User's Manual

Unified Serial Interface

User's Manual

Multimedia Processor for Mobile Applications
EMMA Mobile TM EV2

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

Renesas Electronics
www.renesas.com

Rev.6.00 Dec 2011

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.

The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

> Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

> The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the EMMA Mobile EV2. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	Description	Document Title	Document No.
Data Sheet	Hardware overview and electrical characteristics	EMMA Mobile EV2 Datasheet	R19DS0010EJxxxx
User’s manual (1chip)	Hardware whole specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description	EMMA Mobile EV2 User’s manual 1chip	R19UH0036EJxxxx
User’s manual (module)	Hardware each macro specifications and operation description.	EMMA Mobile EV2 User’s manual each module	See below

Document Name	Document No.	Document Name	Document No.
1chip	R19UH0036EJxxxx	DMA Controller	R19UH0043EJxxxx
System Management Unit	R19UH0037EJxxxx	LP-DDR/DDR2 Controller	R19UH0039EJxxxx
Timer	R19UH0054EJxxxx	SD Memory Card Interface	R19UH0061EJxxxx
System Timer	R19UH0055EJxxxx	SDIO Interface	R19UH0042EJxxxx
HD Video Decoder	R19UH0056EJxxxx	CF Card Interface	R19UH0062EJxxxx
Rotator	R19UH0057EJxxxx	Unified Serial Interface (This manual)	R19UH0047EJxxxx
Resizer	R19UH0058EJxxxx	UART interface	R19UH0040EJxxxx
Image Composer	R19UH0038EJxxxx	USB 2.0 Host Controller	R19UH0045EJxxxx
LCD Interface	R19UH0044EJxxxx	USB 2.0 Function Contoller	R19UH0034EJxxxx
ITU-R BT.656 Interface	R19UH0059EJxxxx	IIC Interface	R19UH0052EJxxxx
Digital Terrestrial TV Interface	R19UH0048EJxxxx	General Purpose I/O Interface	R19UH0041EJxxxx
Camera Interface	R19UH0060EJxxxx	Pulse Width Modulation Interface	R19UH0063EJxxxx

4 digits of end shows the version.

2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples the PM03 bit in the PM0 register
P3_5 pin, VCC pin

(2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b or 11
Hexadecimal: EFA0h
Decimal: 1234

3. Register Notation

The symbols and terms used in register diagrams are described below.

x.x.x XXX register

This register (XXXXXXX: xxxx_xxxxh) …..

7	6	5	4	3	2	1	0
Reserved		CHG_P1_LAT	LATCH_P1_SEL	Reserved		CHG_P0_LAT	CHG_P0_LAT_SEL

Name	R/W	Bit No.	After Reset	Description
LATCH_P2_SEL	R/W	8	0	0: Use the P2_LAT bit of the P2_POWERSW register in the SMU to latch data. 1: Use the CHG_P2_LAT bit to latch data.
Reserved	R	7:6	–	Reserved. If these bits are read, 0 is returned for each bit.
CHG_P1_LAT	R/W	5	1	0: Output data as is. 1: Output latched data.
LATCH_P1_SEL	R/W	4	0	0: Use the P1_LAT bit of the P1_POWERSW register in the SMU to latch data. 1: Use the CHG_P1_LAT bit to latch data.
Reserved	R	3:2	–	Reserved. If these bits are read, 0 is returned for each bit.
CHG_P0_LAT	R/W	1	1	0: Output data as is. 1: Output latched data.
CHG_P0_LAT_SEL	R/W	0	0	0: Use the P0_LAT bit of the P2_POWERSW register in the SMU to latch data. 1: Use the CHG_P0_LAT bit to latch data.

*1 *3 *2

*1

R/W: Read and Write.

R: Read only.

W: Write only.

-: Nothing is assigned.

*2

Reserved bit.

Reserved bit. Set to specified value.

*3

- Nothing is assigned.

  Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.

- Do not set to a value.

  Operation is not guaranteed when a value is set.

- Function varies according to the operating mode.

  The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

4. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communication Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Register
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

Abbreviation	Full Form
AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
FIFO	First In, First Out
GPIO	General Purpose I/O
PCM	Pulse Code Modulation
SIO	Serial Input / Output
SPI	System Packet Interface
USI	Unified Serial Interface

All trademarks and registered trademarks are the property of their respective owners.

EMMA Mobile is registered trademark of Renesas Electronics Corporation in Japan, USA and other countries.

Table of Contents

Unified Serial Interface

EMMA Mobile EV2

R19UH0047EJ0600
Rev.6.00
Dec 21, 2011

1. Overview

The USI is a general-purpose serial interface module has audio serial interface functions, and SPI interface functions. The major USI features are shown below.
The USI supports a 4-wire serial interface SPI and a serial interface that handles audio and voice codecs.

1.1 Features

1.1.1 SPI interface

- Transfer modes
  The SPI macro can be set to CPU control mode or DMA control mode when running as the master and when running as a slave.
  - CPU control mode: In this mode, transfer ends after one frame of data has been transmitted or received, or transmitted-and-received.
  - DMA control mode: In this mode, data can be transmitted and received continuously using two 32-bit × 32-word FIFO buffers (one for transmission and one for reception).
    Transfer stops when the CPU issues a STOP command or according to the status of the FIFO buffers.
- The SCLK polarity (positive or inverted) and delay (half an SCLK clock cycle) can be specified for each CS signal.
- The number of bits to transfer can be specified (between 8 and 32 bits).
- Interrupts can be generated.
- Clock request signals (SIO_DMA_CLKREQ, SIO_PCLKREQ, and SIO_CTLCLKREQ) can be generated.
- Transfer can be stopped automatically (in DMA mode).
  - SPI transmission is stopped if the transmission FIFO buffer becomes empty while the DMA master is transmitting data.
  - SPI reception is stopped if the amount of data received matches the DMA transfer data length specified by SIOx_DMA_LENG while the DMA master is receiving data. If SIOx_DMA_LENG is set to 0, however, transfer continues until the CPU issues a STOP command.
- CS signals can be fixed by specifying a register setting.
- The I/O phases of SI/SO can be switched.

1.1.2 Audio (PCM) interface

- Seven operating modes can be specified (modes 0 to 6).
- Data is transferred using DMA. The PCM macro includes two 32-bit × 32-word FIFO buffers (one for transmission and one for reception).
- The control bus clock (SIO_PCLK), serial clock (SIO_SCLK), and AHB bus clock can be automatically controlled.
- The PCM macro can handle periods when the bus clock is not supplied, such as when the bus clock frequency is switched.
- Can be used to interface with an AC97 controller.
- It is now selectable whether to adjust frames to the right or left when specifying the data transfer format.
- SIO_CS0 is used for sync signal I/O, and SIO_CS1 is used for AC97 reset signal output.

- SIO_CS2 to SIO_CS6 are fixed to 0 (not used).

1.1.3 FIFO buffers for data

- For transmission: 32 bits × 32-word FIFO buffer
- For reception: 32 bits × 32-word FIFO buffer

Restriction

Unspecified data is sometimes output just before the send data by a Tx transmission of audio (PCM).
Refer to 5.2 chapter Audio mode about an avoidance method of this case.

The condition that this case occurs is indicated on below.
When transmission execution (PCM_TXRX_EN.TX_EN='1') is done in the following setting condition.
－mode0＋Left adjust（PCM_FUNC_SEL.MODE_SEL=”000”＋PCM_FUNC_SEL.LR_AJUST=’0’）
－mode1＋Left adjust（PCM_FUNC_SEL.MODE_SEL=”001”＋PCM_FUNC_SEL.LR_AJUST=’0’）
－mode2 （PCM_FUNC_SEL.MODE_SEL=”010”）
－mode3 （PCM_FUNC_SEL.MODE_SEL=”011”）
－mode5 （PCM_FUNC_SEL.MODE_SEL=”101”）
－mode6 （PCM_FUNC_SEL.MODE_SEL=”110”）

