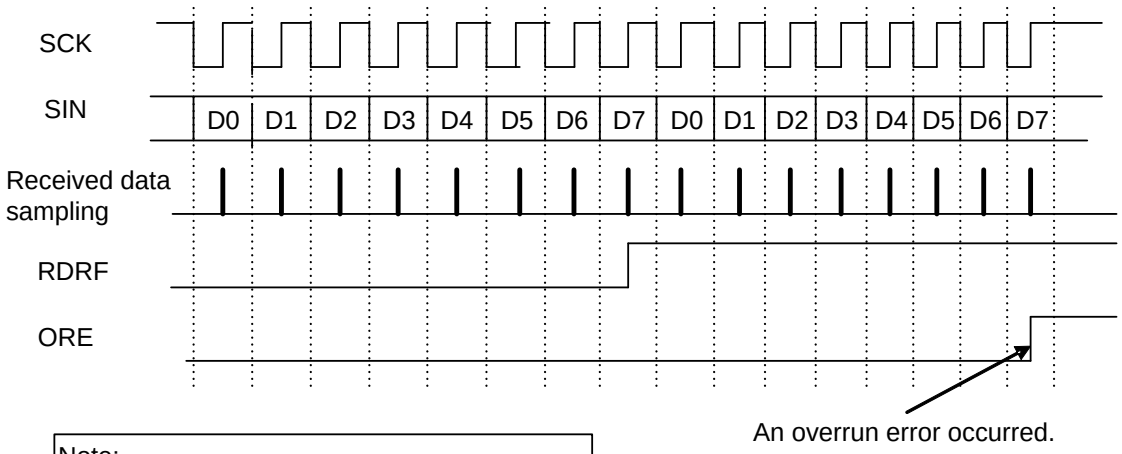


Figure 2-2 ORE (Overrun Error) Flag Set Timing



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

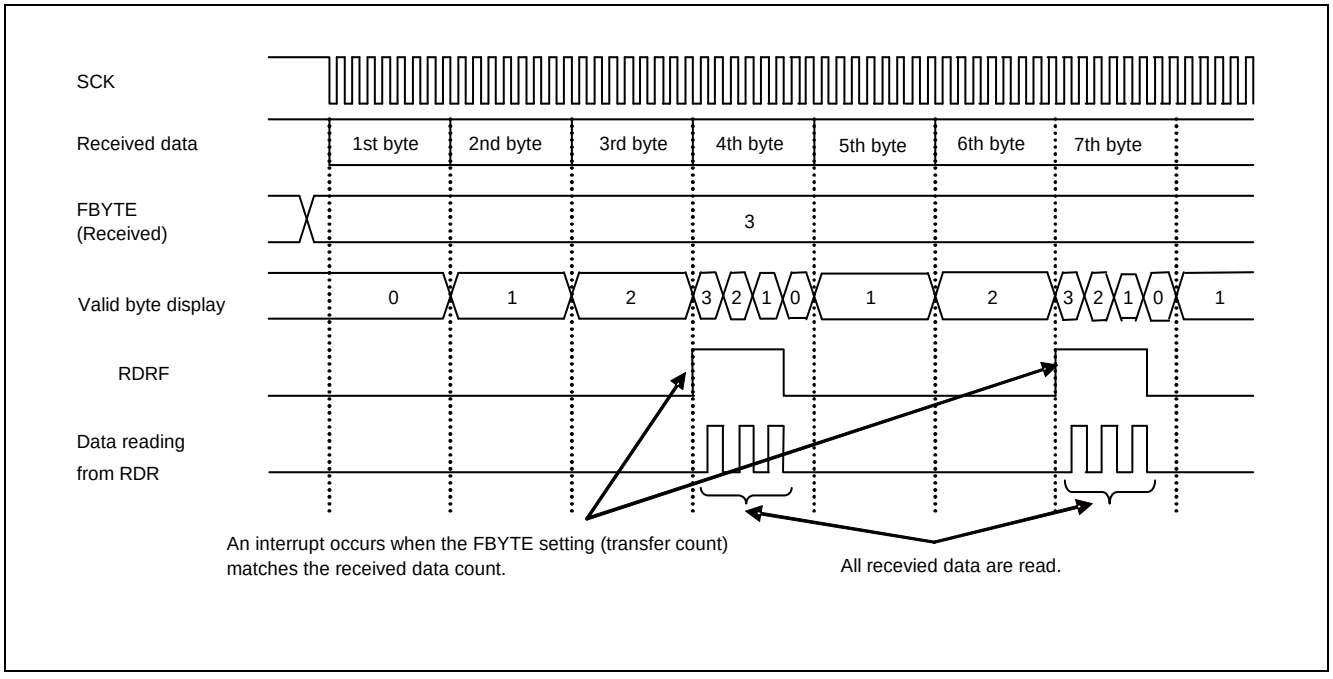
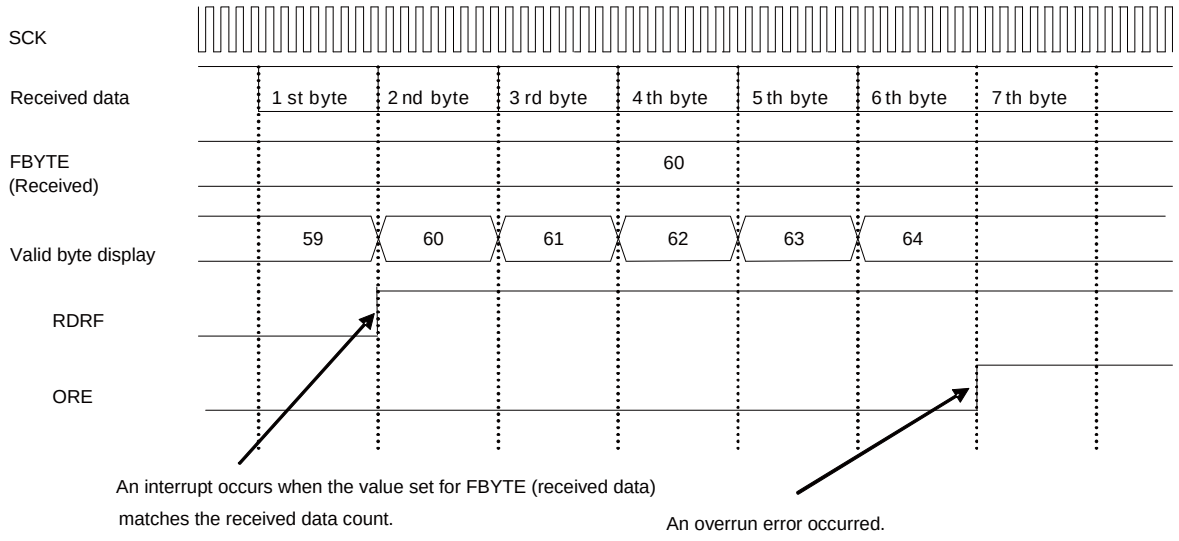


Figure 2-4 ORE (Overrun Error) Flag Bit Set Timing



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

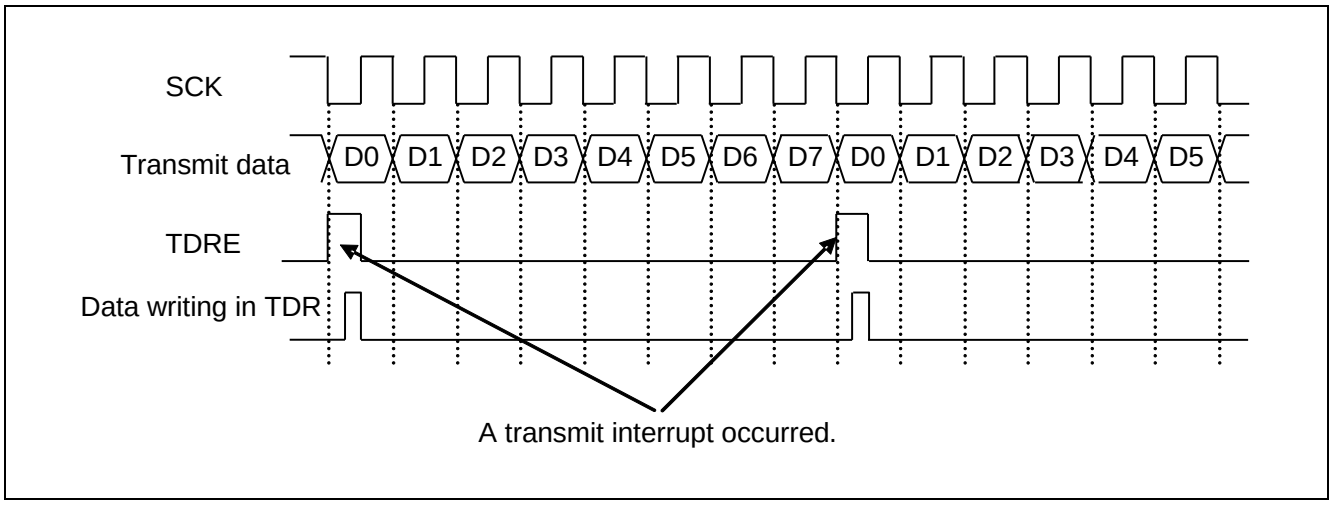
A transmit interrupt occurs if transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE=1) and the data transmission is started, or if no data is transmitted (SSR:TBI=1).

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).

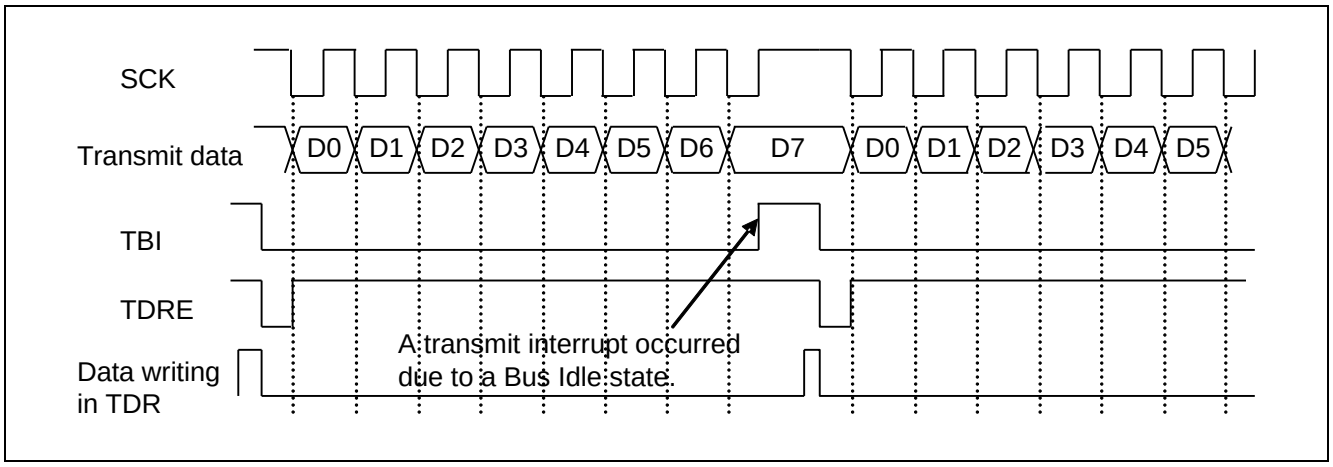
Figure 2-5 Transmit Data Empty Flag (SSR:TDRE) Set Timing



■ 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:CSEN3-0=”0000”, SACSR:TSYNE=0)



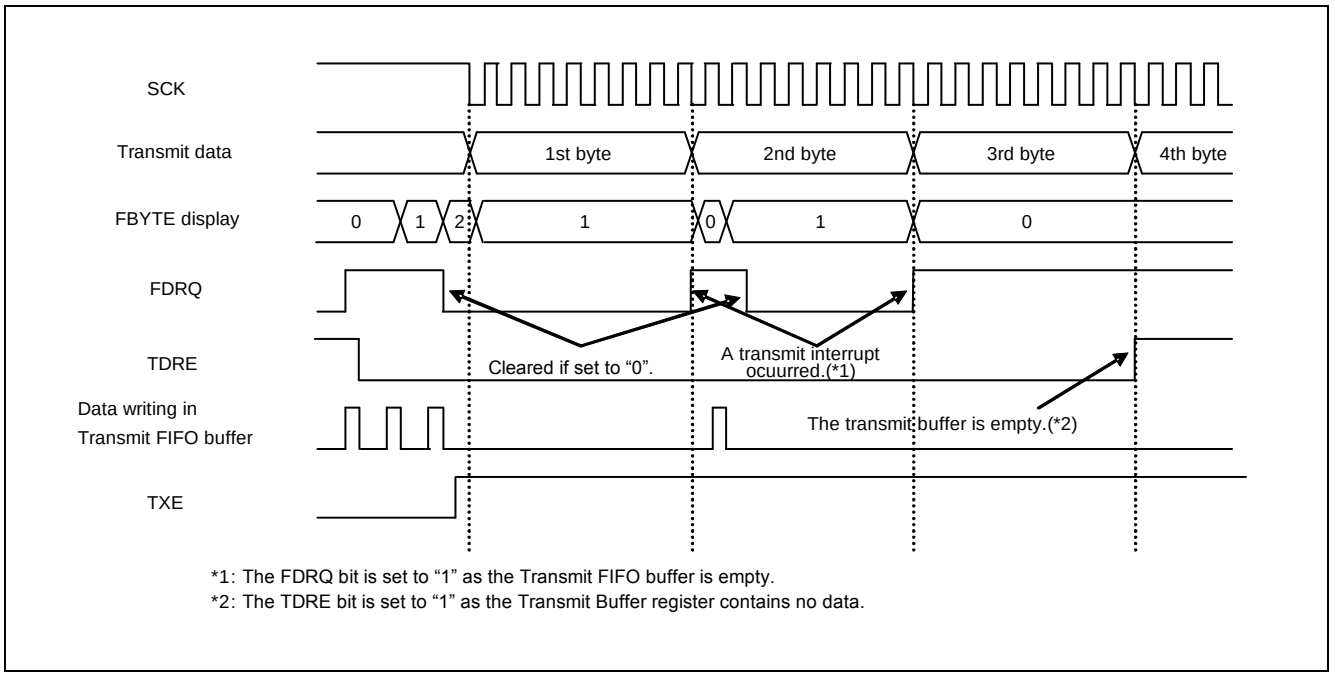
2.4 Interrupt and Flag Set Timing when Transmit FIFO is Used

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.

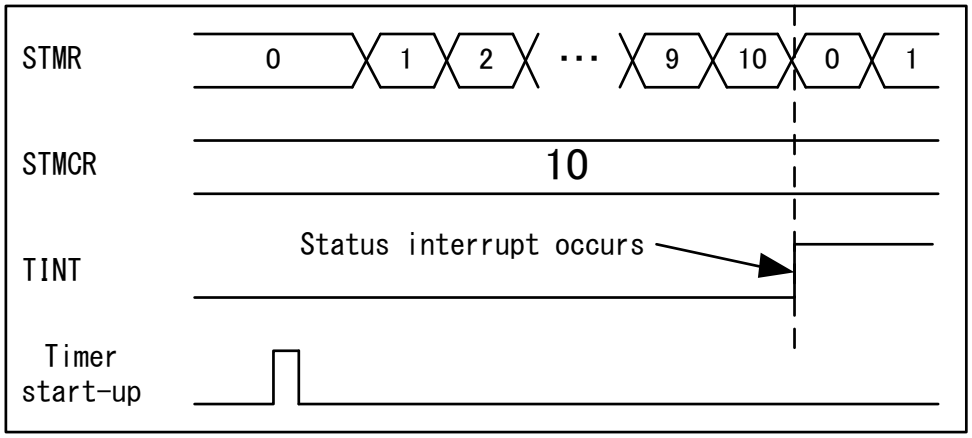
2.5 Timer Interrupt Occurrence and Flag Setting Timing

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

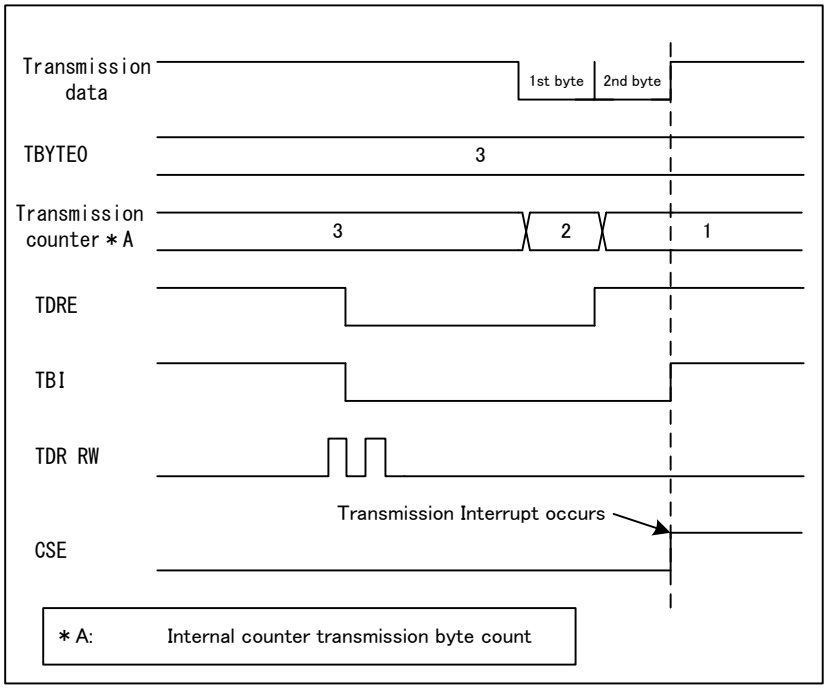
■ Master Mode (SCR:MS=0)

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:CSEN3-0=”0000”, 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)

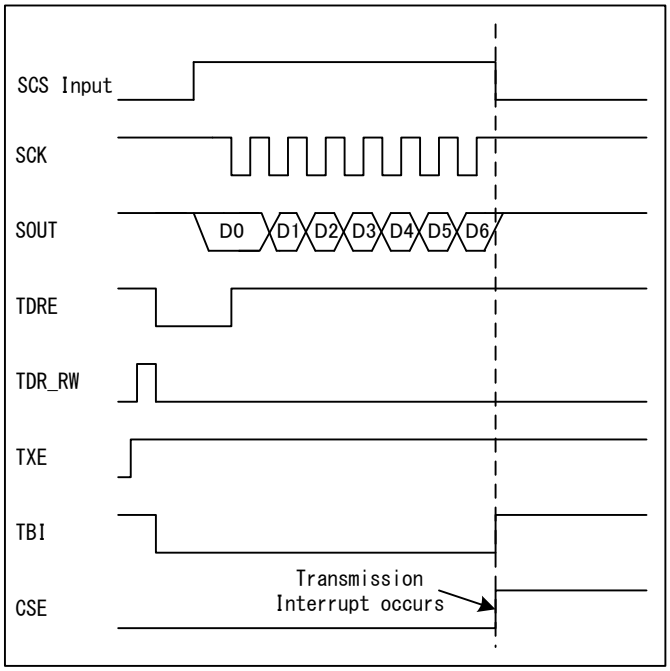
The Chip Select Error occurs when the Serial Chip Select pin becomes inactive at the one of following conditions.

- When serial clock is operating
- When serial clock is changed during not idle state* of transmission module

*: Not idle state means that transmission data is ready and transmission will start by input serial clock.

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)



Note:

- *When Chip Select error(SACSR:CSE=1) occurs at the state that Transmit Data register(TDR) is empty(SSR:TDRE=1), SSR:TBI bit becomes "1" within the period of baud rate.*

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	*	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	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1".

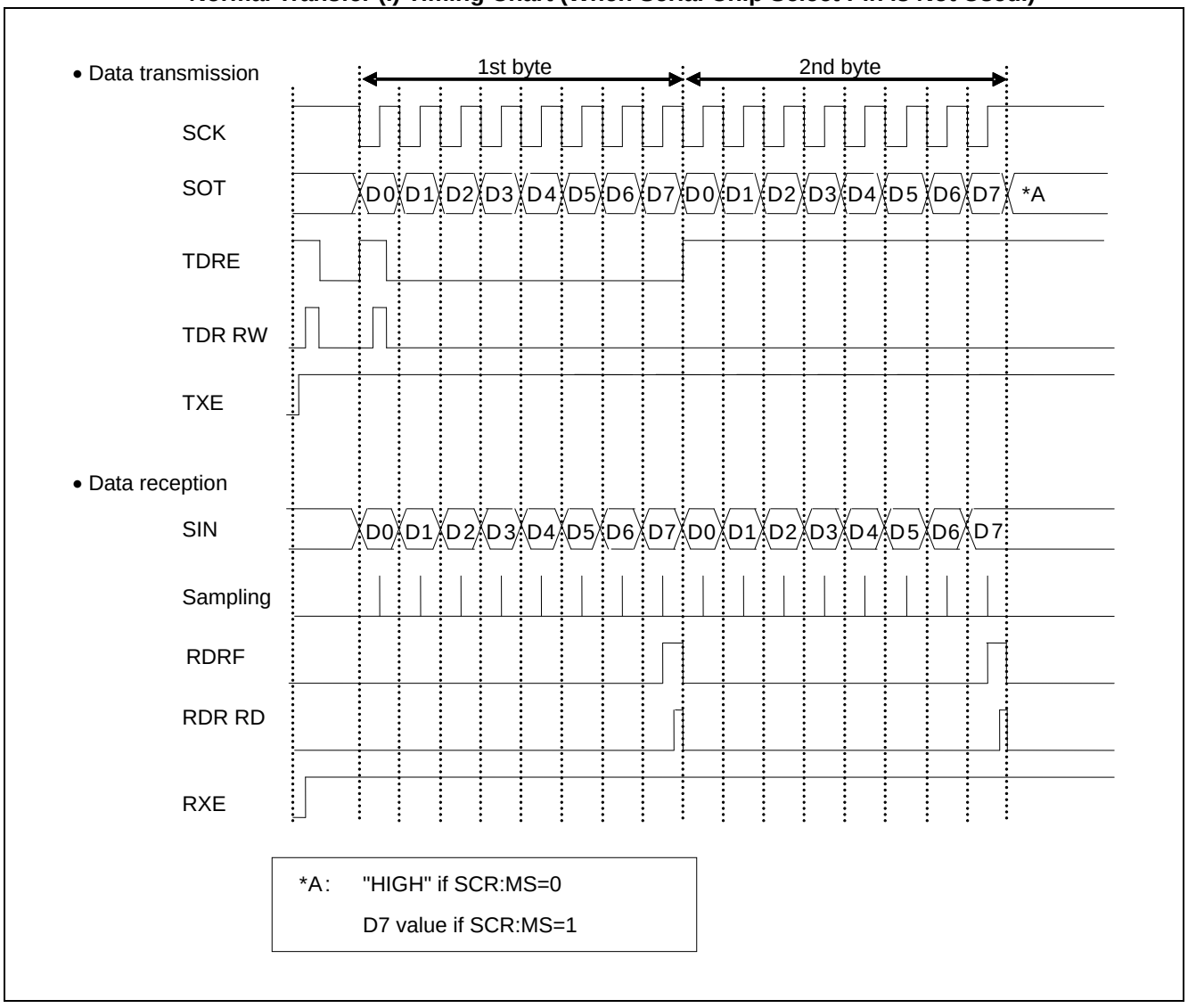
0: Set to "0".

*: 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, SCSCR:CSEN3-0=”0000”)

■ 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.*

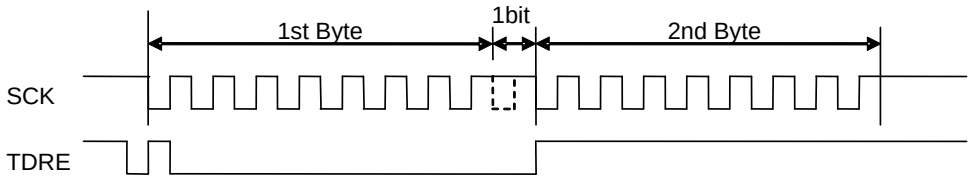
■ Data transmission and reception

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".

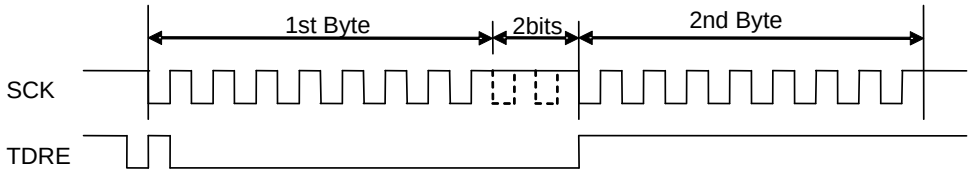
■ Continuous data transmit or reception waiting

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

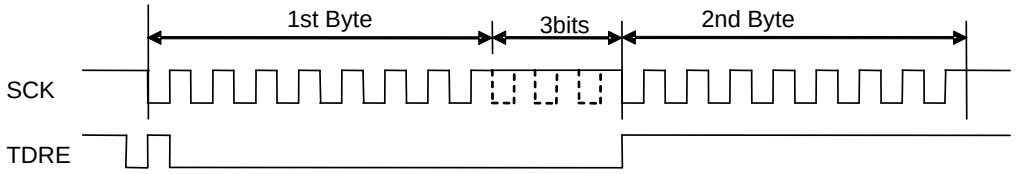
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- 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, SCSCR:CSEN0=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.

Notes:

- *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) and SSR:TBI=1.*

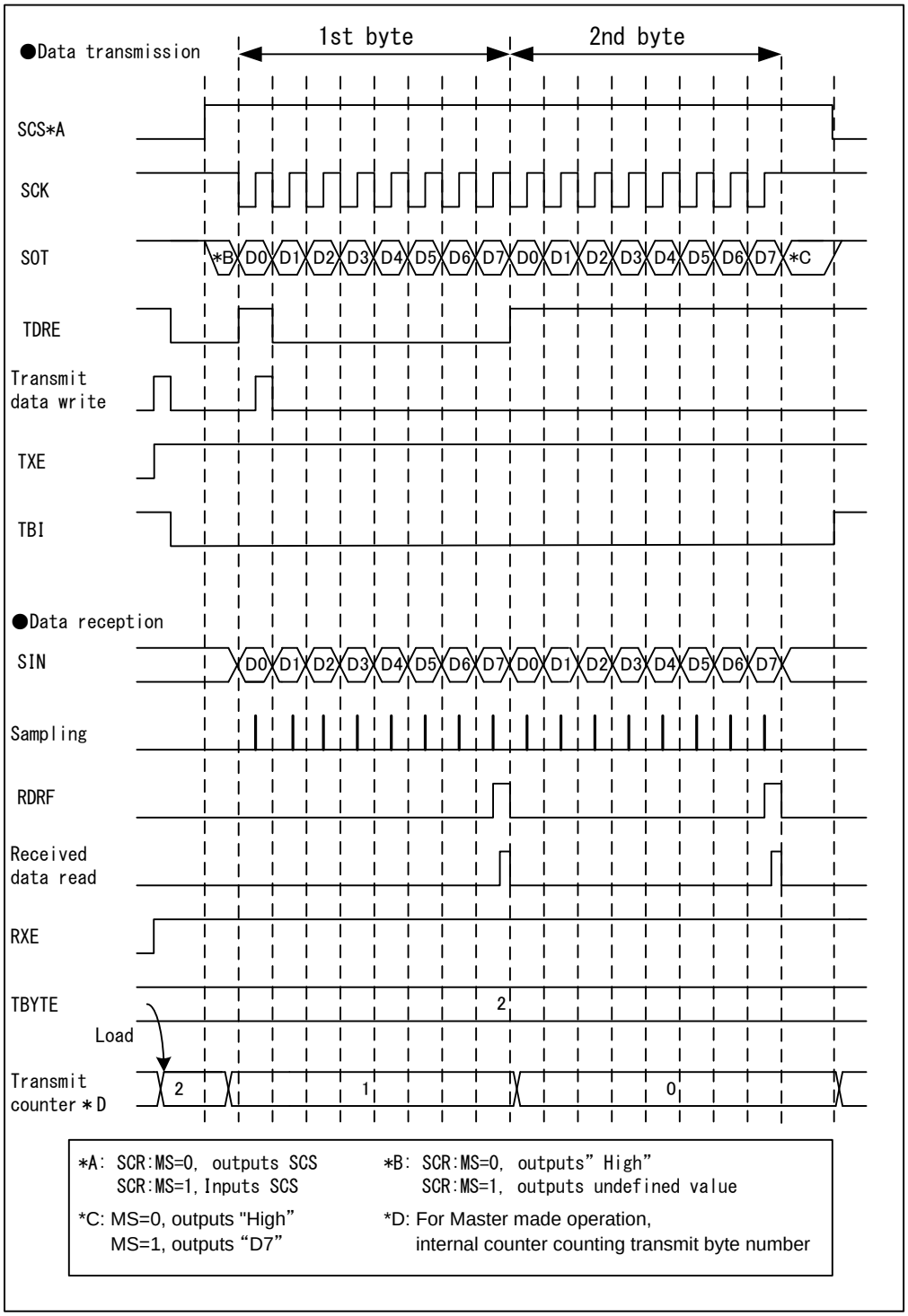
■ Data reception

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).*
- *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.*

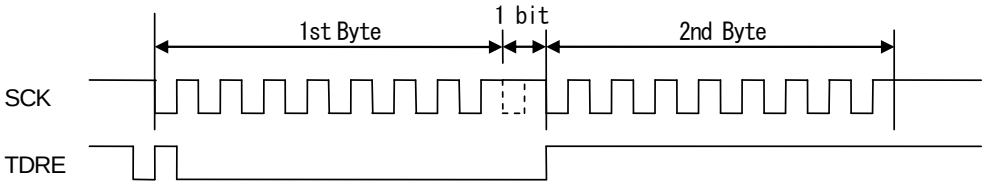
■ Data transmission and reception

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 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.

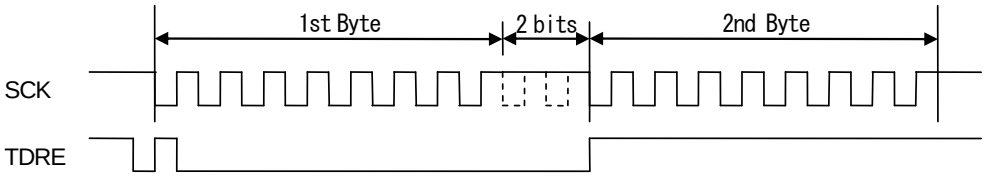
■ Continuous Data Transmit or Reception Waiting

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

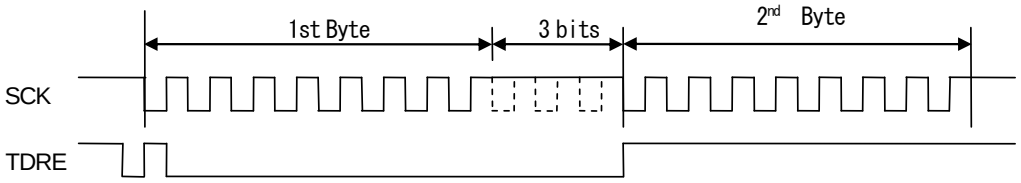
■ ESCR:WT1=0, ESCR:WT0=1(in master mode operation)



■ 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”.

Notes:

- *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) and SSR:TBI=1.*

■ 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	-	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	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1".

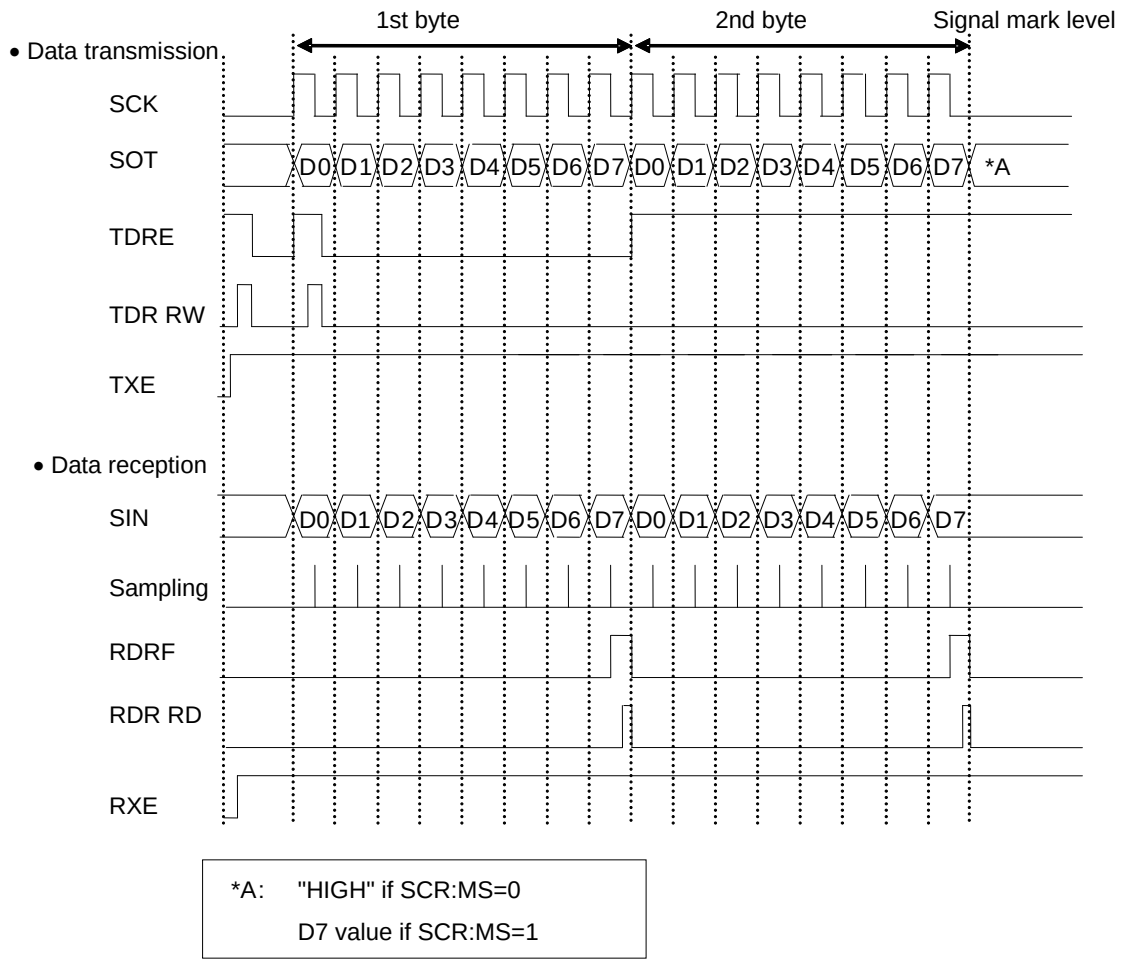
0: Set to "0".

*: 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, SCSCR:CSEN3-0=”0000”)

■ 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.*

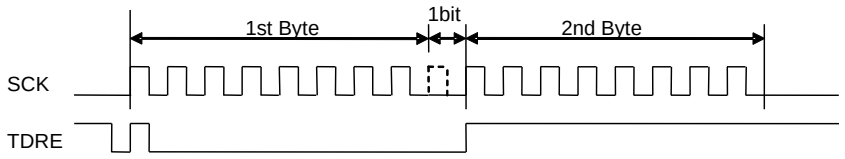
■ Data transmission and reception

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".

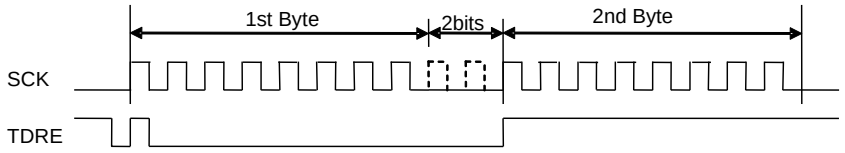
■ Continuous data transmit or reception waiting

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

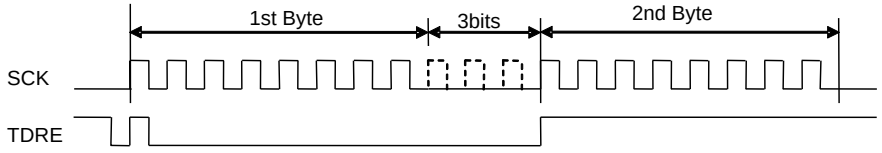
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- 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, SCSCR:CSEN0=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.

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) and SSR:TBI=1.*

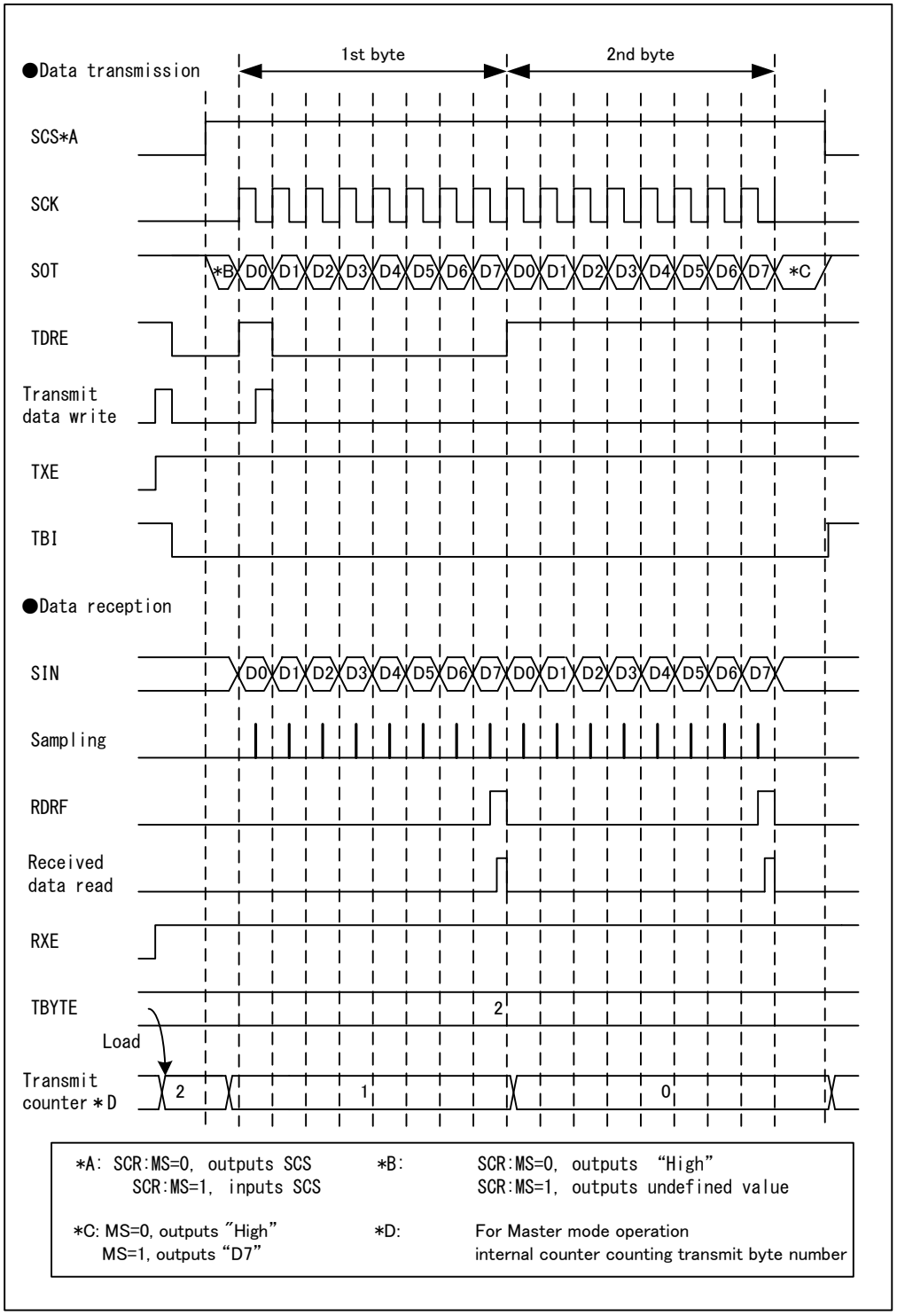
■ Data reception

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.*

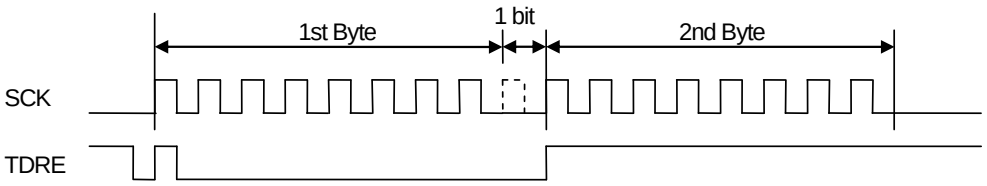
■ Data transmission and reception

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.

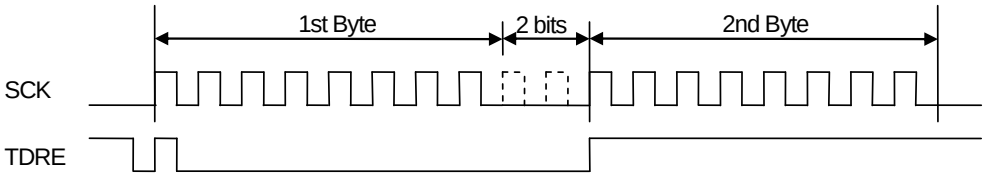
■ Continuous data transmit or reception waiting

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

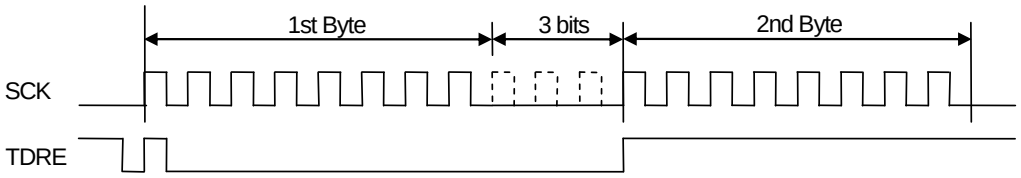
■ ESCR.WT1=0, ESCR.WT0=1(in master mode operation)



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



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



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”.

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) and SSR:TBI=1.*

■ 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	*	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	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1".

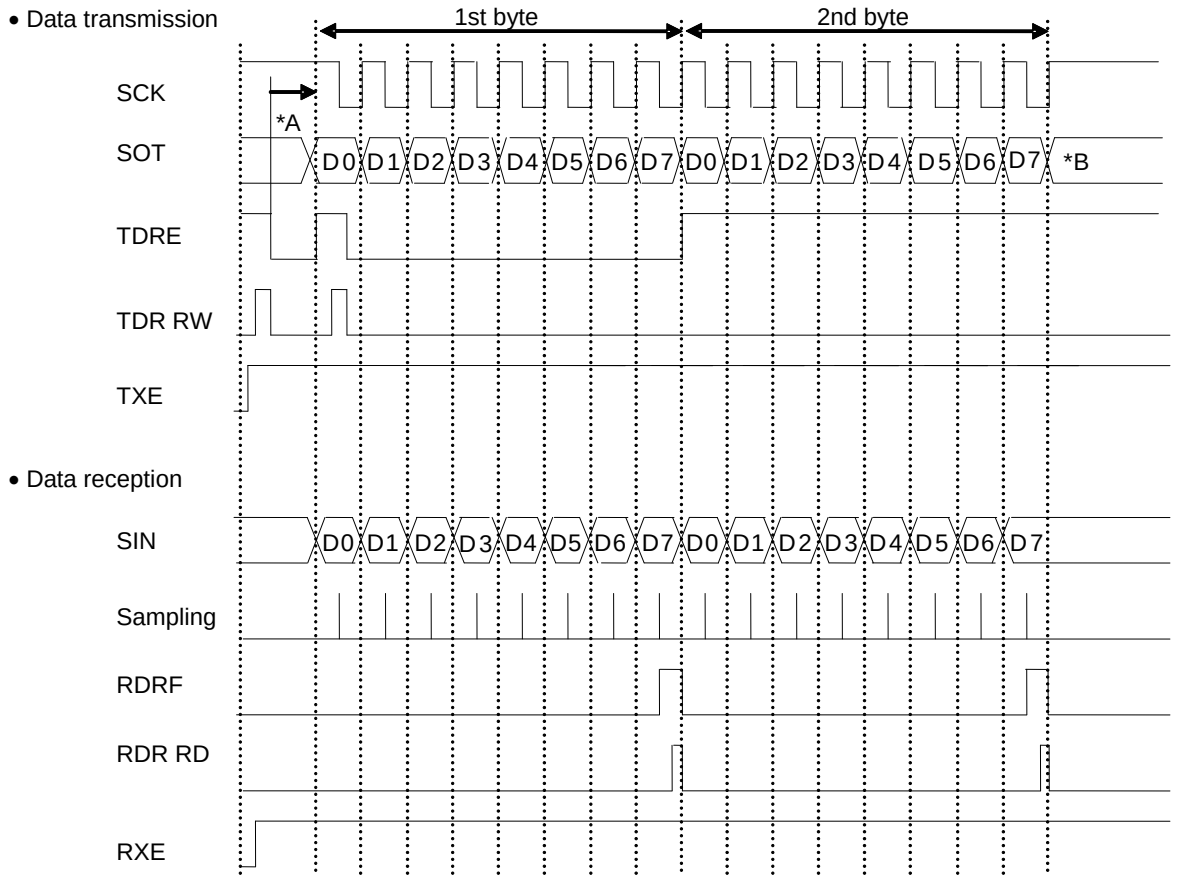
0: Set to "0".

*: 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, SCSCR:CSEN3-0=”0000”)

■ 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.*

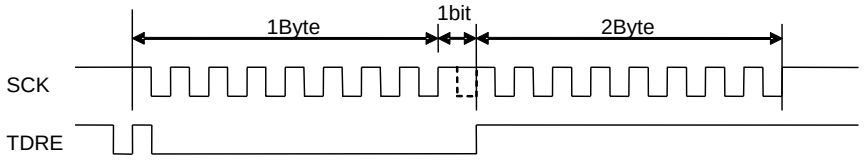
■ Data transmission and reception

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".

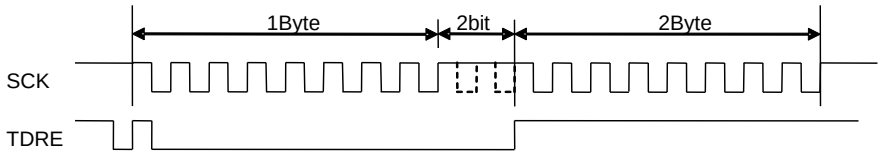
■ Continuous data transmit or reception waiting

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

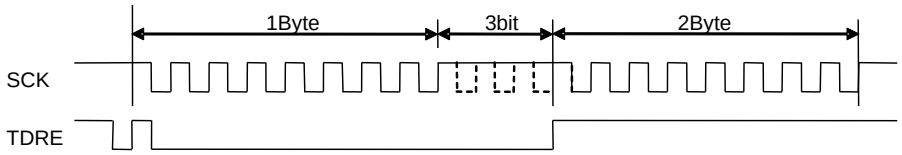
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- 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, SCSCR:CSEN0=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) and SSR:TBI=1.*

■ 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

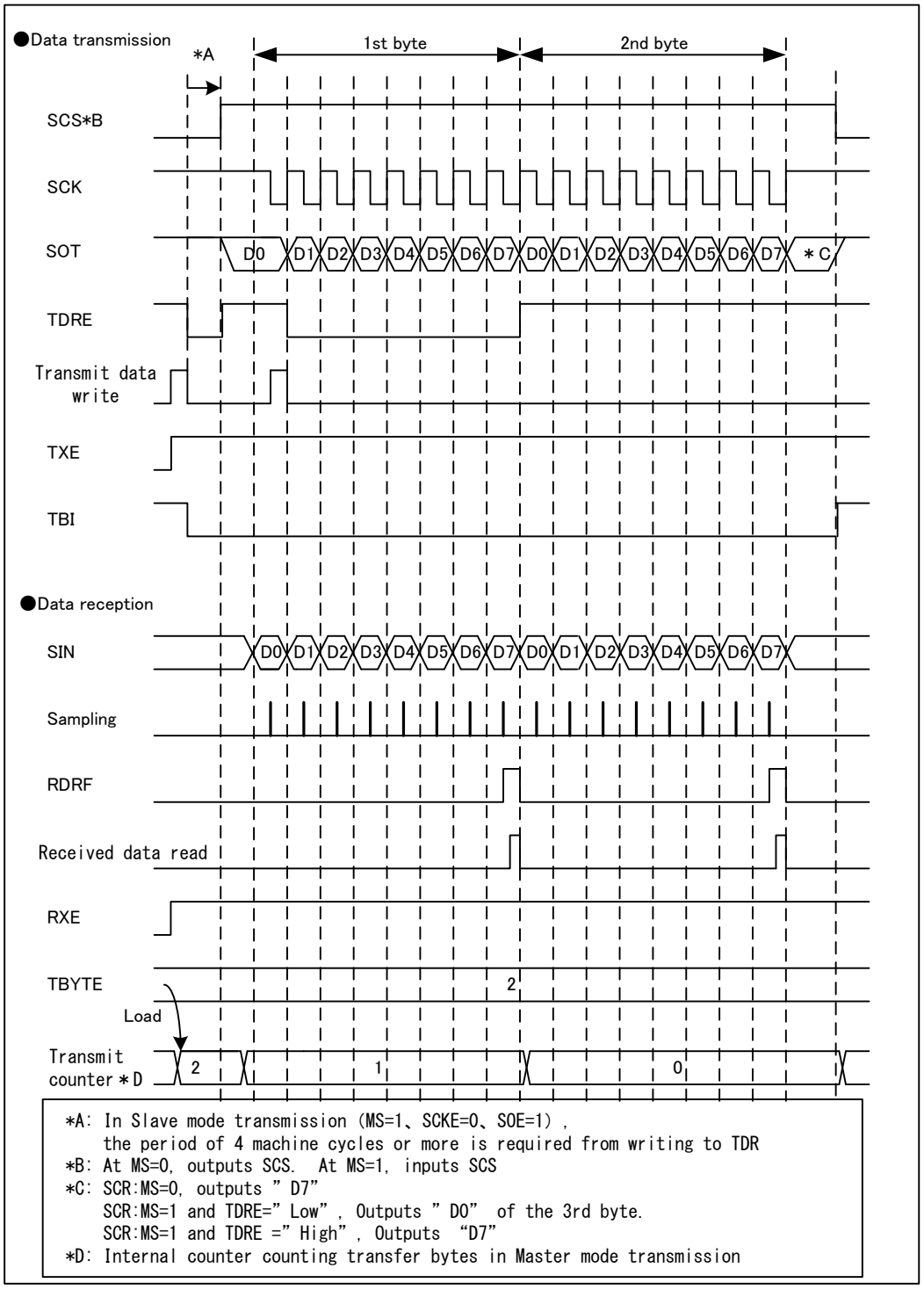
1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is 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.*

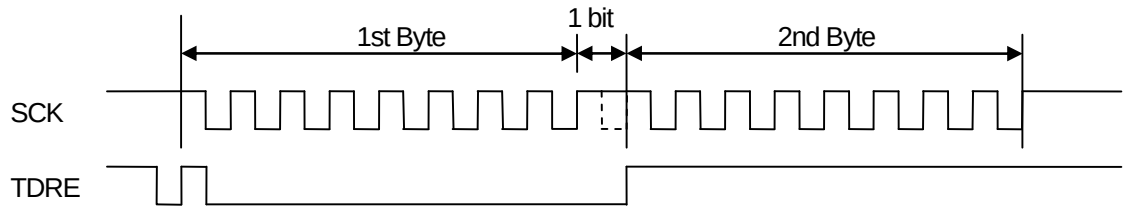
■ Data transmission and reception

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 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.

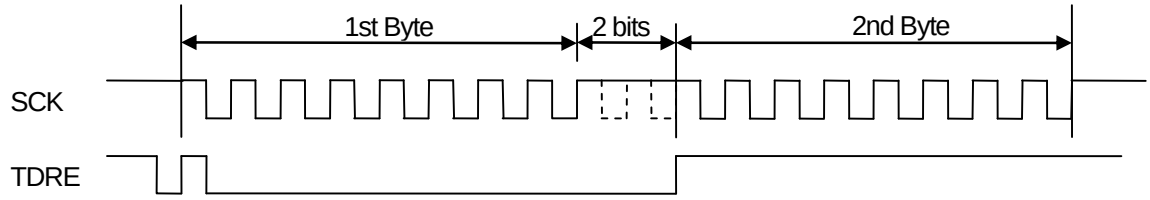
■ Continuous data transmit or reception waiting

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

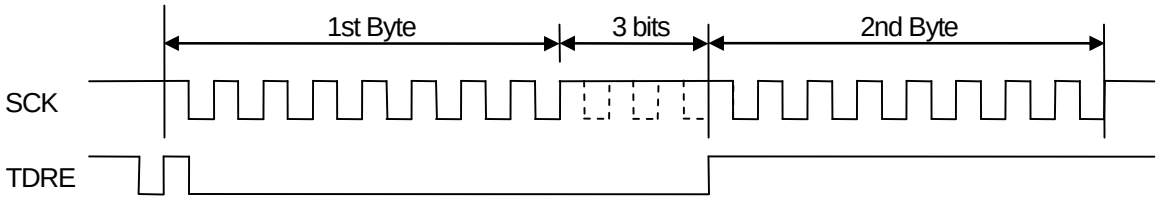
■ ESCR.WT1=0, ESCR.WT0=1(in master mode



■ 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:CSEN0=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 16 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	-	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	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BGR1/	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
BGR0	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1".

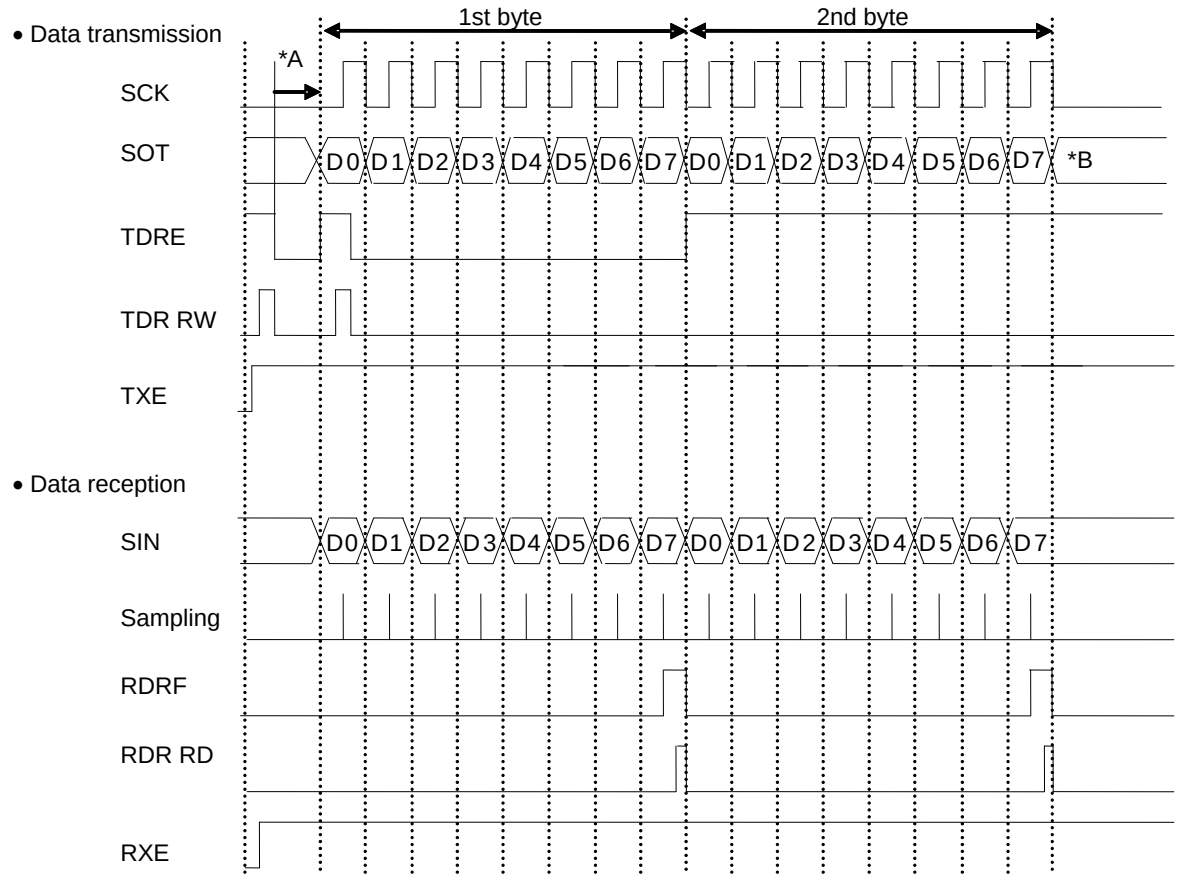
0: Set to "0".

*: 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, SCSCR:CSEN3-0=”0000”)

■ 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.*

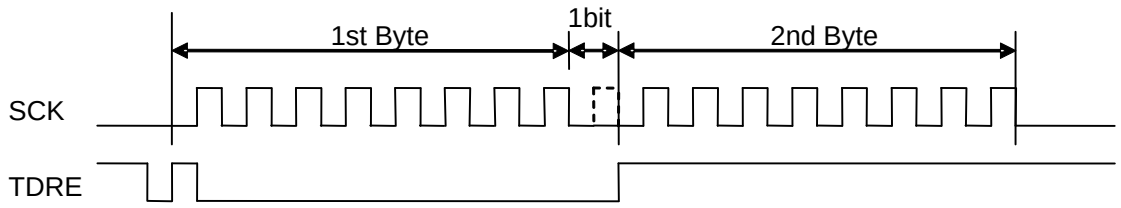
■ Data transmission and reception

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".

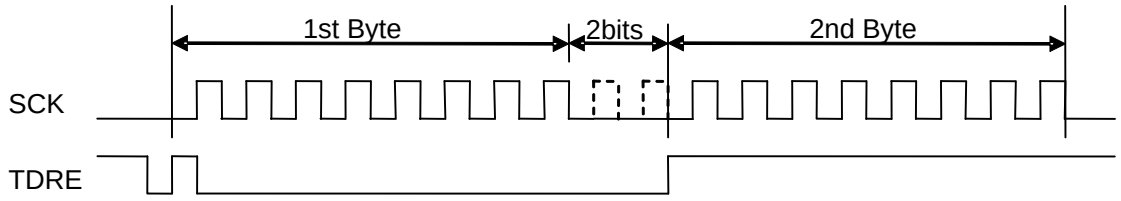
■ Continuous data transmit or reception waiting

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

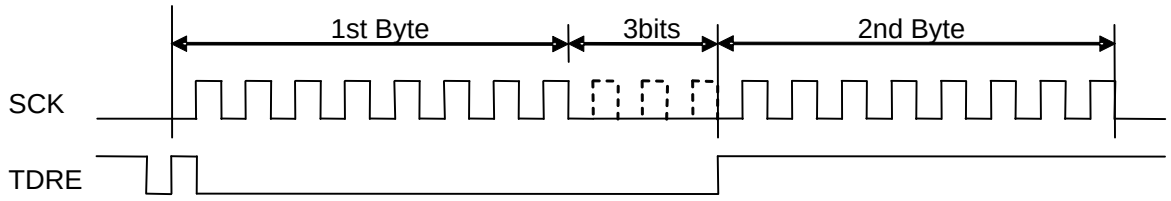
- ESCR:WT1, ESCR:WT0=01 (in master mode operation)



- 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, SCSCR:SCEN0=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) and SSR:TBI=1.*

■ 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

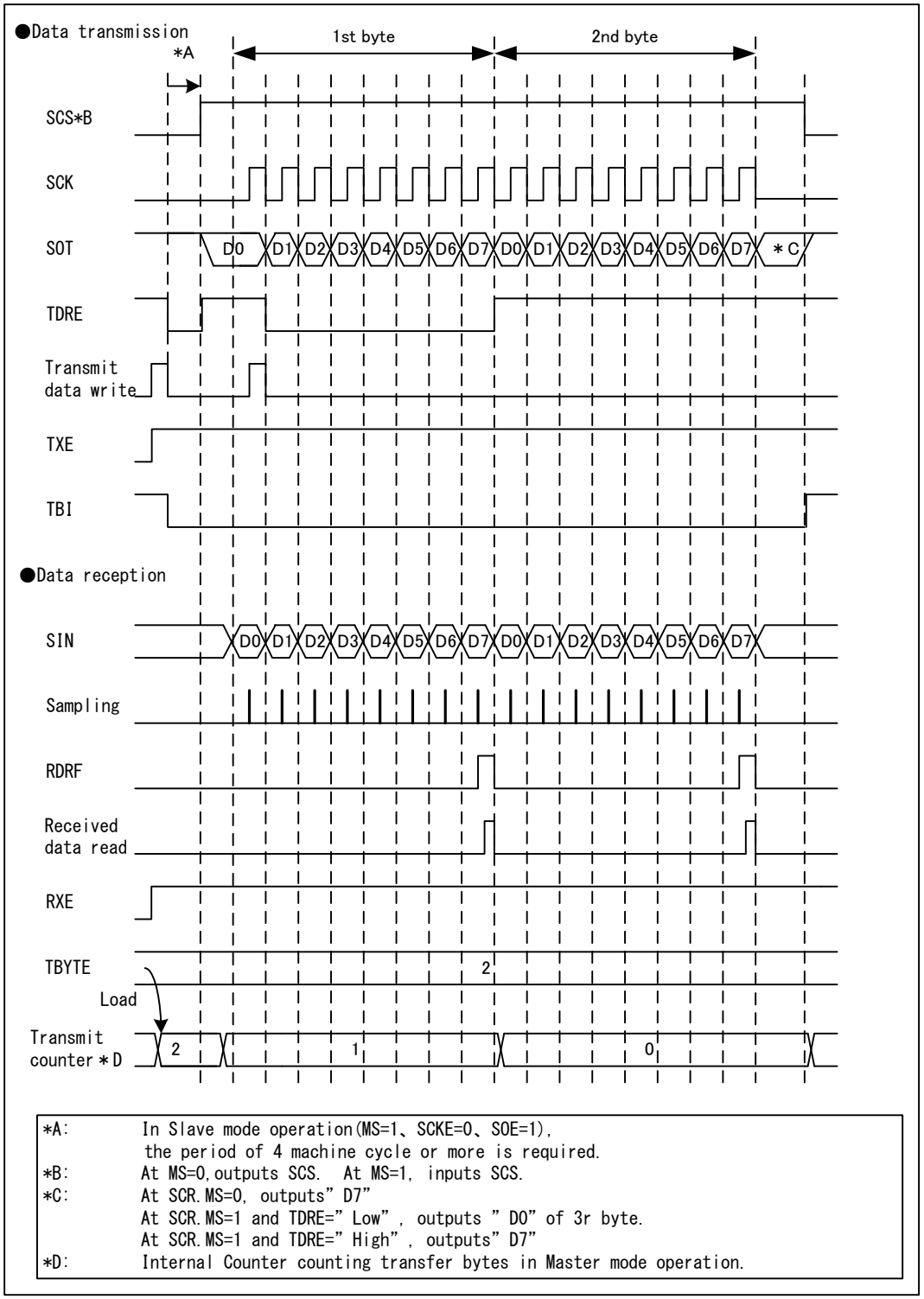
1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).
2. When the transmit data is written in the TDR, the SSR:TDRE is set to "0" and the first bit is output. Then, the transmit data is 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 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

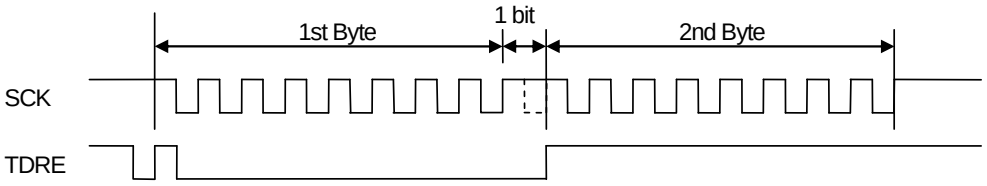
1. To perform data transmission and reception simultaneously, enable the serial data output (SMR:SOE=1) and enable the data transmission and reception (SCR:TXE, RXE=1).

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.

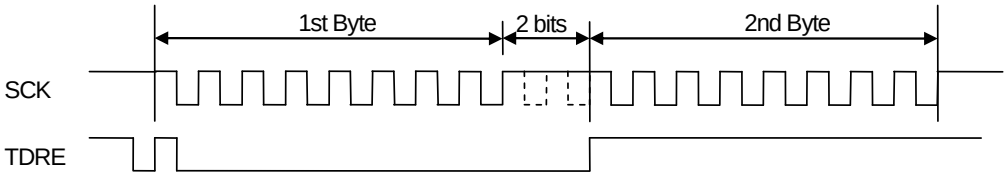
■ Continuous data transmit or reception waiting

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

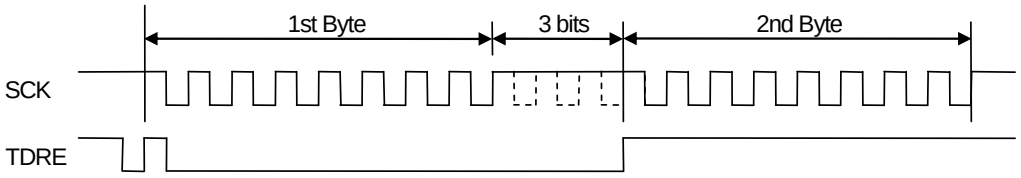
■ ESCR.WT1=0, ESCR.WT0=1(in master mode operation)



■ 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) and SSR:TBI=1.*

■ 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

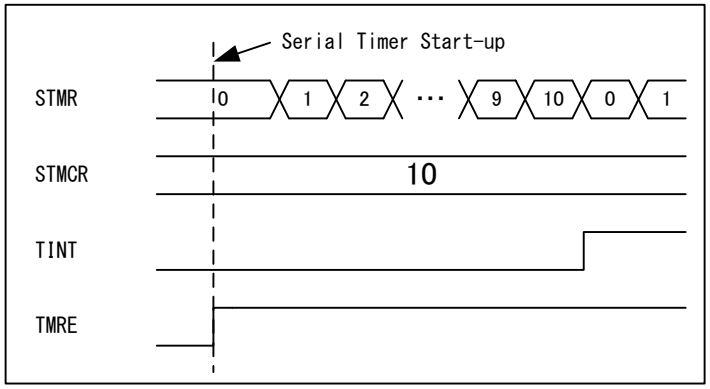
■ Starting method of serial timer

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 a 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)

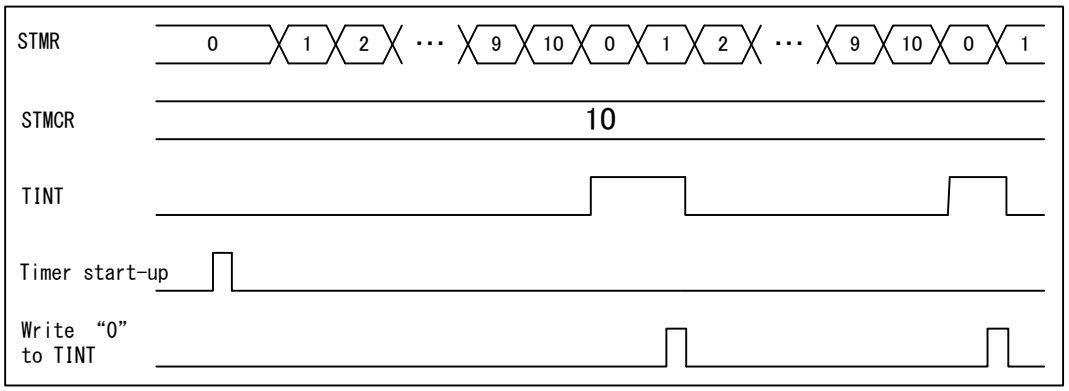


Figure 4-3 Serial Timer Initial Setting Flow Chart

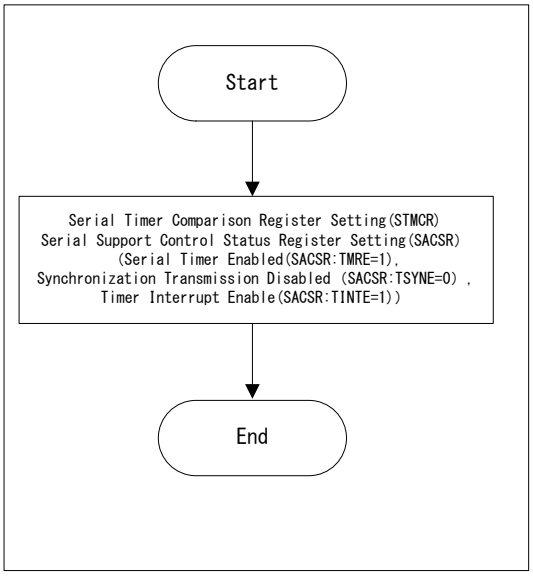
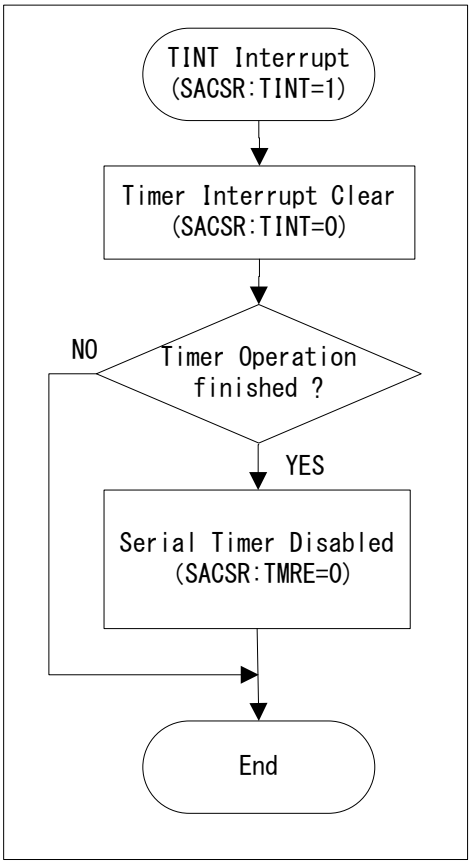


Figure 4-4 Serial Timer Interrupt Process Flow Chart



Notes:

- *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.*

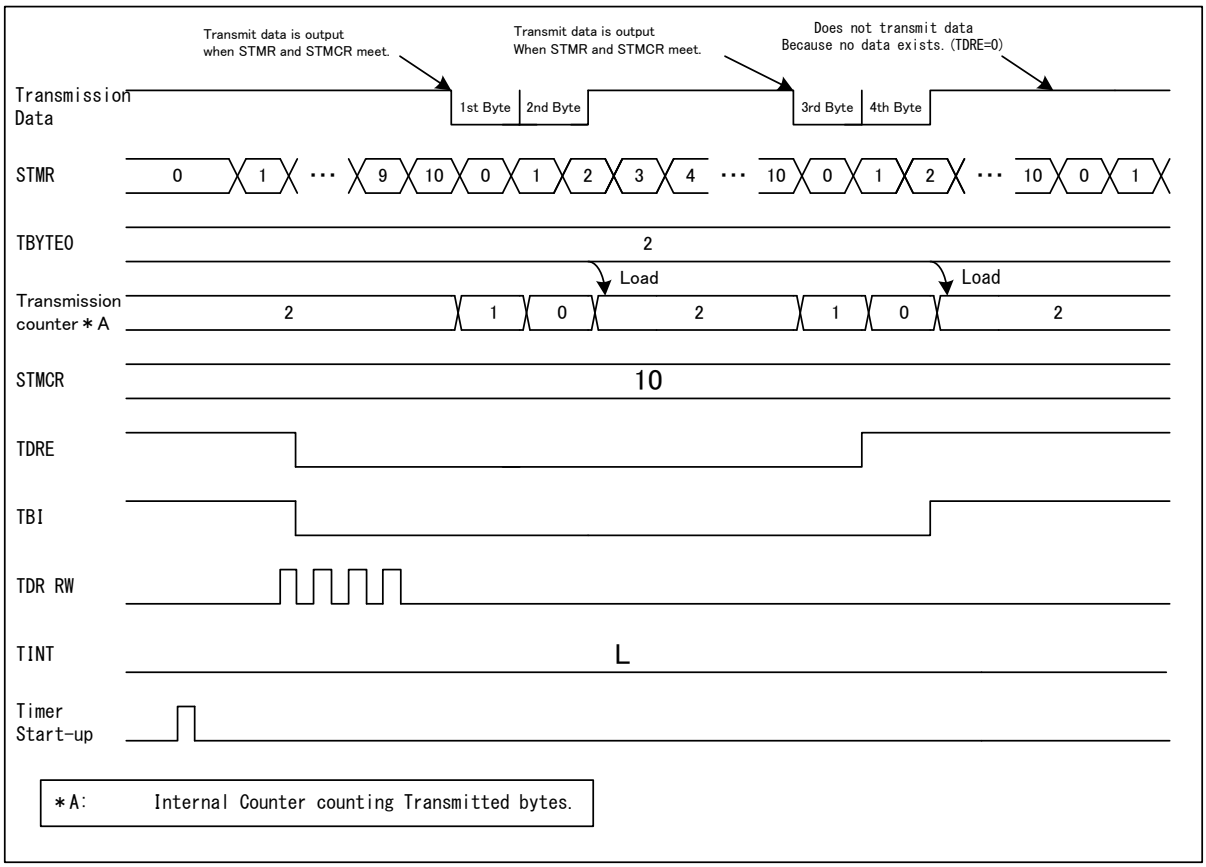
■ Transmission in synchronization with the timer

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

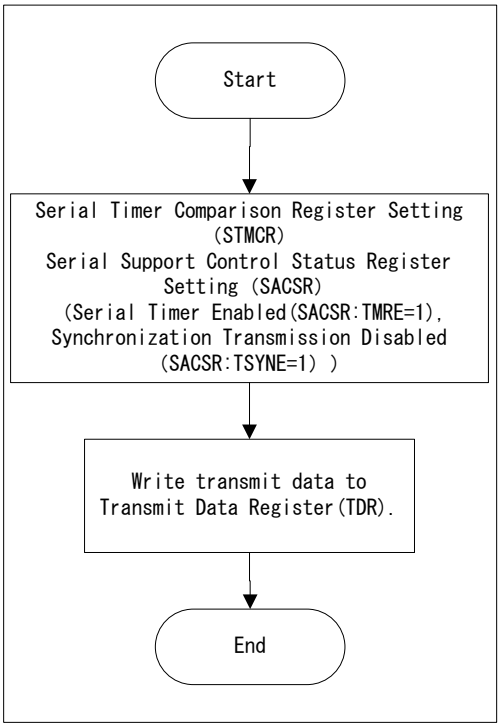
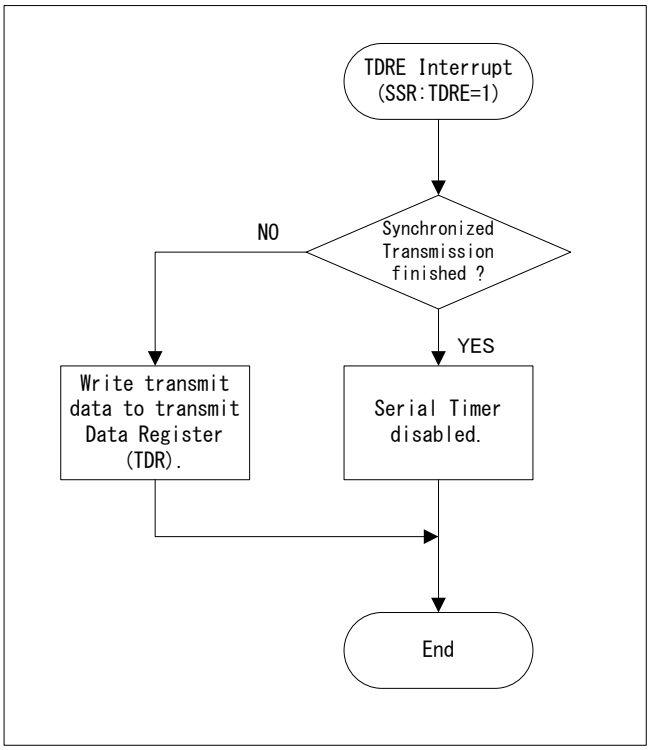


Figure 4-7 Timer Synchronization Transmission Interrupt Handling Flowchart



Notes:

- *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)

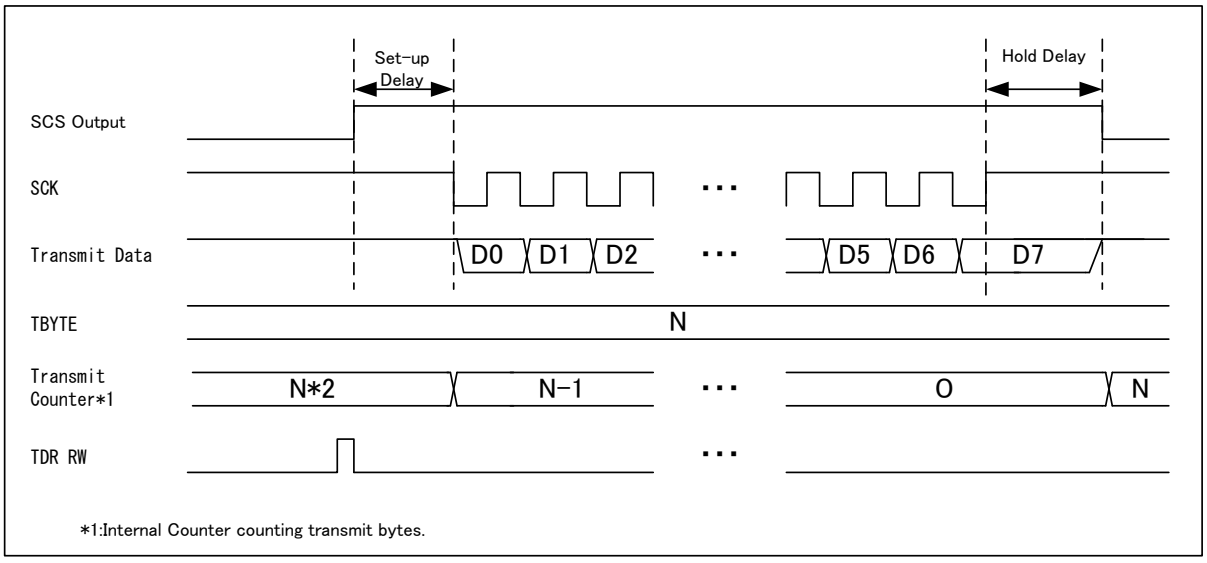
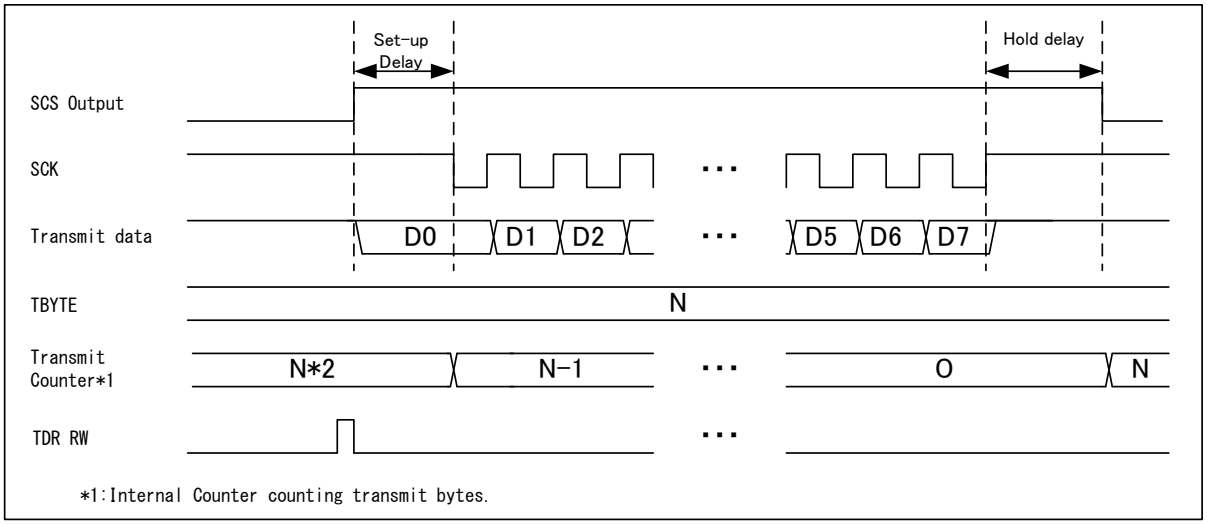


Figure 5-2 Serial Chip Select Operation (Master Transmission (MS=0), SPI Transfer (SPI=1), SCINV=0)



Notes:

- *If the transmission is disabled (SCR:TXE="0") 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:CSEN3-0 is set to "0000" 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)

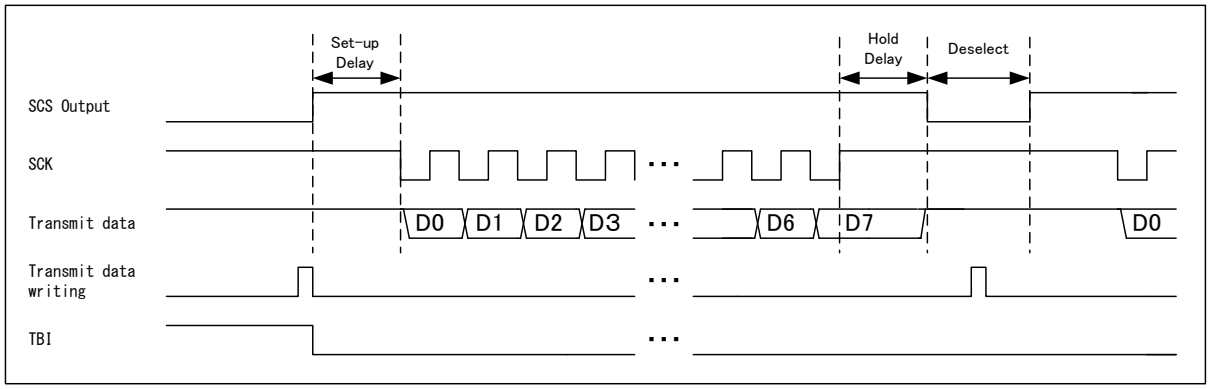
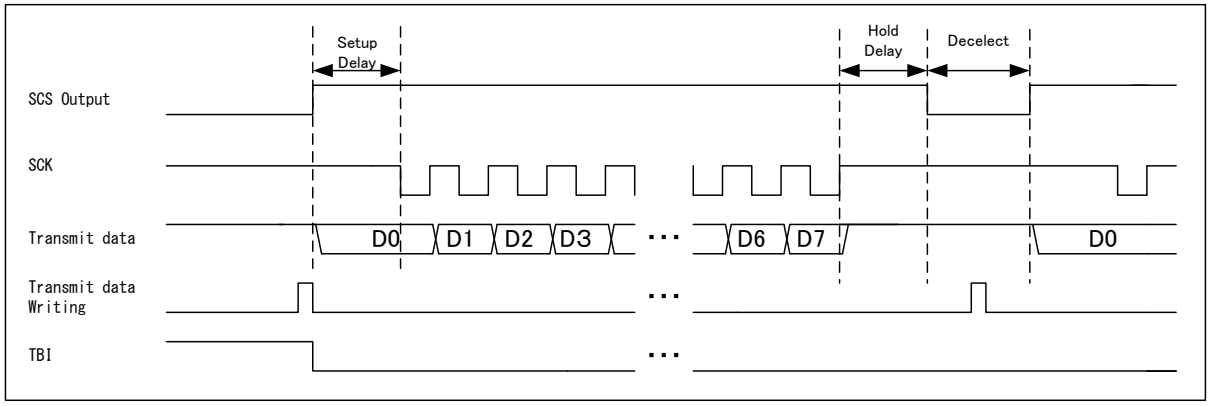


Figure 5-4 Timing Adjustment (SPI Transfer(SPI=1), SCINV=0)



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.*

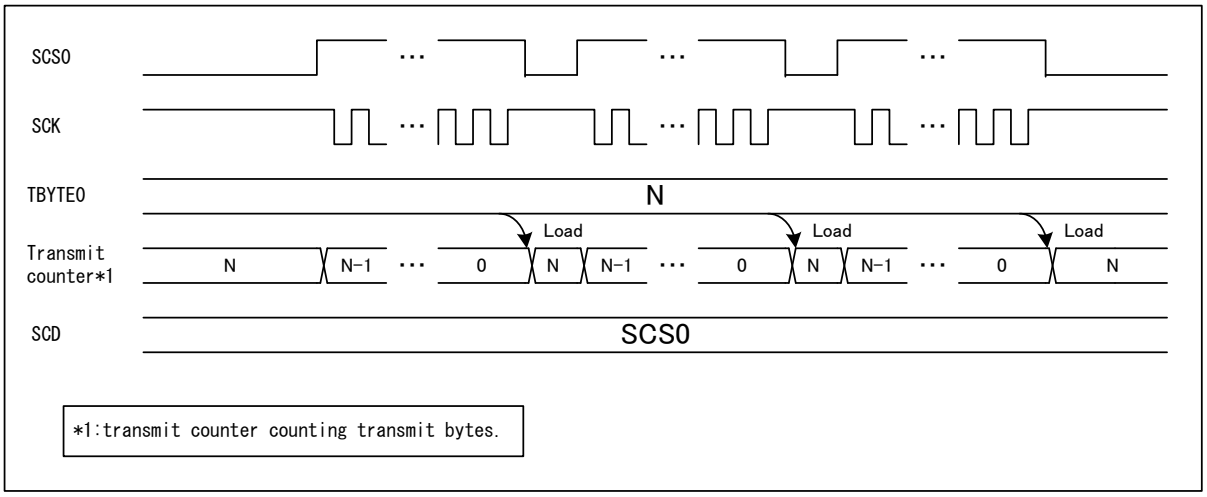
■ Chip Select Pin Independent Operation (Available only in Mater mode operation (SCR:MS=0))

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))

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)

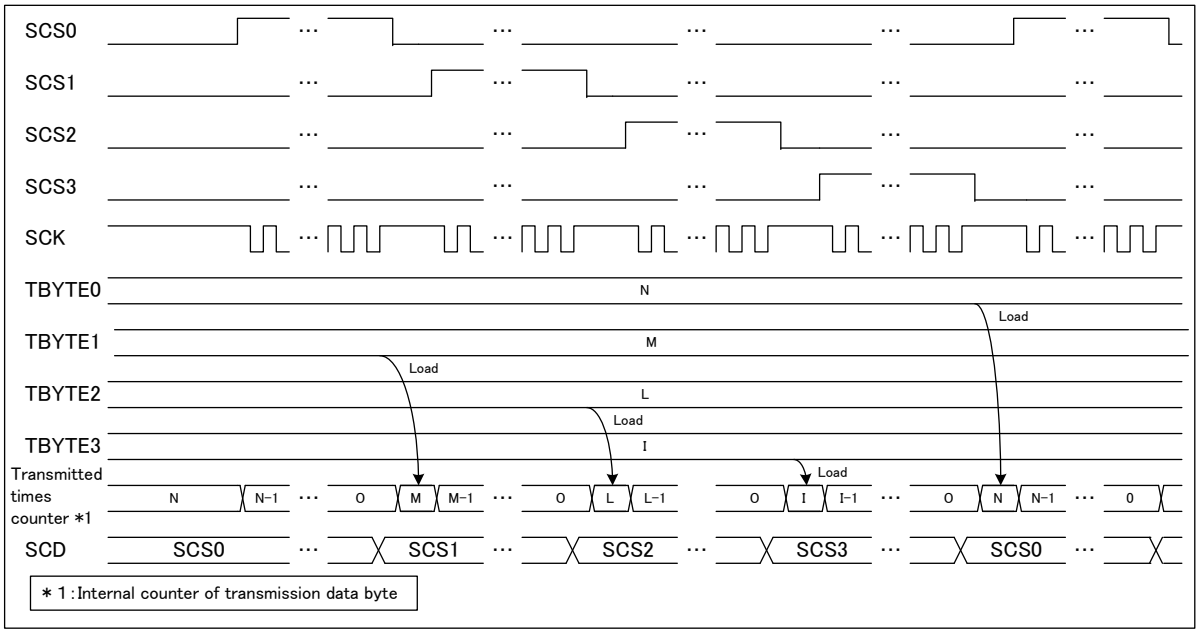


Figure 5-7 is the timing chart explaining the operation when Serial Chip Select Start pin is SCS1(SST1-0=1) and Serial Chip Select End pin is SCS2(SED1-0=2).

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)

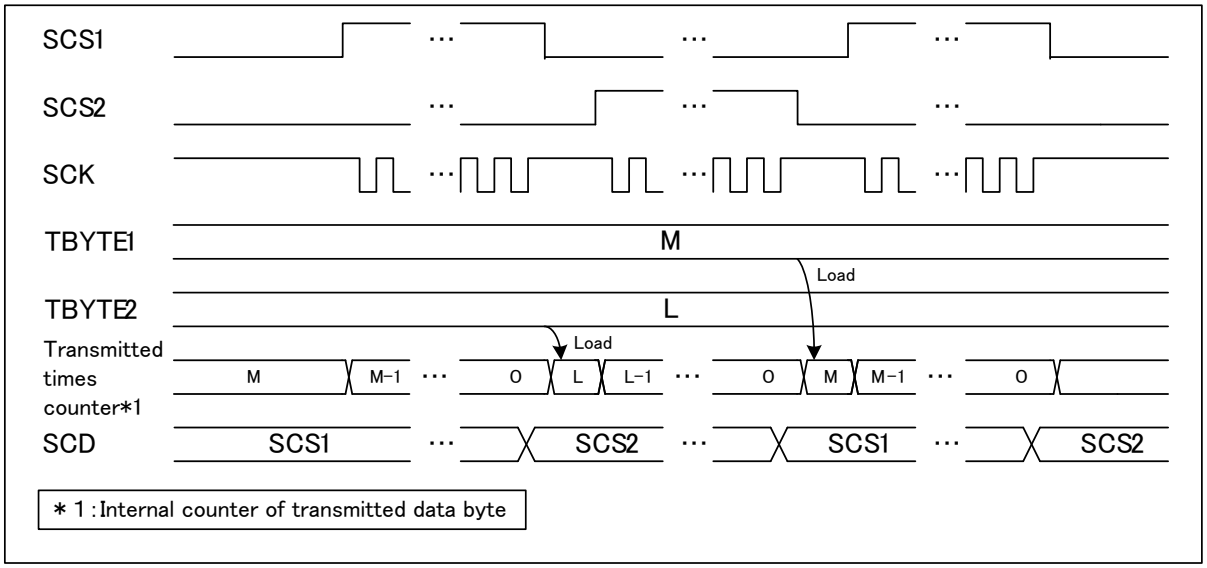
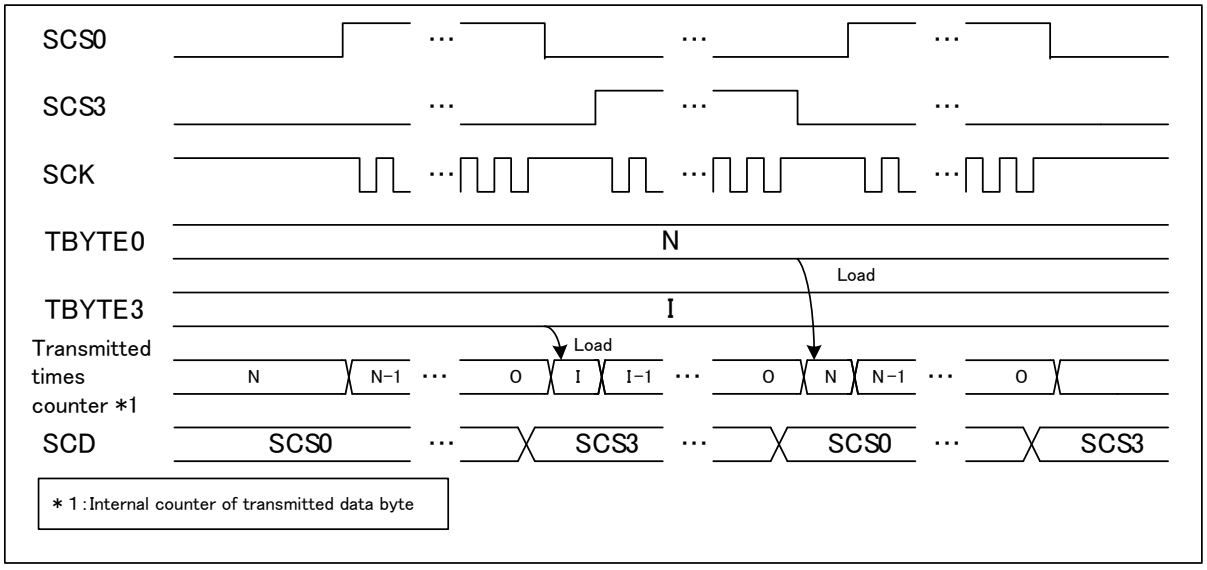


Figure 5-8 is the timing chart explaining the operation when Serial Chip Select Start pin is SCS0(SST1-0=0) and Serial Chip Select End pin is SCS3(SED1-0=3) and SCS1 and SCS3 are disable(CSEN1-2="00").

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)			

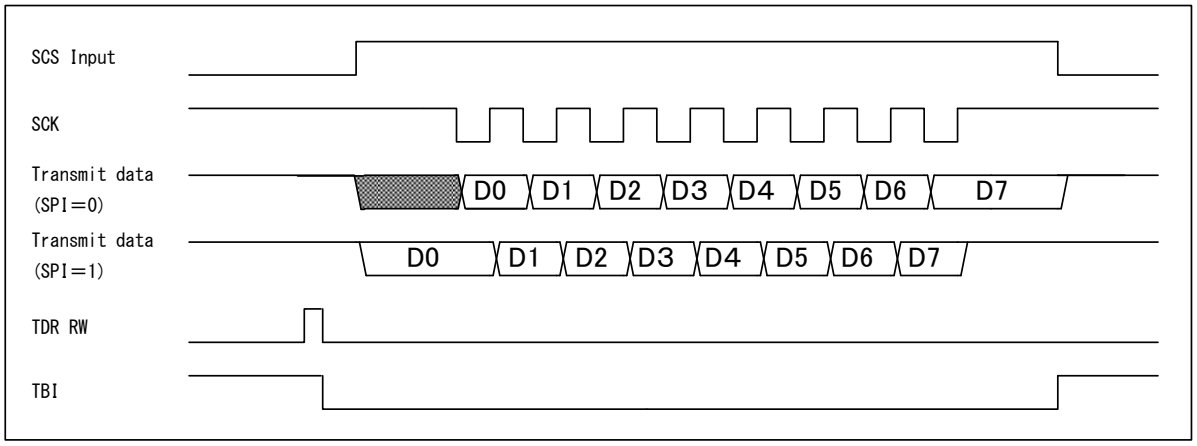
Notes:

- *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

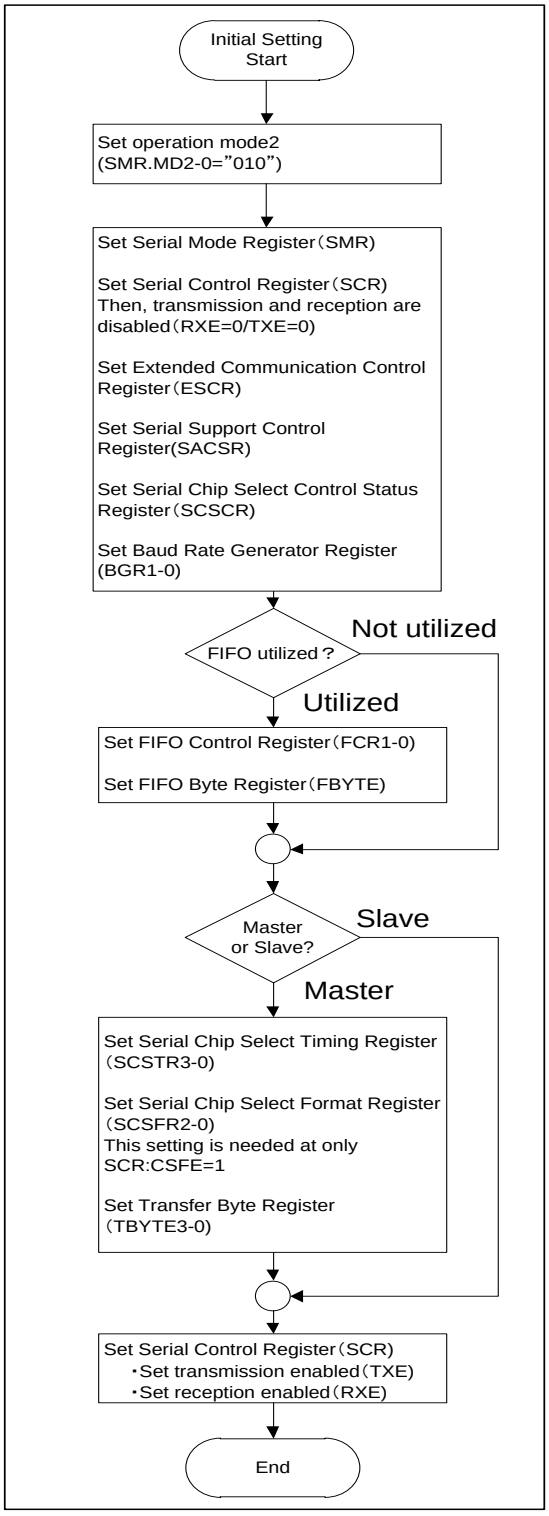
These can be set to each Chip Select Pin by the bits showed at Table 5-2, are chip select active level, clock inversion, SPI mode enabled/disabled, serial data direction and data length.

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 = ϕ / 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 × 1000000) / 19200 – 1 = 832

Therefore, the baud rate is:

b = (16 × 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 × 1000000) / 153600 - 1=129

Baud rate (Calculated value) = (20 × 1000000) / (129 + 1) = 153846 (bps)

Error (%) = (153846 - 153600) / 153600 × 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 (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	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	
	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
4M	9	0	11	0	17	0	19	0
2.5M	15	0	-	-	-	-	31	0
2M	19	0	23	0	35	0	39	0
1M	39	0	47	0	71	0	79	0
500000	79	0	95	0	143	0	159	0
460800	86	-0.22	103	0.16	155	0.16	173	-0.22
250000	159	0	191	0	287	0	319	0
230400	173	-0.22	207	0.16	312	-0.16	346	0.06
153600	259	0.16	312	-0.16	468	-0.05	520	-0.03
125000	319	0	383	0	575	0	639	0
115200	346	0.06	416	-0.08	624	0	693	0.06
76800	520	-0.03	624	0	937	-0.05	1041	-0.03
57600	693	0.06	832	0.04	1249	0	1388	<0.01
38400	1041	-0.03	1249	0	1874	0	2082	0.01
28800	1388	<0.01	1666	-0.02	2499	0	2777	<0.01
19200	2082	0.01	2499	0	3749	0	4166	-0.01
10417	3839	<0.01	4607	<0.01	6911	<0.01	7679	0
9600	4166	<0.01	4999	0	7499	0	8332	0
7200	5555	<0.01	6666	<0.01	9999	0	11110	0
4800	8332	<0.01	9999	0	14999	0	16666	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)

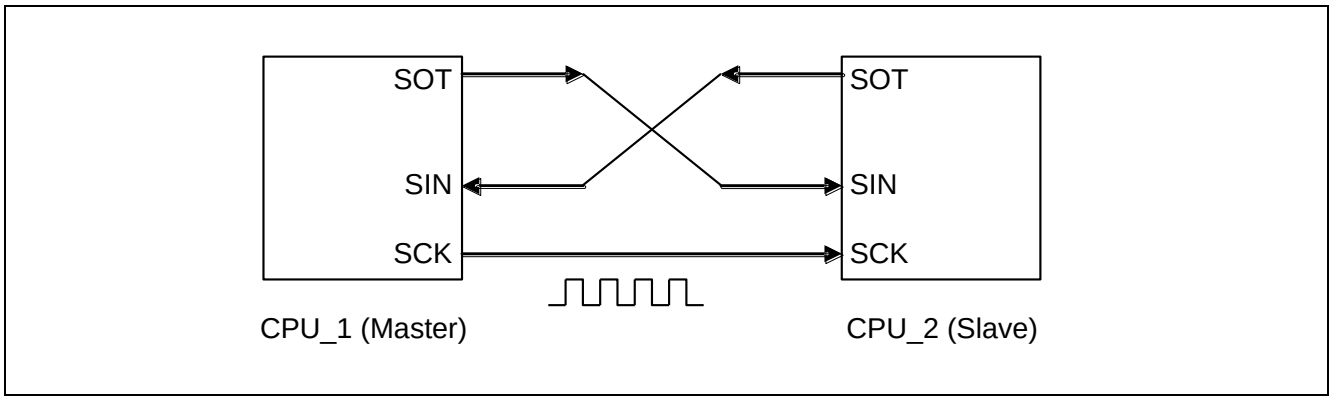
6.2 CSIO (Clock Synchronous Serial Interface) Setup Procedure and Program Flow

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.