1.2 Block Diagram

Figure 1-1. SIO Module Block Diagram

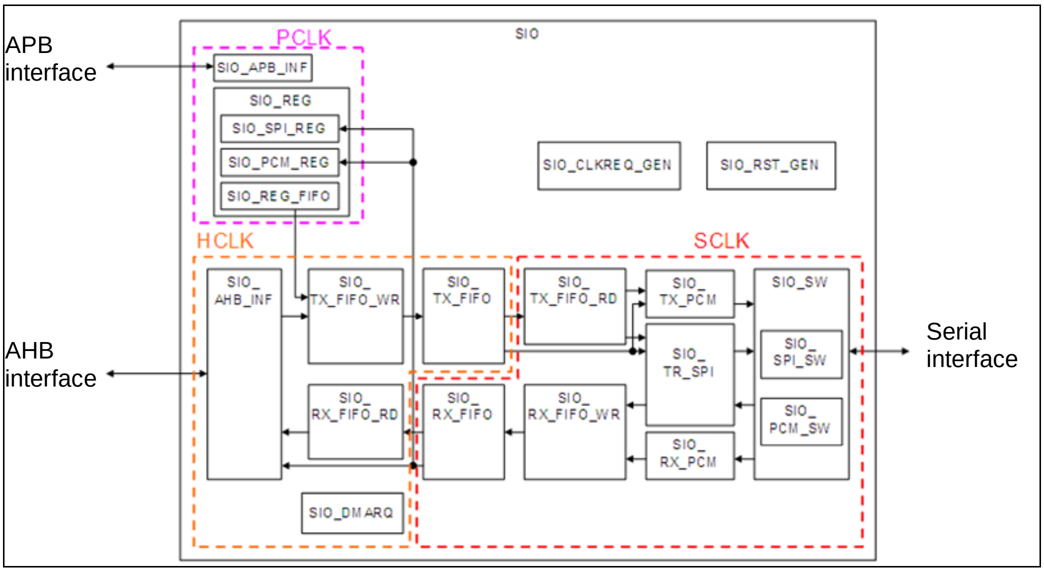


Figure 1-2. USIA Module Block Diagram

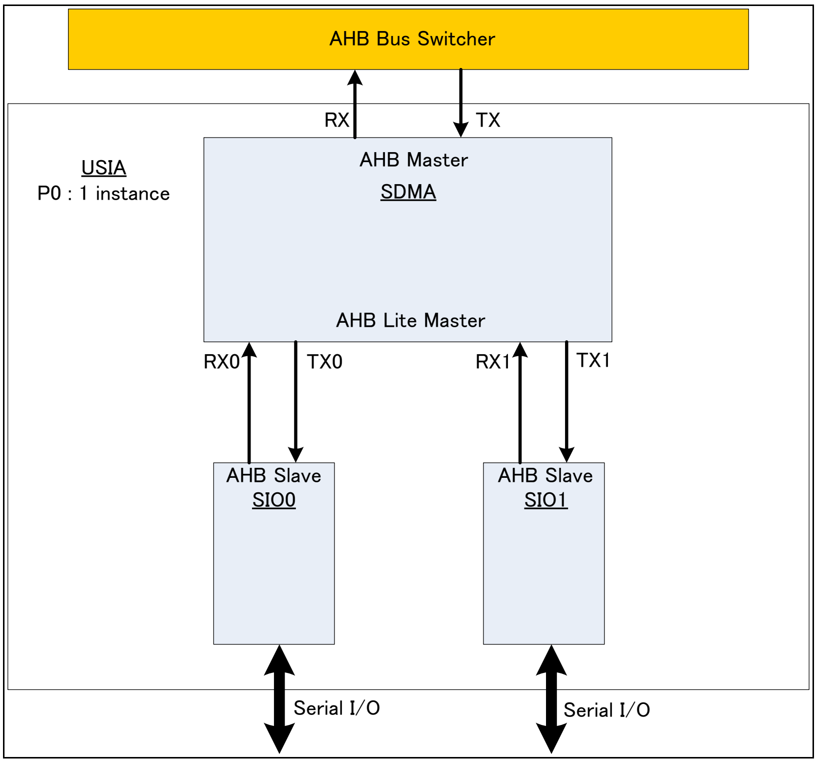
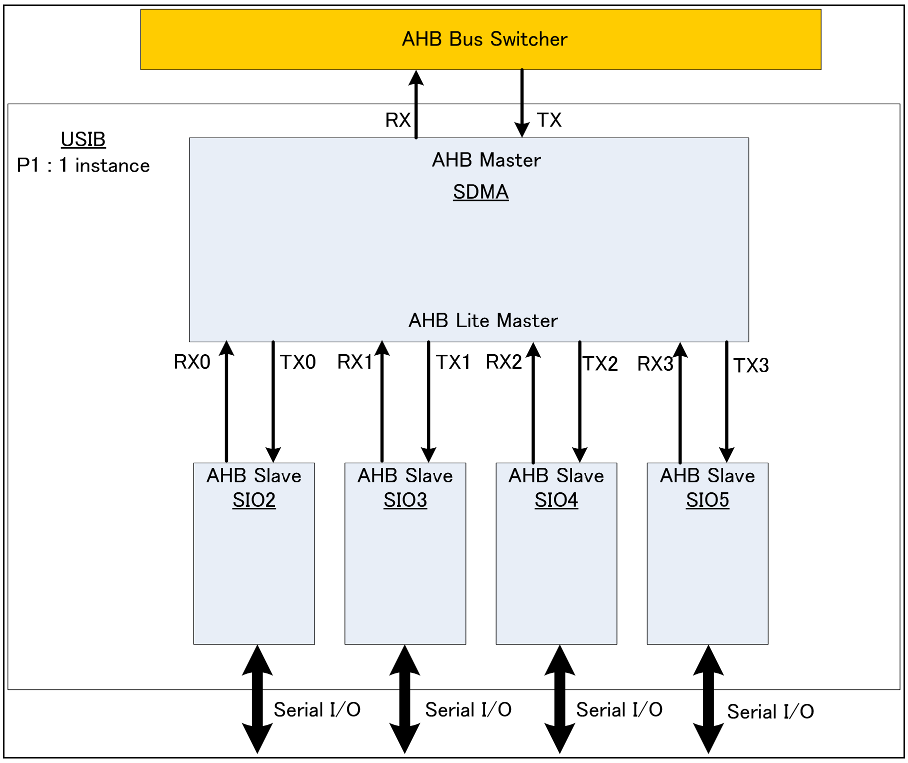


Figure 1-3. USIB Module Block Diagram



USI	Power domain	SIO number	CS	Default communication mode	Interrupt number (INTA)
USIA	P0	SIO0	CS0～6	SPI	44
		SIO1	CS0	PCM	45
USIB	P1	SIO2	CS0～2	SPI	46
		SIO3	CS0	PCM	47
		SIO4	CS0～1	SPI	48
		SIO5	CS0～4 Note1 CS0～2 Note2	SPI	49

Note1 Alternate function pin with a HSI.
Note2 Alternate function pin with a AB.

2. Pin Functions

1.8V/3.3V system

Pin Name	I/O	Function	Alternate Pin
USI0_DI	Input	Port 0 data input	
USI0_DO	Output	Port 0 data output	
USI0_CLK	I/O	Port 0 clock	
USI0_CS6	Output	Port 0 CS6	USI3_CS0
USI0_CS5	Output	Port 0 CS5	USI3_DO
USI0_CS4	Output	Port 0 CS4	USI3_DI
USI0_CS3	Output	Port 0 CS3	USI3_CLK
USI0_CS2	I/O	Port 0 CS2	GPIO_106
USI0_CS1	I/O	Port 0 CS1	GPIO_105
USI0_CS0	I/O	Port 0 CS0	
USI1_DI	I/O	Port 1 data input	
USI1_DO	I/O	Port 1 data output	
USI1_CLK	I/O	Port 1 clock	
USI1_CS0	I/O	Port 1 CS0	
USI2_DI	I/O	Port 2 data input	
USI2_DO	I/O	Port 2 data output	
USI2_CLK	I/O	Port 2 clock	
USI2_CS2	I/O	Port 2 CS2	GPIO_114
USI2_CS1	I/O	Port 2 CS1	GPIO_113
USI2_CS0	I/O	Port 2 CS0	
USI3_DI	I/O	Port 3 data input	
USI3_DO	I/O	Port 3 data output	
USI3_CLK	I/O	Port 3 clock	
USI3_CS0	I/O	Port 3 CS0	
USI4_DI	Input	Port 4 data input	GPIO_120
USI4_DO	Output	Port 4 data output	GPIO_121
USI4_CLK	I/O	Port 4 clock	GPIO_119
USI4_CS1	Output	Port 4 CS1	GPIO_114
USI4_CS0	I/O	Port 4 CS0	GPIO_113
USI5_DI	Input	Port 5 data input	GPIO_086, GPIO_150
USI5_DO	Output	Port 5 data output	GPIO_087, GPIO_144
USI5_CLK	I/O	Port 5 clock	GPIO_085, GPIO_143
USI5_CS4	Output	Port 5 CS4	GPIO_149
USI5_CS3	Output	Port 5 CS3	GPIO_148
USI5_CS2	Output	Port 5 CS2	GPIO_090, GPIO_147
USI5_CS1	Output	Port 5 CS1	GPIO_089, GPIO_146
USI5_CS0	I/O	Port 5 CS0	GPIO_088, GPIO_145

3. Registers

3.1 Register List

The SPI interface registers can only be accessed in word units.
Do not access reserved registers.
Any value written to reserved bits in each register is ignored.

Base address:	SIO0: E012_0000H	SIO1: E001_0000H
	SIO2: E10C_0000H	SIO3: E10D_0000H
	SIO4: E10E_0000H	SIO5: E10F_0000H

(1) Mode setting registers

Address	Register Name	Register Symbol	R/W	After Reset
0000H	Mode selection enable register	SIO_SWITCH_EN	R/W	0000_0000H
0004H	Mode selection register	SIO_MODE_SWITCH	R/W	0000_0000H
0008H to 0FFCH	Reserved	–	–	–

(2) SPI registers

Address	Register Name	Register Symbol	R/W	After Reset
1000H	SPI mode register	SPI_MODE	R/W	0000_0002H
1004H	SPI polarity register	SPI_POL	R/W	0000_7000H
1008H	SPI control register	SPI_CONTROL	R/W	0000_8040H
100CH	Reserved	–	–	–
1010H	SPI transmission data register	SPI_TX_DATA	W	0000_0000H
1014H	SPI reception data register	SPI_RX_DATA	R	0000_00xxH
1018H	SPI interrupt status register	SPI_STATUS	R	0000_0000H
101CH	SPI interrupt raw status register	SPI_RAW_STATUS	R	0000_0000H
1020H	SPI interrupt enable set register	SPI_ENSET	R/W	0000_0000H
1024H	SPI interrupt enable clear register	SPI_ENCLR	W	0000_0000H
1028H	SPI interrupt source clear register	SPI_FFCLR	W	0000_0000H
102CH	Reserved	–	–	–
1030H	SPI reception FIFO pointer register	SPI_RX_FIFO_P	R	0000_0000H
1034H	SPI control register 2	SPI_CONTROL2	R/W	0000_0000H
1038H	SPI CS fixed value setting register	SPI_TIECS	R/W	0000_0000H
103CH to 1FFCH	Reserved	–	–	–

(3) Audio registers

Address	Register Name	Register Symbol	R/W	After Reset
2000H	PCM operating mode setting register	PCM_FUNC_SEL	R/W	0000_0000H
2004H	PCM data transfer enable set register	PCM_TXRX_EN	R/W	0000_0000H
2008H	PCM data transfer enable clear register	PCM_TXRX_DIS	W	0000_0000H
200CH	PCM data transfer cycle setting register	PCM_CYCLE	R/W	0000_0000H
2010H	PCM interrupt raw status register	PCM_RAW	R	0000_0000H
2014H	PCM interrupt status register	PCM_STATUS	R	0000_0000H
2018H	PCM interrupt enable set register	PCM_ENSET	R/W	0000_0000H
201CH	PCM interrupt enable clear register	PCM_ENCLR	W	–
2020H	PCM interrupt clear register	PCM_CLEAR	W	–
2024H	PCM transmission data register	PCM_TXQ	R/W	0000_0000H
2028H	PCM reception data register	PCM_RXQ	R	xxxx_xxxxH
202CH	PCM FIFO counter register	PCM_FIFO_P	R	0000_0000H
2030H	PCM data transfer cycle setting register 2	PCM_CYCLE2	R/W	0000_0000H
2034H	Reserved	–	–	–
2038H	PCM AC97 reset assert register	PCM_AC97_RESET	R/W	0000_0000H
2040H to FFFCH	Reserved	–	–	–

3.2 Register Details

3.2.1 Mode selection enable register

This register (SIO_SWITCH_EN: xxxx_0000H) specifies whether to enable selection of the SIO operating mode.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved							SWCHEN

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:1	–	Reserved. If these bits are read, 0 is returned for each bit.
SWCHEN	R/W	0	0	Specify whether to enable or disable changing the SIO_MODE_SWITCH register settings. 0: Disable 1: Enable

3.2.2 Mode register

This register (SIO_MODE_SWITCH: xxxx_0004H) specifies the SIO operating mode.
Use the registers shown in **3.1 (2) SPI registers** while in the SPI mode, and use those shown in **3.1 (3) Audio registers** while in the audio (PCM) mode.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved			TX_BURST	MODE			

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:5	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_BURST	R/W	4	0	Select the AHB burst transfer type. 0: Single-burst transfer, 1: 8-burst transfer
MODE	R/W	3:0	Note	Select the transfer mode for the CPU side. 0000: Unused macro 0001: SPI mode 0010: Audio mode 0011 to 1111: Reserved

Note SIO0:0001H, SIO1:0010H, SIO2:0001H, SIO3:0010H, SIO4:0001H, SIO5:0001H

3.2.3 SPI mode register

This register (SPI_MODE: xxxx_1000H) specifies the operating mode of the SPI module.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved		CK_PHASE	NB_A				

7	6	5	4	3	2	1	0
CS_SELA[3:0]				Reserved		M_S	DMA

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:14	–	Reserved. If these bits are read, 0 is returned for each bit.
CK_PHASE	R/W	13	0	0: Normal mode 1: Phase change mode (changing to using a common phase of SCLK for transmission and reception.)
NB_A	R/W	12:8	00H	Specify the number of transferred bits[Note]. (NB_A value + 1) bits are transferred. Specifiable value: 7 to 31
CS_SELA[3:0]	R/W	7:4	0H	Select the chip select signal used for transfer. 0000: CS0 0001: CS1 0010: CS2 0011: CS3 0100: CS4 0101: CS5 0110: CS6 0111 to 1111 Non selected
Reserved	R	3:2	–	Reserved. If these bits are read, 0 is returned for each bit.
M_S	R/W	1	1	Specify the operating mode. 0: Master mode 1: Slave mode
DMA	R/W	0	0	Specify the operating mode. 0: CPU mode 1: DMA mode

Note Set the NB_A bit to a value of 7 or more. If a value of less than 7 is specified, data is transferred in the same way as when 7 is specified. The mode cannot be changed during operation.

CS0-6 is different in the terminal output number depending on SIO.

[USIA]

SIO0 : CS0 – 6

SIO1 : CS0

[USIB]

SIO2 : CS0 – 2

SIO3 : CS0

SIO4 : CS0 – 1

SIO5 (HSI sharing terminal) : CS0 – 4

SIO5 (AB sharing terminal) : CS0 – 2

M_S=1 (Slave Mode) operate by input to CS0.

Anything but CS0 can't be used as a slave.