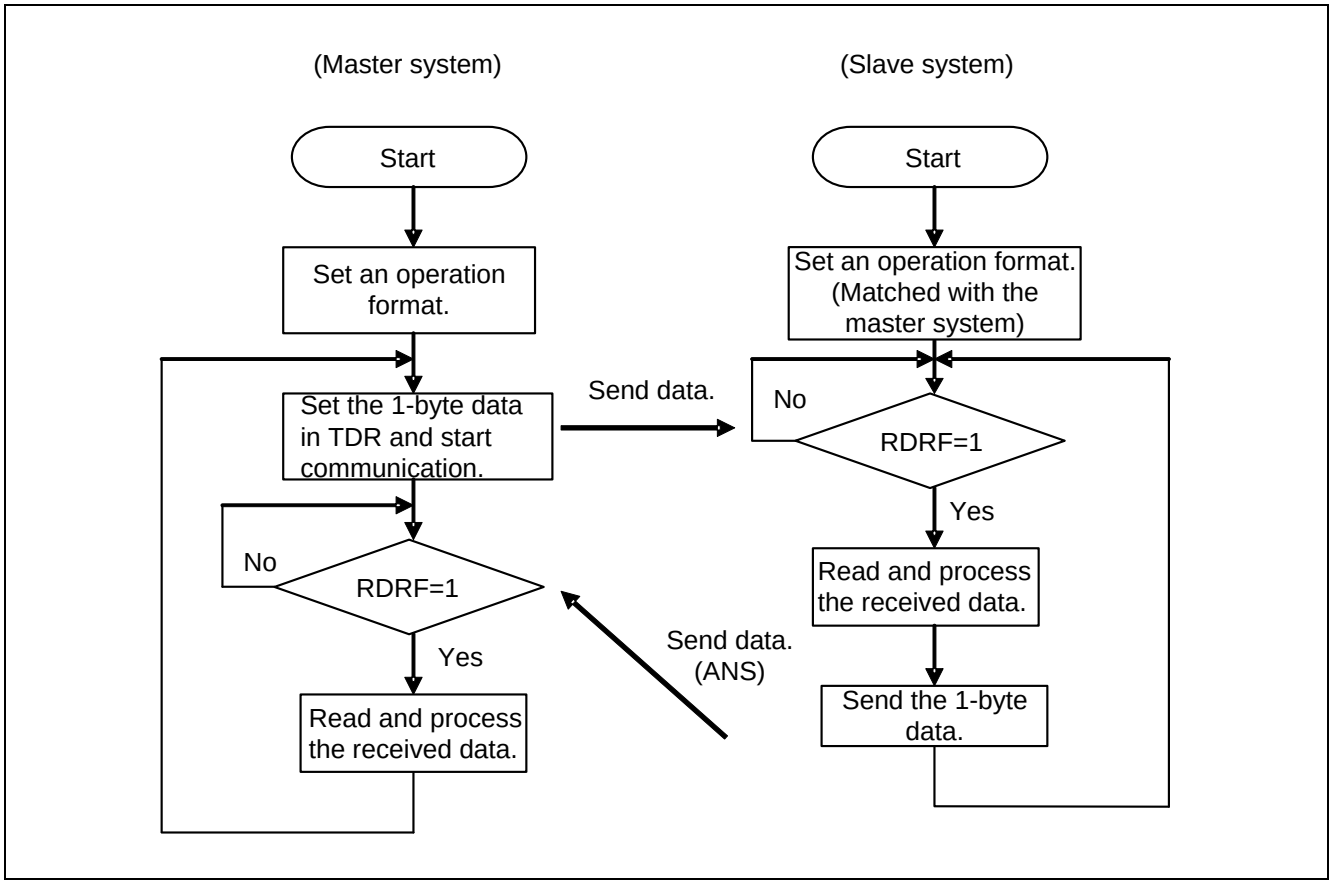
Figure 6-1 Connection Example for CSIO (Clock Synchronous Serial Interface) Bidirectional Communication



Flowcharts

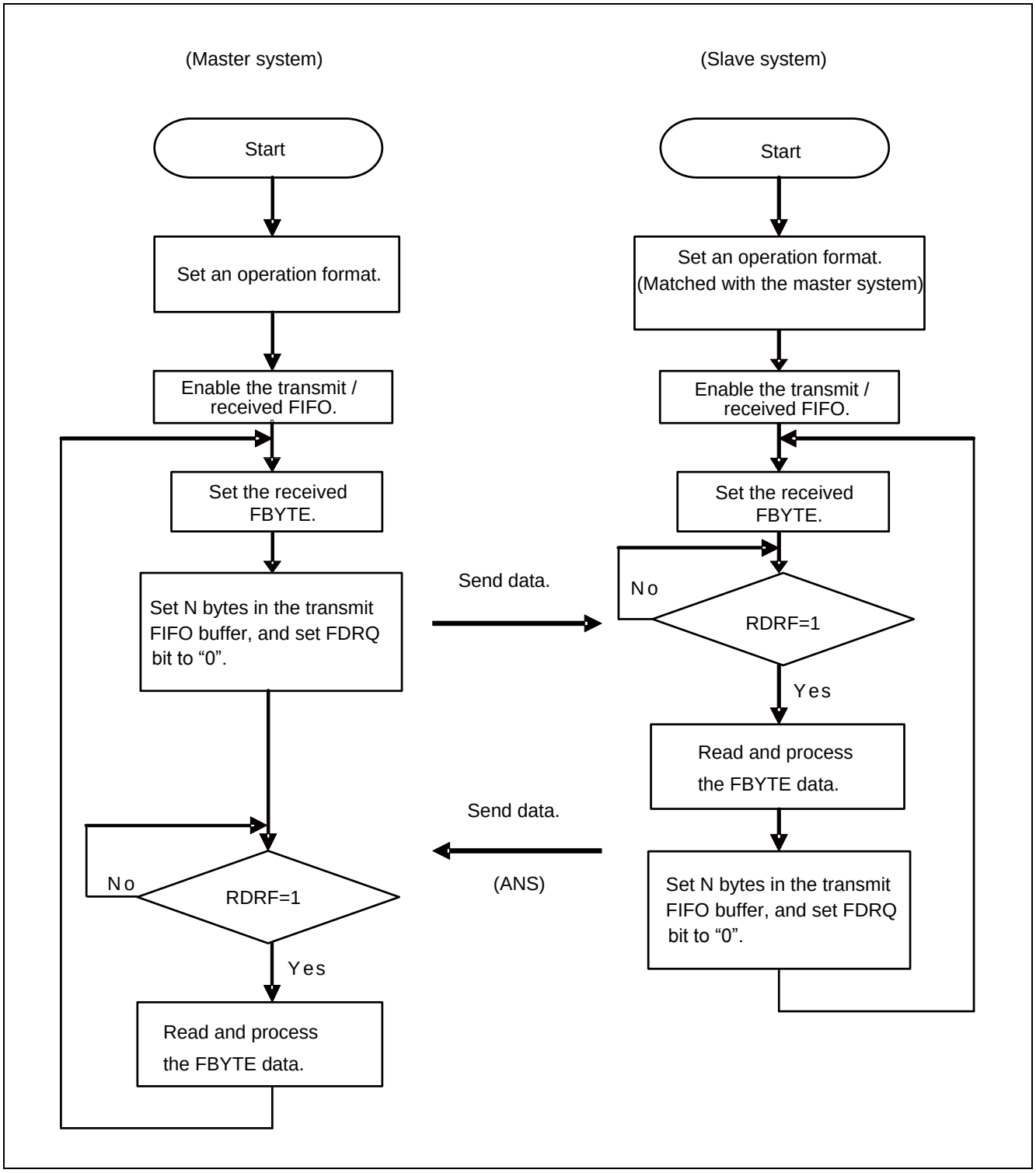
■ If FIFO is not used

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

	bit15 bit8	bit7 bit0
CSIO	SCR (Serial Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	ESCR (Extended Communication Control Register)
	RDR/TDR (Transmit/Received Data register)	
	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 7-2 CSIO (Clock Synchronous Serial Interface) Bit Assignment

	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
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	CS1 TD7	CS1 TD6	CS1 TD5	CS1 TD4	CS1 TD3	CS1 TD2	CS1 TD1	CS1 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.*
- *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 any of the following cases.*
  - *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).*
- *Set this bit when serial clock output is disabled(SCKE=0).*
- *After setting the SCINV bit, enable data reception (SCR:RXE=1).*
- *This bit is used 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)*
  - *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 any of the following cases:*
  - *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 bit

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 ".

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

Note:

- *When Chip Select error(SACSR:CSE=1) occurs at the state that Transmit Data register(TDR) is empty(SSR:TDRE=1), this bit becomes "1" within the period of baud rate.*

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.

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] CSFE: Serial Chip Select Format enable 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 any of the following conditions:*
  - *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 (SOT).
- 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 ".*

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 Reregister (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 exacting 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.

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

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

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

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:*
  - *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.*

[bit4:1]CSEN3-0: Serial Chip Select Enable bit

This bits is 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, 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 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.*
- *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)

This register is used to set the active level, clock Inversion, SPI mode, data direction and data length in each serial chip select pin.

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 Value		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.

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	(TBYTE1)								(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	(TBYTE3)								(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:CSEN3-0=”0000”), 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 bit14 to bit8 of reload counter.
Read	Reads the BGR1 set value.

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

bit7:0	Description
Write	Write data in bit7 to bit0 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 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 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.

[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 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 7-3 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 set the SCR.RXE bit to "1", by setting the SCR:TXE to "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 7-4 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")

CHAPTER1-4: 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_FM0-E03.0_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".

Interrupt	Interrupt	Flag	Interrupt Factor	Interrupt	Operation to Clear
Input capture*2	ICP0/ ICP1	ICSA10/I CSA32	The first rising edge in the LIN Sync field	ICSA10.ICE0 ICSA10.ICE1	Disables ICP0 and ICP1
	ICP0/ ICP1	ICSA10/I CSA32	The fifth falling edge in the LIN Sync field	ICSA32.ICE0 ICSA32.ICE1	

*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

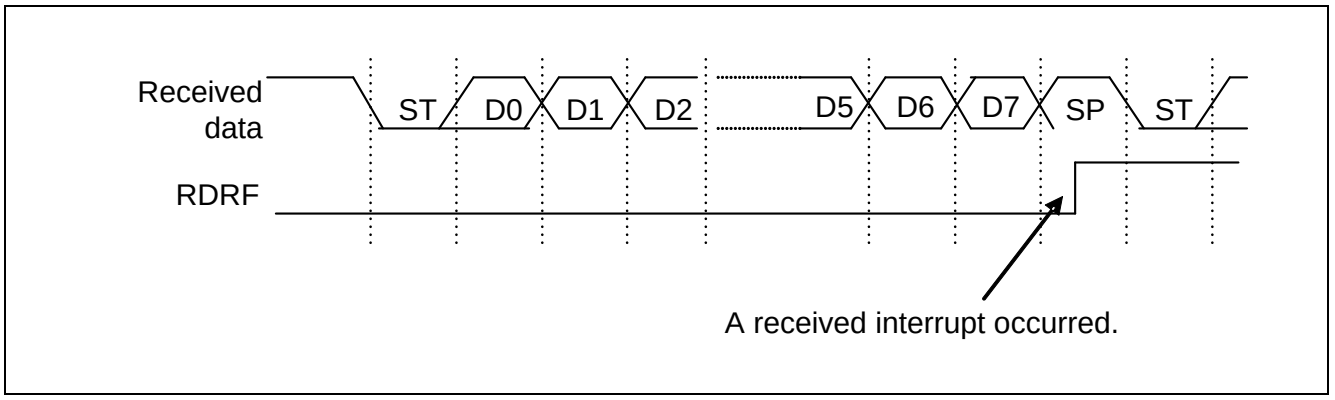
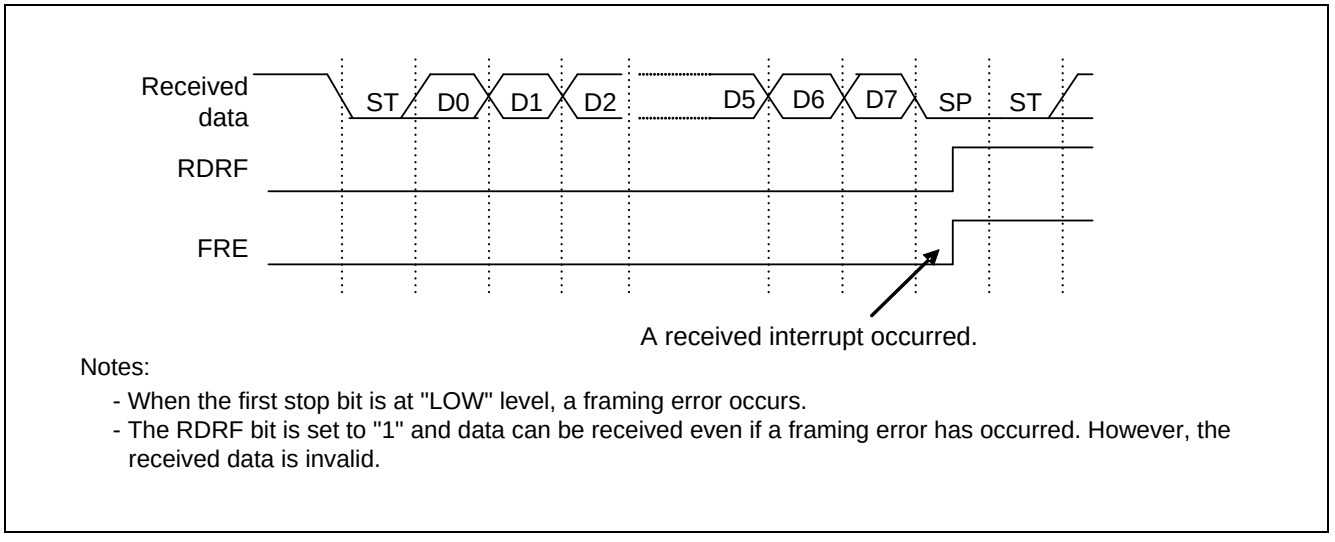


Figure 2-2 FRE (Framing Error Flag Bit) Set Timing



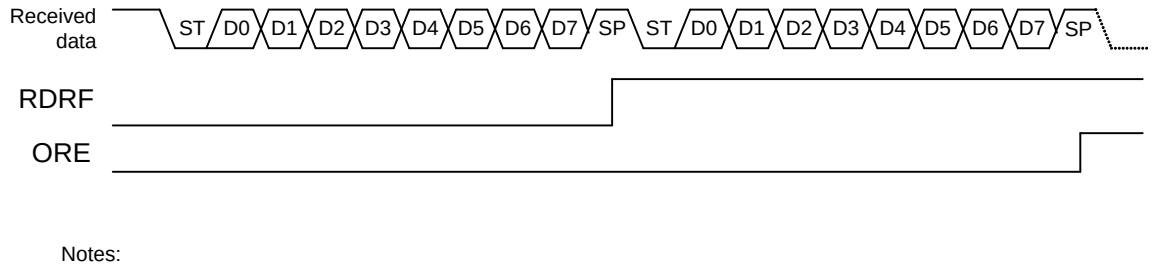
Notes:

- 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



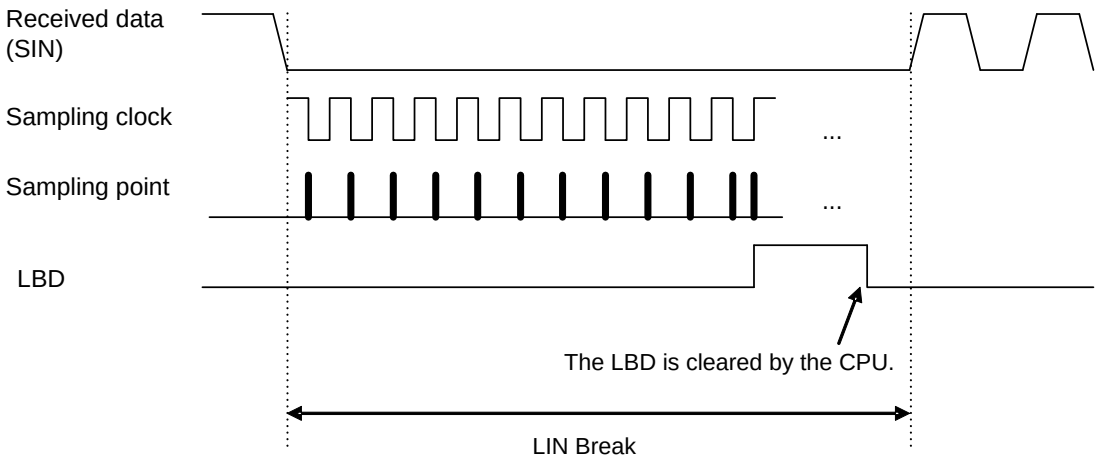
Notes:

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

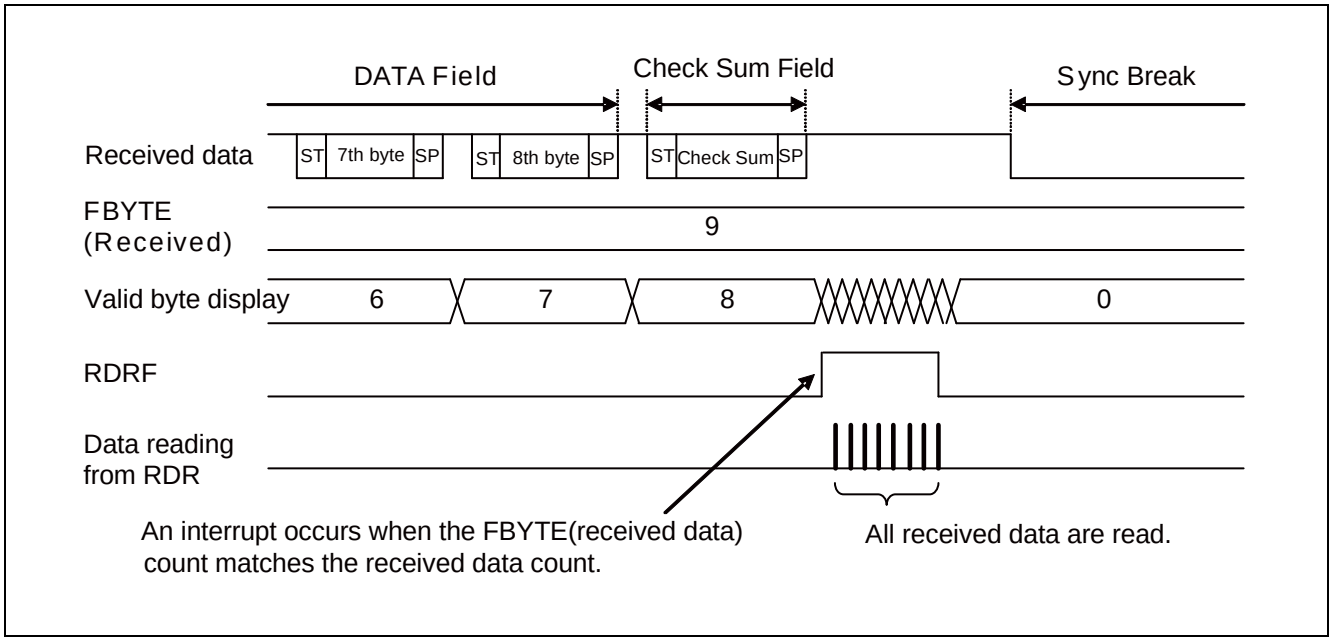
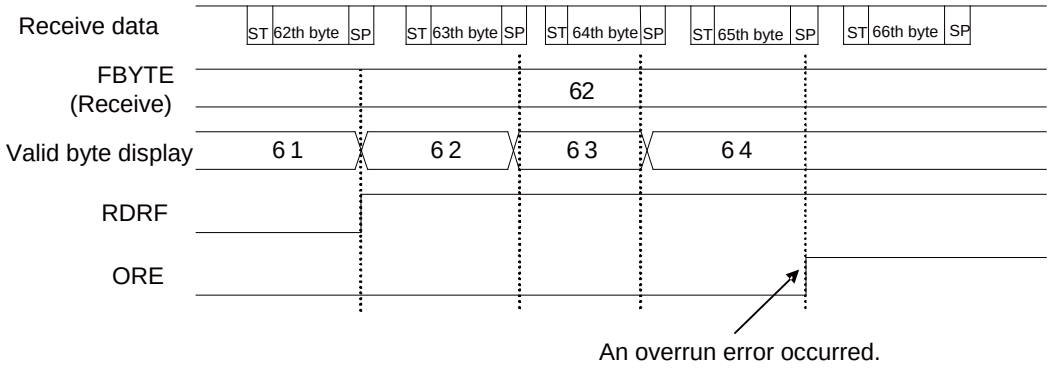


Figure 2-6 ORE (Overrun Error) Flag Bit Set Timing



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

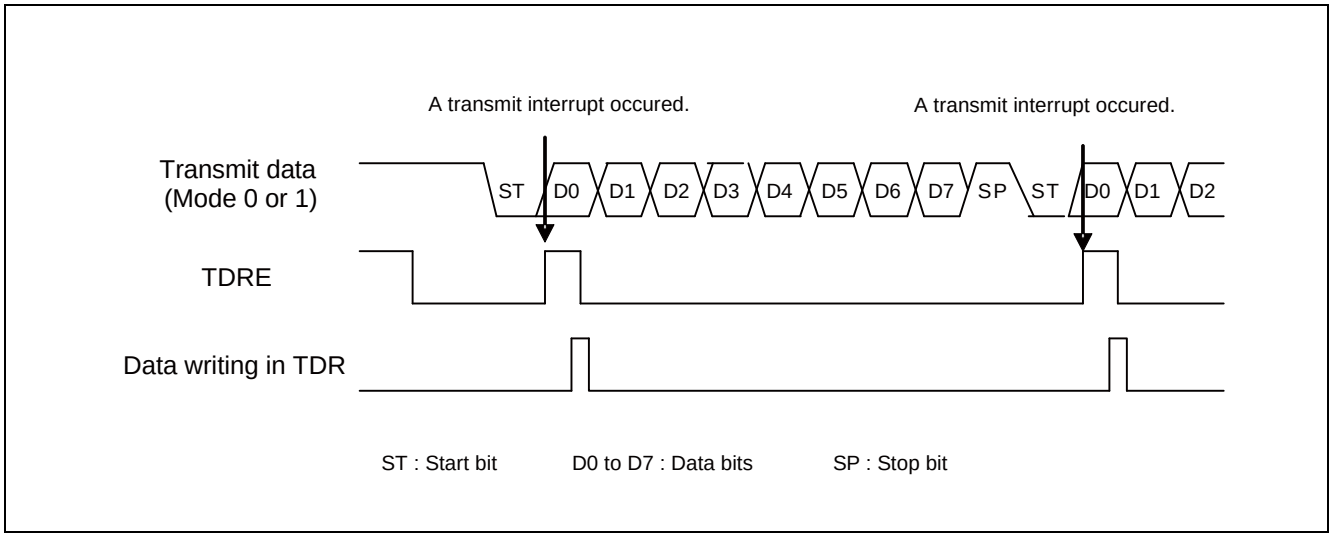
A transmit interrupt occurs when transmit data is transferred from the Transmit Data Register (TDR) to the transmit shift register (SSR:TDRE = 1) and transmission starts and when no transmission is performed (SSR:TBI = 1).

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).

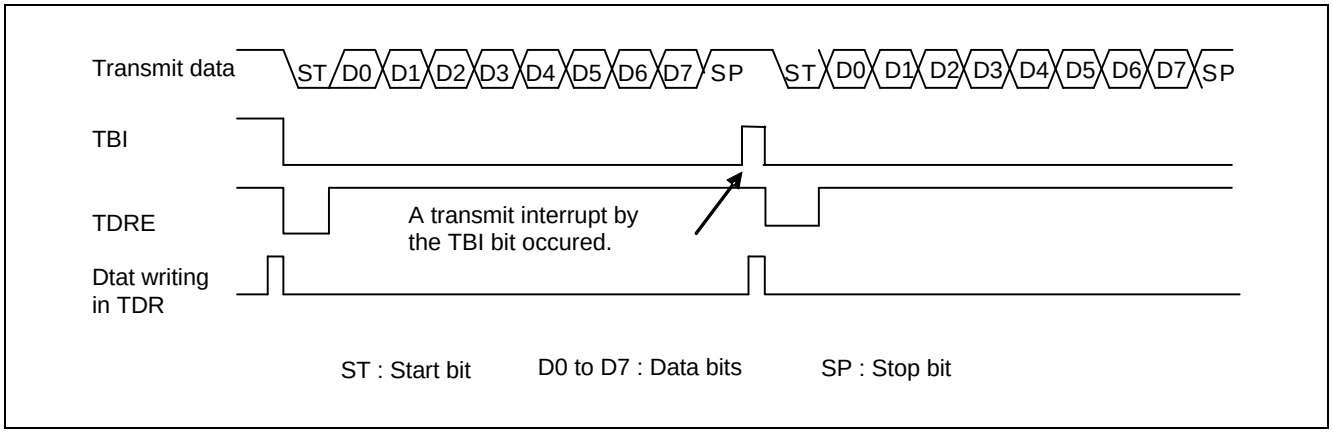
Figure 2-7 Transmit Data Empty Flag (SSR:TDRE) Set Timing



■ 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



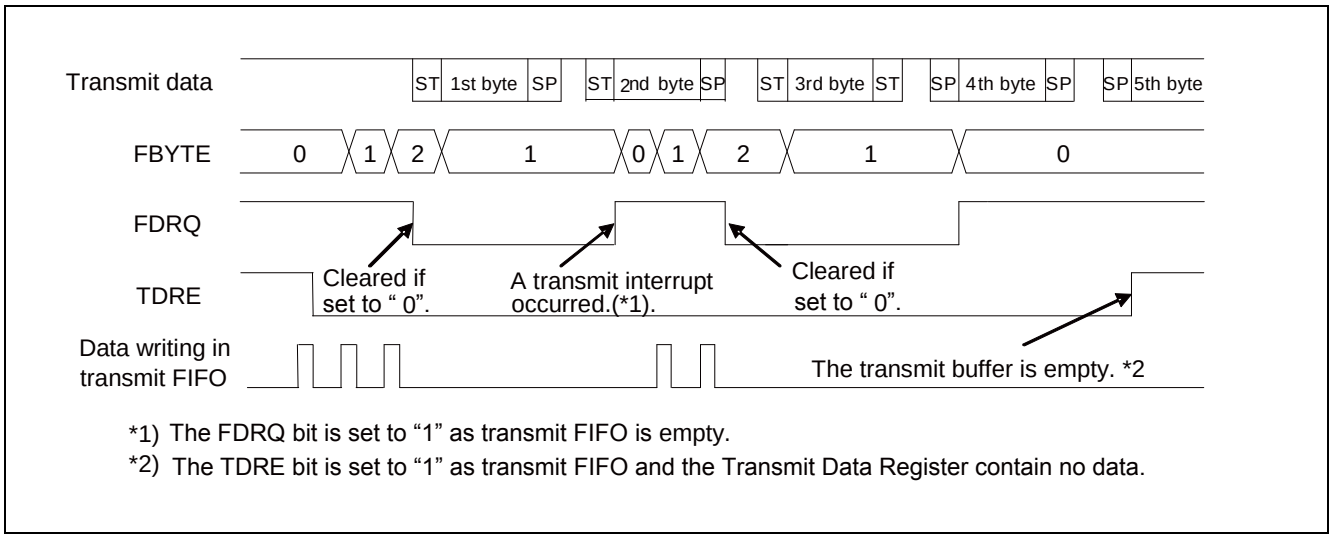
2.4 Interrupt and Flag Set Timing when Transmit FIFO is Used

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

Baud 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 (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	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	
	Value	ERR
8M	4	0
6M	-	-
5M	7	0
4M	9	0
2.5M	15	0
1M	39	0
500000	79	0
460800	86	-0.22
250000	159	0
230400	173	-0.22
153600	259	0.16
125000	319	0
76800	520	-0.03
57600	693	0.06
38400	1041	-0.03
28800	1388	<0.01
19200	2082	0.01
10417	3839	<0.01
9600	4166	<0.01
7200	5555	<0.01
4800	8332	<0.01
2400	16666	<0.01
1200	-	-
600	-	-
300	-	-

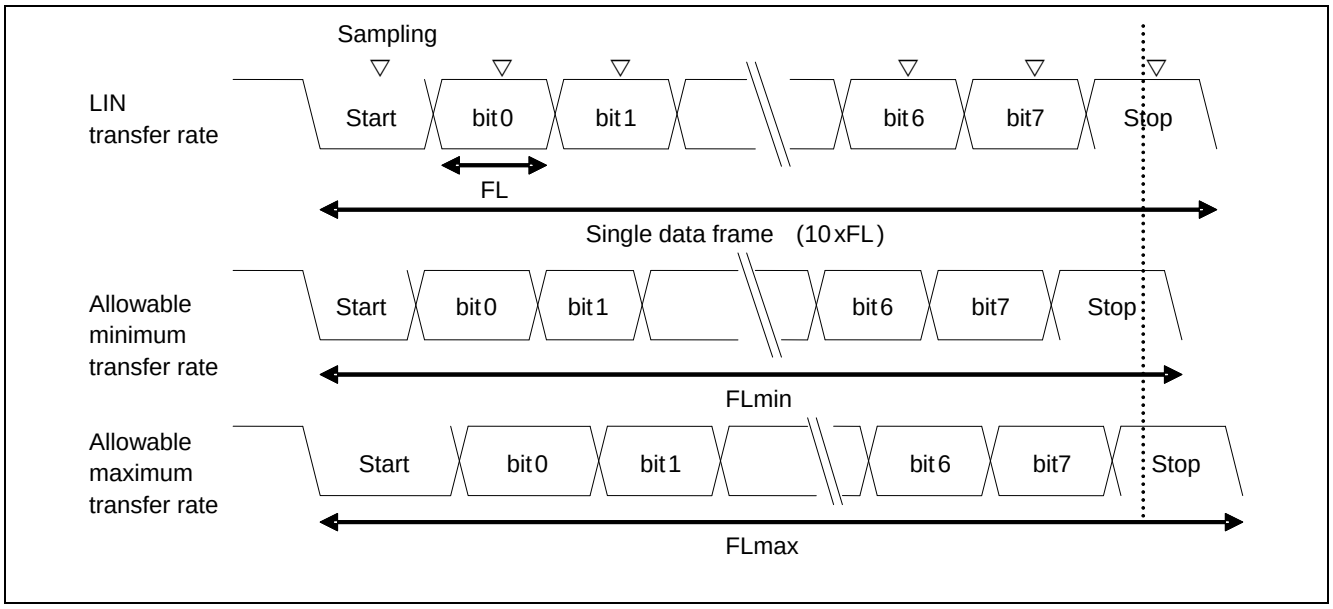
For frequencies not described in Table 3-1 and Table 3-2, calculate them by using formulas in “3.1 Baud rate settings”. (However, for the maximum frequencies, see “Data Sheet” of the product used because they are differed by products)

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

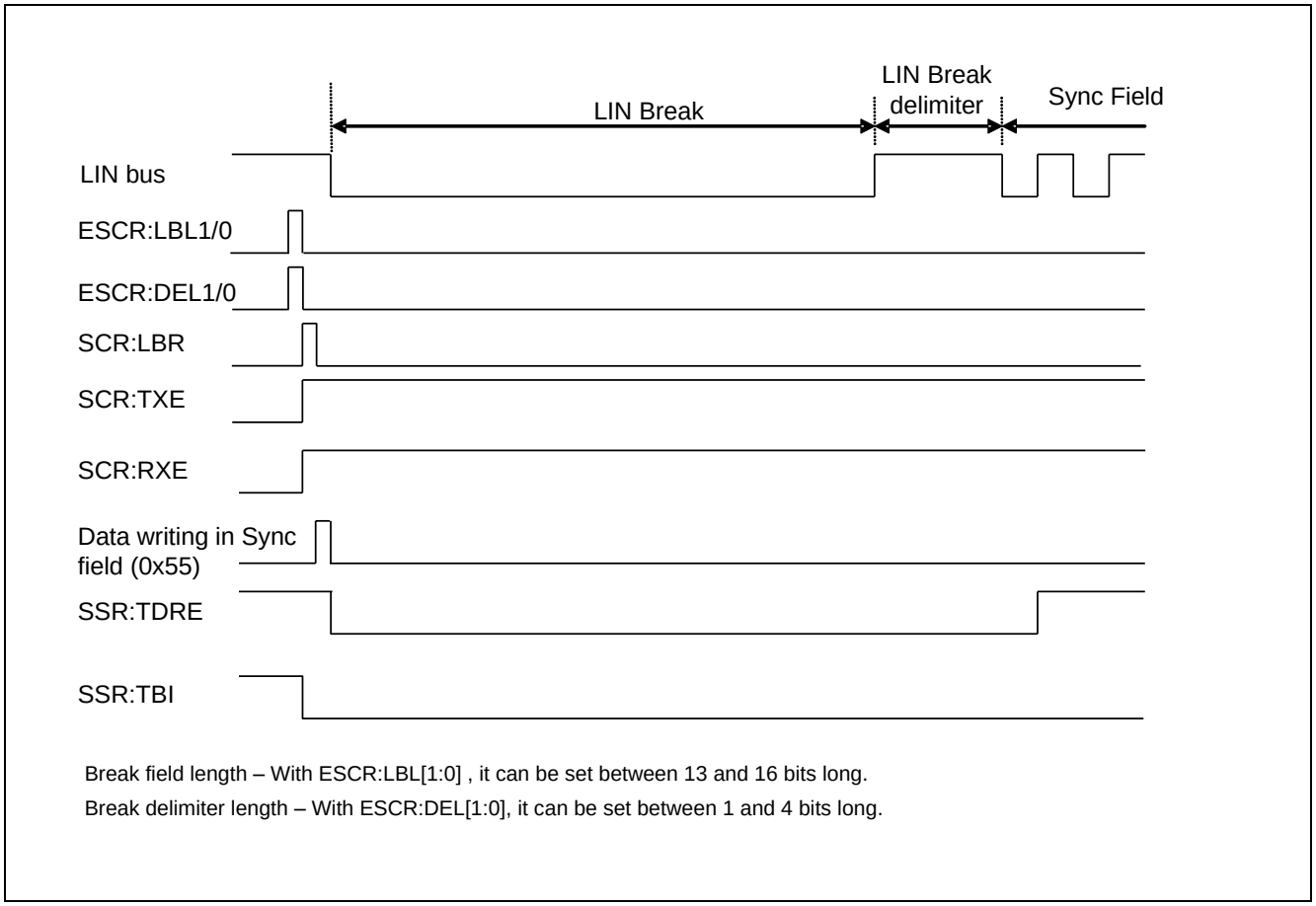
To operate the LIN interface as a master, set the SCR:MS bit to "0".

- 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.*

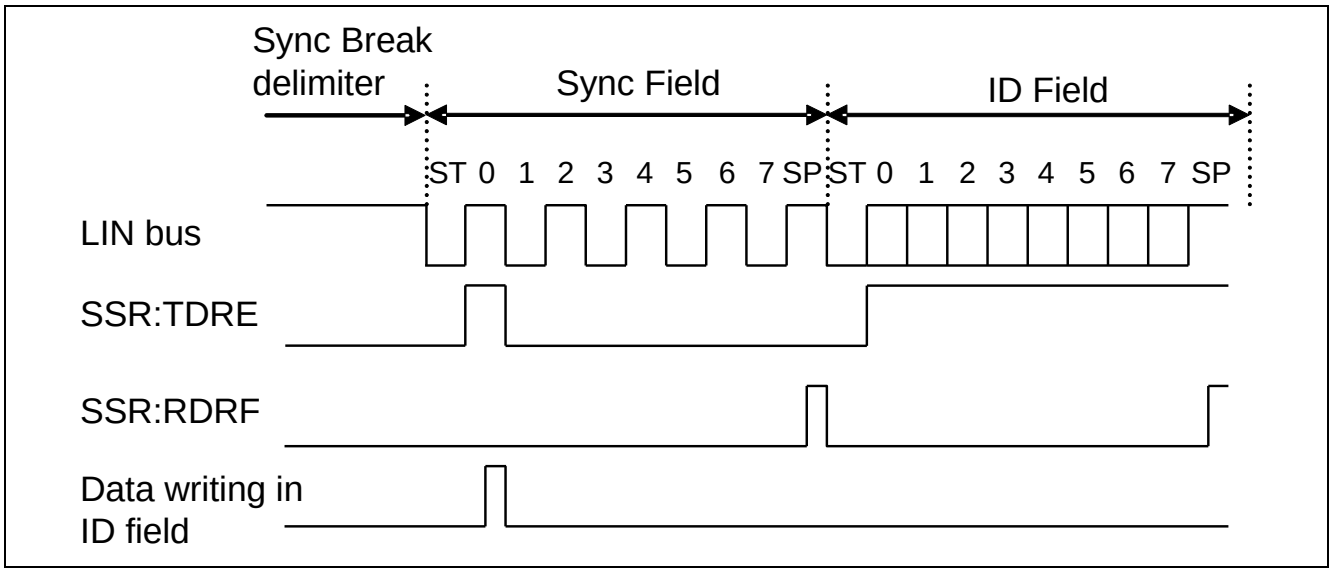
Figure 4-1 Break Field-Sync Field Transmission



Break field length – With ESCR:LBL[1:0] , it can be set between 13 and 16 bits long.

Break delimiter length – With ESCR:DEL[1:0], it can be set between 1 and 4 bits long.

- 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.



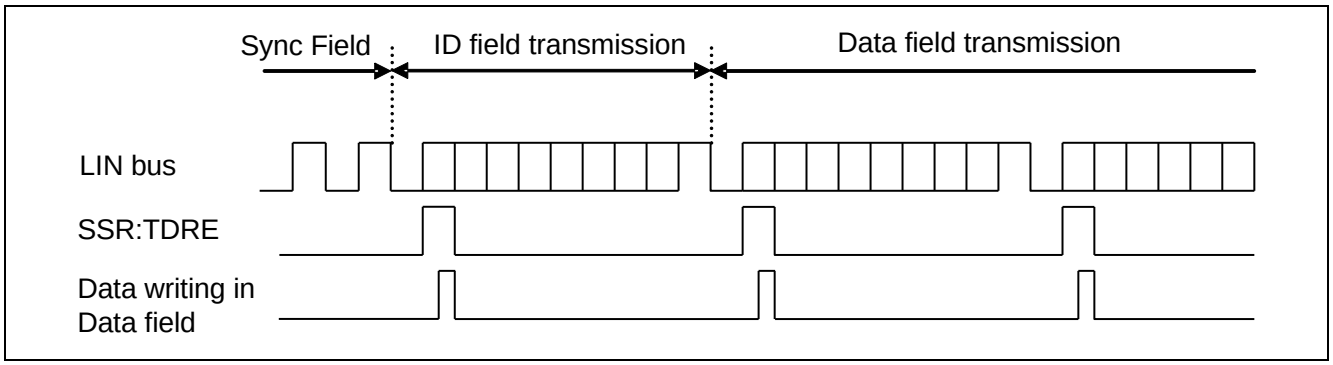
- ID field transmission - DATA field transmission/reception

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.

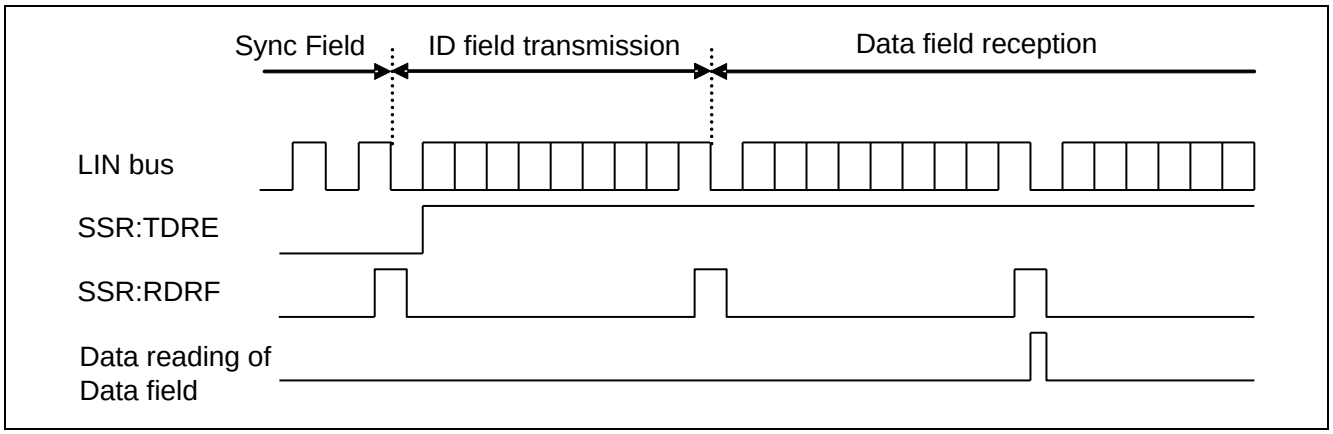
Figure 4-2 ID Field Transmission-DATA Field Transmission



(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)

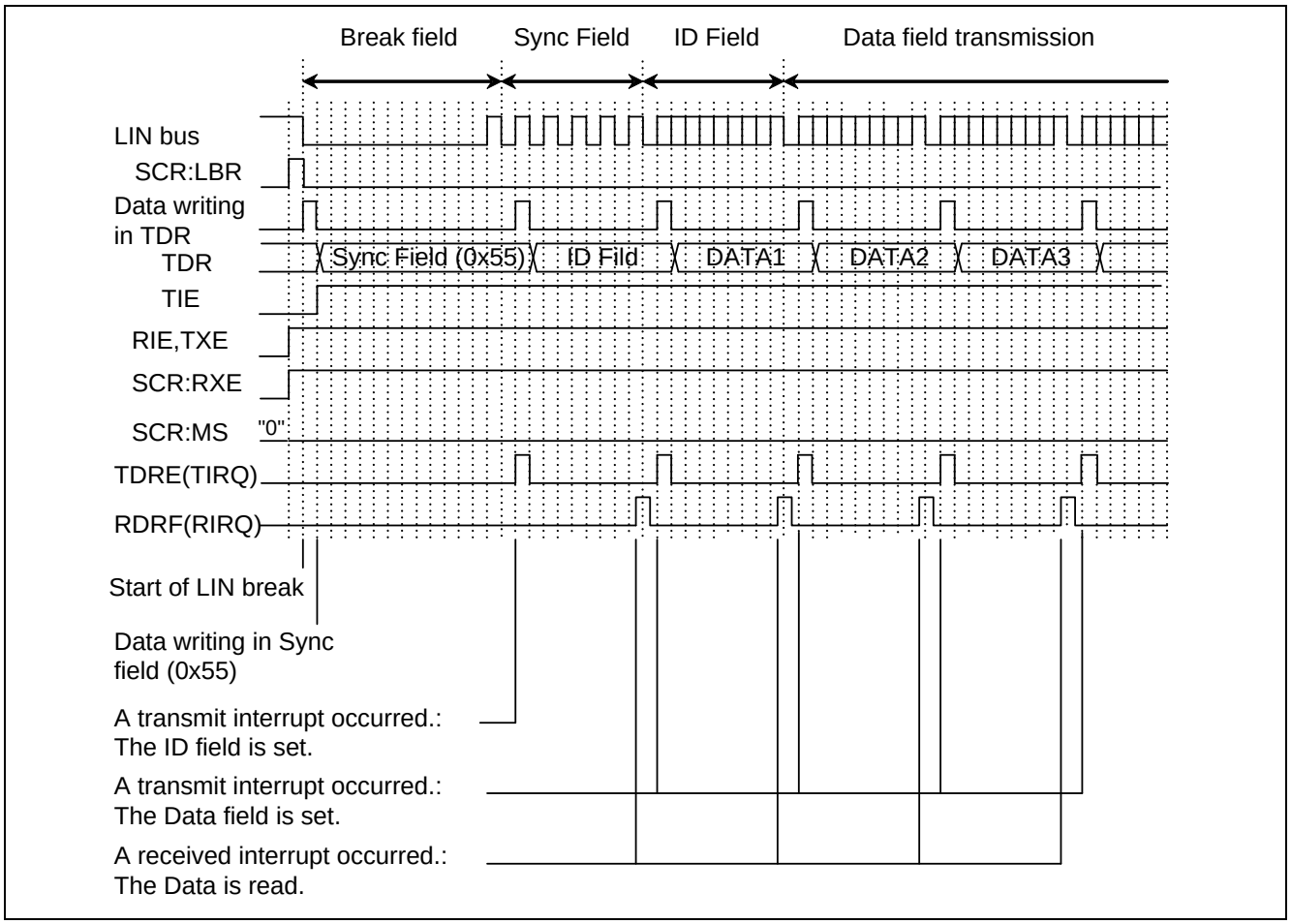
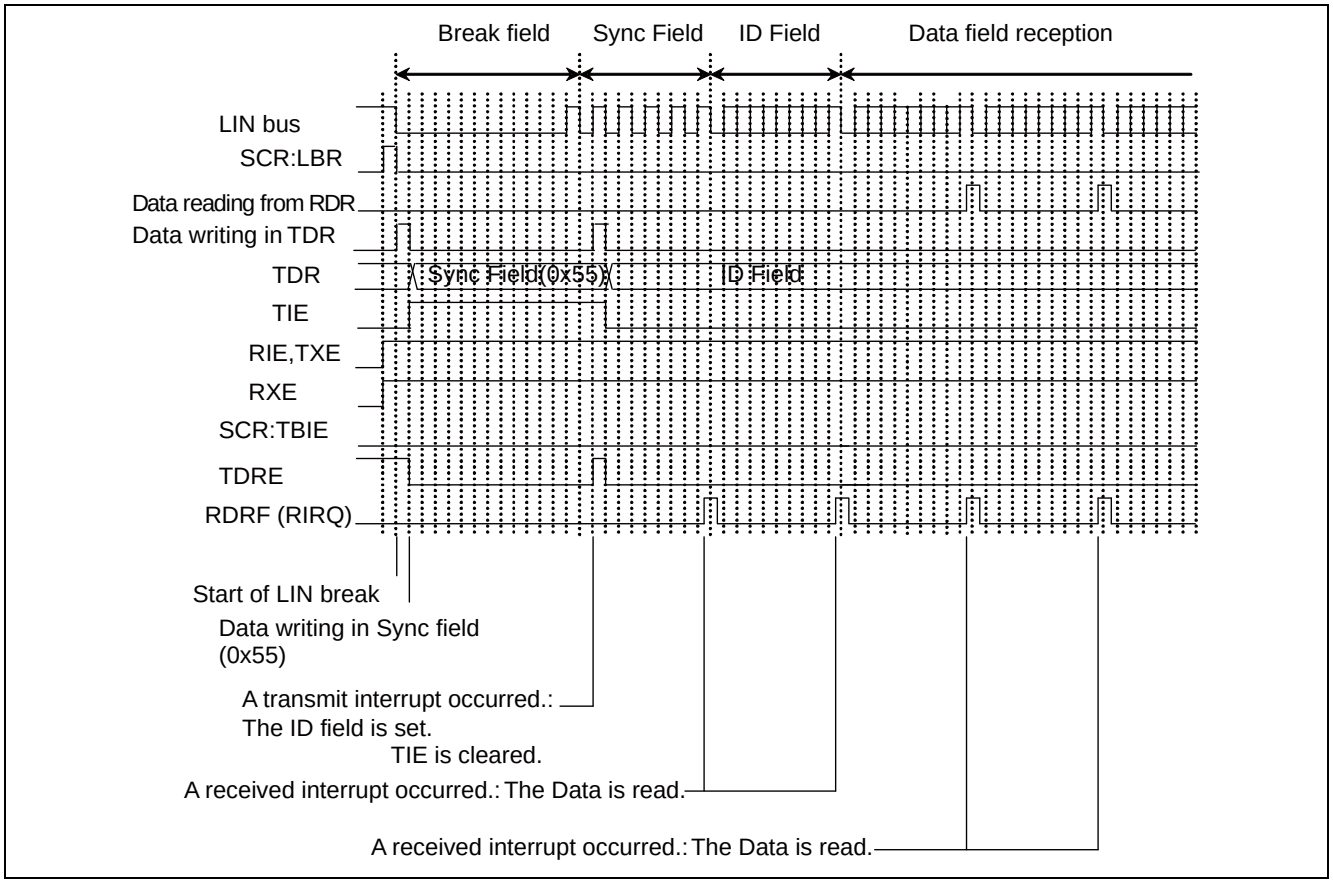


Figure 4-5 LIN Bus Timing (when DATA Field is Received and FIFO is Not Used)



■ Master mode operation timing chart (when FIFO is used)

Figure 4-6 LIN Bus Timing (when DATA Field is Transmitted and FIFO is Used)

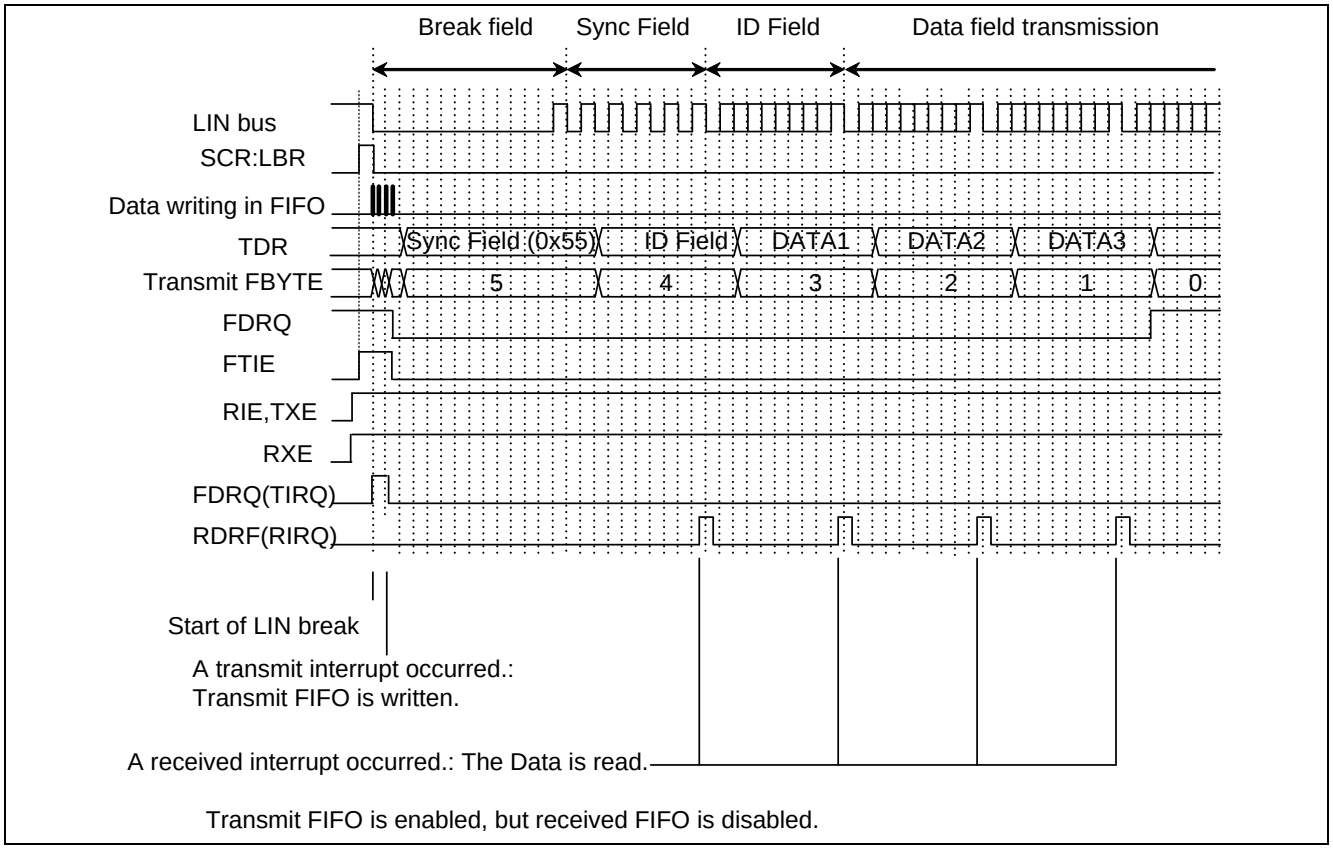
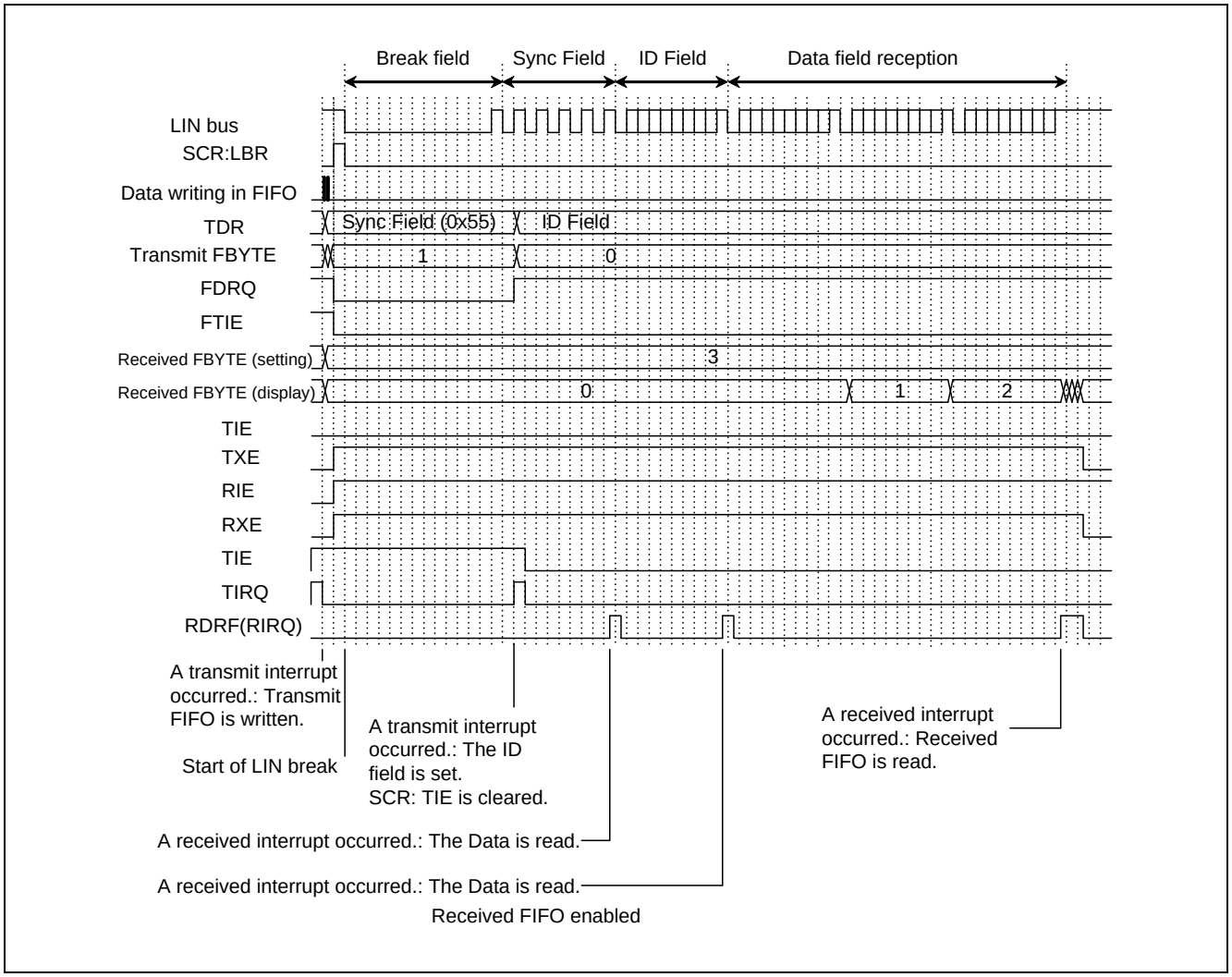


Figure 4-7 LIN Bus Timing (when DATA Field is Received and FIFO is Used)



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 = $(b - a) \times Fe/(8 \times \phi) - 1$

If the free run timer is overflowed:

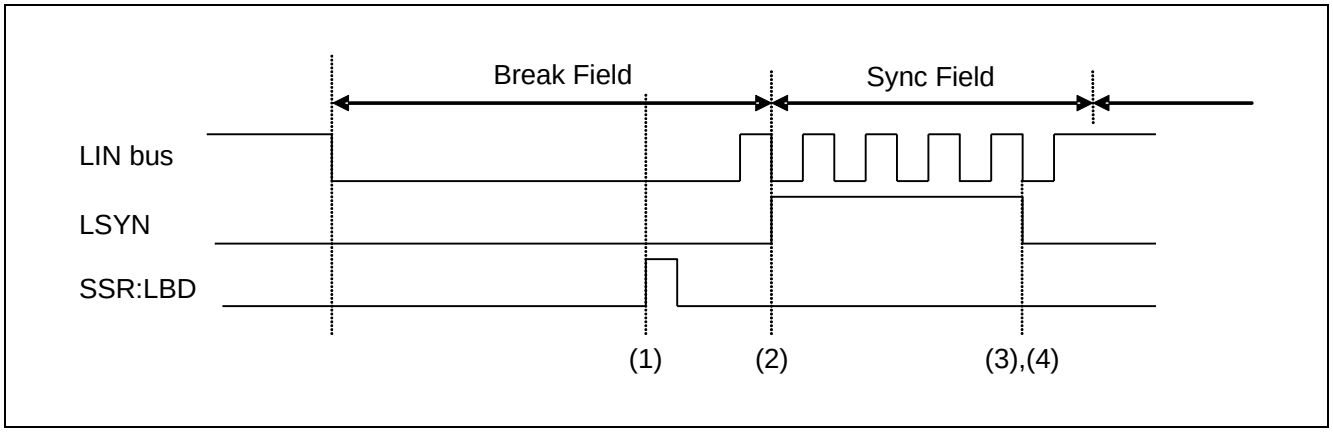
BGR value = $(max + 1 + b - a) \times Fe/(8 \times \phi) - 1$

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



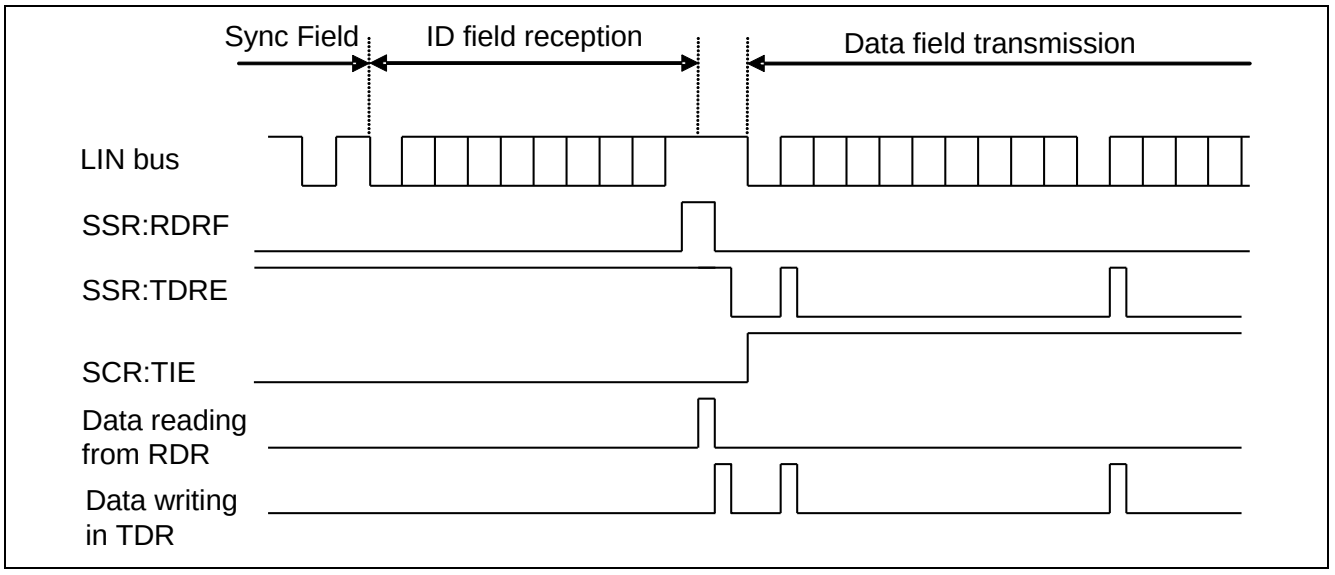
■ ID field reception - DATA field transmission/reception

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.

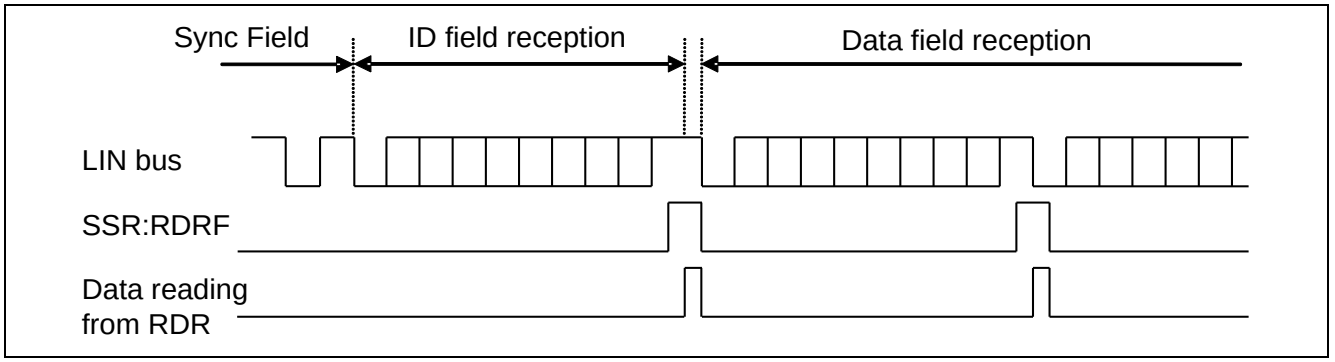
Figure 4-9 ID Field Reception - DATA Field Transmission



(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)

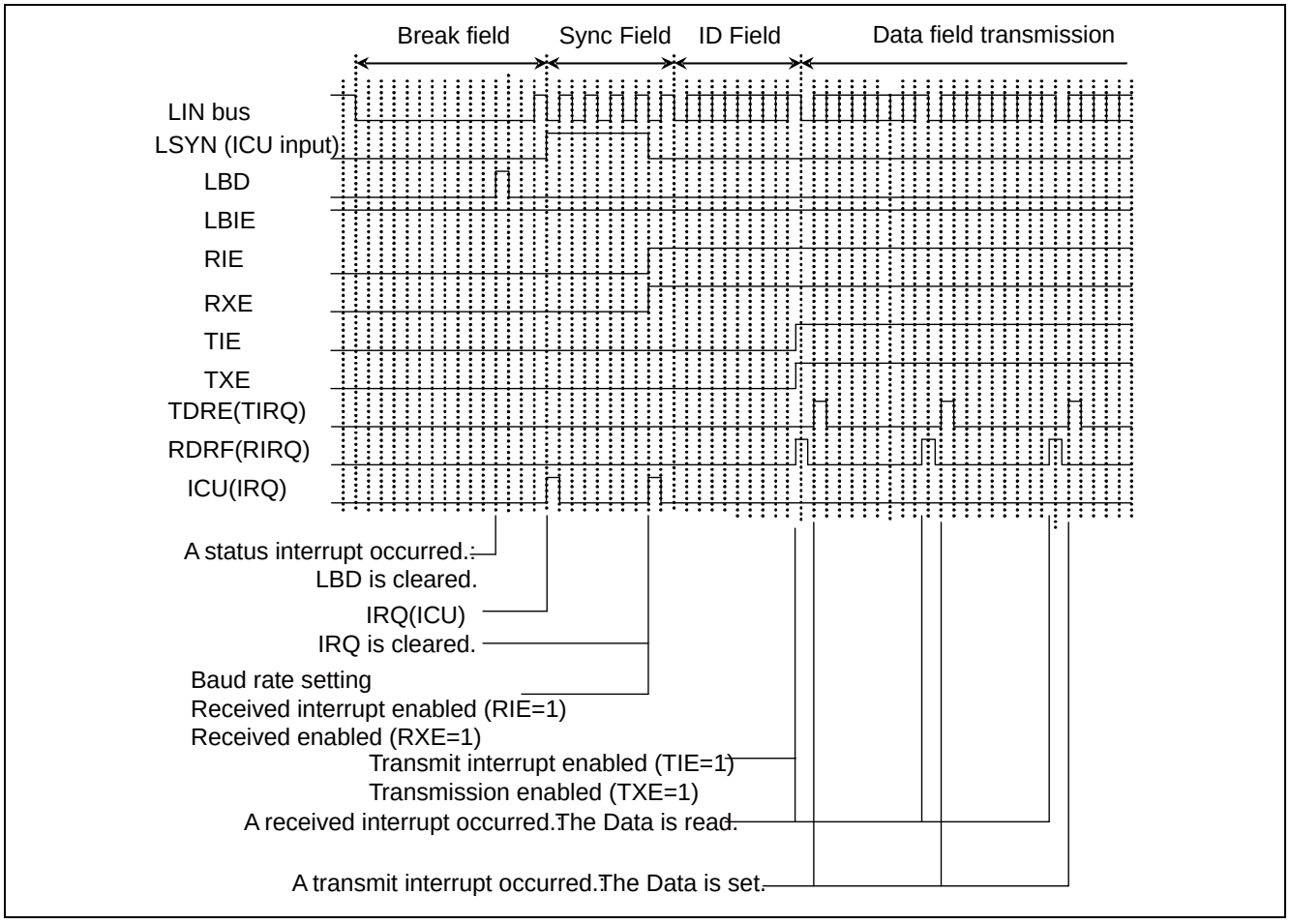
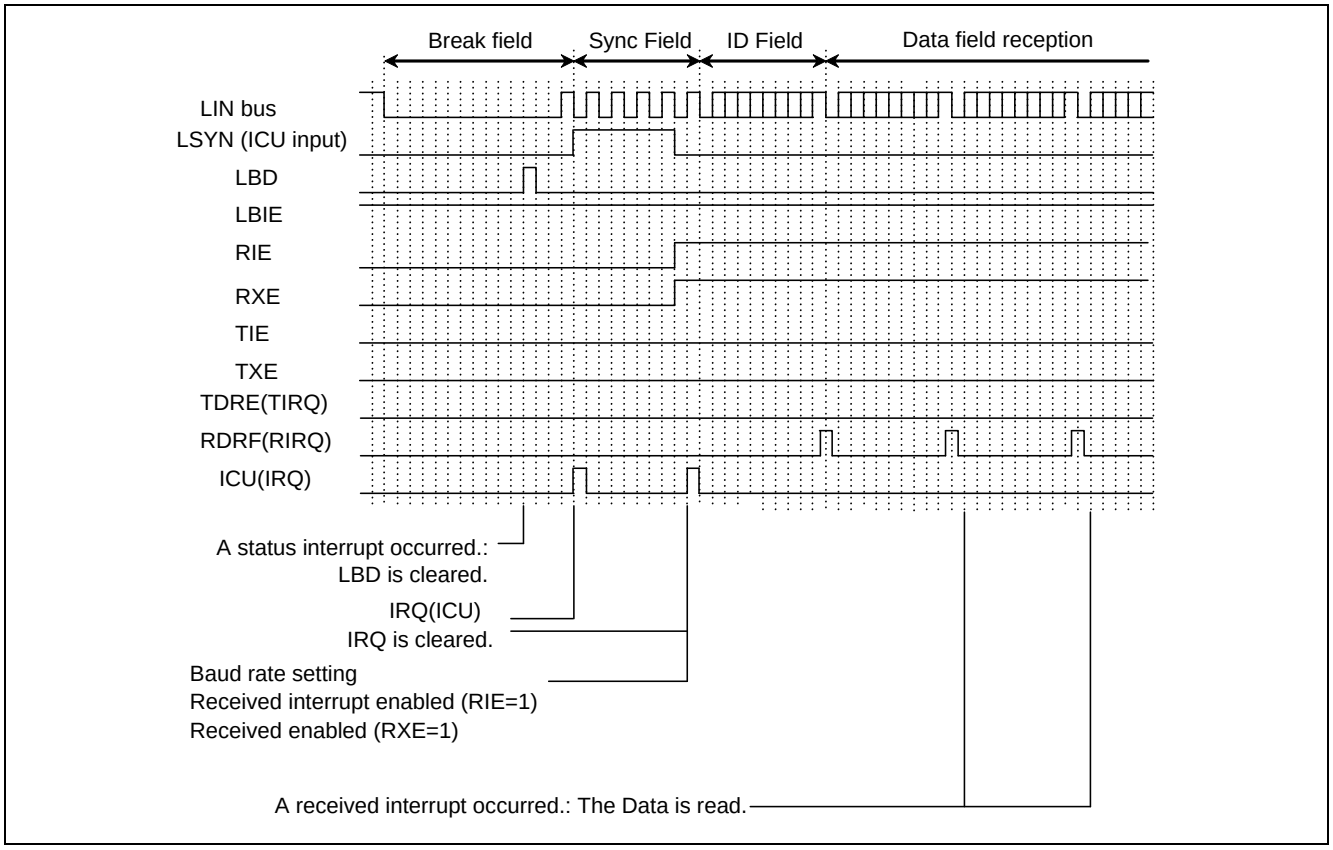


Figure 4-12 LIN Bus Timing (when DATA Field is Received and FIFO is Not Used)



■ If FIFO is used

Figure 4-13 LIN Bus Timing (when DATA Field is Transmitted and FIFO is Used)

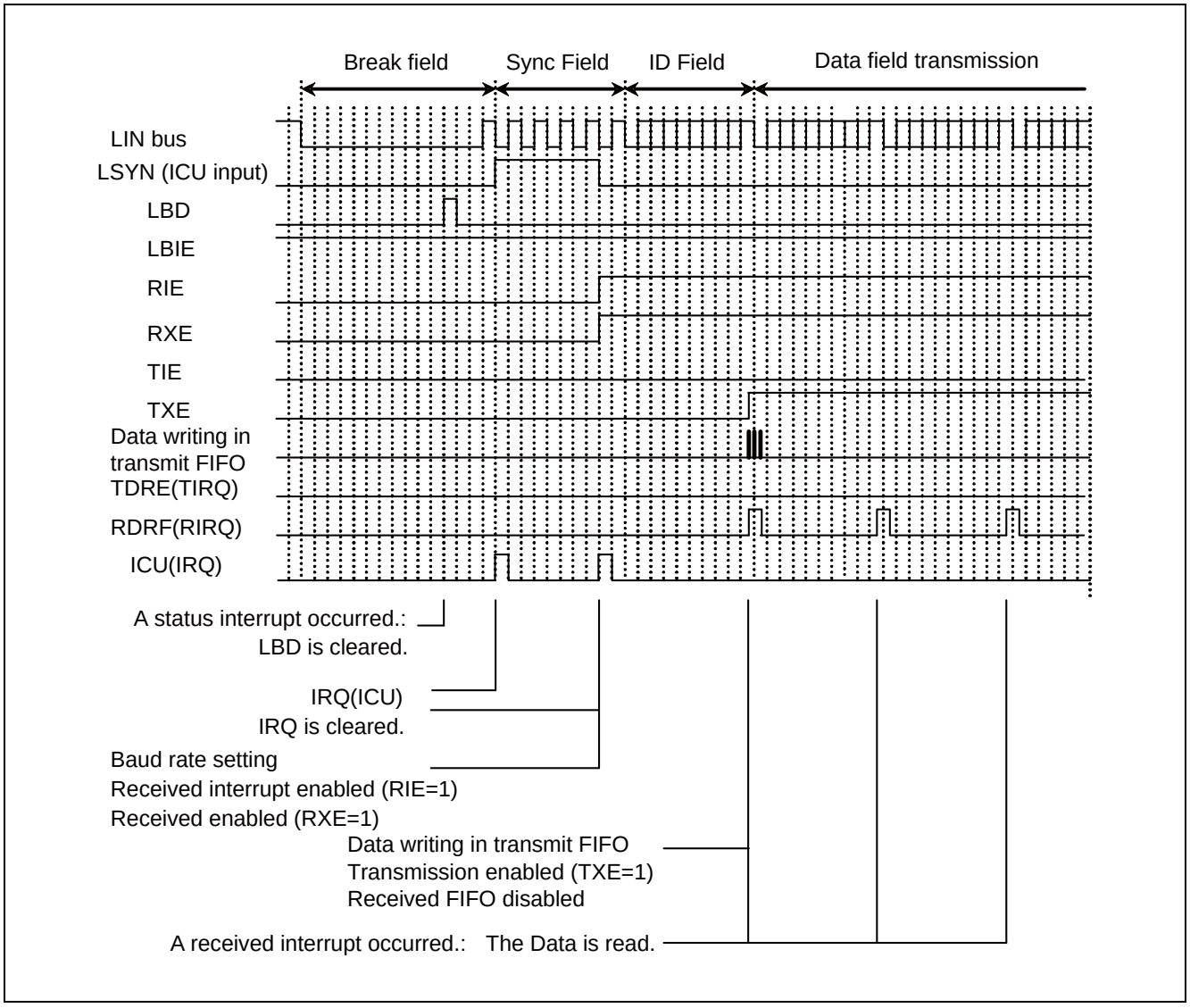
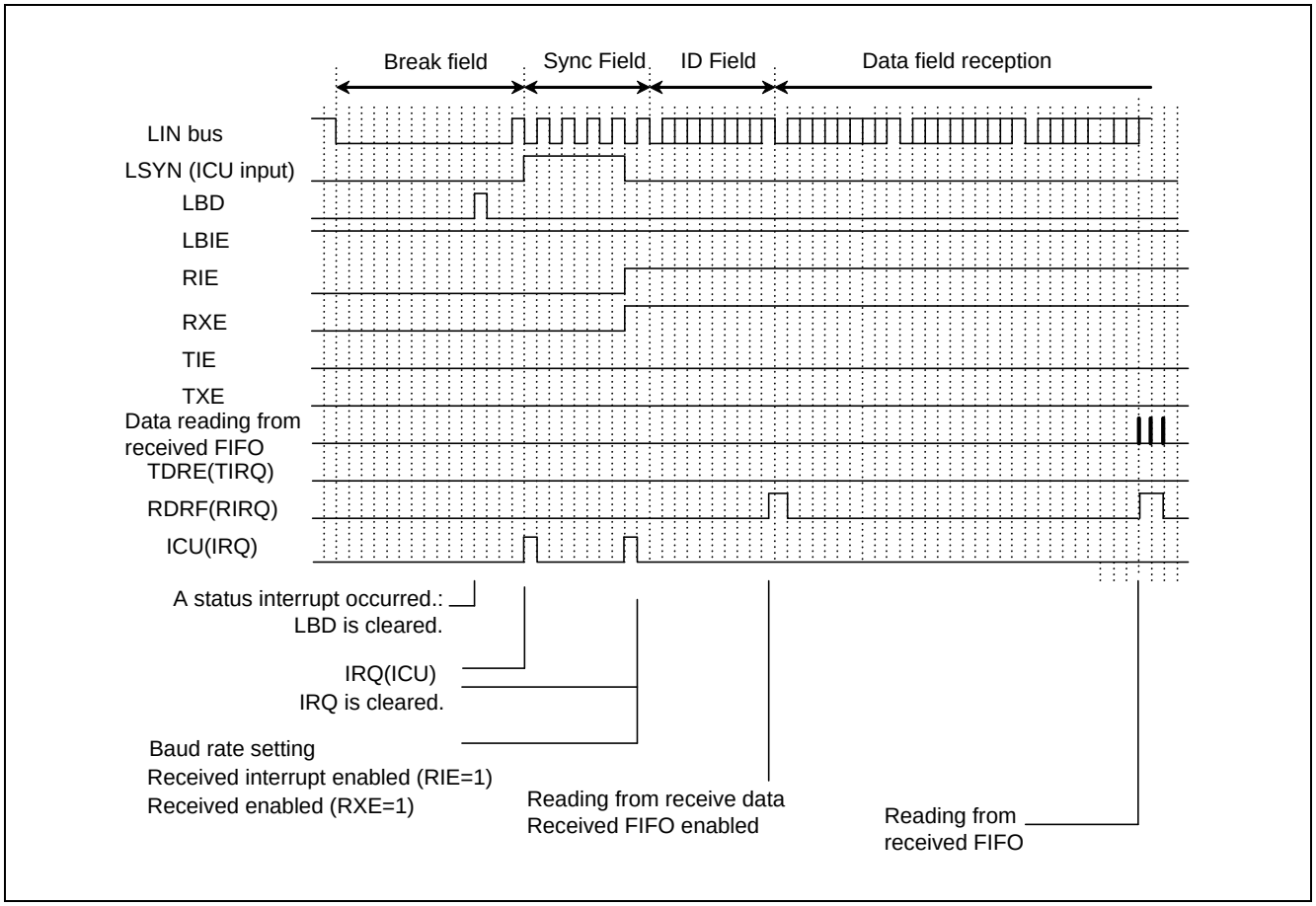


Figure 4-14 LIN Bus Timing (when DATA Field is Received and FIFO is Used)



5. Operation Mode 3 (LIN Communication Mode) Setting Procedure and Program Flow

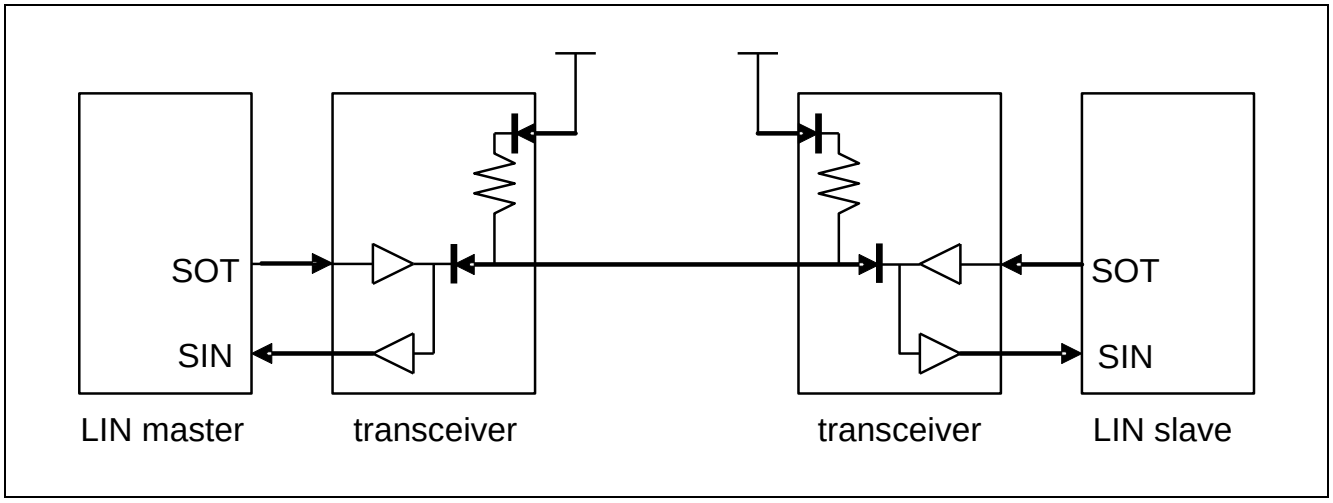
In Operation Mode 3 (LIN communication mode), the LIN interface (Ver. 2.1) can be used for a LIN master or LIN slave system.

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)

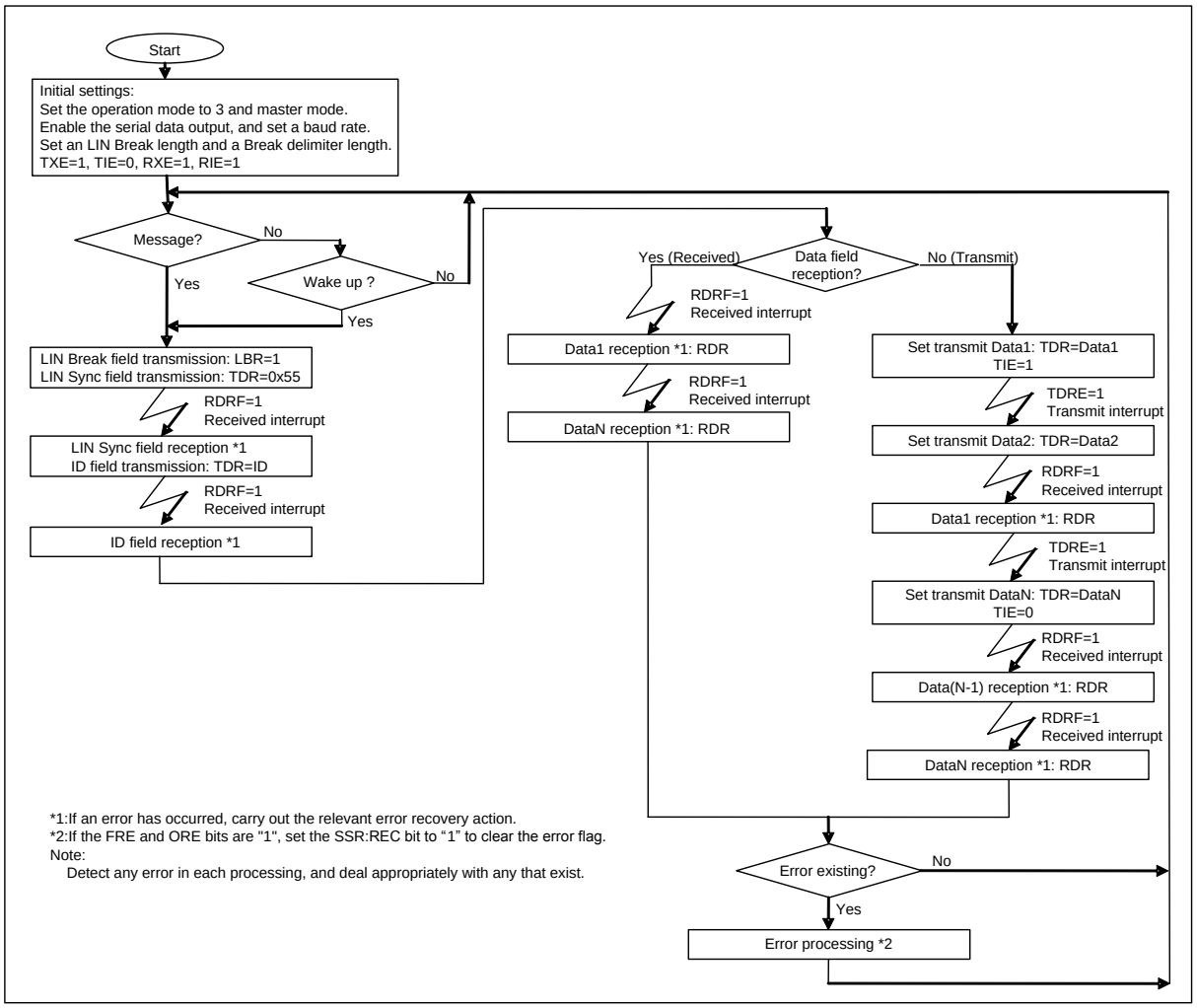
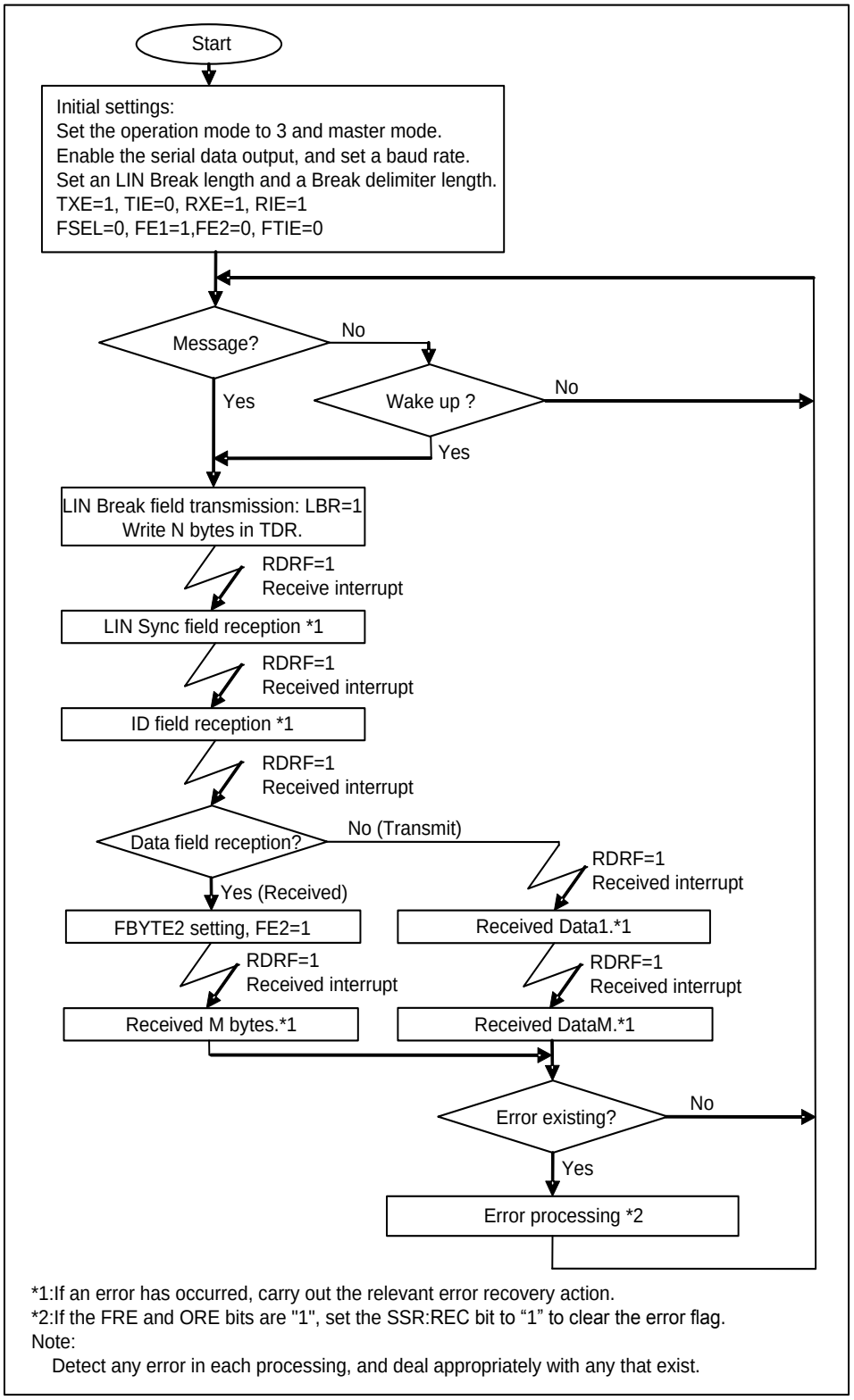


Figure 5-3 Example Flowchart of LIN Communication in Master Mode (when FIFO is Used)



*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)

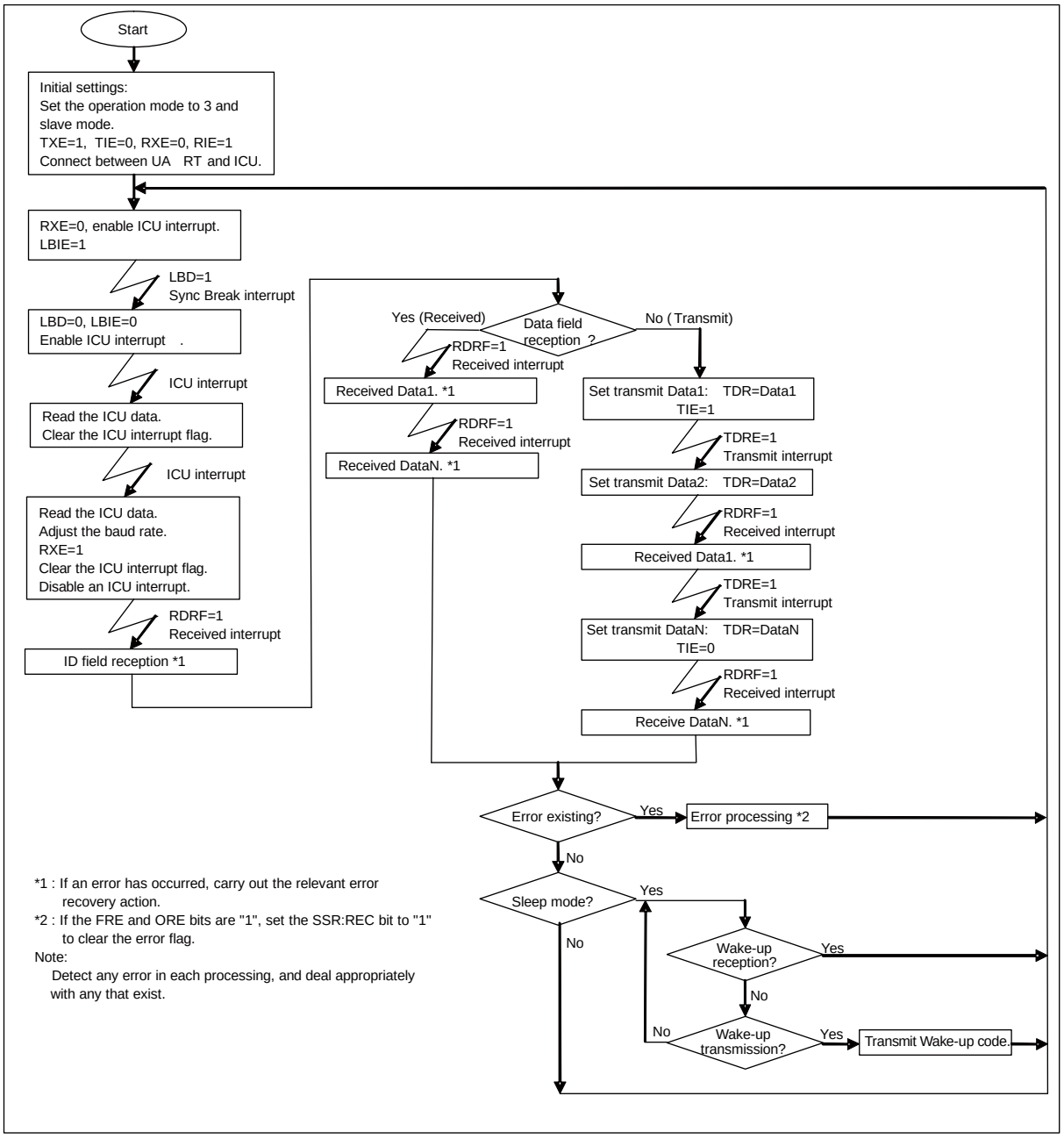
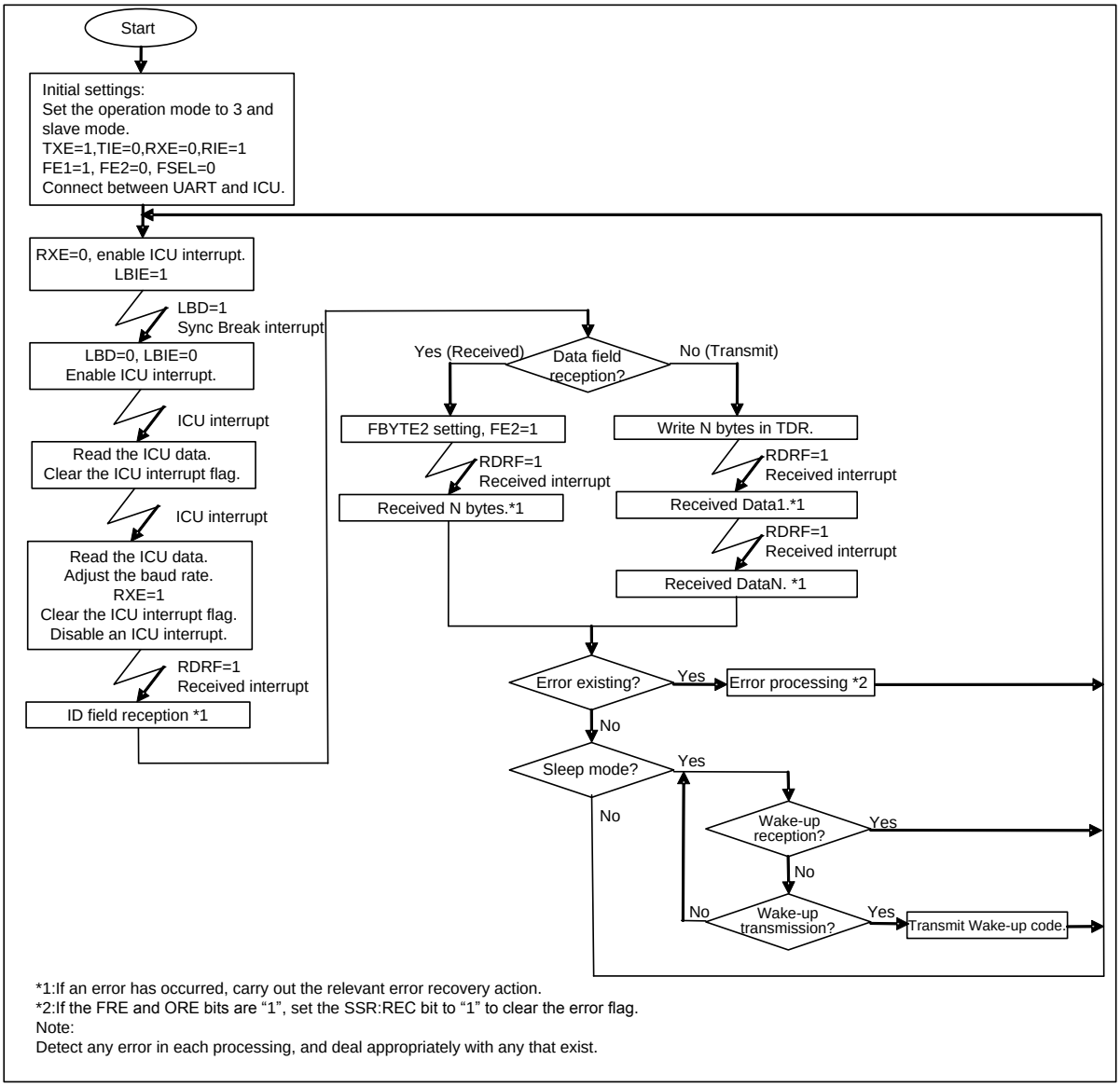


Figure 5-5 Example Flowchart of LIN Communication in Slave Mode (when FIFO is Used)



*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	-	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	Reserved	SBL	Reserved		SOE
Attribute		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 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] Reserved: Reserved bits

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	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 ".

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 ".*

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 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 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 operaion.	"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 6-3 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 6-4 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.*

CHAPTER1-5: 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 Operation

I2C Interface 3. Dedicated Baud Rate Generator

4. I²C Communication Operation Flowchart
5. I²C Interface Registers

CODE: 9BFI2C-E03.0 FM15I-E05.4

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) *1installed.

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 up to 2 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.
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: 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 type.

`

2. I²C Interface Operation

2.1 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 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 is 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".

`

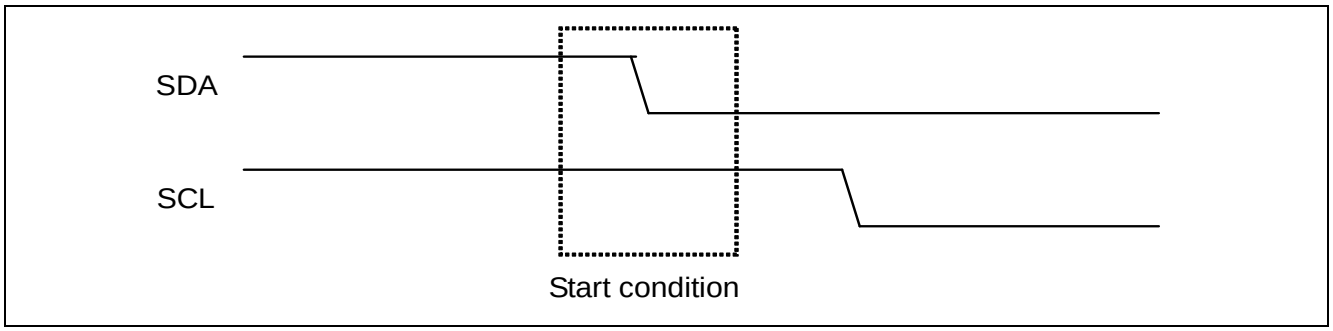
2.2 I²C bus Operation

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.

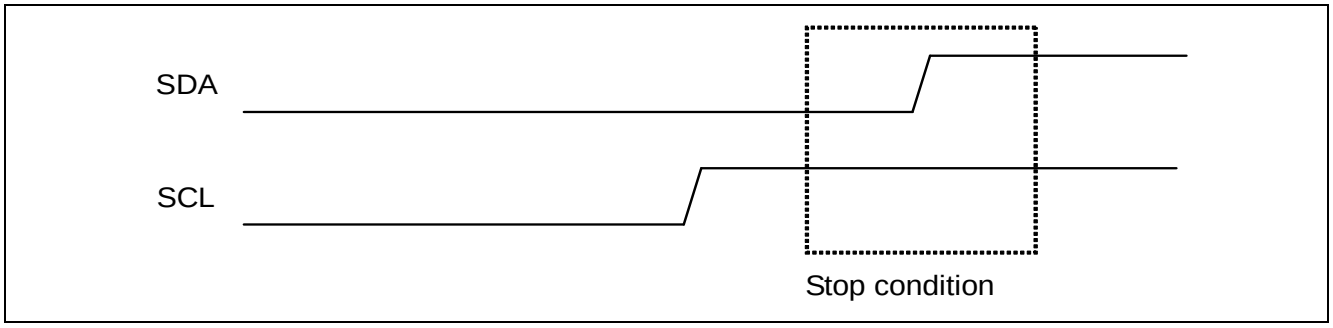
Figure 2-1 Start Condition



I²C Bus Stop Condition

The following shows the I²C bus stop condition.

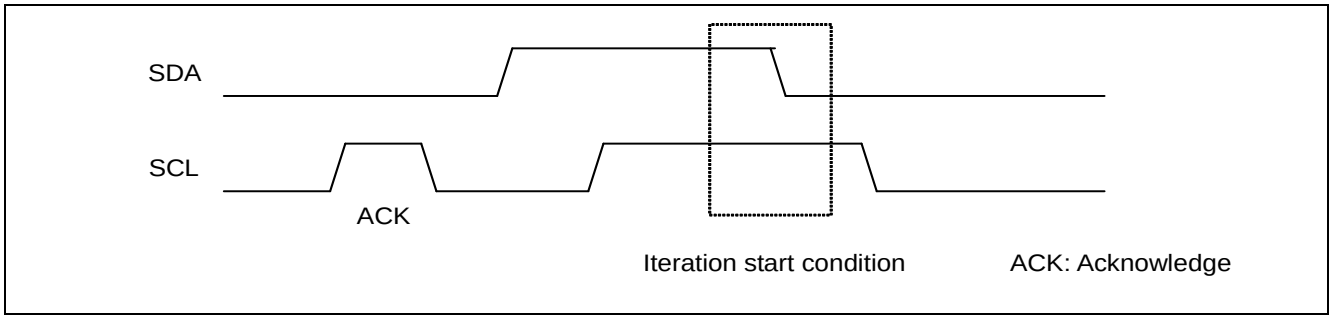
Figure 2-2 Stop Condition



I²C Bus Iteration Start Condition

The following shows the I²C bus iteration start condition.

Figure 2-3 Iteration Start Condition



2.3 Master Mode

Master mode generates the start condition on the I²C bus and outputs clocks to the I²C bus. When the MSS bit in the IBCR register is set to "1" while the I²C 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".