3.2.4 SPI polarity register

This register (SPI_POL: xxxx_1004H) selects the polarity of SCLK and CS (chip select) signals and specifies the SPI_CS inactive period for contiguous transfer (CSW), using the number of SPI_SCLK clock cycles.
This register is used to change the CS output when the CS output is fixed (set to 1) in the CS fixed value setting register.

31	30	29	28	27	26	25	24
Reserved							CK6_DLY

23	22	21	20	19	18	17	16
CK6_POL	CS6_POL	CK5_DLY	CK5_POL	CS5_POL	CK4_DLY	CK4_POL	CS4_POL

15	14	13	12	11	10	9	8
CSW				CK3_DLY	CK3_POL	CS3_POL	CK2_DLY

7	6	5	4	3	2	1	0
CK2_POL	CS2_POL	CK1_DLY	CK1_POL	CS1_POL	CK0_DLY	CK0_POL	CS0_POL

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:25	–	Reserved. If these bits are read, 0 is returned for each bit.
CK6_DLY	R/W	24	0	0: No delay clock is specified. 1: A delay clock is specified.
CK6_POL	R/W	23	0	0: Positive 1: Negative
CS4_POL	R/W	22	0	0: Positive 1: Negative
CK5_DLY	R/W	21	0	0: No delay clock is specified. 1: A delay clock is specified.
CK5_POL	R/W	20	0	0: Positive 1: Negative
CS5_POL	R/W	19	0	0: Positive 1: Negative
CK4_DLY	R/W	18	0	0: No delay clock is specified. 1: A delay clock is specified.
CK4_POL	R/W	17	0	0: Positive 1: Negative
CS4_POL	R/W	16	0	0: Positive 1: Negative
CSW	R/W	15:12	7H	Specify the width of the SPI_CS inactive period during master transfer. 1 to 16 SPI_SCLK clocks[Note] Interrupts such as END occur after the CSW cycles.

(2/2)

Name	R/W	Bit No.	After Reset	Description
CK3_DLY	R/W	11	0	0: No delay clock is specified. 1: A delay clock is specified.
CK3_POL	R/W	10	0	0: Positive 1: Negative
CS3_POL	R/W	9	0	0: Positive 1: Negative
CK2_DLY	R/W	8	0	0: No delay clock is specified. 1: A delay clock is specified.
CK2_POL	R/W	7	0	0: Positive 1: Negative
CS2_POL	R/W	6	0	0: Positive 1: Negative
CK1_DLY	R/W	5	0	0: No delay clock is specified. 1: A delay clock is specified.
CK1_POL	R/W	4	0	0: Positive 1: Negative
CS1_POL	R/W	3	0	0: Positive 1: Negative
CK0_DLY	R/W	2	0	0: No delay clock is specified. 1: A delay clock is specified.
CK0_POL	R/W	1	0	0: Positive 1: Negative
CS0_POL	R/W	0	0	0: Positive 1: Negative

Note The CSW width is set to a value equivalent to (*specified value + 1*) SPI_SCLK cycles.

3.2.5 SPI control register

This register (SPI_CONTROL: xxxx_1008H) controls the SPI module.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
TX_EMP	RX_FULL	Reserved					RST

7	6	5	4	3	2	1	0
TX_FULL	RX_EMP	Reserved		WRT	RD	STOP	START

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:16	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_EMP	R	15	1	0: One or more valid data items exist in the transmission FIFO buffer. 1: The transmission FIFO buffer is empty.
RX_FULL	R	14	0	0: One or more writable areas exist in the reception FIFO buffer. 1: 32-word data is stored in the reception FIFO buffer.
Reserved	R	13:9	–	Reserved. If these bits are read, 0 is returned for each bit.
RST	R/W	8	0	0: Cancel the reset. 1: Reset the SPI module. Some register settings are not reset.
TX_FULL	R	7	0	0: There are writable areas in the transmission FIFO buffer. 1: 32-word data is stored in the transmission FIFO buffer.
RX_EMP	R	6	1	0: Data exists in the reception FIFO buffer. 1: The reception FIFO buffer is empty.
Reserved	R	5:4	–	Reserved. If these bits are read, 0 is returned for each bit.
WRT	R/W	3	0	0: Disable transmission. 0 is always output to SPI_SO. 1: Enable transmission. In CPU mode: Data stored in TX_DATA is output to SPI_SO. In DMA mode: TX_DMAREQ is issued if there is available space in the transmission FIFO buffer.
RD	R/W	2	0	0: Disable reception. No data is written to the reception FIFO buffer. 1: Enable reception. In CPU mode: Generate an RDV interrupt each time a frame is received. In DMA mode: RX_DMAREQ is issued if there is valid data in the reception FIFO buffer.

(2/2)

Name	R/W	Bit No.	After Reset	Description
STOP	W	1	0	0: Ignored 1: Terminate DMA transfer. If this bit is read, 0 is returned. Setting is prohibited when DMA transfer is not performed.
START	W	0	0	0: Ignored 1: Start transfer. (Ignored during slave operation.)
	R			0: Transfer has been stopped. 1: Transfer is in progress. (Ignored during slave operation.)

Cautions 1. Writing 1 to the RST bit executes a software reset, and writing 0 cancels the software reset.

To execute a synchronous reset during the software reset period, the SPI module outputs PCLKREQ and HCLKREQ. After the software reset, wait for at least five SPI_SCLK cycles and then use RCLKREQ and HCLKREQ to cancel the software reset.

The following registers are reset by setting the RST bit.

- **SPI_CONTROL (except for the RST bit)**
- **SPI_STATUS**
- **SPI_RAW_STATUS**
- **TX_FIFO_P**
- **RX_FIFO_P**

If a software reset is executed during SPI transmission or reception, the current transfer is aborted. At this time, frames being transferred are not guaranteed.

2. The STOP bit can be specified only in DMA mode. Setting the STOP bit to 1 is prohibited in a mode other than DMA mode.

If the STOP bit is set to 1 while transfer is stopped, DMA transfer is stopped immediately. If the STOP bit is set to 1 while transfer is being executed, DMA transfer will be stopped after the current transfer ends. When setting the STOP bit to 1, do not change the settings of the WRT and RD bits that were specified when transfer started.

Example To stop transfer that was started by setting the SPI_CONTROL register to 0D in the DMA transmission/reception mode, set the SPI_CONTROL register to 0E.

3. When restarting transfer after setting the STOP bit to 1, make sure that START is set to 0 (transfer has been stopped).

4. Changing the setting of the RD or WRT bit is prohibited during transfer.

5. Reset cancellation (RST = 0) and transfer start (START = 1) must not be specified at the same time.

6. Transfer must not be started (START = 1) while the WRT and RD bits are set to 0.

3.2.6 SPI transmission data register

This register (SPI_TX_DATA: xxxx_1010H) stores the data to be written in the transmission FIFO buffer. If this register is read, 0 is returned for each bit. Store data in LSB order. If the NB_A bit of the SPI_MODE register is set to 7, store transmission data in the TX_DATA[7:0] bits.

31	30	29	28	27	26	25	24
TX_DATA[31:24]							

23	22	21	20	19	18	17	16
TX_DATA[23:16]							

15	14	13	12	11	10	9	8
TX_DATA[15:8]							

7	6	5	4	3	2	1	0
TX_DATA[7:0]							

Name	R/W	Bit No.	After Reset	Description
TX_DATA[31:0]	W	31:0	0000_0000H	Transmission data register

Caution Don't write data to the transmission FIFO buffer while it is full, because the written data will be lost.

3.2.7 SPI reception data register

This register (SPI_RX_DATA: xxxx_1014H) stores the data read from the reception FIFO buffer. Writing to this register is ignored. Received data is stored in LSB order. If the NB_A bit of the SPI_MODE register is set to 7, data is stored in the RX_DATA[7:0] bits.

31	30	29	28	27	26	25	24
RX_DATA[31:24]							

23	22	21	20	19	18	17	16
RX_DATA[23:16]							

15	14	13	12	11	10	9	8
RX_DATA[15:8]							

7	6	5	4	3	2	1	0
RX_DATA[7:0]							

Name	R/W	Bit No.	After Reset	Description
RX_DATA[31:0]	R	31:0	0000_00xx	Reception data register

Caution Don't read the reception FIFO buffer while it is empty, because invalid data will be read.

3.2.8 SPI interrupt status register

This read-only register (SPI_STATUS: xxxx_1018H) is used to read the status of the interrupt sources enabled by the interrupt enable set register (SPI_ENSET).

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	TX_STOP	RX_STOP	TERR	RDV	END	TX_UDR	RX_OVR

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:7	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_STOP	R	6	0	Indicates that data is no longer stored in the transmission FIFO buffer after transmission starts. This flag is valid only during a DMA fixed-length transfer.
RX_STOP	R	5	0	Indicates that the amount of received data has reached the value specified for the RX_FIFO_FULL bit of the SPI_CONTROL2 register. This flag is valid only during a DMA fixed-length transfer.
TERR	R	4	0	Indicates that the number of SPI_SCLK_I cycles does not match the value specified for the NB_A bit of the SPI_MODE register. This flag is not set if the number of SPI_SCLK_I cycles is a multiple of the value specified for the NB_A bit.
RDV	R	3	0	Indicates that reception of one frame is complete on the CPU side in the CPU mode.
END	R	2	0	Indicates that transmission and reception of one frame is complete on the CPU side in the CPU mode.
TX_UDR	R	1	0	Indicates that an underrun has occurred in the transmission FIFO buffer.
RX_OVR	R	0	0	Indicates that an overrun has occurred in the reception FIFO buffer.

Cautions 1. The interrupts END, and RDV occur in the CPU mode.

2. The interrupts RX_STOP, and TX_STOP occur in DMA mode.

3. The interrupts TERR, TX_UDR, and RX_OVR occur only in slave mode.

4. In the DMA transmission/reception master mode, no transmission FIFO underrun interrupt or reception FIFO overrun interrupt occurs because transmission and reception stop when a condition on either side is satisfied.

5. In the DMA transmission/reception slave mode, a transmission FIFO underrun interrupt occurs if CLK/CS is input from the master before storing transmission data in the transmission FIFO buffer, and a reception FIFO overrun interrupt occurs if CLK/CS is input from the master when the reception FIFO buffer is full.

The sources for generating and clearing each interrupt that can be issued for the SPI interface are as follows:

Bit	Interrupt Type	Description	
6	TX_STOP (Transmission stopped)	Generation source	When the number of data items stored in the transmission FIFO buffer becomes 0 during DMA fixed length transmission (specified by setting the TX_STOP_MODE bit of the SPI_CONTROL2 register to 1)
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 6 of the SPI interrupt source clear register (SPI_FFCLR)
5	RX_STOP (Reception stopped)	Generation source	When the number of received data items matches the value specified for the RX_FIFO_FULL bit of the SPI_CONTROL2 register during DMA fixed length reception (specified by setting the RX_STOP_MODE bit of the SPI_CONTROL2 register to 1) or when the reception FIFO buffer becomes full
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 5 of the SPI interrupt source clear register (SPI_FFCLR)
4	TERR (Data length inconsistency error)	Generation source	When the value specified for the NB_A bit of the SPI_MODE register is inconsistent with the number of clocks SPI_SCLK_I while in the slave mode (This interrupt is not generated if the number of clocks is an integer multiple of NB_A.)
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 4 of the SPI interrupt source clear register (SPI_FFCLR)
3	RDV (Received 1 frame)	Generation source	When 1 frame is received in the CPU mode
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 3 of the SPI interrupt source clear register (SPI_FFCLR)
2	END (Transmitted and received 1 frame)	Generation source	When 1 frame is transmitted and received in the CPU mode
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 2 of the SPI interrupt source clear register (SPI_FFCLR)
1	TX_UDR (transmission underrun error)	Generation source	When an underrun occurs for the transmission FIFO buffer while in the slave mode
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 1 of the SPI interrupt source clear register (SPI_FFCLR)
0	RX_OVR (reception overrun error)	Generation source	When an overrun occurs for the reception FIFO buffer while in the slave mode
		Clearing method	Triggering a soft reset (by writing 1 to the RST bit of the SPI_CONTROL register) or writing 1 to bit 0 of the SPI interrupt source clear register (SPI_FFCLR)

Caution When an interrupt is generated, only an interrupt report is sent. No hardware processing such as transfer stopping operation is executed.

3.2.9 SPI interrupt raw status register

This read-only register (SPI_RAW_STATUS: xxxx_101CH) is used read the status of interrupt sources regardless of the setting of the interrupt enable set register (SPI_ENSET).

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	TX_STOP_ RAW	RX_STOP_ RAW	TERR_RAW	RDV_RAW	END_RAW	TX_UDR_ RAW	RX_OVR_ RAW

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:7	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_STOP_RAW	R	6	0	Indicates that data is no longer stored in the transmission FIFO buffer after transmission starts. This flag is valid only during a DMA fixed-length transfer.
RX_STOP_RAW	R	5	0	Indicates that the amount of received data has reached the value specified for the RX_FIFO_FULL bit of the SPI_CONTROL2 register. This flag is valid only during a DMA fixed-length transfer.
TERR_RAW	R	4	0	Indicates that the number of SPI_SCLK_I cycles does not match the value specified for the NB_A bit of the SPI_MODE register. This flag is not set if the number of SPI_SCLK_I cycles is a multiple of the value specified for the NB_A bit.
RDV_RAW	R	3	0	Indicates that reception of one frame on the CPU side is complete in the CPU mode.
END_RAW	R	2	0	Indicates that transmission and reception of one frame on the CPU side is complete in the CPU mode.
TX_UDR_RAW	R	1	0	Indicates that an underrun has occurred in the transmission FIFO buffer.
RX_OVR_RAW	R	0	0	Indicates that an overrun has occurred in the reception FIFO buffer.

Cautions

1. **The interrupts TERR_RAW, TX_UDR_RAW, RX_OVR_RAW, END_RAW, and RDV_RAW occur in the CPU mode.**
2. **The interrupts TERR_RAW, TX_UDR_RAW, RX_OVR_RAW, RX_STOP_RAW, and TX_STOP_RAW occur in DMA mode.**
3. **The interrupts TERR_RAW, TX_UDR_RAW, and RX_OVR_RAW occur only in slave mode.**

3.2.10 SPI interrupt enable set register

This register (SPI_ENSET: xxxx_1020H) enables the issuance of interrupt requests. When the bit corresponding to an interrupt source is set to 1 in this register, the interrupt source is enabled, an interrupt request is issued, and the corresponding bit in the interrupt status register (SPI_STATUS) is set to 1. Writing 0 to this register does not affect the setting. Read this register to check whether issuance of interrupt requests is enabled.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	TX_STOP_EN	RX_STOP_EN	TERR_EN	RDV_EN	END_EN	TX_UDR_EN	RX_OVR_EN

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:7	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_STOP_EN	R	6	0	Indicates whether issuance of the TX_STOP interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the TX_STOP interrupt. 0: Ignored 1: Enable
RX_STOP_EN	R	5	0	Indicates whether issuance of the RX_STOP interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the RX_STOP interrupt. 0: Ignored 1: Enable
TERR_EN	R	4	0	Indicates whether issuance of the TERR interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the TERR interrupt. 0: Ignored 1: Enable

(2/2)

Name	R/W	Bit No.	After Reset	Description
RDV_EN	R	3	0	Indicates whether issuance of the RDV interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the RDV interrupt. 0: Ignored 1: Enable
END_EN	R	2	0	Indicates whether issuance of the END interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the END interrupt. 0: Ignored 1: Enable
TX_UDR_EN	R	1	0	Indicates whether issuance of the TX_UDR interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the TX_UDR interrupt. 0: Ignored 1: Enable
RX_OVR_EN	R	0	0	Indicates whether issuance of the RX_OVR interrupt request is enabled. 0: Not enabled. 1: Enabled.
	W			Enable issuance of the RX_OVR interrupt. 0: Ignored 1: Enable

3.2.11 SPI interrupt enable clear register

This register (SPI_ENCLR: xxxx_1024H) disables the issuance of interrupt requests. When the bit corresponding to an interrupt source is set to 1 in the register, an interrupt request is not issued even if the interrupt source occurs. The status of the bit corresponding to the interrupt status register (SPI_STATUS) also does not change. Writing 0 to this register does not affect the setting.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	TX_STOP_MASK	RX_STOP_MASK	TERR_MASK	RDV_MASK	END_MASK	TX_UDR_MASK	RX_OVR_MASK

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:7	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_STOP_MASK	W	6	0	0: Ignored. 1: Disable issuance of the TX_STOP interrupt request.
RX_STOP_MASK	W	5	0	0: Ignored. 1: Disable issuance of the RX_STOP interrupt request.
TERR_MASK	W	4	0	0: Ignored. 1: Disable issuance of the TERR interrupt request.
RDV_MASK	W	3	0	0: Ignored. 1: Disable issuance of the RDV interrupt request.
END_MASK	W	2	0	0: Ignored. 1: Disable issuance of the END interrupt request.
TX_UDR_MASK	W	1	0	0: Ignored. 1: Disable issuance of the TX_UDR interrupt request.
RX_OVR_MASK	W	0	0	0: Ignored. 1: Disable issuance of the RX_OVR interrupt request.

3.2.12 SPI interrupt source clear register

This write-only register (SPI_FFCLR: xxxx_1028H) clears interrupt sources. When the bit corresponding to an interrupt source is set to 1 in the register, the interrupt source is cleared. Writing 0 to this register does not affect the setting.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	TX_STOP_CLR	RX_STOP_CLR	TERR_CLR	RDV_CLR	END_CLR	TX_UDR_CLR	RX_OVR_CLR

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:7	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_STOP_CLR	W	6	0	0: Ignored. 1: Clear the source of the TX_STOP interrupt.
RX_STOP_CLR	W	5	0	0: Ignored. 1: Clear the source of the RX_STOP interrupt.
TERR_CLR	W	4	0	0: Ignored. 1: Clear the source of the TERR interrupt.
RDV_CLR	W	3	0	0: Ignored. 1: Clear the source of the RDV interrupt.
END_CLR	W	2	0	0: Ignored. 1: Clear the source of the END interrupt.
TX_UDR_CLR	W	1	0	0: Ignored. 1: Clear the source of the TX_UDR interrupt.
RX_OVR_CLR	W	0	0	0: Ignored. 1: Clear the source of the RX_OVR interrupt.

3.2.13 SPI reception FIFO pointer register

This register (SPI_RX_FIFO_P: xxxx_1030H) is used to reference the read and write pointers for the transmission FIFO buffer.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved			RX_FIFO_WP				

7	6	5	4	3	2	1	0
Reserved			RX_FIFO_RP				

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:13	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_FIFO_WP	R	12:8	00H	Use this to reference the reception FIFO buffer write pointer.
Reserved	R	7:5	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_FIFO_RP	R	4:0	00H	Use this to reference the reception FIFO buffer read pointer.

3.2.14 SPI control register 2

This register (SPI_CONTROL2: xxxx_1034H) controls fixed-length transfer in the DMA master mode. The settings of fixed-length transfer are invalid in a mode other than the DMA master mode.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
RX_FIFO_FULL[15:8]							

15	14	13	12	11	10	9	8
Reserved						TX_STOP_MODE	RX_STOP_MODE

7	6	5	4	3	2	1	0
RX_FIFO_FULL[7:0]							

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:24	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_FIFO_FULL[15:8]	R/W	23:16	00H	The higher 8 bits of RX_FIFO_FULL
Reserved	R	15:10	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_STOP_MODE	R/W	9	0	Specify whether to stop SPI transmission/reception when the transmission FIFO buffer becomes empty during a DMA master transmission. 0: Do not stop transfer. 1: Stop transfer.
RX_STOP_MODE	R/W	8	0	Specify whether to stop SPI transmission/reception when the number of received words reaches the value specified for the RX_FIFO_FULL bits during a DMA master reception. 0: Do not stop transfer. 1: Stop transfer.
RX_FIFO_FULL[7:0]	R/W	7:0	00H	The lower 8 bits of RX_FIFO_FULL Specify the threshold of the number of received words that causes an RX_STOP interrupt. Specify this together with the higher 8 bits. 0000H: 1 word 0001H: 2 words ... FFFEH: 65,535 words FFFFH: 65,536 words

Caution **The threshold specified for the RX_FIFO_FULL bits is valid for reception (reception mode or transmission/reception mode) in DMA mode.**

An RX_STOP interrupt is output regardless of the RX_STOP_MODE setting. When transmission and reception are enabled (WRT and RD = 1 in the SPI_CONTROL register), set the TX_STOP_MODE and RX_STOP_MODE bits to 1. Setting fixed-length transfer for one side only is prohibited in this mode.

When the RX_STOP_MODE bit of the SPI_STATUS register is set to 1, reception stops when the reception FIFO buffer becomes full or the number of words specified for the RX_FIFO_FULL bits have been received. (After that, even if space becomes available in the reception FIFO buffer, reception is not resumed.)

3.2.15 SPI CS fixed value setting register

This register (SPI_TIECS: xxxx_1038H) selects normal operation or the fixed-value output mode for the pins SPI_CS0_0, SPI_CS1, SPI_CS2, SPI_CS3, SPI_CS4, SPI_CS5, and SPI_CS6, and, if fixed-value output mode is selected, this register specifies the fixed value to output.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved	E_CS6	E_CS5	E_CS4	E_CS3	E_CS2	E_CS1	E_CS0

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	R_CS6	R_CS5	R_CS4	R_CS3	R_CS2	R_CS1	R_CS0

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:23	–	Reserved. If these bits are read, 0 is returned for each bit.
E_CS6	R/W	22	0	0: Normal operation 1: Output the value specified for the R_CS6 bit.
E_CS5	R/W	21	0	0: Normal operation 1: Output the value specified for the R_CS5 bit.
E_CS4	R/W	20	0	0: Normal operation 1: Output the value specified for the R_CS4 bit.
E_CS3	R/W	19	0	0: Normal operation 1: Output the value specified for the R_CS3 bit.
E_CS2	R/W	18	0	0: Normal operation 1: Output the value specified for the R_CS2 bit.
E_CS1	R/W	17	0	0: Normal operation 1: Output the value specified for the R_CS1 bit.
E_CS0	R/W	16	0	0: Normal operation 1: Output the value specified for the R_CS0 bit.
Reserved	R	15:7	–	Reserved. If these bits are read, 0 is returned for each bit.
R_CS6	R/W	6	0	Specify the value output from the SPI_CS6 pin. 0: Fixed to 0 1: Fixed to 1
R_CS5	R/W	5	0	Specify the value output from the SPI_CS5 pin. 0: Fixed to 0 1: Fixed to 1

(2/2)

Name	R/W	Bit No.	After Reset	Description
R_CS4	R/W	4	0	Specify the value output from the SPI_CS4 pin. 0: Fixed to 0 1: Fixed to 1
R_CS3	R/W	3	0	Specify the value output from the SPI_CS3 pin. 0: Fixed to 0 1: Fixed to 1
R_CS2	R/W	2	0	Specify the value output from the SPI_CS2 pin. 0: Fixed to 0 1: Fixed to 1
R_CS1	R/W	1	0	Specify the value output from the SPI_CS1 pin. 0: Fixed to 0 1: Fixed to 1
R_CS0	R/W	0	0	Specify the value output from the SPI_CS0 pin. 0: Fixed to 0 1: Fixed to 1

3.2.16 PCM operating mode setting register

This register (PCM_FUNC_SEL: xxxx_2000H) specifies the operating mode, selects the serial interface timing, master mode/slave mode, and transmission start timing, and specifies whether to enable DMA transfer.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved					TX_TIM		

15	14	13	12	11	10	9	8
Reserved		AC97_CLK_DIR	AC97_SYNC	LR_AJUST	Reserved	CLK_INV	SEN_INV

7	6	5	4	3	2	1	0
DMA	Reserved		M_S		MODE_SEL		

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:19	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_TIM	R/W	18:16	000b	Select the amount of valid data in the transmission FIFO buffer used to trigger serial transmission. 000: 30 words (default) 001: 24 words 010: 20 words 011: 16 words 100: 12 words 101: 8 words 110: 4 words 111: 1 word
Reserved	R	15:14	–	Reserved. If these bits are read, 0 is returned for each bit.
AC97_CLK_DIR	R/W	13	0	Specify the clock direction in the AC97 mode. 0: Input 1: Output
AC97_SYNC	R/W	12	0	Specify the AC97 mode. This setting is valid only in a multi-channel mode (mode 5 or 6). 0: Enable single-cycle synchronization when starting the device 1: Enable synchronization during the 1st slot period of the first channel (AC97 mode)
LR_AJUST	R/W	11	0	Specify the justification of the transfer data 0: Left (default) 1: Right This setting is valid in modes 0 and 1.