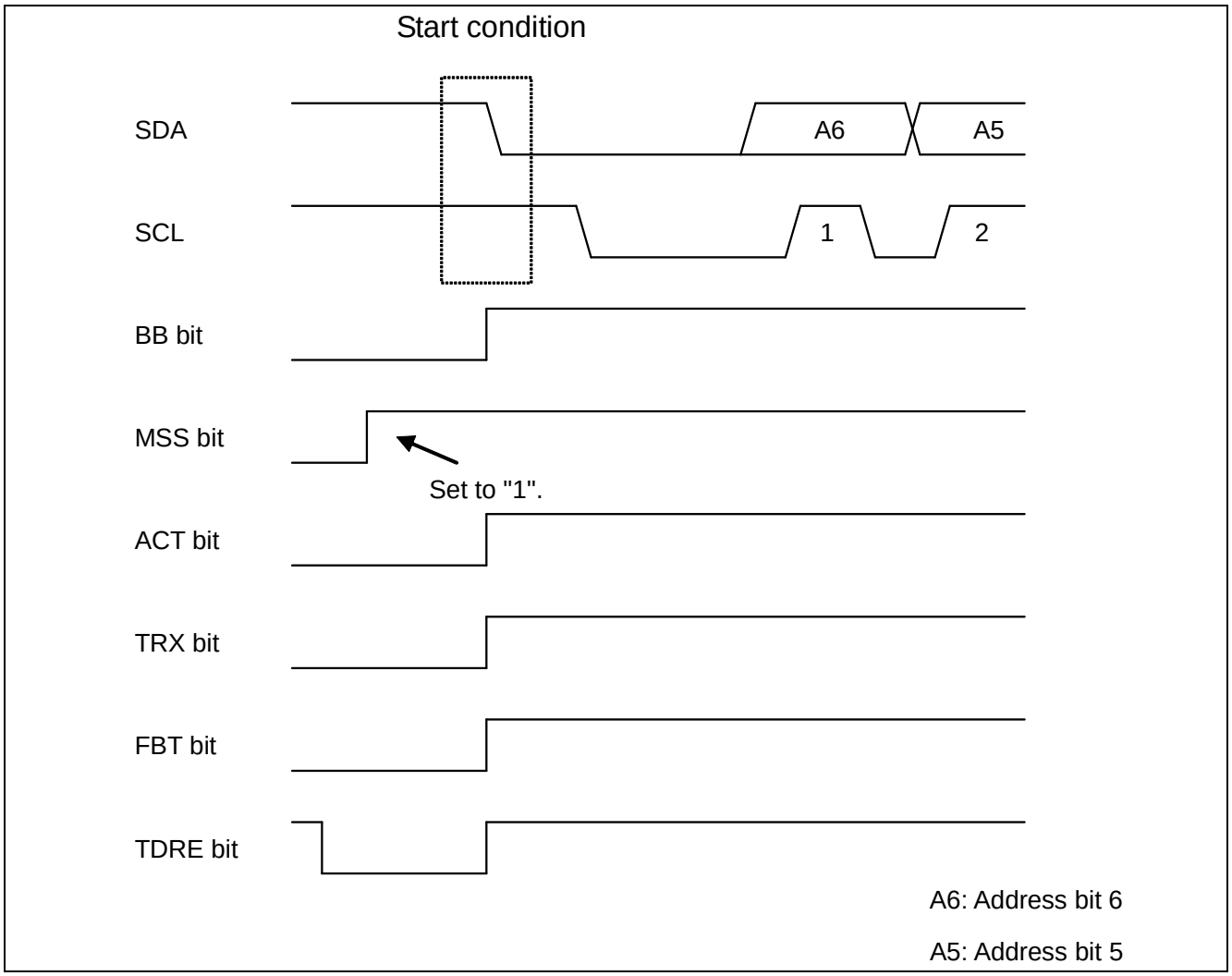
2.3.1 Generating Start Condition

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 I²C 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 I²C bus is carrying out communications. (See Figure 2-4.)

Figure 2-4 Start Condition Output and Relationships with Respective Bits



`

Note:

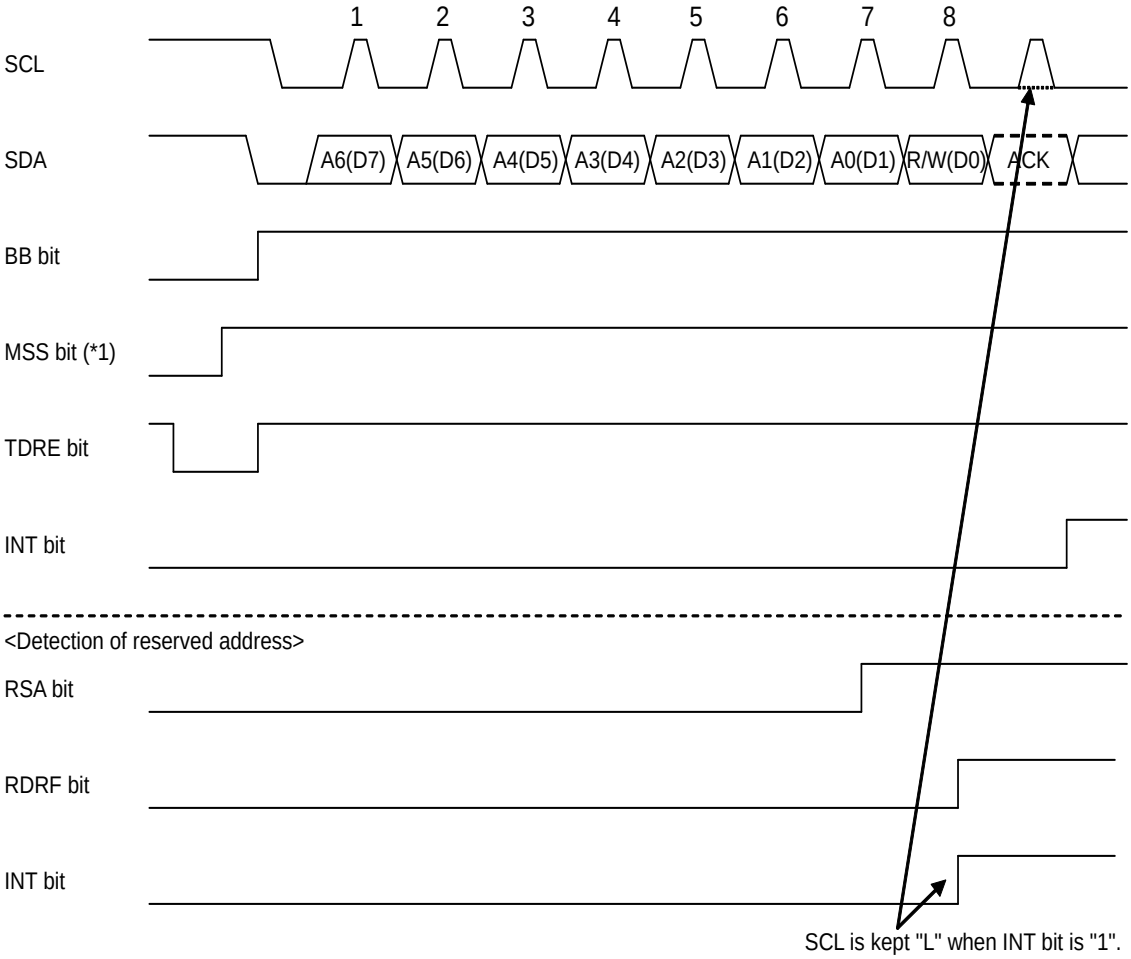
- *In operation mode 4 (I^2C mode), the bus clock is used at a frequency no lower than 8 MHz. Also note that setting of a baud rate generator that exceeds 400 kbps is prohibited.*

2.3.2 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

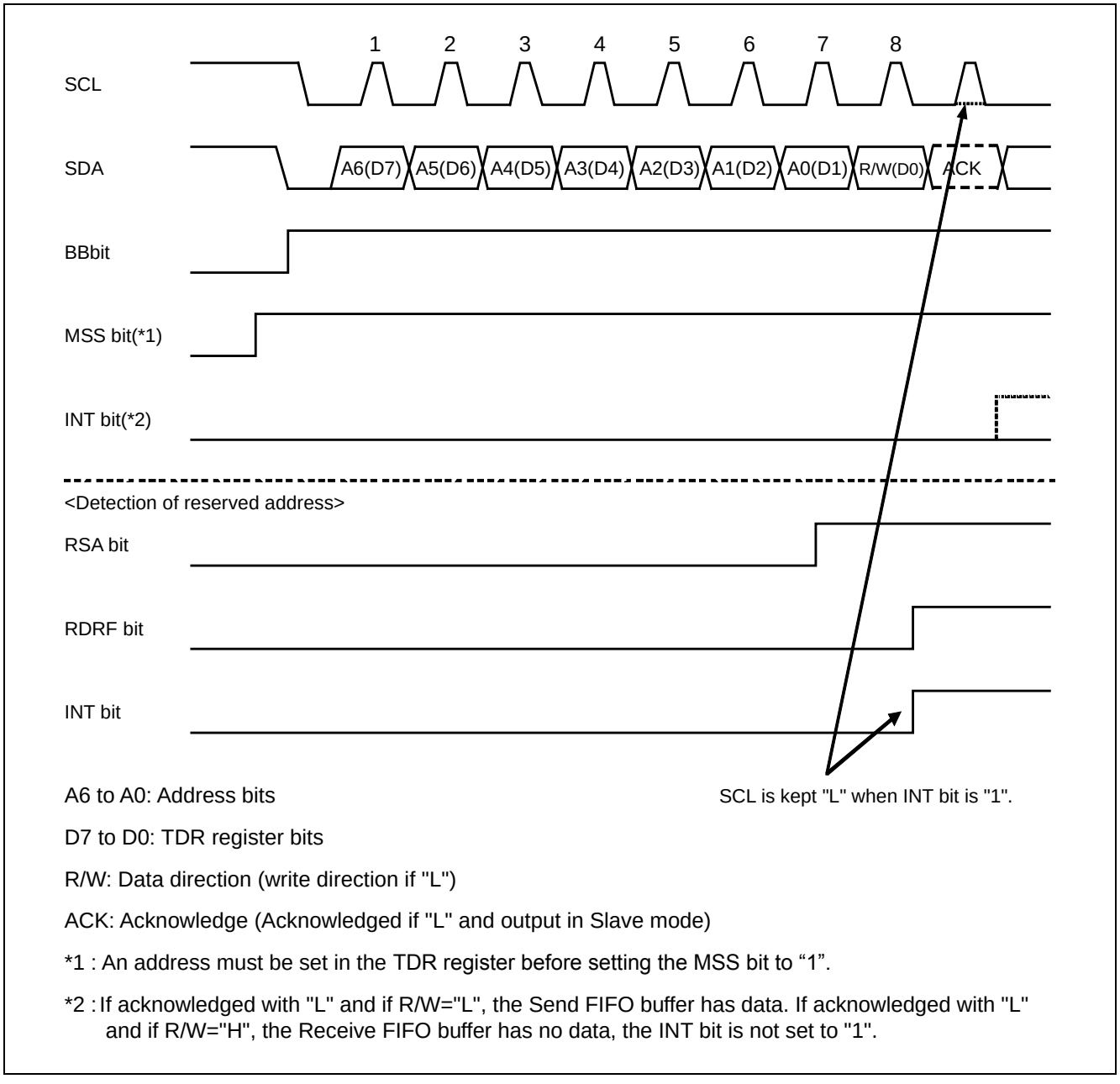
R/W: Data direction (write direction if "L")

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".

2.3.3 Acknowledgement Reception by First Byte Transmission

When the data direction bit (R/W) is output, the I^{2}C 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.

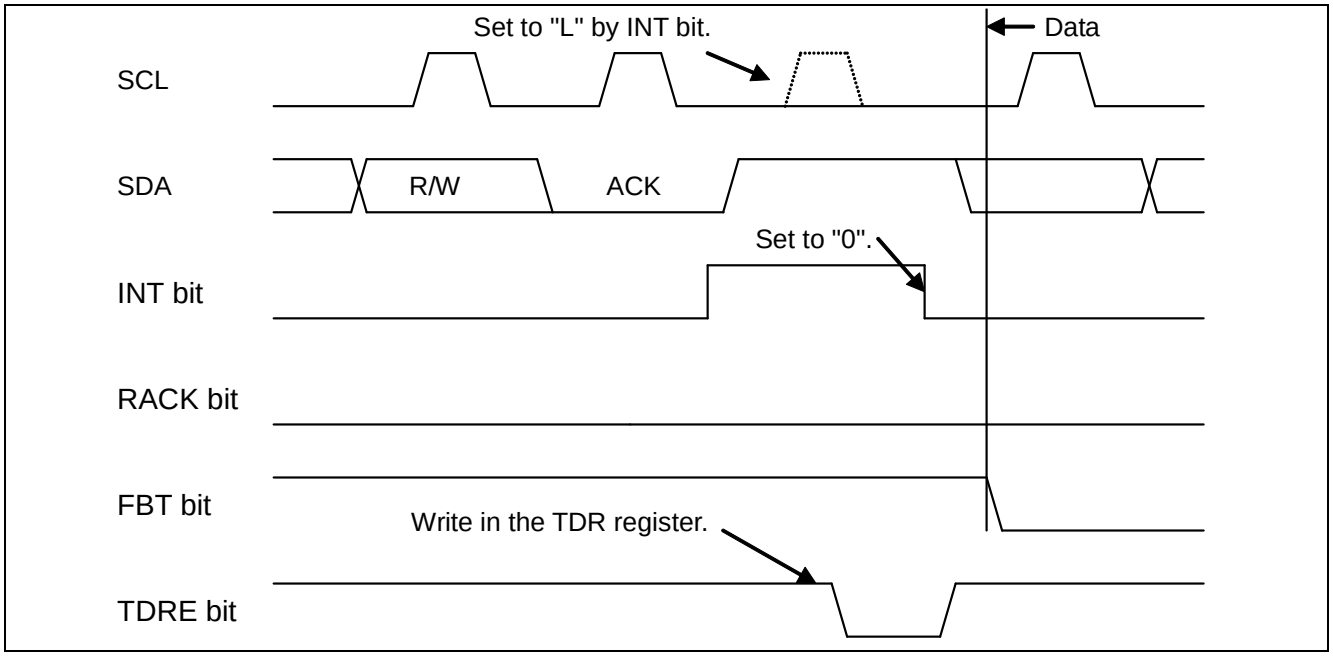
When DMA Mode is Enabled (SSR:DMA=1)

- 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)

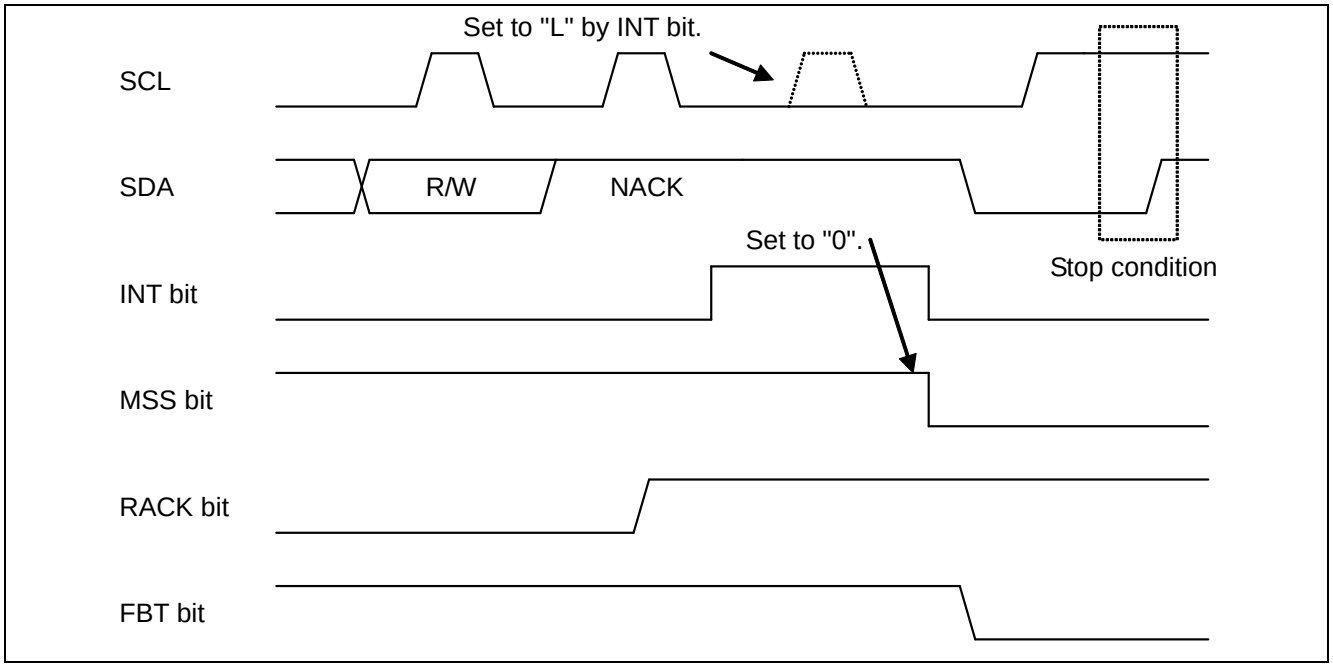


The following describes the wait timing for an address.

- 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)

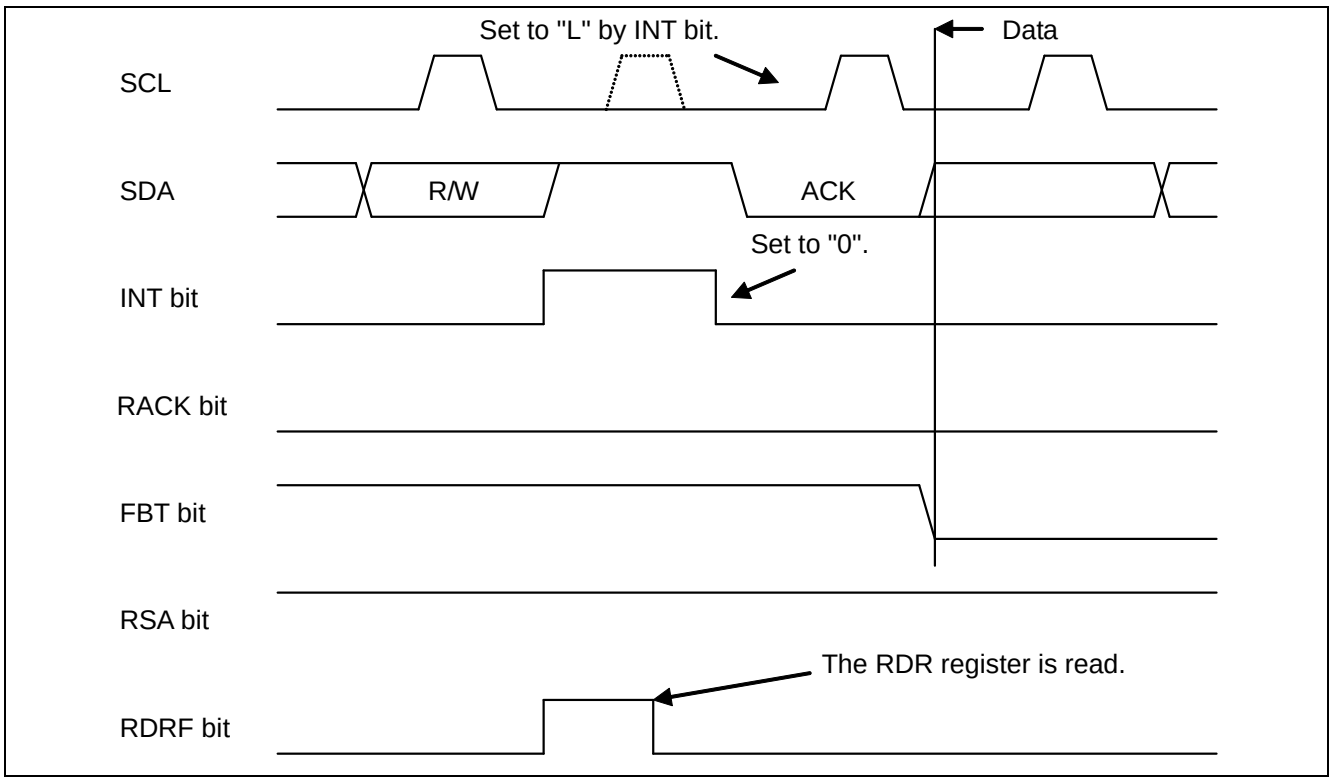
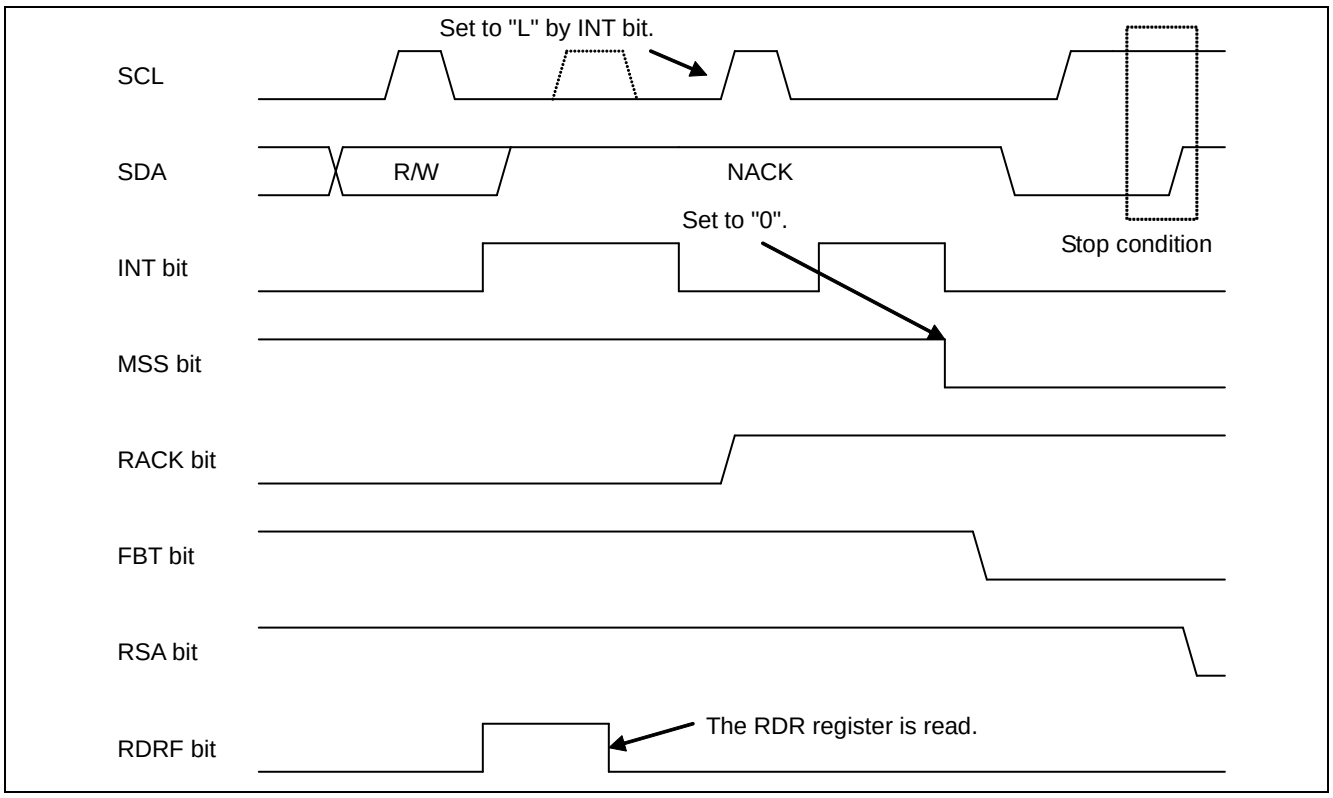
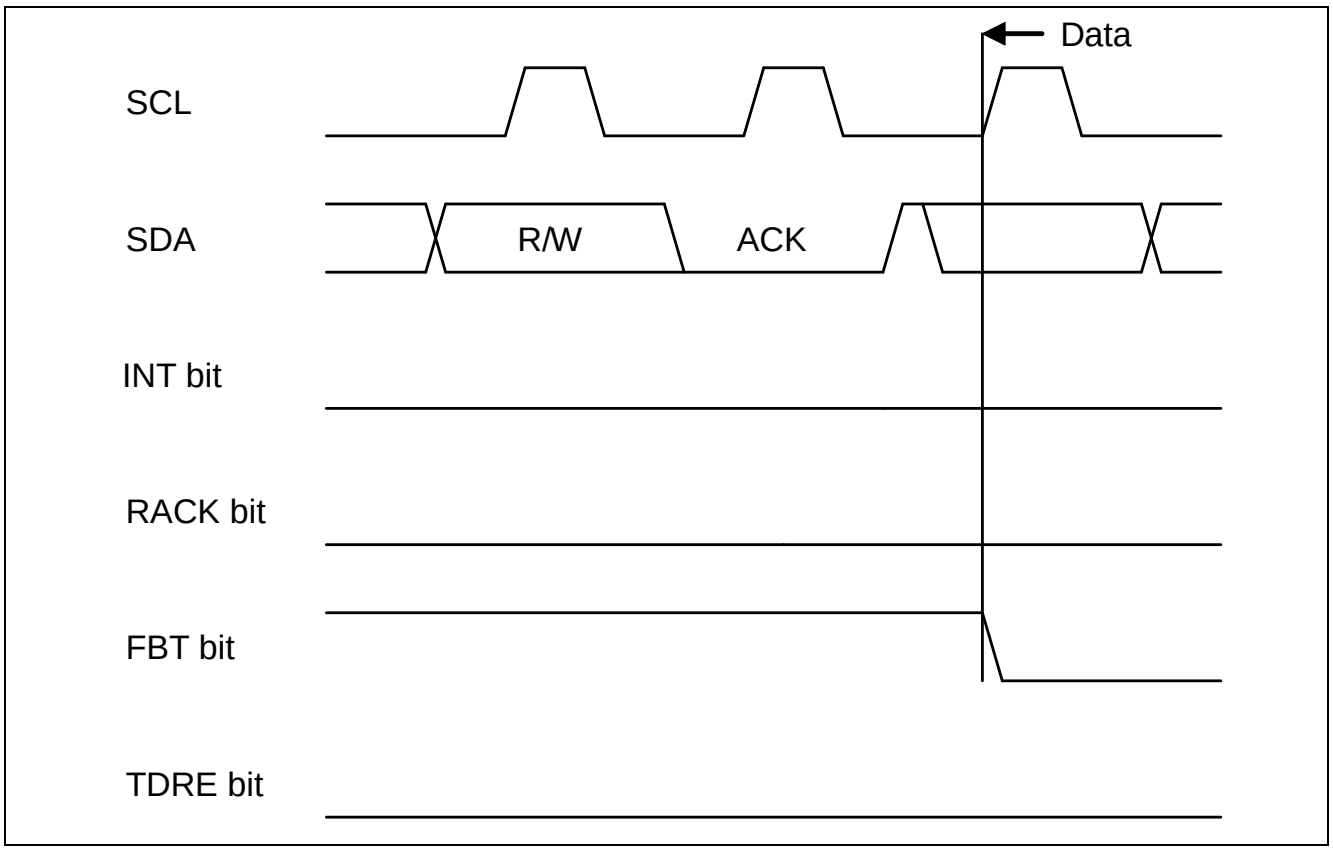


Figure 2-10 Acknowledgement
(when FIFO is Disabled, IBSR:RSA="1", and NACK Response is Selected)



`

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)



2.3.4 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.

`

2.3.4.1 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.

`

2.3.4.2 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^{2}C 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^{2}C 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^{2}C 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.

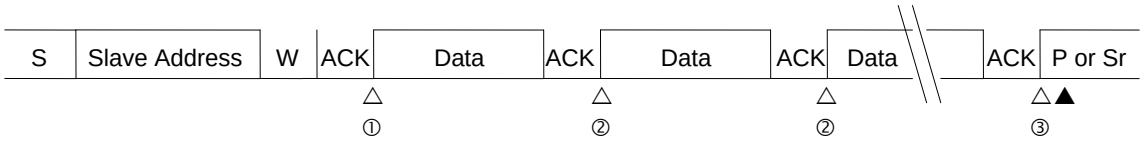
1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

`

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 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".

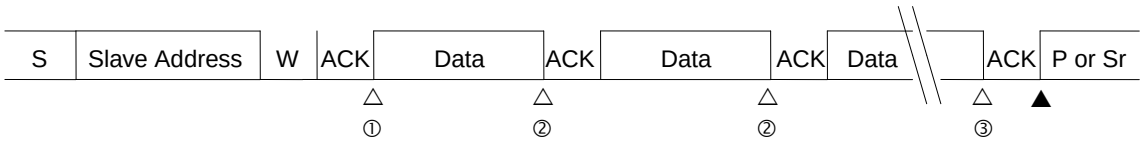
② An interrupt occurs when a single byte is sent and an ACK is received.
- The send data is written in the TDR register, and the INT bit is set to "0".

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".

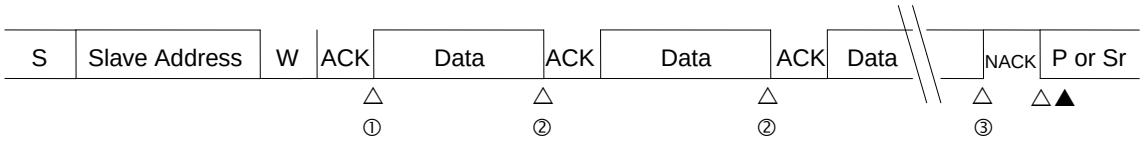
`

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".

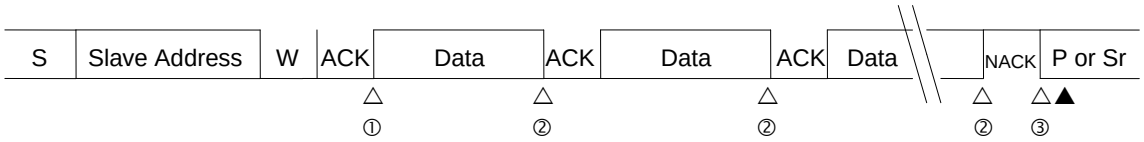
② An interrupt occurs when a single byte is sent.
- The send data is written in the TDR register, and the INT bit is set to "0".

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)



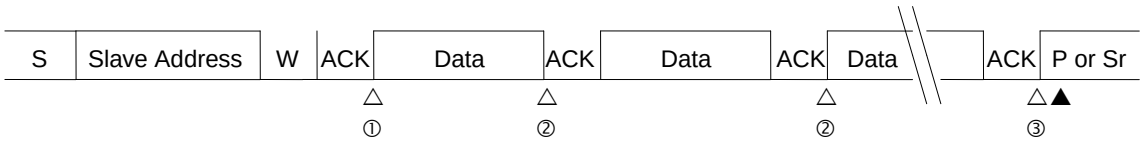
S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△: Interrupt by INTE="1"
▲: Interrupt by CNDE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- The send data is written in the TDR register, 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".

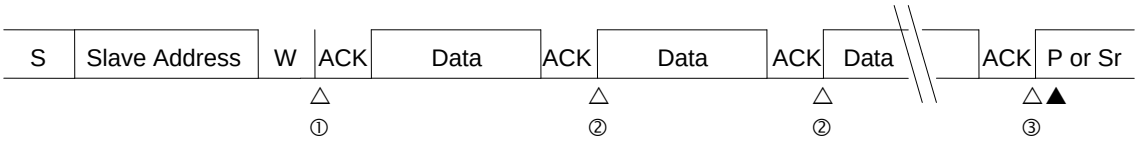
② An interrupt occurs when a single byte is sent.
- The send data is written in the send buffer, and both WSEL and INT bits are set to "0".

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".

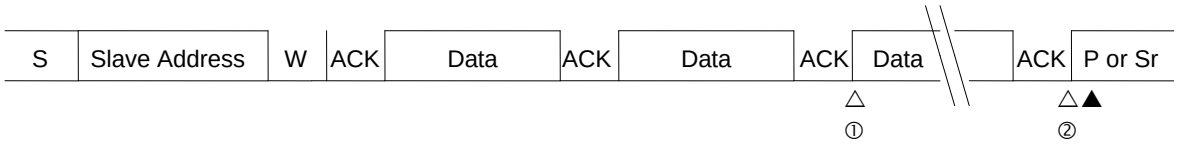
`

Figure 2-17 Master Mode 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".

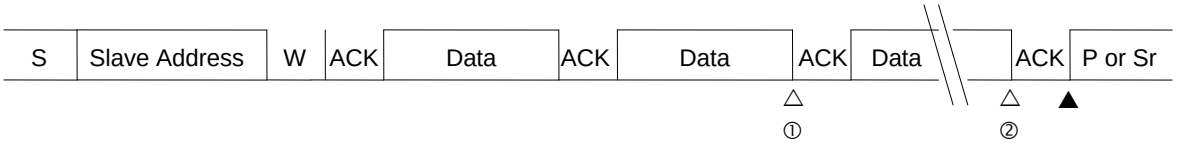
Figure 2-18 Master Mode Transmit Interrupt 7 by Enabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



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")

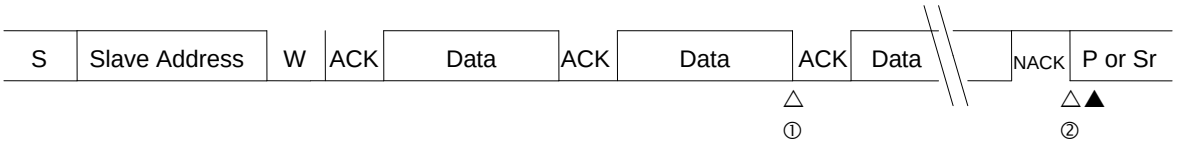


S: Start condition
W: Data direction bit (write direction)
P: Stop condition
Sr: Iteration start condition
△: Interrupt by INTE="1"
▲: Interrupt by CNDE="1"
① An interrupt occurs if the Send FIFO buffer is emptied.
- The send data is written in the Send FIFO buffer, and INT bit is set to "0".

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".

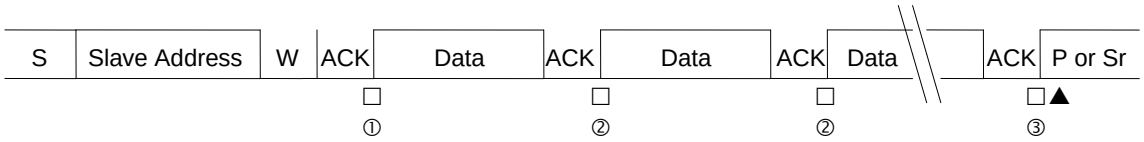
`

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".

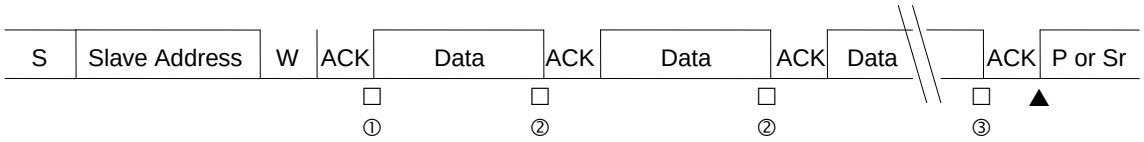
Figure 2-21 Master Mode Interrupt 10 by Disabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



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.

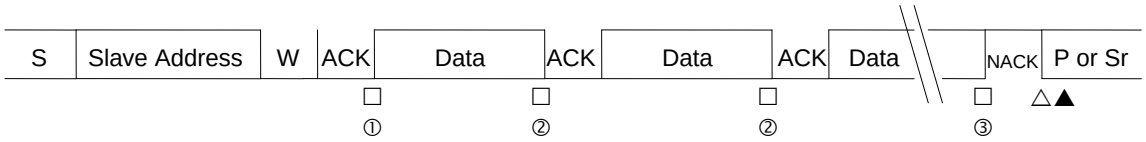
- The send data is written in the TDR register.

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".

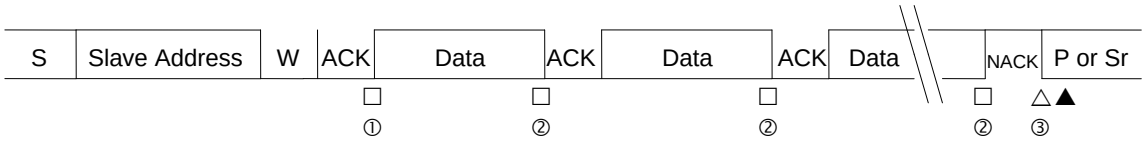
Figure 2-23 Master Mode Transmit Interrupt 12 by Disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)



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".

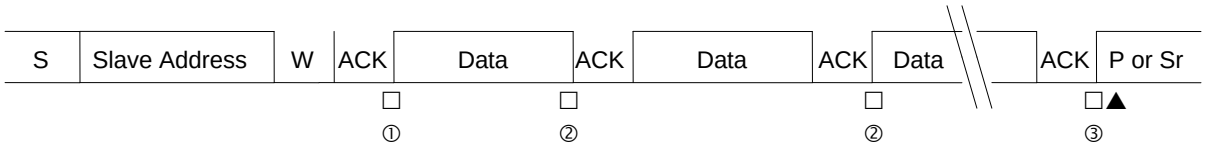
`

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.

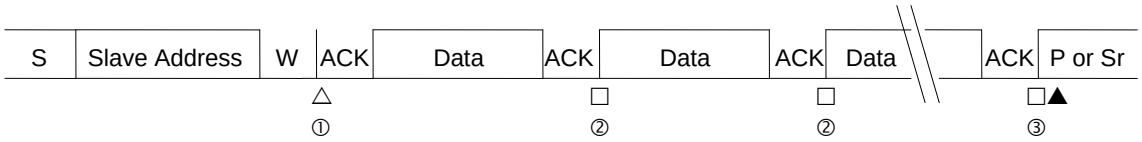
③ An interrupt occurs when a single byte is sent.

- 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".

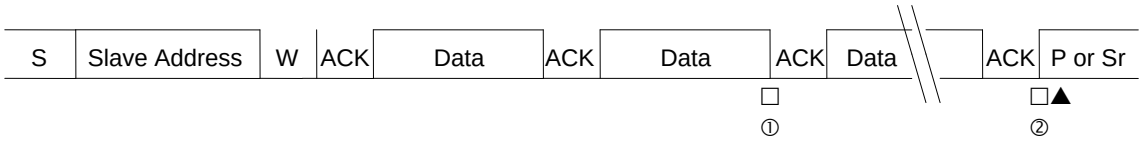
`

Figure 2-26 Master Mode 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".

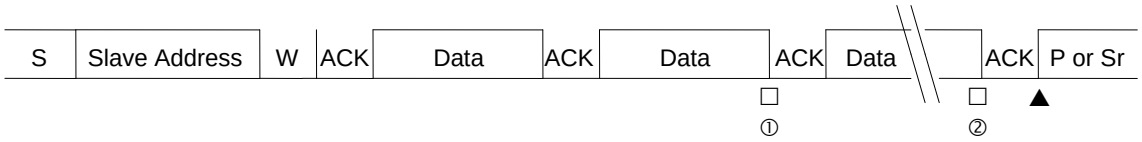
Figure 2-27 Master Mode Transmit Interrupt 16 by Enabling FIFO (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



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".

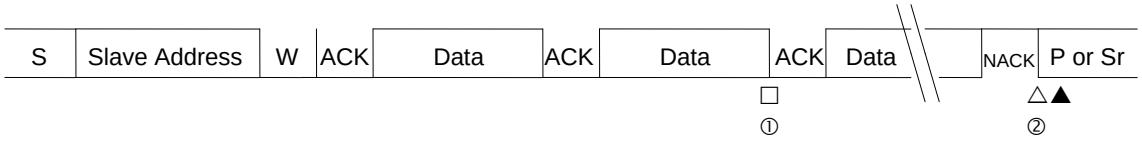
`

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".

2.3.5 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^{2}C 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^{2}C 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.

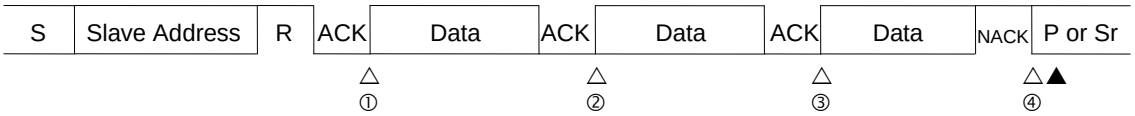
1. Set the IBCR:INT bit to "1".
2. Check that the IBCR:INT bit is set to "1".
3. Write the slave address in the TDR.
4. Set the IBCR:SCC bit to "1".

*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^{2}C 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".

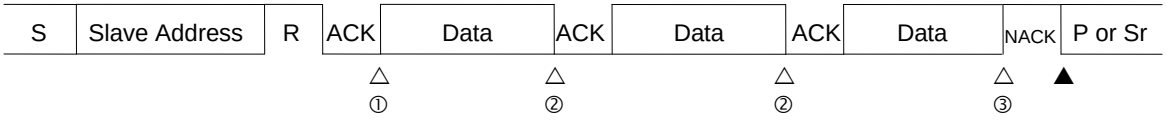
② An interrupt occurs when a single byte is received and an ACK is sent.
- After the received data has been read, the INT bit is set to "0".

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".

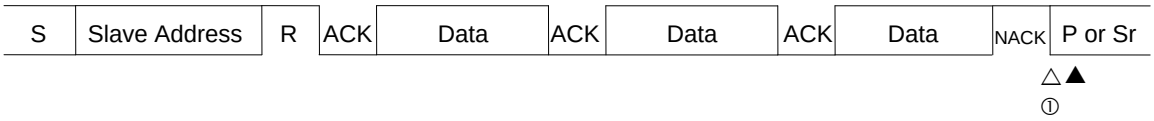
② An interrupt occurs when a single byte is received.
- After the received data has been read, the INT bit is set to "0".

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".

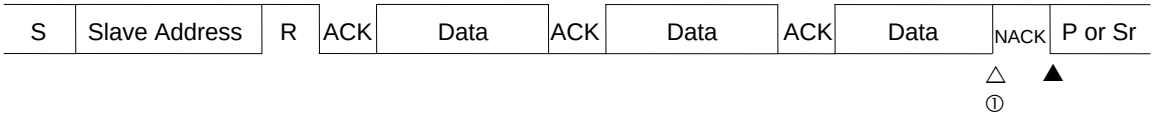
`

Figure 2-32 Master Mode Received Interrupt 3 by Enabling FIFO (SSR:DMA="0", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")



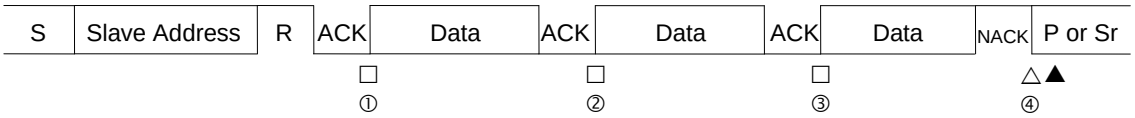
△: Interrupt by INTE="1"
▲: Interrupt by CNDE="1"
① An interrupt occurs if TDRE bit is set to "1".
- The entire data is read from the Received FIFO buffer, and MSS bit is set to "0" or both MSS and SCC bits are set to "1".

Figure 2-33 Master Mode Received Interrupt 4 by Enabling FIFO (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



△: Interrupt by INTE="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".

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.

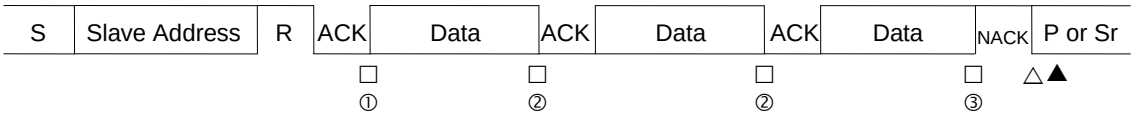
③ An interrupt occurs when a single byte is received and an ACK is sent.
- After the received data has been read, the ACKE bit are set to "0" and the dummy data is written in the TDR register.

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".

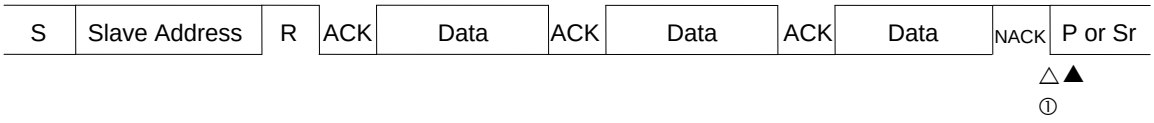
`

Figure 2-35 Master Mode Received Interrupt 6 by Disabling FIFO (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



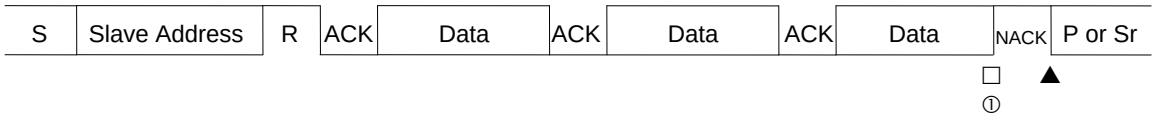
△: Interrupt by INTE="1"
▲: Interrupt by CNDE="1"
□: Interrupt by TBIE="1"
① An interrupt occurs when the slave address is sent, the direction bit is sent, and an ACK is received.
- Dummy data is written in the TDR register.
② An interrupt occurs when a single byte is received.
- After the received data 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".

2.3.6 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.

2.3.7 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

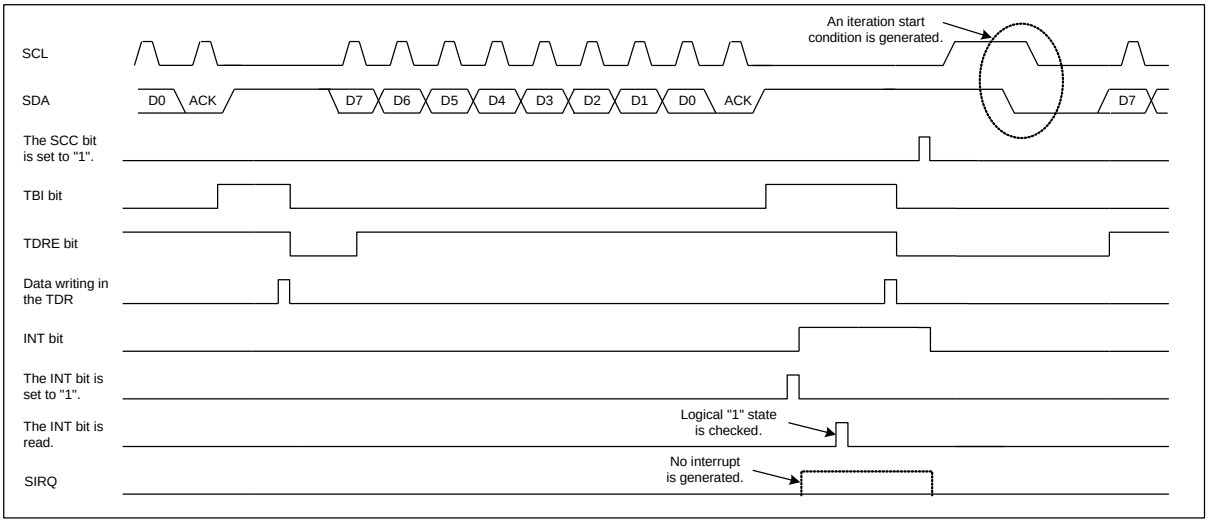
Refer to the IBCR:MSS bit and IBCR:ACT bit to check if master mode is in the wait state or not (in the wait state if the IBCR:MSS=1 and IBCR:ACT=0). After setting the IBCR:MSS bit to "1" and to operate in slave mode, set the IBSR:AL bit to "1", the IBCR:MSS bit to "0", and the IBCR:ACT bit to "1".

2.3.8 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)



2.4 Slave Mode

If a start condition or repeated start condition is detected, the combination of the ISBA and ISMK register values matches the received address, the multifunction serial interface outputs an ACK response and acts as a slave.

<Note>

- *When EIBCR:BEC set to "0", if a second start condition is detected after a first start condition has been detected (while the address field (first byte) or bits 2 to 9 of data field are being transferred), a bus error (IBCR:BER=1) is flagged and reception stops. If this happens, the master must retransmit a start condition after the interrupt flag (IBCR:INT) of multifunction serial interface has been cleared.*

2.4.1 Slave Address Match Detection

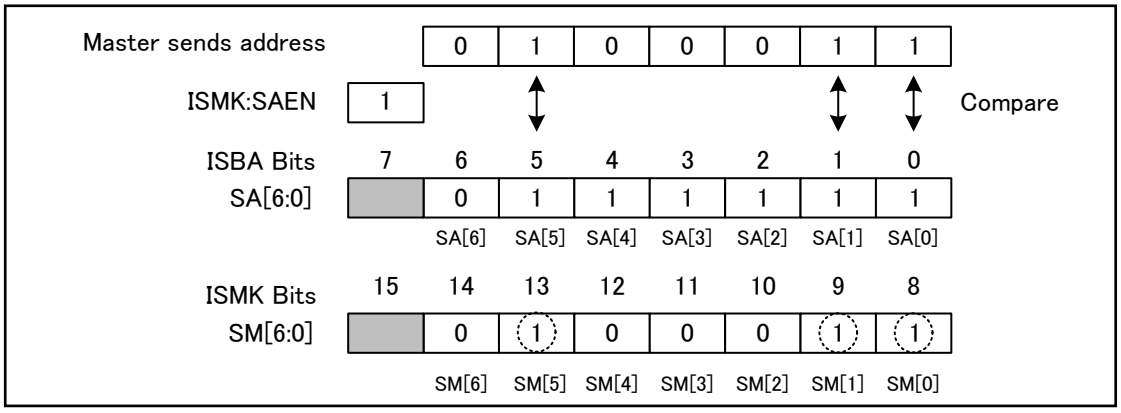
The 7-bit slave address and the direction of a data transfer is contained in the first byte after detection of a start or repeated start condition. The ISMK becomes the value to mask the slave address: a zero mask value designates a don't care, and a 1 must be a direct match. In other words, if a mask bit is set to 0 in the ISMK register, the address bit is not compared.

The SAEN is the enables the slave address detection when set. The address that is sent from master is compared with the slave address bits (SA[6:0]) that sets the mask bits (SM[6:0]) to "1". If they match, an ACK is output. If there is no match, or SAEN is 0, no ACK is output.

■ Example of a slave address detection

Master addresses slave address 0x23.

Figure 2-39 Example of slave address detection.



Only SA5, SA1, and SA0 are compared to the address sent by master because the SM[6] and SM[4:2] are zero and therefore are don't care. the multifunction serial interface outputs an ACK response.

`

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.

2.4.2 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.

`

2.4.3 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)

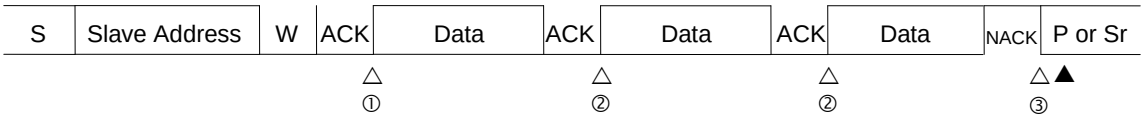
■ When received FIFO is disabled:

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 I²C bus is placed into the wait state. When received FIFO becomes full, place the I²C 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 I²C 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 I²C bus. When received FIFO is full, release the wait state of the I²C 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 2-40 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 .

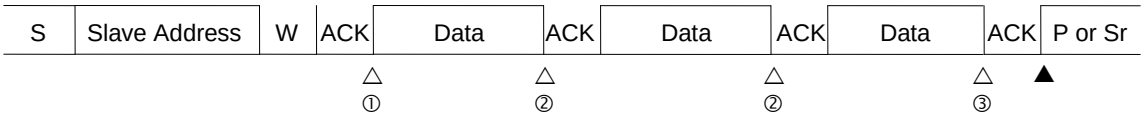
- After the received data has been read from the received buffer, the INT bit is set to "0".

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 2-41 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".

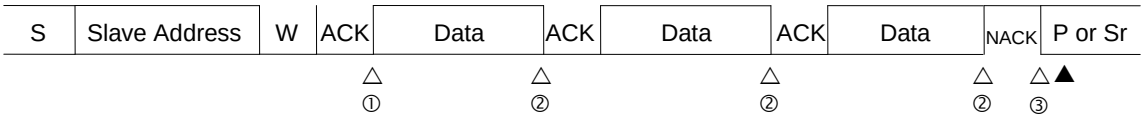
② An interrupt occurs when a single byte is received.

- 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 2-42 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".

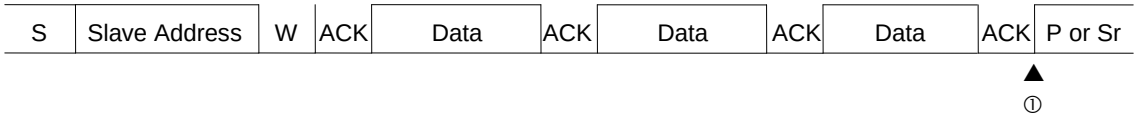
② An interrupt occurs when a single byte is received.

- 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 2-43 Slave Mode Received Interrupt 4 by Enabling Received FIFO (SSR:DMA="0", IBSR:RSA="0")



△ :Interrupt by INTE="1"

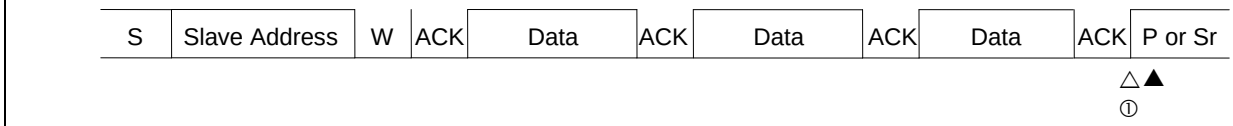
▲ :Interrupt by CNDE="1"

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 2-44 Slave Mode Received Interrupt 5 by Enabling Received FIFO (SSR:DMA="0", IBSR:RSA="0")



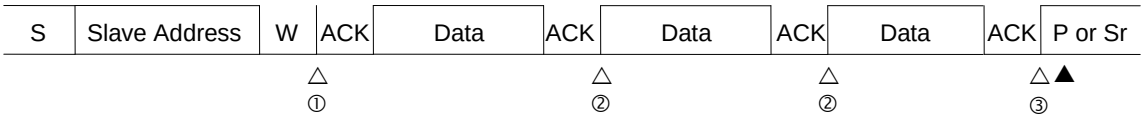
△ :Interrupt by INTE="1"

▲ :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 2-45 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".

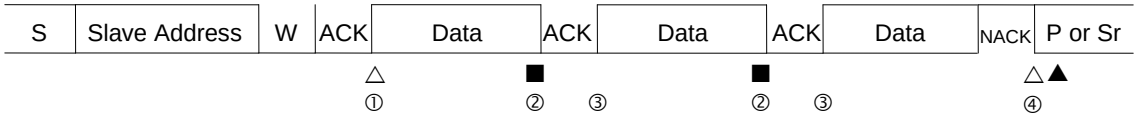
② An interrupt occurs when a single byte is received and an ACK is output .

- 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 2-46 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.

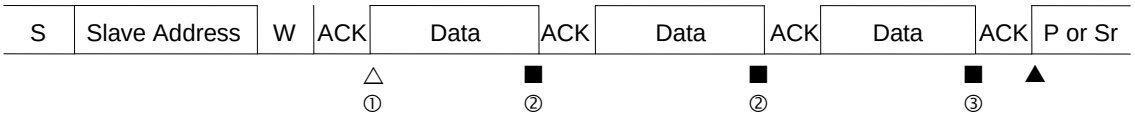
- The received data is read from the received buffer

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 2-47 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".

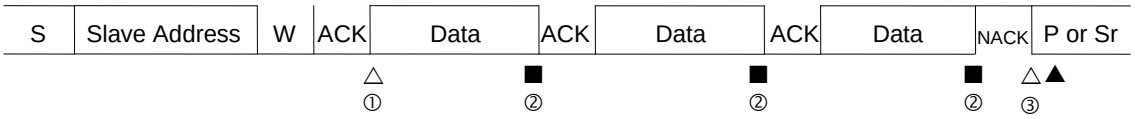
② An interrupt occurs when a single byte is received.

- 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 2-48 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".

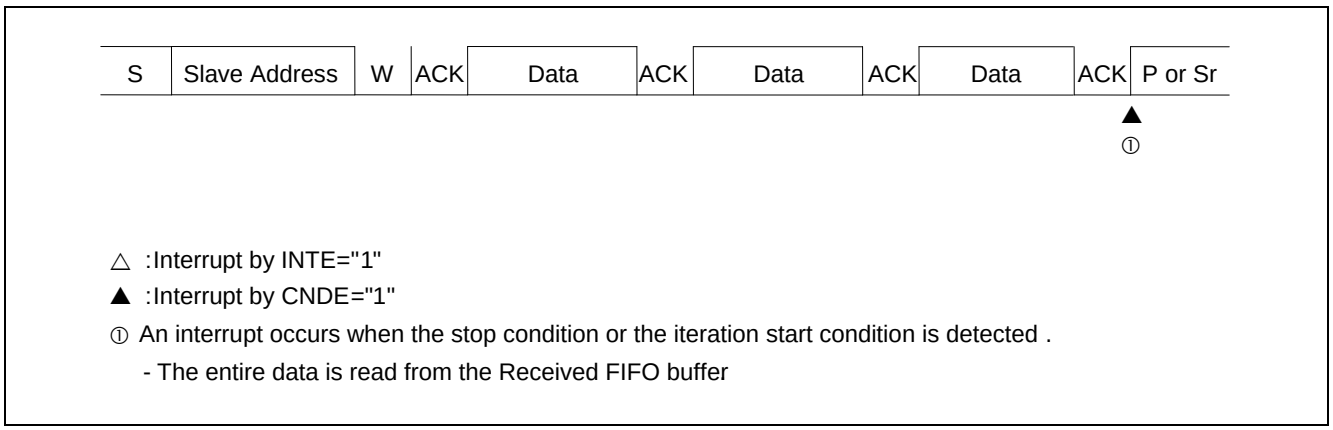
② An interrupt occurs when a single byte is received.

- 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 2-49 Slave Mode Received Interrupt 10 by Enabling Received FIFO (SSR:DMA="1", IBSR:RSA="0")



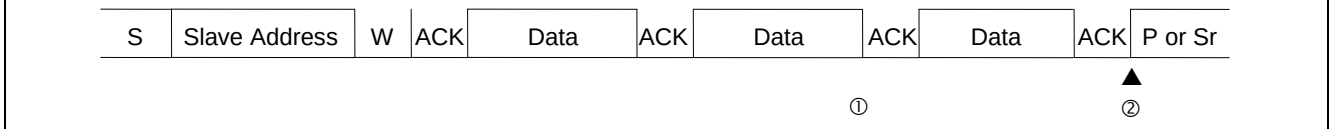
△ :Interrupt by INTE="1"

▲ :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 2-50 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.

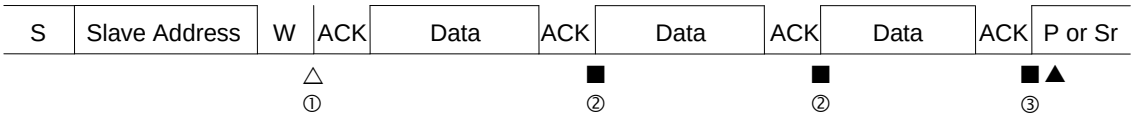
- The waiting is released when data is read even once from the Received FIFO buffer

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 2-51 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.

2.4.4 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.

2.5 Bus Error

If the stop or (iteration) start condition is detected while transmitting or receiving data on the I^{2}C bus, it is handled as a bus error.

2.5.1 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.

2.5.2 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^{2}C 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.

3.1 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.

3.2 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.*

3.3 Reload Values and Baud Rates for Each Bus Clock Frequency

The following shows the reload values and baud rate setting examples.

Table 3-1 Reload Values and Baud Rate Setting Examples 1

Baud Rate [bps]	8 MHz Value	10 MHz Value	16 MHz Value	20 MHz Value	24 MHz Value
400000	19	24	39	49	59
200000	39	49	79	99	119
100000	79	99	159	199	239

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.

Table 3-2 Reload Values and Baud Rate Setting Examples 2

Baud Rate [bps]	32 MHz Value	40 MHz Value
400000	79	99
200000	159	199
100000	319	399

The numeric values above are available when the SCL rising timing of the I^2C bus is 0 s. 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.

3.4 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).

3.5 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

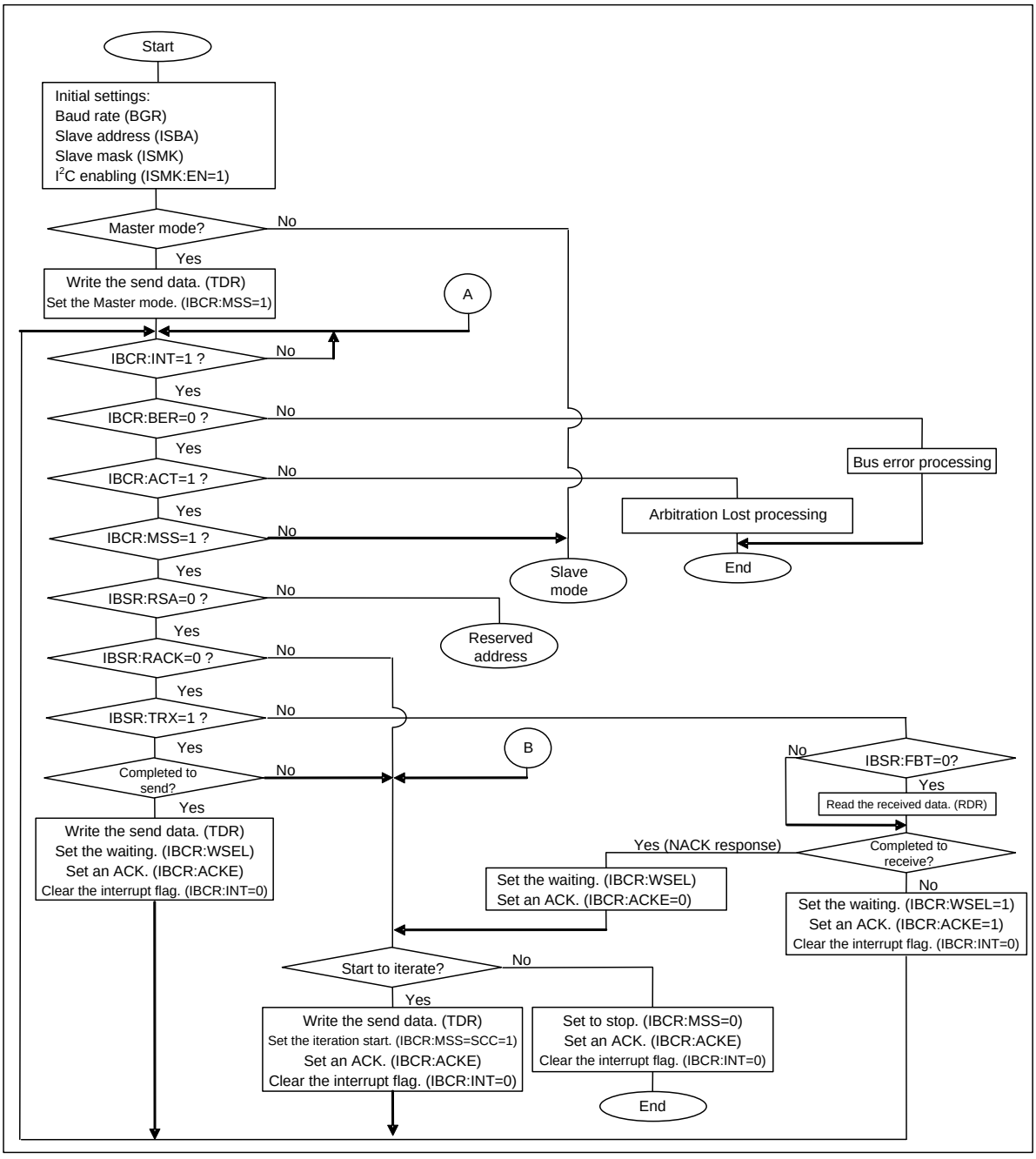


Figure 4-2 I²C Flowchart Example (FIFO Not Used) when DMA Mode is Disabled (SSR:DMA=0) 2/3

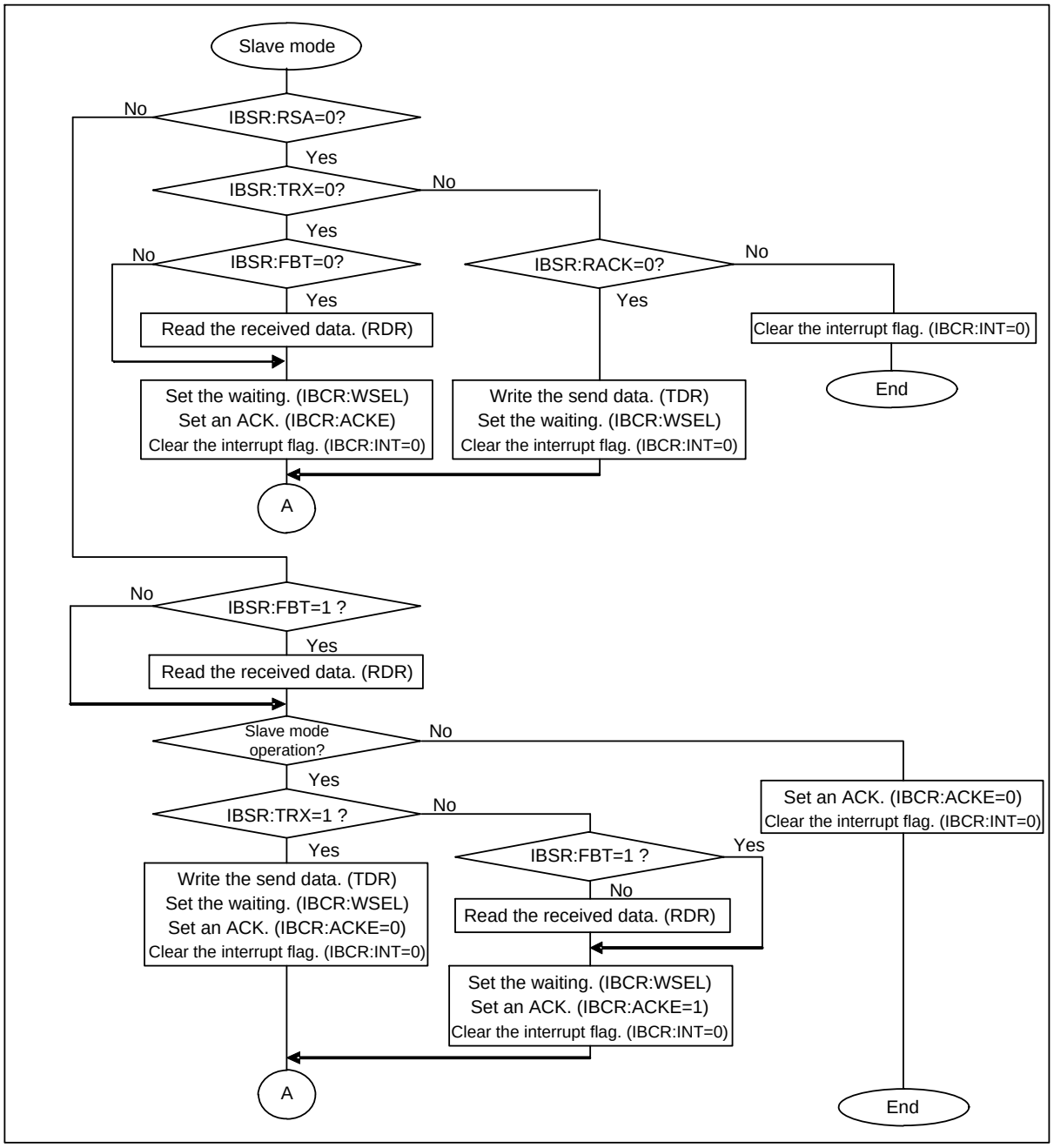
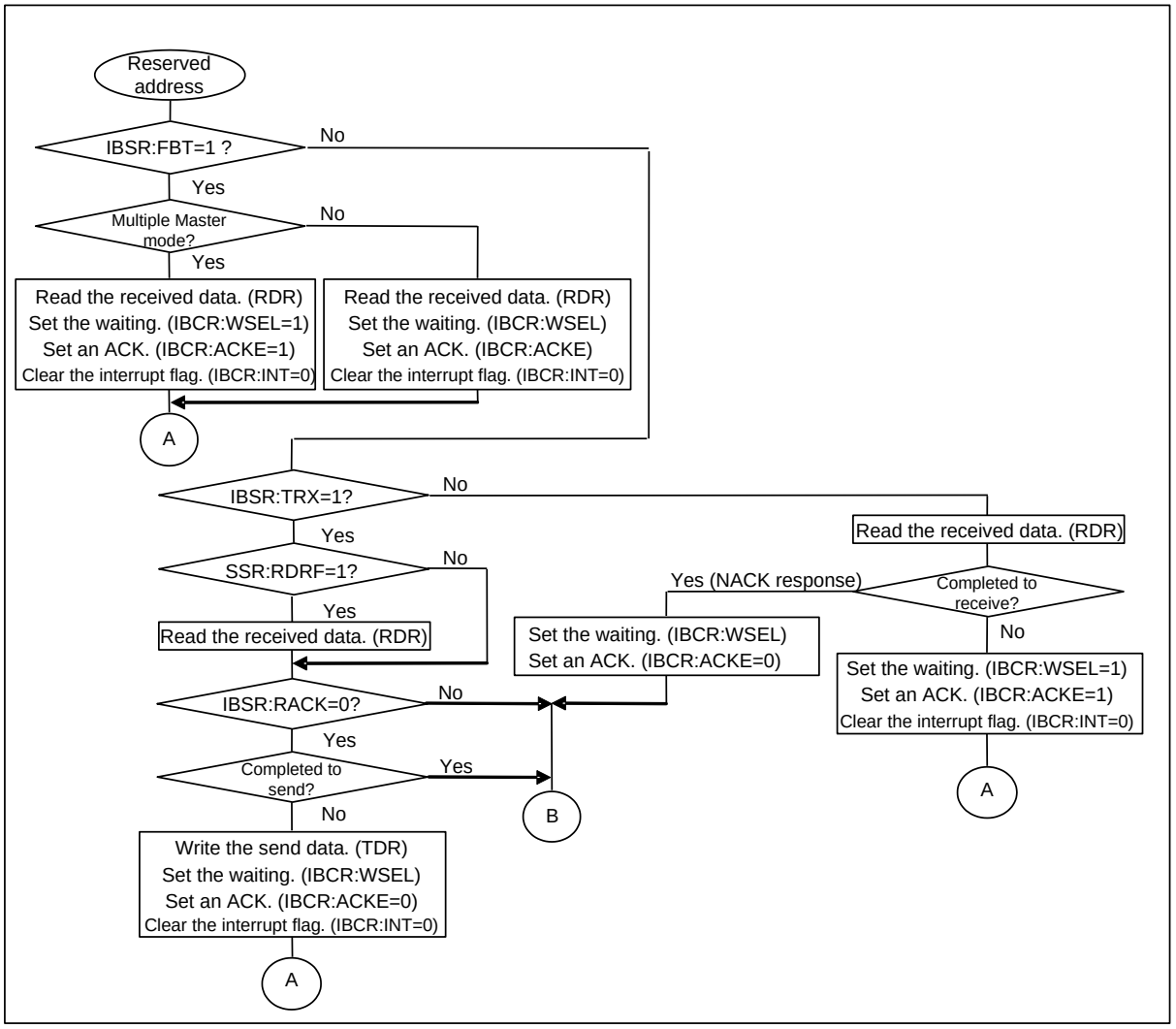
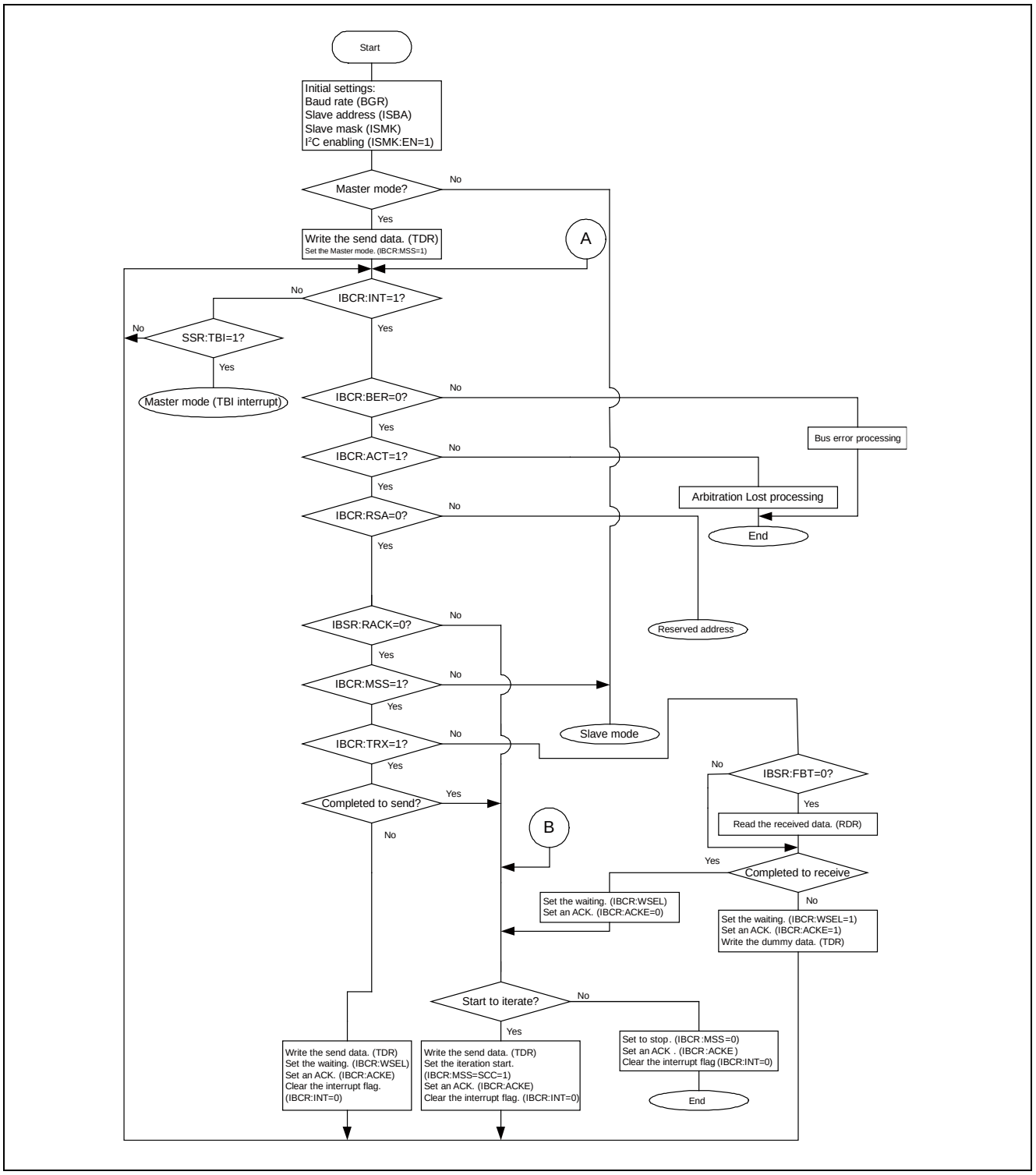


Figure 4-3 I²C Flowchart Example (FIFO Not Used) when DMA Mode is Disabled (SSR:DMA=0) 3/3



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

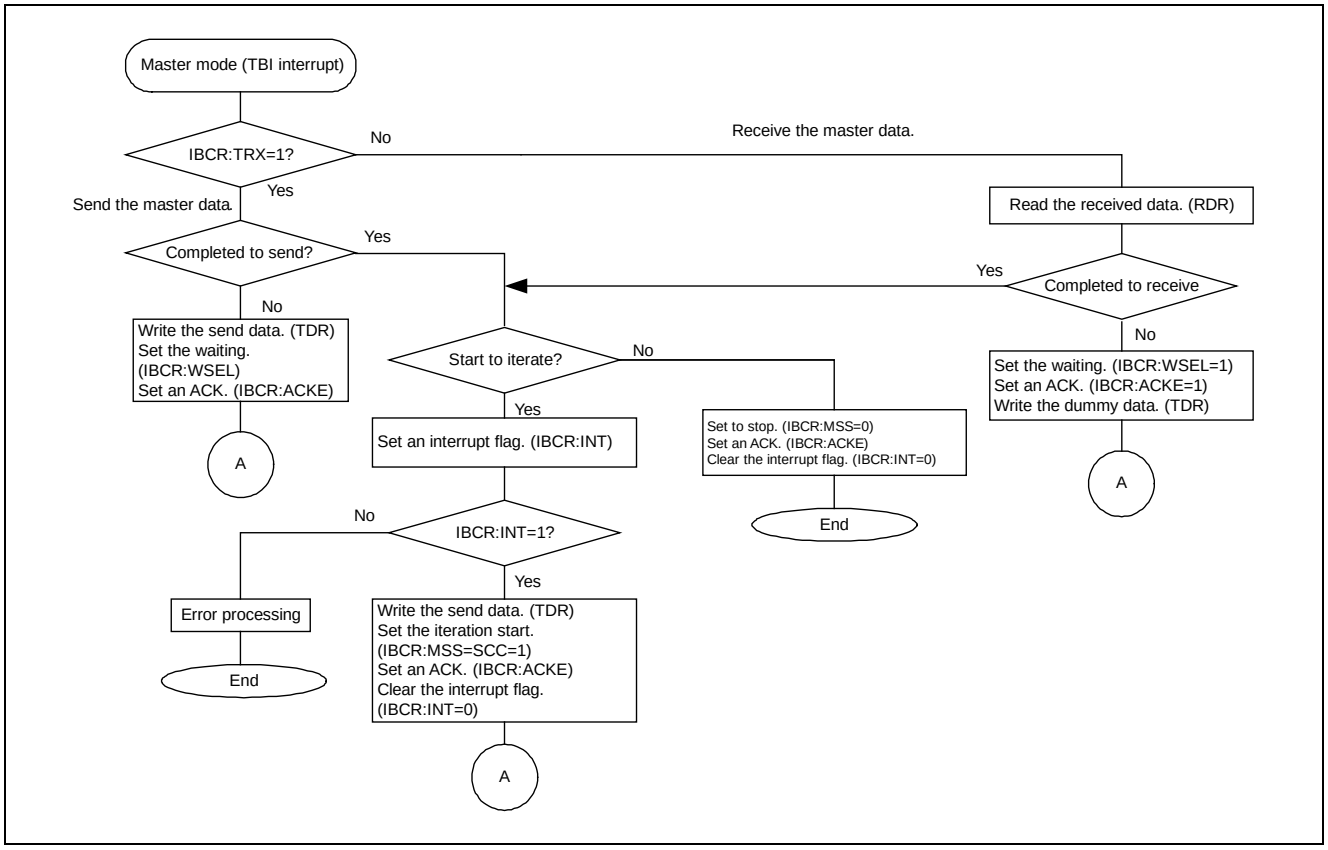


Figure 4-6 I²C Flowchart Example (FIFO Not Used) when DMA Mode is Enabled (SSR:DMA=1) 3/4

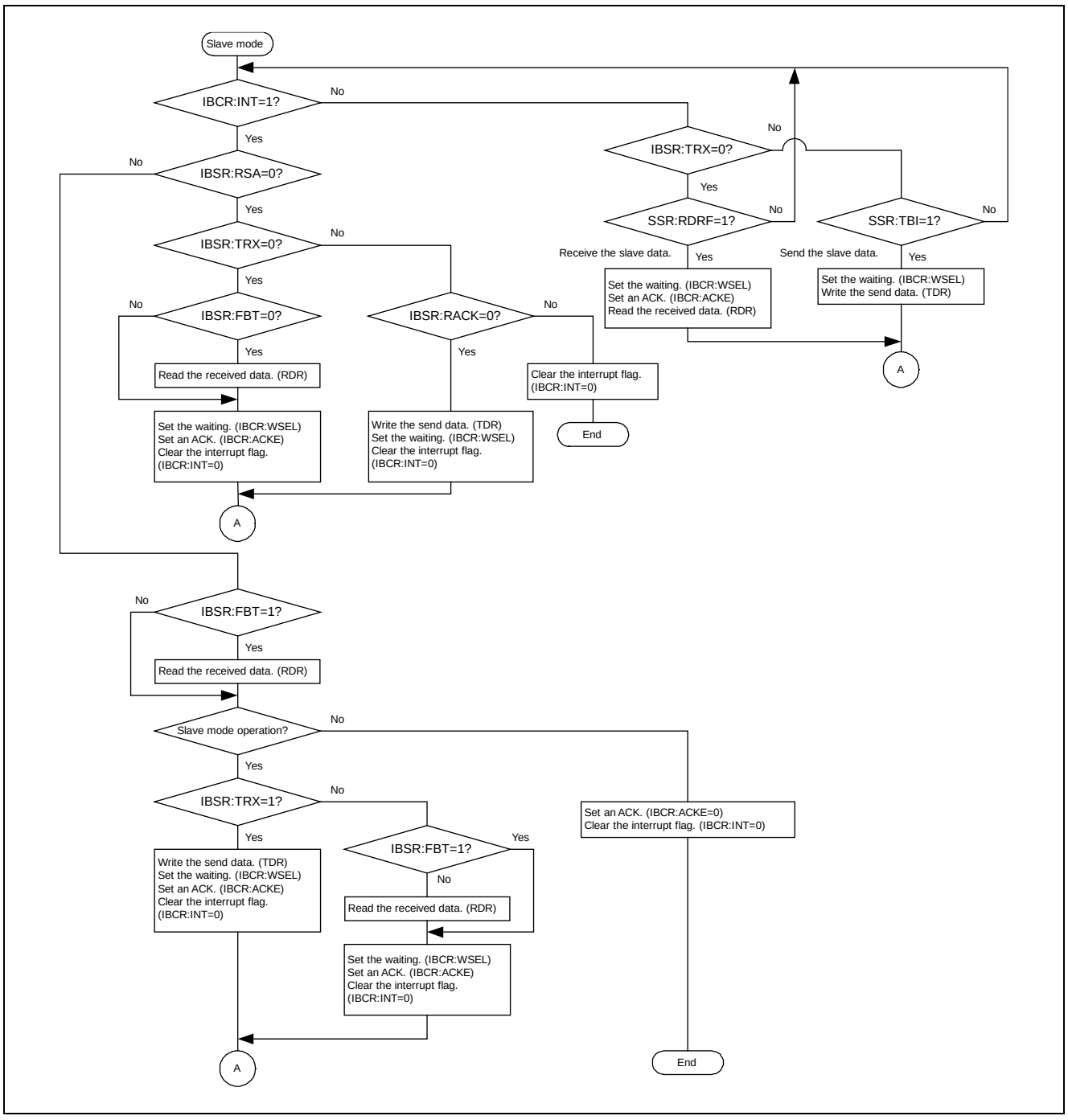
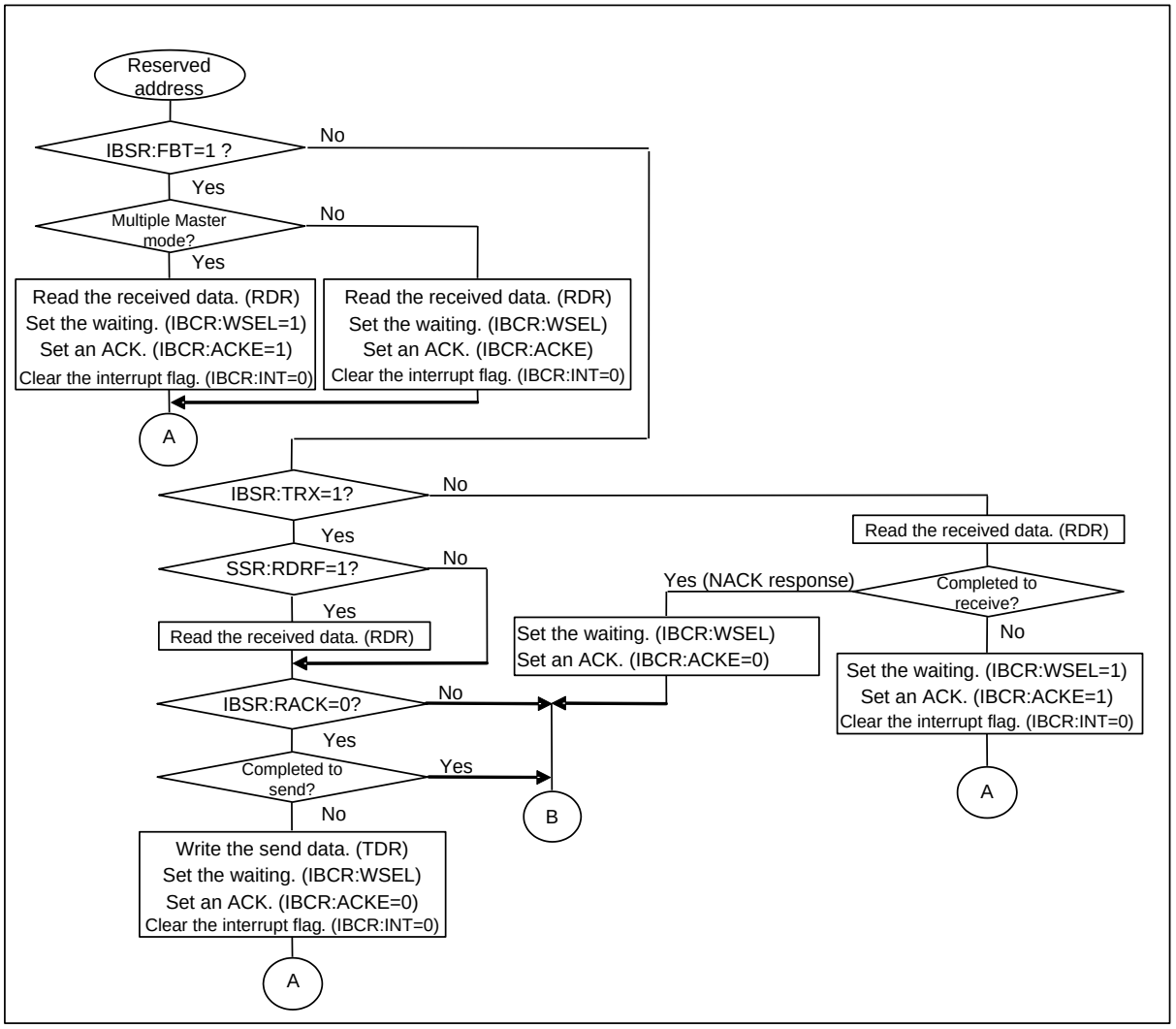


Figure 4-7 I²C Flowchart Example (FIFO Not Used) when DMA Mode is Enabled (SSR:DMA=1) 4/4



<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)	-
	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/ -	-	-	SDAS	SCLS	SDAC	SCLC	SOCE	BEC	-	-	-	-	-	-	-	-
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 relation 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^{2}C 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^{2}C 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 I^{2}C 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 operation
1	Generates an iteration start condition.	During the I^{2}C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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	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 (I^2C 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)
Values other than the above			Setting disabled.

Notes:

- *Any bit setting other than above is inhibited.*
- *To switch the current operation mode, disable the I^2C (ISMK:EN=0) and change 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] 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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 I^2C bus.

The BB bit is reset when:

1. The stop condition is detected.
2. The I^2C 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 I^2C 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 I^2C 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) operation cannot be used.*

5.6 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 I^2C operation.

Bit	Description
0	I^2C operation is aborted.
1	I^2C 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.*

5.7 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".*

5.8 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.

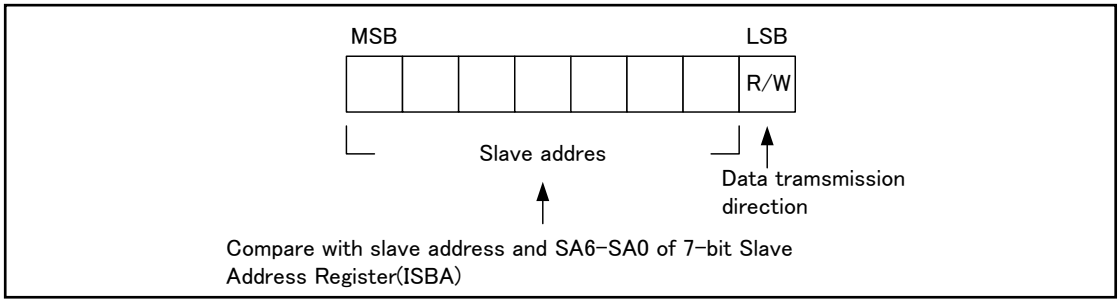
If set to "0": The slave address is not detected.
If set to "1": The ISBA and ISMK settings and the received 1st byte are compared.

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).
- The 7-bit slave address and the direction of a data transfer is contained in the first byte after detection of (iteration) start condition. The slave address which are contained in the received data and these bits are compared.

Figure 5-1 The first byte format after detection of (iteration) start condition



- 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".*

5.9 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 (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.*

`

5.10 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:

- *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 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 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".*

[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).*

`

5.11 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 I^2C 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 I^2C 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 I^2C 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 I^2C 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".*

`

5.12 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 5-3 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 5-4 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.*

CHAPTER1-6: MFS-I^{2}S (Inter-IC Sound Bus)

This chapter explains the functionality of the MFS-I^{2}S interface, which is a serial audio interface.

1. Overview of MFS-I^{2}S
2. Configuration of MFS-I^{2}S Interface
3. Data Structure
4. MFS-I^{2}S interrupts
5. MFS-I^{2}S Registers
6. MFS-I^{2}S Clock Generator Registers
7. MFS-I^{2}S Interface Operation Description
8. User Precaution

1. Overview of MFS-I²S

The MFS-I²S interface can operate as an interface for the transfer of both I²S and MSB-justified by specifying the frame format. It also has transmit/received FIFO (up to 128 bytes each)*1 installed.

MFS-I²S Functions

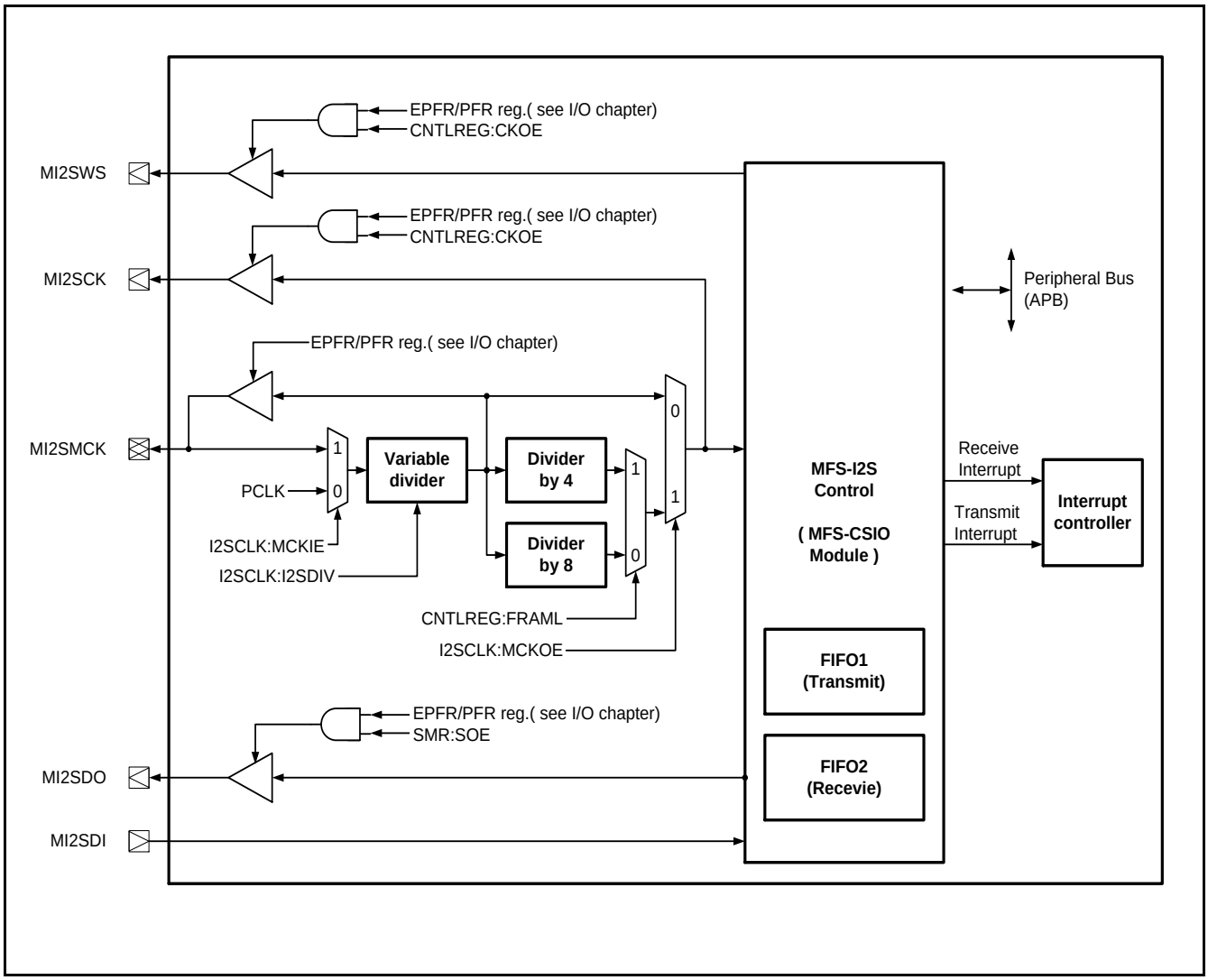
		Function
1	Data buffer	− Transmit/Received FIFO (up to 128 bytes each) *1 − Operation can be done by selecting either transmitting operation or receiving operation (Half-duplex operation)
2	Transfer system	− Clock synchronization − Master operation only
3	Audio sample frequencies	− from 8kHz to 96kHz
4	Data format	− Support 16bit length for the transmit/received data − Support 32xFS and 64xFS for bit clock (MI2SCK) rate
5	Received error detection	− Overrun error
6	Interrupt request	− Received interrupt (a received completion, an overrun error) − Transmit FIFO interrupt (when transmit FIFO is empty) − DSTC (Transmit/Received) transferring support functions are available.
7	Transfer mode	− I²S mode − MSB-justified mode
8	Clock	− The clock source of MI2SCK output can be selected from PCLK(APB Bus Clock) or MI2SMCLK pin input − MI2SMCLK output pin can output the clock of 256 x sampling frequency
9	FIFO options	− FIFO for transmit/received installed (maximum capacity: 128 bytes for transmit FIFO, 128 bytes for received FIFO) * − FIFO resetting is supported independently.

*1: The FIFO capacity size varies depending on the product. For details, see data sheet.

2. Configuration of MFS-I²S Interface

Figure 2-1 shows the configuration of MFS-I²S Interface.

Figure 2-1 MFS-I²S Interface Block Diagram



MFS-I²S Interface use the MFS-CSIO module and FIFO1/FIFO2 are from MFS module. So this MFS-I²S function share the registers and the transmit interrupt (TIRQ) and the received interrupt (RIRQ) with corresponding MFS channel.

3. Data Structure

Figure 3-1 shows the data structure of transmit data.

Figure 3-1 Data Structure of transmit data

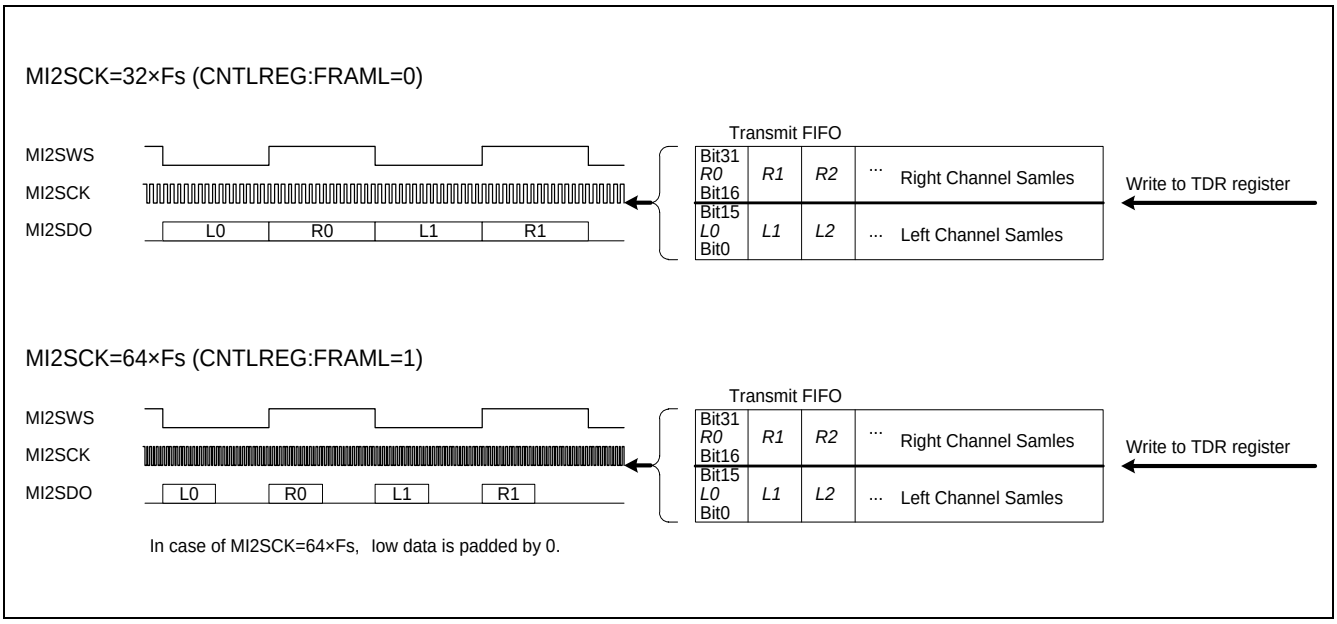
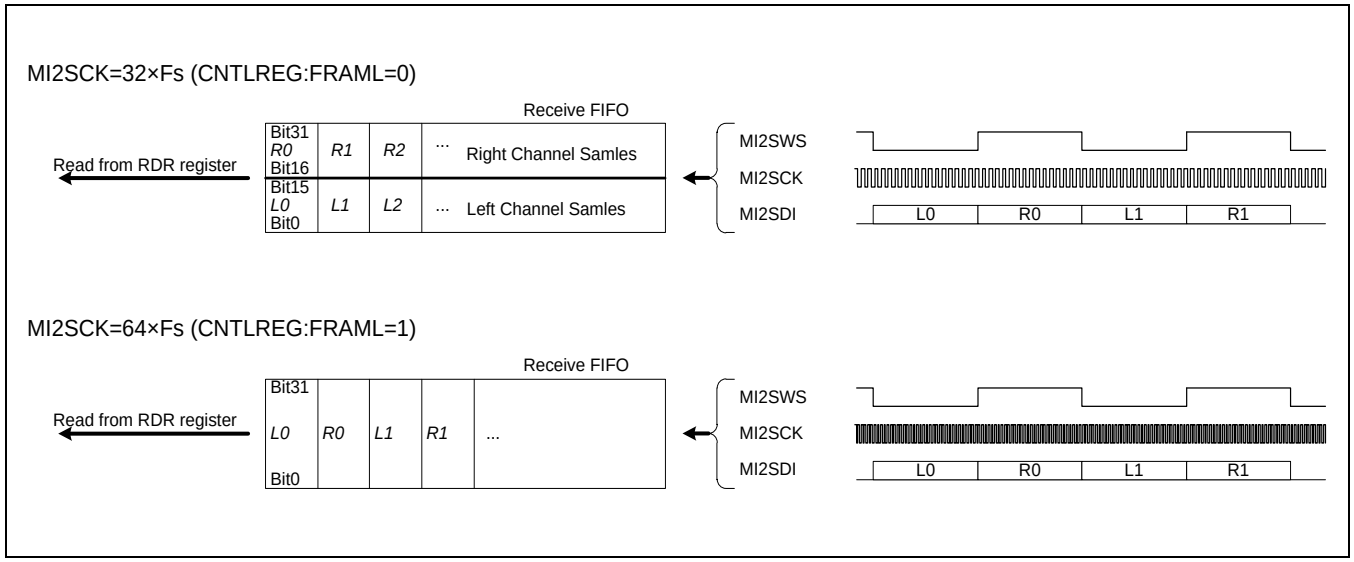


Figure 3-2 shows the data structure of received data.

Figure 3-2 Data Structure of received data



Please refer to “8.2 MFS-I^{2}S and MSB-Justified Protocol” for the details of input and output signal waves.

4. MFS-I²S interrupts

MFS-I²S interrupt has the following interrupt requests.

- Received interrupt requests (RIRQ)
- Transmit interrupt requests (TIRQ)

These interrupt requests are used at the DMA transfer, too.

Table 4-1 shows the relations of MFS-I²S interrupt control bits and the interrupt factors.

Table 4-1 Relation of MFS-I²S 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	Receiving of a data volume matching the value set for FBYTE2.	SCR:RIE	Reading from the Received Data Register (RDR) until data number in received FIFO is less than the value set for FBYTE2.
	ORE	SSR	Overrun error		Writing "1" to the Received Error Flag Clear bit (SSR:REC).
Transmission	FDRQ	FCR1	Transmit FIFO is empty.	FCR1:FTIE	Writing "0" to the FIFO transmit data request bit (FCR1:FDRQ) or transmit FIFO is full.