(2/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	10	–	Reserved. If this bit is read, 0 is returned.
CLK_INV	R/W	9	0	Select whether to invert the serial clock polarity. 0: Do not invert (default). 1: Invert.
SEN_INV	R/W	8	0	Select whether to invert the serial data synchronization signal polarity. 0: Do not invert (default). 1: Invert
DMA	R/W	7	0	Specify the operating mode. 0: DMA mode 1: CPU mode DMARQ is asserted only in the DMA mode.
Reserved	R	6:5	–	Reserved. If these bits are read, 0 is returned for each bit.
M_S	R/W	4:3	00b	Select the master mode/slave mode. 00: Stop (default) 01: Master mode 10: Slave mode 11: Setting prohibited **Remarks 1.** When using the AC97 mode, specify the master mode for these bits.
MODE_SEL	R/W	2:0	000b	Select the serial interface operating mode. For the timing in each mode, see **4.3.1 Serial interface timing.** 000: Mode 0 (default) 001: Mode 1 010: Mode 2 (I^{2}S format) 011: Mode 3 (MSB justified) 100: Mode 4 (LSB justified) 101: Mode 5 (multi-channel mode, AC97) 110: Mode 6 (multi-channel mode) 111: Reserved

3.2.17 PCM data transfer enable set register

This register (PCM_TXRX_EN: xxxx_2004H) specifies the reception enable bit and transmission enable bit. Set up this register while transmission and reception are disabled.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						RX_EN	TX_EN

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:2	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_EN	R	1	0	Indicates whether reception is enabled. 0: Disabled (default) 1: Enabled.
	W			Specify whether to enable reception. 0: The current status is retained. 1: Enable
TX_EN	R	0	0	Indicates whether transmission is enabled. 0: Disabled (default) 1: Enabled.
	W			Specify whether to enable transmission. 0: The current status is retained. 1: Enable

3.2.18 PCM data transfer enable clear register

This register (PCM_TXRX_DIS: xxxx_2008H) clears the reception enable bit and the transmission enable bit.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						RX_ENCLR	TX_ENCLR

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:2	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_ENCLR	W	1	0	Clear the reception enable bit. 0: The current status is retained. 1: Clear the reception enable bit (RX_EN).
TX_ENCLR	W	0	0	Clear the transmission enable bit. 0: The current status is retained. 1: Clear the transmission enable bit (TX_EN).

3.2.19 PCM data transfer cycle setting register

This register (PCM_CYCLE: xxxx_200CH) specifies the frame length, the number of valid reception bits and the number of valid transmission bits, and whether to enable padding when receiving and transmitting data. Be sure to set up this registers before transmission and reception are enabled, and one frame period after transmission and reception are disabled (one word period in phase 1 or phase 2, whichever is specified as longer, in mode 5 or 6), or after a transmission/reception re-enable interrupt occurs. The settings of this register apply to phase 1 in modes 5 and 6.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
TX_PD	Reserved		SOB				

15	14	13	12	11	10	9	8
RX_PD	Reserved		SIB				

7	6	5	4	3	2	1	0
CYC_VAL							

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:24	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_PD	R/W	23	0	Specify whether to enable padding when transmitting data. 0: Disable (default) 1: Enable To enable padding, the number of valid transmission bits must be set to 8 (SOB = 07H) or 16 (SOB = 0FH). Setting this bit to 1 is prohibited in modes 5 and 6.
Reserved	R	22:21	–	Reserved. If these bits are read, 0 is returned for each bit.
SOB	R/W	20:16	00H	Specify the number of valid transmission bits (PCMx_SO signal) according to the number of PCMx_CLK clock cycles. The number of valid transmission bits is (specified value + 1). In modes 0 to 4, CYC_VAL ≥ SOB (the number of valid transmission bits) must be satisfied. Setting range: 07H to 1FH (8 to 32 bits)Note 1
RX_PD	R/W	15	0	Specify whether to enable padding when receiving data. 0: Disable (default) 1: Enable To enable padding, the number of valid reception bits must be set to 8 (SIB = 07H) or 16 (SIB = 0FH). The enable setting is prohibited in modes 5 and 6. Be sure to set this bit to 0.
Reserved	R	14:13	–	Reserved. If these bits are read, 0 is returned for each bit.

(2/2)

Name	R/W	Bit No.	After Reset	Description
SIB	R/W	12:8	00H	Specify the number of valid reception bits (PCMx_SI signal) according to the number of PCMx_CLK clock cycles. The number of valid reception bits is (specified value + 1). In modes 0 to 4, CYC_VAL ≥ SIB (the number of valid reception bits) must be satisfied. Setting range: 07H to 1FH (8 to 32 bits)Note 1
CYC_VAL	R/W	7:0	00H	In modes 0 to 4, the frame length is specified according to the number of PCMx_CLK clock cycles. In mode 5 , the frame length is specified by the number of words of phase 1. The frame length is (specified value + 1). In modes 0 to 4, CYC_VAL ≥ SOB (the number of valid transmission bits) and CYC_VAL ≥ SIB (the number of valid reception bits) must be satisfied. The specifiable number of clock cycles depends on the mode selected. Modes 0 to 4: 07H to 3FH (8 to 64 clock cycles) Mode 5 or 6: 00H to 80H (0 to 128 clock cycles)Note 2

Notes 1. In mode 5 or 6, set the SOB and SIB bits to the same values for a simultaneous transmission/reception operation. A value different from phase 2 (SOB2 and SIB2 bits of PCM_CYCLE2 register) can be specified. (The same values must be specified for SOB and SIB. Setting different values to SOB and SOB2 or SIB and SIB2 is allowed.)

2. In mode 5 or 6, setting the CYC_VAL bit of the PCMx_CYCLE register to 00H is prohibited if the CYC_VAL2 bit of the PCMx_CYCLE2 register is set to 00H. If the CYC_VAL bit is set to 00H, the value specified for the PCMx_CYCLE2 register becomes valid and the single-phase operation is performed. A value different from phase 2 (SOB2 and SIB2 bits of PCM_CYCLE2 register) can be specified. (Setting different values to CYC_VAL and CYC_VAL2 is allowed.)

3.2.20 PCM interrupt raw status register

This register (PCM_RAW: xxxx_2010H) indicates the status of the interrupt sources.

31	30	29	28	27	26	25	24
Reserved				RX_STRRAW	Reserved		

23	22	21	20	19	18	17	16
Reserved				TX_EMPRAW	TX_STRRAW	Reserved	

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RX_RENRAW	RX_ORERAW	RX_URERAW	RX_FRERAW	Reserved	TX_ORERAW	TX_URERAW	TX_FRERAW

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:28	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_STRRAW	R	27	0	Indicates the status of reception data transfer requests. 0: No reception data transfer request 1: A reception data transfer request has been issued.
Reserved	R	26:20	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_EMPRAW	R	19	0	Indicates that there is no more data to transmit. 0: There is data to transmit. 1: There is no more data to transmit.
TX_STRRAW	R	18	1	Indicates the status of data transmission requests. 0: No data transmission request 1: A data transmission request has been issued.
Reserved	R	17:8	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_RENRAW	R	7	0	Indicates the raw status of the reception FIFO buffer. 0: There is no valid reception data (disabling read operation) (default). 1: There is valid reception data (enabling read operation)[Note 1].
RX_ORERAW	R	6	0	Indicates the raw status of the reception overrun error. 0: Normal (default) 1: An overrun error is detected[Note 1].
RX_URERAW	R	5	0	Indicates the raw status of the reception underrun error. 0: Normal (default) 1: An underrun error is detected[Note 1].
RX_FRERAW	R	4	0	Indicates the raw status of the reception frame synchronization error[Note 6]. 0: Normal (default) 1: A frame synchronization error is detected (only in slave operation)[Note 2].
Reserved	R	3	–	Reserved. If this bit is read, 0 is returned.

(2/2)

Name	R/W	Bit No.	After Reset	Description
TX_ORERAW	R	2	0	Indicates the raw status of the transmission overrun error. 0: Normal (default) 1: An overrun error is detected[Note 3].
TX_URERAW	R	1	0	Indicates the raw status of the transmission underrun error. 0: Normal (default) 1: An underrun error is detected[Note 4].
TX_FRERAW	R	0	0	Indicates the raw status of a transmission frame synchronization error[Note 6]. 0: Normal (default) 1: A frame synchronization error is detected (only in slave operation)[Note 5].

Notes
1. This setting is reset by disabling reception (bit 1 of PCM_TXRX_DIS = 1).
2. This setting is reset by disabling reception (bit 1 of PCM_TXRX_DIS = 1) or clearing the interrupt source (bit 4 of PCM_CLEAR = 1).
3. This setting is reset by disabling transmission (bit 0 of PCM_TXRX_DIS = 1).
4. This setting is reset by disabling transmission (bit 0 of PCM_TXRX_DIS = 1) or clearing the interrupt source (bit 1 of PCM_CLEAR = 1).
5. This setting is reset by disabling transmission (bit 0 of PCM_TXRX_DIS = 1) or clearing the interrupt source (bit 0 of PCM_CLEAR = 1).
6. For details about synchronization errors, see **Note** in **3.2.21 PCM interrupt status registers**.

3.2.21 PCM interrupt status register

This register (PCM_STATUS: xxxx_2014H) indicates the status of the interrupt sources.
The result of disabling the interrupt raw status register with values specified for the interrupt enable set register (PCM_ENSET) is shown.

31	30	29	28	27	26	25	24
Reserved				RS_STR	Reserved		

23	22	21	20	19	18	17	16
Reserved				TX_EMPTY	TX_STR	Reserved	

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RX_REN	RX_ORE	RX_URE	RX_FRE	Reserved	TX_ORE	TX_URE	TX_FRE

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:28	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_STR	R	27	0	Indicates the status of reception data transfer requests. 0: No reception data transfer request 1: A reception data transfer request has been issued.
Reserved	R	26:20	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_EMPTY	R	19	0	Indicates the status of the transmission FIFO buffer status report. 0: It has been reported that there is no data to transmit. 1: It has been reported that there is data to transmit.
TX_STR	R	18	0	Indicates the status of the data transmission request report. 0: It has been reported that no data transmission request has been issued. 1: It has been reported that a data transmission request has been issued.
Reserved	R	17:8	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_REN	R	7	0	Indicates the status of the reception FIFO interrupt.
RX_ORE	R	6	0	Indicates the status of the reception overrun error interrupt.
RX_URE	R	5	0	Indicates the status of the reception underrun error interrupt.
RX_FRE	R	4	0	Indicates the status of the reception frame synchronization error interrupt.
Reserved	R	3	0	Reserved. If this bit is read, 0 is returned.
TX_ORE	R	2	0	Indicates the status of the transmission overrun error interrupt.
TX_URE	R	1	0	Indicates the status of the transmission underrun error interrupt.
TX_FRE	R	0	0	Indicates the status of the transmission frame synchronization error interrupt.