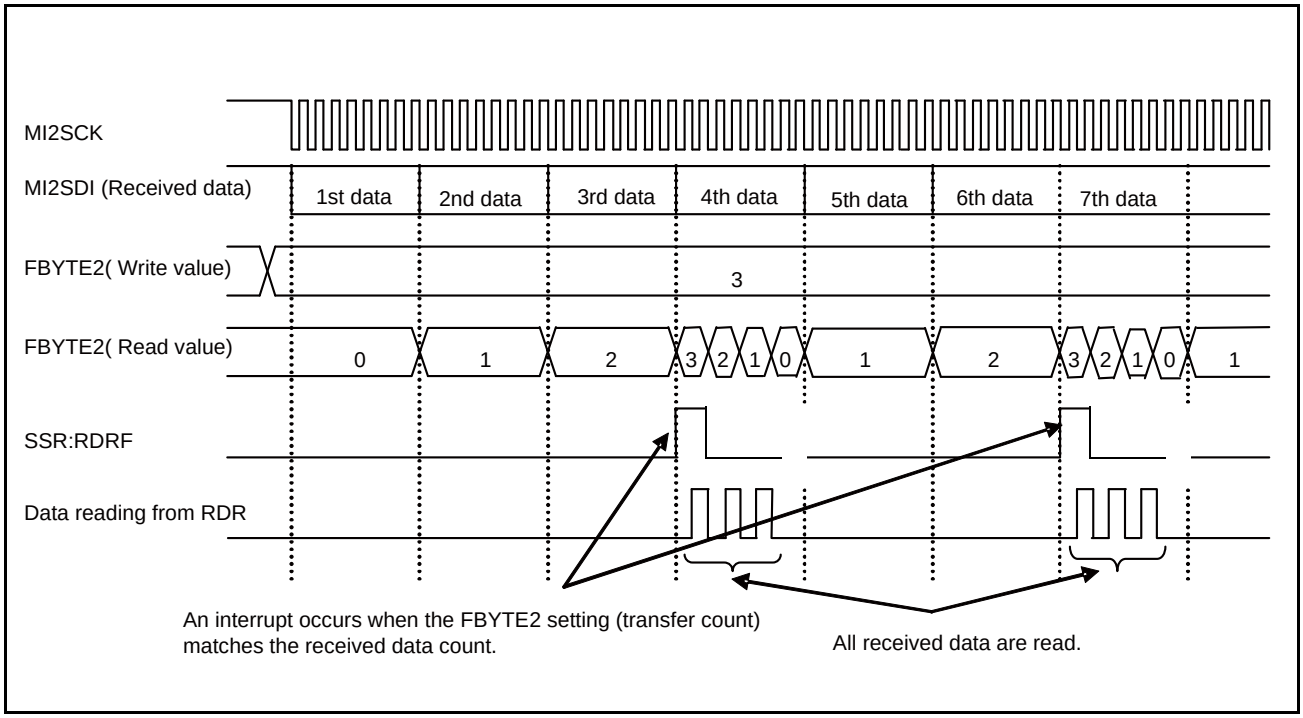
4.1 Interrupt and Flag Set Timing When Received FIFO is Used

If the receive function of MFS-I2S is used, it should be used receive FIFO. 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

- When the amount of data set for transfer count in the FBYTE2 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=1) is enabled during this time, a received interrupt occurs.
- When the valid byte is less than count in the FBYTE2 register, the received data full flag (SSR:RDRF) is cleared to 0.

Figure 4-1 Received Interrupt Occurrence Timing When Received FIFO is Used



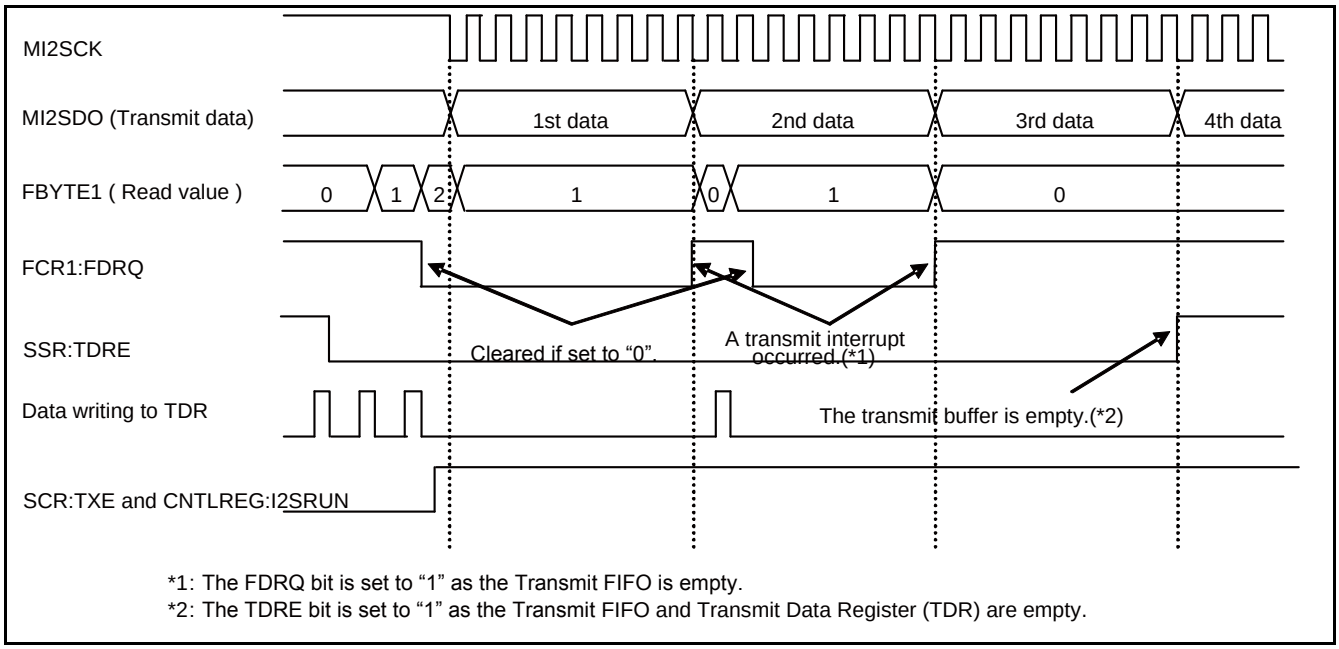
4.2 Interrupt and Flag Set Timing When Transmit FIFO is Used

If the transmit function of MFS-I2S is used, it should be used transmit FIFO. An interrupt occurs if the transmit FIFO 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 writing the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- When transmit FIFO is filled with data and it is in a condition not to be able to write in transmit data, the FIFO transmit data request bit (FCR1:FDRQ) is cleared to "0".
- You can check a presence of data in transmit FIFO by reading the FIFO Byte Register (FBYTE).

Figure 4-2 Transmit Interrupt Occurrence Timing When Transmit FIFO is Used



*1: The FDRQ bit is set to “1” as the Transmit FIFO is empty.
*2: The TDRE bit is set to “1” as the Transmit FIFO and Transmit Data Register (TDR) are empty.

5. MFS-I²S Registers

This section provides a list of MFS-I²S registers.

Table 5-1 MFS-I²S Register List

	bit15 bit8	bit7 bit0
CSIO	SCR (Serial Control Register)	SMR (Serial Mode Register)
	SSR (Serial Status Register)	ESCR (Extended Communication Control Register)
	RDR/TDR (Transmit/Received Data register)	
FIFO	FCR1 (FIFO Control Register 1)	FCR0 (FIFO Control Register 0)
	FBYTE2 (FIFO2 Byte Register)	FBYTE1 (FIFO1 Byte Register)

	bit31 bit24	bit23 bit16
CSIO	RDR/TDR (Transmit/Received Data register)	

Table 5-2 MFS-I²S Bit Assignment

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	-	RIE	-	-	RXE	TXE	MD2	MD1	MD0	-	-	BDS	-	SOE
SSR/ ESCR	REC	-	-	AWC	ORE	RDRF	TDRE	-	-	L3	-	-	-	L2	L1	L0
TDR/ RDR	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
FCR1/ FCR0	-	-	-	-	-	FDRQ	FTIE	FSEL	-	-	-	-	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
TDR/ RDR	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

5.1 Serial Control Register (SCR)

The Serial Control Register (SCR) is used to enable/disable the received interrupts and enable/disable data transmission and received.

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

[bit15] UPCL: Programmable clear bit

Initialize the CSIO internal state.

If set to "1":

- The CSIO is reset directly (software reset). However, the current register settings are kept. The transmission or received state is disconnected immediately.
- All of transmit/received interrupt factors (SSR:TDRE, ORE) are initialized except SSR:RDRF.

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

Select the master or slave mode.

The MS bit should be set to 1 for I^2S mode.

bit	Description
0	Setting is prohibited (Master mode)
1	Slave mode

[bit13] -: Unused bit

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

[bit12] RIE: Received interrupt enable bit

This bit enables or disables an output of received interrupt request to the CPU.

If the RIE bit is “1” as enables the received interrupt, the received data flag bit (SSR:RDRF) are "1" or the error flag bits (ORE) is "1", a received interrupt request is output.

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

[bit11:10] -: Unused bits

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

[bit9] RXE: Data received enable bit

Enables or disables an I^2S data reception.

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

Notes:

- *After you have set the MS bit, enable the data reception (RXE=1).*

[bit8] TXE: Data transmission enable bit

Enables or disables an I^2S data transmission.

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

Notes:

- *Either transmitting operation or receiving operation can be selected. Setting both “RXE=1” and “TXE=1” is prohibited.*

5.2 Serial Mode Register (SMR)

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

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(SCR)	MD2	MD1	MD0	-	-	BDS	-	SOE
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.

In case of I^2S mode, these bits should be written “010” (clock synchronous mode).

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

Notes:

- *After the operation mode has been set, set each register correctly.*

[bit4:3] -: Unused bits

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

[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).

In case of I^2S mode, should be set “1” (MSB first).

bit	Description
0	Setting is prohibited. (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).*

[bit1] -: Unused 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.

In case of I^2S transmission mode, should be set “1”.

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

Note:

- *When performing the data transmission, the GPIO must also be set.*

5.3 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	-	-	AWC	ORE	RDRF	TDRE	-	(ESCR)
Attribute	R/W	-	-	R/W	R	R	R	-	
Initial value	0	-	-	0	0	0	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 (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] AWC: FIFO access width set

This bit determines the access width of FIFO.

■ In case of I^2S mode, should be written “1” (32 bit access).

bit	Description
0	Setting is prohibited (16bit access.)
1	32bit access.

[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 (RIRQ) 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).

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 the received FIFO is used and if the data contained in FIFO is less than the preset amount (FBYTE), this bit is cleared to "0".

The RDRF flag bit will be cleared to “0” after receive FIFO reset.

In case that the value of SCR:RIE bit is 1, a receive interrupt request occurs if RDRF is set to 1.

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 FIFO is used and if transmit data is written in the TDR, this bit is cleared to "0" to indicate that the TDR contains valid data. When the valid data in the transmit FIFO and the transmit data register are empty, this bit is set to "1" to indicate that the TDR does not have the valid data.

When the UPCL bit of the Serial Control Register (SCR) is written to "1", the TDRE bit is set to "1".

For the TDRE bit set/reset timing when transmit FIFO is used, see 4.2" 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] -: Unused bit

The bit value is undefined when read.
This bit has no effect on the operation when written.

5.4 Extended Communication Control Register (ESCR)

The Extended Communication Control Register (ESCR) is used to set a transmit/received data length.

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

[bit7] -: Unused bits

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

[bit5:3] -: Unused bits

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

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

These bits set a length of transmit/received data.

In case of I^2S mode, should be written these bits to “1111” (32-bit length).

L3	L2	L1	L0	Description
1	1	1	1	32-bit length
Other				Setting is prohibit

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.

When Received Data Register (RDR) is read at the Received FIFO is enabled, the Received Data are read out from the Received FIFO.

When Transmit Data Register (TDR) is written at the Transmit FIFO is enabled, the Transmit Data are written it to the Transmit FIFO.

Received Data Register (RDR)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	D 31	D 30	D 29	D 28	D 27	D 26	D 25	D 24	D 23	D 22	D 21	D 20	D 19	D 18	D 17	D 16
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

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	D 15	D 14	D 13	D 12	D 11	D 10	D 9	D 8	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
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 32-bit data buffer register for serial data reception.

- In case of I^2S mode, the Received Data Register (RDR) should be accessed by 32-bit bus width.
- 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).
- In the case of CNTLREG:FRAML= 0, the 16 bits data of the right channel are stored in D31 - D16 and 16bits data of the left channel are stored in D15 - D0. (Please refer to Figure 3-2.)
- In the case of CNTLREG: FRAML= 1, the 32 bits data of the right channel and the left channel are stored in D31-0 by turns. (Please refer to Figure 3-2.)
- When valid data exist in the Received FIFO, please read out the data of the Received Data register (RDR). When valid data does not exist in the Received FIFO, reading the Received Data register (RDR) is prohibited.

Notes:

- *If received FIFO is enabled and if a received error occurs (SSR:ORE=1), the received FIFO's enable bit is cleared to "0" and the received data is not stored in received FIFO.*

Transmit Data Register (TDR)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	D 31	D 30	D 29	D 28	D 27	D 26	D 25	D 24	D 23	D 22	D 21	D 20	D 19	D 18	D 17	D 16
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

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	D 15	D 14	D 13	D 12	D 11	D 10	D 9	D 8	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
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 32-bit data buffer register for serial data transmission.

- In case of I^2S mode, the Transmit Data Register (TDR) should be accessed by 32-bit bus width.
- If data transmission is enabled (SCR:TXE=1, CNTLREG: I2SRUN=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 (MI2SDO).
- The 16 bits data of the right channel are written in D31 - D16 and the 16bits data of the left channel are written in D15 – D0 without depending on a value of CNTLREG: FRAML. (Please refer to Figure 3-1.)
- If the Transmit FIFO and the Transmit Data Register (TDR) are not full, the next transmission data can be written in the Transmit Data Register (TDR). If not, writing data in the Transmit Data Register is prohibited.

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, writing to TDR cannot be supported for read-modify-write (RMW) instructions via the area of the bit-band alias.*

5.6 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			-	-	FDRQ	FTIE	FSEL	(FCR0)		
Attribute	-	-	-	-	-	R/W	R/W	R/W			
Initial value	-	-	-	-	-	1	0	0			

[bit15:13] Reserved: Reserved bits

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

[bit12:11] -: Unused bits

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

[bit10] FDRQ: Transmit FIFO data request bit

This bit shows the requests for the transmit FIFO data.

If this bit is "1", the transmit data is being requested. The FDRQ bit is set when:

- The FBYTE 1 is "0" (Size of valid data of Transmit FIFO is 0).
- Transmit FIFO is reset by CPU

The FDRQ bit is cleared to “0” when:

- This bit is written to "0" by CPU
- The Transmit FIFO and the Transmit Data Register (TDR) are filled with data.

In case that the transmit FIFO interrupt is permitted (FCR1:FTIE=1), if this bit is set to “1” the transmit FIFO interrupt request (TIRQ) occurs.

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

Notes:

- *When this bit is "0", to change the bit of FSEL is prohibition.*
- *If this bit is set to "1", it has no effect on the operation.*
- *If a read-modify-write instruction via area of the bit-band alias 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 or the Received FIFO.

In case of I^2S mode, this bit should be written “0”.

bit	Description
0	Assigned for FIFO1 as Transmit FIFO and FIFO2 as Received FIFO
1	Setting is prohibit (Assigned for FIFO2 as Transmit FIFO and FIFO1 as Received FIFO)

Notes:

- *This bit is not cleared by FIFO reset (FCR0:FCL2=1, FCR0:FCL1=1).*

5.7 FIFO Control Register 0 (FCR0)

The FIFO Control Register 0 (FCR0) is used to enable/disable the FIFO operation, reset FIFO.

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

[bit7] - : Unused bit

"0" is always read.

"0" must always be written.

[bit6] - : Unused bit

The values of these bits are undefined when read.

These bits have no effect on the operation when written.

[bit5:4] - : Unused bit

"0" is always read.

"0" must always be written.

[bit3] FCL2: FIFO2 reset bit

This bit resets the FIFO2 value.

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

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.

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".
- In case of I^2S mode, this bits should be written to “1” (Enable the FIFO2 operation).
  - If the FIFO2 as Received FIFO is selected and if 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 received FIFO, this bits can change when satisfy all following conditions.
    - The Data received enable bit of Serial Control Register (SCR: RXE) is 0.
    - The Received data full flag bit of Serial Status Register (SSR: RDRF) is 0
  - The FIFO2 state is holding even if disables the FIFO2 operation.

bit	Description
0	Setting is prohibit (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".
- In case of I^2S mode, this bits should be set to “1” (Enable the FIFO1 operation).
  - If FIFO1 is used as Transmit FIFO, this bits can change when satisfy all following conditions.
    - The Data transmission enable bit of Serial Control Register (SCR: TXE) is 0.
    - The Transmit data empty flag bit of Serial Status Register (SSR: TDRE) is 1
  - The FIFO1 state is holding even if disables the FIFO1 operation.

bit	Description
0	Setting is prohibit (Disables the FIFO1 operation).
1	Enables the FIFO1 operation.

5.8 FIFO Byte Register (FBYTE)

The FIFO Byte Register (FBYTE) is used to set about FIFO data count.

The definition of the data read out from this register is different to that of the data written to the register.

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

[bit15:8] FBYTE2: FIFO2 valid data count

When read out these bit; the valid data counts of FIFO2 (Received FIFO) are read out.

The value to write in sets counts of received data setting the received data full flag of the Serial Status Register (SSR: RDRF) to 1 to these bit. When counts of valid data of Received FIFO accorded with counts of set data; received data full flag of the Serial Status Register (SSR : RDRF) is set to 1.

Writing	Sets counts of received data setting the received data full flag of the Serial Status Register (SSR: RDRF) to 1 to these bit.
Reading	Valid received data counts of FIFO2 are read out.

Table 5-3 FIFO2 data count and the register value of FBYTE2

FIFO Capacity	Max. set count of FBYTE2 (decimal number)	Max, storage data count (decimal number)
64 byte	14	15 （In case of CNTLREG:FRAML=0）
		16 （In case of CNTLREG:FRAML=1）
128 byte	30	31 （In case of CNTLREG:FRAML=0）
		32 （In case of CNTLREG:FRAML=1）

Notes:

- *The FIFO capacity is different depending on the product to use. Please confirm the capacity from data sheet.*
- *The initial value of the set value of the received data count of these bits is 0x08.*
- *Should be changed these bits at the Data received enable bit of the Serial Control Register (SCR: RXE) is 0.*
- *The value of writing of FBYTE2 must be set to 0x01 or more.*
- *The value of writing of FBYTE2 cannot be set the value bigger than maximum set count of FBYTE2 in Table 5-3. This value shows the upper limit of the value of writing.*
- *When the values of reading of FBYTE2 are accorded to the maximum storage data count in Table 5-3, it shows Received FIFO full state. The output operation of the frame signal (MI2SWS) stops. This value shows the upper limit of the value of reading.*
- *The value of reading of FBYTE2 show the data count which are not begun to read from the Received FIFO. If this value is 0x00, reading the Received Data Register (RDR) is prohibited.*
- *This register cannot support the read-modify-write (RMW) instruction via the area of bit-band alias.*

[bit7:0] FBYTE1: FIFO1 valid data count

When read out these bit; the valid data counts of FIFO1 (Transmit FIFO) are read out.

These bits should be written in values of 0x00.

Writing	Should be written in values of 0x00.
Reading	Valid transmit data counts of FIFO1 are read out. (The number of the data which it has been already written in FIFO, but have not been yet transmitted).

Table 5-4 FIFO1 data count and the register value of FBYTE1

FIFO Capacity	Max, storage data count (decimal number)
64 byte	16
128 byte	32

Notes:

- *The FIFO capacity is different depending on the product to use. Please confirm the capacity from data sheet.*
- *The value of reading of FBYTE1 can read out the value that subtracted 1 from the number of the transmit data to be written. Because there are the valid data in the Transmit Data Register (TDR) other than the Transmit FIFO.*
- *When the values of reading of FBYTE1 are accorded to the maximum storage data count in Table 5-4, writing to the Transmit Data Register (TDR) is prohibited.*
- *When the values of reading of FBYTE1 are 0x00 and there are not the valid data in the Transmit Data Register (TDR), the output operation of the frame signal (MI2SWS) stops.*
- *This register cannot support the read-modify-write (RMW) instruction via the area of bit-band alias.*

6. MFS-I²S Clock Generator Registers

This section provides a list of MFS-I²S clock generator registers.

Table 6-1 MFS-I²S Clock Generator Register List

	bit15	bit8	bit7	bit0
I²S clock generator	CNTL (I²S Control Register)			
	I2SCLK (I²S Clock Setting Register)			
	I2SST (I²S State Register)		I2SRST (I²S Reset Register)	

6.1 Control Register (CNTLREG)

This is the control register for I²S.

bit	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	I2SRUN	-	-
Attribute	-	-	-	-	-	R/W	-	-
Initial Value	-	-	-	-	-	0	-	-

bit	7	6	5	4	3	2	1	0
Field	-	CKOE	I2SEN	FSPL	I2SMOD	-	-	FRAML
Attribute	-	R/W	R/W	R/W	R/W	-	-	R/W
Initial Value	-	0	0	0	0	-	-	1

[bit15:11] - : Unused bit

"0" is always read.

"0" must always be written.

[bit10] I2SRUN: I²S clock generate enable

This bit enables or disables the internal clock generation of I²S.

When I2SRUN is 0, MI2SMCK and MI2SCK and MI2SWS are stopped output.

bit	Description
0	I²S clock generate is disabled.
1	I²S clock generate is enabled.

[bit9:7] - : Unused bit

"0" is always read.

"0" must always be written.

[bit6] CKOE: MI2SCK and MI2SWS(frame sync signal) output enable signal

Clock output enable bit.

In case of I²S mode, this bit should be written “1” (MI2SCK and MI2SWS output enable).

bit	Description
0	Setting is prohibited (MI2SCK and MI2SWS output disable.)
1	MI2SCK and MI2SWS output enable.

[bit5] I2SEN: I²S mode enable

This bit forces the MFS-CSIO module work in MFS-I²S mode.

In case of I²S mode, this bit should be written “1”

bit	Description
0	Setting is prohibited (MFS-CSIO module is used other than MFS-I²S mode)
1	MFS-CSIO module is used as MFS-I²S mode.

[bit4] FSPL: I2SWS polarity set

This bit sets the polarity of the MI2SWS output.

bit	Description
0	When left channel is “Low”, when right channel and idle is “High”.
1	When left channel is “High”, when right channel and idle is “Low”.

[bit3] I2SMOD: I²S mode select

This bit sets a change timing of the MI2SWS output.

bit	Description
0	Output MSB data after 1SCLK of the MI2SWS change.
1	Output MSB data at the time of MI2SWS change.

In the case of I²S Philips standard mode, should be CNTLREG:FSPL=0, CNTLREG:I2SMOD=0.

In the case of MSB-Justified standard mode, should be CNTLREG:FSPL=1, CNTLREG:I2SMOD=1.

For more information about these modes, please refer to “8.2 MFS-I2S and MSB-Justified Protocol”.

[bit2:1] - : Unused bit

"0" is always read.

"0" must always be written.

[bit0] FRAML: Selection of MI2SCK bit rate (Frame length select)

This bit set the bit rate of MI2SCK.

bit	Description
0	MI2SCK bit rate is 32 x Fs (Sampling frequency).
1	MI2SCK bit rate is 64 x Fs (Sampling frequency).

6.2 MFS-I²S Clock Registers (I2SCLK)

This is the clock register for I²S.

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

bit	7	6	5	4	3	2	1	0
Field	I2SDIV[7:0]							
Attribute	R/W							
Initial Value	0x00							

[bit15] MCKIE: Main clock input enable

This bit selects the clock source of MI2SCK.

Refer to the Figure 2-1.

bit	Description
0	Use APB clock (PCLK) as input of the Variable divider.
1	Use input clock from MI2SMCK pin as input of the Variable divider.

[bit14] MCKOE: Main clock output selection

This bit selects the clock source of MI2SCK.

Refer to the Figure 2-1.

bit	Description
0	Use output of the Variable divider as MI2SCK output.
1	Use output after divided for output of the Variable divider as MI2SCK output.

Notes:

- *When use MI2SMCK as the output pin, should be set it the output in the EPFR register of GPIO.*
- *When use MI2SMCK as the output pin, setting of I2SCLK: MCKIE=1 is prohibition.*
- *When CNTLREG: I2SRUN=0, the MI2SMCK output, the MI2SCK output, the MIS2WS output are stopped.*

[bit13:8] - : Unused bits

The values of these bits are undefined when read.

This bit has no effect on the operation when written.

[bit7:0] I2SDIV: I²S clock division set

These bits set the divided value of Variable divider.

Refer to the Figure 2-1.

bit7:0	Description
0x00	Divided by 1 (Bypass clock).
0x01	Divided by 2.
0x02	Divided by 4.
0x03	Divided by 6.
0x04	Divided by 8.
…	…
0xFE	Divided by 508.
0xFF	Divided by 510.

Note:

- *The setting of Variable divider must meet following condition.*
  MI2SCK clock frequency ≦ PCLK(APB bus clock) frequency / 4

Each output clock frequency by setting of MCKIE and MCKOE and I2SDIV and CNTLREG:FRAML are showed in following table.

Table 6-2 Frequency of MI2SWS output

MCKIE	MCKOE	FRAML	Frequency of MI2SWSoutput（= Fs :sampling frequency）
0	0	0	PCLK frequency / (I2SDIV x 32)
0	0	1	PCLK frequency / (I2SDIV x 64)
0	1	X	PCLK frequency / (I2SDIV x 256)
1	0	0	MI2SMCK input frequency / (I2SDIV x 32)
1	0	1	MI2SMCK input frequency / (I2SDIV x 64)
1	1	X	MI2SMCK input frequency / (I2SDIV x 256)

Table 6-3 Frequency of MI2SCKoutput

MCKIE	MCKOE	FRAML	Frequency of MI2SCK output
0	0	X	PCLK frequency / (I2SDIV x 1)
0	1	0	PCLK frequency / (I2SDIV x 8)
0	1	1	PCLK frequency / (I2SDIV x 4)
1	0	X	MI2SMCK input frequency / (I2SDIV x 1)
1	1	0	MI2SMCK input frequency / (I2SDIV x 8)
1	1	1	MI2SMCK input frequency / (I2SDIV x 4)

Table 6-4 Frequency of MI2SMCKoutput

MCKIE	MCKOE	FRAML	Frequency of MI2SMCK output
0	X	X	PCLK frequency / (I2SDIV x 1)
1	X	X	In this setting, MI2SMCK clock cannot be output

Clock setting example 1:

- Clock frequency :
  MI2SWS output = 48 kHz (Fs)
  MI2SCK output = 1536 kHz (32 x Fs)
  MI2SMCK input = 12288 kHz (256 x Fs)
  PCLK input >= 6144 kHz.
- Register setting value
  MCKIE=1, MCKOE=0, I2SDIV=0x04, CNTLREG: FRAML = 0

Clock setting example 2:

- Clock frequency :
  MI2SWS output = 48 kHz (Fs)
  MI2SCK output = 3072 kHz (64 x Fs)
  MI2SMCK input = 12288 kHz (256 x Fs)
  PCLK input >= 12288 kHz.
- Register setting value
  MCKIE=1, MCKOE=0, I2SDIV=0x02, CNTLREG: FRAML = 1

Clock setting example 3:

- Clock frequency :
  MI2SWS output = 48 kHz (Fs)
  MI2SCK output = 1536 kHz (32 x Fs)
  MI2SMCK input = 12288 kHz (256 x Fs)
  PCLK input = 24576 kHz (512 x Fs)
- Register setting value
  MCKIE=0, MCKOE=1, I2SDIV=0x01, CNTLREG: FRAML = 0

Clock setting example 4:

- Clock frequency :
  MI2SWS output = 48 kHz (Fs)
  MI2SCK output = 1536 kHz (32 x Fs)
  MI2SMCK input = 1536 kHz (32 x Fs)
  PCLK input = 36864 kHz (768 x Fs)
- Register setting value
  MCKIE=0, MCKOE=0, I2SDIV=0x0c, CNTLREG: FRAML = 0

6.3 MFS-I2S Status Registers (I2SST)

This is the status register for I^2S.

bit	15	14	13	12	11	10	9	8	7 ... 0
Field	-	-	-	-	-	-	BUSY	CKSTP	(I2SRST)
Attribute	-	-	-	-	-	-	R	R	
Initial value	-	-	-	-	-	-	0	0	

[bit15:10] - : Unused bits

The value of this bit is undefined when read.

This bit has no effect on the operation when written.

[bit9] BUSY: Bus busy indication for transmit

This bit indicates that the I^2S bus is transmitting data.

bit	Description
0	No data transmit
1	Data is being transmitting

[bit8] CKSTP: Clock stop indication

This bit indicates that the MI2SCK output is stopped after CNTLREG:I2SRUN bit is set to 0.

bit	Description
0	MI2SCK output is stopped.
1	MI2SCK output is running.

6.4 MFS-I2S Reset Registers (I2SRST)

This is the software reset register for I^2S.

bit	15 ... 8	7	6	5	4	3	2	1	0
Field	(I2SST)	I2SRST							
Attribute		W	W	W	W	W	W	W	W
Initial value		0	0	0	0	0	0	0	0

[bit7:0] I2SRST: I^2S software reset

- Write 0xA5, generate software reset to reset the internal state and flag signal.
- Write value other than 0xA5, no operation.
- Read data will always be 0x00
- This register should be written by byte access.

7. MFS-I²S Interface Operation Description

7.1 Data transmit operation

Setting procedure examples of the data transmit operation show below.

1. Set CNTLREG register

   Should be set I2SEN=1, CKOE=1, I2SRUN=0. Other register bits setting value is arbitrary.

2. Set I2SCLK register

   Register setting value is arbitrary.

3. Set SMR register

   Should be set MD [2:0] =010, BDS=1, SOE=1.

4. Set SSR register

   Should be set AWC=1. Other register bits setting value is arbitrary.

5. Set ESCR register

   Should be set L3, L2, L1, L0 = 1111.

6. Set FCR0 register

   Should be set FE1=1, FE2=0. Transmit FIFO should be cleared by writing of FCL1=1.

7. Set FCR1 register

   Should be set FSEL=0. Set of FTIE is arbitrary. FDRQ cannot be cleared at the initial state.

8. Set FBYTE1 register

   Should be set FBYTE1=0x0.

9. Set SCR register

   Should be set TXE=1, RXE=0, MS=1. Other register bits setting value is arbitrary.

10. Set CNTLREG register

    Should be set I2SRUN=1. Other register bits are same as #1.

11. Write transmit data to TDR register

Notes:

- *In case of CNTLREG: I2SRUN=1, SCR: TXE=1; When transmit FIFO and TDR are not empty, frame synchronization signal (MI2SWS), the bit clock (MI2SCK), transmit data are output to MI2SDO.*
- *When transmit data disappear during transmitting, frame synchronization signal (MI2SWS) will be stopped the output. The bit clock (MI2SCK), the master clock output (MI2SMCK) are being continue the output.*
- *During transmitting, if write to SCR: TXE = 0, Frame synchronization signal (MI2SWS) will be stopped the output after the output the data at the point in time.*
- *During transmitting, if write to CNTLREG: I2SRUN= 0, Frame synchronization signal (MI2SWS) and the bit clock (MI2SCK) and the master clock output (MI2SMCK) will be stopped the output.*
- *After having written in the data of the number necessary for the Transmit Data Register (TDR), should be cleared the flag by writing 0 to FCR1: FDRQ.*
- *At the start of the data transmit, should be written to CNTLREG: I2SRUN is 1 last.*

7.2 Data received operation

Setting procedure examples of the data received operation show below.

1. Set CNTLREG register

   Should be set I2SEN=1, CKOE=1, I2SRUN=0. Other register bits setting value is arbitrary.

2. Set I2SCLK register

   Register setting value is arbitrary.

3. Set SMR register

   Should be set MD [2:0] =010, BDS=1, SOE=0.

4. Set SSR register

   Should be set AWC=1. Other register bits setting value is arbitrary.

5. Set ESCR register

   Should be set L3, L2, L1, L0 = 1111.

6. Set FCR0 register

   Should be set FE1=0, FE2=1. Transmit FIFO should be cleared by writing of FCL2=1.

7. Set FCR1 register

   Should be set FSEL=0 and FTIE=0.

8. Set FBYTE2 register

   Should be set an appropriate value to FBYTE2.

9. Set SCR register

   Should be set TXE=0, RXE=1, MS=1. Other register bits setting value is arbitrary.

10. Set CNTLREG register

    Should be set I2SRUN=1. Other register bits are same as #1.

11. Read out the received data from RDR register

Notes:

- *In case of CNTLREG: I2SRUN=1, SCR: TXE=1; When received FIFO are not full, frame synchronization signal (MI2SWS) will be output and received data will be received from MI2SDI.*
- *During receiving, if received FIFO are full, Frame synchronization signal (MI2SWS) will be stopped the output. The bit clock (MI2SCK), the master clock output (MI2SMCK) will be being continue the output.*
- *During receiving, if write to SCR: RXE= 0, Frame synchronization signal (MI2SWS) will be stopped the output after receiving the data at the point in time.*
- *During receiving, if write to CNTLREG: I2SRUN= 0, Frame synchronization signal (MI2SWS) and the bit clock (MI2SCK) and the master clock output (MI2SMCK) will be stopped the output.*
- *At the start of the data received, should be written to CNTLREG:I2SRUN is 1 last.*

8. User Precaution

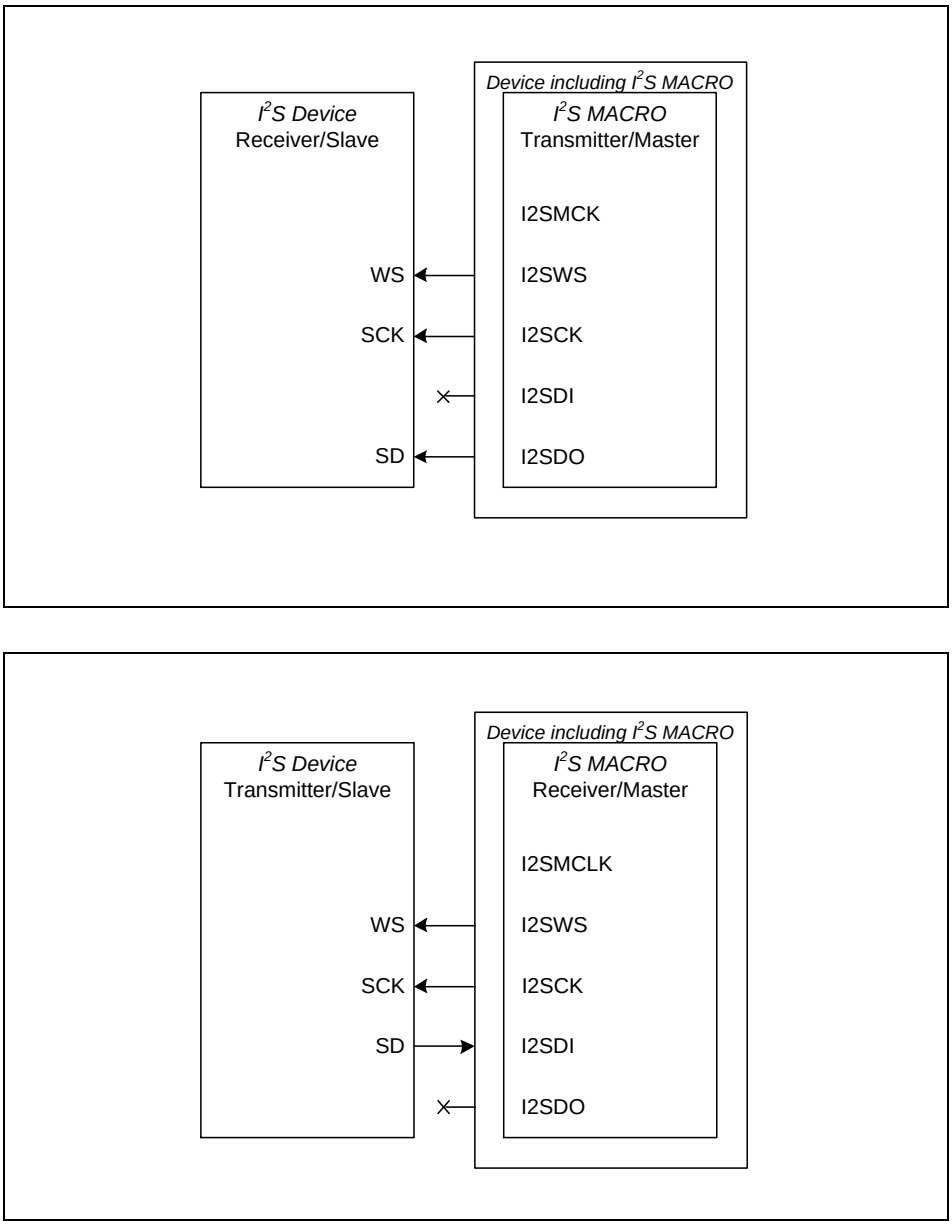
8.1 Connection Diagram

The Figure 8-1 shows the connection diagram. The device including I²S macro is this MCU.

The upper figure is for transmit.

The lower figure is for received.

Figure 8-1 External Connection Diagram



8.2 MFS-I²S and 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. The serial data output from 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. Data sampling is always performed on the rising edge of SCK. Serial data and WS output is always performed on the falling edge of SCK.

The MSB-Justified protocol is similar to I2S. The WS transition point and serial data MSB occur simultaneously. WS indicates the left channel with "1" and indicates the right channel with "0".

Notes:

- *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.*

Figure 8-2 I²S Data Format

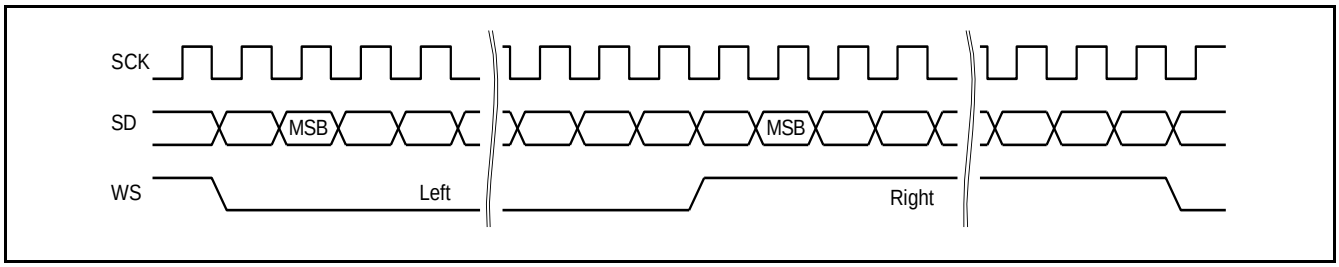
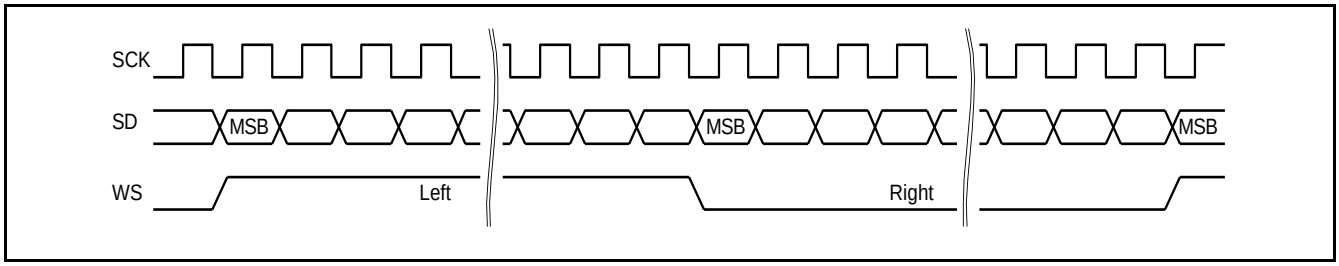


Figure 8-3 MSB-justified Data Format



CHAPTER 2-1: CAN Prescaler

In FM0+ family product, CAN prescaler is not equipped.

This chapter is removed.

CODE: 9BFCANPRE-E01.5

CHAPTER 2-2: CAN Controller

In FM0+ family product, CAN Controller is not equipped.

This chapter is removed.

CODE: FC42L-E02.6

CHAPTER3-1: HDMI-CEC/Remote Control Reception

HDMI-CEC/remote control reception is explained as follows.

1. Configuration
2. Revision
3. Usage notes of HDMI-CEC

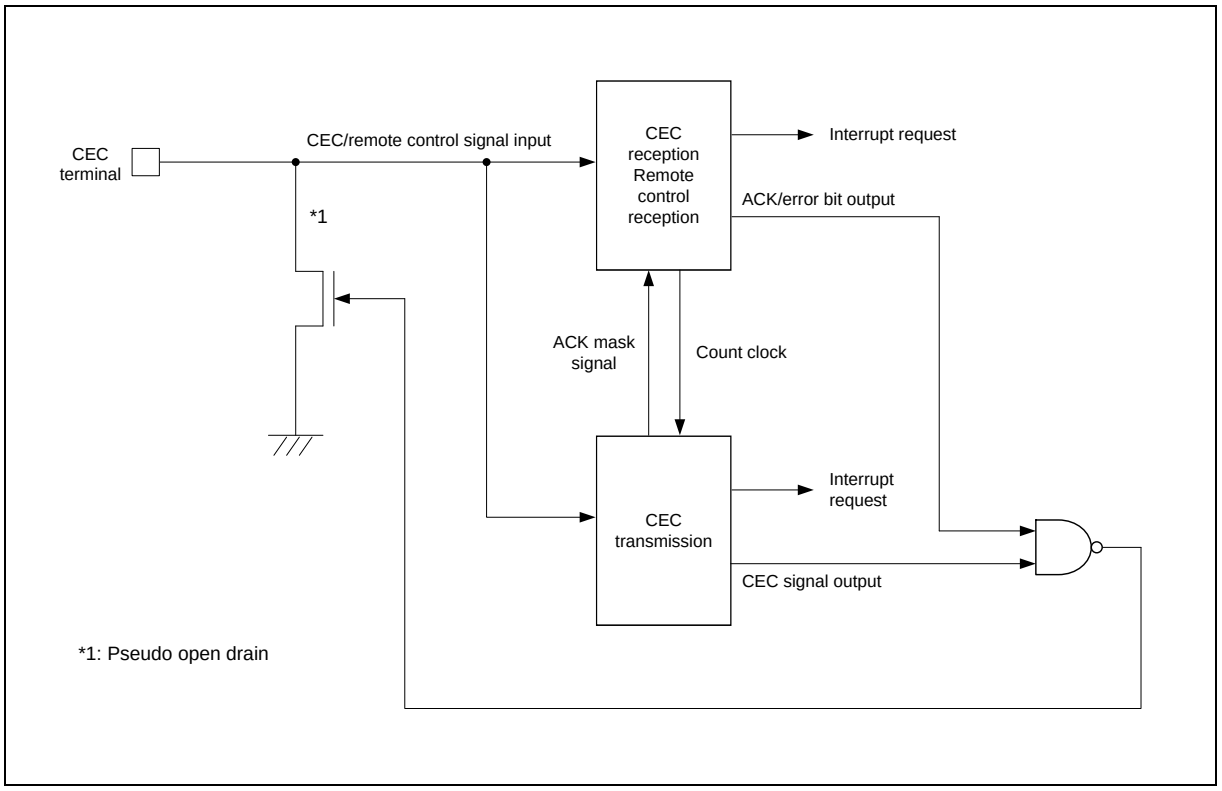
CODE: 9BFRCECTOP-E1.0

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

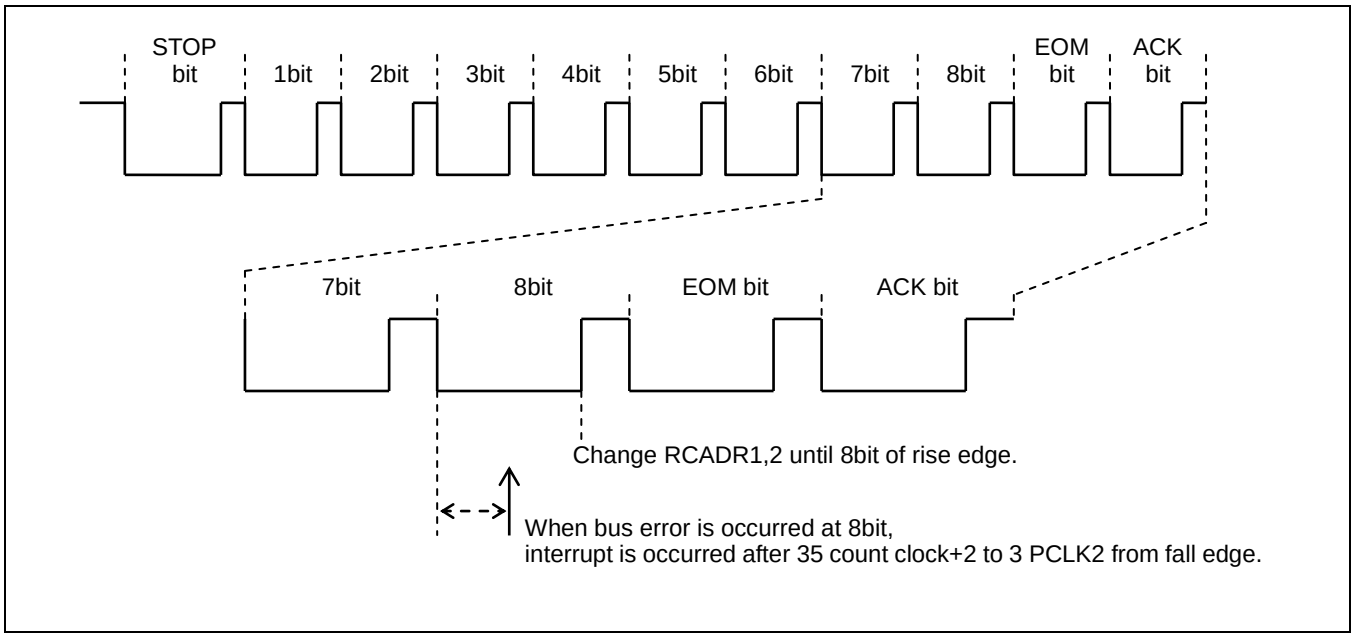
  See a separate chapter "CEC Transmission".

2. Revision

Revision of HDMI-CEC/remote control reception in FM0+ family is RCCEC_rev3.

3. Usage notes of HDMI-CEC

- If RCADR1 or RCADR2 is changed in middle of communication and bus error occurred, change until rise edge of 8bit.



CHAPTER3-2: 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

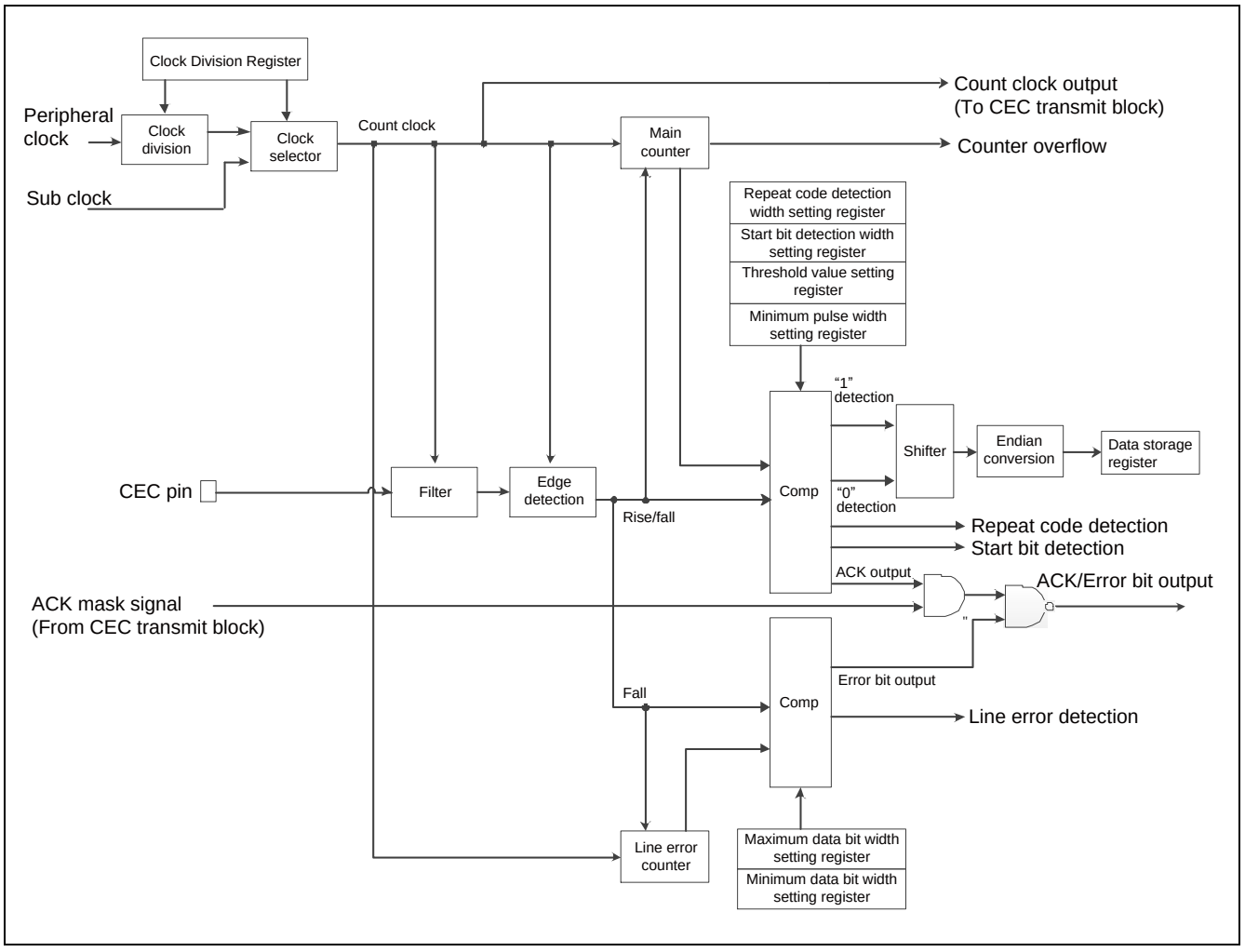
- HDMI-CEC mode
  - Start bit detection and interrupt output
  - Minimum pulse width violation detection
  - Counter overflow detection and interrupt output
  - Device address comparison
  - Minimum data bit width violation detection and interrupt output (supporting HDMI-CEC line error handling standard)
  - 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



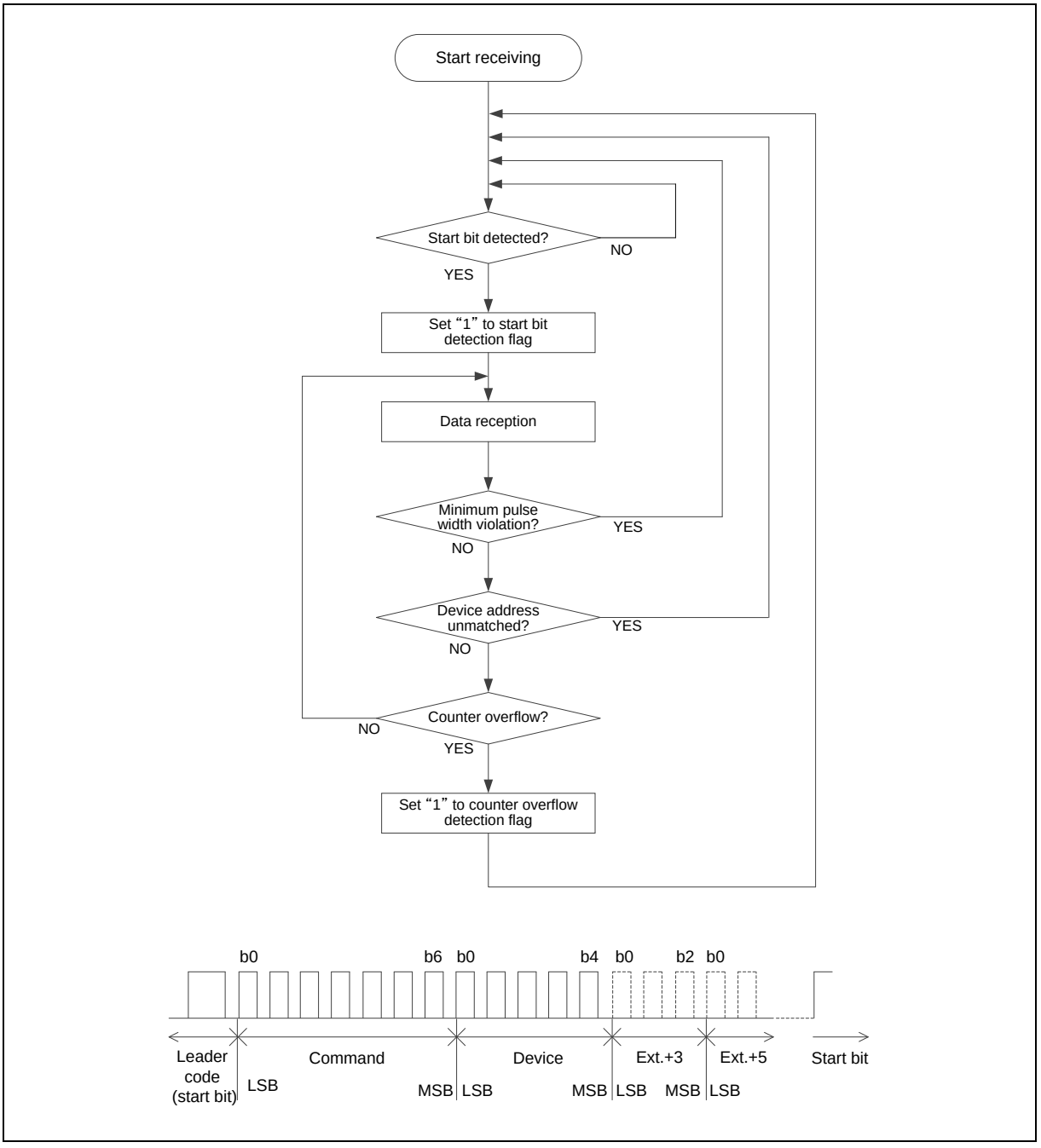
3. Operations

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

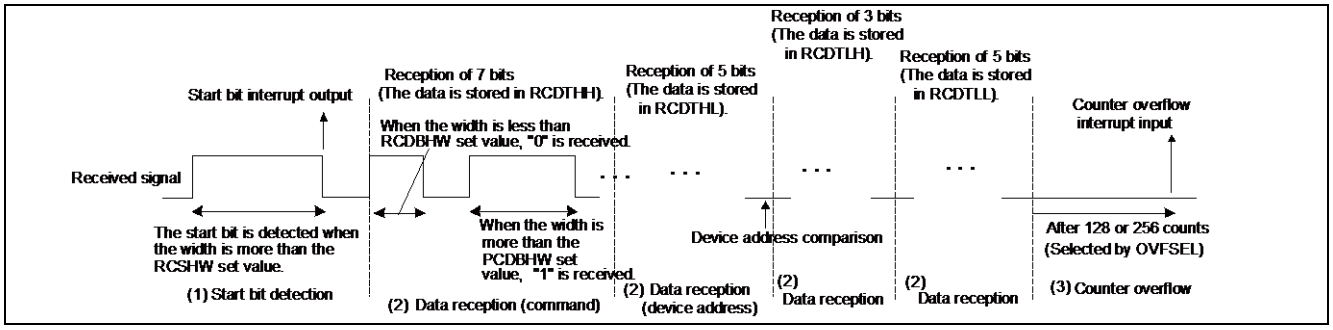
Figure 3-1 Operational Flow Chart and Waves



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

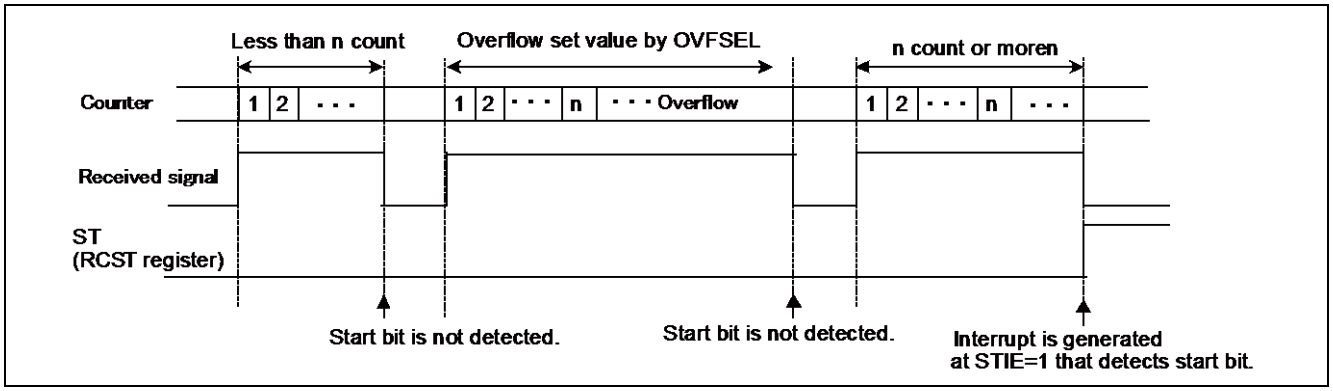


Figure 3-3 explains the start bit detection when RCSHW=n is set.
If the width of "High" duration of "n" or more is input with the start bit detection waited, ST=1 (RCST register) is set by detecting the start bit. Moreover, when STIE=1 (RCST register) is set beforehand, the interrupt is output by detecting 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

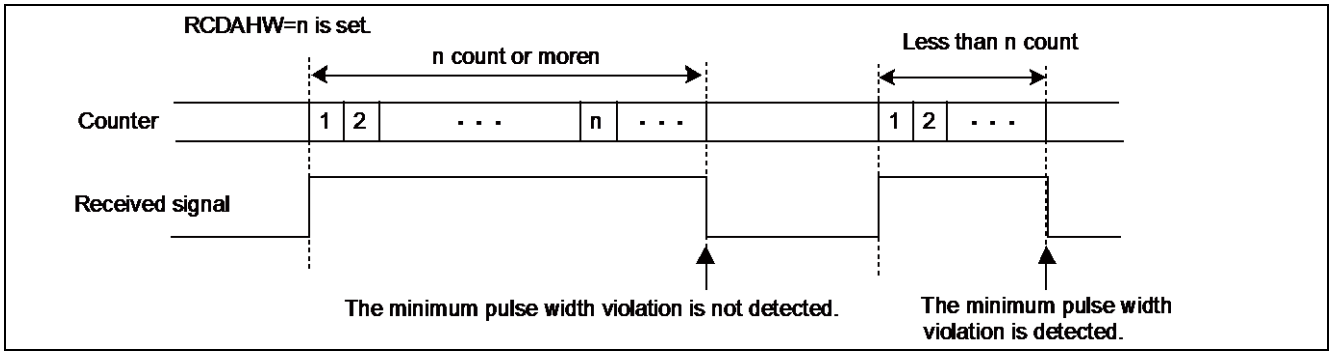


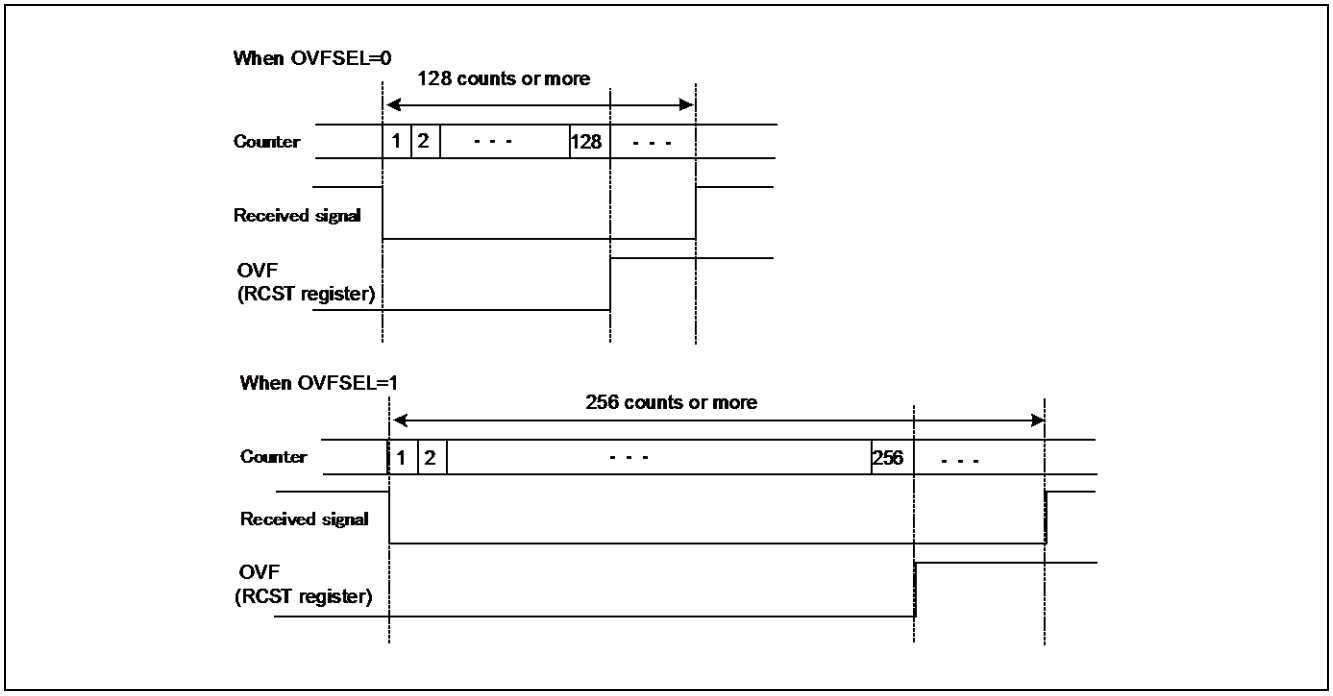
Figure 3-4 explains the minimum pulse width violation when RCDAHW=n is set.
When the signal of less than n is input during the reception operation, the state of the start bit detection waiting is resumed by detecting the minimum pulse width violation.

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



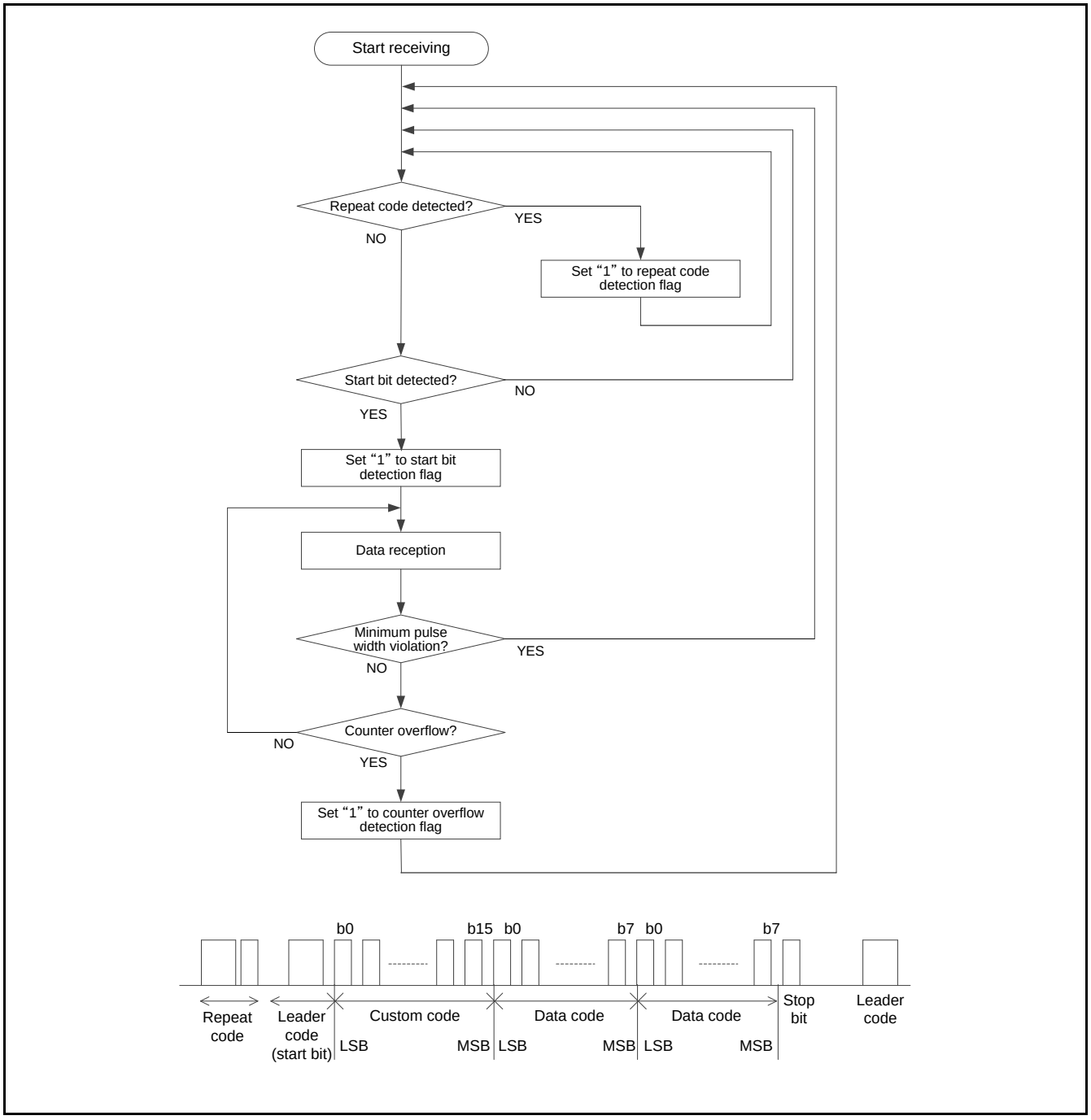
For OVFSEL=0 (RCST register), an overflow occurs and the start bit detection waiting state is resumed when High or Low input continues more than 128 counts. Moreover, for OVFSEL=1, an overflow occurs at 256 counts.

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

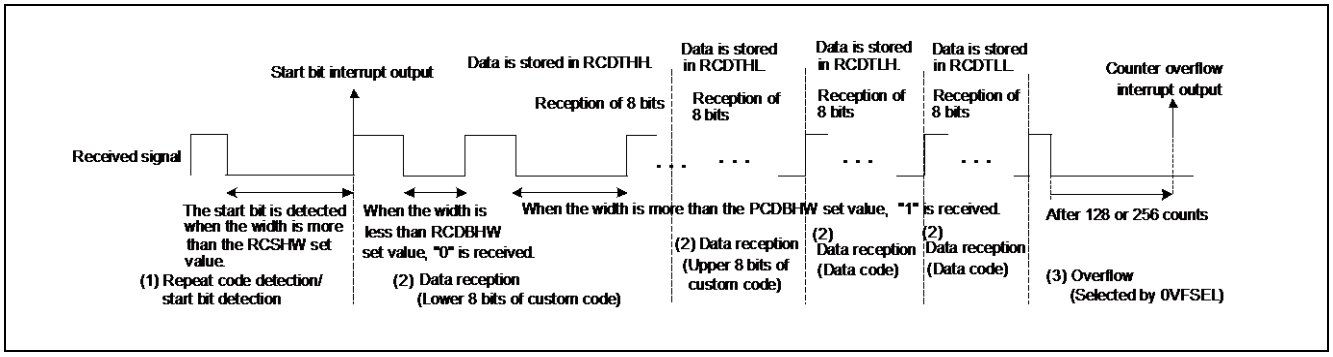
3.2.1 Operational Flow Chart and Waves of NEC/Association for Electric Home Appliances Mode

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

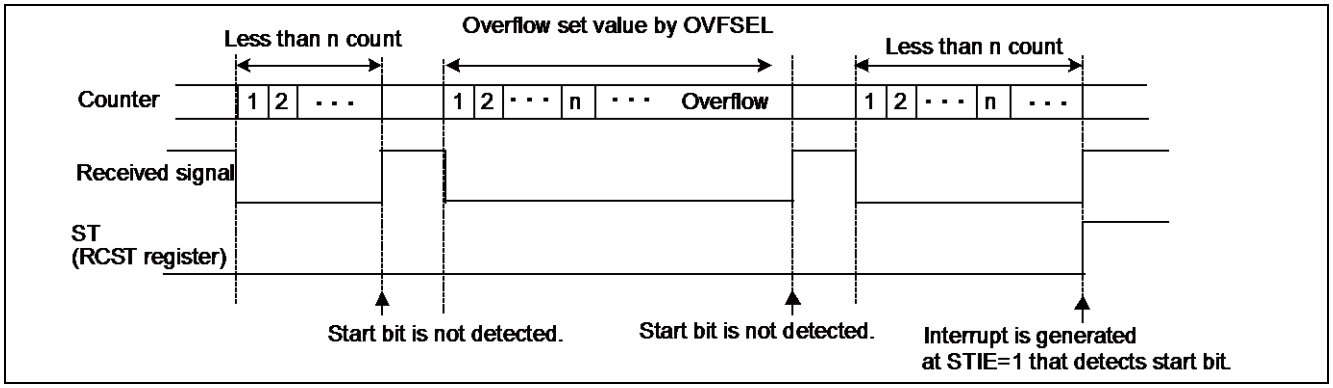


Figure 3-8 explains the start bit detection when "RCSHW=n" is set.
When the width of "Low" duration of n or more is input during the start bit detection waiting, ST=1 (RCST register) is set by detecting the start bit. Moreover, when STIE=1 (RCST register) is set beforehand, the

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

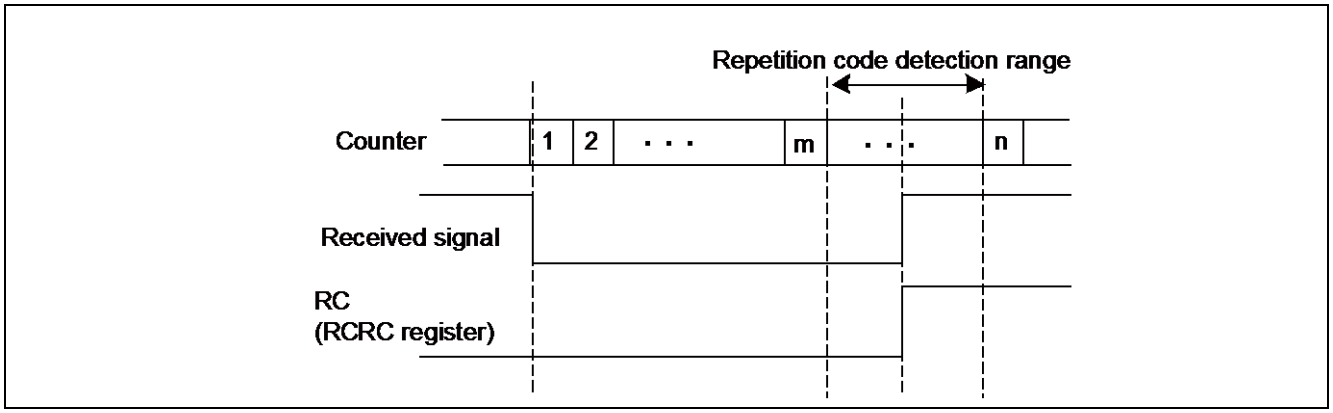


Figure 3-9 explains the start bit detection when RCRHW=m and RCSHW=n are set.
When the "Low" signal of the width of less than n and m or more is input at the reception beginning, RC=1 (RCRC register) is set by detecting the repeat code.
The repeat code is detected only in NEC/Association for Electric Home Appliances mode.

3.2.4 Minimum Pulse Width Violation

Figure 3-10 Minimum Pulse Width Violation

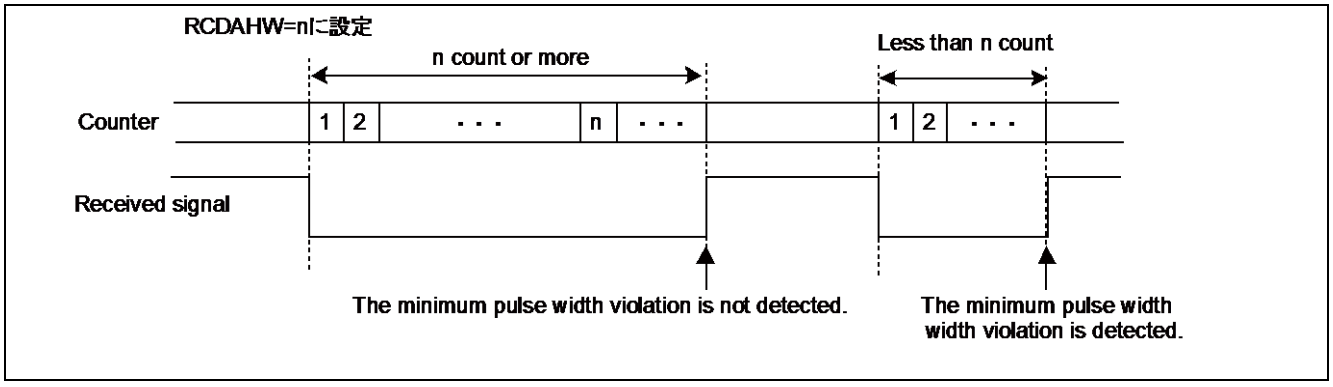
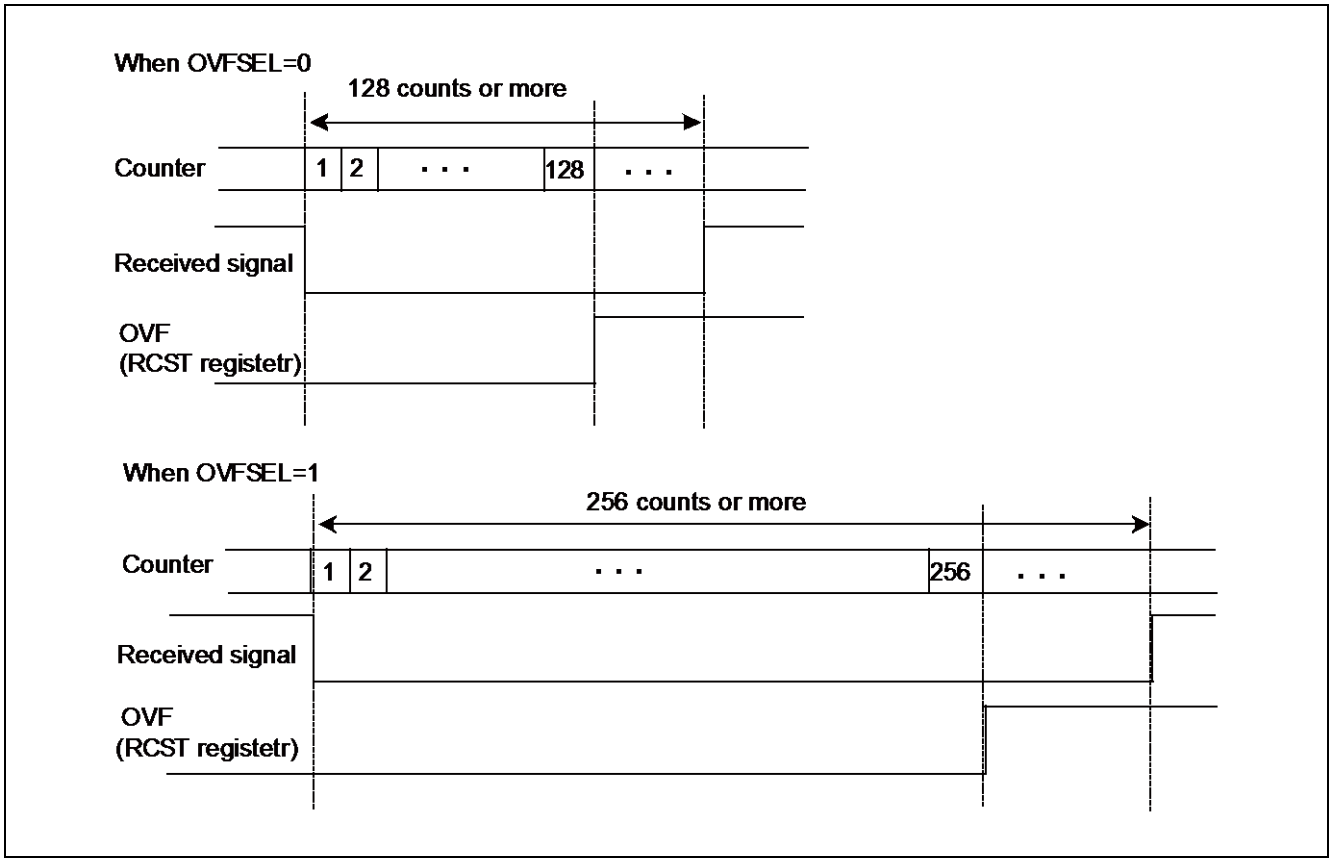


Figure 3-10 explains the minimum pulse width violation when RCDAHW=n is set.
When the width of "Low" duration of less than n is input during the reception operation, the start bit detection waiting state is resumed by detecting the minimum pulse width violation.

3.2.5 Counter Overflow Detection and Interrupt Output

Figure 3-11 Counter Overflow

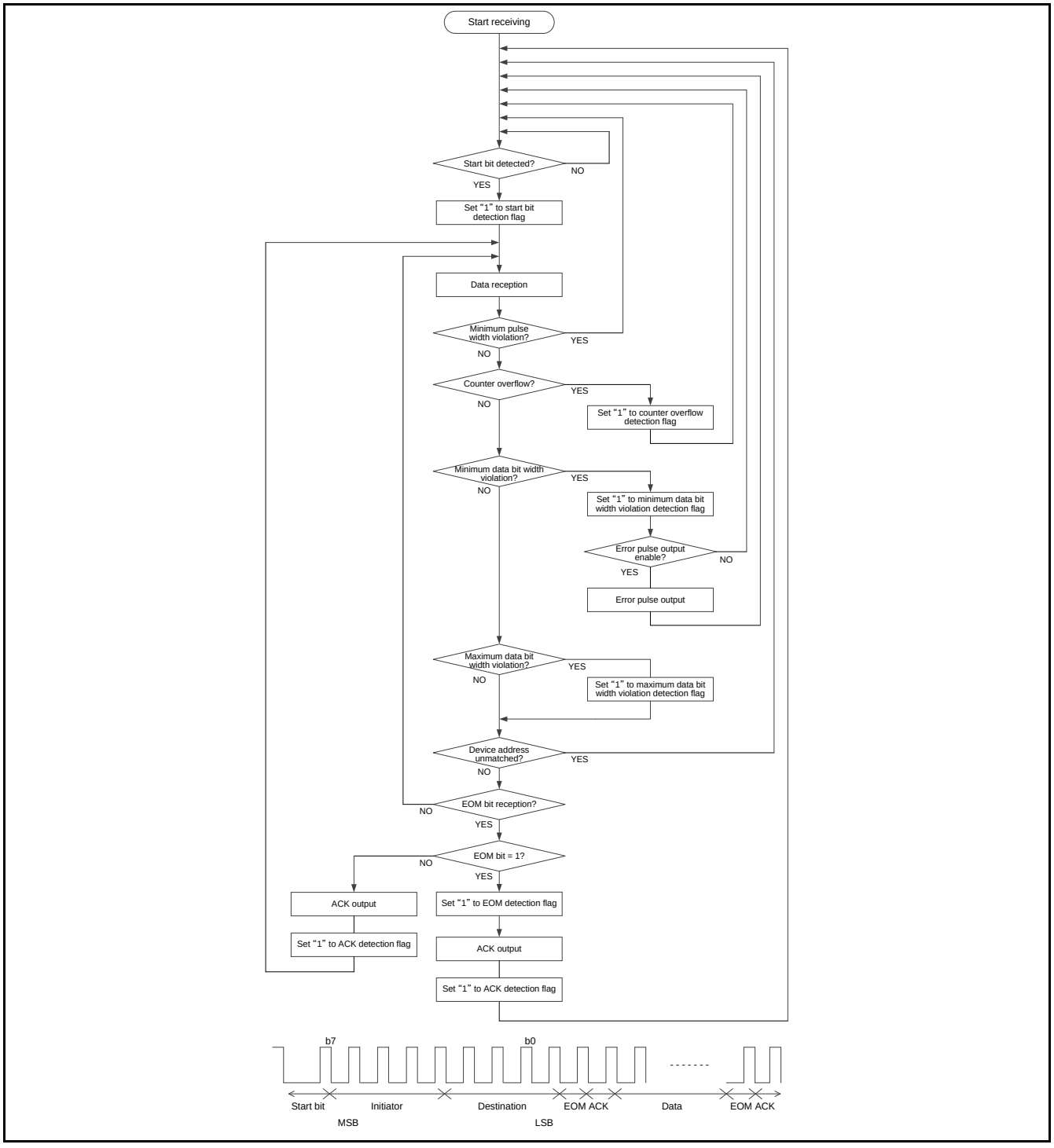


If the "High" or "Low" input of 128 counts or more continues for OVFSEL=0(RCST register), an overflow occurs and the start bit detection waiting state is resumed. Moreover, an overflow occurs with 256 counts of the continuous "High" or "Low" input for OVFSEL=1.
When OVFIE=1 (RCST register) is set beforehand, an overflow occurs and an interrupt is output.

3.3 HDMI-CEC Mode

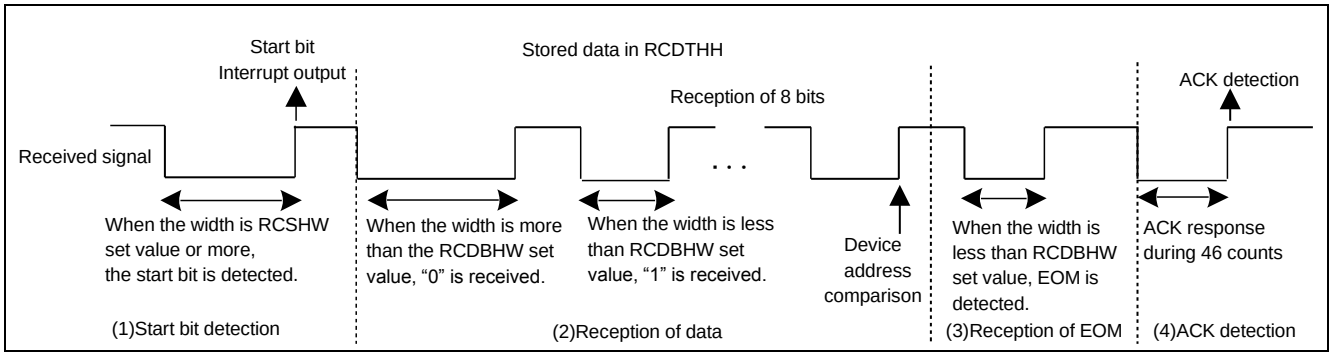
3.3.1 Operational Flow Chart and Waves in HDMI-CEC Mode

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 the RCSHW set value or more 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

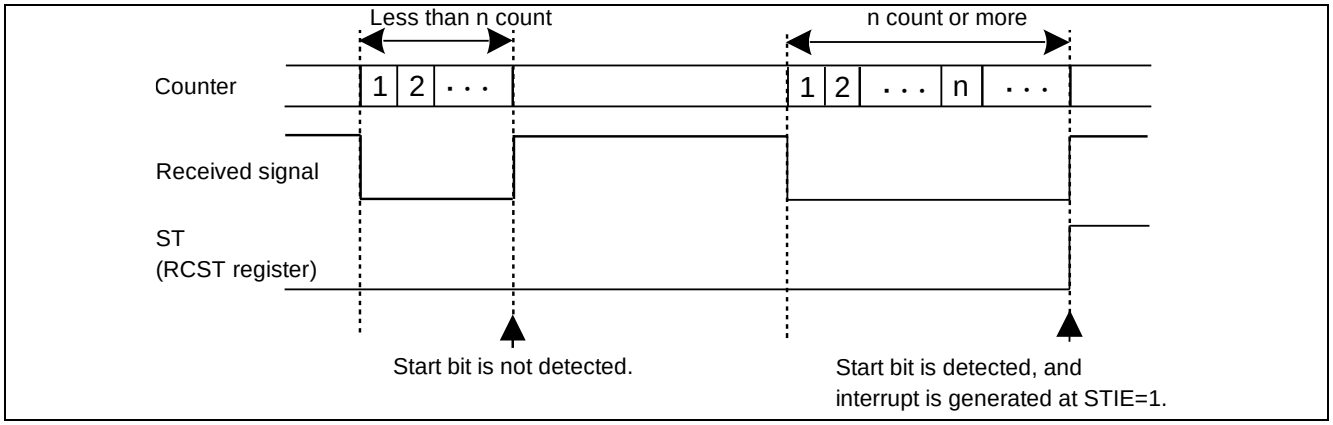


Figure 3-14 shows the start bit detection when "RCSHW=n" is set. When the width of "Low" duration of n or more is input with the start bit detection waiting, the start bit is detected and ST=1 (RCST register) is set. Moreover, when STIE=1 (RCST register) is set beforehand, the interrupt is output by detecting the start bit.

3.3.3 Minimum Pulse Width Violation

Figure 3-15 Minimum Pulse Width Violation

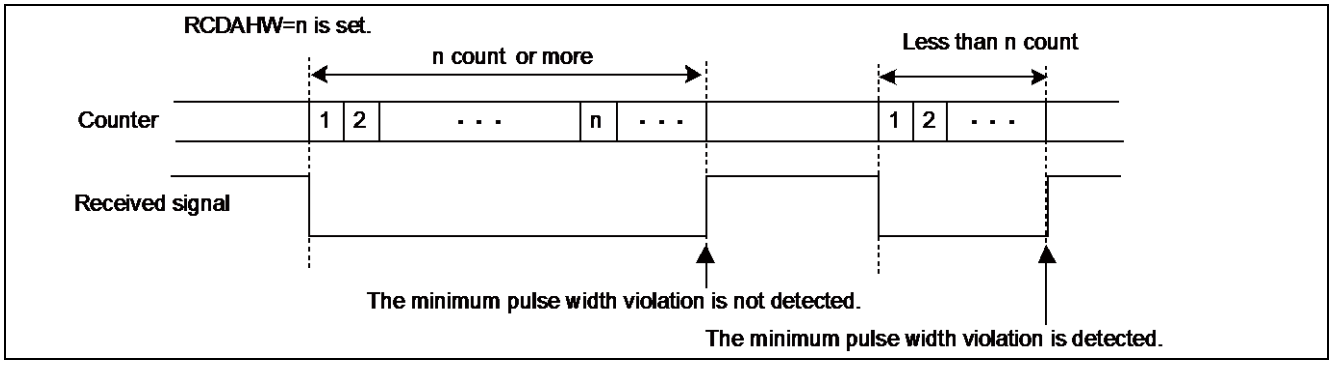
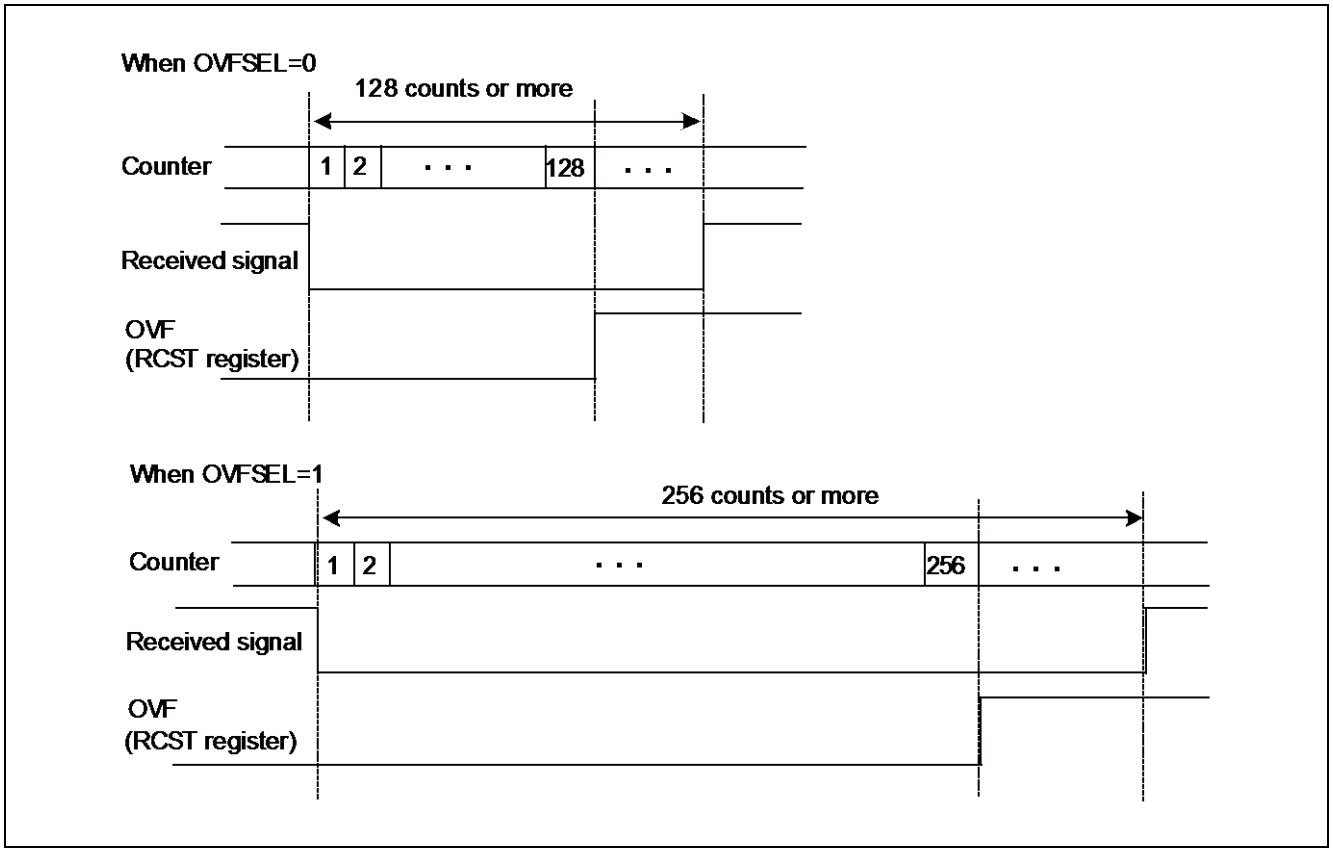


Figure 3-15 shows the minimum pulse width violation when RCDAHW=n is set.
When the signal of less than n is input during the reception operation, the minimum pulse width violation is detected and the start bit detection waiting state is resumed.

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

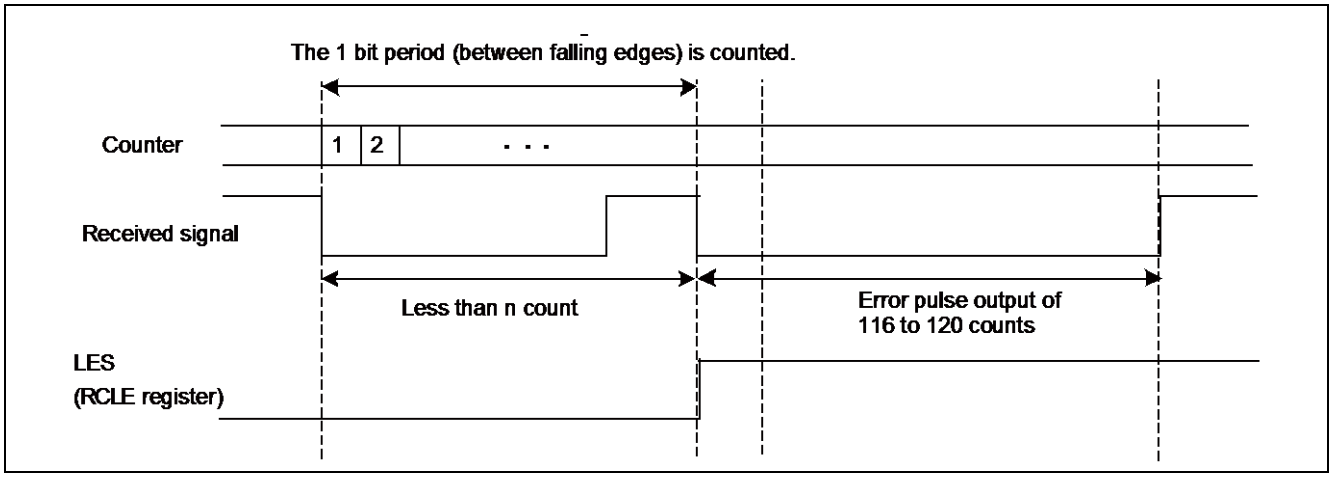


Figure 3-17 explains the minimum data bit width violation when RCLESW=n is set.
At LES=1 (RCLE register), when the 1 bit period (the period between the falling edges) is smaller than the set value of minimum data bit width setting register (RCLESW), the minimum data bit width violation is detected and 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

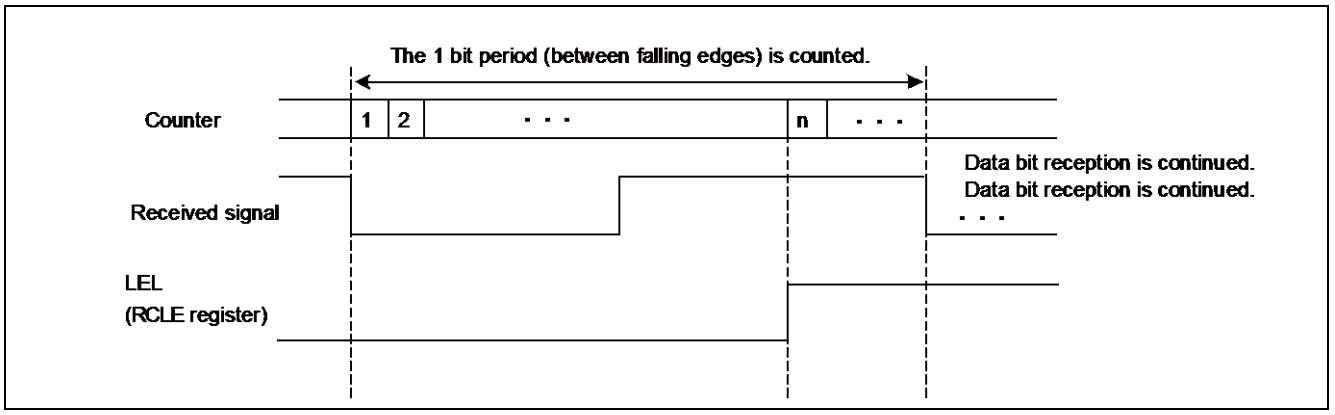


Figure 3-18 explains the minimum data bit width violation when RCLELW=n is set.
For LELE=1 (RCLE register), when the 1 bit period (the period between the falling edges) is more than the set value of maximum data bit width setting register (RCLELW), LEL=1 (RCLE register) is set by detecting the maximum 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

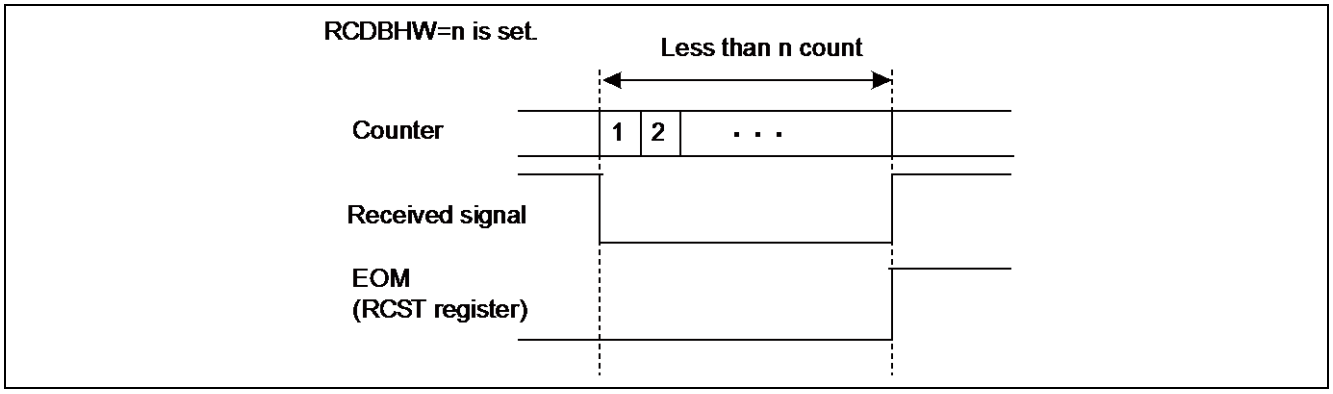
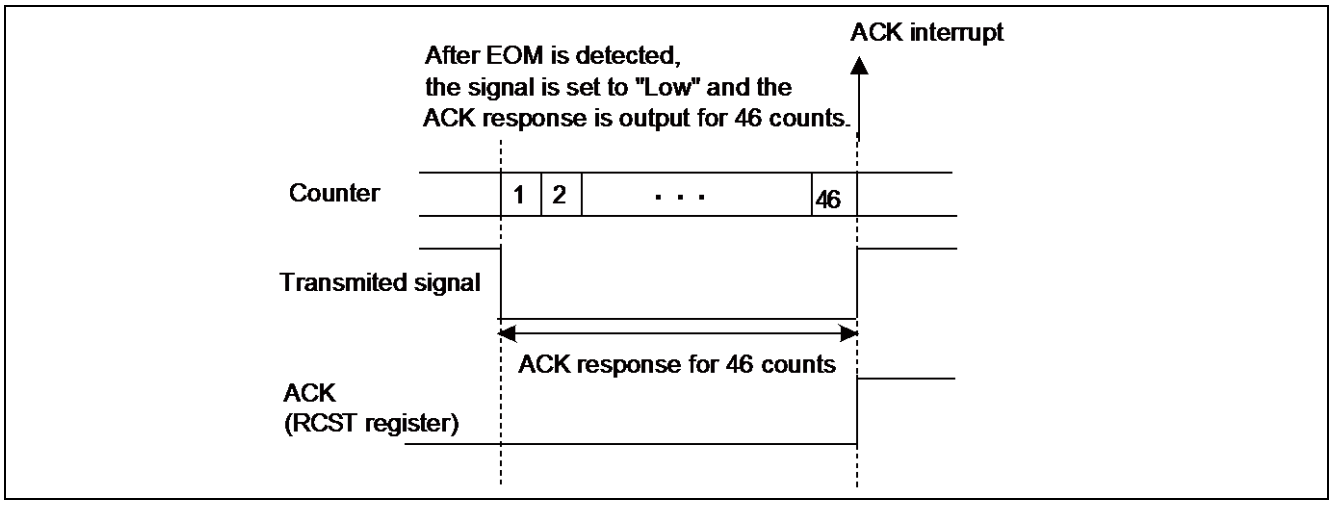


Figure 3-19 shows the operation for THSEL=1 (RCCR register). If the "Low" signal of less than RCDBHW set value is input in EOM bit receiving state, EOM=1 (RCST register) is set by detecting EOM.

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.

Table 3-1 ACK output and ACK interrupt

Received destination address	ADRCE	RCADR1, RCADR2		ACK output*	ACK interrupt
0x0 to 0xE	0	-		ACK	occur
	1	0x00 to 0x0E	match	ACK	occur
			not match	NACK	not occur
		0x0F		NACK	not occur
0xF	-	-		NACK	occur

*: When ACKMEN bit of CEC transmission unit is 1 and during transmission, it will always be NACK.

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, and RCCKD registers
  RCRC, RCRHW, RCLE, RCLELW, and RCLESW registers

  If RCADR1, RCADR2 is changed while this bit is "1", see CHAPTER 3-1: HDMI-CEC/Remote Control Reception 3. Usage notes of HDMI-CEC.

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.*

CHAPTER3-3: 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.