The sources for generating and clearing each interrupt are as follows:

No.	Interrupt Type	Description	
1	Reception data transfer request (RX_STR)	Generation source	When data has accumulated in the reception FIFO buffer.
		Subsequent operation	An RDMARQ request is issued for the data stored in the reception FIFO buffer.
		Clearing method	Disabling reception (RX_EN = 0), reception stopping due to a reception error, or the reception FIFO buffer becoming empty.
2	Transmission data empty (TX_EMP)	Generation source	When transmission is enabled (TX_EN = 1), but there is no valid data in the transmission FIFO buffer.
		Subsequent operation	–
		Clearing method	Disabling transmission or data accumulating in the transmission FIFO buffer.
3	Transmission data transfer request (TX_STR)	Generation source	When transmission is enabled (TX_EN = 1) and there is space for data in the transmission FIFO buffer (space for at least one item of data when executing single transfer and space for at least 8 items of data when executing burst transfer).
		Subsequent operation	TDMARQ is issued.
		Clearing method	Disabling transmission (TX_EN = 0), transmission stopping due to a transmission error, or the transmission FIFO buffer becoming full.
4	RX_FIFO status (RX_REN)	Generation source	When there is valid data in the reception FIFO buffer.
		Subsequent operation	–
		Clearing method	The reception FIFO buffer becoming empty or disabling reception.
5	Reception overrun error (RX_ORE)	Generation source	When data is received from the serial interface while the reception FIFO buffer is full.
		Subsequent operation	Reception is stopped.
		Clearing method	Disabling reception.
6	Reception underrun error (RX_URE)	Generation source	When an attempt is made to read the reception FIFO buffer while it is empty. A dummy read from the parallel interface side
		Subsequent operation	Reception is stopped.
		Clearing method	Disabling reception.
7	Reception frame synchronization error (RX_FRE)	Generation source	In modes 0 to 6, when a synchronization error occurs on the serial interface during reception (the frame synchronization signal is not being input at the prescribed timing).
		Subsequent operation	Reception is stopped.
		Clearing method	Writing to the corresponding bit of the PCM interrupt clear register (PCM_CLEAR) or disabling reception.
8	Transmission overrun error (TX_ORE)	Generation source	When the transmission FIFO buffer is written to while it is full.
		Subsequent operation	Transmission is stopped.
		Clearing method	Disabling transmission.

No.	Interrupt Type	Description	
9	Transmission underrun error (TX_URE)	Generation source	When an attempt is made to read the next data from the empty transmission FIFO buffer while transmission is enabled (TX_EN = 1). (This error is only detected in modes 0 to 6.)
		Subsequent operation	Transmission is stopped.
		Clearing method	Writing to the corresponding bit of the PCM interrupt clear register (PCM_CLEAR) or disabling transmission.
10	Transmission frame synchronization error (TX_FRE)	Generation source	In modes 0 to 6, when a synchronization error occurs on the serial interface during transmission (the frame synchronization signal is not being input at the prescribed timing).
		Subsequent operation	Transmission is stopped.
		Clearing method	Writing to the corresponding bit of the PCM interrupt clear register (PCM_CLEAR) or disabling transmission.

3.2.22 PCM interrupt enable set registers

These registers (PCM_ENSET: xxxx_2018H) specify whether to enable issuance of the interrupt request for each interrupt source.

31	30	29	28	27	26	25	24
Reserved				RX_STRENB	Reserved		

23	22	21	20	19	18	17	16
Reserved				TX_EMPENB	TX_STRENB	Reserved	

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RX_REN_EN	RX_ORE_EN	RX_URE_EN	RX_FRE_EN	Reserved	TX_ORE_EN	TX_URE_EN	TX_FRE_EN

(1/3)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:28	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_STRENB	R	27	0	Indicates whether issuance of the reception data transfer request interrupt request is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the reception data transfer request interrupt. 0: Ignored 1: Enable the interrupt.
Reserved	R	26:20	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_ENPENB	R	19	0	Indicates whether issuance of the transmission data empty interrupt request is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the transmission data empty interrupt. 0: Ignored 1: Enable the interrupt.
TX_STRENB	R	18	0	Indicates whether issuance of the data transmission request interrupt request is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the data transmission request interrupt. 0: Ignored 1: Enable the interrupt.
Reserved	R	17:8	0	Reserved. If these bits are read, 0 is returned for each bit.

(2/3)

Name	R/W	Bit No.	After Reset	Description
RX_REN_EN	R	7	0	Indicates whether issuance of the reception FIFO interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the reception FIFO interrupt. 0: Ignored 1: Enable the interrupt.
RX_ORE_EN	R	6	0	Indicates whether issuance of the reception overrun error interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the reception overrun error interrupt. 0: Ignored 1: Enable the interrupt.
RX_URE_EN	R	5	0	Indicates whether issuance of the reception underrun error interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the reception underrun error interrupt. 0: Ignored 1: Enable the interrupt.
RX_FRE_EN	R	4	0	Indicates whether issuance of the reception frame synchronization error interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the reception frame synchronization error interrupt. 0: Ignored 1: Enable the interrupt.
Reserved	R	3	0	Reserved. If these bits are read, 0 is returned for each bit.
TX_ORE_EN	R	2	0	Indicates whether issuance of the transmission overrun error interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the transmission overrun error interrupt. 0: Ignored 1: Enable the interrupt.
TX_URE_EN	R	1	0	Indicates whether issuance of the transmission underrun error interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the transmission underrun error interrupt. 0: Ignored 1: Enable the interrupt.

(3/3)

Name	R/W	Bit No.	After Reset	Description
TX_FRE_EN	R	0	0	Indicates whether issuance of the transmission frame synchronization error interrupt is enabled. 0: Disabled 1: Enabled
	W			Enable issuance of the transmission frame synchronization error interrupt. 0: Ignored 1: Enable the interrupt.

Cautions 1. When performing DMA transfer, setting bit 7 (RX_REN_EN) to 1 is not recommended. Interrupts corresponding to bits 7 are unnecessary because they can be checked by using the DMA request signal.

2. Interrupts are set up using the three registers shown below. To enable each interrupt, write 1 to the corresponding bit.

Example The execution result depends on the order of writing to the registers.

Set PCM_ENSET to 0101_0101H → Interrupts corresponding to bits 6, 4, 2 and 0 are enabled.

Set PCM_ENCLR to 0001_0001H → Interrupts corresponding to bits 4 and 0 are disabled.

Read PCM_ENSET = 0100_0100H → Interrupts corresponding to bits 6 and 2 remain enabled.

3.2.23 PCM interrupt enable clear register

This register (PCM_ENCLR: xxxx_201CH) clears the interrupt request issuance enable bit (interrupt enable bit) for each interrupt source.

31	30	29	28	27	26	25	24
Reserved				RX_STRDIS	Reserved		

23	22	21	20	19	18	17	16
Reserved				TX_ENPDIS	TX_STRDIS	Reserved	

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RX_REN MASK	RX_ORE MASK	RX_URE MASK	RX_FREMASK	Reserved	TX_OREMASK	TX_UREMASK	TX_FREMASK

(1/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:28	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_STRDIS	W	27	0	Clear the reception data transfer request interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
Reserved	R	26:20	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_ENPDIS	W	19	0	Clear the transmission data empty interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
TX_STRDIS	W	18	0	Clear the data transmission request interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
Reserved	R	17:8	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_RENMASK	W	7	–	Clear the reception FIFO interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
RX_OREMASK	W	6	–	Clear the reception overrun error interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
RX_UREMASK	W	5	–	Clear the reception underrun error interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
RX_FREMASK	W	4	–	Clear the reception frame synchronization error interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.

(2/2)

Name	R/W	Bit No.	After Reset	Description
Reserved	R	3	–	Reserved. If this bit is read, 0 is returned.
TX_OREMASK	W	2	–	Clear the transmission overrun error interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
TX_UREMASK	W	1	–	Clear the transmission underrun error interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.
TX_FREMASK	W	0	–	Clear the transmission frame synchronization error interrupt enable bit. 0: Retain the setting of the interrupt enable bit. 1: Clear the corresponding interrupt enable bit.

3.2.24 PCM interrupt clear register

This register (PCM_CLEAR: xxxx_2020H) clears the interrupt sources.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">Reserved</td></tr>
</table>

<table>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">Reserved</td></tr>
</table>

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">Reserved</td><td>RX _FRECLR</td><td colspan="2">Reserved</td><td>TX_URECLR</td><td>TX_FRECLR</td></tr>
</table>

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:5	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_FRECLR	W	4	–	Clear the RX_FRE source (reception synchronization error) bit. 0: Retain the interrupt source. 1: Clear the corresponding interrupt source.
Reserved	R	3:2	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_URECLR	W	1	–	Clear the TX_URE source (transmission underrun error) bit. 0: Retain the interrupt source. 1: Clear the corresponding interrupt source.
TX_FRECLR	W	0	–	Clear the TX_FRE source (transmission synchronization error) bit. 0: Retain the interrupt source. 1: Clear the corresponding interrupt source.

3.2.25 PCM transmission data register

This register (PCM_TXQ: xxxx_2024H) stores the data to be written in the transmission FIFO buffer. Use this register to read the last data written to the transmission FIFO buffer.
Be sure to write to this register in 32-bit units. Operation is not guaranteed if this register is accessed in 8-bit or 16-bit units.

31	30	29	28	27	26	25	24
TXQ							

23	22	21	20	19	18	17	16
TXQ							

15	14	13	12	11	10	9	8
TXQ							

7	6	5	4	3	2	1	0
TXQ							

Name	R/W	Bit No.	After Reset	Description
TXQ	R/W	31:0	0000_0000H	Control transmission FIFO buffer write data.

The data written to PCM_TXQ is written to the entry pointed to by the write pointer in the 32-bit × 32-word transmission FIFO buffer. The write pointer is automatically controlled by hardware. If PCM_TXQ is read, the last written data is read.

An example of writing data to the transmission FIFO buffer is provided below.
Data longer than the transmission data bit length is ignored (×: Don't Care).

Example Transmission data bit length = 10 bits (SOB = 09H in PCM_CYCLE register)
xxxx_x123H → Write data (PCM_TXQ)
xxxx_x245H → Write data (PCM_TXQ)
xxxx_x067H → Write data (PCM_TXQ)
...

31 TX-FIFO[0] 24

×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	0	1
0	0	1	0	0	0	1	1

7 0

31 TX-FIFO[1] 24

×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	1	0
0	1	0	0	0	1	0	1

7 0

31 TX-FIFO[2] 24

×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	0	0
0	1	1	0	0	1	1	1

7 0

Caution The figure above shows an example of writing with no data padding. If writing is performed with data padding (even if the transmission data bit length is 8 or 16 bits), 32-bit data can be padded (no Don't Care bits are used). For details, see 4.3.2(5) Data padding.

3.2.26 PCM reception data register

This register (PCM_RXQ: xxxx_2028H) is used to read reception data from the reception FIFO buffer. If a reception is stopped, the value of this register is set to 0000_0000H.
Be sure to read this register in 32-bit units. Operation is not guaranteed if this register is accessed in 8-bit or 16-bit units.

31	30	29	28	27	26	25	24
RXQ							

23	22	21	20	19	18	17	16
RXQ							

15	14	13	12	11	10	9	8
RXQ							

7	6	5	4	3	2	1	0
RXQ							

Name	R/W	Bit No.	After Reset	Description
RXQ	R	31:0	0000_0000H	Indicates reception FIFO buffer read data.

Data is read from the entry pointed to by the read pointer in the 32-bit × 32-word reception FIFO buffer. The read pointer is automatically controlled by hardware. No data can be written to PCM_RXQ.

An example of reading data from the reception FIFO buffer is provided below.

Example Reception data bit length = 28 bits (SIB = 1BH in PCM_CYCLE register)
Read data (PCM_RXQ) ← 0123_4567H (RXFIFO[0])
Read data (PCM_RXQ) ← 089A_BCDEH (RXFIFO[1])
Read data (PCM_RXQ) ← 0F24_68ACH (RXFIFO[2])

RX-FIFO[0]

31							24
0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
7							0

RX-FIFO[1]

31							24
0	0	0	0	1	0	0	0
1	0	0	1	1	0	1	0
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	0
7							0

RX-FIFO[2]

31							24
0	0	0	0	1	1	1	1
0	0	1	0	0	1	0	0
0	1	1	0	1	0	0	0
1	0	1	0	1	1	0	0
7							0

Caution The figure above shows an example of reading with no data padding. When reading is performed with data padding (even if the reception data bit length is 8 or 16 bits), 32-bit data can be padded. For details, see 4.3.2(5) Data padding.

3.2.27 PCM FIFO counter register (register used exclusively for debugging)

This register (PCM_FIFO_P: xxxx_202CH) stores the counter (pointer) value of the transmission FIFO buffer and reception FIFO buffer. This register is used exclusively for debugging. This register counts in synchronization with the serial clock; it is not synchronized with the system bus clock. When using this register for debugging, be sure to read the value of the register twice and only use the data when the data read both times is the same.

31	30	29	28	27	26	25	24
Reserved			RX_R_COUNT				

23	22	21	20	19	18	17	16
Reserved			RX_W_COUNT				

15	14	13	12	11	10	9	8
Reserved			TX_R_COUNT				

7	6	5	4	3	2	1	0
Reserved			TX_W_COUNT				

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:29	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_R_COUNT	R	28:24	00H	5-bit counter for reading the reception FIFO buffer This counter is incremented each time PCM_RXQ is read.
Reserved	R	23:21	–	Reserved. If these bits are read, 0 is returned for each bit.
RX_W_COUNT	R	20:16	00H	5-bit counter for writing to the reception FIFO buffer This counter is incremented each time serial data is received.
Reserved	R	15:13	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_R_COUNT	R	12:8	00H	5-bit counter for reading the transmission FIFO buffer This counter is incremented each time serial data is transmitted.
Reserved	R	7:5	–	Reserved. If these bits are read, 0 is returned for each bit.
TX_W_COUNT	R	4:0	00H	5-bit counter for writing to the transmission FIFO buffer This counter is incremented each time PCM_TXQ is written.

3.2.28 PCM data transfer cycle setting register 2

This register (PCM_CYCLE2: xxxx_2030H) specifies the settings for phase 2 in modes 5 and 6. These settings are ignored in other modes.

Set up this register while transmission and reception are disabled.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved			SOB2				

15	14	13	12	11	10	9	8
Reserved			SIB2				

7	6	5	4	3	2	1	0
CYC_VAL2							

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:21	–	Reserved. If these bits are read, 0 is returned for each bit.
SOB2	R/W	20:16	00H	Specify the number of valid transmission bits (PCM_SO signal) according to the number of PCM_CLK clock cycles. The number of valid transmission bits is (specified value + 1). Setting range: 07H to 1FH (8 to 32 bits)Note 1
Reserved	R	15:13	–	Reserved. If these bits are read, 0 is returned for each bit.
SIB2	R/W	12:8	00H	Specify the number of valid reception bits (PCM_SI signal) according to the number of PCM_CLK clock cycles. The number of valid reception bits is (specified value + 1). Setting range: 07H to 1FH (8 to 32 bits)Note 1
CYC_VAL2	R/W	7:0	00H	Specify the number of words of phase 2 per frame. Modes 5 and 6: 00H to 80H (0 to 128 words)Note 2

Notes 1. In mode 5 or 6, set the SOB2 and SIB2 bits to the same values for a simultaneous transmission/reception. A value different from phase 1 (SOB and SIB bits of PCM_CYCLE register) can be specified. (The same values must be specified for SOB2 and SIB2. Setting different values to SOB and SOB2 or SIB and SIB2 is allowed.)

2. Setting the CYC_VAL2 bit of the PCM_CYCLE2 register to 00H is prohibited in modes 5 and 6 if the CYC_VAL bit of the PCM_CYCLE register is set to 00H. If CYC_VAL2 is set to 00H, the value specified for PCM_CYCLE becomes valid and the single-phase operation is performed. A value different from the setting for phase 1 (CYC_VAL bit of the PCM_CYCLE register) can be specified. (Setting different values to CYC_VAL and CYC_VAL2 is allowed.)

3.2.29 PCM AC97 reset assert register

The settings of this register (PCM_AC97_RESET: xxxx_2038H) are enabled when SIO is in audio mode. Set the bits of this register to 1 to assert a reset signal in an AC97-compliant codec.

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						AC97_ CS0RST	AC97_ CS1RST

Name	R/W	Bit No.	After Reset	Description
Reserved	R	31:2	–	Reserved. If these bits are read, 0 is returned for each bit.
AC97_CS0RST	R/W	1	0	If this bit is set to 1 in audio mode, SIO_CS0 is set to 1. 0: Reset signal is not asserted. SIO_CS0 is set to 0. 1: Reset signal is asserted. SIO_CS0 is set to 1.
AC97_CS1RST	R/W	0	0	If this bit is set to 1 in audio mode, SIO_CS1 is set to 1. 0: Reset signal is not asserted. SIO_CS1 is set to 0. 1: Reset signal is asserted. SIO_CS1 is set to 1.

Caution The AC97 specification limits the period during which a reset signal can be asserted in an AC97-compliant codec. However, SIO does not contain logic to control how long the reset signal is asserted. This time must therefore be controlled by the CPU by using an external timer.

4. Description of Functions

4.1 Function Details

4.1.1 4-wire serial interface

Signal Name	I/O	Description
SIO_CLK	I/O	Serial clock (input on the slave side and output on the master side)
SIO_SI	Input	Serial input data
SIO_SO	Output	Serial output data
SIO_CSn	I/O	Device enable (CS0: Input on the slave side and output on the master side. Other CS: Always output)

4.1.2 Transfer mode

The following combinations of transfer modes are supported.
The transfer modes and how transfers start and stop in each mode are shown in **4.1.3 SPI transfer modes (operating or stopped)**.
The interrupts corresponding to each transfer mode are shown in **4.1.4 SPI transfer modes and the corresponding interrupts**.

(1) Transmission and reception mode

(a) Transmission mode (WRT = 1 and RD = 0 in the SPI_CONTROL register):
APB bus input → Serial data output
(b) Reception mode (WRT = 0 and RD = 1 in the SPI_CONTROL register):
Serial data input → APB bus output
(c) Transmission/reception mode (WRT = 1 and RD = 1 in the SPI_CONTROL register):
In this mode, transmission and reception are performed at the same time.

(2) Master mode and slave mode

(a) Master mode (M_S of the SPI_MODE register = 0):
SIO runs as a master and outputs data to the SIO_CSn, SIO_CLKO, and SIO_SO pins.
Transmission and reception are performed using SCLK, which is supplied from the SMU.
(b) Slave mode (M_S of the SPI_MODE = 1):
SIO runs as a slave and inputs data from the SIO_CLKI, SIO_CS0_I, and SIO_SI pins.
Transmission and reception are performed using SCLK, which is supplied from the external pin SIO_CLKI.

(3) CPU mode and DMA control mode

(a) CPU control mode (DMA of the SPI_MODE = 0):

In this mode, the transaction ends when transferring one frame ends in the transmission, reception, or transmission and reception mode.

Transmission: Transmission starts when the CPU writes data to the SPI transmission data register (SPI_TX_DATA) by way of the APB bus and the SPI is started. After transmission finishes, an END interrupt is output.

Reception: After the SPI module is started, reception starts when the CPU reads data from the SPI reception data register (SPI_RX_DATA) by way of the APB bus, and an RDV and END interrupt are output.

(b) DMA control mode (DMA of the SPI_MODE register = 1):

Two 32-word FIFO buffers are provided for transmission and reception.

Transmission: Data is transmitted by way of the DMA bus (AHB) through handshaking by using SIOx_DMA_TDMARQ.

Transmission starts when 1 or more words of data are accumulated in the transmission FIFO buffer.

Transmission stops when the CPU stops it, or the transmission FIFO buffer becomes empty.

When transmission stops because the transmission FIFO buffer becomes empty, a TX_STOP interrupt is output.

Reception: Data is received by way of the DMA bus (AHB) through handshaking by using SIO_DMA_RDMARQ.

Reception stops when the CPU stops it, or the number of received data items matches the value specified for the SIOx_DMA_LENG register.

Transmission is suspended when the reception FIFO buffer becomes full.

When reception stops because the number of received data items matches the SIOx_DMA_LENG value, an RX_STOP interrupt is output.

When 0 is input to SIOx_DMA_LENG, transfers continue until the CPU stops them.

Transmission/reception:

Operation stops if either the above transmission or reception is stopped. At this time, the TX_STOP or RX_STOP interrupt is output, according to the stopped transaction.

If transmission and reception stop at the same time, both interrupts are output.

4.1.3 SPI transfer modes (operating or stopped)

(1/2)

WRT	RD	Operating Mode	Operation Started	Operation Stopped	Remark
1	0	CPU_TX master mode	Transmission is started when 1 is written to START and there is data in the FIFO buffer.	When serial transmission finishes, START is cleared to 0, and END is set to 1.	
0	1	CPU_RX master mode	Reception is started when 1 is written to START.	When serial reception finishes, START is cleared to 0, RDV is set to 1, and END is set to 1.	There is received data in the FIFO buffer when processing finishes.
1	1	CPU_TX/RX master mode	Transmission and reception are started when 1 is written to START and there is data in the FIFO buffer.	When serial transmission finishes, START is cleared to 0, RDV is set to 1, and END is set to 1.	There is received data in the FIFO buffer when processing finishes.
1	0	CPU_TX slave mode	Transmission is started when 1 is written to WRT and there is data in the FIFO buffer.	When serial transmission finishes, END is set to 1.	Transmission starts regardless of whether START is written to.
0	1	CPU_RX slave mode	Reception is started when 1 is written to RD.	When serial reception finishes, RDV is set to 1, and END is set to 1.	Reception starts regardless of whether START is written to. There is received data in the FIFO buffer when processing finishes.
1	1	CPU_TX/RX slave mode	Transmission and reception are started when 1 is written to START and there is data in the FIFO buffer.	When serial transmission finishes, RDV is set to 1, and END is set to 1.	Transmission and reception start regardless of whether START is written to. There is received data in the FIFO buffer when processing finishes.
1	0	DMA_TX master mode	1 is written to DMA. A TXDMA request is output. 1 is written to START. Transmission is started when at least 1 word of data accumulates in the FIFO buffer.	Transmission stops when the transmission FIFO buffer is empty, and then START is cleared to 0, and TX_STOP is set to 1. Alternatively, transmission stops when 1 is written to STOP, and then START is cleared to 0.	
0	1	DMA_RX master mode	Reception is started when 1 is written to START. An RXDMA request is output when data is received in the RXFIFO buffer.	Reception stops when SIOx_DMA_LENG does not equal 0 and the number of data items does not match SIOx_DMA_LENG, and then START is cleared to 0, and RX_STOP is set to 1. Alternatively, reception stops when 1 is written to STOP, and then START is cleared to 0.	