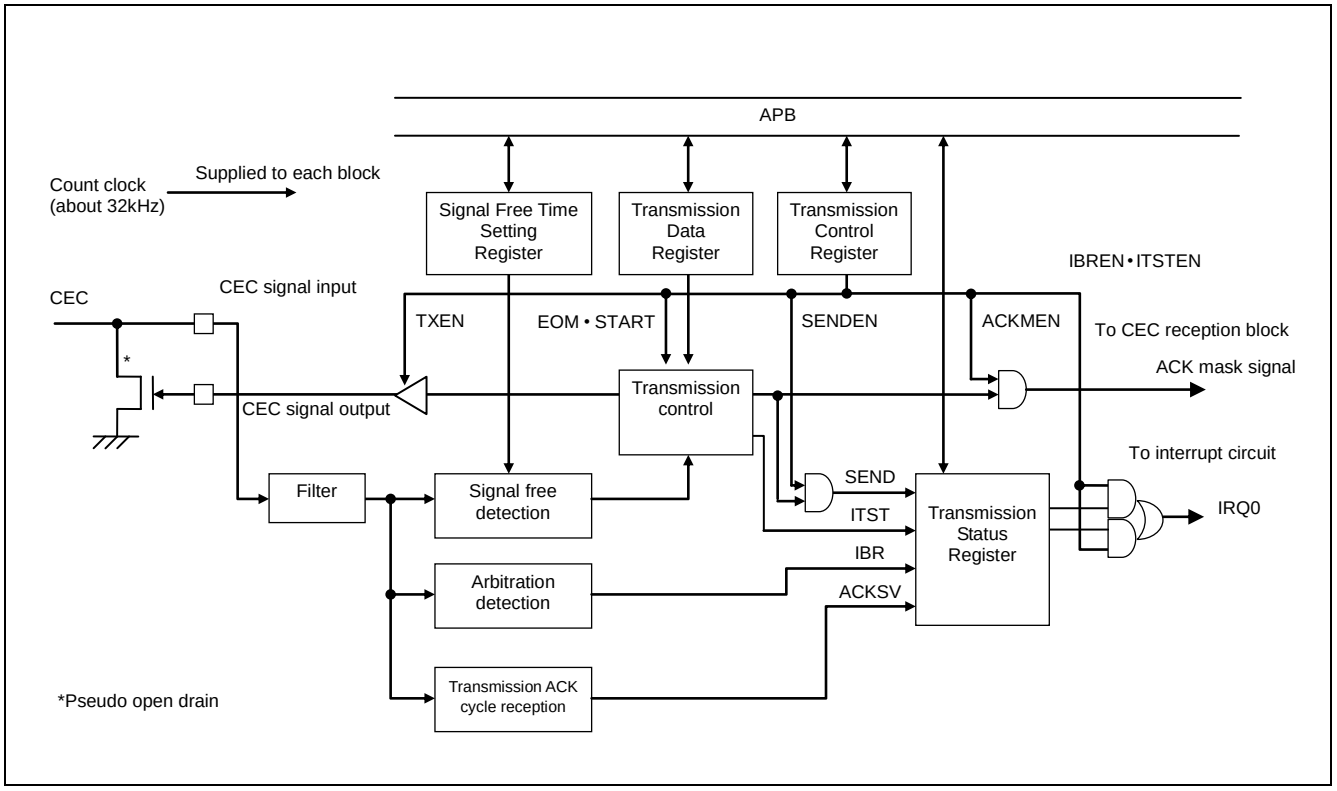
Data Transmission

- 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.

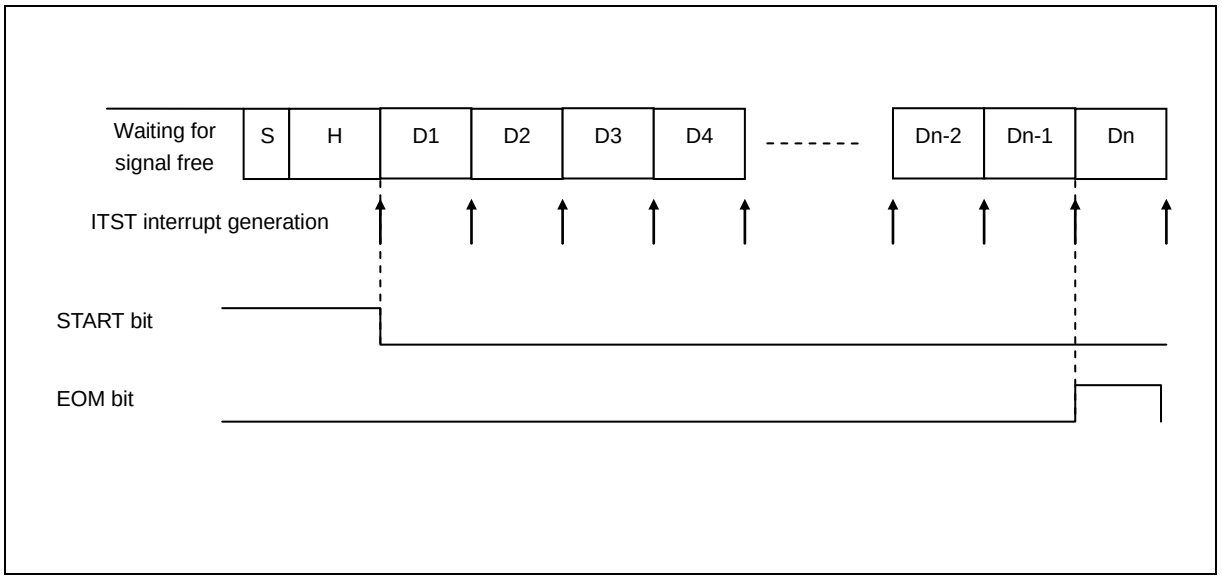
Basic Operations

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



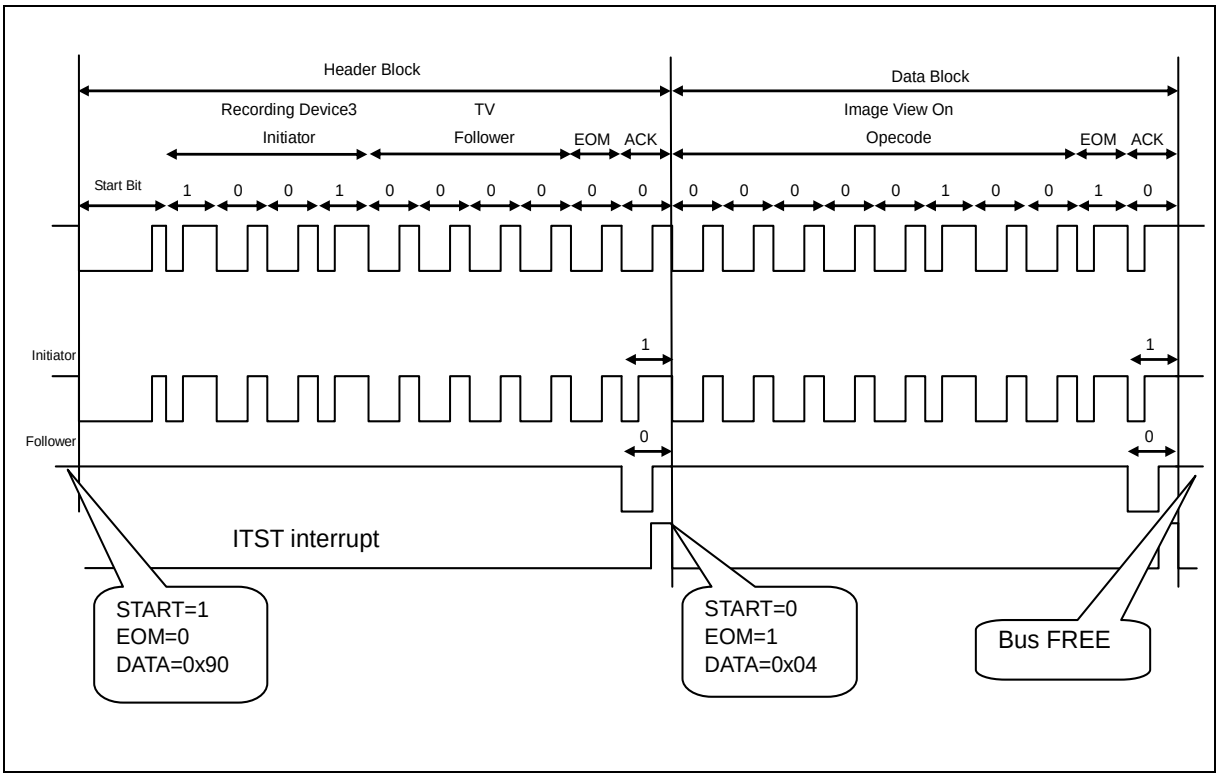
5.2 Interrupt Factors and Timing Chart

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

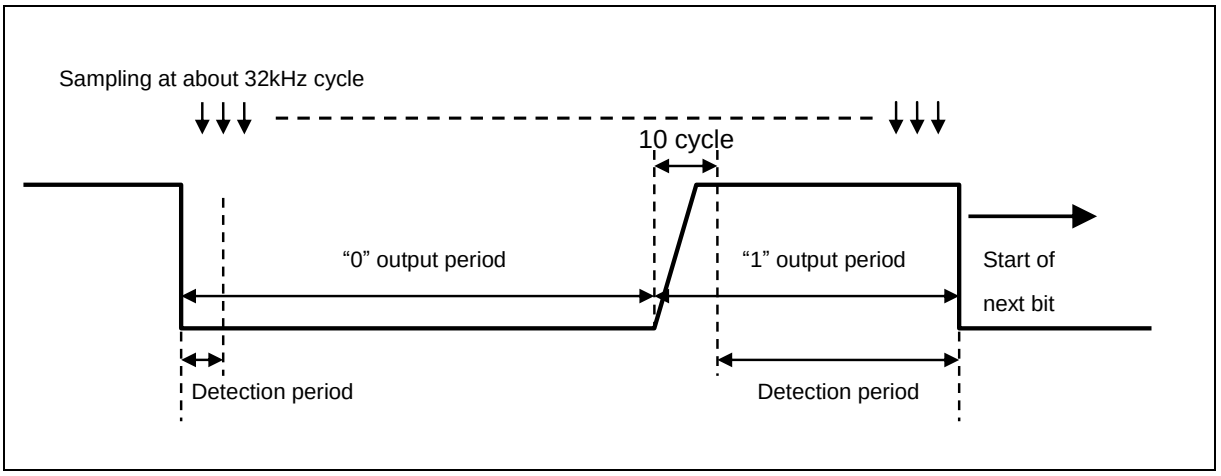
Arbitration lost detection is as follows.

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 continuously 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

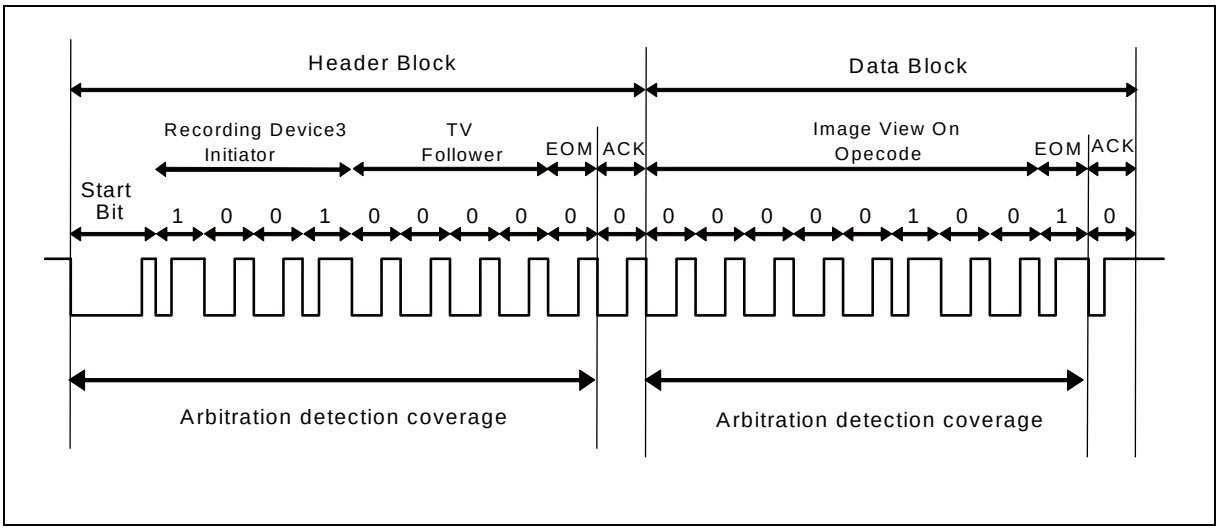


Detection Coverage of Arbitration Lost

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

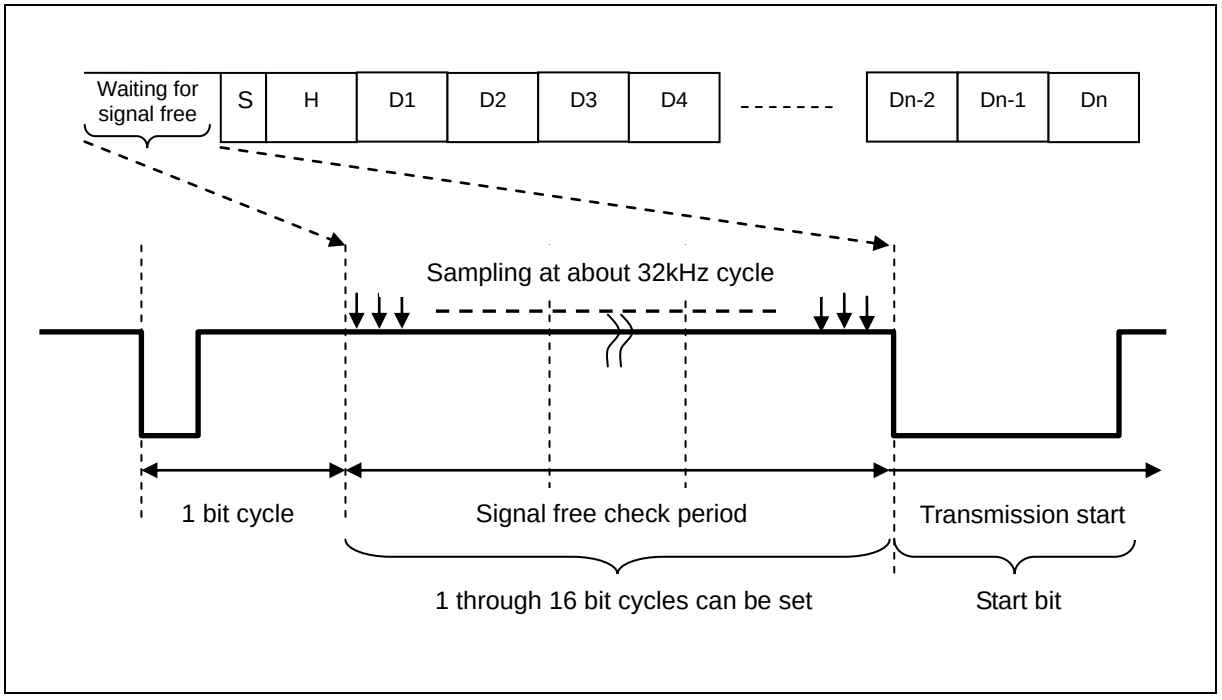
Signal free detection is as follows.

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



If 5bit signal free time from last fall edge of previous frame to fall edge of start bit should be secured, set "3" to signal free setting register.

If signal free time should be secured 5bit after other device transmission except this device address, it is available. If SEND bit at the start bit detection interrupt reception is "0", it is possible to determine the transmission from the other device.

5.5 Filtering

Filtering CEC signal input of transmission side is described as follows.

Filtering CEC Signals

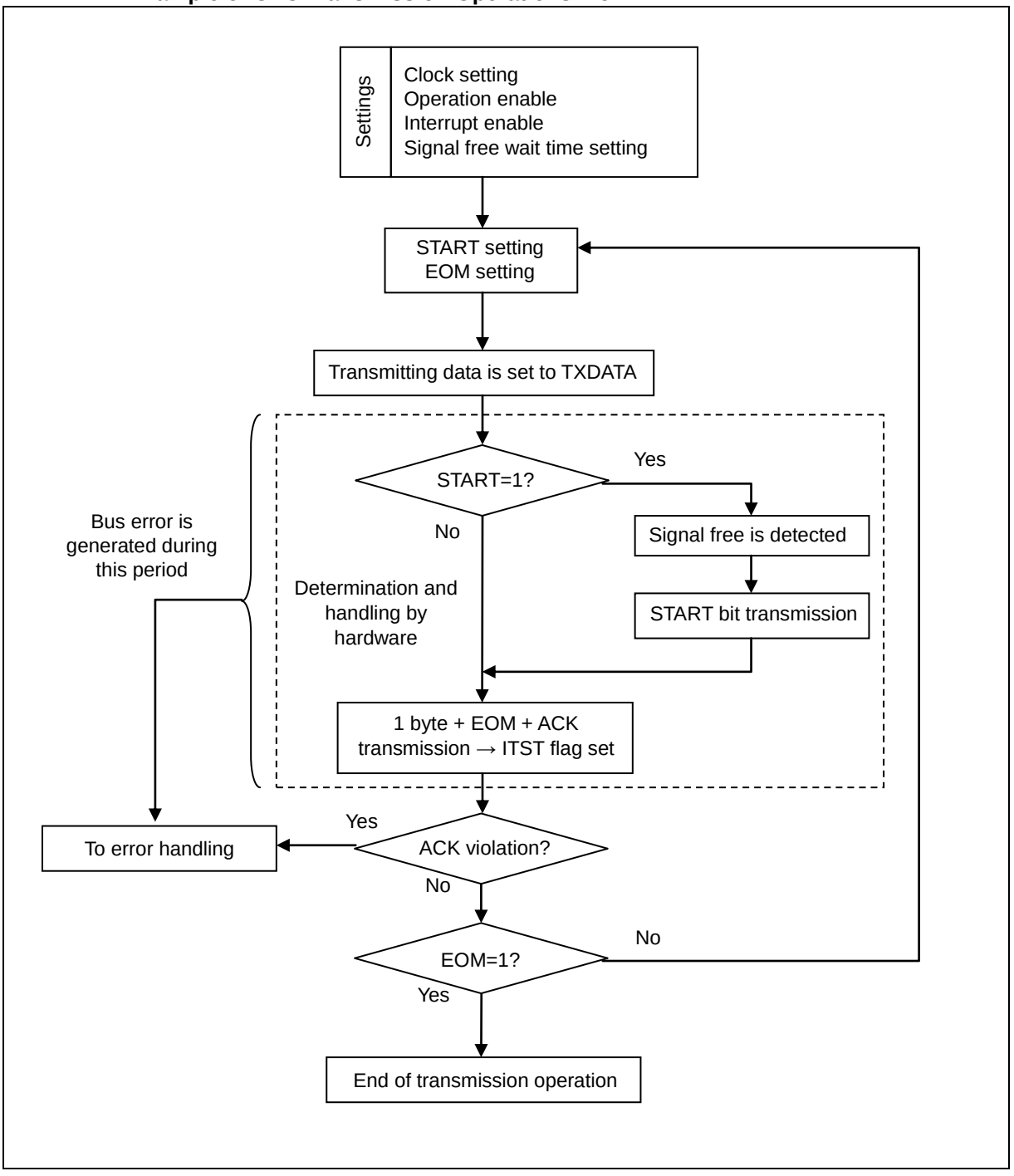
If a CEC signal input is changed within a width less than 2 count clocks, it is determined as noise and the signal will be removed.

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	SENDEN	ACKMEN	IBREN	ITSTEN	EOM	START	Reserved	TXEN
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] SENDEN: Sending flag enable bit

- This bit controls operation of SEND bit in Transmission status register (TXSTS).

Bit	Description
0	Disables SEND bit operation
1	Enables SEND bit operation

[bit6] ACKMEN: ACK mask enable bit

- This bit controls ACK mask.
- When the ACKMEN bit is 1 and sending, ACK output is masked.

Bit	Description
0	Disables ACK mask
1	Enables ACK mask

[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	SEND	Reserved	IBR	ITST	Reserved			ACKSV
Attribute	R	R/W	R/W	R/W	R/W			R
Initial Value	0	0	0	0	000			0

[bit7] SEND: Sending flag bit

- This bit indicates that CEC transmission is sending.
  If SENDEN bit is "1" and CEC transmission is sending from start of start bit to end of ACK bit, this bit is "1".
- When SENDEN bit is "0", this bit is "0".
  It is invalid to set this bit.

Bit	Description
0	Not sending or SEND bit is "0"
1	Sending (When SENDEN is "1")

[bit6] Reserved: Reserved bit

"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	

CHAPTER4-1: USB Clock Generation Block Overview

This chapter explains the USB clock generation block overview.

1. Overview and Configuration

CODE: 9BFBSPLL_FM0-E03.0

1. Overview and Configuration

Generating USB Clock

This block generates a 48 MHz USB clock used in USB macro communication.

Since the function and configuration differ by products, see the chapter "USB Clock Generation (A)" for the TYPE2-M0+ product, and see the chapter "USB Clock Generation (B)" for the TYPE3-M0+ product.

Furthermore, for logic macros of USB 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 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".

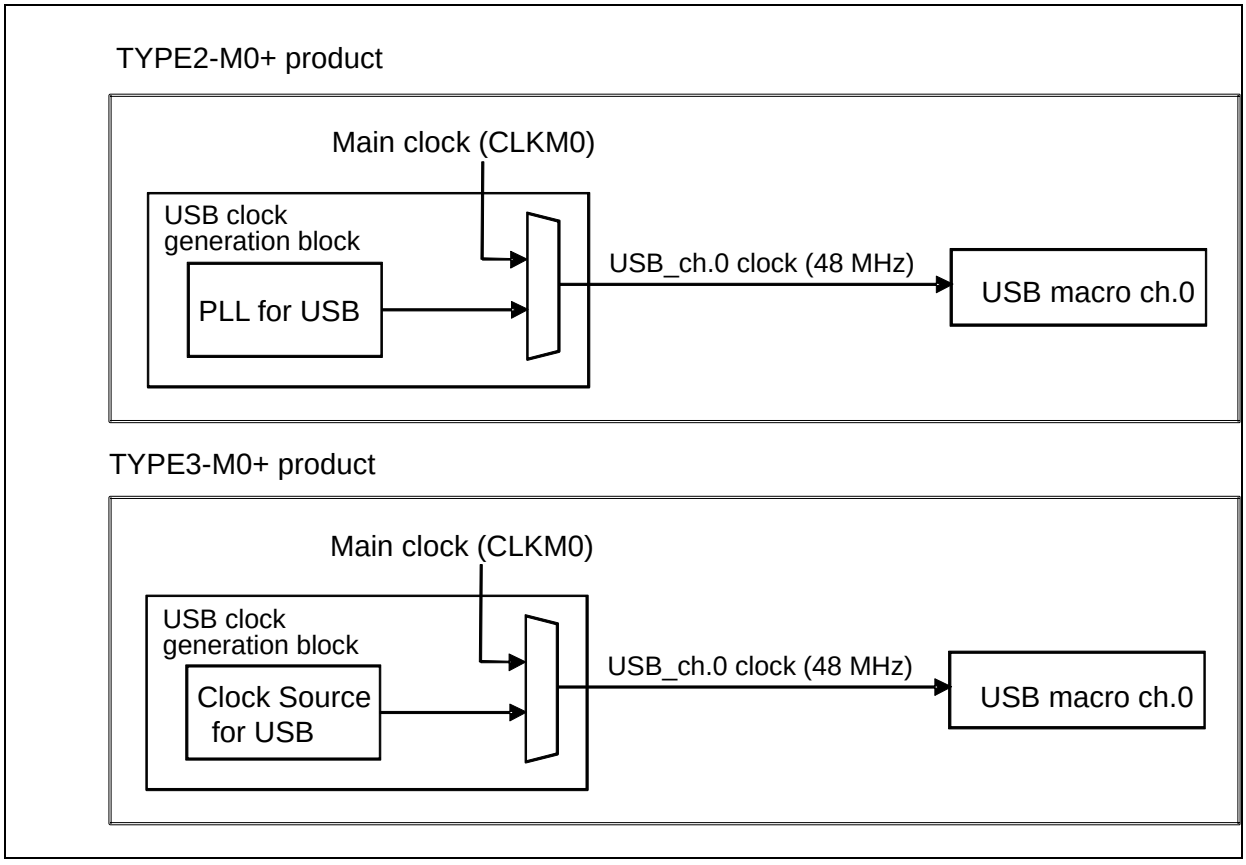
Table 1-1 shows product types and those reference chapter.

Table 1-1 Product type and reference chapter

Product TYPE	Function USB	Reference Chapter
TYPE1-M0+	-	-
TYPE2-M0+	○	USB Clock Generation (A)
TYPE3-M0+	○	USB Clock Generation (B)

Figure 1-1 shows a block diagram of a USB clock and a USB/Ethernet clock generation block.

Figure 1-1 Block Diagram of USB Clock Generation Block



CHAPTER 4-2: USB Clock Generation (A)



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

CODE: 9BFUSBETHERPLL_FM0-E03.0

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..

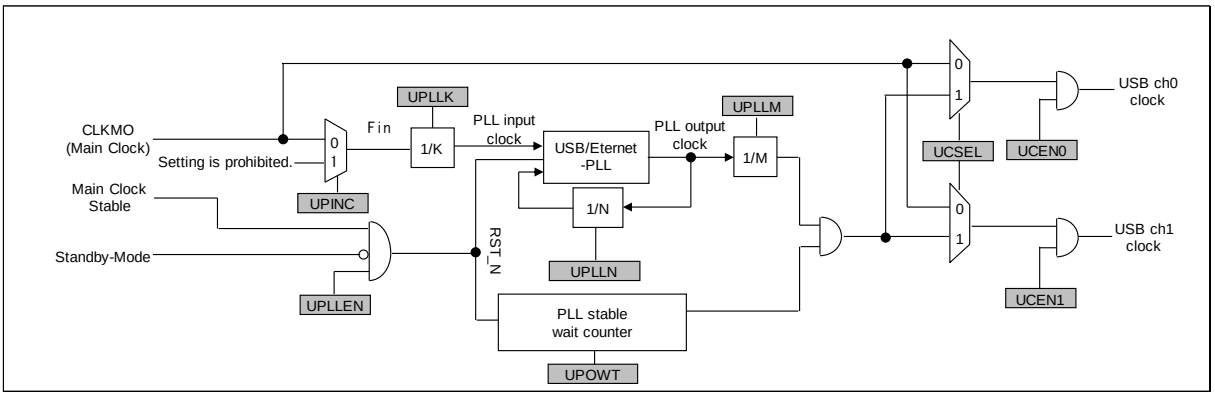
The USB clock generation unit is responsible for the following functions:

- 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.

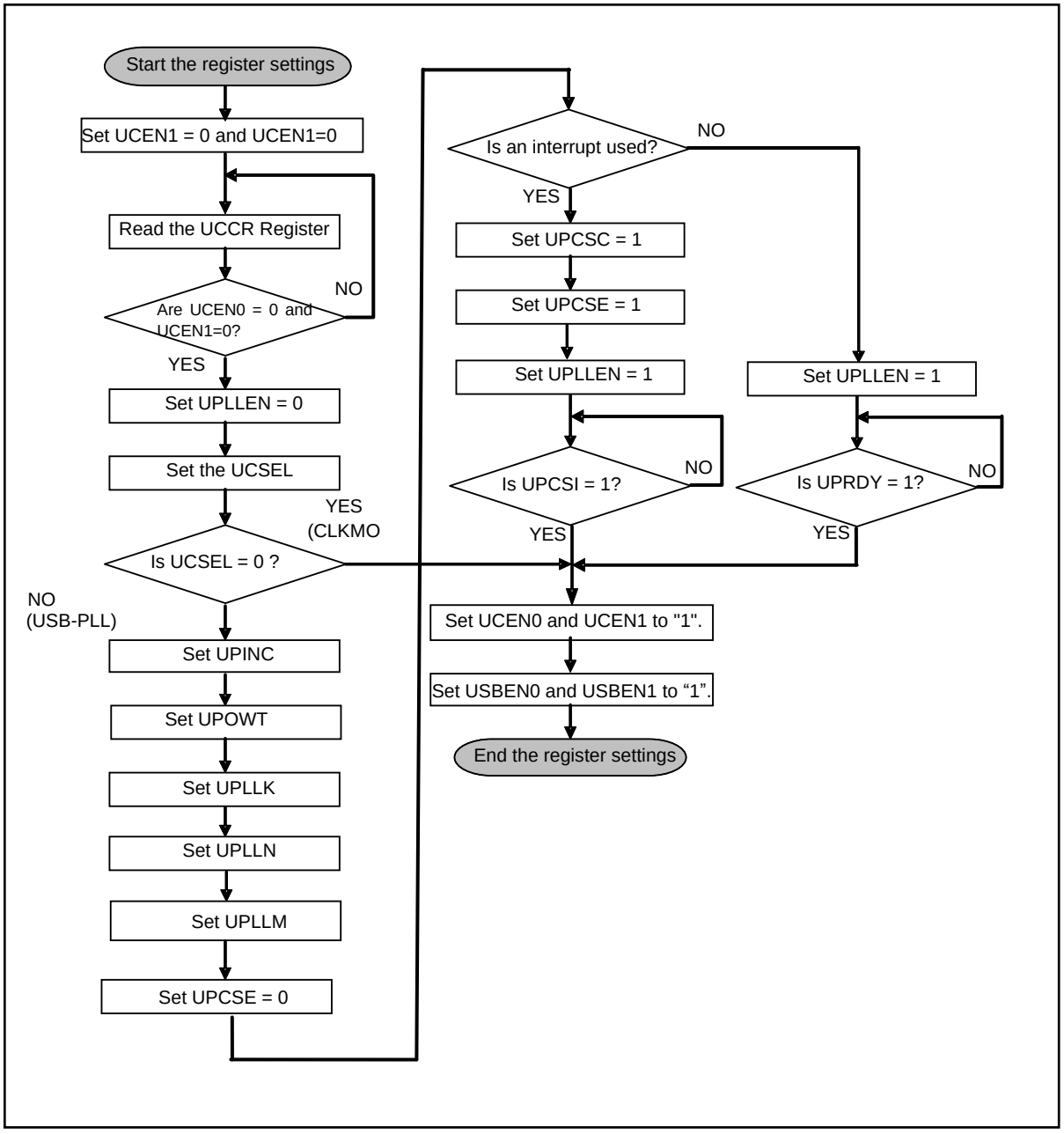
Before enabling the USB-PLL oscillation, configure the oscillation stabilization wait time for USB-PLL and the oscillation stabilization complete interrupt. Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

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.

- 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.

CHAPTER4-3: USB Clock Generation(B)

This chapter explains the USB clock generation.

1. Overview
2. Configuration and Block Diagram
3. Explanation of Operation
4. Setup Procedure Example
5. Register
6. Usage Precautions

CODE: 9BFUSBPLLB_FM0-E03.0

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.
- Main PLL (PLLOUT clock) is used for the clock source..

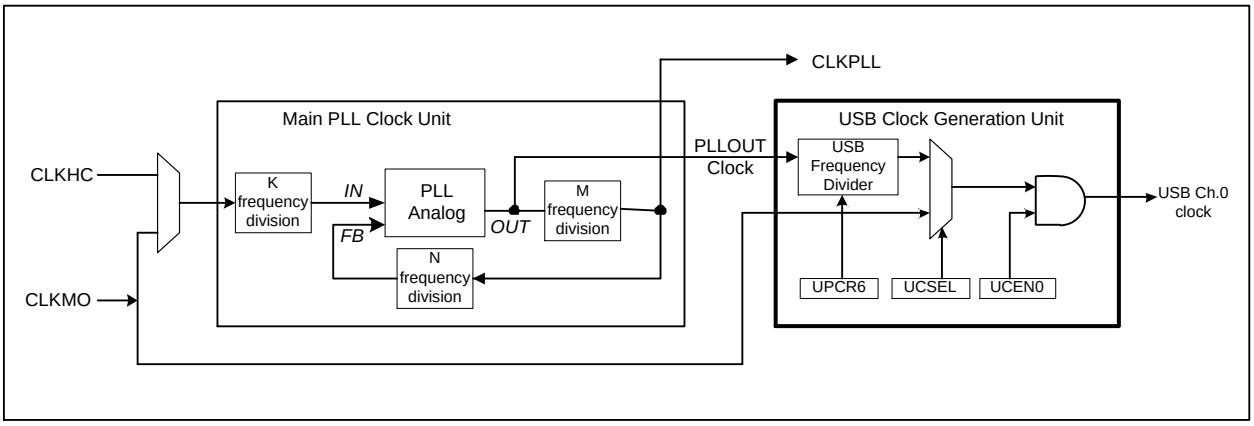
The USB clock generation unit is responsible for the following functions:

- Enables or stops output of the USB clock.
- Selects the USB clock.
- Sets the PLLOUT clock frequency division for USB clock.

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 Frequency Divider

Generate divided clock from PLLOUT clock. The output clock frequency is divided by UPCR6.

USB clock has to be 48MHz. Therefore, Main PLL and UPCR6 have to be set so that 48MHz clock can be provided to USB ch.0.

Output Clock

- Output Clock Select Register (UCSEL)

  Can be selected from CLKMO or divided clock of PLLOUT clock.

- USB Clock Output Enable Register (UCEN0)

  Can set the USB clock output enable.

3. Explanation of Operation

This section explains the operation of the USB clock generation unit.

3.1 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.

■ Main PLL output clock

Divided clock of the Main PLL output clock can be used as the source clock of USB clock.

The Main PLL output clock must be output at 96 MHz or 144 MHz to generate a 48 MHz clock after USB Frequency divider. Below shows the setting example of the division ratio.

Table 3-1 Example of PLL Frequency Division Ratio Settings

Fin (MHz)	Main PLL				USB Frequency Divider		CLKPLL Output Clock Frequency [MHz]
	Division Ratio Settings			PLL Output Frequency [MHz]	Division Rate Settings	USB clock Frequency [MHz]	
	K	N	M				
8	1	4	3	96	2	48	32
12	1	3	4	144	3	48	36
16	1	2	3	96	2	48	32
16	2	4	3	96	2	48	32
24	2	3	4	144	3	48	36
24	3	4	3	96	2	48	32

3.2 Changing to Standby Mode

■ When changing to standby mode

Before changing to standby mode (STOP mode, or TIMER mode), set UCEN0 of UCCR register to "0" to stop the USB clock supply.

1. Set UCCR:UCEN0=0.
2. Read the UCCR Register to check that UCEN0 is set to "0".
3. Changing to standby mode.

When returning from standby mode, set UCEN0 bit 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 Main PLL (PLLOUT) is used

After the Main PLL oscillation has been stabilized, supply the USB clock.

b) When CLKMO (48 MHz) is used

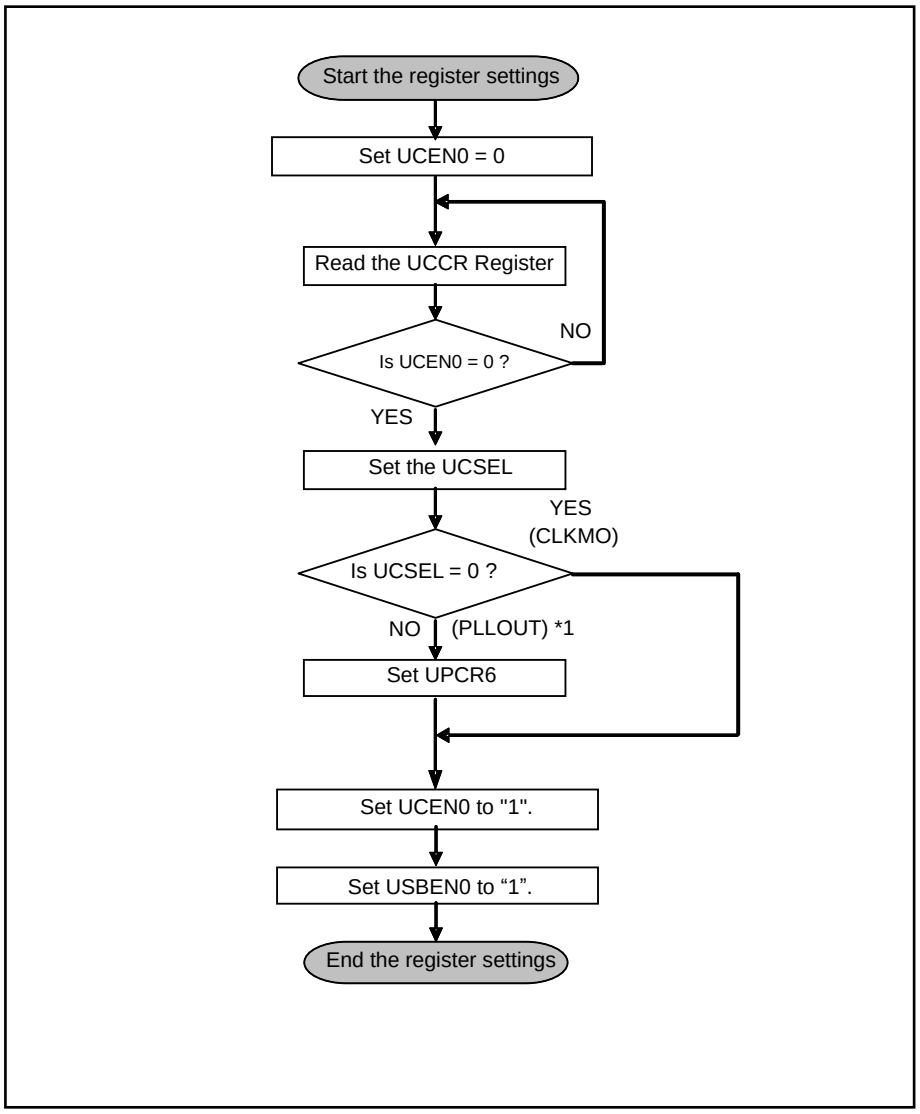
After the CLKMO oscillation has been stabilized, supply the USB clock.

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



*1 : In order to use PLLOUT clock, Main PLL has to be set to generate a clock ahead of setting USB clock.

5. Register

5.1 Register List

This section explains the register list of the USB clock generation unit.

Table 5-1 The Register List of the USB Clock Generation Unit

Abbreviation	Register Name	Reference
UCCR	USB Clock Control Register	5.2
UPCR6	USB-PLL Control Register 6	5.3
USBEN0	USB (ch.0) Enable Register	5.4

5.2 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					UCSEL[1:0]		UCEN0
Attribute	-					R/W		R/W
Initial value	-					00		0

Register Functions

[bit7:3]Reserved: Reserved bit

"0" is read from this bit.

Set this bit to "0" when writing.

[bit2:1] UCSEL[1:0]: USB clock selection bit

Bit	Description
00	CLKMO [Initial value]
10	PLLOUT clock (divided by USB Frequency Divider)
01,11	Setting prohibit

[bit0] UCEN0: 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.*

5.3 USB Frequency Divider Register (UPCR6)

The UPCR6 sets the frequency division ratio of PLLOUT clock.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				UBSR			
Attribute	-				R/W			
Initial value	-				0010			

Register Functions

[bit7:4] Reserved: Reserved bits

"0b0000" is read from these bits.

Set these bits to "0b0000" when writing.

[bit3:0] UBSR: Frequency division ratio setting bits of the PLLOUT clock

bit3:0	Description
0000	Divides the frequency by (UBSR+1). The division ratio of 1 to 16 can be set by using the UBSR value. (Example) USBR = "0010" => 1/3 frequency [Initial value]
0001	
•	
•	
1111	

Note:

- *This register is not initialized by software reset.*

5.4 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".*

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) at the same time.

Be sure to disable the USB clock output before selecting the USB clock.

- 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 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 5-1: USB Device (USB Function)



This chapter explains the USB Device (USB function).

1. Overview of USB Device (USB Function
2. Configuration of USB Device (USB Function
3. Operations of USB Device (USB Function
4. Examples of USB Device (USB Function) Setting Procedures
5. USB Device (USB Function) Registers

CODE: FW03F-E19.5

1. Overview of USB Device (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 Device (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:

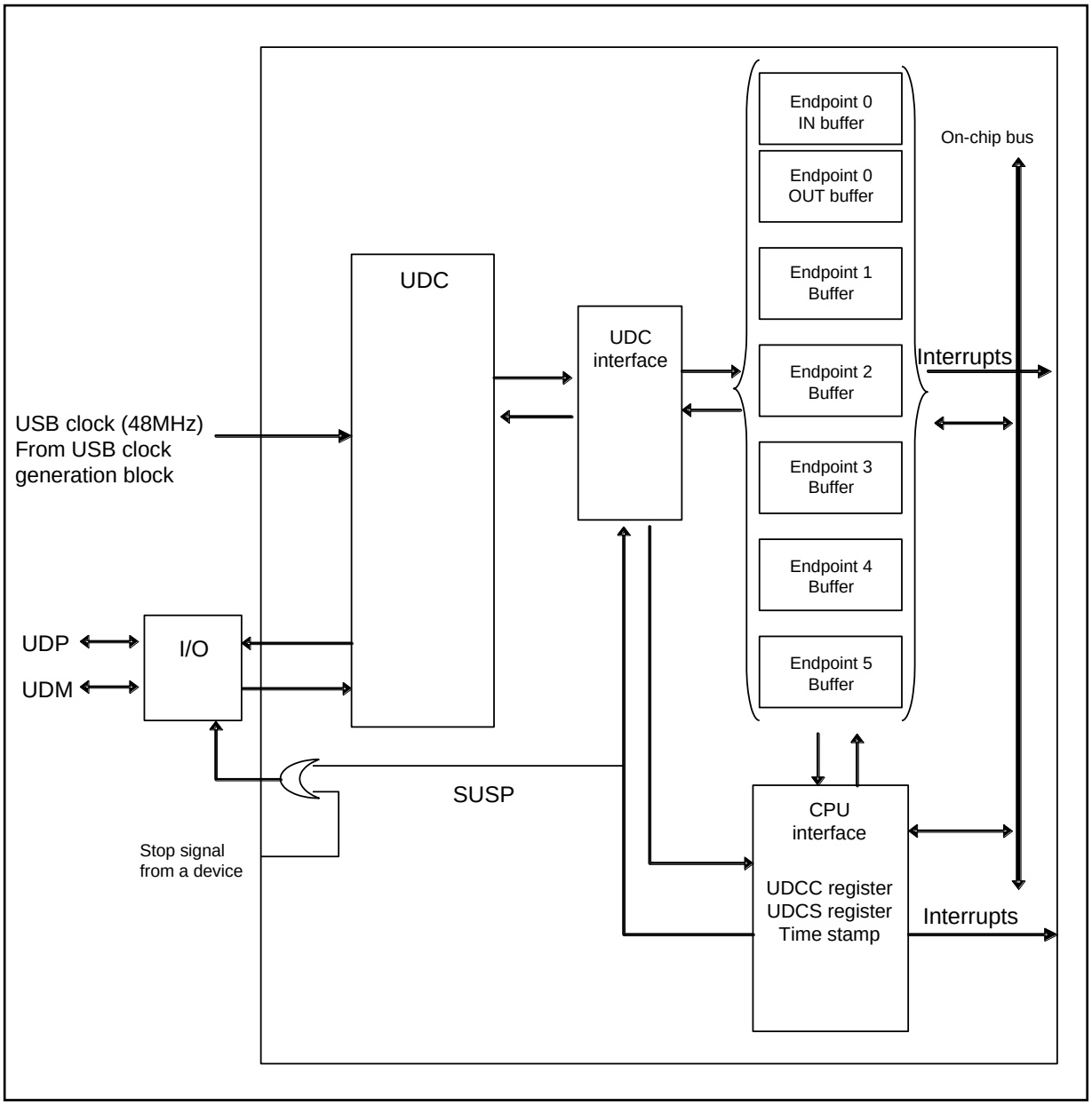
- *Set the base clock (HCLK) to 13 MHz or higher when using the USB function.*

2. Configuration of USB Device (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 Device (USB Function)

The USB function supports the USB (Universal Serial Bus) communication protocol. It’s hardware supports the basic protocol operation (handshake). Therefore, USB communication can be implemented by processing only transfer data.

3.1. USB Device (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 Device (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.

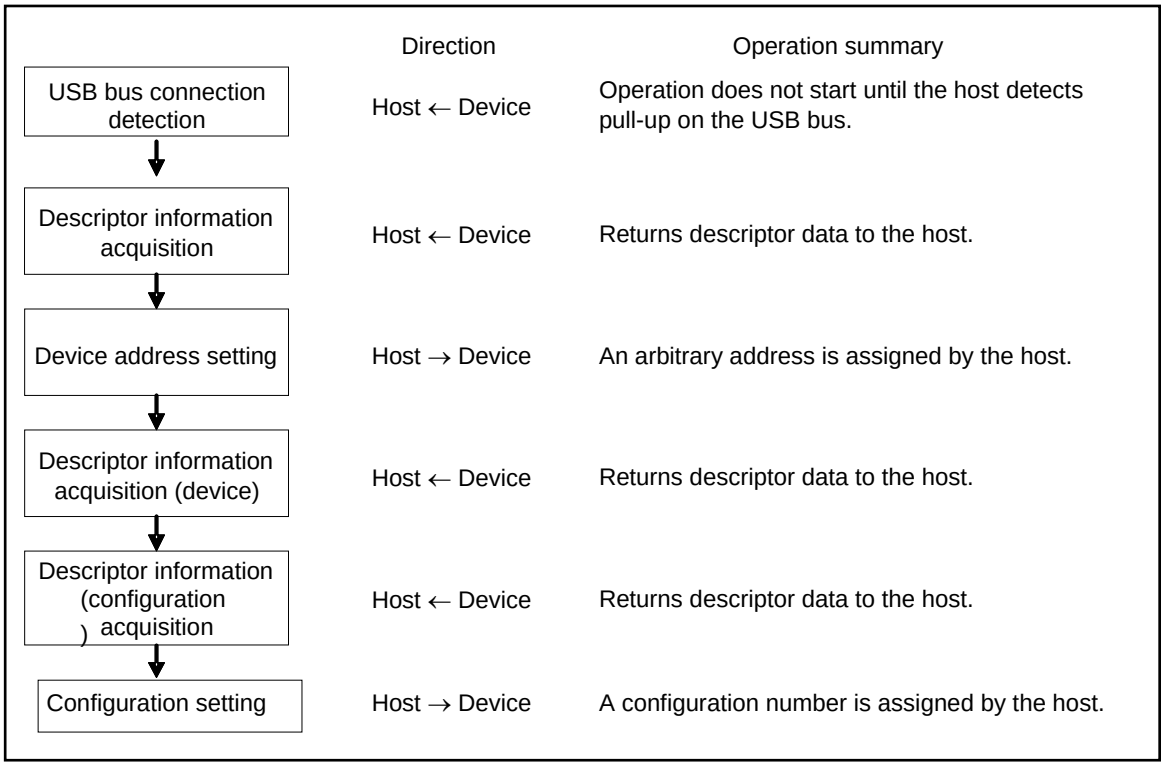
Enumeration

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 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

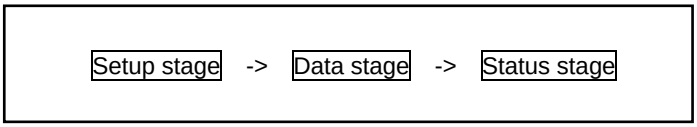
The USB device core is initialized when the host executes a bus reset on the device, but register and buffer states are not initialized.
Take the following steps to process the device. (The process is not required in the initial bus reset after USB connection.)

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.

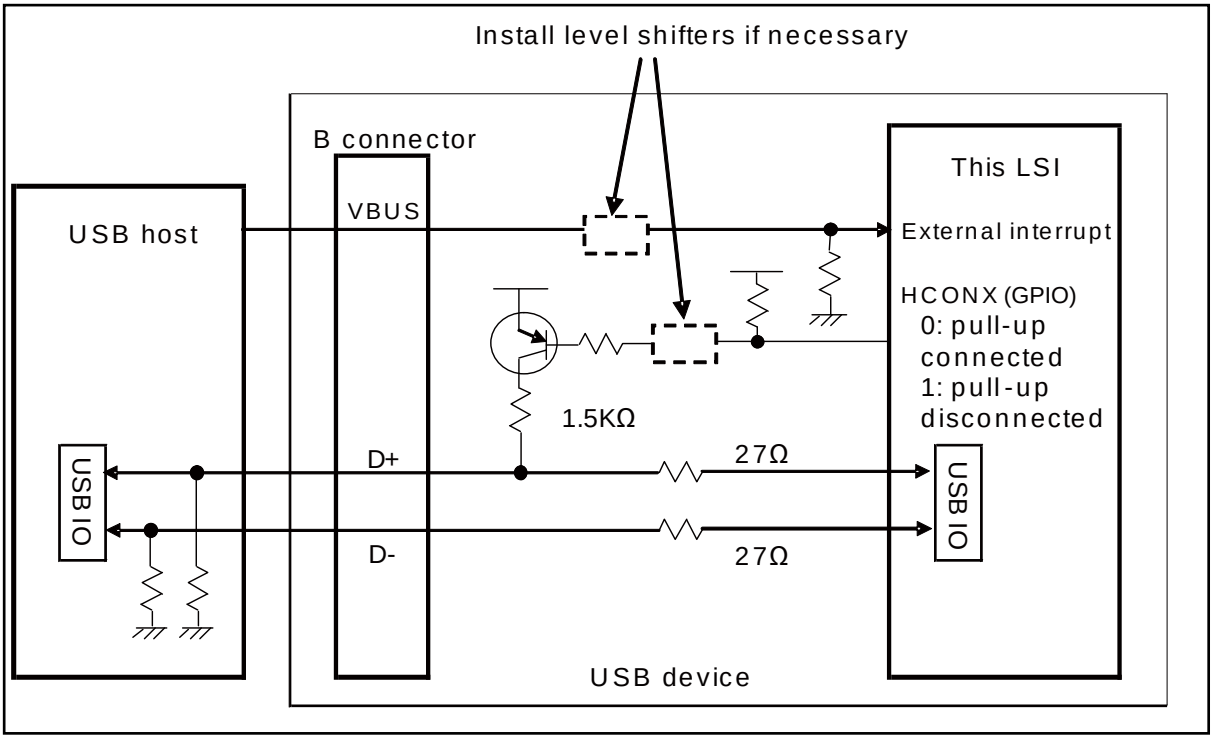
3.2 Detection of Connection and Disconnection

The following explains about detecting connection and disconnection to/from the USB host.

Example of USB System Connection

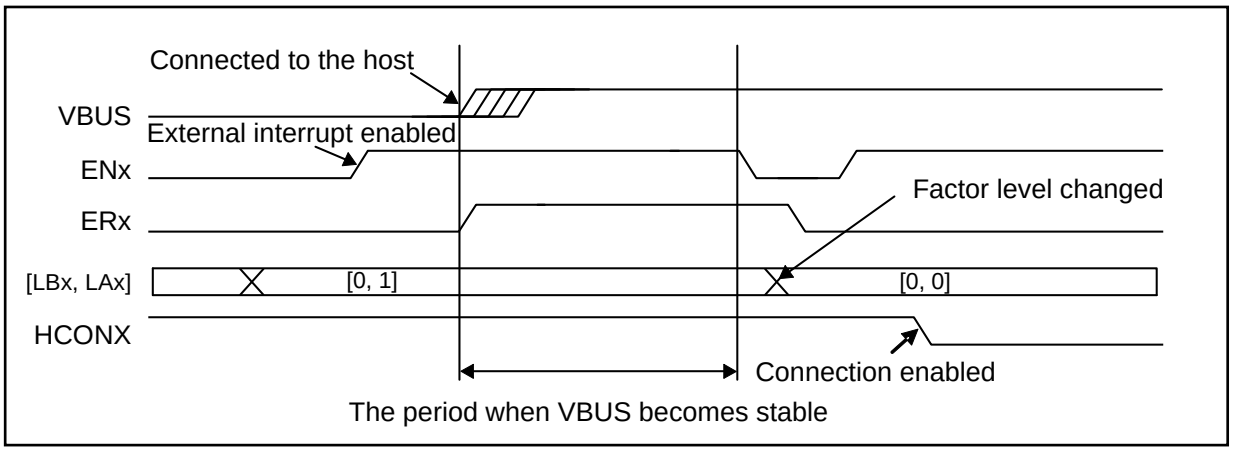
By connecting an external interrupt pin to the VBUS pin of the USB connector, and installing a pull-down resistor onto the VBUS signal, disconnection from the USB host can be detected. Figure 3-3 shows an example connection of USB connector with D+, D- and VBUS.

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.

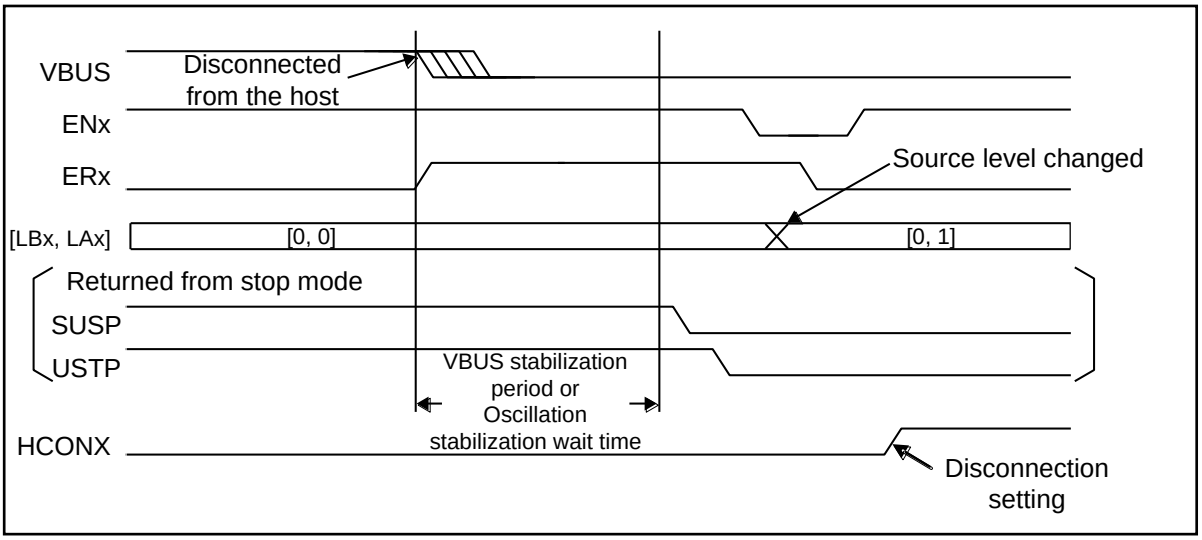
*2: Clear the HCONX bit even if the pull-up resistor is not controlled.

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.*

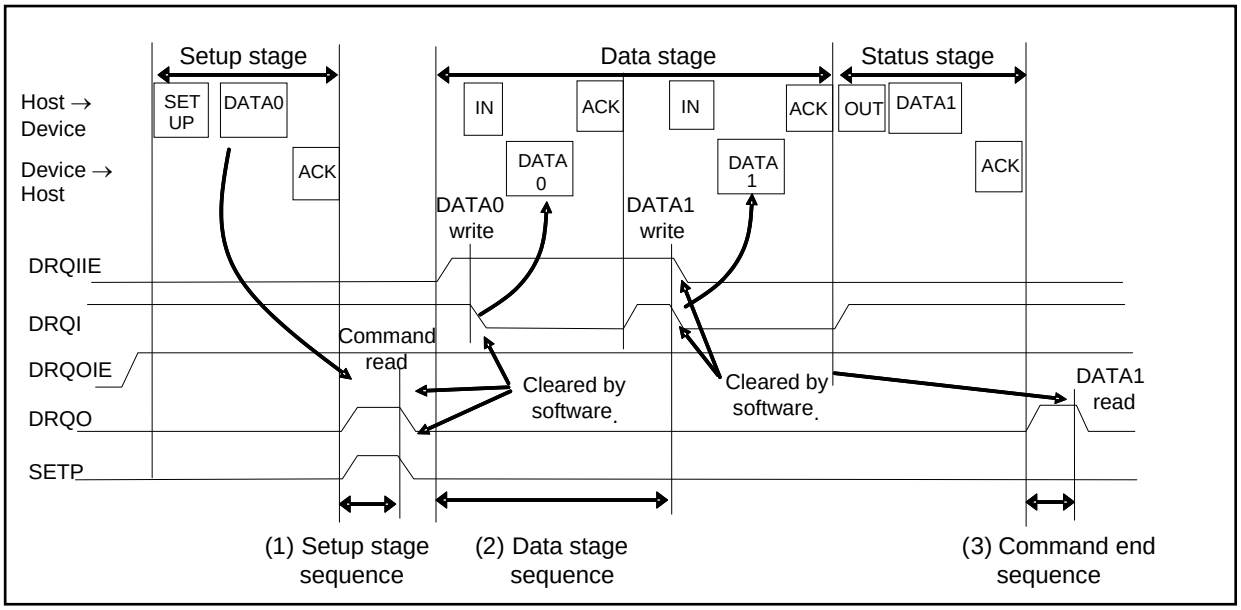
3.3 Operation of Each Register in Response to a Command

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 DRQIIE,* 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:

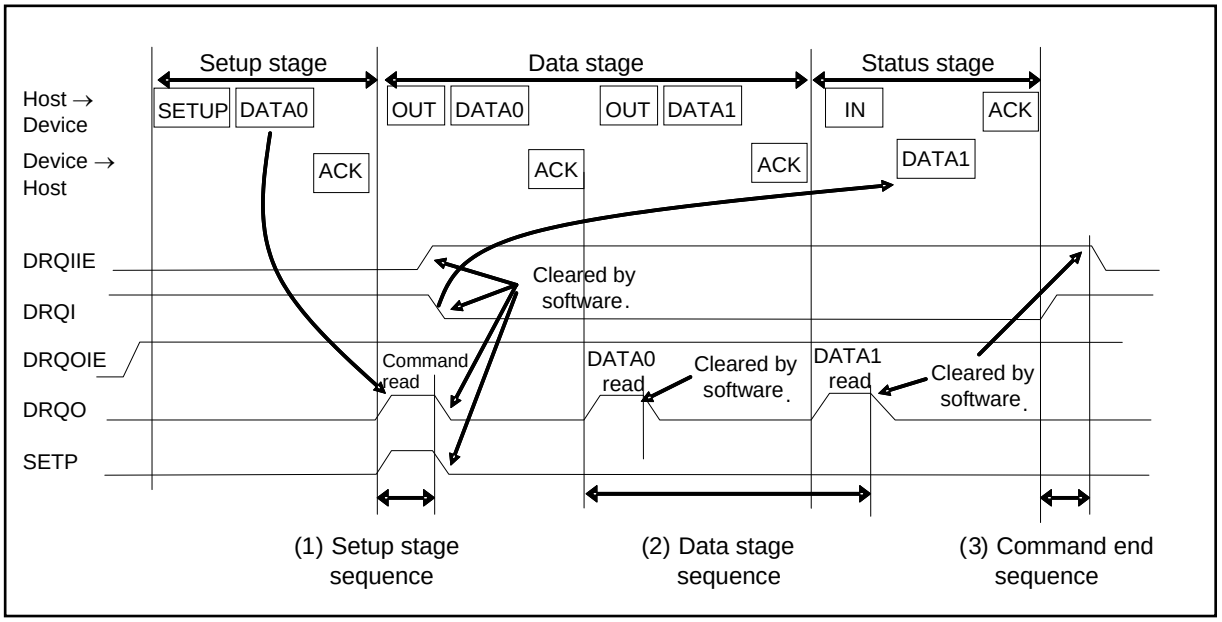
- *When next setup stage is received without (3) Command end sequence being carried out due to the process of an interrupt which has higher priority than USB, the device makes no response to the next setup stage. In order to avoid this phenomenon, carry out any of the following.*
  - *Increase the interrupt 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

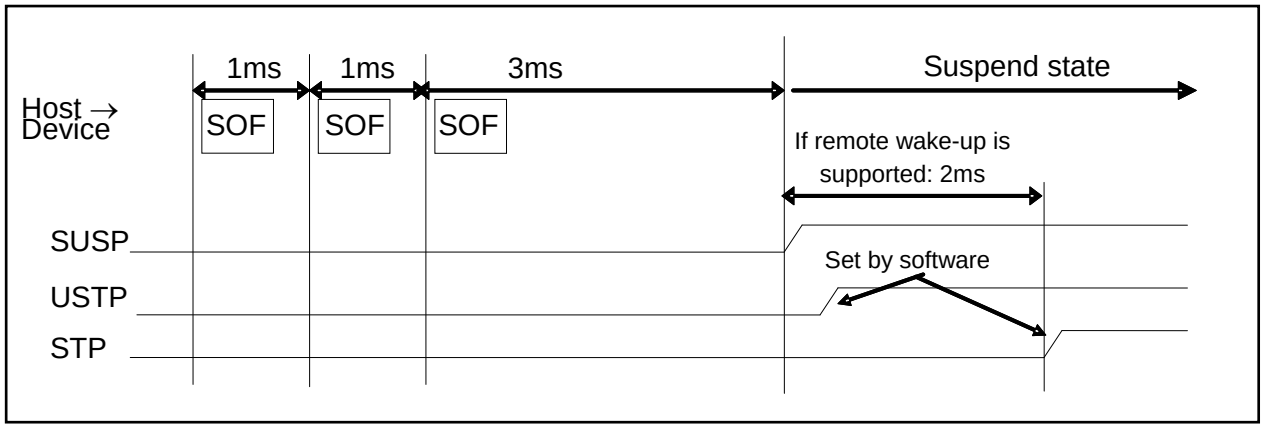
Depending on the bus power configuration, USB devices must drop the power consumption to 500 μA or less in suspend state. The following explains the sequence the USB device makes transition to suspend state, and then stop mode or timer mode.

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.*

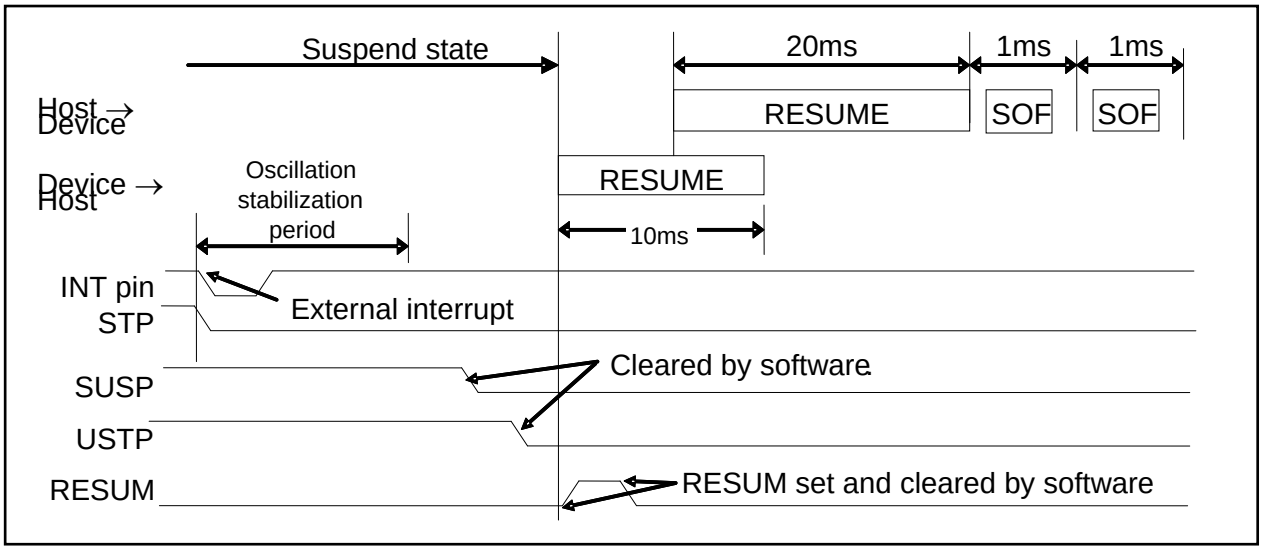
3.5 Wake-up Function

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

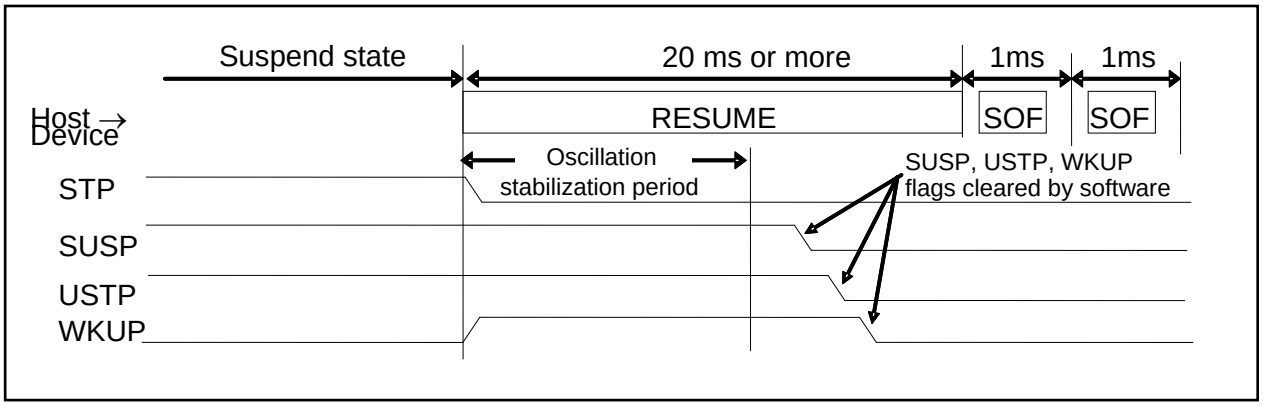


The device must be processed in the following sequence:

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.

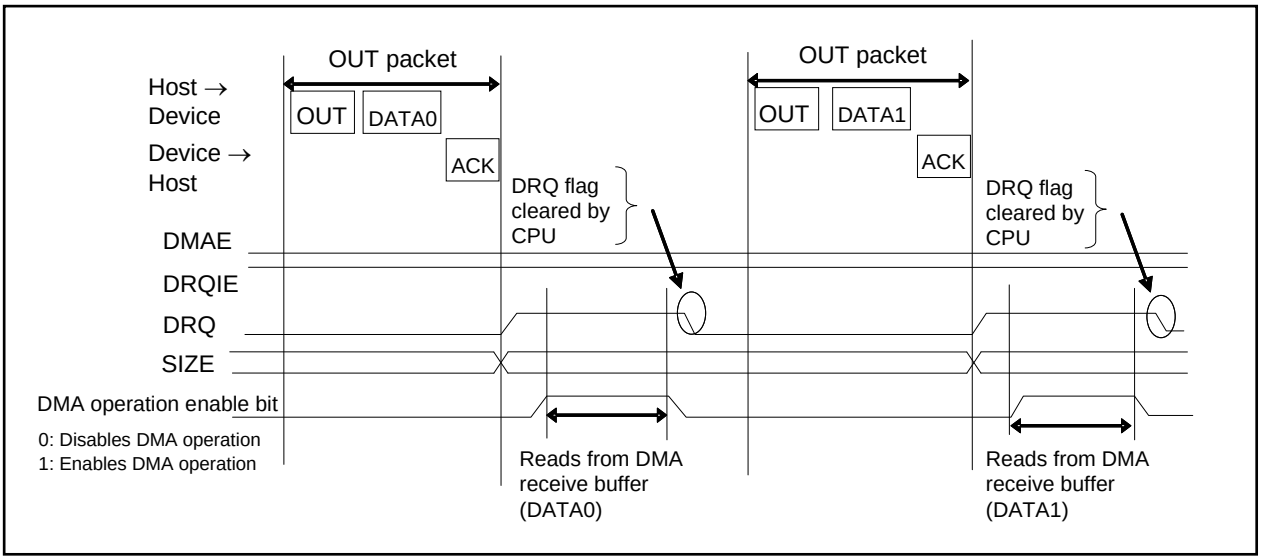
Packet Transfer Mode

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

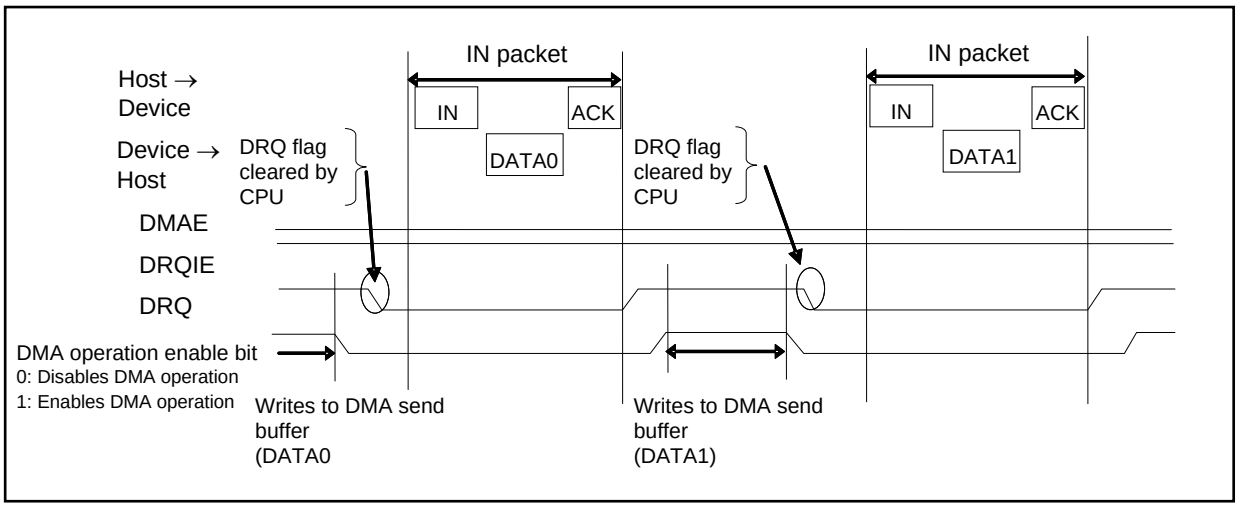


In the OUT direction transfer, the device must be processed in the following sequence:

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

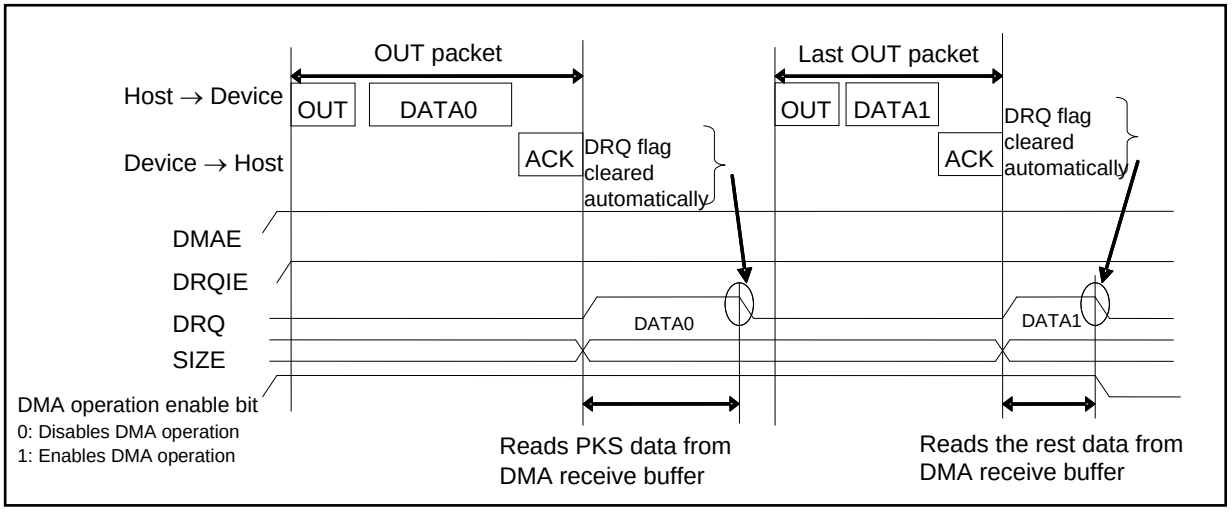
This mode can transfer even bytes. To transfer odd bytes in the OUT direction transfer, a CPU transfer sequence is required. (See Figure 3-14.) To transfer odd bytes in the IN direction transfer, see the following information.

- 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:

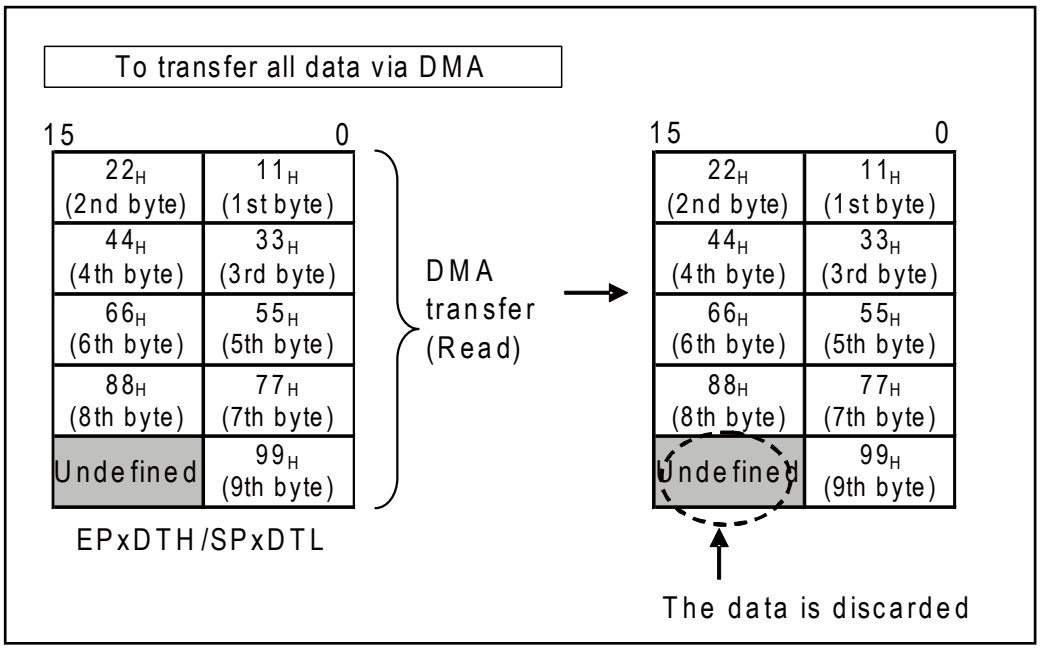
1. Configure the DMA register setting relevant to the number of transfers and block size corresponding to the total data size, and then enable DMA to start the transfer.
2. Enable DMAE and DRQIE.

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:

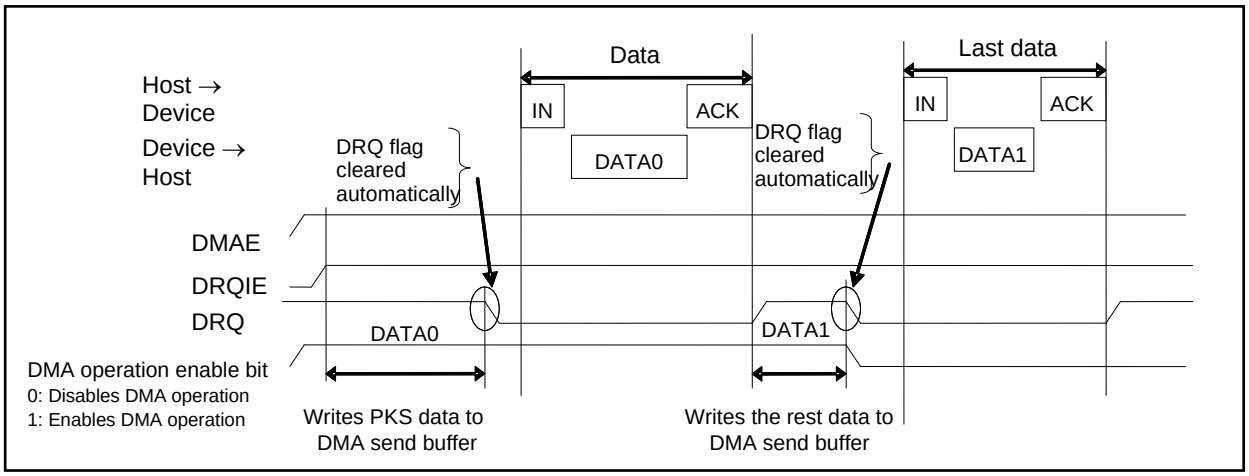
- Transfer all the data + 1 byte via DMA, and discard the last data after an endian conversion.

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.

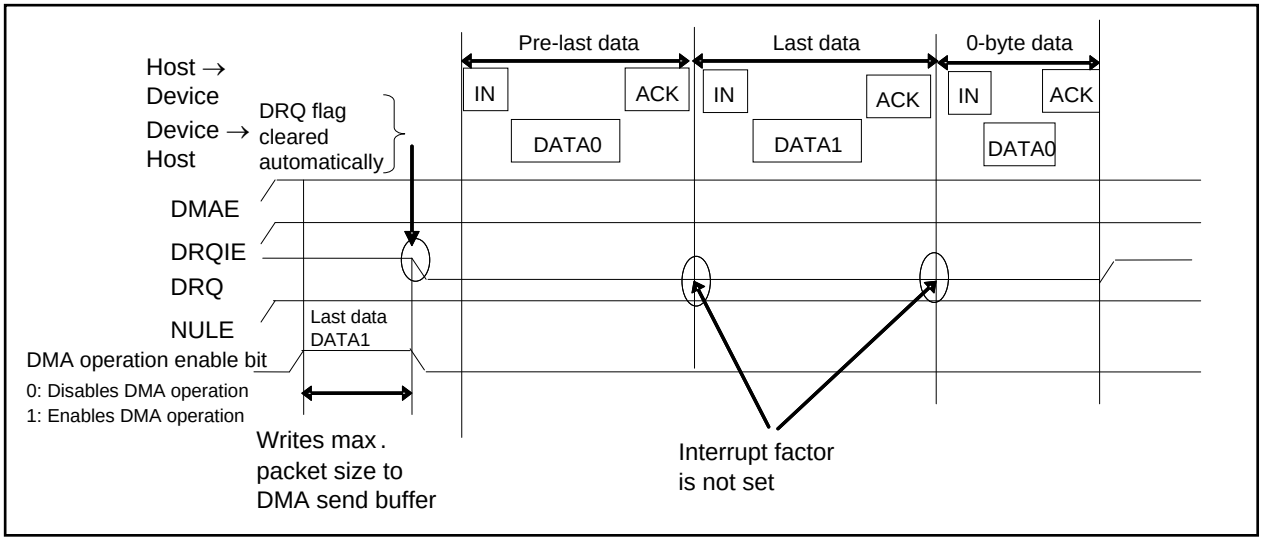
NULL transfer mode works when the following conditions are met:

- 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:

Enable EPxC:DMAE, EPxS:DRQIE, and EPxC:NULE bits by setting to 1.

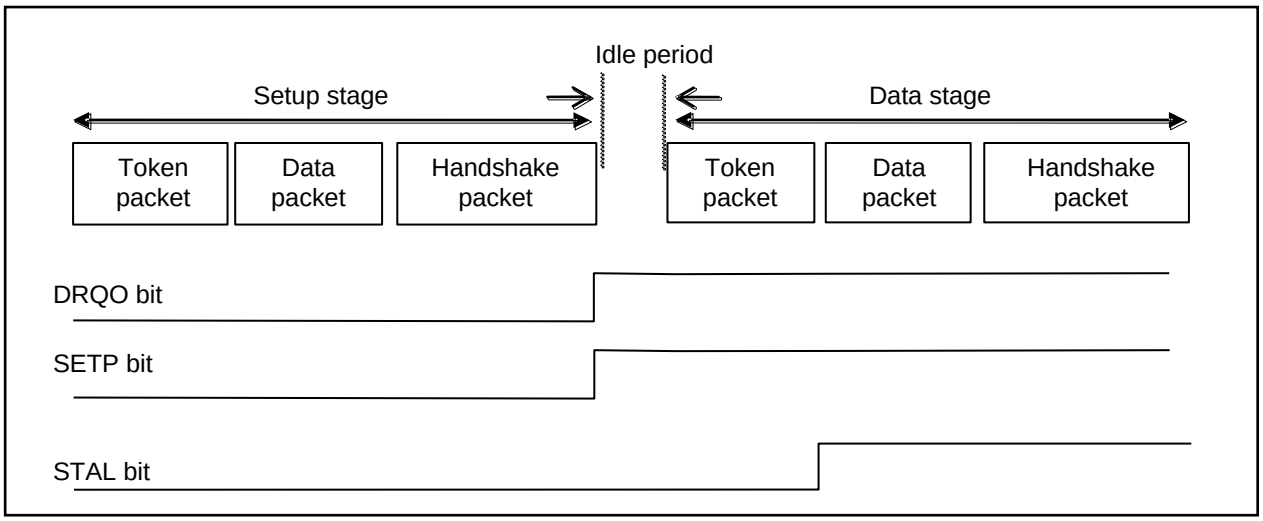
3.8 STALL Response/release of Endpoint 0

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).

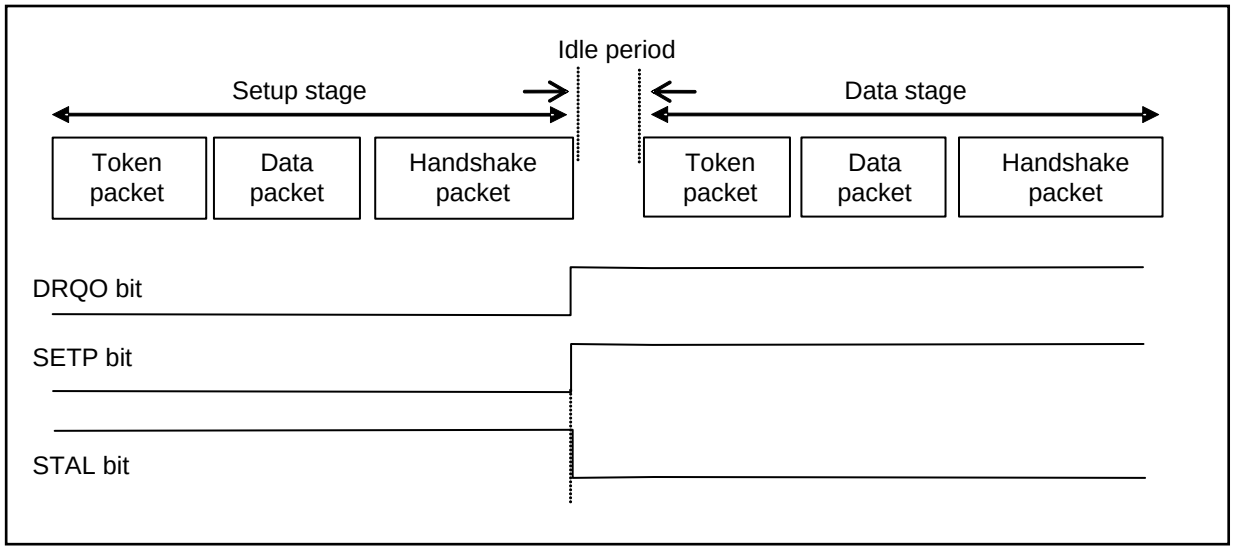
Figure 3-17 STAL Bit Set Timing



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

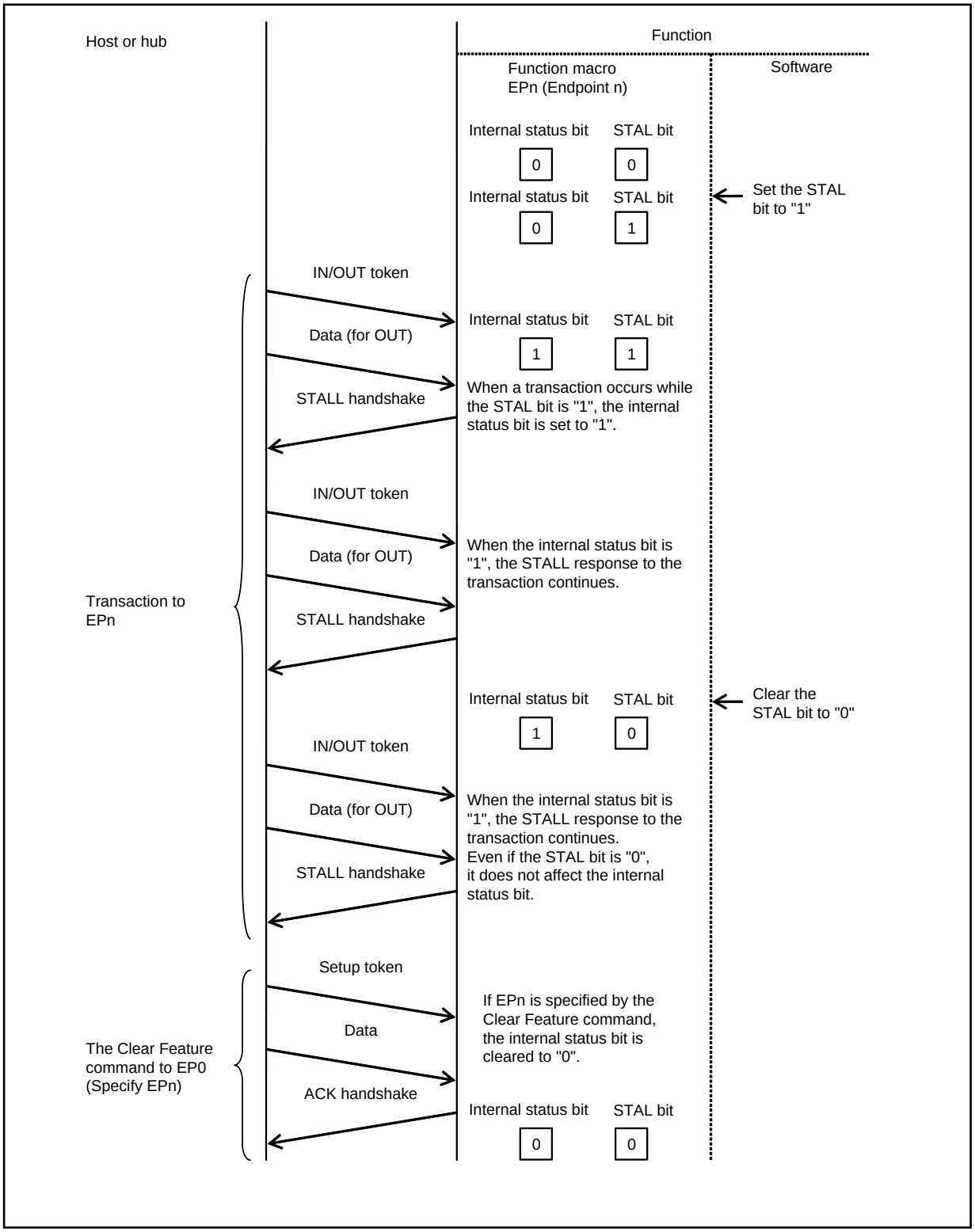
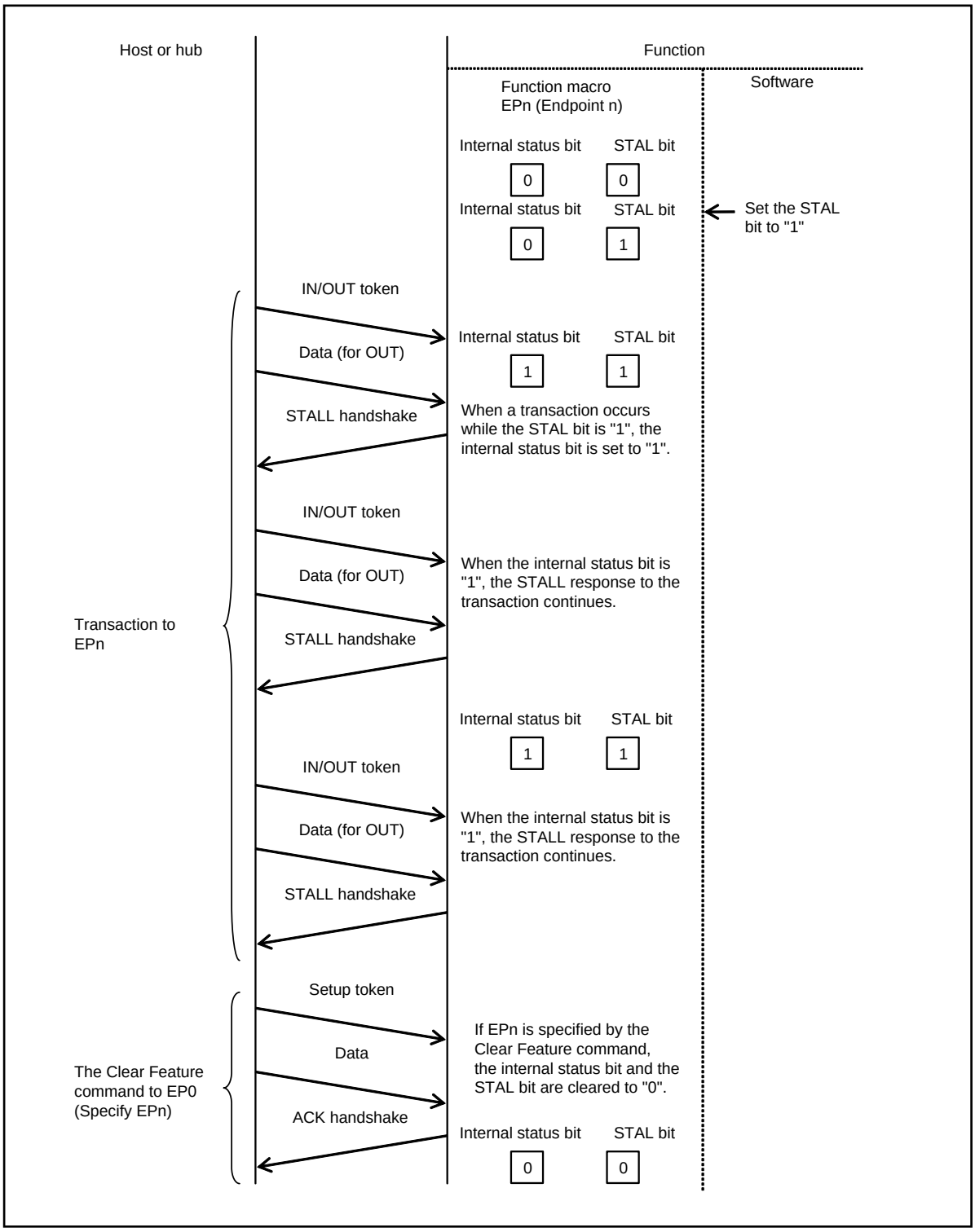


Figure 3-20 To Process the STALL Response by Software (the STAL Bit is Cleared by Hardware) UDCC.STALCLREN=1



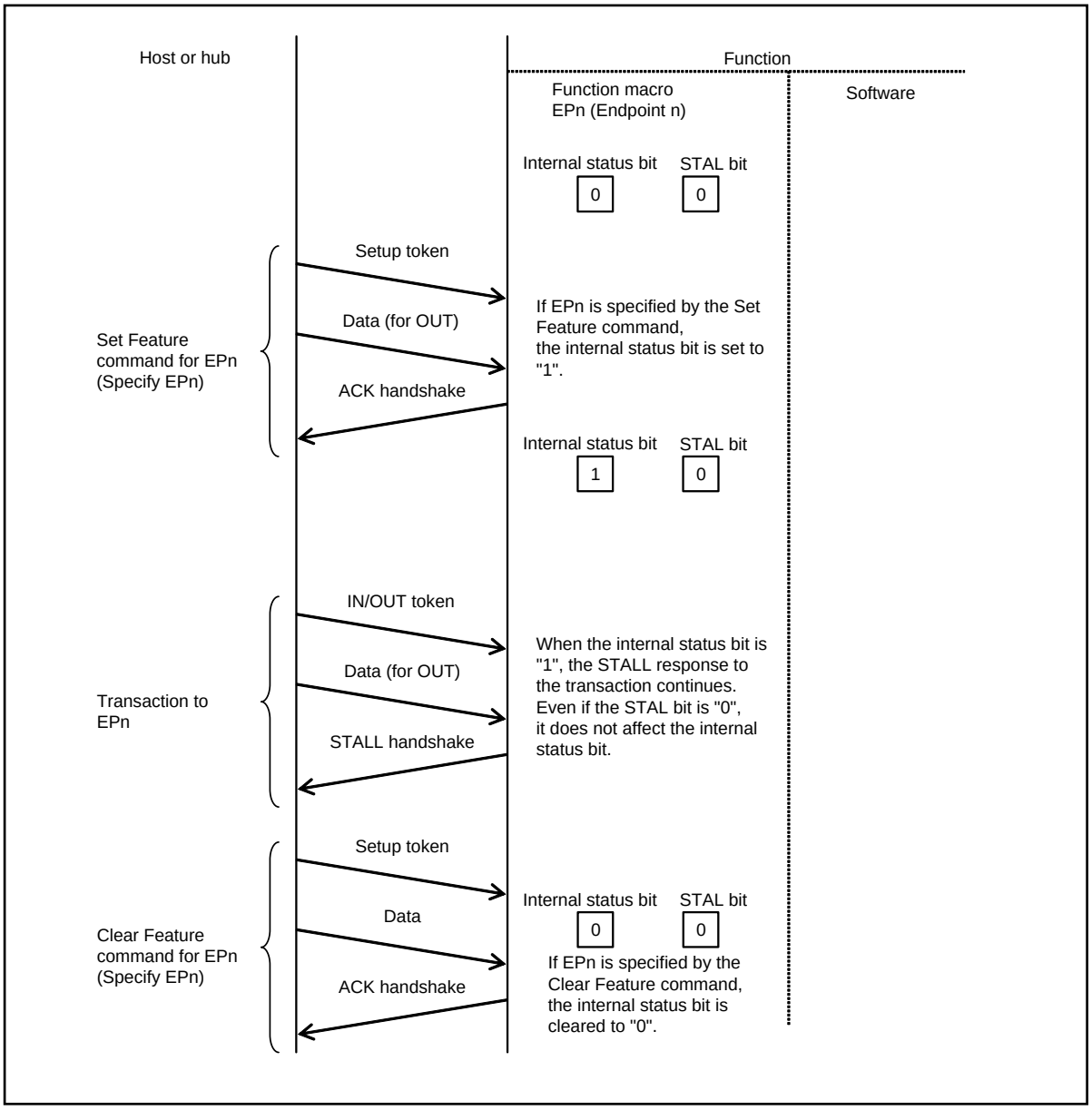
Automatic STALL Response by Hardware

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.

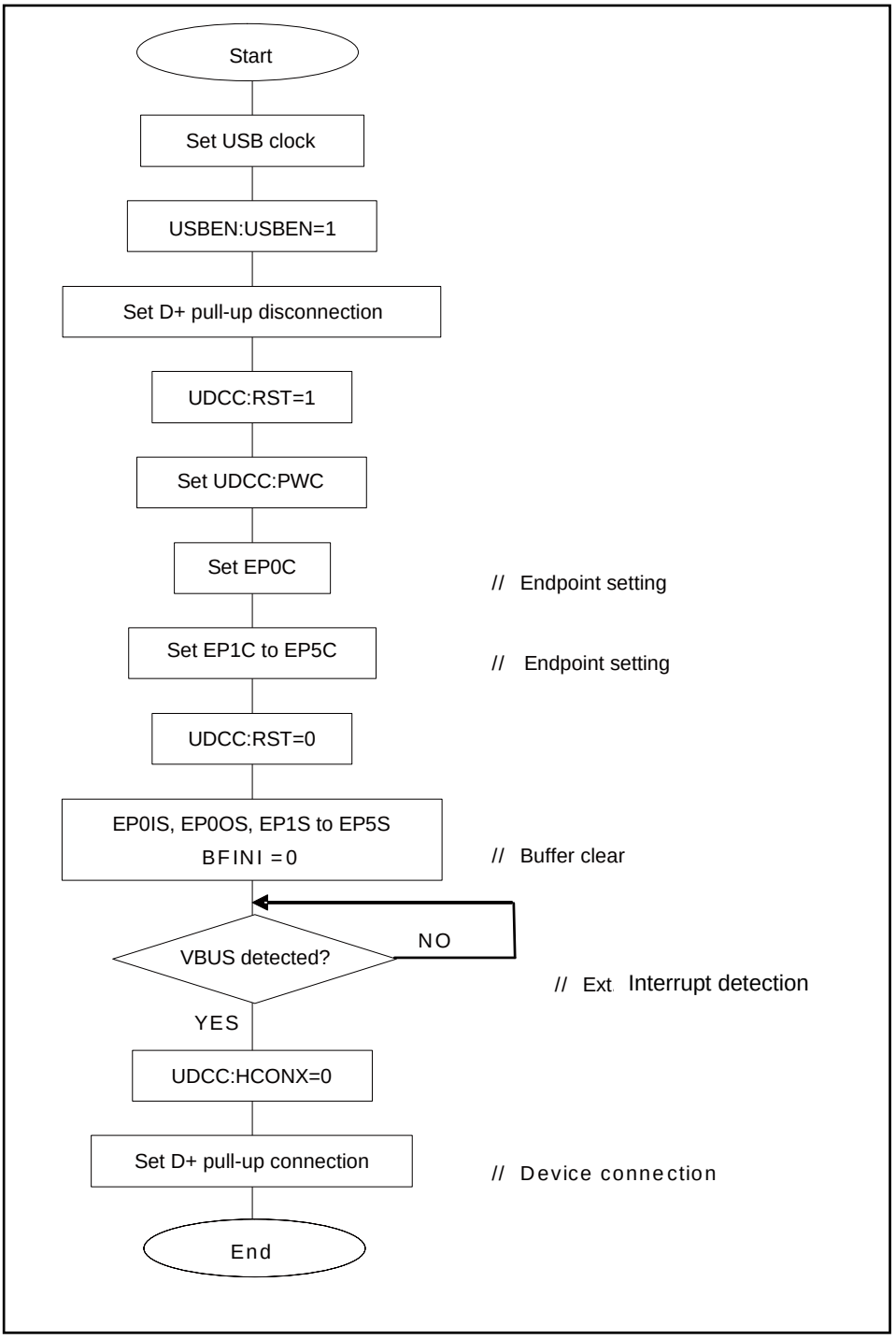
Figure 3-21 Automatic STALL Response by Hardware



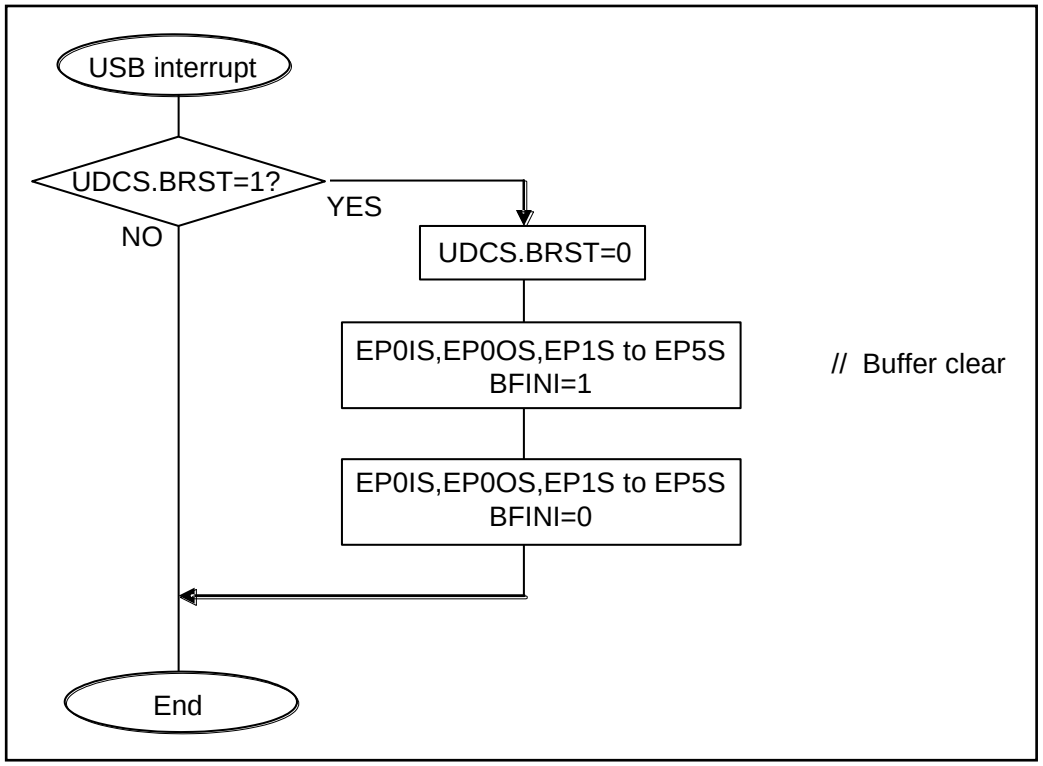
4. Examples of USB Device (USB Function) Setting Procedures

This section provides flowcharts for initialization, bus reset, CPU transfer, packet transfer (IN/OUT) and automatic data size transfer (IN/OUT).

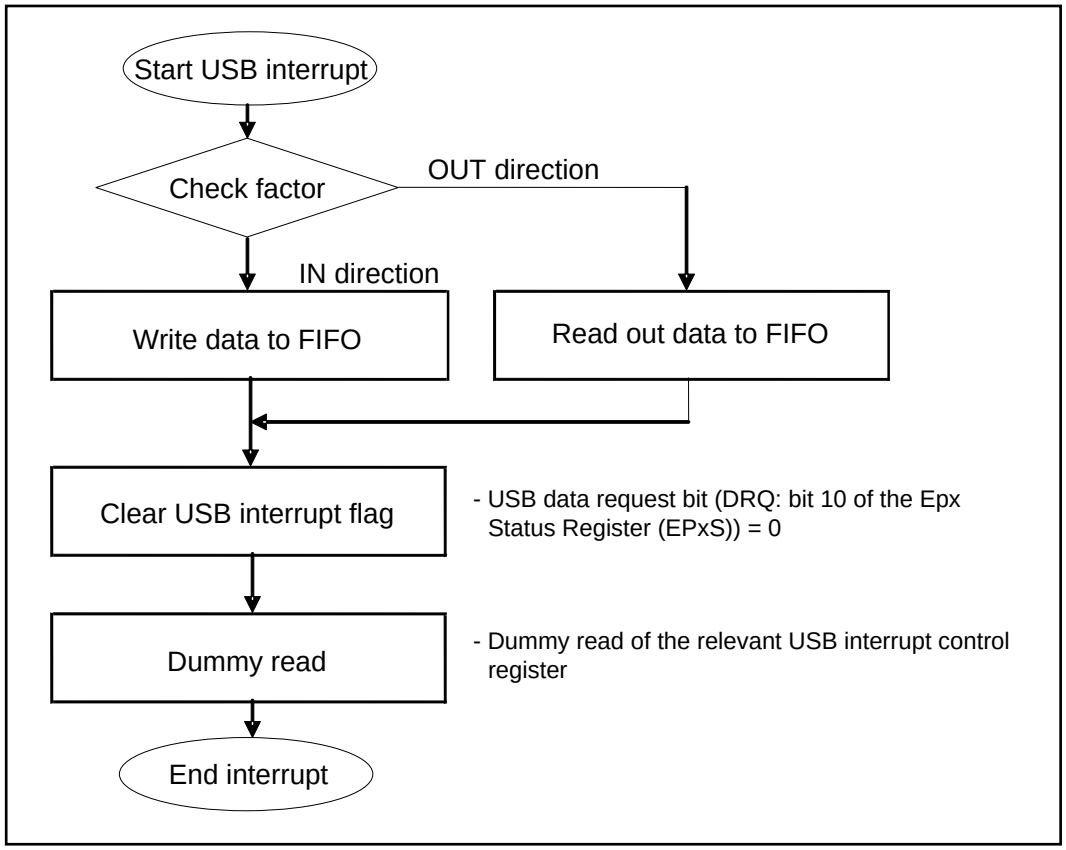
Initialization



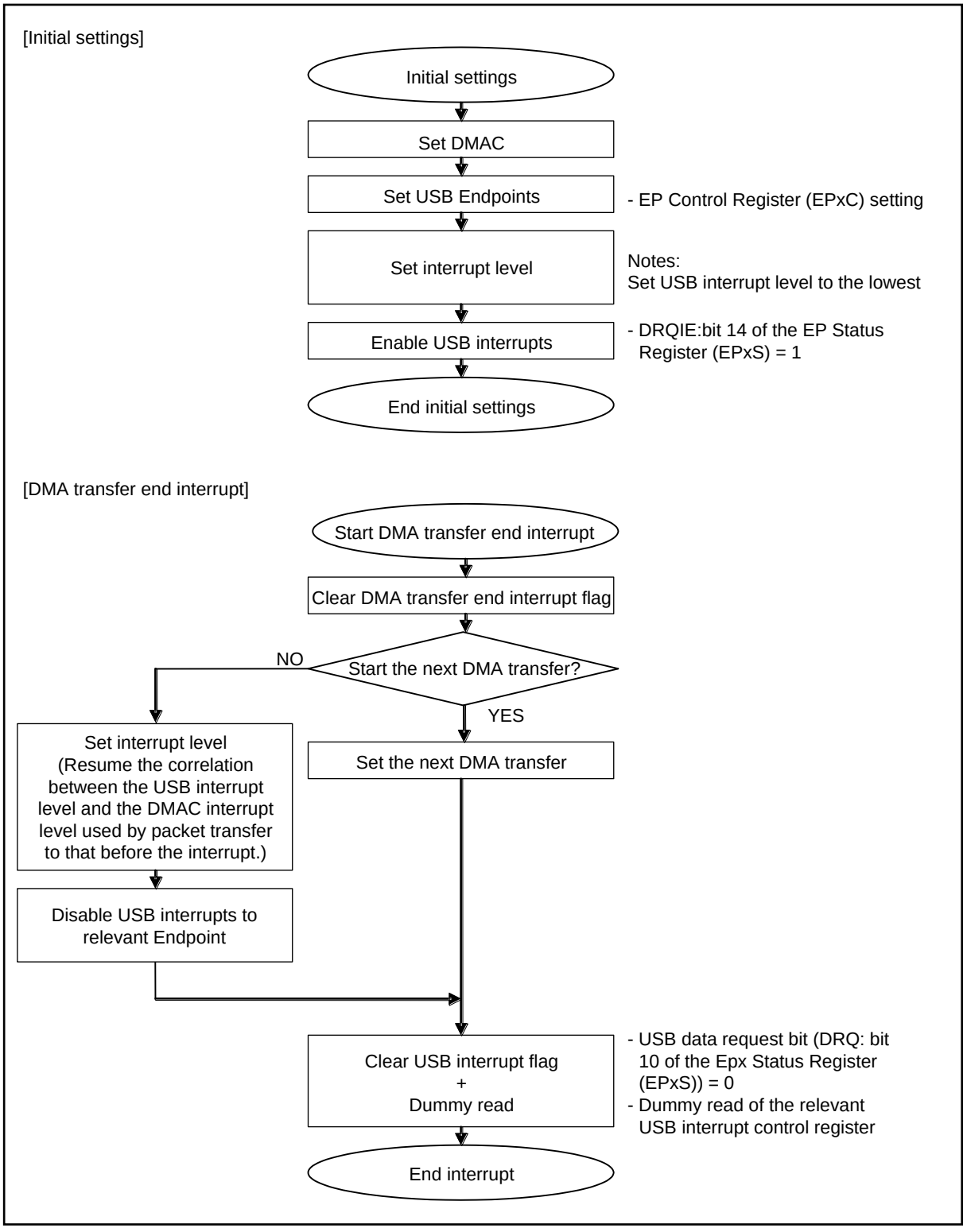
Bus reset


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

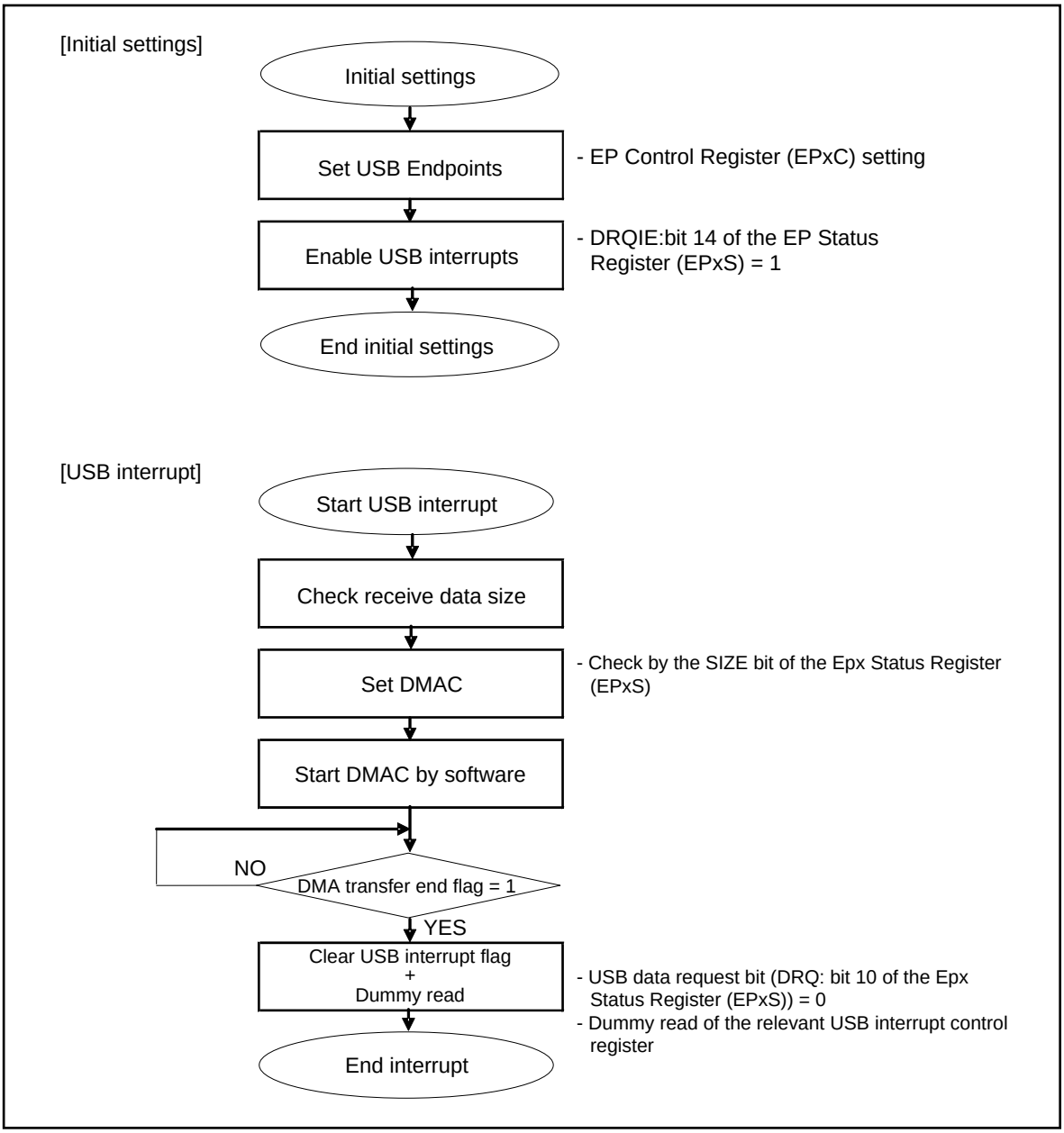
Example Control for CPU Transfer



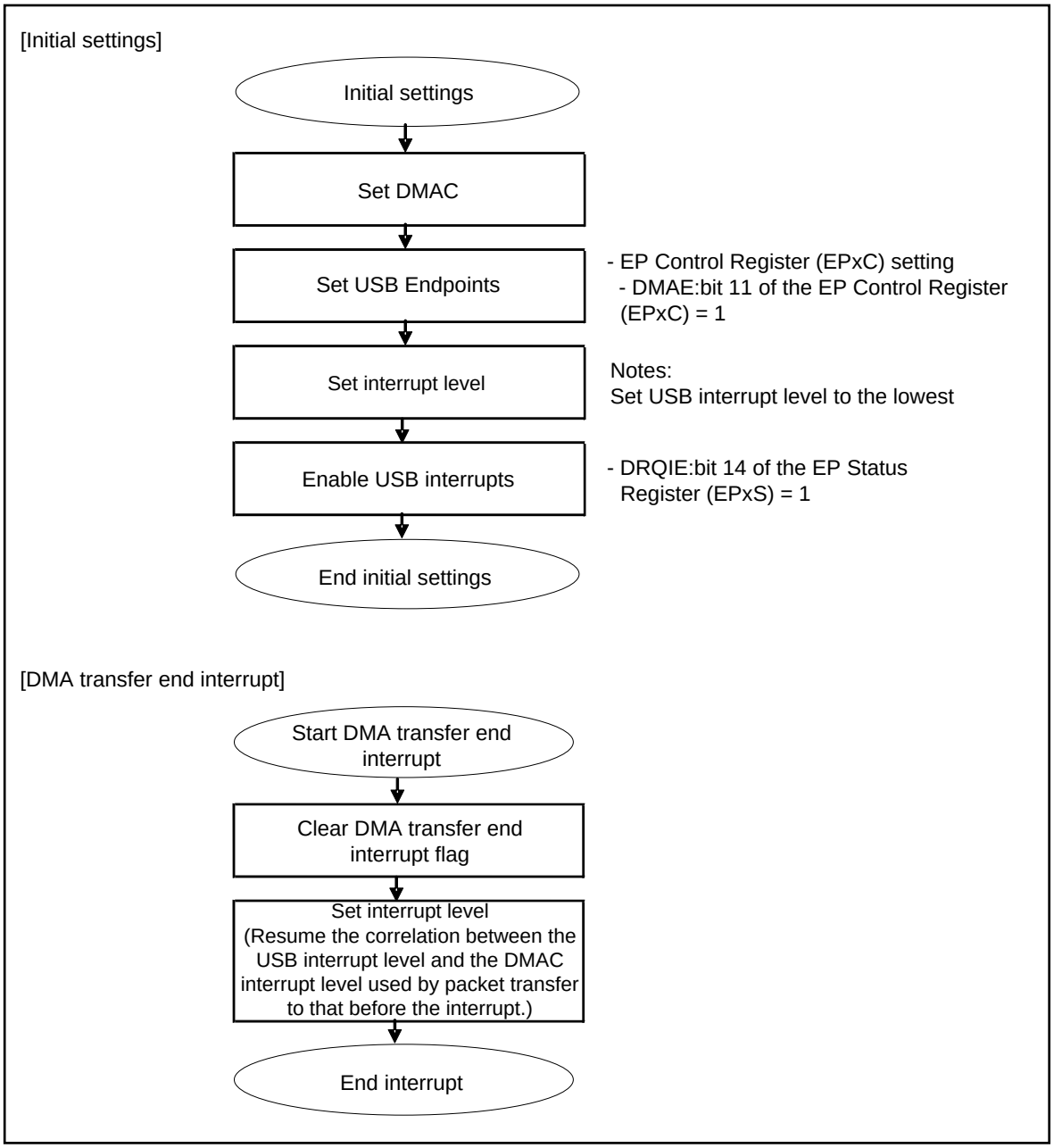
Example Control for Packet Transfer in IN Direction



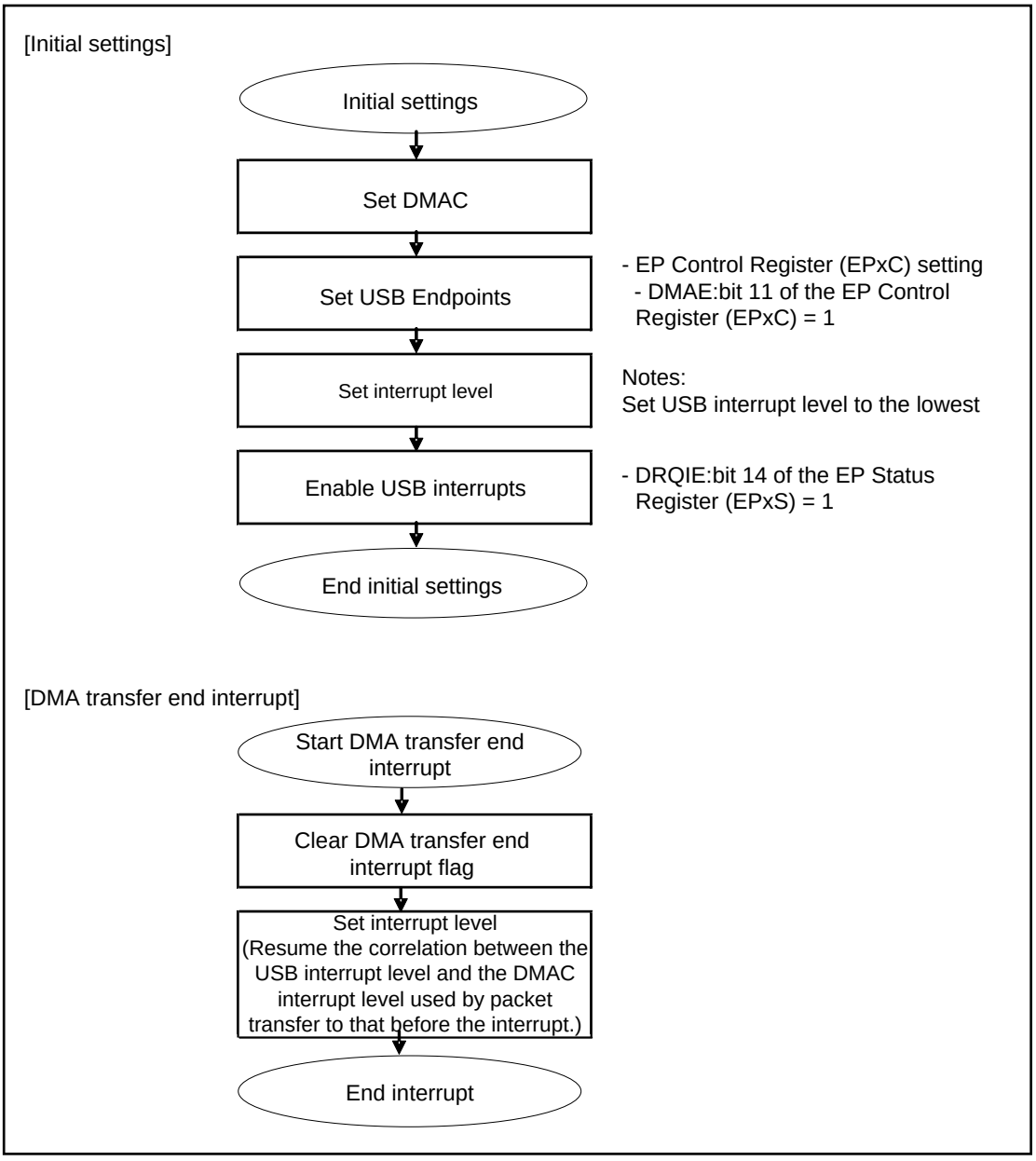
Example Control for Packet Transfer in OUT Direction



Example Control for Automatic Data Size Transfer in IN Direction



Example Control for Automatic Data Size Transfer in OUT Direction



5. USB Device (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 ClearR 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 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 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 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 Frame)

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

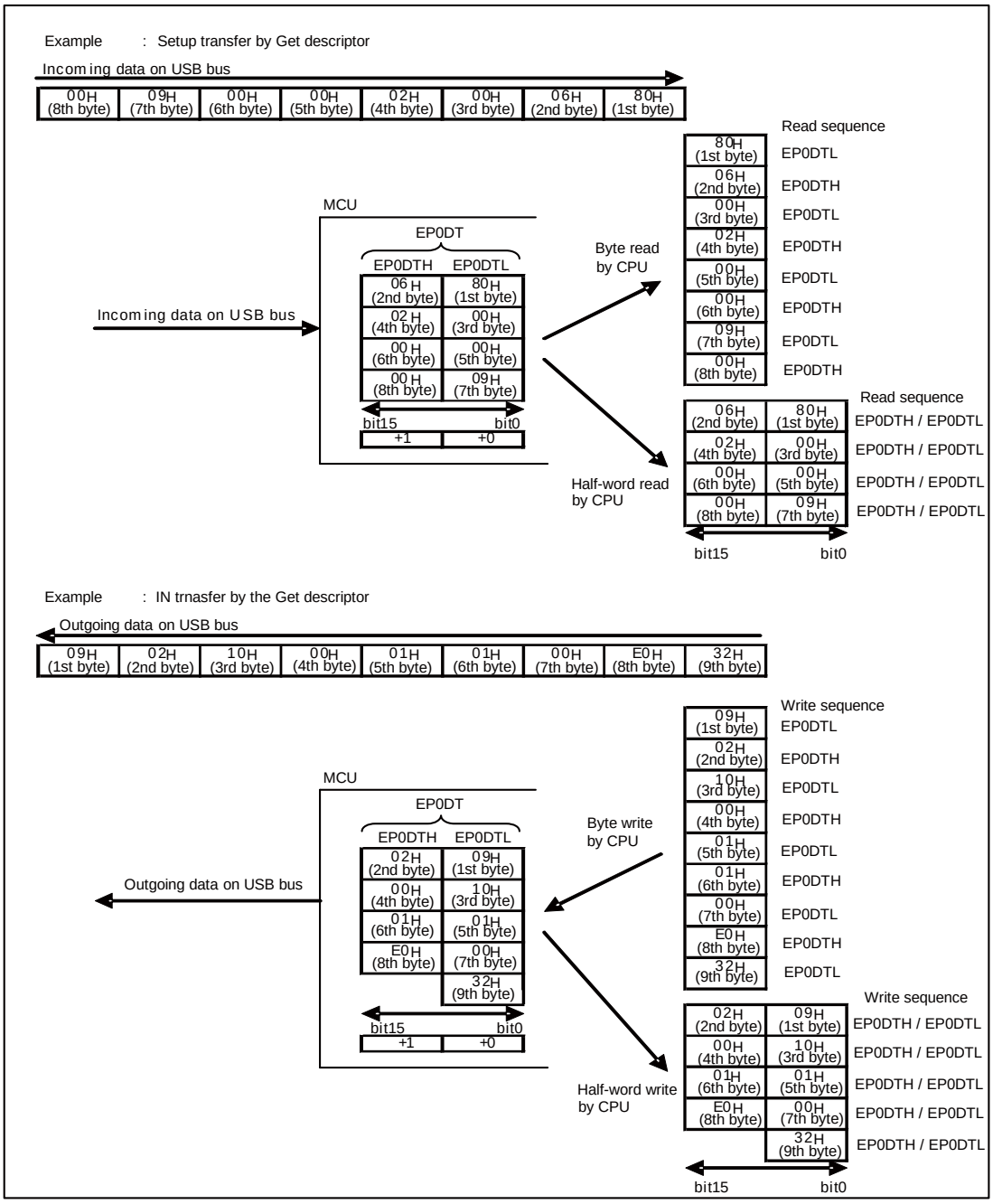
The following explains the function of each bit in the EP0 to EP5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

[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 Automatic Data Size Transfer Mode of 3.6 DMA Transfer Function.)

CHAPTER 5-2: 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.

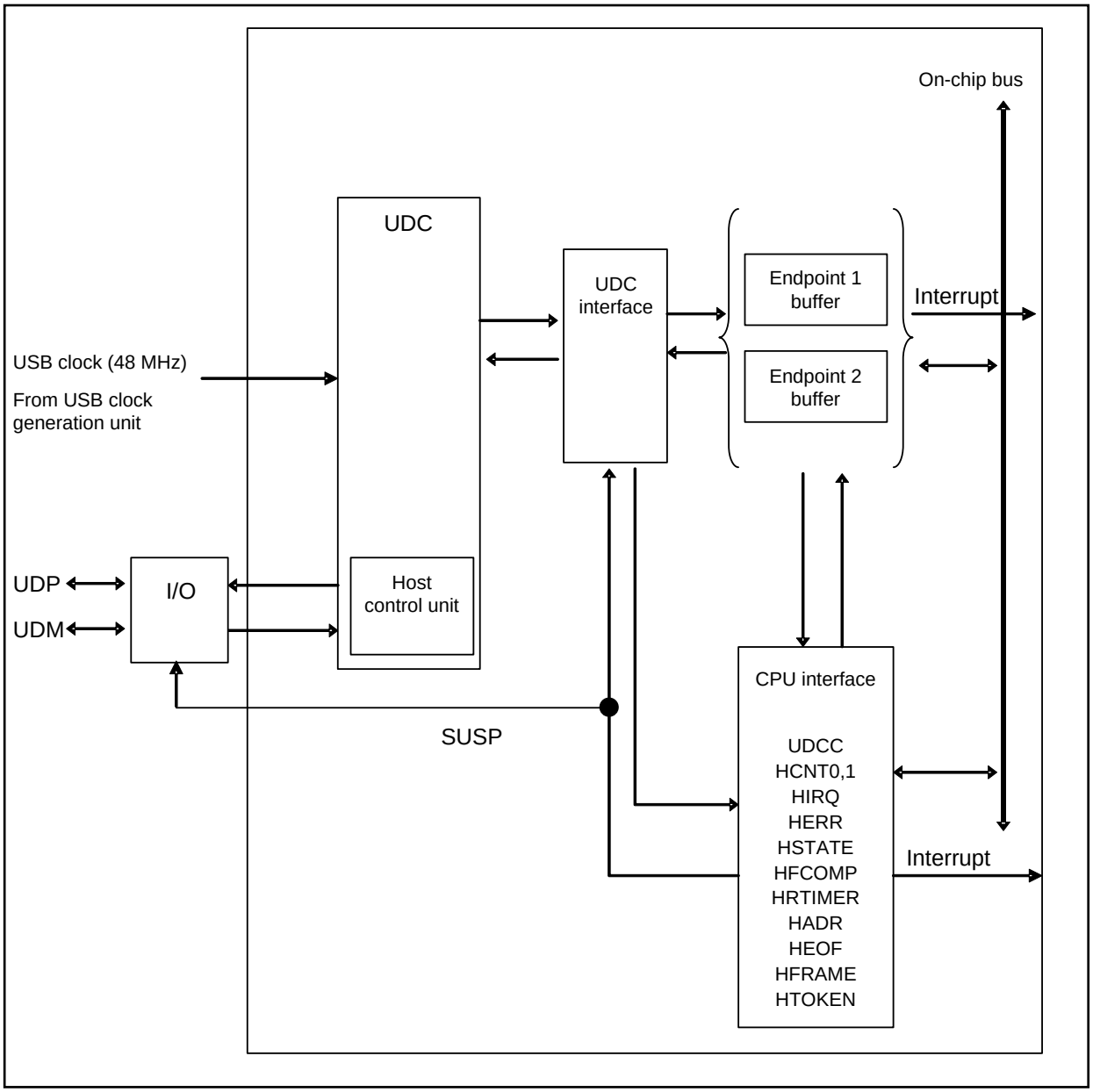
*1: Supports a hub of up to one stage in only the full-speed mode.

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.

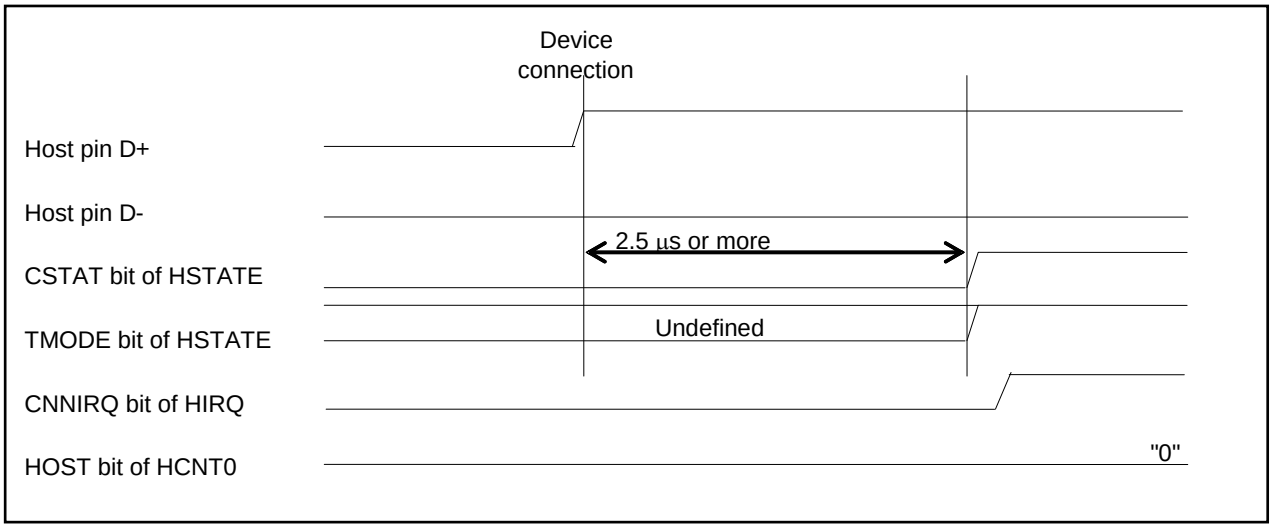
Obtaining the transfer speed of the remote USB device and selecting clocks

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

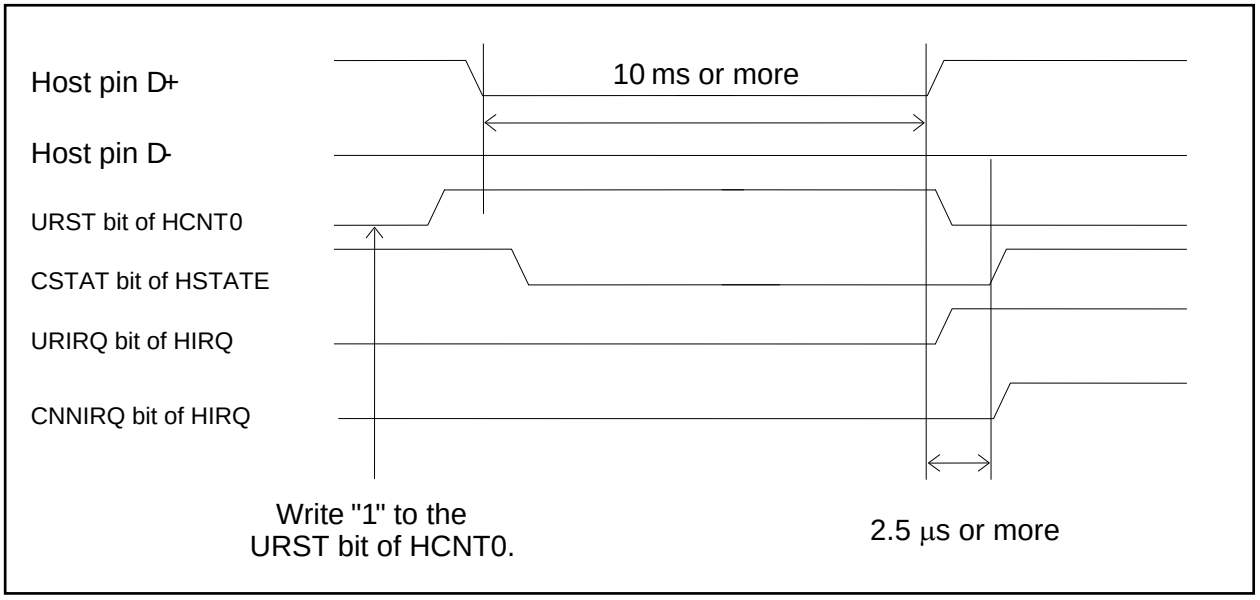
The USB bus is reset by sending SE0 for 10 ms or more if the URST bit of Host Control Register 0 (HCNT0) is set to 1 in the host mode. After USB bus resetting has been completed, the URST bit of Host Control Register 0 (HCNT0) is set to 0, and the URIRQ bit of the Host Interrupt Register (HIRQ) is set 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.

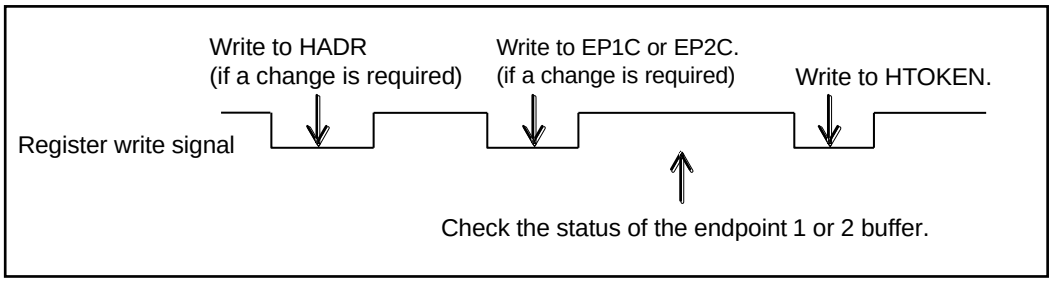
If the DIR bit of the EP1 Control Register (EP1C) is 1, the endpoint 1 buffer is used as an OUT buffer. The endpoint 2 buffer is used as an IN buffer. Then set 0 to the DIR bit of the EP2 Control Register (EP2C).
If the DIR bit of the EP1 Control Register (EP1C) is 0, the endpoint 1 buffer is used as 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").

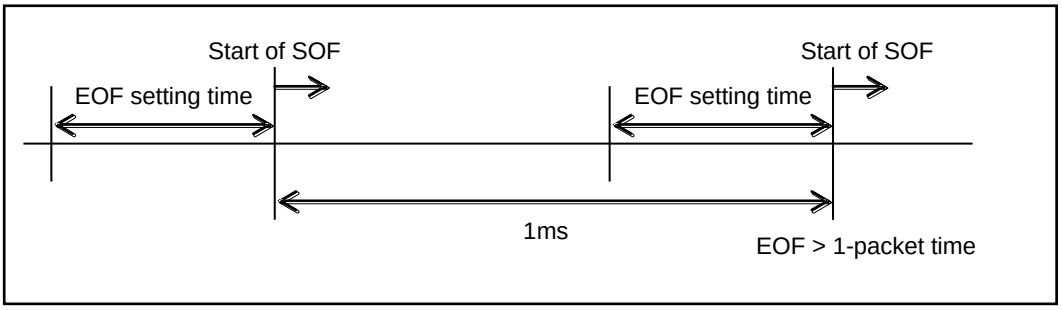
Take the following steps to change the USB from the host mode to the function mode.

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

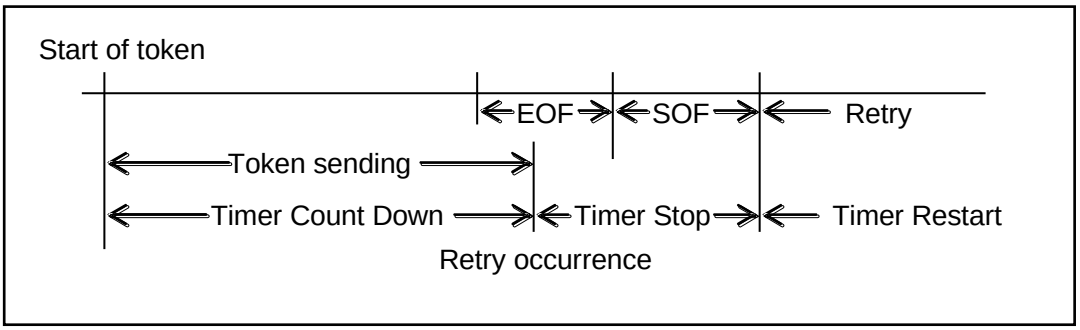
When an error* other than STALL or device disconnection occurs, the target token is retried if the RETRY bit of Host Control Register 1 (HCNT1) is 1. The following shows the conditions to end retry processing.

*: 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.

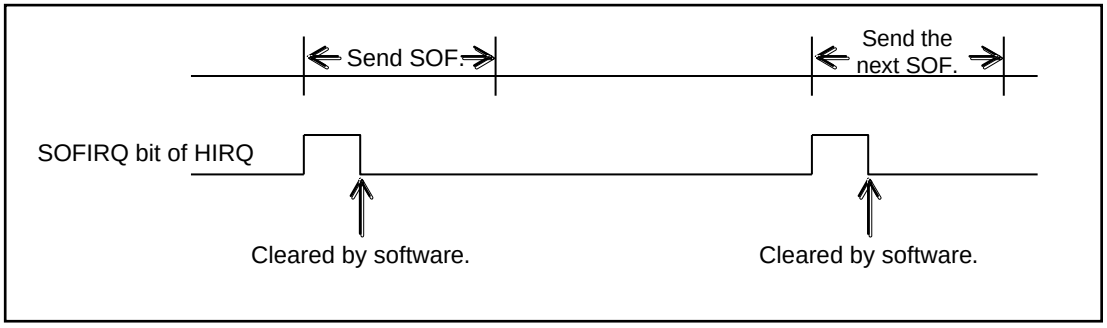
3.7 SOF Interrupt

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.

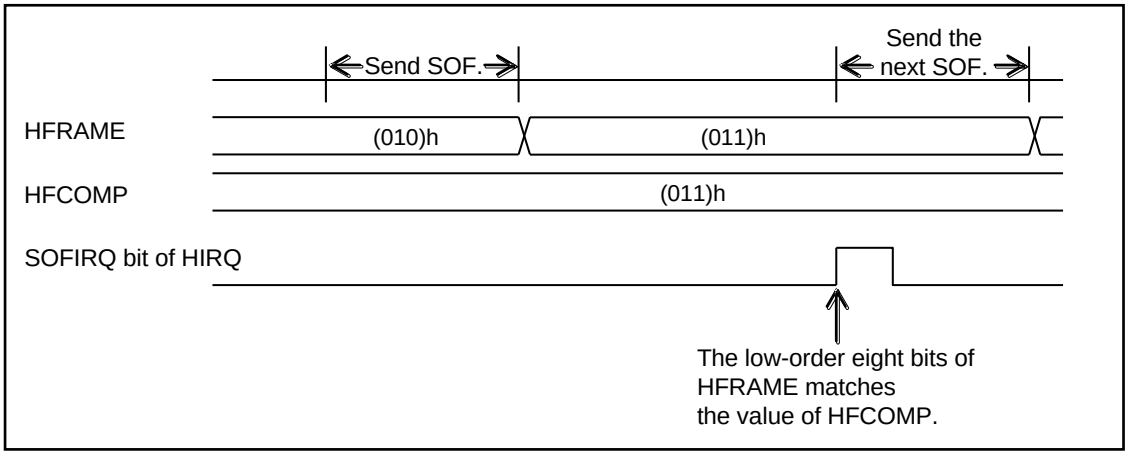
SOF Interrupt

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

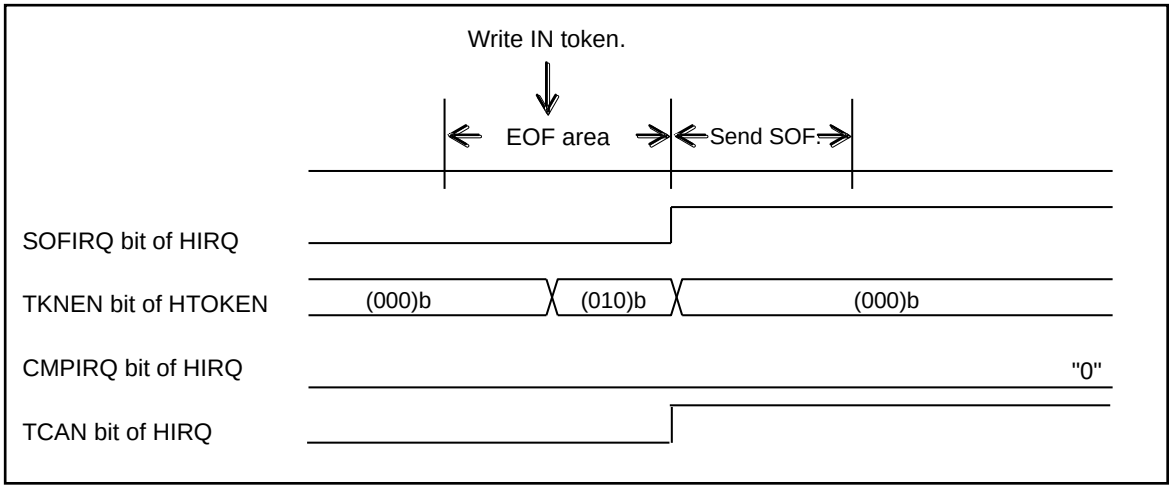
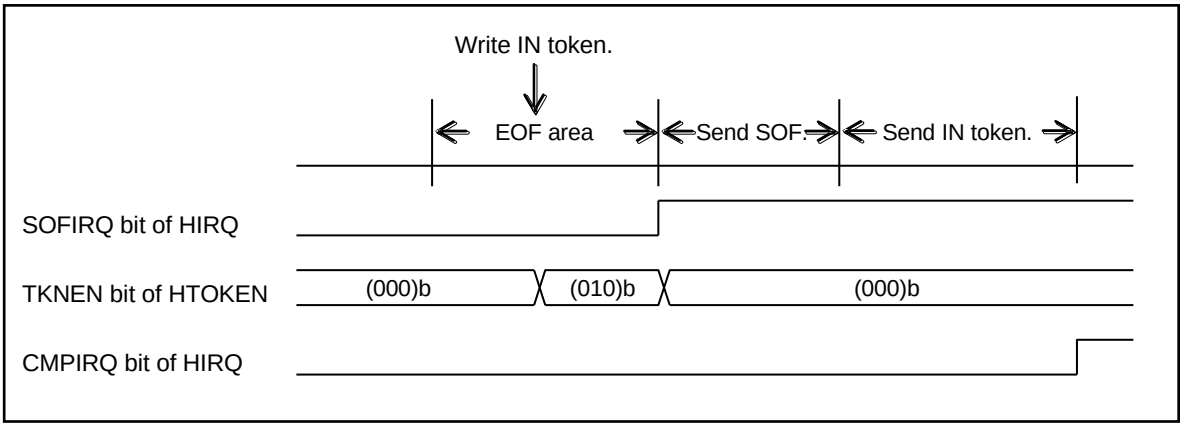


Figure 3-7 Token Operation Example at CANCEL Bit of HCNT1 = 0



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

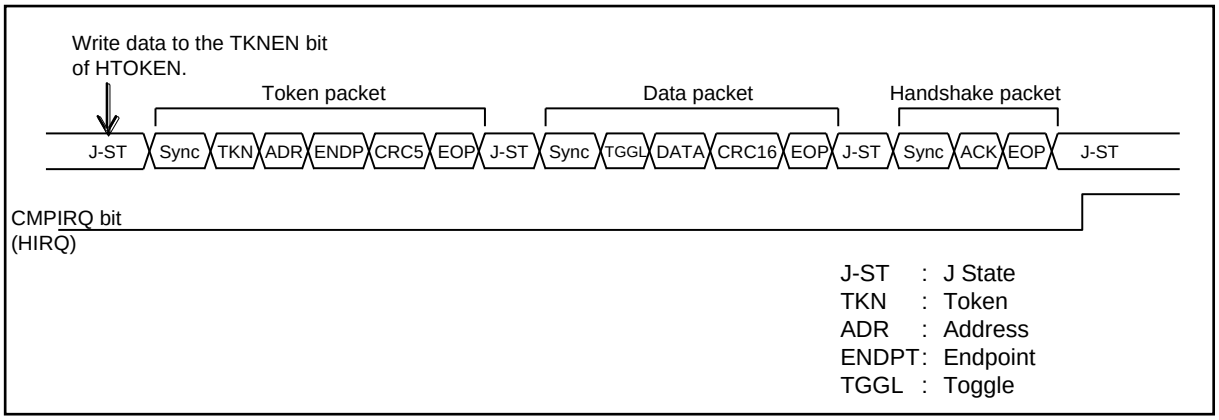
If one packet is ended in the USB host, the CMPIRQ bit of the Host Interrupt Register (HIRQ) is set to 1. At this time, if the CMPIRE bit of Host Control Register 0 (HCNT0) is 1, an interrupt occurs.

Packet End Timing

When one packet ends, the interrupt flag is generated when:

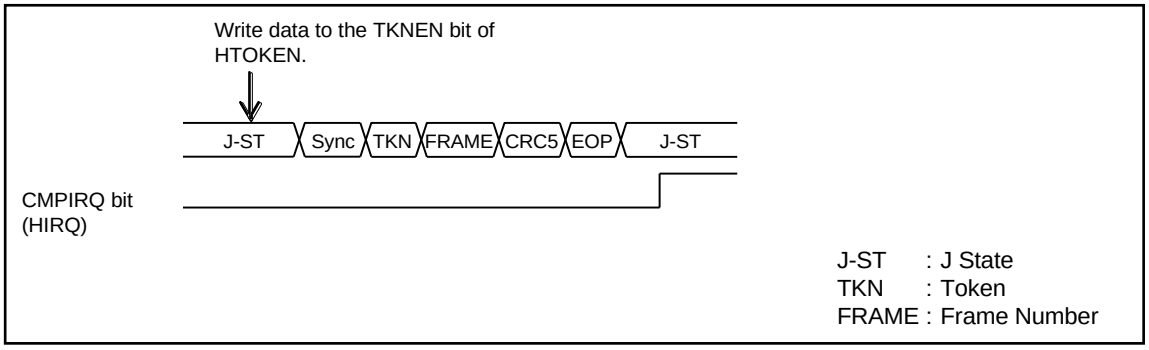
- The TKNEN bit of the Host Token Endpoint Register (HTOKEN) is (001)b, (010)b, or (011)b (SETUP token, IN token, or OUT token).

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

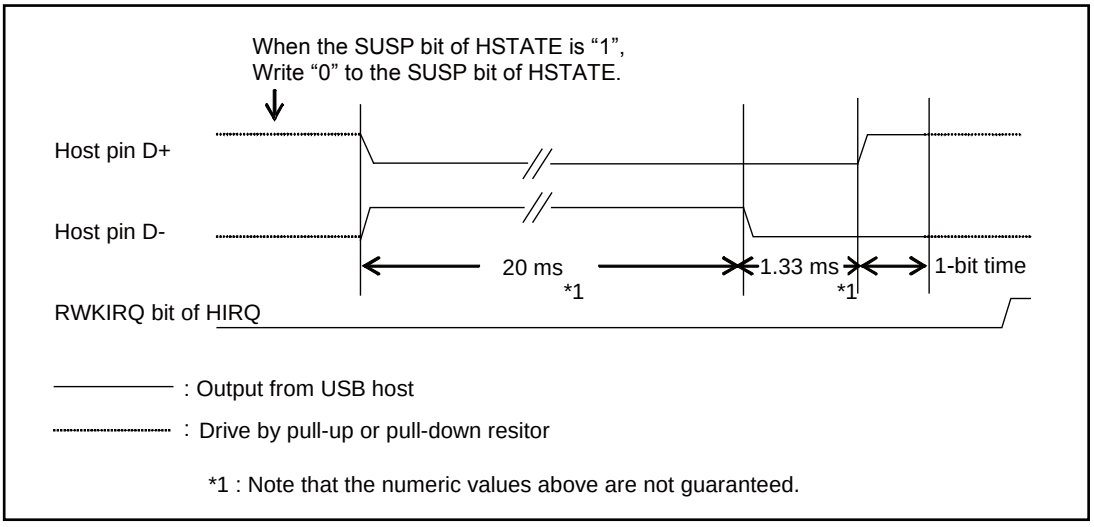
The USB bus changes from the suspend state to the resume state to resume processing when one of the following conditions is satisfied.

- 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.

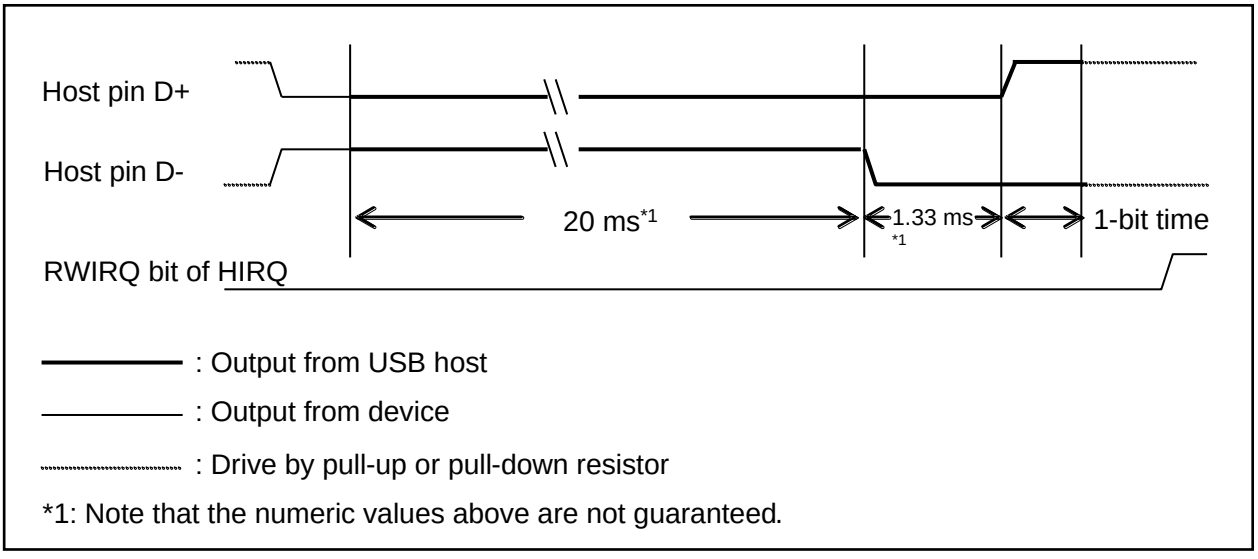
- 0 is set to the SUSP bit of the Host Status Register (HSTATE).

Figure 3-10 Resume Operation with Register (Full-speed Mode)



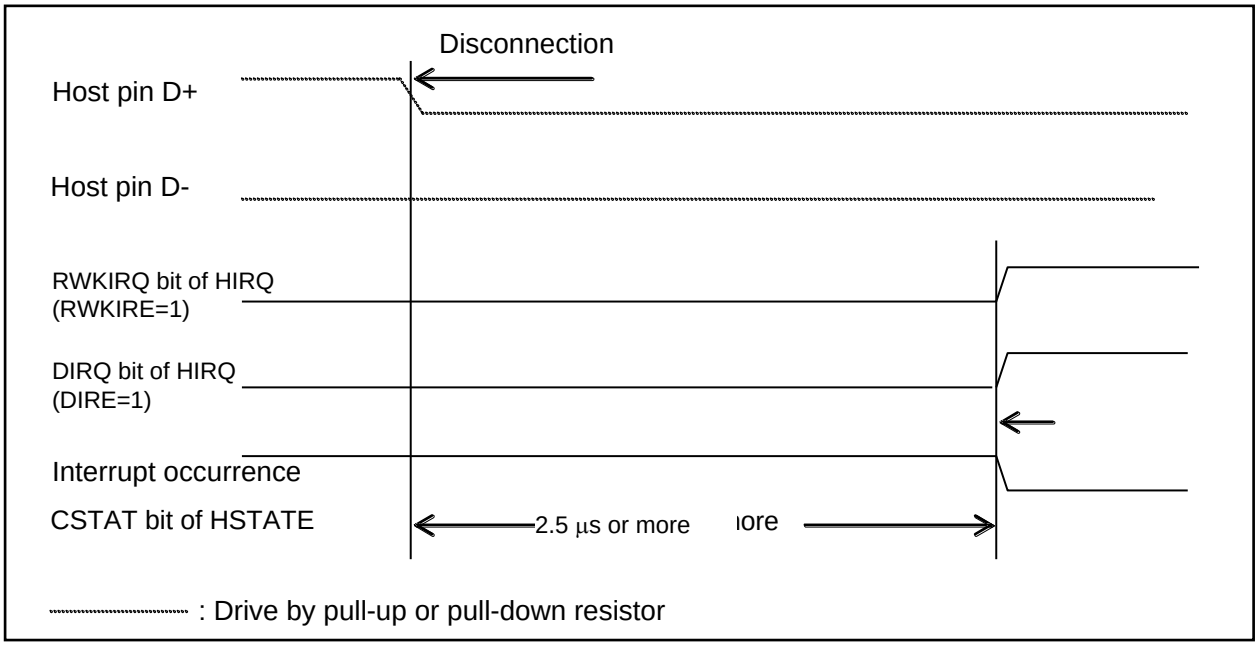
- The state that host pin D+ or D- is placed in the K-state mode has been detected.

Figure 3-11 Resume Operation by Device (Full-Speed mode)



- A device disconnection is detected.

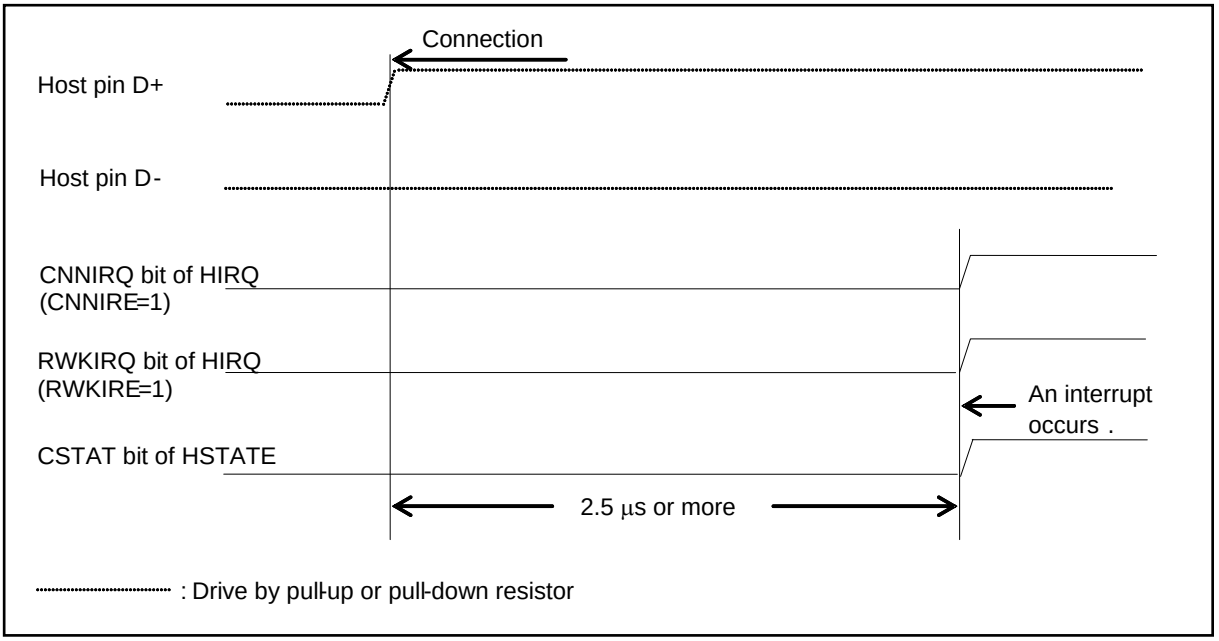
Figure 3-12 Resume Operation by Device Disconnection



-

- 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

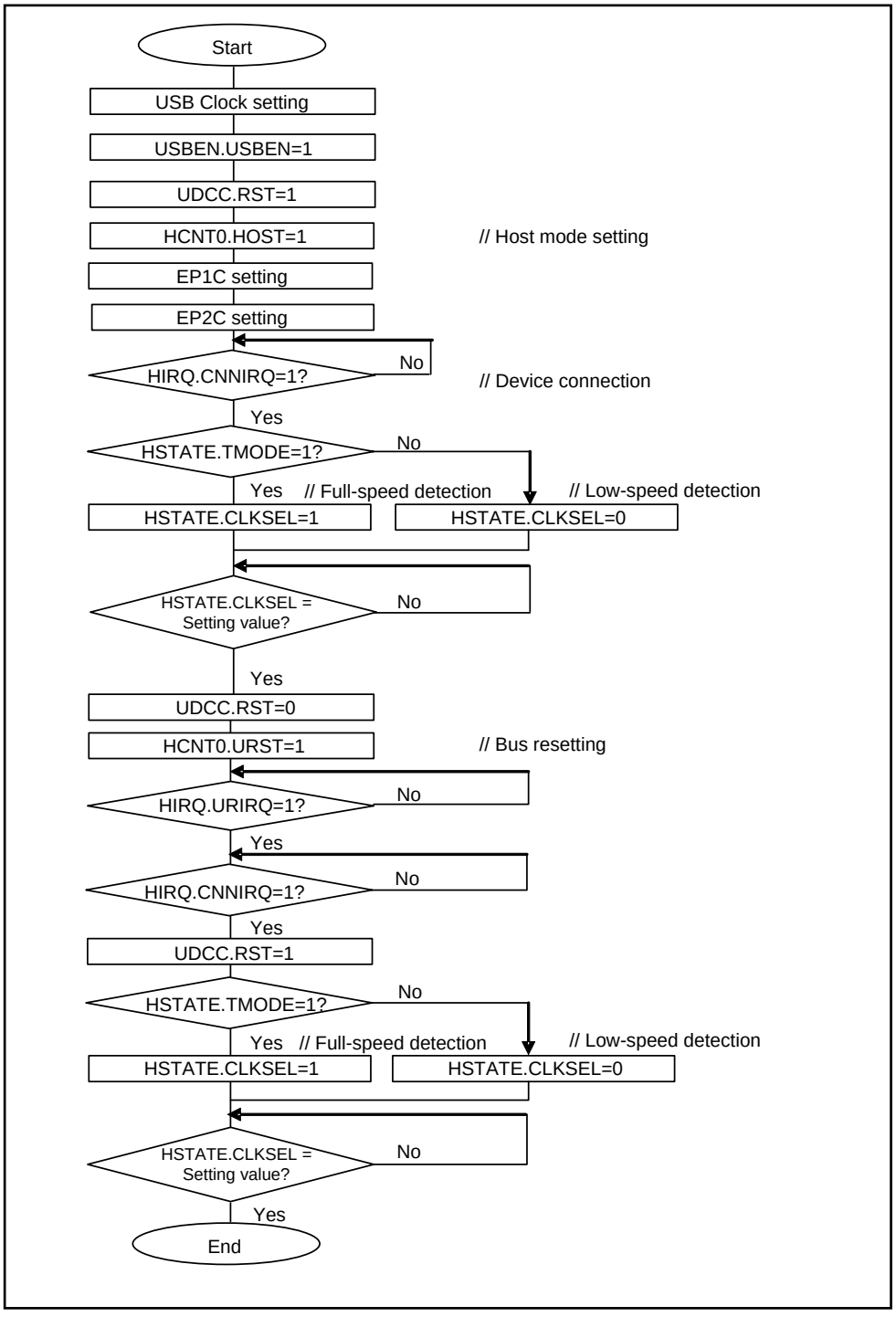
If both the host pins D+ and D- remain set to LOW for 2.5 μs or more regardless of the host or function mode, it is judged that the device has been disconnected. This then sets 0 to the CSTAT bit of the Host Status Register (HSTATE) and 1 to the DIRQ bit of the Host Interrupt Register (HIRQ). At this 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

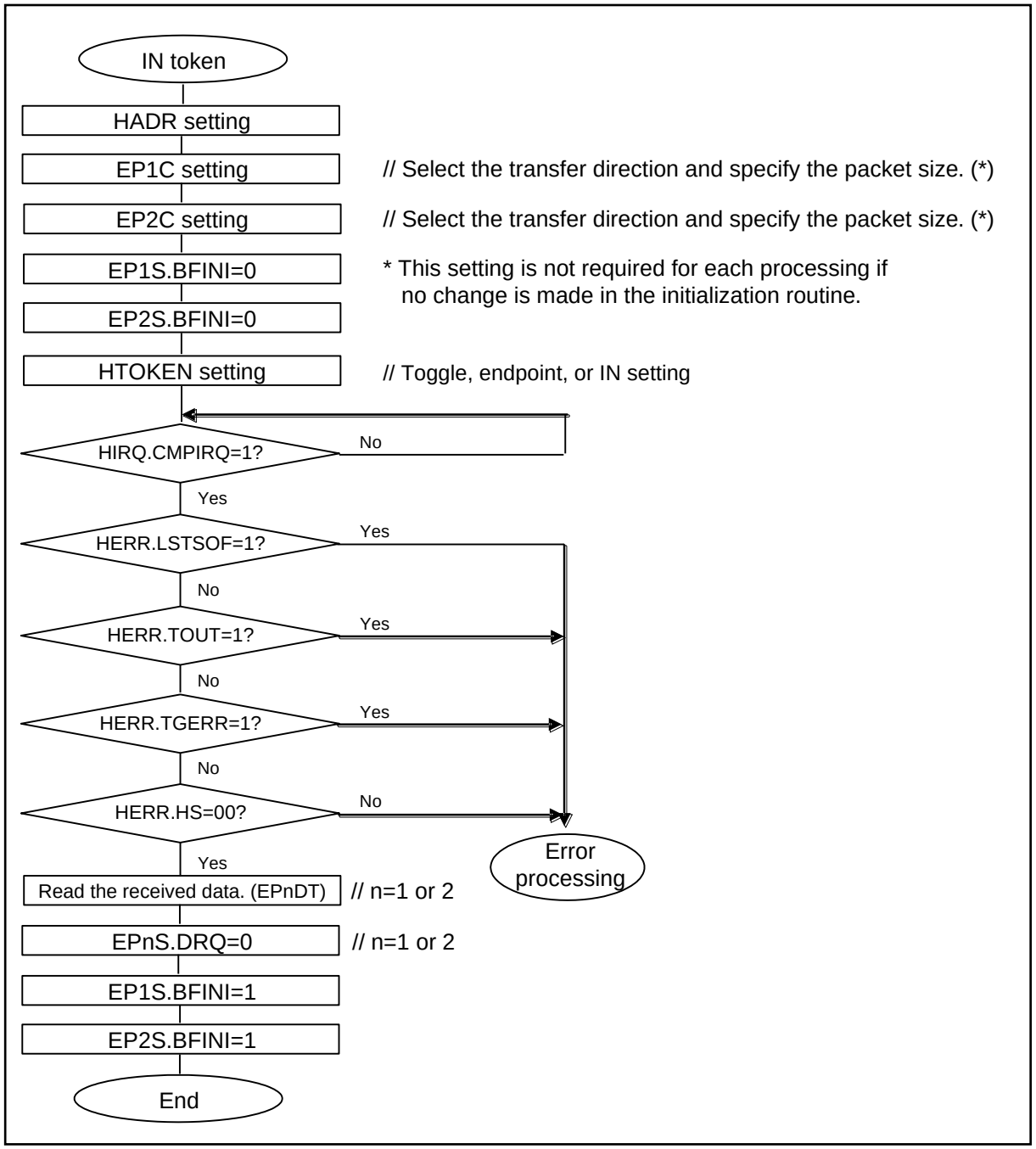
The following shows the flowchart for the USB host tokens.

Initialization and Device Detection

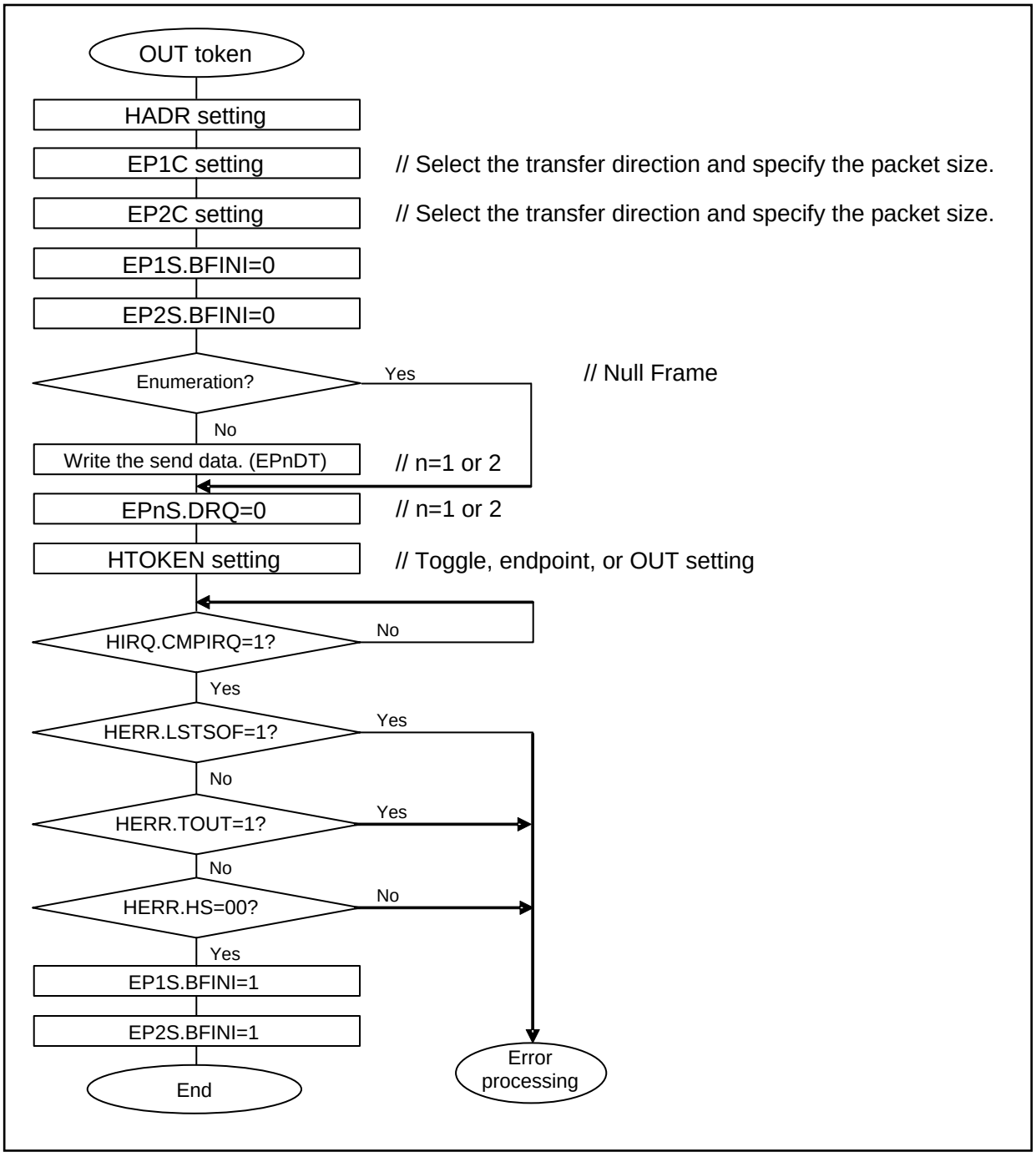


IN, OUT, or SETUP Token

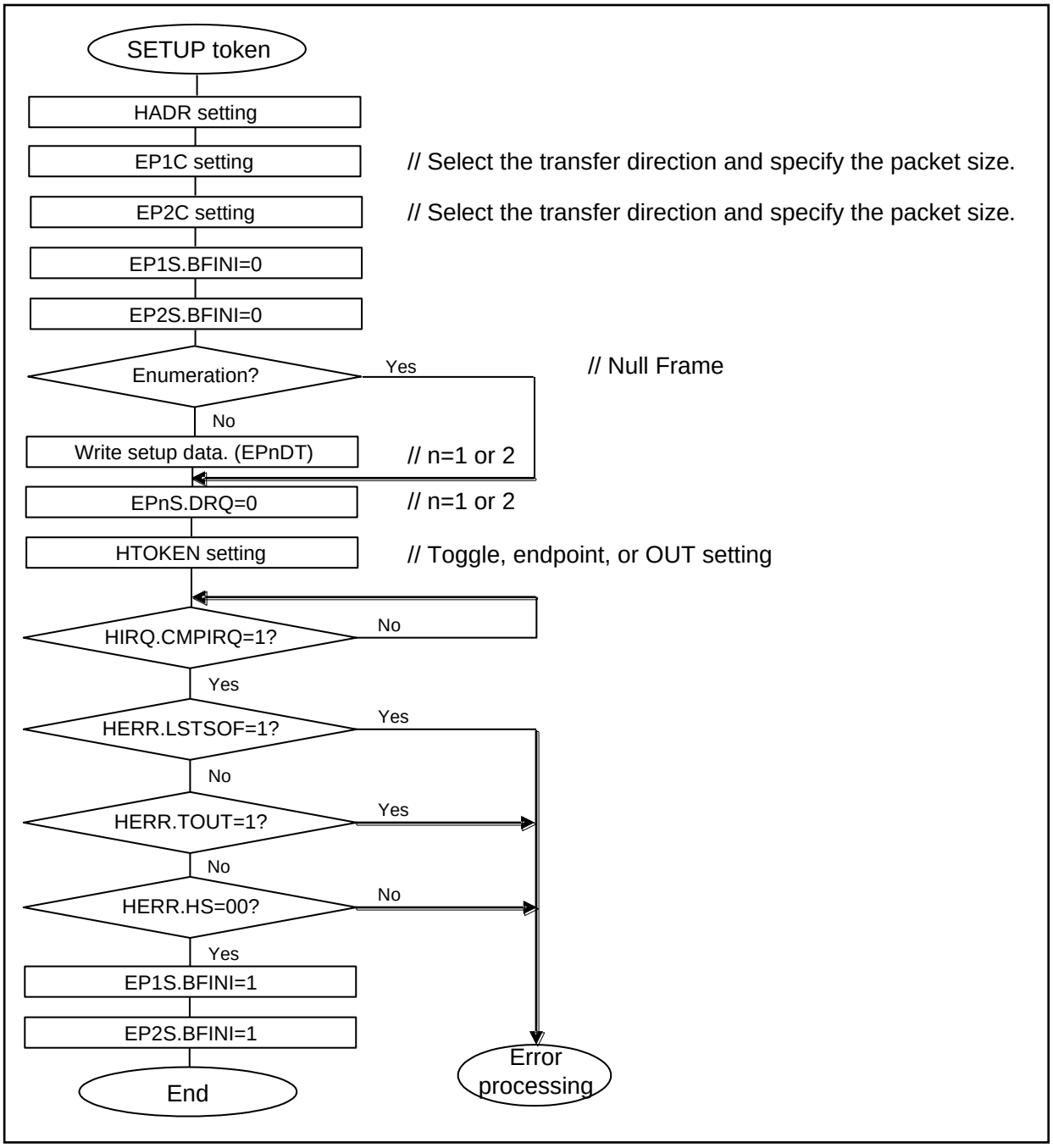
■ IN token



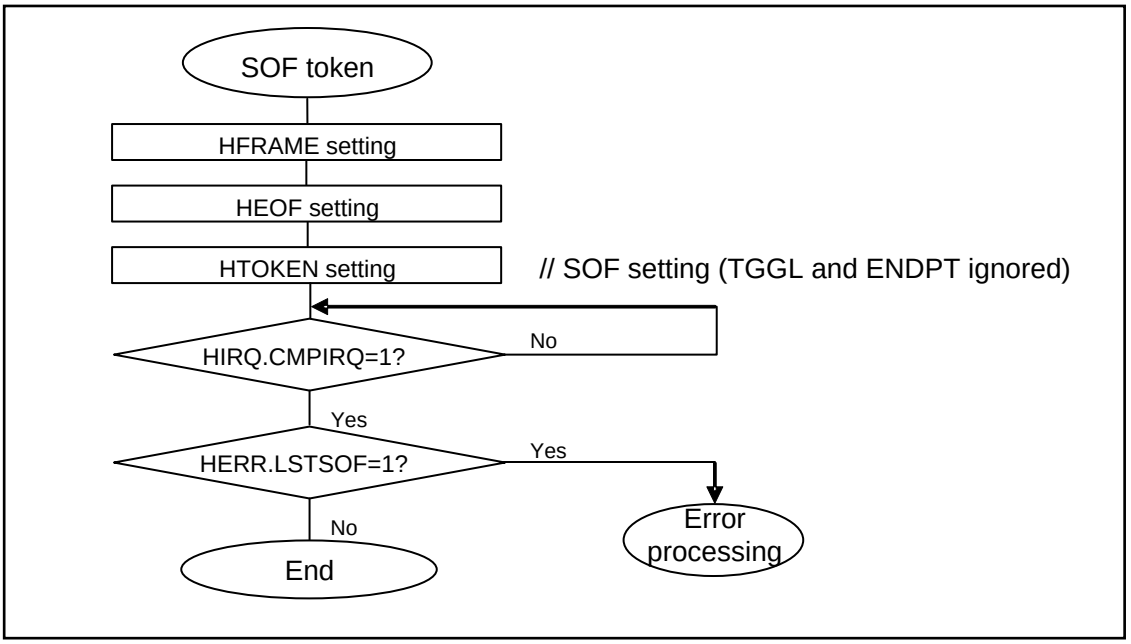
■ OUT token


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
Error processing
End

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.*
- *The setting of this bit is also effective in the device mode.*
  In the device mode, this bit 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).*

CHAPTER6: Smart Card Interface

This chapter explains the smart card interface function.

1. Overview of Smart Card Interface
2. Smart Card Interface Configuration
3. Smart Card Interface Operation
4. Smart Card Interface Interrupt
5. Smart Card Interface Setting Procedure and Program Flow
6. Smart Card Interface Registers

1. Overview of Smart Card Interface

Smart card interface is for communicating with ISO 7816 smart cards. Only asynchronous cards are supported. The interfaces contain a parallel-to-serial and serial-to-parallel converter with timer support, 16-byte transmit and receive FIFOs and a control-logic. The data transfer to and from the smart cards is controlled by the CPU. The smart card Interfaces handle the interface timing and offer limited support for data framing, timing, and error handling. The smart card pins are shared with GPIO pins. The physical smart card pins are configured using the GPIO module. Please refer to the GPIO document section for information on configuring the physical pins of the smart card.

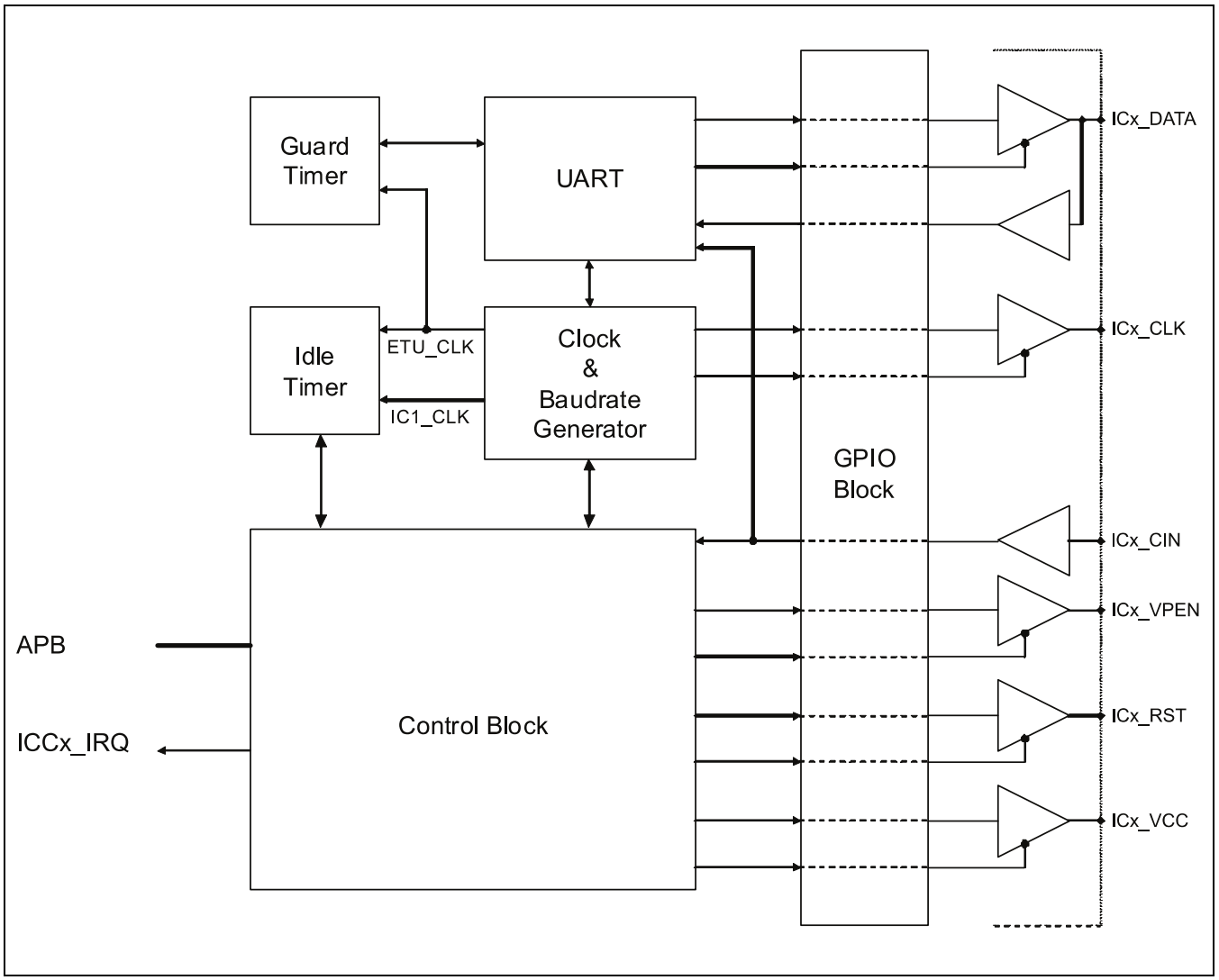
Feature of Smart Card Interface

- ISO 7816-3 supported
- Card clock frequency adjustable up to 20 MHz
- Programmable baud rate
- Available protocols:
  - Transmitter: 8E2, 8O2, 8N2
  - Receiver: 8E1, 8O1, 8N2, 8N1, 9N1
  - Inverse mode
- Resend option:
  - Transmitter: if receiver request a resend, data will be send again and interrupt is delayed
  - Receiver: if parity bit is wrong, block can request a resend
- Inversion of output data is programmable
- Card inserted or removed detection (used for interrupt generation)
- Programmable guard time
- FIFO size:
  - For receiver: 16-bytes
  - For transmitter: 16-bytes
- Programmable idle timer (interrupt may occur when expired)
- Interrupt controlled

2. Smart Card Interface Configuration

The UART block (UART = Universal Asynchronous Receiver Transmitter) controls the protocol of the serial asynchronous data. The baud rate clock is provided by the baud rate generator. In addition, two timers are available; the guard timer, which allows a gap between two successive transmitted bytes, and the idle timer, which can be clocked by the ETU clock (ETU= Elementary Time Unit) or the card clock (ICx_CLK). The idle timer can be used as a general purpose timer. It can be triggered by either a start bit, or directly triggered by a register access. The control block interfaces to the control bus and the interrupt (ICCx_IRQ) is provided to the CPU.

Figure 2-1 Smart Card Interface Block Diagram



Note:

- *x stands for channel number. The following are likewise.*

3. Smart Card Interface Operation

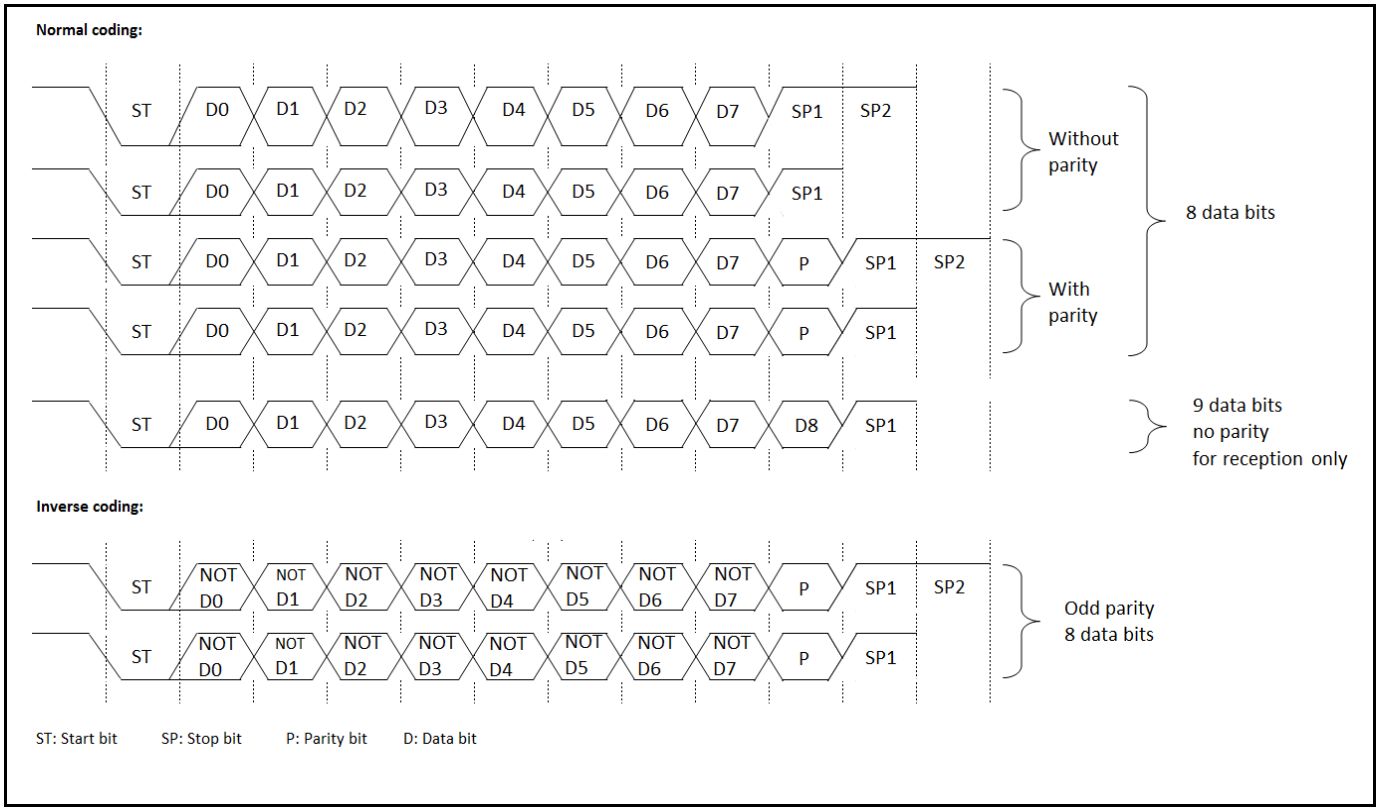
Smart card interface operates in half-duplex serial asynchronous communication with smart cards.

3.1 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 Frm1 bit of Global Control1 Register (GLOBALCONTROL1) determines the coding style of data transmission (normal coding or inverse coding). The Frm1 bit has no influence on start bit or parity bit. If parity is used, the parity bit is always placed between the last data bit and the first stop bit.
- In normal coding style, LSB is transmitted first and low level is logic zero.
- In inverse coding style, MSB is transmitted first and high level is logic zero. Odd parity should be configured for inverse coding.

Figure 3-1 shows the transmit/received data formats for normal coding and inverse coding.

Figure 3-1 Example Transmit/Received Data Format (Normal/Inverse Coding)



Notes:

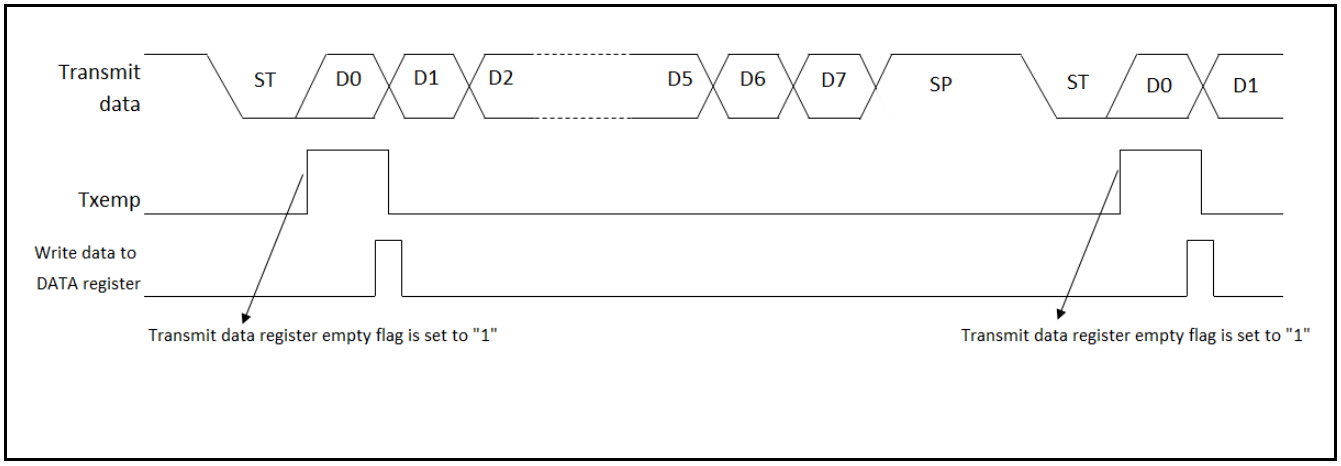
- *The above figure shows formats when the data length is set to 8 or 9 bits.*
- *If Frm1 bit of the Global Control1 Register (GLOBALCONTROL1) is set to 1 (MSB first, inverse coding), bits D0-D7 are logically inversed. This has no effects on Start bit, Stop bit and Parity bit.*

3.2 Data Transmission

- If the transmit data register empty flag bit (Txemp) of the Status Register (STATUS) is 1, the transmit data can be written in the Transmit Data Register (DATA). (When write FIFO is enabled, transmit data comes from FIFO and write to DATA register has no effect.)
- If transmit data is written in the Transmit Data Register (DATA), the transmit data register empty flag bit (STATUS.Txemp) is set to 0.
- Once smart card interface is enabled by setting IccDisable bit of Global Control2 Register (GLOBALCONTROL2) to 0, transmit data is loaded to the transmit shift register when Txemp status flag is set to 0 if no resend require occurs (STATUS.TxResend = 0) or resend function is disabled (GLOBALCONTROL1.Resnd = 0), followed by sequential transmission starting with the start bit.
- When transmission starts, the transmit data register empty flag bit (STATUS.Txemp) is set to 1 again.

Figure 3-2 shows the timing when the STATUS.Txemp flag bit is set and cleared.

Figure 3-2 Timing of Transmit Data Register Empty Flag Bit (STATUS.Txemp)



- When the transmitter starts to transmit the start bit of data, Txact bit of Status Register is set to 1 (STATUS.Txact = 1), indicating that the transmitter is active. In the case when guard timer is disabled (GLOBALCONTROL1.Guaen = 0), the Txact bit is set to 0 when transmitter finishes transmitting the stop bits. When guard timer is enabled (GLOBALCONTROL1.Guaen = 1), Txact bit is set to 0 when the configured guard time (GUARDTIMER.Gtreg) expires.

Figure 3-3 shows the timing of STATUS.Txact flag bit when guard timer is disabled.

Figure 3-3 Timing of Transmitter Active Flag Bit (STATUS.Txact) (Guard Timer Disabled)

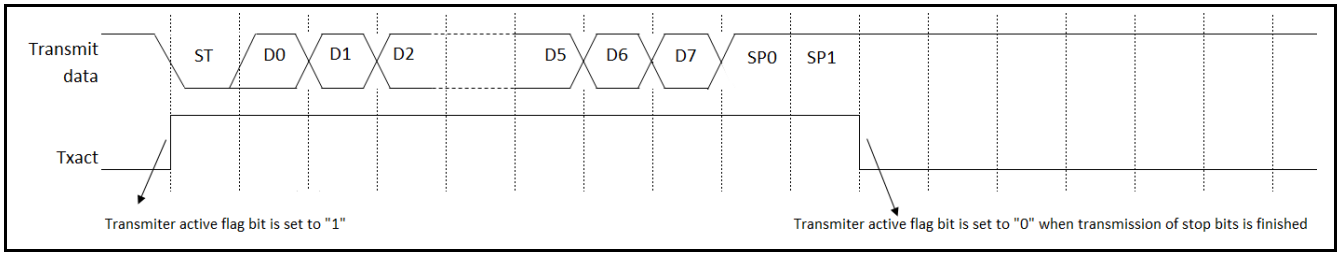
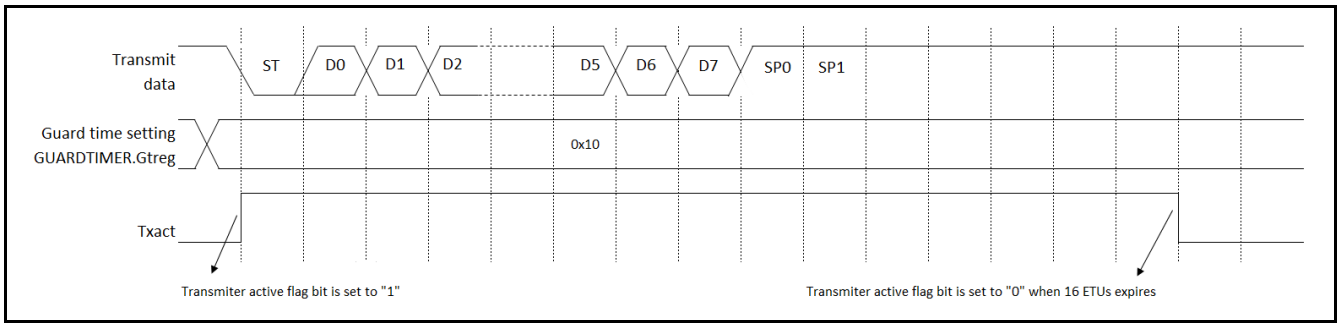


Figure 3-4 shows the timing of STATUS.Txact flag bit when guard timer is enabled.

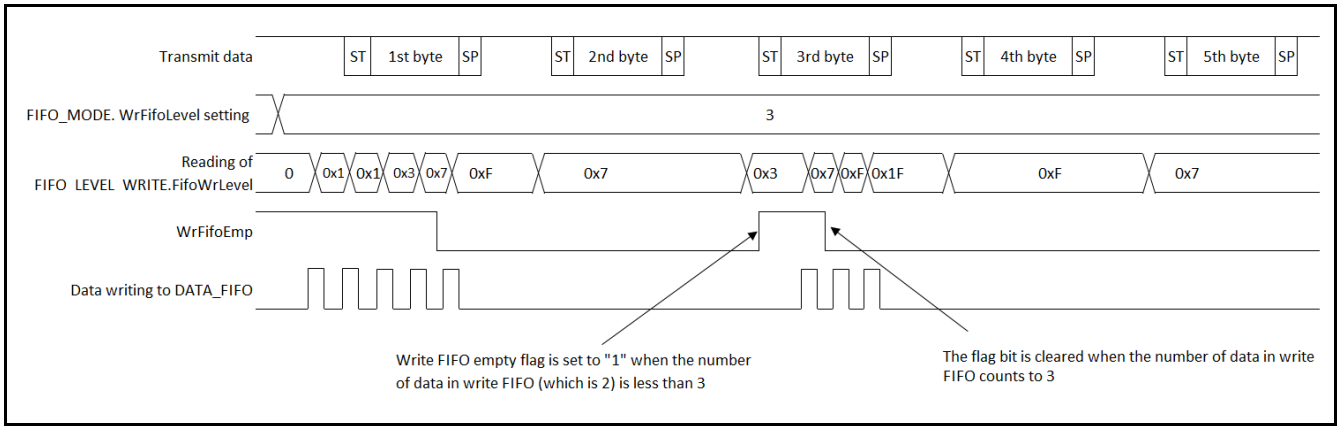
Figure 3-4 Timing of Transmitter Active Flag Bit (STATUS.Txact) (Guard Timer Enabled)



- When write FIFO is enabled (FIFO_MODE.FifoEn = 1), the data write to FIFO is transmitted. If the number of data in write FIFO is less than write FIFO level (FIFO_MODE.WrFifoLevel), write FIFO empty flag bit (STATUS.WrFifoEmp) is set to 1. When the number of data in write FIFO is equal to or more than write FIFO level, the WrFifoEmp flag bit is cleared to 0.

Figure 3-5 shows the timing of STATUS.WrFifoEmp flag bit.

Figure 3-5 Timing of Write FIFO Empty Flag Bit (STATUS.WrFifoEmp)

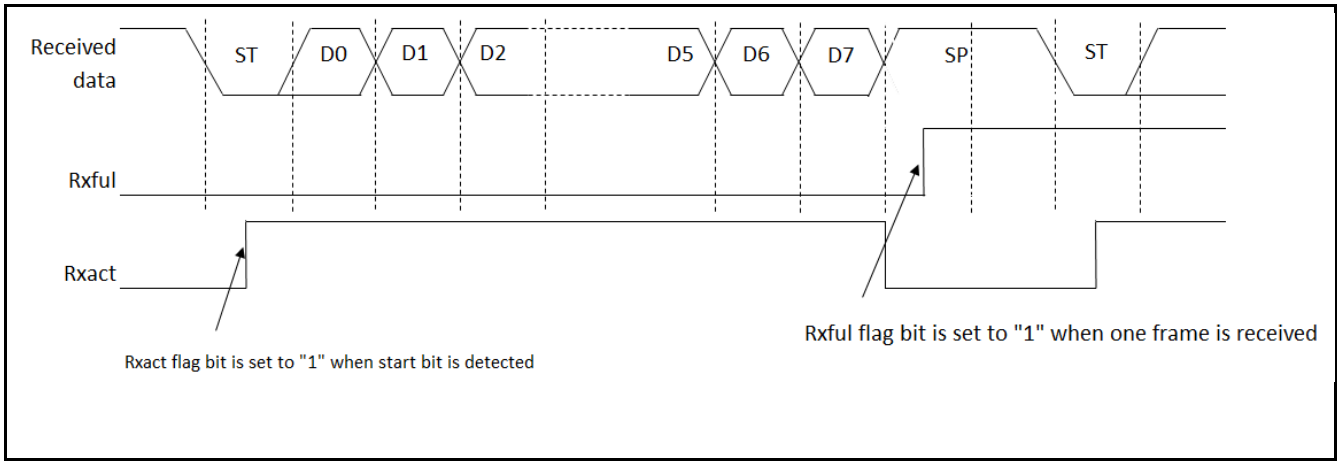


3.3 Data Reception

- Once smart card interface is enabled by setting IccDisable bit of Global Control2 Register (GLOBALCONTROL2) to 0, the interface performs reception when a start bit is detected on ICx_DATA line.
- Upon detection of the start bit, one-frame data reception takes place according to the data format set in the global control1 register (GLOBALCONTROL1.Parity, Frm0, Mode8n1, Rx8n1). A start bit is detected (Status.Rxact = 1) when falling edge is detected on ICx_DATA line and the low level keeps till the sampling point (half ETU after the falling edge).
- When one-frame reception is completed, the received data is put to receiver load register and the received register full flag bit (STATUS.Rxful) is set to 1. Data frame can be read from DATA register. Be noted that the received data will not be loaded to receiver load register if parity error occurs and resend function is enabled, and the setting of received register full flag bit is postponed.
- To read received data, perform reading of the received data register (DATA) after one-frame data received.
- Reading of the received data causes the received register full flag bit (STATUS.Rxful) to be cleared to 0.

Figure 3-6 shows the timing of the received data register full (STATUS.Rxful) and receiver active (Status.Rxact) flag bits.

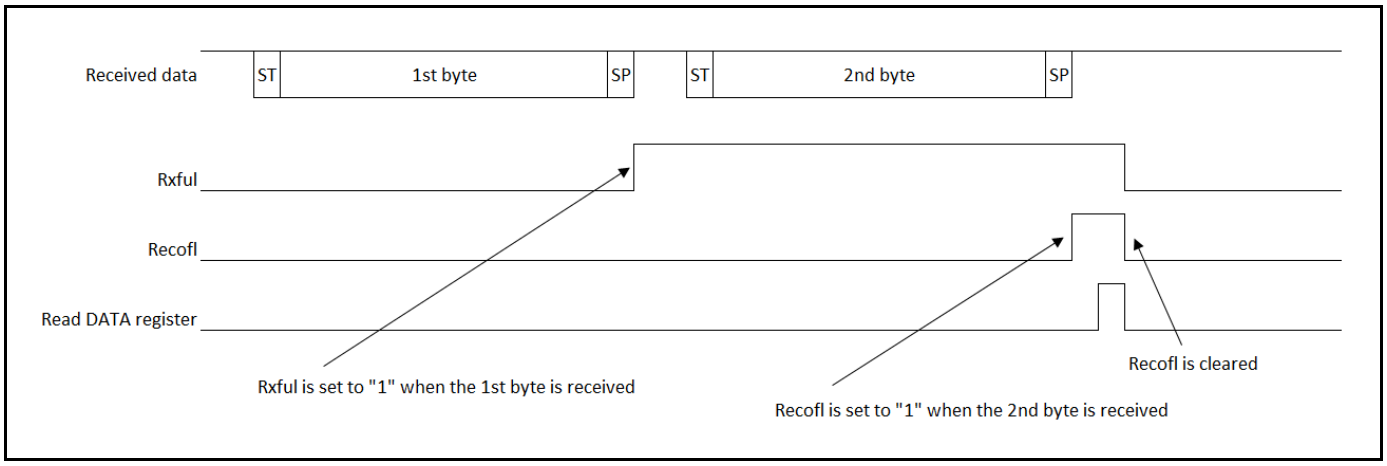
Figure 3-6 Timing of STATUS.Rxful and STATUS.Rxact Flag Bits



- If the received data is not read by CPU, and the receiver finishes receiving the next data frame, the received register overflow flag bit (STATUS.Recofl) is set to 1.
- The received register overflow flag bit bit (STATUS.Recofl) is cleared to 0 by reading DATA register.

Figure 3-7 shows the timing of the received data register overflow (STATUS.Recofl) flag bit.

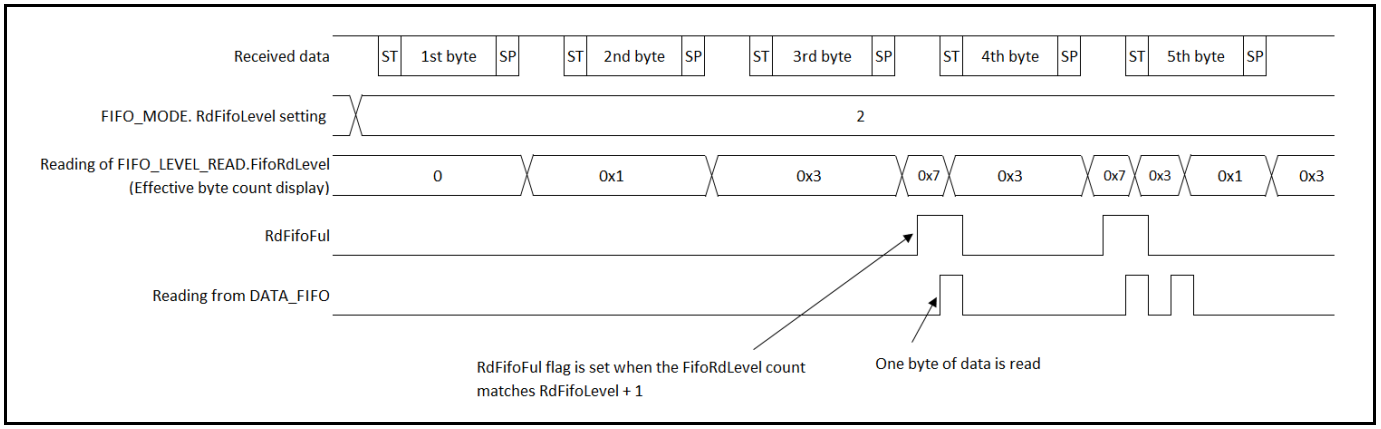
Figure 3-7 Timing of Received Data Register Overflow (STATUS.Recofl) Flag Bit



- If read FIFO is enabled, the read FIFO full flag bit (STATUS.RdFifoFul) is set to 1 when the number of received frames reaches the value set for read FIFO level plus 1 (FIFO_MODE.RdFifoLevel + 1).
- If read FIFO is enabled, the received data frame is not stored in read FIFO if parity error occurs (STATUS.Rxresend = 1) and resend function is enabled (GLOBALCONTROL1.Rxrsnd = 1).
- The read FIFO full flag is set to 0 when the number of valid data in read FIFO is less than FIFO_MODE.RdFifoLevel + 1.

Figure 3-8 shows the timing of the read FIFO full (STATUS.RdFifoFul) flag bit.

Figure 3-8 Timing of Read FIFO Full (STATUS.RdFifoFul) Flag Bit



3.4 Baud Rate Configuration

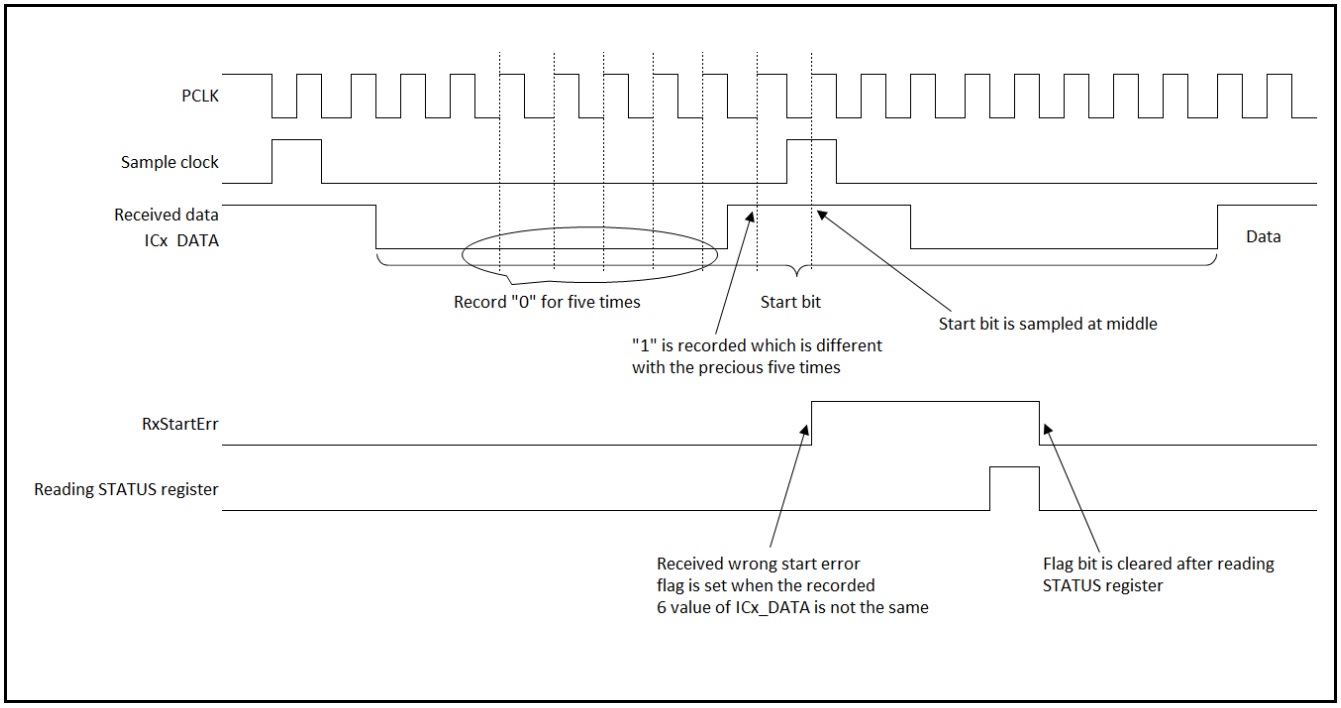
- The frequency of card clock output ICx_CLK depends on PCLK (maximum 40MHz) and card clock divider configured by ClkDivider bits of CARDCLOCK register.
  That is CardClock[Hz] = PCLK[Hz]/ CARDCLOCK.ClkDivider
- The baud rate of data transmission depends on card clock frequency and the value of Brreg bits of baud rate register (BAUDRATE.Brreg).
- According to ISO7816-3 standard, 1 ETU = (F/D) * (1/CardClock[Hz]). F/D is configured by Brreg bits of baud rate register (BAUDRATE.Brreg).
  For example, to get F/D = 31, the value 0x1F has to be programmed to BAUDRATE.Brreg. to achieve F/D = 31.5, set the LittleStep bit of BAUDRATE register to 1.

3.5 Start Bit Detection

- The start bit of received data is recognized based on detection of the falling edge of the ICx_DATA pin.
- Upon detection of the falling edge of start bit, the receiver records ICx_DATA level for 6 times before sampling at middle of start bit. If the recorded 6 values are not the same or the sampled value of start bit is not 0, the Received Wrong Start Bit error occurs and RxStartErr bit of STATUS register is set (STATUS.RxStartErr = 1).
- Once the STATUS register is read, the RxStartErr bit is cleared to 0.

Figure 3-9 shows the timing of received wrong start bit error flag (STATUS.RxStartErr)

Figure 3-9 Timing of Received Wrong Start Bit Error (STATUS.RxStartErr) Flag Bit

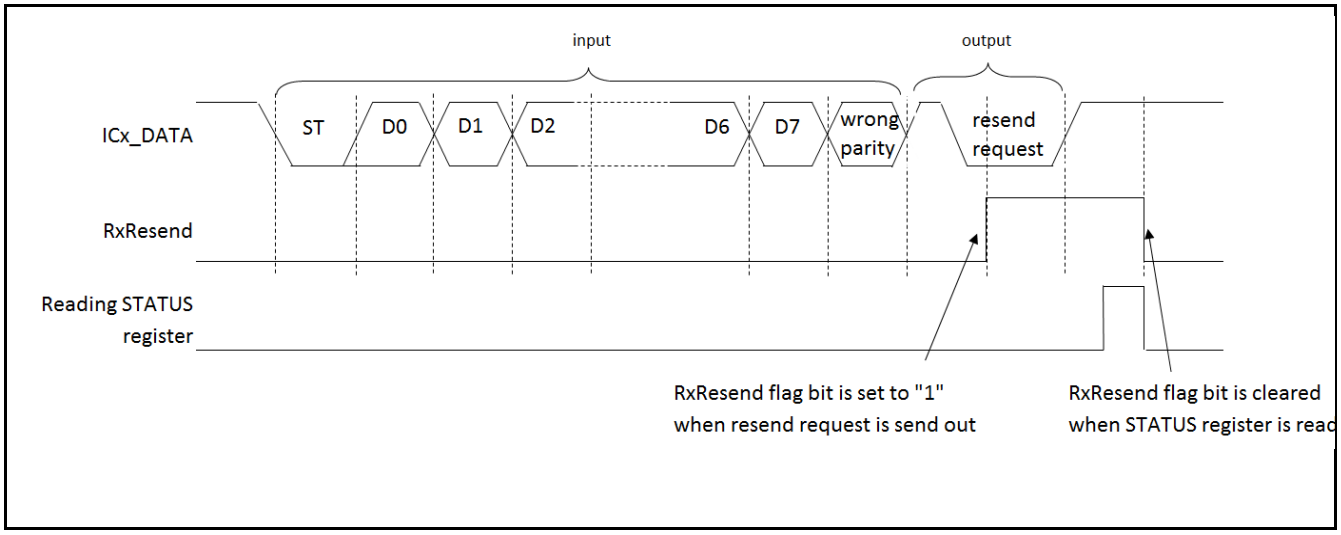


3.6 Error Detection

- When receiver detected wrong parity bit and the resend function s enabled (GLOBALCONTROL1.Resnd = 1), resend request will be sent out by putting ICx_DATA to low level for 1.5 ETUs half ETU after parity bit.
- When resend request is sent out, the Receiver Resend flag bit of STATUS register (STATUS.RxResend) is set to 1.
- When resend function is enabled, the received frame with wrong parity is not stored in data register and the received data register full flag bit (STATUS.Rxful) is not set.
- The Receiver Resend flag bit is cleared by reading STATUS register.

Figure 3-10 shows the timing of Receiver Resend flag bit (STATUS.RxResend)

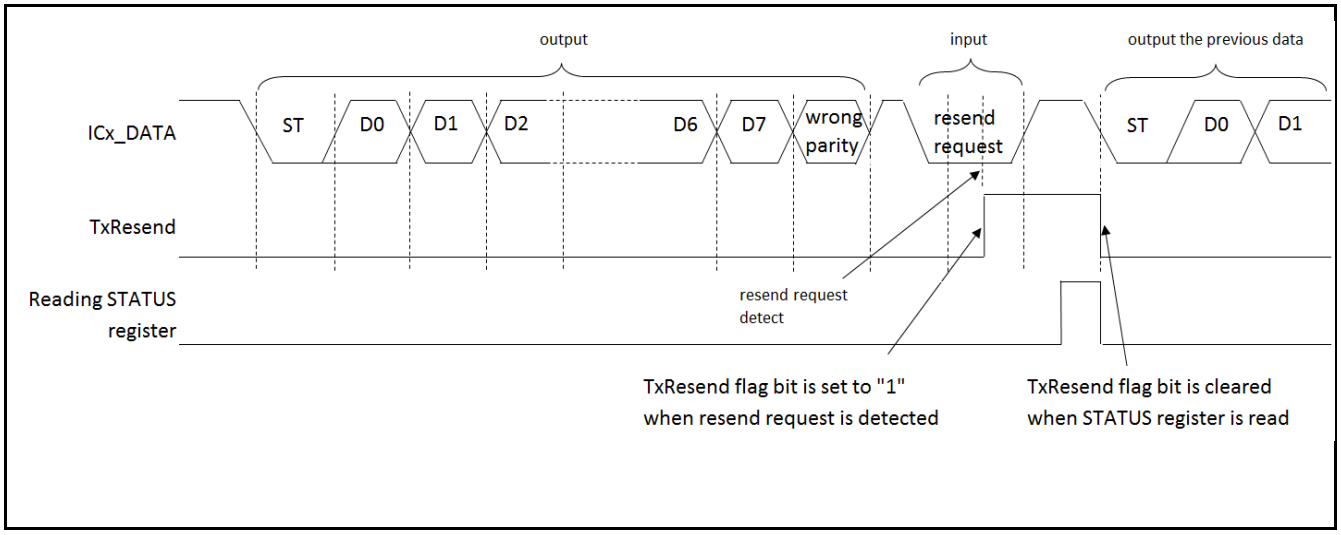
Figure 3-10 Timing of Receiver Resend (STATUS.RxResend) Flag Bit



- When transmitter detected resend request (low level on ICx_DATA half ETU after parity bit) and the resend function is enabled (GLOBALCONTROL1.Resnd = 1), transmitter will send the current data frame again and Transmitter Resend flag bit of STATUS register (STATUS.TxResend) is set to 1.
- The Transmitter Resend flag bit is cleared by reading STATUS register.

Figure 3-11 shows the timing of Transmitter Resend flag bit (STATUS.TxResend)

Figure 3-11 Timing of Transmitter Resend (STATUS.TxResend) Flag Bit



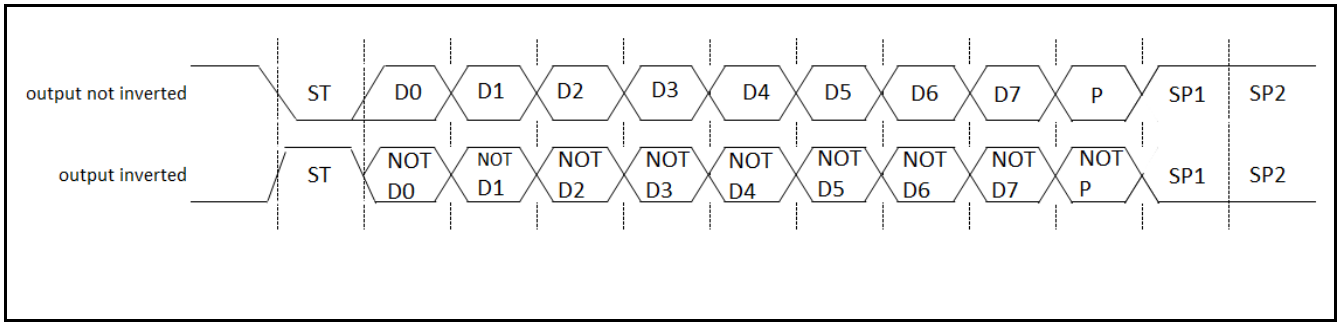
3.7 Output Inverse

By setting the InvDataOut bit of Global Control2 Register to 1, the output data on ICx_DATA line can be logically inversed.

- To inverse the output data, set GLOBALCONTROL2.InvDataOut to 1.
- The Start bit, Stop bit and Parity bit are all inversed.
- Only the output data can be inversed. This setting has no effect on input data.

Figure 3-12 shows the data frame when output data is inversed.

Figure 3-12 Data Frame when Output Data is Inversed



3.8 Port Control

The level on ICx_VPEN, ICx_VCC and ICx_RST outputs are controlled by software. The level on ICx_DATA and ICx_CLK can be controlled by hardware or sorftware.

- The output level of ICx_VPEN, ICx_VCC and ICx_RST pins depend on the value configured to bits VpenBuf, VccBuf, and RstBuf of PORTCONTROL register respectively. Configure the register bits to 0 if low level is needed and to 1 vice versa.
- If bit Ckmod of GLOBALCONTROL1 register is set to 1 (GLOBALCONTROL1.Ckmod = 1), the level on ICx_CLK depends on the value configured to Clkpt bit of PORTCONTROL register (PORTCONTROL.Clkpt).
- If bit Ckmod of GLOBALCONTROL1 register is set to 0 (GLOBALCONTROL1.Ckmod = 0), the level on ICx_CLK is controlled by UART block automatically (hardware).
- If bit Iomod of GLOBALCONTROL1 register is set to 1 (GLOBALCONTROL1.Iomod = 1), the level on ICx_DATA depends on the value configured to Io1 bit of PORTCONTROL register (PORTCONTROL.Io1).
- When ICx_DATA output level is controlled by software (GLOBALCONTROL1.Iomod = 1), Trimod bit of PORTCONTROL register should be configured to 1 (PORTCONTROL.Trimod = 1) to enable controlling of data output enable signal by software.
- If PORTCONTROL.Trimod is configured to 1, set Io1en bit of PORTCONTROL register to 0/1 to enable/disable data output on ICx_DATA pin.
- If bit Iomod of GLOBALCONTROL1 register is set to 0 (GLOBALCONTROL1.Iomod = 0), the level on ICx_DATA is controlled by UART block automatically (hardware).

4. Smart Card Interface Interrupt

Smart card interface generates transmit, received, card event detect or idle timer expired interrupts. These interrupt requests can be generated if:

- Received data is set in the Data Register (DATA) or the receiver is active.
- Transmit data is transferred from the Data Register (DATA) to the transmit shift register and the data transmission is started.
- An event is detected on the ICx_CIN (x stands for 0 or 1. The following are likewise) pin.
- The idle timer is expired.

Table 4-1 shows the relationship between the smart card interface interrupt control bits and the interrupt factors.

Table 4-1 Smart Card Interface 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
Received	RxfullIrq	IRQ_STATUS	A single-byte received	GLOBALCONTROL1.Maskrxful	Reading the Received Data Register (DATA)
	RxstbiIrp	IRQ_STATUS	Received start-bit detected, receiver is active	GLOBALCONTROL1.Masksti	Reading the IRQ Status Register (IRQ_STATUS)
	RdFifoIrq	IRQ_STATUS	Received data volume matching the value set for RdFifoLevel	FIFO_MODE. RdFifoIrqEn	Reading the Read Data FIFO Register (DATA_FIFO) till the number of data in read FIFO is no larger than RdFifoLevel
	RdFifoOvrIrq	IRQ_STATUS	Read FIFO overflow	FIFO_MODE. RdFifoOvrIrqEn	Flushing read FIFO by setting FIFO_CLEAR_MSB_READ.ClrRdFifo to 1
Transmit	TxempIrp	IRQ_STATUS	The Transmit Data Register is empty	GLOBALCONTROL1. Masktxemp	Writing to the Transmit Data Register (DATA)
	WrFifoIrq	IRQ_STATUS	Transmit data volume matching the value set for WrFifoLevel	FIFO_MODE. WrFifoIrqEn	Writing to the Write Data FIFO Register (DATA_FIFO) till the number of data in write FIFO is no smaller than WrFifoLevel
Card event	CardEventIrq	IRQ_STATUS	Block detected a change on the card detect input (ICx_CIN)	GLOBALCONTROL1. Maskcaevent	Reading the IRQ Status Register (IRQ_STATUS)
Idle timer	IdtexpIrq	IRQ_STATUS	Idle timer expired	GLOBALCONTROL1.Maskitexp	This bit can only be cleared by restarting or disabling the idle timer

4.1 Received Interrupt and Flag Set Timing

Data reception can be interrupted by a Received Completion (IRQ_STATUS: RxfullIrq =1) and receiver activation can be detected (IRQ_STATUS: RxstbiIrp =1).

Received Interrupt and Flag Set Timing

Upon detection of the first stop bit, received data is stored in the Received Data Register (DATA), and the received data register full flag is set (IRQ_STATUS. RxfullIrq =1). If received interrupt is enabled (GLOBALCONTROL1.Maskrxful = 1), a received interrupt occurs. The received interrupt is cleared when data is read from DATA register.

When a start bit is detected and the received start bit interrupt is enabled (GLOBALCONTROL1.Masksti = 1), the received start bit flag is set (IRQ_STATUS.RxstbiIrq=1) and a received start bit interrupt occurs. This interrupt is cleared by reading IRQ_STATUS register.

Note:

- *If parity error occurs and resend function is enabled (GLOBALCONTROL1.Resnd = 1), data is not put into DATA register and interrupt is postponed.*

Figure 4-1 RxfullIrq (Received Data Register Full) Flag Bit Set Timing

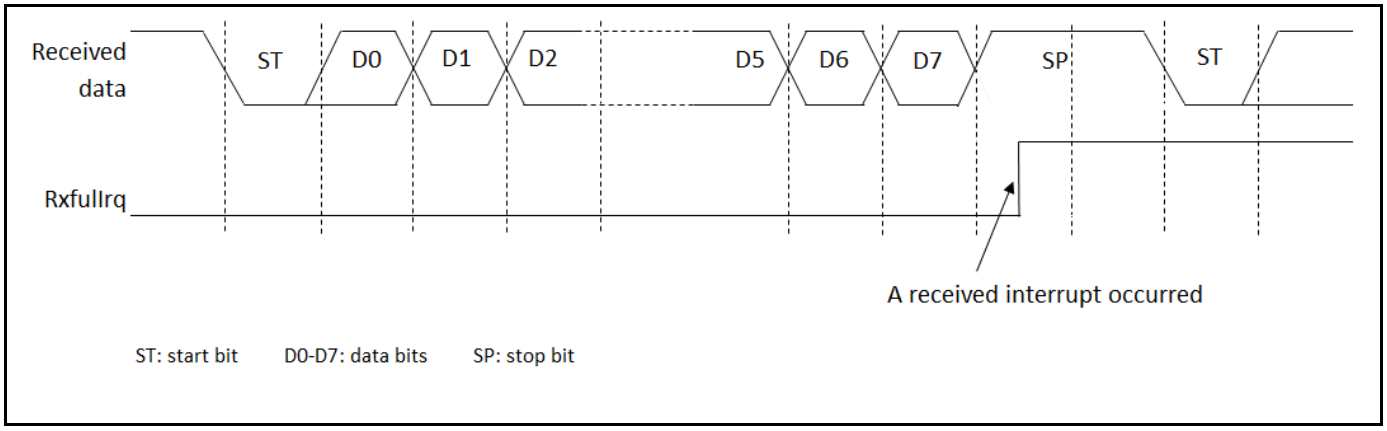
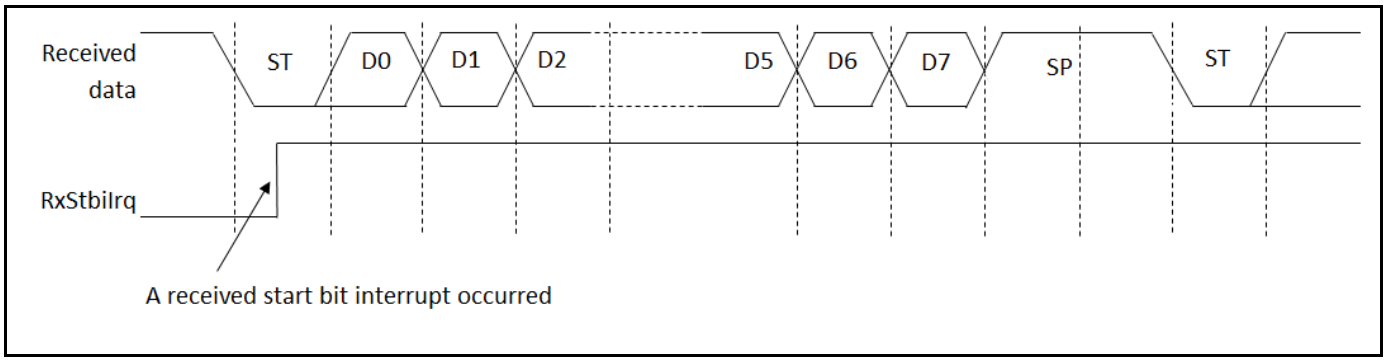


Figure 4-2 RxStbiIrq (Received Start Bit) Flag Bit Set Timing



4.2 Interrupt and Flag Set Timing when Read FIFO is Used

If the read FIFO is used, an interrupt occurs when the number of data in read FIFO reaches RdFifoLevel (preset for the FIFO_MODE register).

Interrupt and Flag Set Timing when Read FIFO is Used

If the read FIFO is used, an interrupt occurs depending on the value set for FIFO_MODE. RdFifoLevel register bits.

- When the number of received data in read FIFO reaches RdFifoLevel + 1, and read FIFO full interrupt is enabled (FIFO_MODE. RdFifoIrqEn = 1), the read FIFO full interrupt flag (IRQ_STATUS. RdFifoIrq) of the IRQ Status register is set to 1 and a read FIFO full interrupt occurs.
- When data is read from the Read Data FIFO Register (DATA_FIFO) till the number of data in read FIFO is less than RdFifoLevel + 1, the read FIFO full flag (IRQ_STATUS. RdFifoIrq) is cleared.
- If the valid received data amount is the same as the FIFO capacity and if the next data is received when read FIFO overflow interrupt is enabled (FIFO_MODE. RdFifoOvrIrqEn = 1), the read FIFO overflow flag (IRQ_STATUS. RdFifoOvrIrq =1) of the IRQ Status register is set to 1 and a read FIFO overflow interrupt occurs.

When read FIFO is flushed by writing 1 to Read FIFO Clear Register (FIFO_CLEAR_MSB_READ. ClrRdFifo), the read FIFO overflow interrupt is cleared.

Figure 4-3 RdFifoIrq (Read FIFO Full Interrupt) Flag Bit Set Timing

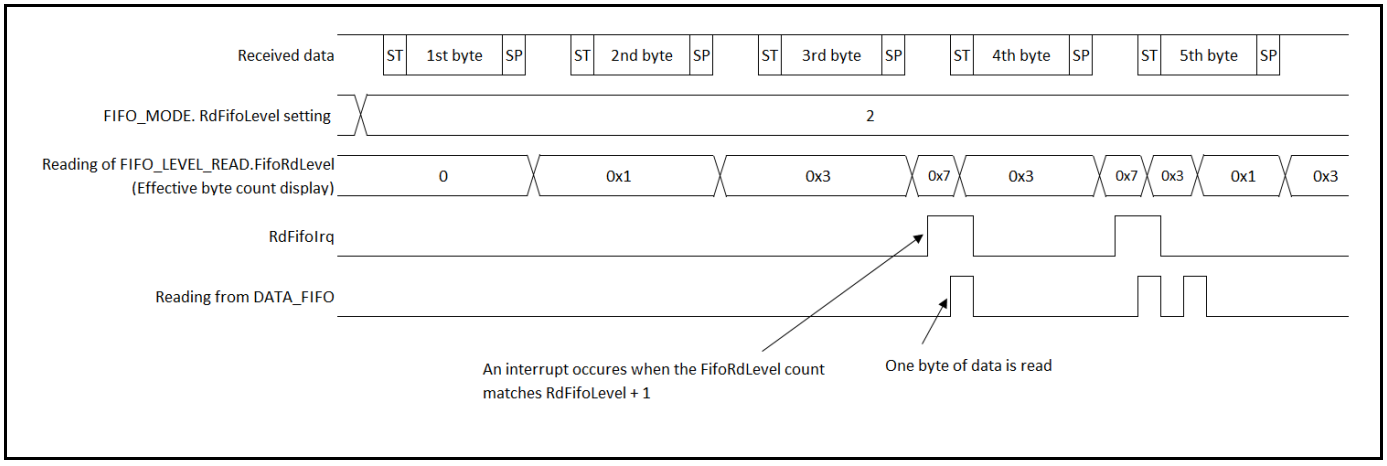
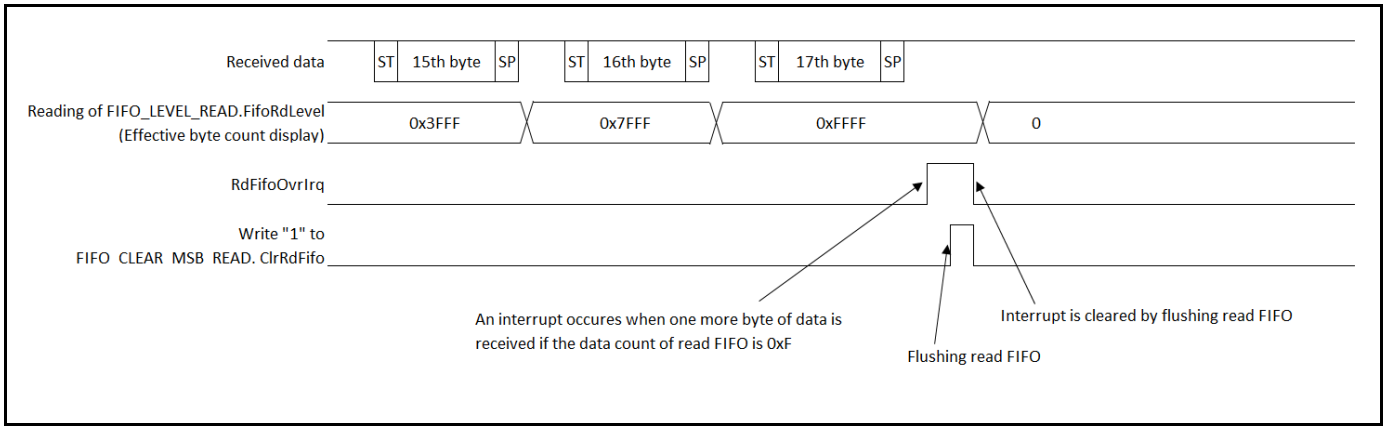


Figure 4-4 RdFifoOvrIrq (Read FIFO Overflow) Flag Bit Set Timing



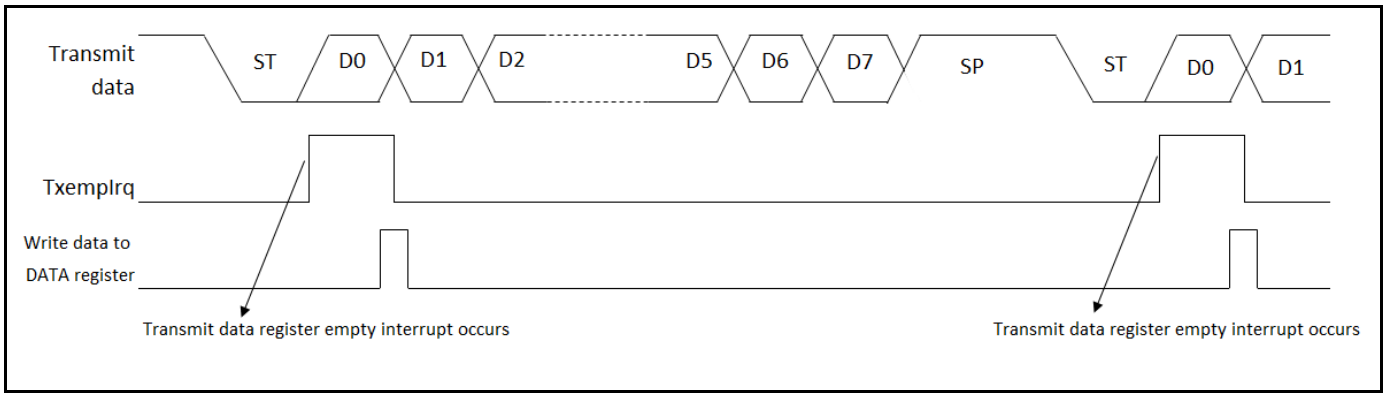
4.3 Transmit Interrupt and Flag Set Timing

A transmit interrupt occurs when transmit data is transferred from the Transmit Data Register (DATA) to the transmit shift register (IRQ_STATUS. TxempIrq = 1). Transmission starts when no transmission is performed (STATUS.Txact = 0).

Transmit Data Register Empty Flag (IRQ_STATUS.TxempIrq) Set Timing

After data has been transferred from the Transmit Data Register (DATA) to the transmit shift register (STATUS.Txemp = 1), the next data can be written to DATA register. If transmit interrupt is enabled (GLOBALCONTROL1.Masktxemp = 1) during this time, transmit data register empty flag is set (IRQ_STATUS.TxempIrq = 1) and a transmit interrupt occurs. IRQ_STATUS.TxempIrq bit is cleared to 0 when data is written to the Transmit Data Register (DATA).

Figure 4-5 Transmit Data Register Empty Flag (IRQ_STATUS.TxempIrq) Set Timing



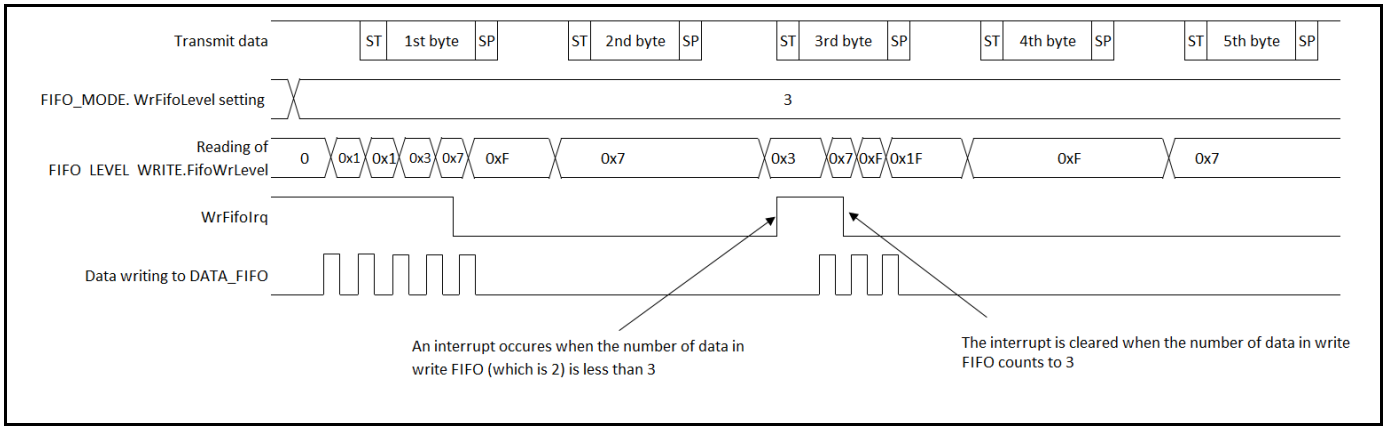
4.4 Interrupt and Flag Set Timing when Write FIFO is Used

When the write FIFO is used, an interrupt occurs if the number of valid data in write FIFO (FIFO_LEVEL_WRITE.FifoWrLevel) is less than the value set to FIFO_MODE.WrFifoLevel.

Transmit Interrupt and Flag Set Timing when Write FIFO is Used

- When the write FIFO empty interrupt is enabled (FIFO_MODE. WrFifoIrqEn = 1), an interrupt occurs if the number of valid data in write FIFO is less than FIFO_MODE.WrFifoLevel, and the write FIFO empty interrupt flag is set to 1 (IRQ_STATUS. WrFifoIrq = 1).
- If the number of valid data in write FIFO is equal to or more than FIFO_MODE.WrFifoLevel, the write FIFO empty interrupt is cleared (IRQ_STATUS. WrFifoIrq = 0).

Figure 4-6 Transmit Interrupt Timing when Write FIFO is Used



4.5 Card Event Interrupt and Flag Set Timing

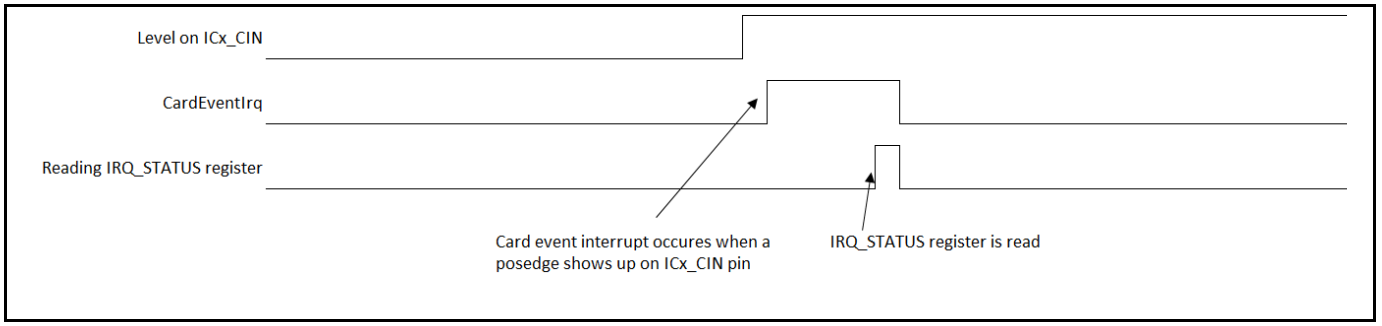
If there is an event (card plug-in or plug-out) on ICx_CIN pin and the card event interrupt is enabled (GLOBALCONTROL1. Maskcaevent = 1), the card event interrupt occurs.

Card Event Interrupt and Flag Set Timing

- When card event interrupt is enabled (GLOBALCONTROL1. Maskcaevent = 1), an interrupt occurs when the level on ICx_CIN input changes, and the card event interrupt flag is set to 1 (IRQ_STATUS.CardEventIrq = 1).
- The card event interrupt is cleared by reading the IRQ status register (IRQ_STATUS).

Figure 4-7 shows the timing of card event interrupt.

Figure 4-7 Card Event Interrupt Flag (IRQ_STATUS. CardEventIrq) Set Timing



4.6 Idle Timer Expired Interrupt and Flag Set Timing

If the idle timer is expired (down count to 0) and the idle timer expired interrupt is enabled (GLOBALCONTROL1.Maskitexp = 1), the idle timer expired interrupt occurs.

Idle Timer Expired Interrupt and Flag Set Timing

- When idle timer expired interrupt is enabled (GLOBALCONTROL1.Maskitexp = 1), an interrupt occurs when the idle timer counts to 0, and the idle timer expired interrupt flag is set to 1 (IRQ_STATUS.IdtexpIrq = 1).
- The idle timer is a general purpose 16-bit down counter which can either be clocked by the card clock (ICx_CLK) or the ETU clock (baud rate clock) by setting the Idtsc bit of global control1 register bit (GLOBALCONTROL1.Idtsc) to 0 and 1 respectively.
- The idle timer can be triggered by the transmitter when sending a start bit (also after resend request) or by writing 1 to Stidt bit of global control1 register bit (GLOBALCONTROL1.Stidt).
- Each timer the idle timer is triggered, the start value configured by Idtreg bits of idle timer register (IDLETIMER.Idtreg -1) is reloaded to the down counter, and idle timer starts running.
- The idle timer expired interrupt can be cleared by restarting the idle timer by writing a data to transmit data register (DATA) or by writing 1 to GLOBALCONTROL1.Stidt.

Figure 4-8 shows the timing of idle timer expired interrupt flag (IRQ_STATUS.IdtexpIrq) when idle timer is clocked by card clock and triggered by software (writing 1 to GLOBALCONTROL1.Stidt).

Figure 4-8 Idle Timer Expired Interrupt Flag (IRQ_STATUS.IdtexpIrq) Set Timing when Idle Timer is Triggered by Software and Clocked by Card Clock

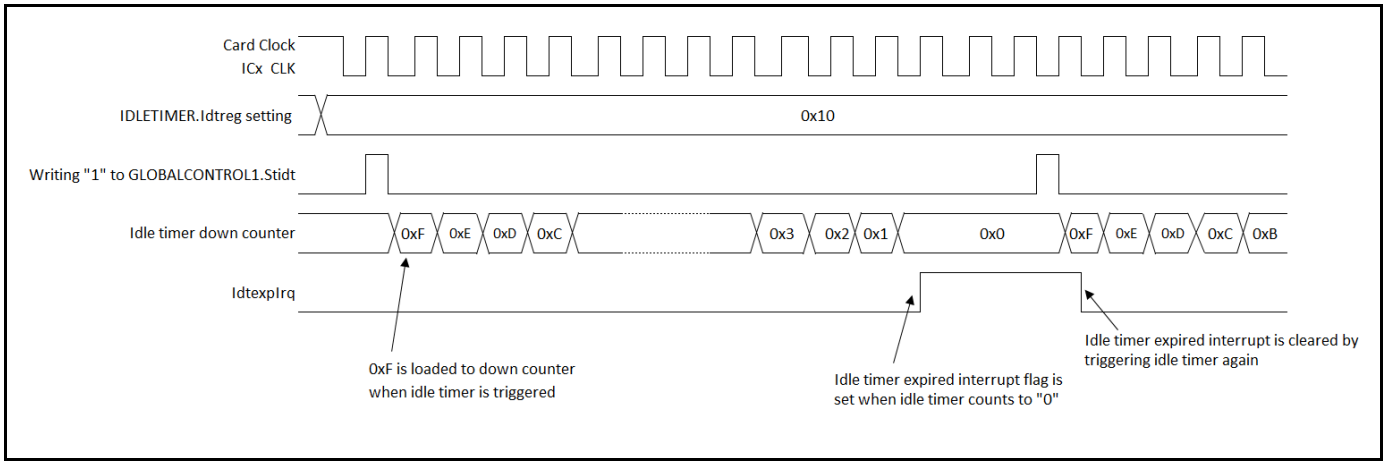
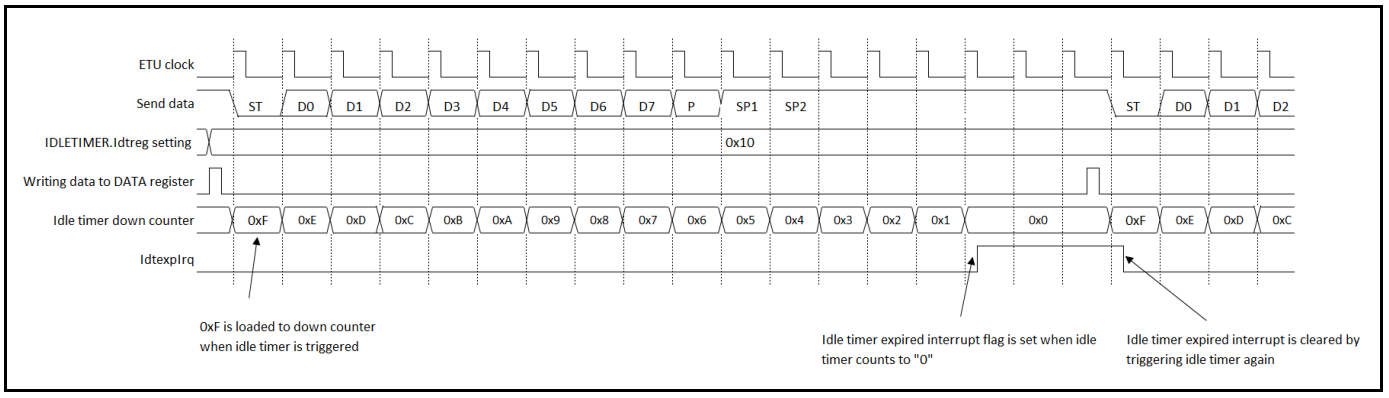


Figure 4-9 shows the timing of idle timer expired interrupt flag (IRQ_STATUS.IdtexpIrq) when idle timer is clocked by ETU clock and triggered by sending a start bit.

Figure 4-9 Idle Timer Expired Interrupt Flag (IRQ_STATUS.IdtexpIrq) Set Timing when Idle Timer is Triggered by Sending a Start Bit and Clocked by ETU Clock



5. Smart Card Interface Setting Procedure and Program Flow

MCU to Smart Card Connection

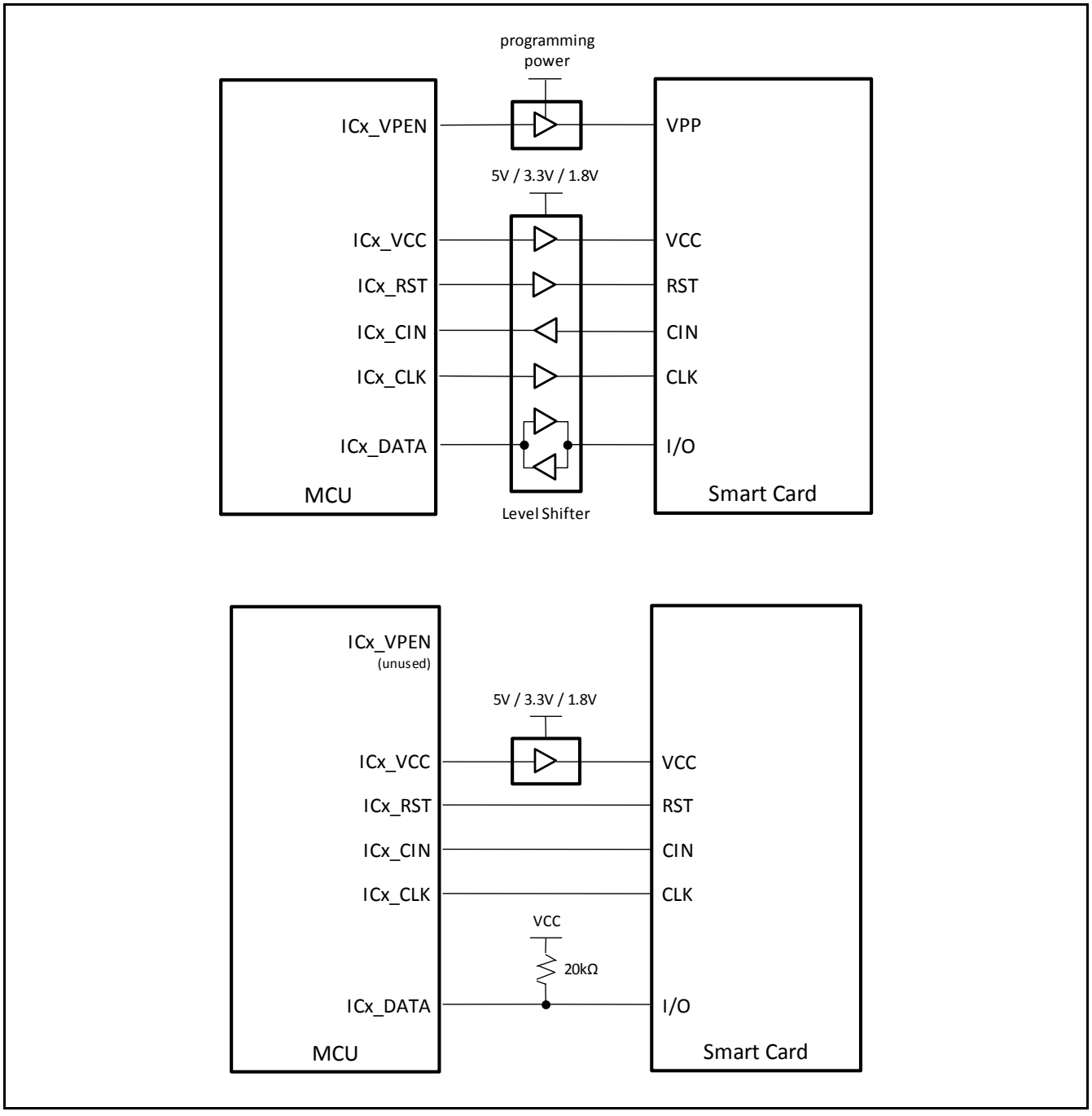
Illustrate connection MPU to smart card as shown in Figure 5-1.

Depends on MCU IO level and Smart Card (Class-A / Class-B / Class-C) level shifter is needed.

If Smart Card supports ISO 7816-3 2006 and later version, programming power outsource is not necessary.

MCU ICx_VCC cannot drive maximum current of Smart Card. Therefore, outer current source is necessary.

Figure 5-1 A Connection Example of Half-Duplex Communication

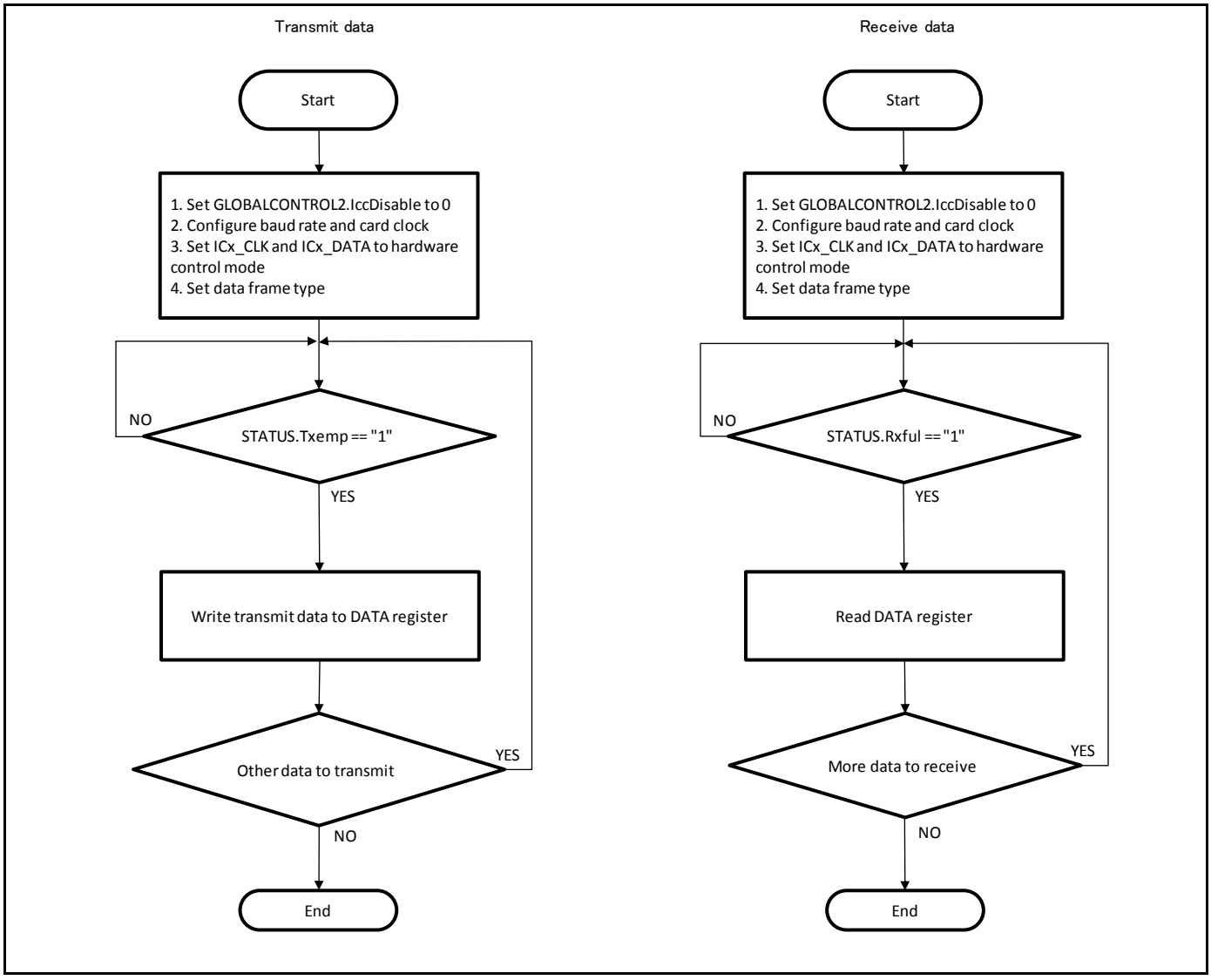


Flowcharts

■ If FIFO is not used

Figure 5-2 shows the flow chart of data transmission when FIFO is not used, and ICx_DATA/ICx_CLK are controlled by hardware.

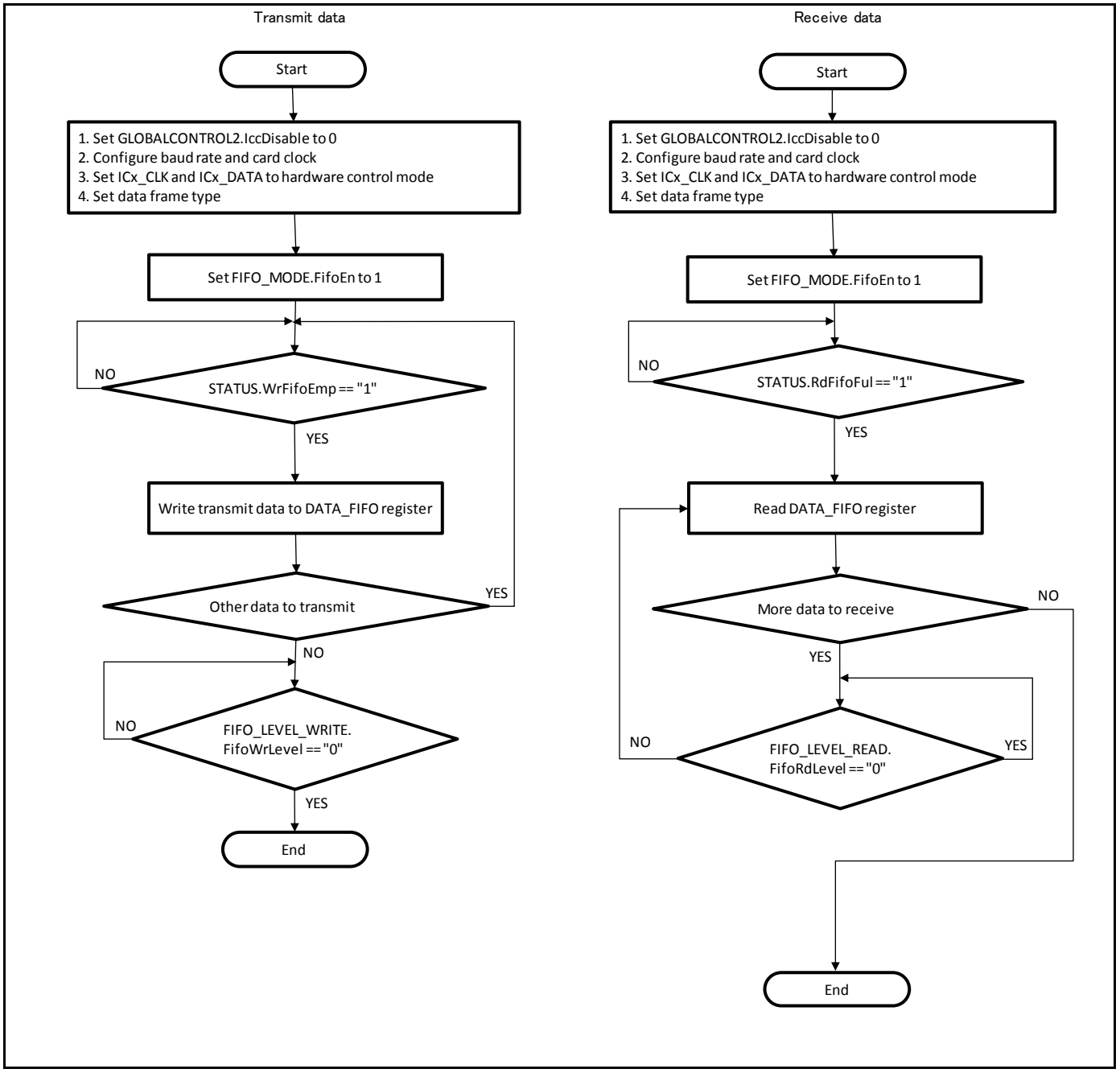
Figure 5-2 An Example of Half-Duplex Communication Flowchart (if FIFO is Not Used)



■ If FIFO is used

Figure 5-3 shows the flow chart of data transmission when FIFO is used, and ICx_DATA/ICx_CLK are controlled by hardware.

Figure 5-3 An Example of Half-Duplex Communication Flowchart (if FIFO is Used)



6. Smart Card Interface Registers

This section provides a list of smart card interface registers.

Table 6-1 List of Smart Card Interface Registers

Abbreviation	Register name	Reference
GLOBALCONTROL1	Global Control Register 1	6.1
STATUS	Status Register	6.2
PORTCONROL	Port Control Register	0
DATA	Data Register	6.4
CARDCLOCK	Card Clock Frequency Register	6.5
BAUDRATE	Baud Rate Register	6.6
GUARDTIMER	Guard Timer Register	6.7
IDLETIMER	Idle Timer Register	6.8
GLOBALCONTROL2	Global Control Register 2	6.9
DATA_FIFO	FIFO Access Register	0
FIFO_LEVEL_READ	Read FIFO Level Register	6.11
FIFO_LEVEL_WRITE	Write FIFO Level Register	6.12
FIFO_MODE	FIFO Mode Register	6.13
FIFO_CLEAR_MSB_WRITE	Write FIFO Clear Register	6.14
FIFO_CLEAR_MSB_READ	Read FIFO Clear Register	6.15
IRQ_STATUS	Interrupt Status Register	6.16

6.1 Global Control Register 1 (GLOBALCONTROL1)

This register allows the configuration of the smart card interface.
Here the Interrupts are enabled and the protocol is selected.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	-	Idtsc	Stidt	Guaen	Resnd	Ckmod	Iomod	Maskitexp
Attribute	-	R/W	W	R/W	R/W	R/W	R/W	R/W
Initial value	-	0	0	0	1	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Maskcaevent	Masksti	Masktxemp	Maskrxful	Mode8n1	Frm1	Frm0	Parity
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] - : Unused bit

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

[bit14] Idtsc: Idle timer clock select bit

This bit selects the clock source for the down counter of idle timer.

Value	Description
0	Idle timer is clocked by card clock (ICx_CLK).
1	Idle timer is clocked by baud clock (ETU clock).

[bit13] Stidt: Start idle timer bit

This bit is used to start the idle timer. Set this bit to 1 to start the idle timer.
Write 0 to this bit has no effect. Always read as 0.

Value	Description
0	No effect.
1	Start the idle timer.

[bit12] Guaen: Guard timer enable bit

This bit enables or disables the guard timer.

Value	Description
0	Disables the guard timer.
1	Enables the guard timer.

[bit11] Resnd: Transmiter and receiver resend function enable bit

Enables or disables transmitter and receiver resend function.
In data transmission, when the resend function is enabled and resend request is detected (parity error happens), the transmitter will resend the present data again.
In data reception, when the resend function is enabled and parity error happens, the receiver will send out resend request and wait to receive the next frame.

Value	Description
0	Disables resend function.
1	Enables resend function.

Notes:

- *When transmit resend happens, the data frame which is requested to resend instead of the data in DATA register is to be sent. The data in DATA register will be sent out till no resend request is detected.*
- *When receive resend happens, the current data frame (with parity error) is not loaded to DATA register, and STATUS.Rxful is not set to 1 when FIFO is not used; when FIFO is used, the data frame with parity error is not pushed to read FIFO.*

[bit10] Ckmod: Clock generation mode select bit

Decide the clock generation method to be software or hardware.
When software is selected, the output level on ICx_CLK depends on the value configured to Clkpt bit of PORTCONTROL register (PORTCONTROL.Clkpt).
When hardware is selected, ICx_CLK is internally generated by UART block.

Value	Description
0	ICx_CLK is generated by hardware.
1	ICx_CLK is generated by software.

[bit9] Iomod: Data generation mode select bit

- Decide the data generation method to be software or hardware.
- When software is selected, the output level on ICx_DATA depends on the value configured to Io1 bit of PORTCONTROL register (PORTCONTROL.Io1).
- When hardware is selected, ICx_DATA is internally generated by UART block.

Value	Description
0	ICx_DATA is generated by hardware.
1	ICx_DATA is generated by software.

Notes:

- *When Iomod is configured to be 1, the output enable for ICx_DATA also needs to be configured by Io1en bit of PORTCONTROL register.*

[bit8] Maskitexp: Idle timer expired interrupt enable bit

Enables or disables the idle timer expired interrupt.

Value	Description
0	Disable idle timer expired interrupt.
1	Enable idle timer expired interrupt.

[bit7] Maskcaevent: Card event detect interrupt enable bit

Enables or disables the interrupt caused by event detected on ICx_CIN input.

Value	Description
0	Disable card event interrupt.
1	Enable card event interrupt.

[bit6] Masksti: Start bit detect interrupt enable bit

Enables or disables the interrupt caused by start bit detected on ICx_DATA in reception mode.

Value	Description
0	Disable start bit detect interrupt.
1	Enable start bit detect interrupt.

[bit5] Masktxemp: Transmit data register empty interrupt enable bit

Enables or disables the interrupt caused by transmit data register empty.

Value	Description
0	Disable transmit data register empty interrupt.
1	Enable transmit data register empty interrupt.

[bit4] Maskrxful: Receive data register rull interrupt enable bit

Enables or disables the interrupt caused by receive data register full.

Value	Description
0	Disable receive data register full interrupt.
1	Enable receive data register full interrupt.

[bit3] Mode8n1: Transmitter 8N2 protocol select bit

This bit selects the 8N2 protocol for transmitter. This bit has no influence on receiver.

When protocol 8E2/8O2 is selected for transmitter, set this bit to 0.

Value	Description
0	Transmitter 8N2 protocol is not selected.
1	Transmitter 8N2 protocol is selected.

[bit2] Frm1: Data frame coding style select bit

Select the data frame coding style to be normal coding or inverse coding.
This bit has no influence on start and stop bit.
When this bit is set to 1, odd parity should be configured (GLOBALCONTROL1.Parity=1).

Value	Description
0	Normal coding (LSB is transmitted first, low level is logical zero).
1	Inverse coding (MSB is transmitted first and inverted, high level is logical zero).

[bit1] Frm0: Received data bit length configure bit

Configure the data bit length to be 8 or 9.
Only effective for receiver. For transmitter, the data bit length is always 8.
When 8E1 or 8O1 protocol is selected, configure this bit to 0. Configure this bit to 1 when 9N1 protocol is selected.

Value	Description
0	8 bit data length.
1	9 bit data length.

[bit0] Parity: Odd/Even parity select bit

Odd/Even parity select bit for data transmission and reception.
For data transmission, the output parity bit will be generated according to this bit's configuration.
For data reception, the parity bit of input data will be checked according to this bit's configuration.

Value	Description
0	Even parity.
1	Odd parity.

6.2 Status Register (STATUS)

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

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved		RxResend	TxResend	RxStartErr	WrFifoEmp	RdFifoFul	RdFifoOvr
Attribute	-		R(*1)	R(*1)	R(*1)	R	R	R
Initial value	-		0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Idtrun	Recofl	CardEvent	CardDetect	Txact	Rxact	Rxful	Txemp
Attribute	R	R	R(*1)	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	1

*1: This bit is read clear.

Register functions

[bit15:14] Reserved: Reserved bits

These bits always read as 0. Write has no effect.

[bit13] RxResend: Receiver resend flag bit

If a parity error occurs during data reception with GLOBALCONTROL1.Resnd =1, smart card interface will send resend request, and this bit is set to 1.
This bit is cleared to 1 when STATUS register is read.

Value	Description
0	No receiver resend occurs.
1	Receiver resend occurs.

[bit12] TxResend: Transmitter resend flag bit

If a resend request is detected while data transmission with GLOBALCONTROL1.Resnd =1, the transmitter will resend the previous data frame again, and this flag bit is set to 1.
This bit is cleared to 1 when STATUS register is read.

Value	Description
0	No transmitter resend occurs.
1	Transmitter resend occurs.

[bit11] RxStartErr: Received start bit error flag bit

This bit indicates whether a wrong start bit is received.

When a wrong start bit is received, the receiver will wait for the next start bit on data line.

Value	Description
0	No start bit is wrong.
1	Received wrong start bit.

[bit10] WrFifoEmp: Write FIFO empty flag bit

This flag shows the state of write FIFO.

When the number of valid data in write FIFO is less than the value configured to FIFO_MODE. WrFifoLevel, this bit is set to 1.

When the number of valid data in write FIFO is equal to or more than the value configured to FIFO_MODE. WrFifoLevel, this bit is cleared to 0.

Value	Description
0	The write FIFO is not empty.
1	The write FIFO is empty.

[bit9] RdFifoFul: Read FIFO full flag bit

This flag shows the state of read FIFO.

When the number of received data in read FIFO is more than the value configured to FIFO_MODE. RdFifoLevel, this bit is set to 1.

When the number of received data in read FIFO is equal to or less than the value configured to FIFO_MODE. RdFifoLevel, this bit is cleared to 0.

Value	Description
0	The read FIFO is not full.
1	The read FIFO is full.

[bit8] RdFifoOvr: Read FIFO overflow flag

This bit indicates whether the read FIFO is overflow.

When there is 16 bytes of data received in read FIFO, and one more data frame is received, the read FIFO is overflow, and this bit is set to 1.

This bit can be cleared to 0 by writing 1 to ClrRdFifo bit of FIFO_CLEAR_MSB_READ register.

Value	Description
0	Read FIFO is not overflow.
1	Read FIFO is overflow.

[bit7] Idtrun: Idle timer running flag

This bit indicates whether idle timer is still running.

When idle timer is started, the down counter of idle timer starts running and this bit is set to 1.

When idle timer is expired, this bit is set to 0 and the down counter stops running.

Value	Description
0	Idle timer has stopped.
1	Idle timer still running.

[bit6] Recofl: Received data register overflow flag

This bit indicates whether the received data register is overflow.

When the received data frame is not read by CPU and another data frame is received, this bit is set to 1.

This bit can be cleared to 0 by reading DATA register.

Value	Description
0	Received data register is not overflow.
1	Received data register is overflow.

[bit5] CardEvent: Card event flag

This bit indicates that a change on the card detect input (ICx_CIN) is detected.

When there is a change of level on the ICx_CIN input this bit is set to 1.

This bit is cleared to 0 when the STATUS register is read.

Value	Description
0	No card event.
1	Card event detected.

[bit4] CardDetect: Level on ICx_CIN input pin

This bit shows the level on ICx_CIN input pin.

Value	Description
0	The level on ICx_CIN pin is low.
1	The level on ICx_CIN pin is high.

[bit3] Txact: Transmitter status flag

This bit shows the operation status of transmitter.

When the serial data transmission is ongoing, the transmitter is active and this bit is set to 1.

When there is no data to be transmitted, the transmitter is idle and this bit is set to 0.

Value	Description
0	Transmitter is idle.
1	Transmitter is active.

[bit2] Rxact: Receiver status flag

This bit shows the operation status of receiver.

When the serial data reception is ongoing, the receiver is active and this bit is set to 1.

When there is no data to be received, the receiver is idle and this bit is set to 0.

Value	Description
0	Receiver is idle.
1	Receiver is active.

[bit1] Rxful: Received data register status flag

This bit indicates the status of received data register.

When a data frame is received, this bit is set to 1.

When the DATA register is read, this bit is set to 0.

Value	Description
0	The received data register is empty.
1	The received data register is full.

[bit0] Txemp: Transmit data register status flag

This bit indicates the status of transmit data register.

When a data frame is written into DATA register, this bit is set to 1.

When there is no data in DATA register, this bit is set to 0.

Value	Description
0	The transmit data register is full.
1	The transmit data register is empty.

6.3 Port Control Register (PORTCONTROL)

The Port Control Register is used to control the status of smart card interface ports.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	VpenOutEn	VccOutEn	RstOutEn	ClkOutEn	Reserved		Vpen	Vccen
Attribute	R/W	R/W	R/W	R/W	-		R/W	R/W
Initial value	0	0	0	0	-		0	0

bit	7	6	5	4	3	2	1	0
Field	Rst	Clkpt	Reserved	Io1en	Reserved	Io1	Reserved	Trimod
Attribute	R/W	R/W	-	R/W	-	R/W	-	R/W
Initial value	0	0	-	0	-	0	-	0

Register functions

[bit15] VpenOutEn: ICx_VPEN output enable bit

This bit enables the output level on ICx_VPEN.

Before data communication, write 1 to this bit to enable output of ICx_VPEN.

The level on ICx_VPEN is not guaranteed when this bit is set to 0.

Value	Description
0	ICx_VPEN output level not guaranteed.
1	ICx_VPEN output enabled.

[bit14] VccOutEn: ICx_VCC output enable bit

This bit enables the output level on ICx_VCC.

Before data communication, write 1 to this bit to enable output of ICx_VCC.

The level on ICx_VCC is not guaranteed when this bit is set to 0.

Value	Description
0	ICx_VCC output level not guaranteed.
1	ICx_VCC output enabled.

[bit13] RstOutEn: ICx_RST output enable bit

This bit enables the output level on ICx_RST.

Before data communication, write 1 to this bit to enable output of ICx_RST.

The level on ICx_RST is not guaranteed when this bit is set to 0.

Value	Description
0	ICx_RST output level not guaranteed.
1	ICx_RST output enabled.

[bit12] ClkOutEn: ICx_CLK output enable bit

This bit enables the output level on ICx_CLK.

Before data communication, write 1 to this bit to enable output of ICx_CLK.

The level on ICx_CLK is not guaranteed when this bit is set to 0.

Value	Description
0	ICx_CLK output level not guaranteed.
1	ICx_CLK output enabled.

[bit11:10] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit9] Vpen: ICx_VPEN output value

Write to this bit to set the output level on ICx_VPEN output.

Value	Description
0	Low level.
1	High level.

[bit8] Vccen: ICx_VCC output value

Write to this bit to set the output level on ICx_VCC output.

Value	Description
0	Low level.
1	High level.

[bit7] Rst: ICx_RST output value

Write to this bit to set the output level on ICx_RST output.

Value	Description
0	Low level.
1	High level.

[bit6] Clkpt: ICx_CLK output value

Write to this bit to set the output level on ICx_CLK when GLOBALCONTROL1.Ckmod = 1.

Value	Description
0	Low level.
1	High level.

[bit5] Reserved: Reserved bit

Always read as 0. Write has no effect.

[bit4] Io1en: ICx_DATA output enable control bit

Write to this bit to enable/disable ICx_DATA output when PORTCONTROL.Trimod = 1.

Value	Description
0	ICx_DATA output enabled.
1	ICx_DATA output disabled.

[bit3] Reserved: Reserved bit

Always read as 0. Write has no effect.

[bit2] Io1: Level on ICx_DATA

Write to this bit to set output level on ICx_DATA pin when GLOBALCONTROL1.Iomod = 1.
This bit shows the level on ICx_DATA when read.

Value	Description
0	Low level.
1	High level.

[bit1] Reserved: Reserved bit

Always read as 0. Write has no effect.

[bit0] Trimod: ICx_DATA output enable generation mode select bit

This bit selects the generation mode of ICx_DATA output enable in data transmission.

Value	Description
0	ICx_DATA output enable is controlled by UART block internally.
1	ICx_DATA output enable is controlled by PORTCONTROL.Io1en.

6.4 Data Register (DATA)

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

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							Data[8]
Attribute	-							R/W
Initial value	-							0

bit	7	6	5	4	3	2	1	0
Field	Data[7:0]							
Attribute	R/W							
Initial value	0x00							

Register functions

[bit15:9] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit8:0] Data: Transmit/Received data

In transmission mode, write to DATA register, and Data[7:0] will be transmitted. Parity is calculated automatically.

A write access to this register causes an immediate start of transmission.

In reception mode of 8-bit frame, Data[7:0] stores the received data byte. Data[8] stores the received parity bit if any.

In reception mode of 9N1, Data[8:0] stores the received data frame.

Transmission Mode	Data Length	Data[8]	Data[7:0]
Data receive	9 bits	Data bits	Data bits
	8 bits (without parity)	Invalid	Data bits
	8 bits (with parity)	Parity bit	Data bits
Data transmit	8 bits	Invalid	Data bits

6.5 Card Clock Frequency Register (CARDCLOCK)

The Card Clock Frequency Register configures the card clock frequency divider based on PCLK. PCLK is divided to generate the card clock output (ICx_CLK).

Register configuration

bit	15	…	0
Field	ClkDivider[15:0]		
Attribute	R/W		
Initial value	0x0028		

Register functions

[bit15:0] ClkDivider: Card clock frequency divider

- Configure even value to this field as card clock frequcy divider.
- When an odd value is configured to this field, the effective divider becomes the odd value + 1.

6.6 Baud Rate Register (BAUDRATE)

The Baud Rate Register allows the adjustment of the baud rate. The reference value for the baud rate calculation is the card clock frequency. For example, to achieve the relation of F/D = 31, 0x1F has to be programmed to BAUDRATE.Brreg.

Register configuration

bit	15	14	…	0
Field	LittleStep	Brreg[14:0]		
Attribute	R/W	R/W		
Initial value	0	0x0174		

Register functions

[bit15] LittleStep: Little step bit for baud rate

By activating this bit (‘1’), the baud clock generation will add a +0.5 card clock step. The Baud rate can be adjusted more accurate that way.

Value	Description
0	Disable little step function.
1	Enable little step function.

[bit14:0] Brreg: Baud rate register bits

These bits configure the reload value for down counter of the baud rate generator.

6.7 Guard Timer Register (GUARDTIMER)

The guard timer is activated by setting GLOBALCONTROL1.Guaen to 1. The UART transmitter waits for GUARDTIMER.Gtreg ETUs before transmitting the next character. The guard timer is activated by any sent or received start bit. The receiver is not affected by the guard timer.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Gtreg[7:0]							
Attribute	R/W							
Initial value	0x00							

Register functions

[bit15:8] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit7:0] Gtreg: Guard time in ETUs

This bits configures the length of a single transmission counted from the start bit in ETUs.

6.8 Idle Timer Register (IDLETIMER)

The idle timer is an independent down counter which can be clocked by the card clock (IC1_CLK) or the ETU clock from the baud rate generator (see register bit: GLOBALCONTROL1.Idtsc). The start value is reloaded each time the transmitter sends a start bit or by setting GLOBALCONTROL1.Stidt to 1. When the idle timer is started, (Idtreg - 1) is programmed into the down counter.

Register configuration

bit	15	…	0
Field	Idtreg[15:0]		
Attribute	R/W		
Initial value	0x0000		

Register functions

[bit15:0] Idtreg: Reload value for idle timer

These bits configure the reload value for down counter of idle timer. Idtreg -1 is programmed into idle timer.

6.9 Global Control Register 2 (GLOBALCONTROL2)

The Global Control Register 2 (GLOBALCONTROL2) is used to enable/disable smart card interface and to configure protocol for data frame.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved				IccDisable	Reserved	InvDataOut	Rx8n1
Attribute	-				R/W	-	R/W	R/W
Initial value	0				1	0	0	0

Register functions

[bit15:4] Reserved: Reserved bits

The read values are all 0. Writes have no effects.

[bit3] IccDisable: Smart card interface disable/enalbe bit

This bit disables or enables smart card interface control block.

Data transmission or reception can only start when smart card interface is enabled.

Don’t set this bit to 1 during data transmission.

Value	Description
0	Enable smart card interface.
1	Disable smart card interface.

[bit2] Reserved: Reserved bit

The read value is 0. Writes have no effects.

[bit1] InvDataOut: Output inversion enable bit

This bit disables or enables the inversion of level on ICx_DATA during data transmission.

Only output data can be inversed. Start bit and stop bit are also inversed.

This bit is only effective when output data is generated by hardware (GLOBALCONTROL1.Iomod=0).

Value	Description
0	Disable inversion of ICx_DATA.
1	Enable inversion of ICx_DATA.

[bit0] Rx8n1: Receiver 8N1/8N2 protocol select bit

This bit selects 8N1/8N2 protocol for data reception.

Value	Description
0	8N1/8N2 protocol for receiver is not selected.
1	8N1/8N2 protocol for receiver is selected.

6.10 FIFO Access Register (DATA_FIFO)

The smart card interface includes two 16-byte FIFOs for data transmission and reception separately. The FIFOs may be configured to generate an IRQ when they reach a particular level. The FIFOs may also be flushed if desired.

The FIFO Access Register (DATA_FIFO) is used to read/write FIFOs. Received data stored in read FIFO can be read out by reading this register. Data write to this register is sent out. A write to this register will cause immediate data transmission.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							Data[8]
Attribute	-							R/W
Initial value	-							0

bit	7	6	5	4	3	2	1	0
Field	Data[7:0]							
Attribute	R/W							
Initial value	0x00							

Register functions

[bit15:9] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit8:0] Data: Transmit/Received data

Transmission Mode	Data Length	Data[8]	Data[7:0]
Data receive	9 bits	Data bits	Data bits
	8 bits (without parity)	Invalid	Data bits
	8 bits (with parity)	Parity bit	Data bits
Data transmit	8 bits	Invalid	Data bits

6.11 Read FIFO Level Register (FIFO_LEVEL_READ)

The Read FIFO Level Register (FIFO_LEVEL_READ) shows the number of data frame in read FIFO.

Register configuration

bit	15	…	0
Field	FifoRdLevel[15:0]		
Attribute	R		
Initial value	0x0000		

Register functions

[bit15:0] FifoRdLevel: Read FIFO level

Each bit represents a single FIFO slot. When the bit is '1' the FIFO slot is used, otherwise it is unused.

6.12 Write FIFO Level Register (FIFO_LEVEL_WRITE)

The Write FIFO Level Register (FIFO_LEVEL_ WRITE) shows the number of data frame in write FIFO.

Register configuration

bit	15	…	0
Field	FifoWrLevel[15:0]		
Attribute	R		
Initial value	0x0000		

Register functions

[bit15:0] FifoRdLevel: Read FIFO level

Each bit represents a single FIFO slot. When the bit is '1' the FIFO slot is used, otherwise it is unused.

6.13 FIFO Mode Register (FIFO_MODE)

The FIFO Mode Register (FIFO_MODE) sets FIFO levels for interrupt generation or status setting. It also enables/disables FIFOs and interrupts related with FIFO.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	RdFifoLevel[3:0]				WrFifoLevel[3:0]			
Attribute	R/W				R/W			
Initial value	0000				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved				RdFifoIrqEn	WrFifoIrqEn	RdFifoOvrIrqEn	FifoEn
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				0	0	0	0

Register functions

[bit15:12] RdFifoLevel: Read FIFO level

These bits set read FIFO level for generation of read FIFO full interrupt or status flag.
Interrupt can be generated when at least RdFifoLevel + 1 data frame is stored in read FIFO.

[bit11:8] WrFifoLevel: Write FIFO level

These bits set write FIFO level for generation of write FIFO empty interrupt or status flag.
Interrupt can be generated when data frame in write FIFO is less than WrFifoLevel.

[bit7:4] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit3] RdFifoIrqEn: Read FIFO full interrupt enable bit

This bit enables/disables the generation of read FIFO full interrupt.

Value	Description
0	Disable read FIFO full interrupt.
1	Enable read FIFO full interrupt.

[bit2] WrFifoIrqEn: Write FIFO empty interrupt enable bit

This bit enables/disables the generation of write FIFO empty interrupt.

Value	Description
0	Disable write FIFO empty interrupt.
1	Enable write FIFO empty interrupt.

[bit1] RdFifoOvrIrqEn: Read FIFO overflow interrupt enable bit

This bit enables/disables the generation of read FIFO overflow interrupt.

Value	Description
0	Disable read FIFO overflow interrupt.
1	Enable read FIFO overflow interrupt.

[bit0] FifoEn: FIFO enable bit

This bit enables/disables read/write FIFOs.

Value	Description
0	Disable read and write FIFOs.
1	Enable read and write FIFOs.

6.14 Write FIFO Clear Register (FIFO_CLEAR_MSB_WRITE)

Data in write FIFO can be cleared by writing to this register.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved							ClrWrFifo
Attribute	-							R/W
Initial value	-							0

Register functions

[bit15:1] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit0] ClrWrFifo: Write FIFO clear bit

By writing 1 to this bit, data in write FIFO is flushed and write FIFO becomes totally empty.

Value	Description
0	Do not flush write FIFO.
1	Flush write FIFO.

6.15 Read FIFO Clear Register (FIFO_CLEAR_MSB_READ)

Data in read FIFO can be cleared by writing to this register.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved							ClrRdFifo
Attribute	-							R/W
Initial value	-							0

Register functions

[bit15:1] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit0] ClrRdFifo: Read FIFO clear bit

By writing 1 to this bit, data in read FIFO is flushed and read FIFO becomes totally empty.

Value	Description
0	Do not flush read FIFO.
1	Flush read FIFO.

6.16 Interrupt Status Register (IRQ_STATUS)

This register allows reading out the interrupt status of the smart card interface. The software can use this register to check which event has caused the interrupt.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	RxfullIrq	TxempIrq	RxstbiIrq	CardEventIrq	IdtexpIrq	RdFifoIrq	WrFifoIrq	RdFifoOvrIrq
Attribute	R	R	R(*1)	R(*1)	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

*1: This bit is read clear.

Register functions

[bit15:8] Reserved: Reserved bits

Always read as 0. Write has no effect.

[bit7] RxfullIrq: Received data register full interrupt flag bit

This bit indicates the status of interrupt caused by received data register full.

When one data frame is received by DATA register with GLOBALCONTROL1.Maskrxful = 1, this bit is set to 1.

This bit is cleared to 0 by reading the DATA register.

Value	Description
0	No received data register full interrupt.
1	Received data register full interrupt occurs.

[bit6] TxempIrq: Transmit data register empty interrupt flag bit

This bit indicates the status of interrupt caused by transmit data register empty.

When there is no data in DATA register in transmission mode with GLOBALCONTROL1.Masktxemp = 1, this bit is set to 1.

This bit is cleared to 0 by writing to DATA register.

Value	Description
0	No transmit data register empty interrupt.
1	Transmit data register empty interrupt occurs.

[bit5] RxstbiIrq: Received start bit interrupt flag bit

This bit indicates the status of interrupt caused by receiver detected a start bit on ICx_DATA pin.
When a start bit is detected by receiver with GLOBALCONTROL1.Masksti = 1, this bit is set to 1.
This bit is cleared to 0 by reading the IRQ_STATUS register.

Value	Description
0	No received start bit interrupt.
1	Received start bit interrupt occurs.

[bit4] CardEventIrq: Card event interrupt flag bit

This bit indicates the status of interrupt caused by detecting of a level change on ICx_CIN input.
When a level change on ICx_CIN input is detected with GLOBALCONTROL1.Maskcaevent = 1, this bit is set to 1.
This bit is cleared to 0 by reading the IRQ_STATUS register.

Value	Description
0	No card event interrupt occurs.
1	Card event interrupt occurs.

[bit3] IdtexpIrq: Idle timer expired interrupt flag bit

This bit indicates the status of interrupt caused by idle timer expired.
When idle timer is expired with GLOBALCONTROL1.Maskitexp = 1, this bit is set to 1.
This bit is cleared to 0 by restarting the idle timer.

Value	Description
0	No idle timer expired interrupt.
1	Idle timer expired interrupt occurs.

[bit2] RdFifoIrq: Read FIFO full interrupt flag bit

This bit indicates the status of interrupt caused by read FIFO full.
When the number of data frame in read FIFO is more than the value configured to FIFO_MODE.RdFifoLevel with FIFO_MODE.RdFifoIrqEn= 1, this bit is set to 1.
When the number of data frame in read FIFO is equal to or less than the value configured to FIFO_MODE.RdFifoLevel, this bit is cleared to 0.

Value	Description
0	No read FIFO full interrupt.
1	Read FIFO full interrupt occurs.

[bit1] WrFifoIrq: Write FIFO empty interrupt flag bit

This bit indicates the status of interrupt caused by write FIFO empty.
When the number of data frame in write FIFO is less than the value configured to FIFO_MODE.WrFifoLevel with FIFO_MODE.WrFifoIrqEn= 1, this bit is set to 1.
When the number of data frame in write FIFO is equal to or more than the value configured to FIFO_MODE.WrFifoLevel, this bit is cleared to 0.

Value	Description
0	No write FIFO empty interrupt.
1	Write FIFO empty interrupt occurs.

[bit0] RdFifoOvrIrq: Read FIFO overflow interrupt flag bit

This bit indicates the status of interrupt caused by read FIFO overflow.
When the number of data frame in read FIFO reaches 16 and another data frame is received with FIFO_MODE.RdFifoOvr = 1, this bit is set to 1.
This bit is cleared to 0 by writing 1 to FIFO_CLEAR_MSB_READ. ClrRdFifo to flush the read FIFO.

Value	Description
0	No read FIFO overflow interrupt.
1	Read FIFO overflow interrupt occurs.

CHAPTER7: I2CSLAVE

This chapter explains the function of I2CSLAVE.

CHAPTER 7:1. Overview

CHAPTER 7:2. Configuration

CHAPTER 7:3. Operation

CHAPTER 7:4. Registers

CODE: FIP024-E01.3

1. Overview

I2CSLAVE (I2C Slave Function With Wake-Up) supports the slave function of I2C and Wake-Up function by the match with the slave address.

The features of I2CSLAVE

- The match with the slave address

I2CSLAVE has the I2C Bus 7-bit Slave Address Register (IBSADR). If the value of the I2C Bus Slave Address Register (IBSADR) matches with the received slave address, I2CSLAVE outputs "L" to the I2C data line (SDA) at the timing of acknowledge automatically. In addition, it is possible to disable the detection of the slave address.

- The mask of the slave address

For the slave address bits set to the I2C Bus 7-bit Slave Address Register (IBSADR), the address bits masked by the I2C Bus 7-bit Slave Address Mask Register (IBSMSKR) perform the detection of slave address as a match.

If the received slave address matches with the value of I2C Bus 7-bit Slave Address Register (IBSADR) masked by the I2C Bus 7-bit Slave Mask Register (IBSMSKR), the received slave address is written to the I2C Bus Slave Address Register (IBSADR).

- Supports the reserved addresses

If one of the reserved addresses (0000xxx or 1111xxx) is detected, the interrupt request is generated before the timing of acknowledge, then the received data is read. The received data is the received reserved address. Software judges whether the address is support or not, then I2CSLAVE outputs "L" or "H" to the I2C data line (SDA) at the timing of acknowledge. I2CSLAVE supports the reserved addresses (0000xxx or 1111xxx) how to this.

- Interrupt Request

I2CSLAVE supports transmission interrupt request, reception interrupt request and status interrupt request.

- Automatic Wake-Up

When CPU is the state of the stand-by mode, I2CSLAVE supports the function which wakes up CPU automatically if the slave address matches with received address. Until the interrupt request flag is cleared, I2CSLAVE outputs "L" to the I2C clock line (SCL).

- The speed of I2C Bus

I2CSLAVE supports the Standard-mode (Sm), Fast-mode (Fm) and Fast-mode Plus (Fm+).

I2CSLAVE doesn't support the High-speed mode (Hs-mode) and Ultra Fast-mode (UFm).

- Supports I2C Bus

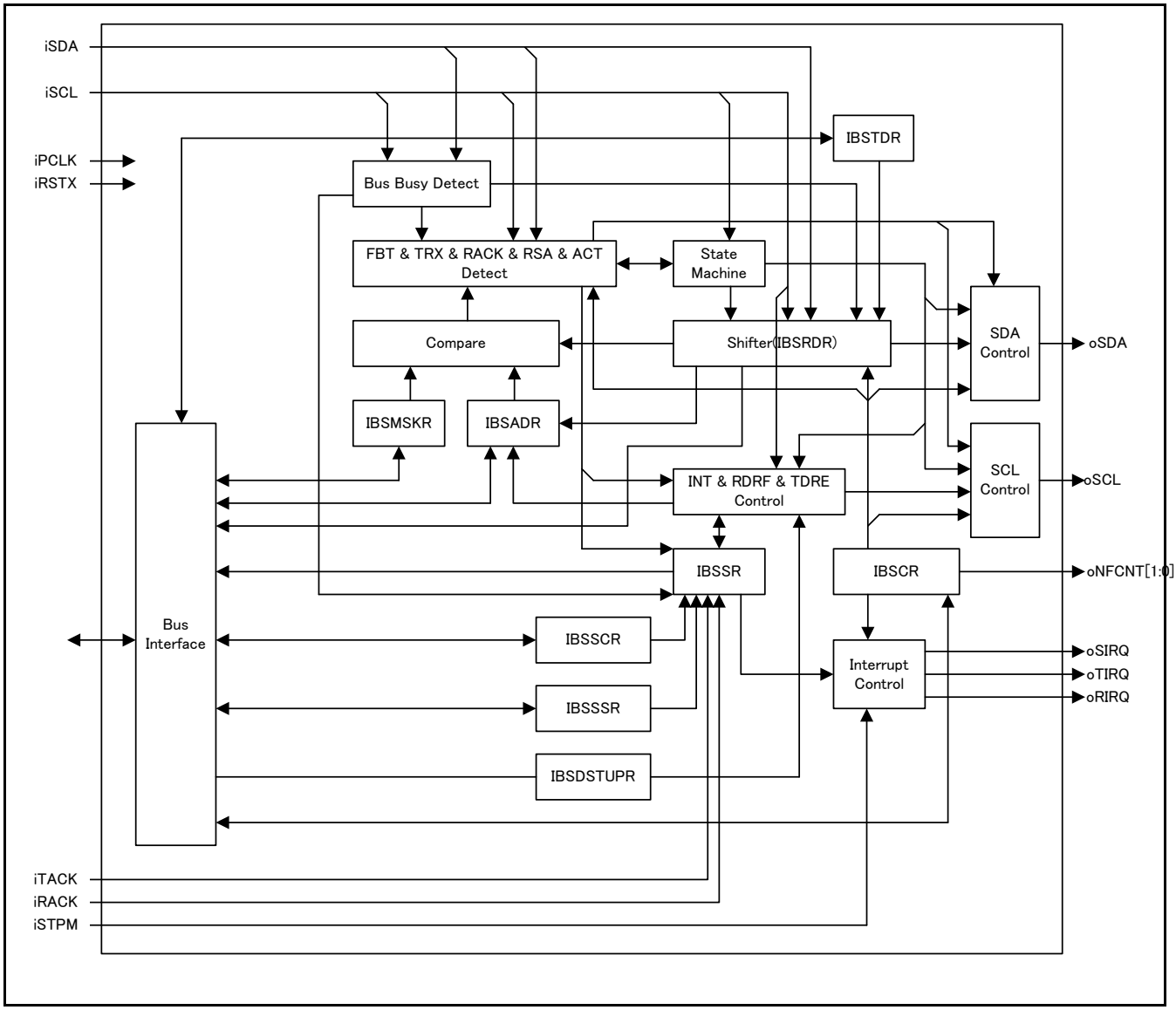
  Slave only. I2CSLAVE doesn't support the following protocol.

  CBUS, SMBus, PMBus, IPMI, ATCA, DDC

2. Configuration

This section shows a configuration of I2CSLAVE.

Figure 2-1 The block diagram of the I2CSLAVE



3. Operation

This section describes the operation of I2CSLAVE.

3.1. Slave Address

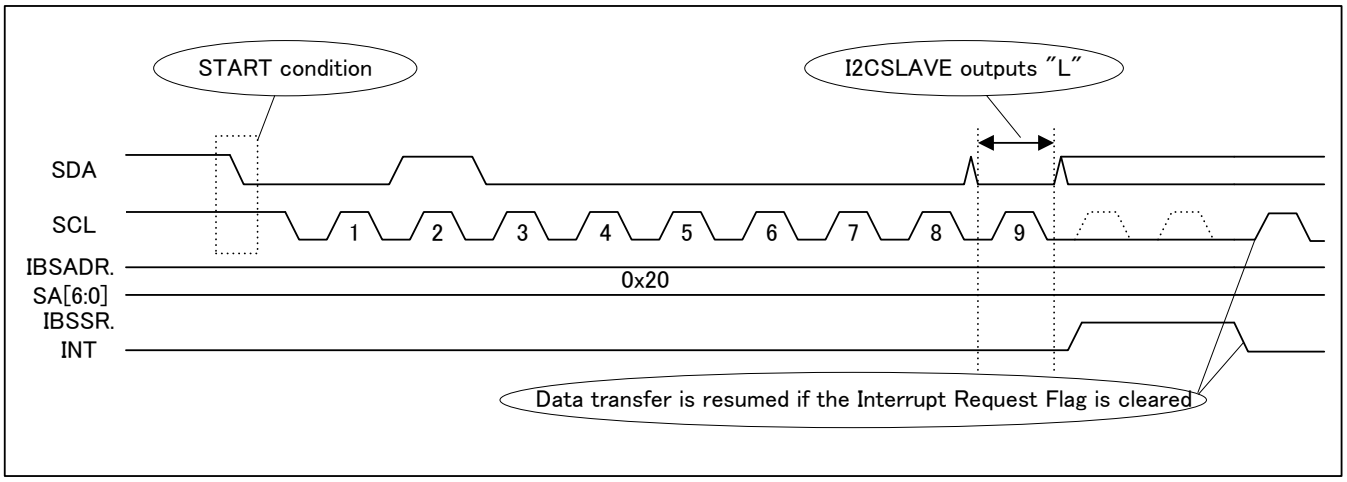
After the detection of START condition, I2CSLAVE operates as follows if the received slave address matched with the slave address set to the I2C Bus 7-bit Slave Address Register (IBSADR) (If IBSADR.SAEN="1").

1) Output "L" to the I2C data line (SDA) at the timing of acknowledge. Then,
2) Output "L" to the I2C clock line (SCL) after the timing of acknowledge. Then,
3) Set "1" to the Status Interrupt Request flag (IBSSR.INT).

The data transfer stops while the I2C clock line (SCL) is "L". If the Status Interrupt Request flag (IBSSR.INT) is cleared to "0" ("1" is written to the IBSSCR.INTC), the clock of I2C is resumed and the transmission or reception of data is started.

If the received slave address didn't match with the slave address set to the I2C Bus 7-bit Slave Address Register (IBSADR), the I2C data line (SDA) is the state of "H" at the timing of acknowledge, the Status Interrupt Request flag (IBSSR.INT) keeps "0" and I2CSLAVE doesn't perform the transmission and reception.

Figure 3-1 The operation for the match of the slave address (Ex. IBSADR.SA[6:0]=0x20)



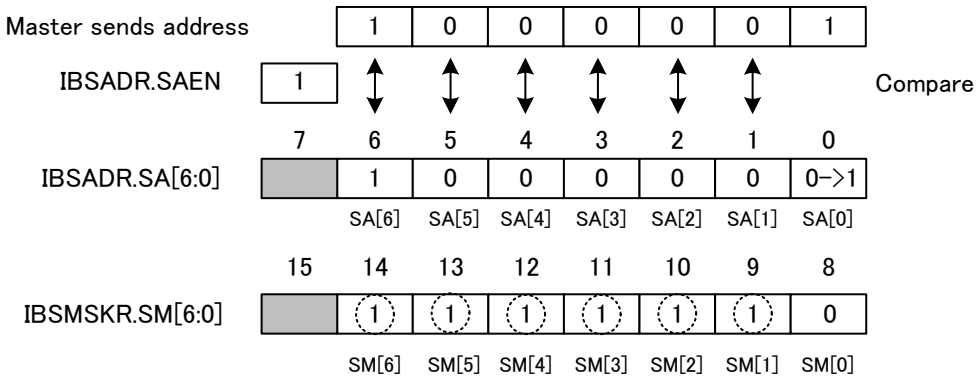
There is the function masking the value of the I2C Bus 7-bit Slave Address Register (IBSADR).

If "0" is set to any of the SM[6:0] bits of the I2C Bus 7-bit Slave Address Mask Register (IBSMSKR), the address bits are masked , the bits tread as matches and the received data is written to the SA[6:0] bits of the I2C Bus 7-bit Slave Address Register (IBSADR) after the timing of acknowledge.

Ex) In case of IBSADR.SA[6:0]=0x40, IBSMSKR.SM[6:0]=0x7E :

In case that the received slave address is 0x41, the slave address is matched as IBSMSKR.SM[0] is "0", then IBSADR.SA[6:0] bits are updated to 0x41.

Figure 3-2 Example of slave address detection



3.2. Reserved Address

If one of the reserved addresses (0000xxx or 1111xxx) is detected after the reserved address detection is enabled (IBSCR.RSVEN="1"), "1" is set to the Status Interrupt Request flag (IBSSR.INT), Reception Interrupt Request flag (IBSSR.RDRF) and Reserved Address Detection flag (IBSSR.RSA) before the timing of acknowledge, and I2CSLAVE outputs "L" to the I2C clock line (SCL). If the Reserved Address Detection flag (IBSSR.RSA) is "1" and the Status Interrupt Request flag (IBSSR.INT) is "1", the received address and direction (transmission or reception) is confirmed by reading the received data (IBSRDR). If it should be the corresponding address, the Status Interrupt Request flag (IBSSR.INT) is cleared to "0" after the Wait Select bit (IBSCR.WSEL) and Data Byte Acknowledge Enable bit (IBSCR.ACKE) are written. If the process is changed by the received data of second byte, the Status Interrupt Request flag (IBSSR.INT) should be cleared to "0" after "1" is set to the Wait Select bit (IBSCR.WSEL).

When the next byte is received after the detection of the reserved address, I2CSLAVE outputs "L" to the I2C clock line (SCL), and "1" is set to the Status Interrupt Request flag (IBSSR.INT) and Reception Interrupt Request flag (IBSSR.RDRF) before the timing of acknowledge if the Wait Select bit (IBSCR.WSEL) is "1". If it continues to receive data by the data after the data of the second byte is read, the Acknowledge Enable bit (IBSCR.ACKE) is "1". If not, the Data Byte Acknowledge Enable bit (IBSCR.ACKE) is "0". And the Status Interrupt Request flag (IBSSR.INT) is cleared. Refer to Figure 3-4.

Figure 3-3 The Interrupt request timing of the reserved address
(Ex. General Call Address, IBSCR.WSEL="0")

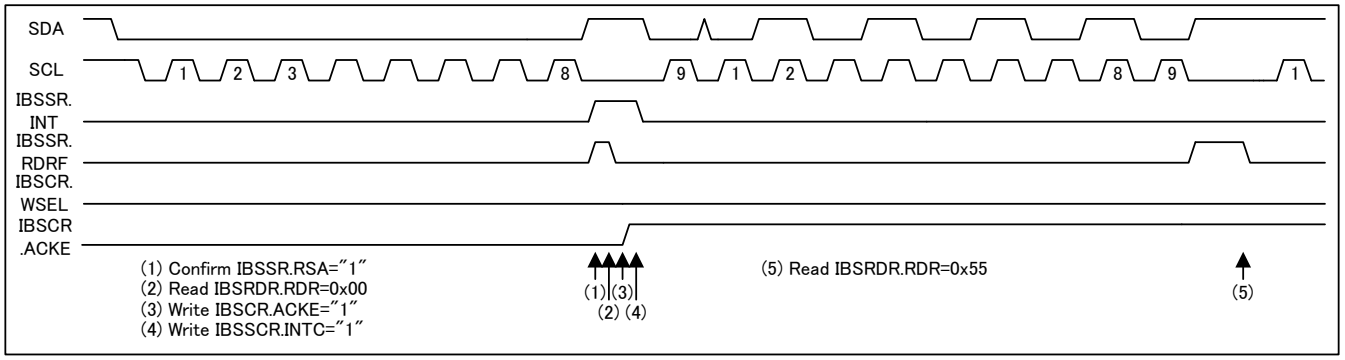
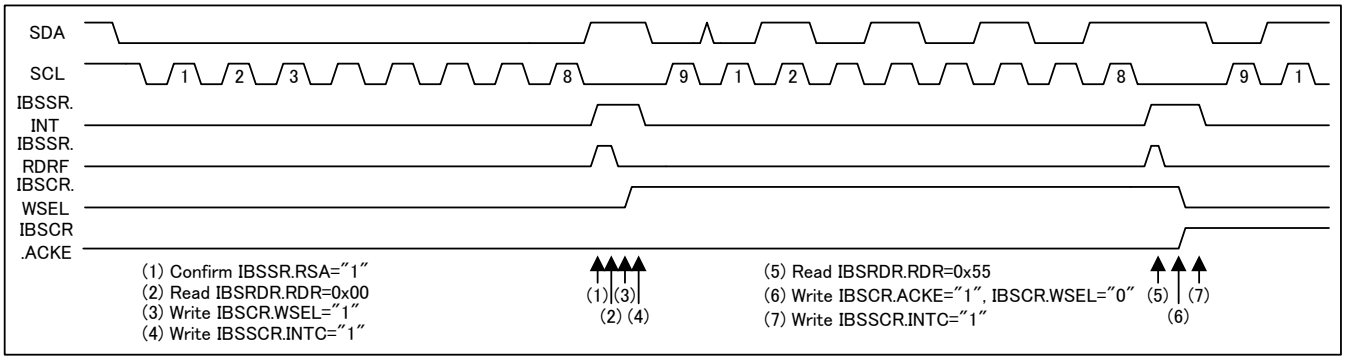


Figure 3-4 The Interrupt request timing of the reserved address
(Ex. General Call Address, IBSCR.WSEL="1")



Note:

- *As I2CSLAVE doesn't support the "START byte", "CBUS address", "reserved" and "Hs-mode master code" among the reserved addresses, I2CSLAVE must respond by "Not Acknowledge (NACK)" if they are received.*

3.3. Transmission and Reception of data

I2CSLAVE displays the Data Direction flag (IBSSR.TRX) by detecting the direction of transmission or reception when the slave address is received. If the slave address is matched and "1" is set to the Status Interrupt Request flag (IBSSR.INT), the Data Direction flag (IBSSR.TRX) is read. If this flag is "0", the direction is the reception. If "1", the direction is the transmission. The data process is operated by this flag.

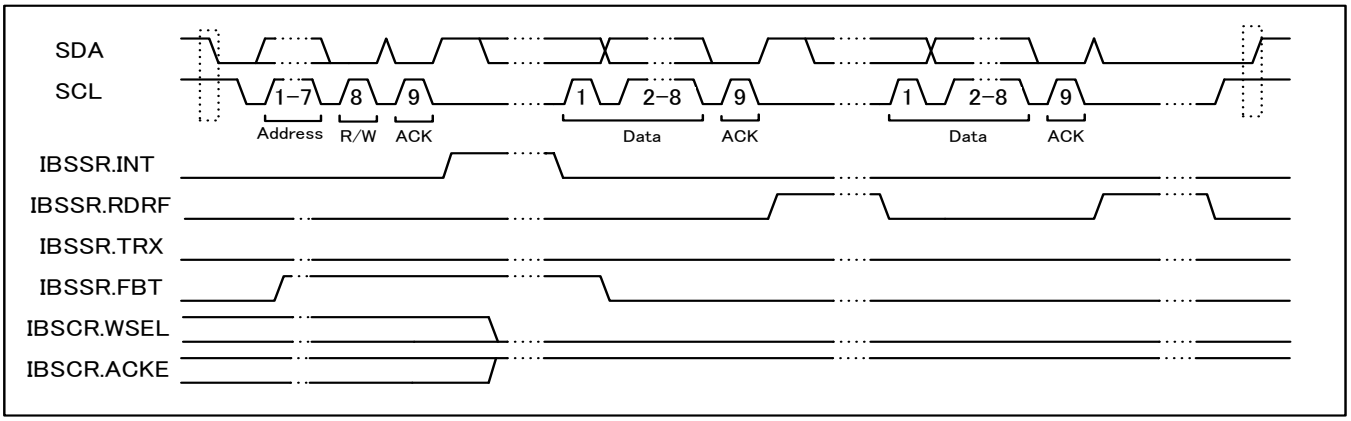
3.3.1. Reception of data

When the Status Interrupt Request flag (IBSSR.INT) by matching with the slave address is set to "1", the steps are performed as follows.

1) Write "0" to the Wait Select bit (IBSCR.WSEL), and write "1" to the Data Byte Acknowledge Enable bit (IBSCR.ACKE). Then,
2) Clear the Status Interrupt Request flag (IBSSR.INT).
(Write "1" to the Status Interrupt Request Flag Clear bit (IBSSCR.INTC))

The reception of data is started by these steps. After that, I2CSLAVE outputs "L" to the I2C data line (SDA) at the timing of acknowledge, and after the timing of acknowledge, I2CSLAVE outputs "L" to the I2C clock line (SCL), outputs "H" to the I2C data line (SDA) and "1" is set to the Reception Interrupt Request flag (IBSSR.RDRF). The received data in the I2C Bus Slave Reception Data Register (IBSRDR) is read, the Reception Interrupt Request flag (IBSSR.RDRF) is clear to "0" automatically, the I2C clock line (SCL) is released from "L" and the next reception of data is started. It continues until the STOP Condition or Re-START Condition is detected.

Figure 3-5 Reception of data



3.3.2. Transmission of data

When the Status Interrupt Request flag (IBSSR.INT) by matching with the slave address is set to "1", the steps are performed as follows.

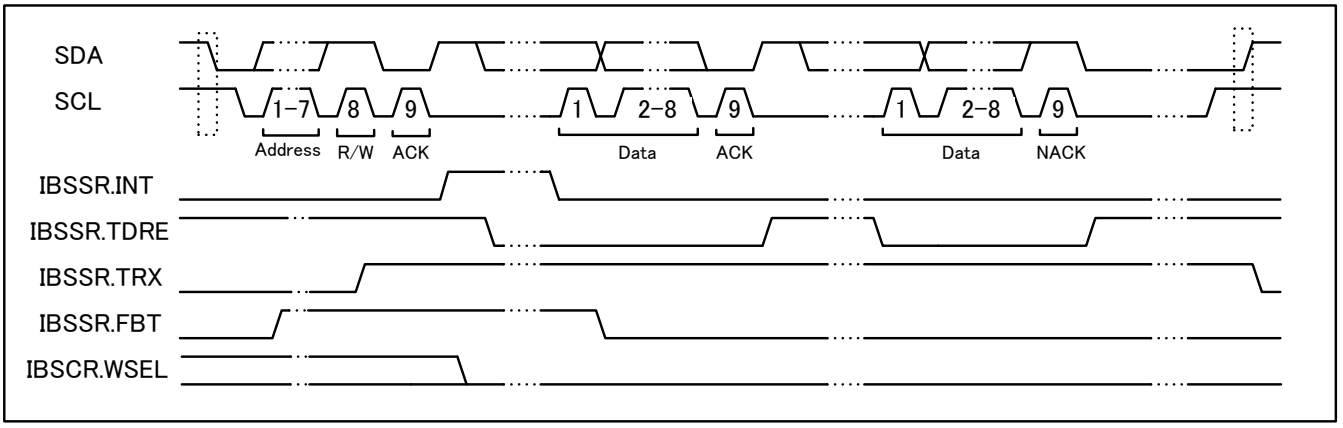
1) Write "0" to the Wait Select bit (IBSCR.WSEL), and write "1" to the Data Byte Acknowledge Enable bit (IBSCR.ACKE). Then,
2) Write the transmission data to the I2C Bus Slave Transmission Data Register (IBSTDR). Then,
3) Clear the Status Interrupt Request flag (IBSSR.INT).
(Write "1" to the Status Interrupt Request Flag Clear bit (IBSSCR.INTC))

The transmission of data is started by these steps. After that, I2CSLAVE outputs "L" to the I2C clock line (SCL) after the timing of acknowledge, "1" is set to the Transmission Data Empty flag (IBSSR.TDRE). The Acknowledge flag (IBSSR.RACK) is checked, and the one of the following steps should be performed by this flag.

- If the Acknowledge flag (IBSSR.RACK) is "0", the transmission data is written to the I2C Bus Slave Transmission Data Register (IBSTDR). Or
- If the Acknowledge flag (IBSSR.RACK) is "1", "0" is written to the Transmission Interrupt Request Enable bit (IBSCR.TIE) and the Status Interrupt Request flag (IBSSR.INT) is cleared to "0".

The transmission continues until I2CSLAVE detects "H" to the I2C data line (SDA) at the timing of acknowledge (IBSSR.RACK="1"). If the I2C master outputs "H" to the I2C data line (SDA), next, the I2C master generates the STOP Condition or Re-START Condition, and the transmission is completed.

Figure 3-6 Transmission of data



3.4. Wake-Up

When CPU is the state of stand-by mode, the operating clock (iPCLK) is stopped and I2CSLAVE is the state of low power mode. When the Status Interrupt Request Enable bit (IBSCR.INTE) is "1", the interrupt request is generated in one of the following conditions and CPU changes to the state of operation. I2CSLAVE outputs "L" to the I2C clock line (SCL) until the Status Interrupt Request flag (IBSSR.INT) is cleared to "0". If the Status Interrupt Request flag (IBSSR.INT) is cleared to "0", the transmission or reception is started.

- When the Slave Address Enable bit (IBSADR.SAEN) is "1", the received slave address is matched. Or
- When the Reserved Address Enable bit (IBSCR.RSVEN) is "1", the received slave address is one of the reserved addresses (0000xxx or 1111xxx).

3.5. Interrupt Request

The conditions generating the interrupt request show Table 3-1.

Table 3-1 Conditions generating the interrupt request

Interrupt Request	Interrupt Request flag	Interrupt Request Enable	Interrupt Request is cleared if	Interrupt Request is set if
Status Interrupt Request flag	IBSSR.INT="1"	IBSCR.INTE="1"	Write "1" to the IBSSCR.INTC	Refer to 4.1
Re-Start Condition Detection flag	IBSSR.RSC="1"	IBSCR.CNDE="1"	Write "1" to the IBSSCR.RSCC	Detect the START Condition when IBSSR.ACT="1"
Stop Condition Detection flag	IBSSR.SPC="1"		Write "1" to the IBSSCR.SPCC	Detect the STOP Condition when IBSSR.ACT="1"
Reception Interrupt Request flag	IBSSR.RDRF="1"	IBSCR.RIE="1"	Read the IBSRDR	Receive a data or Receive a received address
Transmission Data Empty flag	IBSSR.TDRE="1"	IBSCR.TIE="1"	Write to the IBSTDR	Write "1" to the IBSSSR.TDRES, or IBSTDR and Shifter (IBSSR.TRX="1") are empty

IBSSR：I2C Bus Slave Status Register

IBSCR：I2C Bus Slave Control Register

IBSSCR：I2C Bus Slave Status Clear Register

IBSSSR：I2C Bus Slave Status Set Register

IBSTDR：I2C Bus Slave Transmission Data Register

IBSRDR：I2C Bus Slave Reception Data Register

3.6. Data setup time

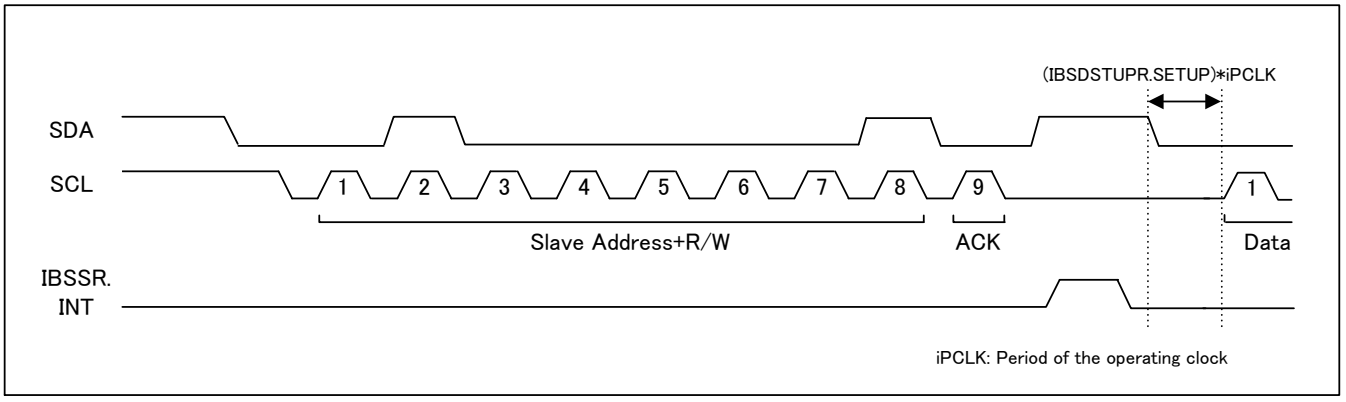
The Data setup time is time from SDA to SCL rising edge. The data setup time can be set by IBSDSTUPR.SETUP[7:0].

The data setup time is controlled by any following conditions.

- The Status Interrupt Request after received slave address (direction of transmission) is cleared.
- IBSCR.ACKE bit is set to "1" at the Status Interrupt Request after received reserved address (direction of reception), and the Status Interrupt Request is cleared.
- IBSCR.ACKE bit is set to "1" at the Status Interrupt Request after received reserved address (direction of transmission), and the Status Interrupt Request is cleared.
- IBSCR.ACKE bit is set to "1" at Reception Interrupt Request after received data, and the Reception Interrupt Request is cleared.
- Transmission Interrupt Request after transmission data and reception acknowledge (IBSCR.WSEL="0") is cleared.

For example, Figure 3-7 shows the data setup time if the Status Interrupt Request after received slave address (direction of transmission) is cleared.

Figure 3-7 The data setup time if the Status Interrupt Request after received slave address (direction of transmission) is cleared



3.7. Example for the operation steps

The section describes the example for the operation steps.

- Initialization
  1) Setting the IBSCR
     Write NFCNT="00", RSVEN="0" or "1", ACKE="1", WSEL="0", CNDE="0", INTE="1", RIE="1", TIE="0".
  2) Setting the IBSDSTUPR
     Write the data setup time between the I2C data line (SDA) and I2C clock line (SCL).
  3) Setting the IBSADR
     Write SAEN="1", SA[6:0]=7-bit Slave Address.
  4) Setting the IBSMSKR
     Write EN="1", SM[6:0]=7-bit Slave Address Mask.

- Status Interrupt Request(The interrupt request is generated by the IBSSR.INT)
  1) If the IBSSR.RSA="0", jump to 2). If the IBSSR.RSA="1", jump to 15).
  2) If the IBSSR.FBT="1", jump to 3). If the IBSSR.FBT="0", jump to 11).
  3) If the IBSSR.TRX="1", jump to 4). If the IBSSR.TRX="0", jump to 8).
  4) Write the transmission data to the IBSTDR.
  5) Write the IBSCR.TIE="1", IBSCR.WSEL="0", IBSCR.ACKE="1".
  6) Write the IBSSCR.INTC="1".
  7) Finished.
  8) Write the IBSCR.RIE="1", IBSCR.WSEL="0", IBSCR.ACKE="1".
  9) Write the IBSSCR.INTC="1".
  10) Finished.
  11) Write the TIE="0".
  12) Write the IBSSSR.TDRES="1".
  13) Write the IBSSCR.INTC="1".
  14) Finished.
  15) If the IBSSR.FBT="1", jump to 16). If the IBSSR.FBT="0", jump to 26).
  16) Read the IBSRDR.
  17) If supporting the reserved address, jump to 18). If not, jump to 23).
  18) If the confirmation of data of second byte is necessary, jump to 19). If not, 20).
  19) Write the IBSCR.WSEL="1".
  20) Write the IBSCR.ACKE="1".
  21) Write the IBSSCR.INTC="1".
  22) Finished.
  23) Write the IBSCR.ACKE="0".
  24) Write the IBSSCR.INTC="1".
  25) Finished.
  26) Read the IBSRDR (data of second byte).
  27) If supporting the data (second address), jump to 28). If not, jump to 31).
  28) Write the IBSCR.WSEL="0", IBSCR.ACKE="1".
  29) Write the IBSSCR.INTC="1".
  30) Finished.
  31) Write the IBSCR.WSEL="0", IBSCR.ACKE="0".

32) Write the IBSSCR.INTC="1".
33) Finished.

If not supporting the reserved addresses, the steps 15) and its subsequent are not necessary.

- Reception Interrupt Request (The interrupt request is generated by the IBSSR.RDRF)
  1) Read the IBSRDR.
  2) Finished.

- Transmission Interrupt Request (The interrupt request is generated by the IBSSR.TDRE)
  1) Write the transmission data to the IBSTDR.
  2) Finished.

4. Registers

This section describes the registers of I2CSLAVE.

Table 4-1 List of I2CSLAVE register

Abbreviation	Register name	See
IBSSR	I2C Bus Slave Status Register	4.1
IBSCR	I2C Bus Slave Control Register	4.2
IBSADR	I2C Bus 7-bit Slave Address Register	4.3
IBSMSKR	I2C Bus 7-bit Slave Address Mask Register	4.4
IBSDSTUPR	I2C Bus Slave Data Setup Register	4.5
IBSTDR	I2C Bus Slave Transmission Data Register	4.6
IBSRDR	I2C Bus Slave Reception Data Register	4.7
IBSSCR	I2C Bus Slave Status Clear Register	4.8
IBSSSR	I2C Bus Slave Status Set Register	4.9

4.1. I2C Bus Slave Status Register (IBSSR)

I2C Bus Slave Status Register (IBSSR) displays the generation of each interrupt request flags, first byte, acknowledge data, reserved address, direction of the transmission or reception, state of I2CSLAVE, detection of Re-START Condition and STOP Condition, state of I2C Bus. This Register is initialized if I2CSLAVE is disabled (IBSMSKR.EN="0").

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	INT	RDRF	TDRE
Attribute	-	-	-	-	-	R	R	R
Initial Value	0	0	0	0	0	0	0	1

bit	7	6	5	4	3	2	1	0
Field	FBT	RACK	RSA	TRX	ACT	RSC	SPC	BB
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

[bit15:11] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit10] INT : Status Interrupt Request flag

This flag is set to "1" when the following one of conditions is satisfied.

- When the IBSADR.SAEN is "1", the received slave address is matched with the value of the IBSADR.SA[6:0]. (The address bits set to "0" in IBSMSKR.SM[6:0] are judged as a match) Or
- When the IBSCR.RSVEN is "1", one of the reserved addresses (0000xxxx or 1111xxx) is received. Or
- When the IBSCR.WSEL is "1" and the IBSSR.RSA is"1", the second byte is received. Or
- When the IBSSR.TRX is "1", "Not Acknowledge (NACK)" is received.

This flag is cleared to "0" if "1" is written to the Status Interrupt Request Flag Clear bit (INTC) in the I2C Bus Slave Status Clear Register (IBSSCR).

bit	Description	
	Read	Write
0	Not detect the condition for the Status Interrupt Request	No effect
1	Detect the condition for the Status interrupt Request	No effect

[bit9] RDRF : Reception Interrupt Request flag

When data for the second and subsequent bytes or one of the reserved addresses (0000xxx or 1111xxx) for the first byte is received, this flag is set to "1".

This flag is cleared to "0" when the received data is read from the I2C Bus Slave Reception Data Register (IBSRDR).

bit	Description	
	Read	Write
0	Not detect the condition for the Reception Interrupt Request	No effect
1	Detect the condition for the Reception Interrupt Request	No effect

[bit8] TDRE : Transmission Data Empty flag

This flag is set to "1" if the transmission data is empty during the transmission or "1" is written to the Transmission Data Empty Flag Set bit (TDRES) in the I2C Bus Slave Status Set Register (IBSSSR).

This flag is cleared to "0" when the transmission data is written to the I2C Bus Slave Transmission Data Register (IBSTDR).

bit	Description	
	Read	Write
0	Not detect the condition for the Transmission Interrupt Request	No effect
1	Detect the condition for the Transmission Interrupt Request	No effect

[bit7] FBT : First Byte flag

This flag is set to "1" if the first byte is received after the START Condition. This flag is cleared to "0" if the I2C Bus transmits the data of transmission or reception for the second and subsequent bytes.

bit	Description	
	Read	Write
0	I2C Bus Idle, or the transmission or reception for the second and subsequent bytes	No effect
1	Reception for the first byte	No effect

Note:

- *This flag is valid when any of the following conditions are satisfied.*
  - *Status Interrupt Request flag (INT) is "1".*
  - *Reception Interrupt Request flag (RDRF) is "1".*
  - *After receiving 1st byte, Transmission Data Empty flag (TDRE) is "1" and Data Direction flag (TRX) is "1".*

[bit6] RACK : Acknowledge flag

This flag shows the received data at the timing of acknowledge.

bit	Description	
	Read	Write
0	The I2C data line (SDA) was "L" at the timing of acknowledge	No effect
1	The I2C data line (SDA) was "H" at the timing of acknowledge	No effect

Note:

- *This flag is valid when any of the following conditions are satisfied.*
  - *Status Interrupt Request flag (INT) is "1".*
  - *Reception Interrupt Request flag (RDRF) is "1".*
  - *After receiving 1st byte, Transmission Data Empty flag (TDRE) is "1" and Data Direction flag (TRX) is "1".*

[bit5] RSA : Reserved Address Detection flag

This flag is set to "1" if one of the reserved addresses in the first byte is detected. This flag is cleared to "0" if the START Condition or STOP Condition is detected.

bit	Description	
	Read	Write
0	Not detect a reserved address	No effect
1	Detect a reserved address	No effect

Note:

- *This flag is valid when any of the following conditions are satisfied.*
  - *Status Interrupt Request flag (INT) is "1".*
  - *Reception Interrupt Request flag (RDRF) is "1".*
  - *After receiving 1st byte, Transmission Data Empty flag (TDRE) is "1" and Data Direction flag (TRX) is "1".*

[bit4] TRX : Data Direction flag

This flag shows the bit0 in the first byte. The bit0 in the first byte decides the transmission or reception.

bit	Description	
	Read	Write
0	Reception	No effect
1	Transmission	No effect

Note:

- *This flag is valid when any of the following conditions are satisfied.*
  - *Status Interrupt Request flag (INT) is "1".*
  - *Reception Interrupt Request flag (RDRF) is "1".*

- *After receiving 1st byte, Transmission Data Empty flag (TDRE) is "1" and Data Direction flag (TRX) is "1".*

[bit3] ACT : Active flag

This flag shows whether I2CSLAVE is active as the slave of I2C Bus or not.

bit	Description	
	Read	Write
0	Not active as the slave	No effect
1	Active as the slave	No effect

Note:

- *This flag is valid when any of the following conditions are satisfied.*
  - *Status Interrupt Request flag (INT) is "1".*
  - *Reception Interrupt Request flag (RDRF) is "1".*
  - *After receiving 1st byte, Transmission Data Empty flag (TDRE) is "1" and Data Direction flag (TRX) is "1".*

[bit2] RSC : Re-Start Condition Detection flag

When ACT flag is "1", this flag is set to "1" if the START Condition is detected. This flag is cleared to "0" if "1" is written to the Re-Start Condition Detection Flag Clear bit (RSCC) in the I2C Bus Slave Status Clear Register (IBSSCR).

bit	Description	
	Read	Write
0	Not detect the Re-Start Condition	No effect
1	Detect the Re-Start Condition	No effect

[bit1] SPC : Stop Condition Detection flag

When ACT flag is "1", this flag is set to "1" if the STOP Condition is detected. This flag is cleared to "0" if "1" is written to the Stop Condition Detection Flag Clear bit (SPCC) in the I2C Bus Slave Status Clear Register (IBSSCR).

bit	Description	
	Read	Write
0	Not Detect the STOP Condition	No effect
1	Detect the STOP Condition	No effect

[bit0] BB : Bus Busy flag

This flag is set to "1" if the START Condition is detected. This flag is cleared to "0" if the STOP Condition is detected.

bit	Description	
	Read	Write
0	I2C Bus idle	No effect
1	I2C Bus Busy	No effect

4.2. I2C Bus Slave Control Register (IBSCR)

I2C Bus Slave Control Register (IBSCR) is used to control the noise filter, enable to detect the reserved address, enable to output acknowledge, select the waiting timing, or enable to generate each interrupt requests. This Register has no effect if I2CSLAVE is disabled (IBSMSKR.EN="0").

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	NFCNT[1:0]	
Attribute	-	-	-	-	-	-	R/W	
Initial Value	0	0	0	0	0	0	00	

bit	7	6	5	4	3	2	1	0
Field	RSVEN	Reserved	ACKE	WSEL	CNDE	INTE	RIE	TIE
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

[bit15:10] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit9:8] NFCNT[1:0] : Noise Filter Control bits

These bits enable or disable the noise filter.

NFCNT[1:0]	Description
00	Enable
01	Not allowed
10	Not allowed
11	Disable

Note:

- *This bit must be changed when I2CSLAVE is disabled (IBSMSKR.EN="0").*

[bit7] RSVEN : Reserved Address Enable bit

This bit enables or disables the detection of the reserved addresses (0000xxx or 1111xxx).

bit	Description
0	Disable
1	Enable

Note:

- *This bit must be changed when I2CSLAVE is disabled (IBSMSKR.EN="0").*

[bit6] Reserved : reserved bit

Always write "0" to this bit. The read value is "0".

[bit5] ACKE : Acknowledge Enable bit

When I2CSLAVE is the state of the reception (IBSSR.TRX="0"), this bit selects whether I2CSLAVE outputs "L" to the I2C data line (SDA) at the timing of acknowledge or not.

bit	Description
0	Output NACK (output "H")
1	Output ACK (output "L")

Note:

- *This bit must be changed when one of the following conditions is satisfied.*
  - *I2CSLAVE is disabled (IBSMSKR.EN="0"), or*
  - *The Status Interrupt Request flag (IBSSR.INT) is "1", or*
  - *The Reception Interrupt Request flag (IBSSR.RDRF) is "1"*

[bit4] WSEL : Wait Select bit

This bit selects that I2C Bus is stopped (SCL="L") before the timing of acknowledge or after the timing of acknowledge about the transmission or reception for the second and sequent bytes. If I2C Bus is stopped, any of the Status Interrupt Request, Reception Interrupt Request and/or Transmission Interrupt Request are generated.

bit	Description
0	After the timing of acknowledge
1	Before the timing of acknowledge

Note:

- *This bit must be changed when one of the following conditions is satisfied.*
  - *I2CSLAVE is disabled (IBSMSKR.EN="0"), or*
  - *The Status Interrupt Request flag (IBSSR.INT) is "1", or*
  - *The Reception Interrupt Request flag (IBSSR.RDRF) is "1", or*
  - *The Transmission Data Empty flag (IBSSR.TDRE) is "1" and I2CSLAVE is the state of the transmission (IBSSR.TRX="1").*

[bit3] CNDE : Condition Detection Interrupt Request Enable bit

The generation of interrupt request by the Re-Start Condition Detection flag (IBSSR.RSC) or Stop Condition Detection flag (IBSSR.SPC) is enabled or disabled.

bit	Description
0	Disable
1	Enable

[bit2] INTE : Status Interrupt Request Enable bit

The generation of interrupt request by the Status Interrupt Request flag (IBSSR.INT) is enabled or disabled.

bit	Description
0	Disable
1	Enable

[bit1] RIE : Reception Interrupt Request Enable bit

The generation of the interrupt request by the Reception Interrupt Request flag (IBSSR.RDRF) is enabled or disabled.

bit	Description
0	Disable
1	Enable

[bit0] TIE : Transmission Interrupt Request Enable bit

The generation of the interrupt request by the Transmission Data Empty flag (IBSSR.TDRE) is enabled or disabled.

bit	Description
0	Disable
1	Enable

4.3. I2C Bus 7-bit Slave Address Register (IBSADR)

I2C Bus 7-bit Slave Address Register (IBSADR) is used to enable the detection of the match with the received slave address and be set to the value of the slave address. This Register must be changed when I2CSLAVE is disabled (IBSMSKR.EN="0").

bit	7	6	5	4	3	2	1	0
Field	SAEN	SA[6:0]						
Attribute	R/W	R/W						
Initial Value	0	0000000						

[bit7] SAEN : Slave Address Enable bit

This bit enables or disables to compare the value of the slave address with the received slave address.

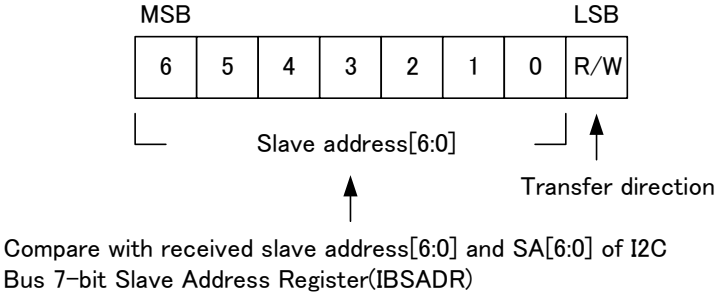
bit	Description
0	Disable
1	Enable

[bit6:0] SA[6:0] : Slave Address Value bit

These bits are set to the value of 7-bit slave address

- If the slave address detection is enabled (SAEN="1"), the slave address value bit (SA[6:0]) is compared to the 7-bit data which has been received after detection of (Re-)START Condition. If match, I2CSLAVE operates as the slave mode and an Acknowledge (ACK) is output. At this time, the received slave address is set in this register (if SAEN="0", Not Acknowledge (NACK) is output).
- The 7-bit slave address and the transfer direction are contained in the first byte after the detection of the (Re-)START Condition. The received slave address and these bits are compared.

Figure 4-1 First byte format after the detection of the (Re-)START Condition



- If the IBSMSKR.SM[n] (n=6-0) is set to "0", slave address[n] (n=6-0) corresponding to it is not compared.

SA[6:0]	Description
0x00-0x3F	Value of 7-bit slave address

4.4. I2C Bus 7-bit Slave Address Mask Register (IBSMSKR)

I2C Bus 7-bit Slave Address Mask Register (IBSMSKR) is used to enable I2CSLAVE or be set to the masked information of the 7-bit slave address. SM[6:0] bits must be changed when I2CSLAVE is disabled (EN="0").

bit	7	6	5	4	3	2	1	0
Field	EN	SM[6:0]						
Attribute	R/W	R/W						
Initial Value	0	1111111						

[bit7] EN : Operating Enable bit

I2CSLAVE is enabled or disabled about the operating.

bit	Description
0	Disable
1	Enable

[bit6:0] SM[6:0] : Slave Address Mask bits

These bits are set to the masked information of the 7-bit slave address. As the masked bit isn't compared, it is treaded as a match.

SM[n]	Description
0	Not compare
1	Compare

n=6, 5, 4, 3, 2, 1, 0

4.5. I2C Bus Slave Data Setup Register (IBSDSTUPR)

I2C Bus Data Setup Register (IBSDSTUPR) is set to the setup time between the I2C data line (SDA) and I2C clock line (SCL). The unit is the period of the operating clock (iPCLK). This Register must be changed when I2CSLAVE is disabled (IBSMSKR.EN="0").

bit	7	6	5	4	3	2	1	0
Field	SETUP[7:0]							
Attribute	R/W							
Initial Value	11111111							

[bit7:0] SETUP[7:0] : Setup bits

These bits are set to the setup time between the I2C data line (SDA) and I2C clock line (SCL).

SETUP[7:0]	Description
0x00	0
0x01	1 x iPCLK period
0x02	2 x iPCLK period
0x03	3 x iPCLK period
0xFF	255 x iPCLK period

4.6. I2C Bus Slave Transmission Data Register (IBSTDR)

I2CSLAVE outputs the data written to the I2C Bus Slave Transmission Data Register (IBSTDR) on the I2C Bus.

bit	7	6	5	4	3	2	1	0
Field	TDR[7:0]							
Attribute	R/W							
Initial Value	11111111							

[bit7:0] TDR[7:0] : Transmission Data bits

The data of the transmission is written to these bits.

Note:

- *Don't write value to this Register when Transmission Data Empty flag (IBSSR.TDRE) is "0".*
- *Write transmission data to this register if Status Interrupt Request flag(IBSSR.INT) is "1" and Data Direction flag(IBSSR.TRX) is "1" in the 1st byte received.*

4.7. I2C Bus Slave Reception Data Register (IBSRDR)

I2C Bus Slave Reception Data Register (IBSRDR) saves the received data.

bit	7	6	5	4	3	2	1	0
Field	RDR[7:0]							
Attribute	R							
Initial Value	11111111							

[bit7:0] RDR[7:0] : Reception Data bits

The received data is saved to these bits.

Note:

- *This Register is valid when Reception Interrupt Request flag (IBSSR.RDRF) is "1".*

4.8. I2C Bus Slave Status Clear Register (IBSSCR)

I2C Bus Slave Status Clear Register (IBSSCR) is used to clear the Status Interrupt Request flag (IBSSR.INT), Re-Start Condition Detection flag (IBSSR.RSC) and Stop Condition Detection flag (IBSSR.SPC) to "0".

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	INTC	Reserved	Reserved
Attribute	-	-	-	-	-	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	RSCC	SPCC	Reserved
Attribute	-	-	-	-	-	W	W	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:11] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit10] INTC : Status Interrupt Request Flag Clear bit

The Status Interrupt Request flag (IBSSR.INT) is cleared to "0" if "1" is written to this bit. The read value is"0".

bit	Description
	Write
0	No effect
1	The Status Interrupt Request flag (IBSSR.INT) is cleared to "0"

[bit9:3] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit2] RSCC : Re-Start Condition Detection Flag Clear bit

The Re-Start Condition Detection flag (IBSSR.RSC) is cleared to "0" if "1" is written to this bit. The read value is"0".

bit	Description
	Write
0	No effect
1	The Re-Start Condition Detection flag (IBSSR.RSC) is cleared to "0"

[bit1] SPCC : Stop Condition Detection Flag Clear bit

The Stop Condition Detection flag (IBSSR.SPC) is cleared to "0" if "1" is written to this bit. The read value is"0".

bit	Description
	Write
0	No effect
1	The Stop Condition Detection flag (IBSSR.SPC) is cleared to "0"

[bit0] Reserved : Reserved bit

Always write "0" to this bit. The read value is "0".

4.9. I2C Bus Slave Status Set Register (IBSSSR)

I2C Bus Slave Status Set Register (IBSSSR) is used to set the Transmission Data Empty flag (IBSSR.TDRE) to "1".

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TDRES
Attribute	-	-	-	-	-	-	-	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	Reserved	Reserved	Reserved
Attribute	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0

[bit15:9] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

[bit8] TDRES : Transmission Data Empty Flag Set bit

The Transmission Data Empty flag (IBSSR.TDRE) is set to "1" if "1" is written to this bit. The read value is"0".

bit	Description
	Write
0	No effect
1	The Transmission Data Empty flag (IBSSR.TDRE) is set to "1"

[bit7:0] Reserved : Reserved bits

Always write "0" to these bits. The read value is "0".

APPENDIXES

This chapter shows the register map and list of notes.

A. Product Type

B. Register Map (TYPE1-M0+)

C. Register Map (TYPE2-M0+)

D. Register Map (TYPE3-M0+)

E. List of Notes

CODE: 9AFAPPENDIXES-E03.0

A. Product Type

This section describes the product TYPE.

1. Product TYPE List

CODE: xxxx

1. Product TYPE List

In this manual, the products are classified into the following groups and are described as follows.

For the descriptions such as "TYPE1-M0+","TYPE2-M0+" and “TYPE3-M0+”, see the relevant items of the target FM0+ product in the lists below.

Table 1-1 FM0+ family TYPE1-M0+ Product list

TYPE	Flash memory size	
	56 Kbytes	**88 Kbytes**
TYPE1-M0+	S6E1A11B S6E1A11C	S6E1A12B S6E1A12C

Table 1-2 FM0+ family TYPE2-M0+ Product list

TYPE	Flash memory size	
	304K bytes	560K bytes
TYPE2-M0+	S6E1B84E S6E1B84F S6E1B84G S6E1B34E S6E1B34F S6E1B34G	S6E1B86E S6E1B86F S6E1B86G S6E1B36E S6E1B36F S6E1B36G

Table 1-3 FM0+ family TYPE3 Product list

TYPE	Flash memory size	
	64K bytes	128K bytes
TYPE3-M0+	S6E1C31B S6E1C31C S6E1C31D S6E1C11B S6E1C11C S6E1C11D	S6E1C32B S6E1C32C S6E1C32D S6E1C12B S6E1C12C S6E1C12D

B. Register Map (TYPE1-M0+)

This chapter shows the register map.

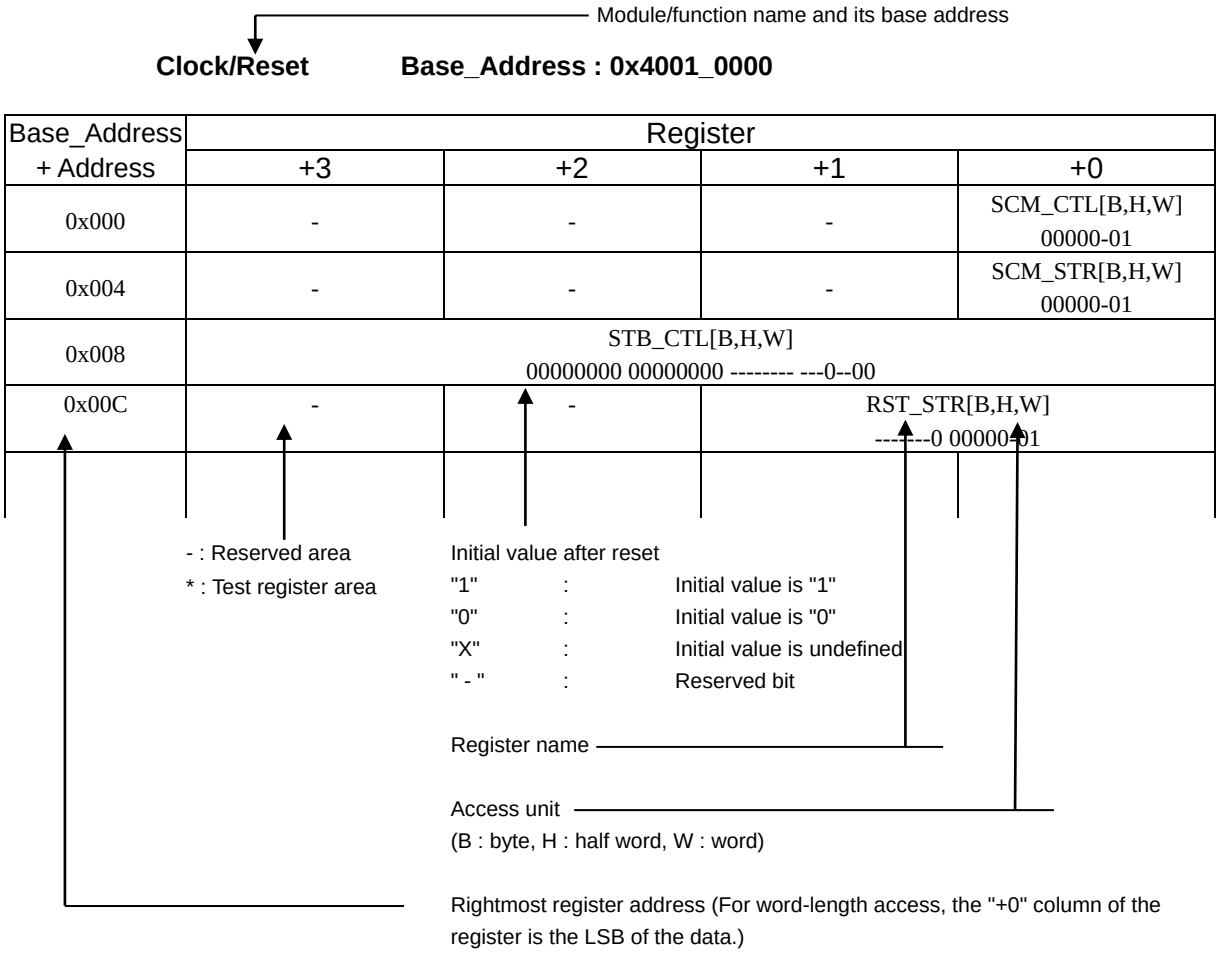
1. Register Map

CODE: 9AFREGMAP-E01.0

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.
- When the register is accessed by larger unit than register size, for the reserved area to access at the same time, the read value is undefined, and writing is invalid.

1.1 Flash I/F

Flash I/F Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C				
0x010	FSYNDN[B,H,W]			
0x014 - 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, refer to "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
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	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
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	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----
0x058	-	-	-	*

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address	Register			
+ Address	+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 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW WDT

SW WDT Base_Address : 0x4001_2000

Base_Address	Register			
+ Address	**+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	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 Dual Timer

Dual Timer Base_Address : 0x4001_5000

Base_Address	Register			
+ Address	+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	-	-	-	-

1.7 MFT

MFT unit0 Base_Address : 0x4002_0000

MFT unit1 Base_Address : 0x4002_1000

MFT unit2 Base_Address : 0x4002_2000

Base_Address	Register			
+ Address	+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,H,W]	OCSA54[B,H,W]
		00000000	00000000	00000000
0x124	-	-	OCSC[B,H,W]	-
			--000000	
0x128	-	-	OCSE0[H,W]	
			00000000 00000000	
0x12C	OCSE1[H,W]			
	00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[H,W]	
			00000000 00000000	
0x134	OCSE3[H,W]			
	00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[H,W]	
			00000000 00000000	
0x13C	OCSE5[H,W]			
	00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W]		-	-
	11111111 11111111			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x144	TCDT0[H,W]		-	-
	00000000 00000000			
0x148	TCSC0[B,H,W]		TCSA0[B,H,W]	
	00000000 00000000		000---00 01000000	
0x14C	TCCP1[H,W]		-	-
	11111111 11111111			
0x150	TCDT1[H,W]			
	00000000 00000000			
0x154	TCSC1[B,H,W]		TCSA1[B,H,W]	
	00000000 00000000		000---00 01000000	
0x158	TCCP2[H,W]		-	-
	11111111 11111111			
0x15C	TCDT2[H,W]		-	-
	00000000 00000000			
0x160	TCSC2[B,H,W]		TCSA2[B,H,W]	
	00000000 00000000		000---00 01000000	
0x164	TCAL[B,H,W] (only in unit 0)			
	00000000 00000000 11111111 11111111			
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	
0x1A4	-	-	WFSA10[H,W]	
			---00000 000000	
0x1A8	-	-	WFSA32[H,W]	
			---00000 000000	
0x1AC	-	-	WFSA54[H,W]	
			---00000 000000	
0x1B0	-	-	WFIR[H,W]	
			00000000 00000000	
0x1B4	-	-	NZCL[H,W]	
			-000--00 ---00000	
0x1B8	ACMP0		-	-
	00000000 00000000			
0x1BC	ACMP1		-	-
	00000000 00000000			
0x1C0	ACMP2		-	-
	00000000 00000000			
0x1C4	ACMP3		-	-
	00000000 00000000			
0x1C8	ACMP4		-	-
	00000000 00000000			
0x1CC	ACMP5		-	-
	00000000 00000000			
0x1D0	-	-	ACSA[B,H,W]	
			--000000 --000000	
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

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x1E8	-	-	ACSD5[B,H,W]	ACSC5[B,H,W]
			00000000	00000000
0x1EC - 0xFFC	-	-	-	-

1.8 PPG

PPG Base_Address : 0x4002_4000

Base_Address	Register			
+ Address	+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	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	-	-	TRG0[B,H,W]	
			00000000 00000000	
0x104	-	-	REVC0[B,H,W]	
			00000000 00000000	
0x108 - 0x13C	-	-	-	-
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--00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8[B,H,W]	PPGC9[B,H,W]
			00000000	00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
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--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

Base_Address	Register			
+ Address	+3	+2	+1	+0
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--00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	IGBTC[B,H,W]
				00000000
0x384 - 0xFFC	-	-	-	-

1.9 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	Register			
+ Address	+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	0000-000
0x014 - 0x03C	-	-	-	-

1.10 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) **Base Address : 0x4002_5100**

Base_Address	Register			
+ Address	+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	Register			
+ Address	+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	Register			
+ Address	+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	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSELCDEF[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simultaneous Startup (Base Timer) **Base Address : 0x4002_5F00**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR[B,H,W] XXXXXXXX XXXXXXXX	

1.11 QPRC

QPRC ch.0 **Base Address : 0x4002_6000**

QPRC ch.1 **Base Address : 0x4002_6040**

QPRC ch.2 **Base Address : 0x4002_6080**

Base_Address	Register			
+ Address	+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 - 0x0038	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12 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**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	NFCTLA[B,H,W]			
	-------- -------- -------- --00-000			
0x0004	NFCTLB[B,H,W]			
	-------- -------- -------- --00-000			
0x0008	NFCTLC[B,H,W]			
	-------- -------- -------- --00-000			
0x000C	-	-	-	-

1.13 A/DC

12-bit A/DC unit0 Base_Address : 0x4002_7000

12-bit A/DC unit1 Base_Address : 0x4002_7100

12-bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address	Register			
+ Address	+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]
			1000-000	--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	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			--000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.14 D/AC

10-bit D/AC Base_Address : 0x4002_8000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	DACR0[B,H,W]	DADR0[B,H,W]	
		-------0	------XX XXXXXXXX	
0x04	-	DACR1[B,H,W]	DADR1[B,H,W]	
		-------0	------XX XXXXXXXX	
0x08 - 0xFC	-	-	-	-

1.15 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	+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[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+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[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C - 0x0FC	-	-	-	-

1.17 INT-Req. READ

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DRQSEL[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	*			
0x008 - 0x00B	-	-	-	-
0x00C	-	-	-	IRQCMODE[B,H,W]
				-------0
0x010	EXC02MON[B,H,W]			
	-------- -------- -------- ------00			
0x014	IRQ00MON[B,H,W]			
	-------- -------- -------- -------0			
0x018	IRQ01MON[B,H,W]			
	-------- -------- -------- -------0			
0x01C	IRQ02MON[B,H,W]			
	-------- -------- -------- -------0			
0x020	IRQ03MON[B,H,W]			
	-------- -------- ----0000 00000000			
0x024	IRQ04MON[B,H,W]			
	-------- -------- -------- 00000000			
0x028	IRQ05MON[B,H,W]			
	-------- 00000000 00000000 00000000			
0x02C	IRQ06MON[B,H,W]			
	-------- ----0000 00000000 00000000			
0x030	IRQ07MON[B,H,W]			
	-------- -------- -------- ------00			
0x034	IRQ08MON[B,H,W]			
	-------- -------- -------- ----0000			
0x038	IRQ09MON[B,H,W]			
	-------- -------- -------- ------00			
0x03C	IRQ10MON[B,H,W]			
	-------- -------- -------- ----0000			
0x040	IRQ11MON[B,H,W]			
	-------- -------- -------- ------00			
0x044	IRQ12MON[B,H,W]			
	-------- -------- -------- ----0000			
0x048	IRQ13MON[B,H,W]			
	-------- -------- -------- ------00			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x04C	IRQ14MON[B,H,W]			
	-------- -------- -------- ----0000			
0x050	IRQ15MON[B,H,W]			
	-------- -------- -------- ------00			
0x054	IRQ16MON[B,H,W]			
	-------- -------- -------- ----0000			
0x058	IRQ17MON[B,H,W]			
	-------- -------- -------- ------00			
0x05C	IRQ18MON[B,H,W]			
	-------- -------- -------- ----0000			
0x060	IRQ19MON[B,H,W]			
	-------- -------- -------- ---0--00			
0x064	IRQ20MON[B,H,W]			
	-------- -------- -------- ---00000			
0x068	IRQ21MON[B,H,W]			
	-------- -------- -------- ---0--00			
0x06C	IRQ22MON[B,H,W]			
	-------- -------- -------- ---00000			
0x070	IRQ23MON[B,H,W]			
	-------- -------- -------0 00000000			
0x074	IRQ24MON[B,H,W]			
	-------- -------- -------- --00-000			
0x078	IRQ25MON[B,H,W]			
	-------- -------- -------- ---00000			
0x07C	IRQ26MON[B,H,W]			
	-------- -------- -------- ---00000			
0x080	IRQ27MON[B,H,W]			
	-------- -------- -------- --000000			
0x084	IRQ28MON[B,H,W]			
	-------- ------00 00000000 00000000			
0x088	IRQ29MON[B,H,W]			
	-------- -------- ----0000 00000000			
0x08C	IRQ30MON[B,H,W]			
	-------- ------00 00000000 00000000			
0x090	IRQ31MON[B,H,W]			
	----0--- -------- 00000000 00000000			
0x094 - 0x20C	-	-	-	-
0x210	RCINTSEL0[B,H,W] 00000000 00000000 00000000 00000000			
0x214	RCINTSEL1[B,H,W] 00000000 00000000 00000000 00000000			
0x218 - 0xFFC	-	-	-	-

1.18 GPIO

GPIO Base_Address : 0x4003_3000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
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	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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]			
	---- ---- ---- ---- ---- ---- ---- 0101			
0x584 - 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- ---- ---- ---1 ---- ---- 0000 -000			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 --00 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 ----			
0x620	EPFR08[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x628 - 0x62C	-	-	-	-
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			
0x63C	EPFR15[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x644	EPFR17[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x648	EPFR18[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0000			
0x64C - 0x650	-	-	-	-
0x654	EPFR21[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- -000			
0x658	EPFR22[B,H,W]			
	---- ---- ---- ---- 0000 ---- 0000 ----			
0x65C - 0x6FC	-	-	-	-
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 - 0x7FC	-	-	-	-
0x800	*			
0x804	*			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x808 - 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	FPOER7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924	FPOER9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x928	FPOERA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x92C	FPOERB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x930	FPOERC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x934	FPOERD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C	FPOERF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x940 - 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	Register			
+ Address	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				--0000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				--00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 - 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 - 0xFC	-	-	-	-

1.20 LVD

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	LVD_CTL[B,H,W]	
			100000-- 000011--	
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
				01------
0x014 - 0x0FC	-	-	-	-

1.21 DS Mode

DS Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	REG_CTL[B,H,W]
				-------0
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]
				-----000
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	-	-	-	-

1.22 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

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0--00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	-	-	RDR/TDR[H,W]	
			00000000 00000000	
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/ EIBCR[B,H,W]	SCSTR0 [B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000--0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x044 - 0x0FC	-	-	-	-

1.23 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			

1.24 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 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 - 0xFFC	-	-	-	-

1.25 RTC

RTC Base_Address : 0x4003_B000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WTCR1[B,H,W]			
	00000000 00000000 ---00000 -00000-0			
0x004	WTCR2[B,H,W]			
	-------- -------- -----000 -------0			
0x008	WTBR[B,H,W]			
	-------- 00000000 00000000 00000000			
0x00C	WTDR[B,H,W]	WTHR[B,H,W]	WTMIR[B,H,W]	WTSR[B,H,W]
	--000000	--000000	-0000000	-0000000
0x010	-	WTYR[B,H,W]	WTMOR[B,H,W]	WTDW[B,H,W]
		00000000	---00000	-----000
0x014	ALDR[B,H,W]	ALHR[B,H,W]	ALMIR[B,H,W]	-
	--000000	--000000	-0000000	
0x018	-	ALYR[B,H,W]	ALMOR[B,H,W]	-
		00000000	---00000	
0x01C	WTTR[B,H,W]			
	-------- ------00 00000000 00000000			
0x020	-	-	WTCLKM[B,H,W]	WTCLKS[B,H,W]
			------00	-------0
0x024	-	WTCALEN[B,H,W]	WTCAL[B,H,W]	
		-------0	------00 00000000	
0x028	-	-	WTDIVEN[B,H,W]	WTDIV[B,H,W]
			------00	----0000
0x02C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x030	-	-	-	WTCOSEL[B,H,W]
				-------0
0x034 - 0xFFC	-	-	-	-

1.26 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
0x000 - 0x0FC	-	-	-	-

1.27 Peripheral Clock Gating

Peripheral Clock Gating Base_Address : 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1---1 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	-------0 ----0000 00000000 00000000			
0x008 - 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000 ----0000 ----0000			
0x018 - 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -------- -------- --00----			
0x024	MRST2[B,H,W]			
	-------- -------- -------- --00----			
0x028 - 0x0FC	-	-	-	-

1.28 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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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 00000000 00000000 00000000			
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	-	-	-	-

1.29 MTB_DWT

MTB_DWT Base_Address : 0xF000_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 - 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF8	CID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.30 Fast GPIO

Fast GPIO Base_Address : 0xF800_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	FPDIR7[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024	-	-	FPDIR9[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x028	-	-	FPDIRA[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x02C	-	-	FPDIRB[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x030	-	-	FPDIRC[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x034	-	-	FPDIRD[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	FPDIRF[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	FPDOR7[B,H,W]	
			00000000 00000000	
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064	-	-	FPDOR9[B,H,W]	
			00000000 00000000	
0x068	-	-	FPDORA[B,H,W]	
			00000000 00000000	
0x06C	-	-	FPDORB[B,H,W]	
			00000000 00000000	
0x070	-	-	FPDORC[B,H,W]	
			00000000 00000000	
0x074	-	-	FPDORD[B,H,W]	
			00000000 00000000	
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	FPDORF[B,H,W]	
			00000000 00000000	
0x080	-	-	-	M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088	-	-	-	M_FPDIR2[B,H,W]
				XXXXXXXX
0x08C	-	-	-	M_FPDIR3[B,H,W]
				XXXXXXXX
0x090	-	-	-	M_FPDIR4[B,H,W]
				XXXXXXXX
0x094	-	-	-	M_FPDIR5[B,H,W]
				XXXXXXXX

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x098	-	-	-	M_FPDIR6[B,H,W]
				XXXXXXXX
0x09C	-	-	-	M_FPDIR7[B,H,W]
				XXXXXXXX
0x0A0	-	-	-	M_FPDIR8[B,H,W]
				XXXXXXXX
0x0A4	-	-	-	M_FPDIR9[B,H,W]
				XXXXXXXX
0x0A8	-	-	-	M_FPDIRA[B,H,W]
				XXXXXXXX
0x0AC	-	-	-	M_FPDIRB[B,H,W]
				XXXXXXXX
0x0B0	-	-	-	M_FPDIRC[B,H,W]
				XXXXXXXX
0x0B4	-	-	-	M_FPDIRD[B,H,W]
				XXXXXXXX
0x0B8	-	-	-	M_FPDIRE[B,H,W]
				XXXXXXXX
0x0BC	-	-	-	M_FPDIRF[B,H,W]
				XXXXXXXX
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8	-	-	-	M_FPDOR2[B,H,W]
				00000000
0x0CC	-	-	-	M_FPDOR3[B,H,W]
				00000000
0x0D0	-	-	-	M_FPDOR4[B,H,W]
				00000000
0x0D4	-	-	-	M_FPDOR5[B,H,W]
				00000000
0x0D8	-	-	-	M_FPDOR6[B,H,W]
				00000000
0x0DC	-	-	-	M_FPDOR7[B,H,W]
				00000000
0x0E0	-	-	-	M_FPDOR8[B,H,W]
				00000000
0x0E4	-	-	-	M_FPDOR9[B,H,W]
				00000000
0x0E8	-	-	-	M_FPDORA[B,H,W]
				00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0EC	-	-	-	M_FPDORB[B,H,W]
				00000000
0x0F0	-	-	-	M_FPDORC[B,H,W]
				00000000
0x0F4	-	-	-	M_FPDORD[B,H,W]
				00000000
0x0F8	-	-	-	M_FPDORE[B,H,W]
				00000000
0x0FC	-	-	-	M_FPDORF[B,H,W]
				00000000
0x100 - 0xFFC	-	-	-	-

C. Register Map (TYPE2-M0+)

This chapter shows the register map.

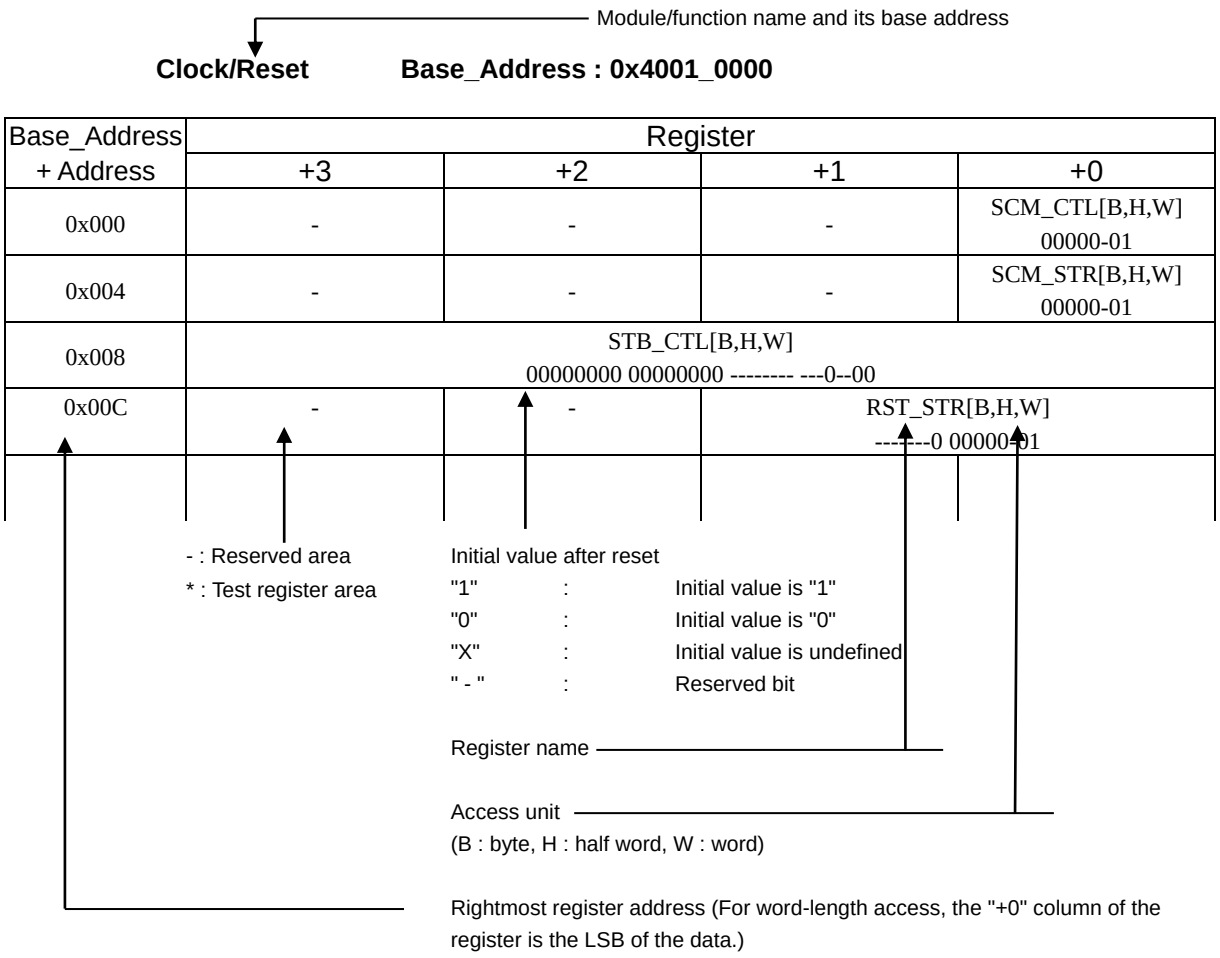
1. Register Map

CODE: 9AFREGMAP-E01.0

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.
- When the register is accessed by larger unit than register size, for the reserved area to access at the same time, the read value is undefined, and writing is invalid.

1.1 Flash I/F

Flash I/F Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C	FRVRC[B,H,W]			
0x010	FSYNDN[B,H,W]			
0x014 - 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, refer to "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
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	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
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	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address	Register			
+ Address	+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[R]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW_WDT

SW WDT Base_Address : 0x4001_2000

Base_Address	Register			
+ Address	+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[R]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			
0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 Dual Timer

Dual Timer Base_Address : 0x4001_5000

Base_Address	Register			
+ Address	+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	-	-	-	-

1.7 MFT

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		-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x144	TCDT0[H,W] 00000000 00000000		-	-
0x148	TCSC0[H,W] 00000000 00000000		TCSA0[B,H,W] 00000000 01000000	
0x14C	TCCP1[H,W] 11111111 11111111		-	-
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	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x194	WFTF32[H,W] 00000000 00000000		-	-
0x198	WFTB32[H,W] 00000000 00000000		WFTA32[H,W] 00000000 00000000	
0x19C	WFTF54[H,W] 00000000 00000000		-	-
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

Base_Address + Address	Register			
	+3	+2	+1	+0
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	-	-	-	-

1.8 PPG

PPG Base_Address : 0x4002_4000

Base_Address	Register			
+ Address	+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	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	-	-	TRG0[B,H,W]	
			00000000 00000000	
0x104	-	-	REVC0[B,H,W]	
			00000000 00000000	
0x108 - 0x13C	-	-	-	-
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--00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8[B,H,W]	PPGC9[B,H,W]
			00000000	00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
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--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]

Base_Address	Register			
+ Address	+3	+2	+1	+0
			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--00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	IGBTC[B,H,W]
				00000000
0x384 - 0xFFC	-	-	-	-

1.9 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	Register			
+ Address	+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	0000-000
0x014 - 0x03C	-	-	-	-

1.10 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) Base Address : 0x4002_5100

Base_Address	Register			
+ Address	+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	Register			
+ Address	+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	Register			
+ Address	+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	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSELCDEF[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simultaneous Startup (Base Timer) Base Address : 0x4002_5F00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR[B,H,W] XXXXXXXX XXXXXXXX	

1.11 QPRC

QPRC ch.0 **Base Address : 0x4002_6000**

QPRC ch.1 **Base Address : 0x4002_6040**

QPRC ch.2 **Base Address : 0x4002_6080**

Base_Address	Register			
+ Address	+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 - 0x0038	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12 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**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	NFCTLA[B,H,W]			
	-------- -------- -------- --00-000			
0x0004	NFCTLB[B,H,W]			
	-------- -------- -------- --00-000			
0x0008	NFCTLC[B,H,W]			
	-------- -------- -------- --00-000			
0x000C	-	-	-	-

1.13 A/DC

12-bit A/DC unit0 Base_Address : 0x4002_7000

12-bit A/DC unit1 Base_Address : 0x4002_7100

12-bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address	Register			
+ Address	+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]
			1000-000	--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	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			--000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.14 D/AC

10-bit D/AC Base_Address : 0x4002_8000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	DACR0[B,H,W]	DADR0[B,H,W]	
		-------0	------XX XXXXXXXX	
0x04	-	DACR1[B,H,W]	DADR1[B,H,W]	
		-------0	------XX XXXXXXXX	
0x08 - 0xFC	-	-	-	-

1.15 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	+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]
				-0111111
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+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[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C – 0x0FC	-	-	-	-

1.17 INT-Req. READ

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	*			
0x008 – 0x00B	-	-	-	-
0x00C	-	-	-	IRQCMODE[B,H,W] -------0
0x010	EXC02MON[B,H,W] -------- -------- -------- ------00			
0x014	IRQ00MON[B,H,W] -------- -------- -------- -------0			
0x018	IRQ01MON[B,H,W] -------- -------- -------- -------0			
0x01C	IRQ02MON[B,H,W] -------- -------- -------- -------0			
0x020	IRQ03MON[B,H,W] -------- -------- -------- ----0000			
0x024	IRQ04MON[B,H,W] -------- -------- -------- 00000000			
0x028	IRQ05MON[B,H,W] -------- -------- 00000000 00000000			
0x02C	IRQ06MON[B,H,W] -------- -------- -------- ------00			
0x030	IRQ07MON[B,H,W] -------- -------- -------- -------0			
0x034	IRQ08MON[B,H,W] -------- -------- -------- ------00			
0x038	IRQ09MON[B,H,W] -------- -------- -------- -------0			
0x03C	IRQ10MON[B,H,W] -------- -------- -------- ------00			
0x040	IRQ11MON[B,H,W] -------- -------- -------- -------0			
0x044	IRQ12MON[B,H,W] -------- -------- -------- ------00			
0x048	IRQ13MON[B,H,W] -------- -------- -------- -------0			
0x04C	IRQ14MON[B,H,W] -------- -------- -------- ------00			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x050	IRQ15MON[B,H,W]			
	-------- -------- -------- -------0			
0x054	IRQ16MON[B,H,W]			
	-------- -------- -------- ------00			
0x058	IRQ17MON[B,H,W]			
	-------- -------- -------- -------0			
0x05C	IRQ18MON[B,H,W]			
	-------- -------- -------- ------00			
0x060	IRQ19MON[B,H,W]			
	-------- -------- -------- -------0			
0x064	IRQ20MON[B,H,W]			
	-------- -------- -------- ------00			
0x068	IRQ21MON[B,H,W]			
	-------- -------- -------- -------0			
0x06C	IRQ22MON[B,H,W]			
	-------- -------- -------- ------00			
0x070	IRQ23MON[B,H,W]			
	-------- -------- ---0000- -----000			
0x074	IRQ24MON[B,H,W]			
	-------- -------- -----00- --000000			
0x078	IRQ25MON[B,H,W]			
	-------- -------- -------- ---00000			
0x07C	IRQ26MON[B,H,W]			
	-------- -------- -------- --000000			
0x080	IRQ27MON[B,H,W]			
	-------- -------- -------- --0-----			
0x084	IRQ28MON[B,H,W]			
	-------- -------- -------- --000000			
0x088	IRQ29MON[B,H,W]			
	-------- -------- -------0 00000000			
0x08C	IRQ30MON[B,H,W]			
	-------- -------- -------- --000000			
0x090	IRQ31MON[B,H,W]			
	----0--- -------- 00000000 00000000			
0x094 – 0x20C	-	-	-	-
0x210	RCINTSEL0[B,H,W]			
	00000000 00000000 00000000 00000000			
0x214	RCINTSEL1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x218 – 0xFFC	-	-	-	-

1.18 LCDC

LCDC Base_Address : 0x4003_2000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	LCDCC3[B,H,W]	LCDCC2[B,H,W]	LCDCC1[B,H,W]
		0011111-	--010100	-00000--
0x04	LCDC_PSR[B,H,W]			
	-------- -0000000 00000000 00000000			
0x08	LCDC_COMEN[B,H,W]			
	-------- -------- -------- 00000000			
0x0C	LCDC_SEGEN1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x10	LCDC_SEGEN2[B,H,W]			
	-------- -------- -------- 00000000			
0x14	-	-	LCDC_BLINK[B,H,W]	
			00000000 00000000	
0x18	-	-	LCDC_BOOSTER[B,H,W]	
			--001110	----0011
0x1C	LCDRAM03[B,H,W]	LCDRAM02[B,H,W]	LCDRAM01[B,H,W]	LCDRAM00[B,H,W]
	00000000	00000000	00000000	00000000
0x20	LCDRAM07[B,H,W]	LCDRAM06[B,H,W]	LCDRAM05[B,H,W]	LCDRAM04[B,H,W]
	00000000	00000000	00000000	00000000
0x24	LCDRAM11[B,H,W]	LCDRAM10[B,H,W]	LCDRAM09[B,H,W]	LCDRAM08[B,H,W]
	00000000	00000000	00000000	00000000
0x28	LCDRAM15[B,H,W]	LCDRAM14[B,H,W]	LCDRAM13[B,H,W]	LCDRAM12[B,H,W]
	00000000	00000000	00000000	00000000
0x2C	LCDRAM19[B,H,W]	LCDRAM18[B,H,W]	LCDRAM17[B,H,W]	LCDRAM16[B,H,W]
	00000000	00000000	00000000	00000000
0x30	LCDRAM23[B,H,W]	LCDRAM22[B,H,W]	LCDRAM21[B,H,W]	LCDRAM20[B,H,W]
	00000000	00000000	00000000	00000000
0x34	LCDRAM27[B,H,W]	LCDRAM26[B,H,W]	LCDRAM25[B,H,W]	LCDRAM24[B,H,W]
	00000000	00000000	00000000	00000000
0x38	LCDRAM31[B,H,W]	LCDRAM30[B,H,W]	LCDRAM29[B,H,W]	LCDRAM28[B,H,W]
	00000000	00000000	00000000	00000000
0x3C	LCDRAM35[B,H,W]	LCDRAM34[B,H,W]	LCDRAM33[B,H,W]	LCDRAM32[B,H,W]
	00000000	00000000	00000000	00000000
0x40	LCDRAM39[B,H,W]	LCDRAM38[B,H,W]	LCDRAM37[B,H,W]	LCDRAM36[B,H,W]
	00000000	00000000	00000000	00000000
0x44 – 0xFC	-	-	-	-

1.19 GPIO

GPIO Base_Address : 0x4003_3000

Base_Address	Register			
+ Address	+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	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x43C	PDORF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x440 – 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	---- ---- 1111 1111 1111 1111 1111 1111			
0x504 – 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0101			
0x584 – 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- ---- ---- ---1 ---- ---- 0000 -000			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 –00 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 ----			
0x620	EPFR08[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	---- ---- ---- ---- 0000 ---- ---- ----			
0x628 – 0x638	-	-	-	-
0x63C	EPFR15[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x644	-			
	-	-	-	-
0x648	EPFR18[B,H,W]			
	--00 0000 0000 0000 0000 0000 0000 0000			
0x64C – 0x658	-	-	-	-
0x65C	EPFR23[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x660 – 0x680	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x684	EPFR33[B,H,W]			
	---- 0000 0000 0000 ---- 0000 0000 0000			
0x688	EPFR34[B,H,W]			
	---- ---- ---- ---- ---- ---- 0000 ----			
0x68C – 0x690	-	-	-	-
0x694	EPFR37[B,H,W]			
	---- 0000 0000 0000 ---- ---- ---- ----			
0x698	EPFR38[B,H,W]			
	---- ---- ---- ---- ---- 0000 0000 0000			
0x69C – 0x6FC	-	-	-	-
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			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x740	LVDIE[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- ---0			
0x744 – 0x7FC	-	-	-	-
0x800	*			
0x804	*			
0x808 – 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	FPOER7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924	FPOER9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x928	FPOERA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x92C	FPOERB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x930	FPOERC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x934	FPOERD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C	FPOERF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x940 – 0xFFC	-	-	-	-

1.20 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	Register			
+ Address	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				000000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				0-00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 – 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 – 0xFC	-	-	-	-

1.21 LVD

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	-	LVD_CTL [B,H,W]	
			100000–000011--	
0x004	-	-	LVD_STR [B,H,W]	
			0------1 0------1	
0x008	-	-	LVD_CLR [B,H,W]	
			1------- 1-------	
0x00C	LVD_RLR [W]			
	---------- 00000000 00000001			
0x010	-	-	LVD_STR2 [B,H,W]	
			0------- 01------	
0x014	-	-	LVD_CTL2 [B,H,W]	
			-------0 000011--	
0x018	-	-	-	LVD2_CTL [B,H,W]
				000011--
0x01C	-	-	LVD2_CTL2 [B,H,W]	
			0------0 000011--	
0x020 – 0x0FC	-	-	-	-

1.22 DS Mode

DS Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	REG_CTL [B,H,W]
				---- -10-
0x004	-	-	-	RCK_CTL [B,H,W]
				---- --01
0x008	-	-	-	REG_CTL2 [B,H,W]
				---- -011
0x00C	-	-	-	-
0x010	-	-	-	MOSC_CTL [B,H,W]
				---- -10-
0x014 – 0x0FC	-	-	-	-
0x100	-	-	-	CAL_CTL [B,H,W]
				---- 0000
0x104	-	-	-	CAL_SET [B,H,W]
				---1 0001
0x108	CAL_KEY [W]			
	00000000 00000000 00000000 00000001			
0x10C – 0x1FC	-	-	-	-
0x200 – 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL [B,H,W]
				---- ---0
0x704	-	-	-	WRFSR [B,H,W]
				---- --00
0x708	-	-	WIFSR [B,H,W]	
			00000000 00000000	
0x70C	-	-	WIER [B,H,W]	
			00000000 00000-00	
0x710	-	-	WILVR [B,H,W]	
			-----000 00000000	
0x714	-	-	-	DSRAMR [B,H,W]
				---- --00
0x718 – 0x71C	-	-	-	-
0x720	-	-	-	STBFLASHPDX [B,H,W]
				---- ---0
0x724	RST_MSK [W]			
	00000000 00000000 -------- -------0			
0x728 – 0x7FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
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	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	UCCR [B,H,W]
				-000 0000
0x004	-	-	-	UPCR [B,H,W]
				---- --00
0x008	-	-	-	UPCR2 [B,H,W]
				---- -000
0x00C	-	-	-	UPCR3 [B,H,W]
				---0 0000
0x010	-	-	-	UPCR4 [B,H,W]
				-011 1011
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]
				---- 0001
0x028	-	-	-	UPCR6 [B,H,W]
				---- 0010
0x02C	-	-	-	UP_CR7 [B,H,W]
				---- ---0
0x030	-	-	-	USBEN0 [B,H,W]
				---- -100
0x034	-	-	-	USBEN1 [B,H,W]
				---- -100
0x038 – 0xFFC	-	-	-	-

1.24 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

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0–00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	RDR/TDR[H,W]			
	00000000 00000000 00000000 00000000			
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/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000–0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x044	-	-	FTICR2[B,H,W] 00000000	FTICR1[B,H,W] 00000000
0x048 – 0x0FC	-	-	-	-

Note:

- *RDR/TDR register's higher 16 bits can be accessed by word operation in MFS-I^{2}S mode.*

1.25 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			

1.26 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 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 – 0xFFC	-	-	-	-

1.27 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

Base_Address + Address	Register			
	+3	+2	+1	+0
0x14C	-	-	-	WTTR0[B,H,W] 00000000
0x150	-	-	-	WTTR1[B,H,W] 00000000
0x154	-	-	-	WTTR2[B,H,W] ------00
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_CLKDIV[B,H,W] 00000111
0x178	-	-	-	WTOSCCNT[B,H,W] ------01
0x17C	-	-	-	CCS[B,H,W] 11001110
0x180	-	-	-	CCB[B,H,W] 11001110
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

Base_Address + Address	Register			
	+3	+2	+1	+0
0x1A8	-	-	-	VBDIR[B,H,W] ----0000
0x1AC	-	-	-	VBDOR[B,H,W] ----1111
0x0B0	-	-	-	VBPZR[B,H,W] ------11
0x1B4-1FF	-	-	-	-
0x200	BREG03[B,H,W] 00000000	BREG02[B,H,W] 00000000	BREG01[B,H,W] 00000000	BREG00[B,H,W] 00000000
0x204	BREG07[B,H,W] 00000000	BREG06[B,H,W] 00000000	BREG05[B,H,W] 00000000	BREG04[B,H,W] 00000000
0x208	BREG0B[B,H,W] 00000000	BREG0A[B,H,W] 00000000	BREG09[B,H,W] 00000000	BREG08[B,H,W] 00000000
0x20C	BREG0F[B,H,W] 00000000	BREG0E[B,H,W] 00000000	BREG0D[B,H,W] 00000000	BREG0C[B,H,W] 00000000
0x210	BREG13[B,H,W] 00000000	BREG12[B,H,W] 00000000	BREG11[B,H,W] 00000000	BREG10[B,H,W] 00000000
0x214	BREG17[B,H,W] 00000000	BREG16[B,H,W] 00000000	BREG15[B,H,W] 00000000	BREG14[B,H,W] 00000000
0x218	BREG1B[B,H,W] 00000000	BREG1A[B,H,W] 00000000	BREG19[B,H,W] 00000000	BREG18[B,H,W] 00000000
0x21C	BREG1F[B,H,W] 00000000	BREG1E[B,H,W] 00000000	BREG1D[B,H,W] 00000000	BREG1C[B,H,W] 00000000
0x220	BREG23[B,H,W] 00000000	BREG22[B,H,W] 00000000	BREG21[B,H,W] 00000000	BREG20[B,H,W] 00000000
0x224	BREG27[B,H,W] 00000000	BREG26[B,H,W] 00000000	BREG25[B,H,W] 00000000	BREG24[B,H,W] 00000000
0x228	BREG2B[B,H,W] 00000000	BREG2A[B,H,W] 00000000	BREG29[B,H,W] 00000000	BREG28[B,H,W] 00000000
0x22C	BREG2F[B,H,W] 00000000	BREG2E[B,H,W] 00000000	BREG2D[B,H,W] 00000000	BREG2C[B,H,W] 00000000
0x230	BREG33[B,H,W] 00000000	BREG32[B,H,W] 00000000	BREG31[B,H,W] 00000000	BREG30[B,H,W] 00000000
0x234	BREG37[B,H,W] 00000000	BREG36[B,H,W] 00000000	BREG35[B,H,W] 00000000	BREG34[B,H,W] 00000000
0x238	BREG3B[B,H,W] 00000000	BREG3A[B,H,W] 00000000	BREG39[B,H,W] 00000000	BREG38[B,H,W] 00000000
0x23C	BREG3F[B,H,W] 00000000	BREG3E[B,H,W] 00000000	BREG3D[B,H,W] 00000000	BREG3C[B,H,W] 00000000
0x240	BREG43[B,H,W] 00000000	BREG42[B,H,W] 00000000	BREG41[B,H,W] 00000000	BREG40[B,H,W] 00000000
0x244	BREG47[B,H,W] 00000000	BREG46[B,H,W] 00000000	BREG45[B,H,W] 00000000	BREG44[B,H,W] 00000000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x248	BREG4B[B,H,W] 00000000	BREG4A[B,H,W] 00000000	BREG49[B,H,W] 00000000	BREG48[B,H,W] 00000000
0x24C	BREG4F[B,H,W] 00000000	BREG4E[B,H,W] 00000000	BREG4D[B,H,W] 00000000	BREG4C[B,H,W] 00000000
0x250	BREG53[B,H,W] 00000000	BREG52[B,H,W] 00000000	BREG51[B,H,W] 00000000	BREG50[B,H,W] 00000000
0x254	BREG57[B,H,W] 00000000	BREG56[B,H,W] 00000000	BREG55[B,H,W] 00000000	BREG54[B,H,W] 00000000
0x258	BREG5B[B,H,W] 00000000	BREG5A[B,H,W] 00000000	BREG59[B,H,W] 00000000	BREG58[B,H,W] 00000000
0x25C	BREG5F[B,H,W] 00000000	BREG5E[B,H,W] 00000000	BREG5D[B,H,W] 00000000	BREG5C[B,H,W] 00000000
0x260	BREG63[B,H,W] 00000000	BREG62[B,H,W] 00000000	BREG61[B,H,W] 00000000	BREG60[B,H,W] 00000000
0x264	BREG67[B,H,W] 00000000	BREG66[B,H,W] 00000000	BREG65[B,H,W] 00000000	BREG64[B,H,W] 00000000
0x268	BREG6B[B,H,W] 00000000	BREG6A[B,H,W] 00000000	BREG69[B,H,W] 00000000	BREG68[B,H,W] 00000000
0x26C	BREG6F[B,H,W] 00000000	BREG6E[B,H,W] 00000000	BREG6D[B,H,W] 00000000	BREG6C[B,H,W] 00000000
0x270	BREG73[B,H,W] 00000000	BREG72[B,H,W] 00000000	BREG71[B,H,W] 00000000	BREG70[B,H,W] 00000000
0x274	BREG77[B,H,W] 00000000	BREG76[B,H,W] 00000000	BREG75[B,H,W] 00000000	BREG74[B,H,W] 00000000
0x278	BREG7B[B,H,W] 00000000	BREG7A[B,H,W] 00000000	BREG79[B,H,W] 00000000	BREG78[B,H,W] 00000000
0x27C	BREG7F[B,H,W] 00000000	BREG7E[B,H,W] 00000000	BREG7D[B,H,W] 00000000	BREG7C[B,H,W] 00000000
0x280-0xFFC	-	-	-	-

1.28 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
0x000 – 0x0FC	-	-	-	-

1.29 Peripheral Clock Gating

Peripheral Clock Gating Base_Address : 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1–11 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	------00 ----0000 00000000 00000000			
0x008 – 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000 ----0000 ----0000			
0x018 – 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- ---1-1-1 1111-1-- --00–00			
0x024	MRST2[B,H,W]			
	-------- ---0-0-0 0000-0-- --00–00			
0x028 – 0x0FC	-	-	-	-

1.30 Smart Card I/F

Smart Card I/F ch.0Base_Address : 0x4003_C900

Smart Card I/F ch.1Base_Address : 0x4003_C980

Base_Address + Address	Register			
	+3	+2	+1	+0
0x00	-	-	GLOBALCONTROL1[H,W] -0001000 00000000	
0x04	-	-	STATUS[H,W] --000000 00000001	
0x08	-	-	PORTCONTROL[H,W] 0000–00 00-0-0-0	
0x0C	-	-	DATA[H,W] -------0 00000000	
0x10	-	-	CARDCLOCK [H,W] 00000000 00101000	
0x14	-	-	BAUDRATE[H,W] 0000001 01110100	
0x18	-	-	GUARDTIMER[H,W] -------- 00000000	
0x1C	-	-	IDLETIMER[H,W] 00000000 00000000	
0x20	-	-	GLOBALCONTROL2[H,W] -------- ----1-00	
0x24	-	-	DATA_FIFO[H,W] -------0 00000000	
0x28	-	-	FIFO_LEVEL_READ[H,W] 00000000 00000000	
0x2C	-	-	FIFO_LEVEL_WRITE[H,W] 00000000 00000000	
0x30	-	-	FIFO_MODE[H,W] 00000000 ----0000	
0x34	-	-	FIFO_CLEAR_MSB_WRITE[H,W] -------- -------0	
0x38	-	-	FIFO_CLEAR_MSB_ READ[H,W] -------- -------0	
0x3C	-	-	-	-
0x40	-	-	IRQ_STATUS[H,W] -------- 00000000	
0x44- 0x7C	-	-	-	-

1.31 MFSI2S

MFSI2S ch.5 Base_Address : 0x4003_CA00

MFSI2S ch.6 Base_Address : 0x4003_CA80

Base_Address	Register			
+Address	+3	+2	+1	+0
0x00	-		CNTLREG [H,W] -------- -------- -----000 00000001	
0x04	-		I2SCLK [H,W] -------- -------- 000----- 00000000	
0x08	-		I2SST [B] ------00	I2SRST[B] 00000000

1.32 High Resilience

High Resilience Base_Address : 0x4003_E000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	RTR_CTL3 [B,H,W]	RTR_CTL2 [B,H,W]	RTR_CTL1 [B,H,W]	RTR_CTL0 [B,H,W]
	000- 000-	000- ----	---- ----	1111 1111
0x004	RTR_RTS3 [B,H,W]	RTR_RTS2 [B,H,W]	RTR_RTS1 [B,H,W]	RTR_RTS0 [B,H,W]
	1111 1111	1111 1111	1111 1111	1111 1111
0x008	RTR_TGS3 [B,H,W]	RTR_TGS2 [B,H,W]	RTR_TGS1 [B,H,W]	RTR_TGS0 [B,H,W]
	1111 1111	1111 1111	1111 1111	1111 1111
0x00C	RTR_STR3 [B,H,W]	RTR_STR2 [B,H,W]	RTR_STR1 [B,H,W]	RTR_STR0 [B,H,W]
	00-- ----	---- ----	00-- ----	---- ----
0x010	RTR_RLR [W]			
	00000000 00000000 00000000 00000000			
0x014	RTR_CT23 [B,H,W]	RTR_CT22 [B,H,W]	RTR_CT21 [B,H,W]	RTR_CT20 [B,H,W]
	0000 0000	0000 0000	0000 0000	---0 ---0
0x018	RTR_REV [B,H,W]			
	00000000 00010101 00000001 00000000			
0x01C – 0xFFC	-	-	-	-

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] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 00000000 00000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------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] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	
0x2148	-	-	EP0OS[H,W] 100–00- -XXXXXXX	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
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] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 – 0x217C	-	-	-	-

1.34 DSTC

DSTC Base_Address : 0x4006_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x00C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x018-0x02C	-			
0x030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x038-0x04C	-			
0x050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x058-0x06C	-			
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x078-0x08C	-			
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x098-0xFFC	-			

1.35 MTB_DWT

MTB_DWT Base_Address : 0xF000_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 – 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF8	CID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.36 Fast GPIO

Fast GPIO Base_Address : 0xF800_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	FPDIR7[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024	-	-	FPDIR9[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x028	-	-	FPDIRA[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x02C	-	-	FPDIRB[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x030	-	-	FPDIRC[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x034	-	-	FPDIRD[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	FPDIRF[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	FPDOR7[B,H,W]	
			00000000 00000000	
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064	-	-	FPDOR9[B,H,W]	
			00000000 00000000	
0x068	-	-	FPDORA[B,H,W]	
			00000000 00000000	
0x06C	-	-	FPDORB[B,H,W]	
			00000000 00000000	
0x070	-	-	FPDORC[B,H,W]	
			00000000 00000000	
0x074	-	-	FPDORD[B,H,W]	
			00000000 00000000	
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	FPDORF[B,H,W]	
			00000000 00000000	
0x080	-	-	-	M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088	-	-	-	M_FPDIR2[B,H,W]
				XXXXXXXX
0x08C	-	-	-	M_FPDIR3[B,H,W]
				XXXXXXXX
0x090	-	-	-	M_FPDIR4[B,H,W]
				XXXXXXXX
0x094	-	-	-	M_FPDIR5[B,H,W]
				XXXXXXXX

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x098	-	-	-	M_FPDIR6[B,H,W]
				XXXXXXXX
0x09C	-	-	-	M_FPDIR7[B,H,W]
				XXXXXXXX
0x0A0	-	-	-	M_FPDIR8[B,H,W]
				XXXXXXXX
0x0A4	-	-	-	M_FPDIR9[B,H,W]
				XXXXXXXX
0x0A8	-	-	-	M_FPDIRA[B,H,W]
				XXXXXXXX
0x0AC	-	-	-	M_FPDIRB[B,H,W]
				XXXXXXXX
0x0B0	-	-	-	M_FPDIRC[B,H,W]
				XXXXXXXX
0x0B4	-	-	-	M_FPDIRD[B,H,W]
				XXXXXXXX
0x0B8	-	-	-	M_FPDIRE[B,H,W]
				XXXXXXXX
0x0BC	-	-	-	M_FPDIRF[B,H,W]
				XXXXXXXX
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8	-	-	-	M_FPDOR2[B,H,W]
				00000000
0x0CC	-	-	-	M_FPDOR3[B,H,W]
				00000000
0x0D0	-	-	-	M_FPDOR4[B,H,W]
				00000000
0x0D4	-	-	-	M_FPDOR5[B,H,W]
				00000000
0x0D8	-	-	-	M_FPDOR6[B,H,W]
				00000000
0x0DC	-	-	-	M_FPDOR7[B,H,W]
				00000000
0x0E0	-	-	-	M_FPDOR8[B,H,W]
				00000000
0x0E4	-	-	-	M_FPDOR9[B,H,W]
				00000000
0x0E8	-	-	-	M_FPDORA[B,H,W]
				00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0EC	-	-	-	M_FPDORB[B,H,W]
				00000000
0x0F0	-	-	-	M_FPDORC[B,H,W]
				00000000
0x0F4	-	-	-	M_FPDORD[B,H,W]
				00000000
0x0F8	-	-	-	M_FPDORE[B,H,W]
				00000000
0x0FC	-	-	-	M_FPDORF[B,H,W]
				00000000
0x100 – 0xFFC	-	-	-	-

D. Register Map (TYPE3-M0+)

This chapter shows the register map.

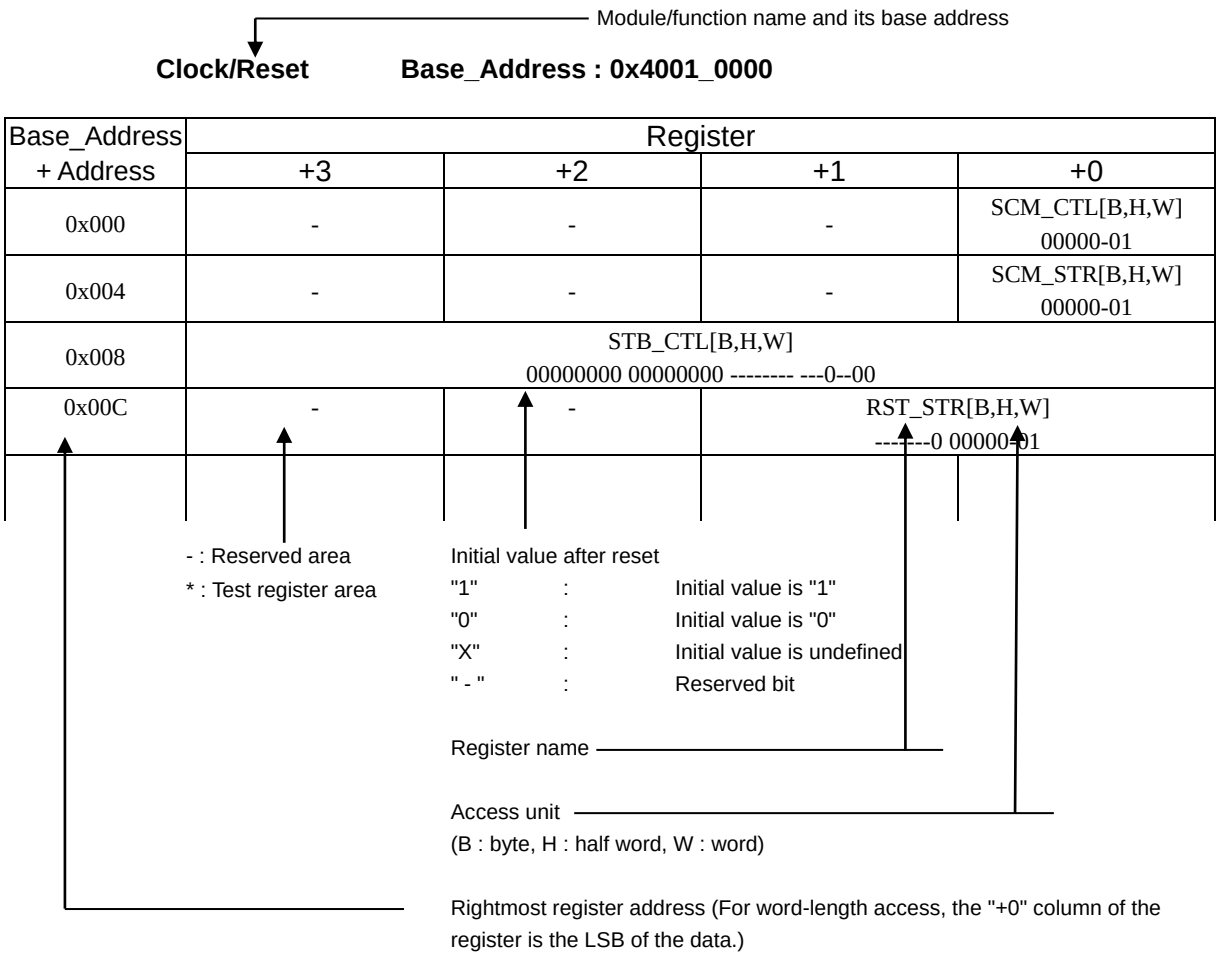
1. Register Map

CODE: 9AFREGMAP-E03.0

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.
- When the register is accessed by larger unit than register size, for the reserved area to access at the same time, the read value is undefined, and writing is invalid.

1.1 Flash I/F

Flash I/F Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
	-------- -------- -------- -----011			
0x008	FSTR[B,H,W]			
	-------- -------- -------- --00000X			
0x00C	-	-	-	-
0x010	FSYNDN[B,H,W]			
	-------- -------- -------- ----0001			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
	-------- -------- -------- ------00			
0x024	FISR[B,H,W]			
	-------- -------- -------- ------00			
0x028	FICLR[B,H,W]			
	-------- -------- -------- ------00			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x104 - 0x1FC	-	-	-	-

Note:

- *For details of Flash I/F registers, refer to "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 00000-01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
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	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address	Register			
+ Address	+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 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW WDT

SW WDT Base_Address : 0x4001_2000

Base_Address	Register			
+ Address	+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	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 Dual Timer

Dual Timer Base_Address : 0x4001_5000

Base_Address	Register			
+ Address	+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	-	-	-	-

1.7 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_Address	Register			
+ Address	+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	-	-	-	-

1.8 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) Base Address : 0x4002_5100

Base_Address	Register			
+ Address	+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	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simultaneous Startup (Base Timer) Base Address : 0x4002_5F00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR [B,H,W] -------- XXXXXXXX	

1.9 A/DC

12-bit A/DC unit0 Base_Address : 0x4002_7000

Base_Address	Register			
+ Address	+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	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			00000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.10 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	-	-	MCR_PSR[B,H,W]
				-----001
0x004	-	-	MCR_FTRM[B,H,W]	
			------10 00000110	
0x008	-	-	-	MCR_TTRM[B,H,W]
				-1111111
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.11 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+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[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C	ELVR2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x020	-	-	-	NMIENR[B,H,W]
	-	-	-	-------0
0x024 – 0x0FC	-	-	-	-

1.12 INT-Req. READ

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 – 0x004	-	-	-	-
0x008	VIR_OFFSET[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	-	-	-	ODDPKS[B,H,W]
				---00000
0x014 – 0x1FC	-	-	-	-
0x200	EXC02MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x204	IRQ00MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x208	IRQ01MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x20C	IRQ02MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x210	IRQ03MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x214	IRQ04MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x218	IRQ05MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x21C	IRQ06MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x220	IRQ07MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x224	IRQ08MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x228	IRQ09MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x22C	IRQ10MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x230	IRQ11MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x234	IRQ12MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x238	IRQ13MON[B,H,W]			
	00000000 00000000 00000000 00000000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x23C	IRQ14MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x240	IRQ15MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x244	IRQ16MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x248	IRQ17MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x24C	IRQ18MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x250	IRQ19MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x254	IRQ20MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x258	IRQ21MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x25C	IRQ22MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x260	IRQ23MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x264	IRQ24MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x268	IRQ25MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x26C	IRQ26MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x270	IRQ27MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x274	IRQ28MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x278	IRQ29MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x27C	IRQ30MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x280	IRQ31MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x284 – 0xFFC	-	-	-	-

1.13 GPIO

GPIO Base_Address : 0x4003_3000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
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	-	-	-	-
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024 – 0x034	-	-	-	-
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C – 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
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 – 0x134	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C – 0x1FC	-	-	-	-
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	-	-	-	-
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224 – 0x234	-	-	-	-
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C – 0x2FC	-	-	-	-
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	-	-	-	-
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324 – 0x334	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C – 0x3FC	-	-	-	-
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	-	-	-	-
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424 – 0x434	-	-	-	-
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x43C – 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	1111 1111 1111 1111 1111 1111 1111 1111			
0x504 – 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---0 0101			
0x584 – 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- --00 ---- --01 ---- --0- ---- --00			
0x604 – 0x60C	-	-	-	-
0x610	EPFR04[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x614	EPFR05[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
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 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
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 – 0x654	-	-	-	-
0x658	EPFR22[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x65C	EPFR23[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x660 – 0x678	-	-	-	-
0x67C	EPFR31[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x680	-	-	-	-
0x684	EPFR33[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x688 – 0x690	-	-	-	-
0x694	EPFR37[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x698	EPFR38[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x69C – 0x6FC	-	-	-	-
0x700	-	-	-	-
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	-	-	-	-
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710 – 0x714	-	-	-	-
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C – 0x7FC	-	-	-	-
0x800 – 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	-	-	-	-
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924 – 0x934	-	-	-	-
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C – 0xFFC	-	-	-	-

1.14 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	Register			
+ Address	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				000000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				0-00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 – 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 – 0xFC	-	-	-	-

1.15 LVD

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	LVD_CTL [B,H,W]	
			10000000 00001100	
0x004	-	-	LVD_STR [B,H,W]	
			00000000 0000000-	
0x008	-	-	LVD_CLR [B,H,W]	
			00000000 10000000	
0x00C	LVD_RLR [W]			
	00000000 00000000 00000000 00000001			
0x010	-	-	LVD_STR2 [B,H,W]	
			00000000 01000000	
0x014 – 0x0FC	-	-	-	-

1.16 DS Mode

DS Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	REG_CTL [B,H,W]
				---- -10-
0x004	-	-	-	RCK_CTL [B,H,W]
				---- --01
0x008 – 0x00C	-	-	-	-
0x010	-	-	-	MOSC_CTL [B,H,W]
				---- -10-
0x014 – 0x0FC	-	-	-	-
0x100	-	-	-	CAL_CTL [B,H,W]
				---- 0000
0x104	-	-	-	-
0x108	CAL_KEY [W]			
	00000000 00000000 00000000 00000001			
0x10C – 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL [B,H,W]
				---- ---0
0x704	-	-	-	WRFSR [B,H,W]
				---- --00
0x708	-	-	WIFSR [B,H,W]	
			00000000 00000000	
0x70C	-	-	WIER [B,H,W]	
			00000000 00000-00	
0x710	-	-	WILVR [B,H,W]	
			-----000 00000000	
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 – 0x8FC	-	-	-	-

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x900	WIOLC_CTL [B,H,W]			
	-------0 -------1 -------0 -------0			
0x904	-	-	-	SUBOSC_CTL[B,H,W]
				------01
0x908	-	-	-	CEC_CTL [B,H,W]
				----0000
0x90C	-	-	-	DEBUG_SW_CTL[B,H,W]
				-------1
0x910 – 0xEFC	-	-	-	-

1.17 USB Clock

USB Clock Base_Address : 0x4003_6000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	UCCR [B,H,W]
				-----000
0x004 – 0x024	-	-	-	-
0x028	-	-	-	UPCR6[B,H,W]
				----0010
0x02C	-	-	-	-
0x030	-	-	-	USBEN0[B,H,W]
				-------0
0x038 – 0x0FC	-	-	-	-

1.18 I2CSLAVE

I2CSLAVE ch.6 Base_Address : 0x4003_7980

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	IBSCR[B,H,W]		IBSSR[B,H,W]	
	------00 0-000000		-----001 00000000	
0x04	-	IBSDSTUPR[B,H,W]	IBSMSKR[B,H,W]	IBSADR[B,H,W]
	-	11111111	01111111	00000000
0x08	-	-	-	IBSTDR[B,H,W]
	-	-	-	11111111
0x0C	-	-	-	IBSRDR[B,H,W]
	-	-	-	11111111
0x10	-	-	IBSSCR[B,H,W]	
	-	-	-----0-- -----00-	
0x14	-	-	IBSSSR[B,H,W]	
	-	-	-------0 --------	
0x18 – 0x3F	-	-	-	-

1.19 MFS

MFS ch.0 Base_Address : 0x4003_8000

MFS ch.1 Base_Address : 0x4003_8100

MFS ch.3 Base_Address : 0x4003_8300

MFS ch.4 Base_Address : 0x4003_8400

MFS ch.6 Base_Address : 0x4003_8600

MFS ch.7 Base_Address : 0x4003_8700

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0–00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	RDR/TDR[H,W]			
	00000000 00000000 00000000 00000000			
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/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000–0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
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
0x044 – 0x0FC	-	-	-	-

Note:

- *RDR/TDR register's higher 16 bits can be accessed by word operation in I^2S mode.*

1.20 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			

1.21 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 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 – 0xFFC	-	-	-	-

1.22 RTC

RTC Base_Address : 0x4003_B000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WTCR1[B,H,W]			
	00000000 00000000 ---00000 -00000-0			
0x004	WTCR2[B,H,W]			
	-------- -------- -----000 -------0			
0x008	WTBR[B,H,W]			
	-------- 00000000 00000000 00000000			
0x00C	WTDR[B,H,W]	WTHR[B,H,W]	WTMIR[B,H,W]	WTSR[B,H,W]
	--000000	--000000	-0000000	-0000000
0x010	-	WTYR[B,H,W]	WTMOR[B,H,W]	WTDW[B,H,W]
		00000000	---00000	-----000
0x014	ALDR[B,H,W]	ALHR[B,H,W]	ALMIR[B,H,W]	-
	--000000	--000000	-0000000	
0x018	-	ALYR[B,H,W]	ALMOR[B,H,W]	-
		00000000	---00000	
0x01C	WTTR[B,H,W]			
	-------- ------00 00000000 00000000			
0x020	-	-	WTCLKM[B,H,W]	WTCLKS[B,H,W]
			------00	-------0
0x024	-	WTCALEN[B,H,W]	WTCAL[B,H,W]	
		-------0	------00 00000000	
0x028	-	-	WTDIVEN[B,H,W]	WTDIV[B,H,W]
			------00	----0000
0x02C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x030	-	-	-	WTCOSEL[B,H,W]
				-------0
0x034 – 0xFFC	-	-	-	-

1.23 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
0x000 – 0x0FC	-	-	-	-

1.24 Peripheral Clock Gating

Peripheral Clock Gating Base_Address : 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1---- -------1 -------- 11-11-11			
0x004	MRST0[B,H,W]			
	-------- -------0 -------- 00-00-00			
0x008 – 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- -------- -------- ------11			
0x014	MRST1[B,H,W]			
	-------- -------- -------- ------00			
0x018 – 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -----1–111----- -------0			
0x024	MRST2[B,H,W]			
	-------- -----0–000----- -------0			
0x028 – 0x0FC	-	-	-	-

1.25 Smart Card I/F

Smart Card I/F ch.1 Base_Address : 0x4003_C980

Base_Address + Address	Register			
	+3	+2	+1	+0
0x00	-	-	GLOBALCONTROL1[H,W] -0001000 00000000	
0x04	-	-	STATUS[H,W] --000000 00000001	
0x08	-	-	PORTCONTROL[H,W] 0000–00 00-0-0-0	
0x0C	-	-	DATA[H,W] -------0 00000000	
0x10	-	-	CARDCLOCK [H,W] 00000000 00101000	
0x14	-	-	BAUDRATE[H,W] 0000001 01110100	
0x18	-	-	GUARDTIMER[H,W] -------- 00000000	
0x1C	-	-	IDLETIMER[H,W] 00000000 00000000	
0x20	-	-	GLOBALCONTROL2[H,W] -------- ----1-00	
0x24	-	-	DATA_FIFO[H,W] -------0 00000000	
0x28	-	-	FIFO_LEVEL_READ[H,W] 00000000 00000000	
0x2C	-	-	FIFO_LEVEL_WRITE[H,W] 00000000 00000000	
0x30	-	-	FIFO_MODE[H,W] 00000000 ----0000	
0x34	-	-	FIFO_CLEAR_MSB_WRITE[H,W] -------- -------0	
0x38	-	-	FIFO_CLEAR_MSB_ READ[H,W] -------- -------0	
0x3C	-	-	-	-
0x40	-	-	IRQ_STATUS[H,W] -------- 00000000	
0x44- 0x7C	-	-	-	-

1.26 MFSI2S

MFSI2S ch.4 Base_Address : 0x4003_CA00

MFSI2S ch.6 Base_Address : 0x4003_CA80

Base_Address + Address	Register			
	+3	+2	+1	+0
0x00	-	-	CNTLREG[B, H,W] -----0-0 -0000-01	
0x04	-	-	I2SCLK[B, H,W] 00------ 00000000	
0x08	-	-	I2SST[B,H,W] ------00	I2SRST[B,H,W] 00000000
0x0C- 0x3C	-	-	-	-

1.27 USB

USB ch.0 Base_Address : 0x4004_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 00000000 00000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------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] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	
0x2148	-	-	EP0OS[H,W] 100–00- -XXXXXXX	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
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] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 – 0x217C	-	-	-	-

1.28 DSTC

DSTC Base_Address : 0x4006_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x00C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x018-0x02C				
0x030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x038-0x04C				
0x050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x058-0x06C				
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x078-0x08C				
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x098 – 0xFFC	-	-	-	-

1.29 MTB_DWT

MTB_DWT Base_Address : 0xF000_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 – 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF8	CID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.30 Fast GPIO

Fast GPIO Base_Address : 0xF800_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	-	-
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024 – 0x034	-	-	-	-
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	-	-
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064 – 0x074	-	-	-	-
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	-	-
0x080				M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088 – 0x0BF	-			
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8 – 0x0FC	-	-	-	-

1.31 VIR

VIR Base_Address : 0xF800_0100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	VIR00[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x004	VIR01[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	VIR02[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x00C	VIR03[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	VIR04[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x014	VIR05[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x018	VIR06[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x01C	VIR07[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x020	VIR08[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x024	VIR09[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x028	VIR10[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x02C	VIR11[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	VIR12[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x034	VIR13[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x038	VIR14[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x03C	VIR15[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x040	VIR16[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x044	VIR17[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	VIR18[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x04C	VIR19[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x050	VIR20[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x054	VIR21[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x058	VIR22[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x05C	VIR23[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x060	VIR24[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x064	VIR25[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x068	VIR26[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x06C	VIR27[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x070	VIR28[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x074	VIR29[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x078	VIR30[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x07C	VIR31[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

E. List of Notes

This section explains notes for each function.

1. Notes when High-speed CR Is Used for Master Clock

CODE: 9APRECAUTION-FM0-E03.0

1. Notes when High-speed CR Is Used for 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	The maximum frequency of the high-speed CR shall not exceed the upper limit of the internal operation clock frequency specified in the "Data Sheet" of the product used.
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 Tim	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" of the product used.
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.
Multi Function Serial Interface	UART	Because of the frequency change of the High-Speed CR, the Board rate set may turn worse more. In case of over range of board rate error, this Function/Mode cannot be used.
	CSIO, I2C, MFS-I2S	The frequency variation of the high-speed CR should be considered for the communication of each macro.
	LIN	As the required frequency accuracy cannot be met, this function cannot be used as master. As slave, this function can be used. 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.
Smart-card interface	-	Because of the frequency change of the High-Speed CR, the Board rate set may turn worse more. In case of over range of board rate error, this Function/Mode cannot be used.

F. Major Changes

Spansion Publication Number: MN710-00004

Page	Section	Changes
Revision 1.0		
-	-	Initial release
Revision 2.0		
6	The target product in this manual	Added TYPE2-M0+ products.
18	CHAPTER1-2:UART 1 Overview of UART	Added Description to the Hardware flow control(*3).
372	CHAPTER2-1:CAN Prescaler 1 Overview and configuration	Added Frequency
600	CHAPTER5-2:USB host registers 5.4 Host Status Register	Added note.
Revision 3.0		
6	The target product in this manual	Added TYPE3-M0+ products.
18	C HAPTER1-1: Multi-function Serial Interface 1. Overview of the Multi-Function Serial Interface	Edit " Transmission/Reception FIFO"
19	CHAPTER1-2: UART (Asynchronous Serial Interface)	Added " The external clock is input from SCK." Revised Table 4-1
241	CHAPTER1-5: I2C Interface (I2C Communications Control Interface)	Revised Table 2-1, Figure 4-2 Added "5.6 Noise Filter Control Register (NFCR)" Added Table 5-4
495	CHAPTER4-1:USB Clock Generation Block Overview	Added this chapter newly.
497	CHAPTER4-2:USB Clock Generation Block (A)	Renamed this chapter. Original is CHAPTER4-1:USB Clock Generation Block
517	CHAPTER4-3:USB Clock Generation Block (B)	Added this chapter newly.
653	CHAPTER6:Smart Card Interface 5.Smart Card Interface Setting Procedure and Program Flow	Revised Figure 5-2, Figure 5-3
683	CHAPTER7:I2CSLAVE	Added this chapter newly.
805	D. Register Map (TYPE3-M0+)	Added this chapter newly.

NOTE: Please see “Revision History” about later revised information.

Revision History

Document Revision History

Document Title:	32-Bit Microcontroller FM0+ Family Peripheral Manual Communication Macro Part
Document Number:	002-05024

Revision	ECN No.	Origin of Change	Description of Change
**	-	TOYO	New Specification
*A	5340921	KEMU	Updated to Cypress format. Updated description of FBYTE register of CHAPTER 1-3: CSIO on page 77. Updated section structure of I2C Interface Operation of CHAPTER 1-5: I2C Interface on page 265. Updated description of CHAPTER 1-6: MFS-I2S on page 373. Removed contents due to unsupported function of CHAPTER 2-1: CAN Prescaler on page 409. Removed contents due to unsupported function of CHAPTER 2-2: CAN Controller on page 411. Updated Section 3.3 of CHAPTER 3-2: CEC Reception/Remote Reception on page 429. Updated Table 1-2 of APPENDIX Product Type List on page 710. Updated description of APPENDIX B Register Map (TYPE1-M0+) on page 711. Updated description of APPENDIX C Register Map (TYPE2-M0+) on page 765. Updated description of APPENDIX D Register Map (TYPE3-M0+) on page 829.
*B	5742745	AESATMP8	Updated logo and Copyright.
*C	6014665	HTER	P.3 Added the URL for "Microcontroller support information" P.7 Modified the part numbers in Table 1, 2, 3 from 10 digits to 8 digits discription. P.710 CHAPTER1: Modified the part numbers in "1. Product TYPE List" from 10 digits to 8 digits description. P.876 APPENDIX E: Deleted the discription “CAN” in the table of "Notes on each macro".