(2/2)

WRT	RD	Operating Mode	Operation Started	Operation Stopped	Remark
1	1	DMA_TX/RX master mode	1 is written to DMA. A TXDMA request is output. 1 is written to START. An RXDMA request is output when data is received in the RXFIFO buffer. Transmission is started when at least 1 word of data accumulates in the FIFO buffer.	If the TXFIFO buffer is empty or the number of received data items matches SIOx_DMA_LENG, START is cleared to 0. If the transmission FIFO buffer is empty, TX_STOP is set to 1. If SIOx_DMA_LENG does not equal 0 and the number of received data items does not match SIOx_DMA_LENG, RX_STOP is set to 1. Alternatively, reception stops when 1 is written to STOP, and then START is set to 1.	
1	0	DMA_TX slave mode	A TXDMA request is output when 1 is written to DMA. 1 is written to WRT. Transmission is started when there is CS or CLK input from the master.	If 1 is written to STOP, processing immediately stops.	
0	1	DMA_RX slave mode	1 is written to RD. The system waits to receive CS or CLK. An RXDMA request is output when data is received in the RXFIFO buffer.	If 1 is written to STOP, processing immediately stops.	
1	1	DMA_TX/RX slave mode	1 is written to DMA. A TXDMA request is output. 1 is written to WRT and RD. Transmission is started when there is CS or CLK input from the master. An RXDMA request is output when data is received in the RXFIFO buffer.	If 1 is written to STOP, processing immediately stops.	

Caution TXDMAREQ is output by setting DMA to 1.

If there is data in the reception FIFO buffer while DMA is set to 1, RXDMAREQ is output.

In CPU/DMA slave mode, data is transmitted and received regardless of whether the START bit is written or not.

4.1.4 SPI transfer modes and the corresponding interrupts

(1/2)

DMA	M_S	WRT	RD	Operating Mode	TX_ STOP	RX_ STOP	TERR	RDV	END	TX_ UDR	RX_ OVR	Detection or Output Timing	Remark
0	0	1	0	CPU_TX master mode					√			After serial transmission finishes	- TX_UDR is not generated because no transfer starts when the transmission FIFO buffer is empty.
0	0	0	1	CPU_RX master mode				√	√			After serial reception finishes	- RX_OVR is not generated because no transfer starts when the reception FIFO buffer is full.
0	0	1	1	CPU_TX/ RX master mode				√	√			After serial transmission and reception finish	- TX_UDR is not generated because no transfer starts when the transmission FIFO buffer is empty. - RX_OVR is not generated because no transfer starts when the reception FIFO buffer is full.
0	1	1	0	CPU_TX slave mode			√		√	√		TERR, END: After serial transmission finishes TX_UDR: When serial transmission starts[Note]	- When TX_UDR is generated, the data is not guaranteed because the internal FIFO buffer pointer is updated. A software reset is necessary to resume processing.
0	1	0	1	CPU_RX slave mode			√	√	√		√	TERR, RDV, END: After serial reception finishes RX_OVR: When serial reception starts	- If the reception FIFO buffer becomes full once, even if the buffer is read to cancel this status, the RX_OVR interrupt is generated the next time data is received. A software reset is necessary to resume processing.
0	1	1	1	CPU_TX RX slave mode			√	√	√	√	√	TERR, END: After serial transmission and reception finish RDV: After serial reception finishes TX_UDR: When serial transmission starts[Note] RX_OVR: When serial reception starts	- When TX_UDR is generated, the data is not guaranteed because the internal FIFO buffer pointer is updated. A software reset is necessary to resume processing. - If the reception FIFO buffer becomes full once, even if the buffer is read to cancel this status, the RX_OVR interrupt is generated the next time data is received. A software reset is necessary to resume processing.

Note If data is written to the transmission FIFO buffer after the CS signal has been input during slave transmission, TX_UDR is generated. However, TX_UDR might not be generated if the timing of inputting CS conflicts with writing data to the transmission FIFO buffer, causing the first bit of the data to be output incorrectly. This problem does not occur if the data is written to the transmission FIFO buffer before CS is input.

(2/2)

DMA	M_S	WRT	RD	Operating Mode	TX_ STOP	RX_ STOP	TERR	RDV	END	TX_ UDR	RX_ OVR	Detection or Output Timing	Remark
1	0	1	0	DMA_TX master mode	√							After serial transmission finishes	- TX_UDR is not generated because no transfer starts when the transmission FIFO buffer is empty.
1	0	0	1	DMA_RX master mode		√						After serial transmission finishes	- RX_OVR is not generated because no transfer starts when the reception FIFO buffer is full.
1	0	1	1	DMA_TX/ RX master mode	√	√						TX_STOP: After serial transmission finishes RX_STOP: After serial reception finishes	- TX_UDR is not generated because no transfer starts when the transmission FIFO buffer is empty. - RX_OVR is not generated because no transfer starts when the reception FIFO buffer is full.
1	1	1	0	DMA_TX slave mode	√		√			√		TX_STOP, TERR: After serial transmission finishes TX_UDR: When serial transmission starts[Note]	- When TX_UDR is generated, the data is not guaranteed because the internal FIFO buffer pointer is updated. A software reset is necessary to resume processing.
1	1	0	1	DMA_RX slave mode		√	√				√	RX_STOP, TERR: After serial reception finishes RX_OVR: When serial reception starts	- If the reception FIFO buffer becomes full once, even if the buffer is read to cancel this status, the RX_OVR interrupt is generated the next time data is received. A software reset is necessary to resume processing.
1	1	1	1	DMA_TX/ RX slave mode	√	√	√			√	√	TX_STOP: After serial transmission finishes RX_STOP: After serial reception finishes TERR: After serial transmission and reception finish TX_UDR: When serial transmission starts[Note] RX_OVR: When serial reception starts	- When TX_UDR is generated, the data is not guaranteed because the internal FIFO buffer pointer is updated. A software reset is necessary to resume processing. - If the reception FIFO buffer becomes full once, even if the buffer is read to cancel this status, the RX_OVR interrupt is generated the next time data is received. A software reset is necessary to resume processing.
					DMA TX only	DMA RX only	Slave only	CPU RX only	CPU only	TX slave only	RX slave only		

Note If data is written to the transmission FIFO buffer after the CS signal has been input during slave transmission, TX_UDR is generated.
However, TX_UDR might not be generated if the timing of inputting CS conflicts with writing data to the transmission FIFO buffer, causing the first bit of the data to be output incorrectly. This problem does not occur if the data is written to the transmission FIFO buffer before CS is input.

4.1.5 Interrupt generation

Interrupt signals (SIO_INT) can be generated by the following sources:

○ Interrupt signals (SIO_INT) are generated:

- When the transmission FIFO buffer becomes empty during master transfer in DMA mode. (TX_STOP)
- When the amount of received data reaches the value specified for the input SIOx_DMA_LENG bits of the SPI_CONTROL2 register during master transfer in DMA mode. (RX_STOP)
- When the number of SPI_SCLK_I clock cycles is not a multiple of the value specified for the NB_A bits of the SPI_MODE register during slave transfer. (TERR)

  When a TERR interrupt occurs, an END interrupt does not occur upon the completion of transfer. A software reset is required to resume transfer.
- When reception of one frame is completed in the CPU mode. (RDV)
- When transmission or reception of one frame is completed in the CPU mode. (END)
- When the transmission FIFO buffer underruns during slave transfer. (TX_UDR)

  At that time, the data being transmitted is not guaranteed. A software reset is required to resume transfer.
- When the reception FIFO buffer overruns during slave transfer. (RX_OVR)

  At that time, the data being received is not guaranteed. A software reset is required to resume transfer.

  Once the reception FIFO buffer becomes full, an RX_OVR interrupt occurs when the next data is received, even if data in the reception FIFO buffer is read to release it from the full state.
- When SCLK or CS is input before transmission data is set to the transmission FIFO buffer for slave transfer in DMA mode when transmission and reception is specified (WRT and RD = 1). (TX_UDR)

4.1.6 Clocks

Each SPI clock can be controlled (constant supply, automatic control and stop) by setting a clock setting register in the SMU. PCLK is requested by the SPI by pulling PCLKREQ, HCLKREQ, and SCLKREQ high.

4.1.7 Data buffers

Two data buffers of 32 bits × 32 words are provided for transmission and reception, respectively.
The data buffers consist of FIFO blocks. The transmission FIFO buffer is referenced from the system bus, as a port register on the write side, and the reception FIFO buffer is referenced as a port register on the read side.

4.1.8 Fixing CS by setting a register

The SPI_CS0 to SPI_CS5 signals can be fixed to a certain level (high or low) by setting the SPI_TIECS register.
The level to which the signals are to be fixed is specified using the SPI_POL register.
The CS level can only be fixed by the SPI_TIECS register; it is not affected by the SPI internal operation.

4.1.9 Switching the I/O phases of SPI_SI and SPI_SO (at the rising/falling edge of SCLK)

Using the CK_PHASE bit of the SPI_MODE register, the reverse phase of SPI_SI/SPI_SO input/output (default) can be changed to a common phase. The F/F phase of the reception side is changed during master operation, and the F/F phase of the transmission side is changed during slave operation.

4.1.10 Reset control

○ Software reset control

The SPI module is in the software reset state in the period from when the RST bit of the SPI_CONTROL register (SPI_CTRL) is set to 1 until it is set to 0.

The following registers are reset by a software reset:

- SIO_MODE_SWITCH
- SPI_CONTROL (except for the RST bit)
- SPI_STATUS
- SPI_RAW_STATUS
- TX_FIFO_P
- RX_FIFO_P

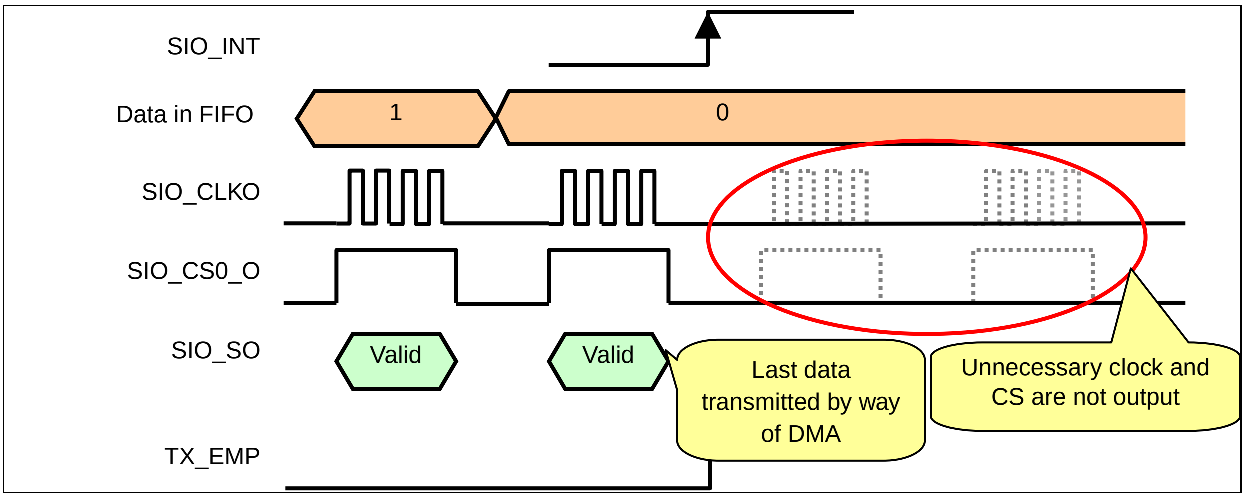
Caution Transfer is stopped if a software reset is executed during transfer. (The data being transferred is not guaranteed.)

4.1.11 Stopping output of CLK and CS0 when FIFO buffer is empty

- Outputting null values stops when the FIFO buffer becomes empty during DMA transfer.

When the master is transmitting data using DMA transfer, output of the clock and CS signal stops if the transmission FIFO buffer becomes empty.

Figure 4-1. Stopping Unnecessary Clock When FIFO Buffer Becomes Empty During Normal Transmission


SIO_INT
Data in FIFO
1
0
SIO_CLKO
SIO_CS0_O
SIO_SO
Valid
Valid
Last data transmitted by way of DMA
Unnecessary clock and CS are not output
TX_EMP

4.2 Detailed Timing

4.2.1 SPI interface timing 1 (CKn_DLY = 0 in master mode)

Figure 4-2 shows SPI interface timing 1.

Figure 4-2. SPI Interface Timing 1

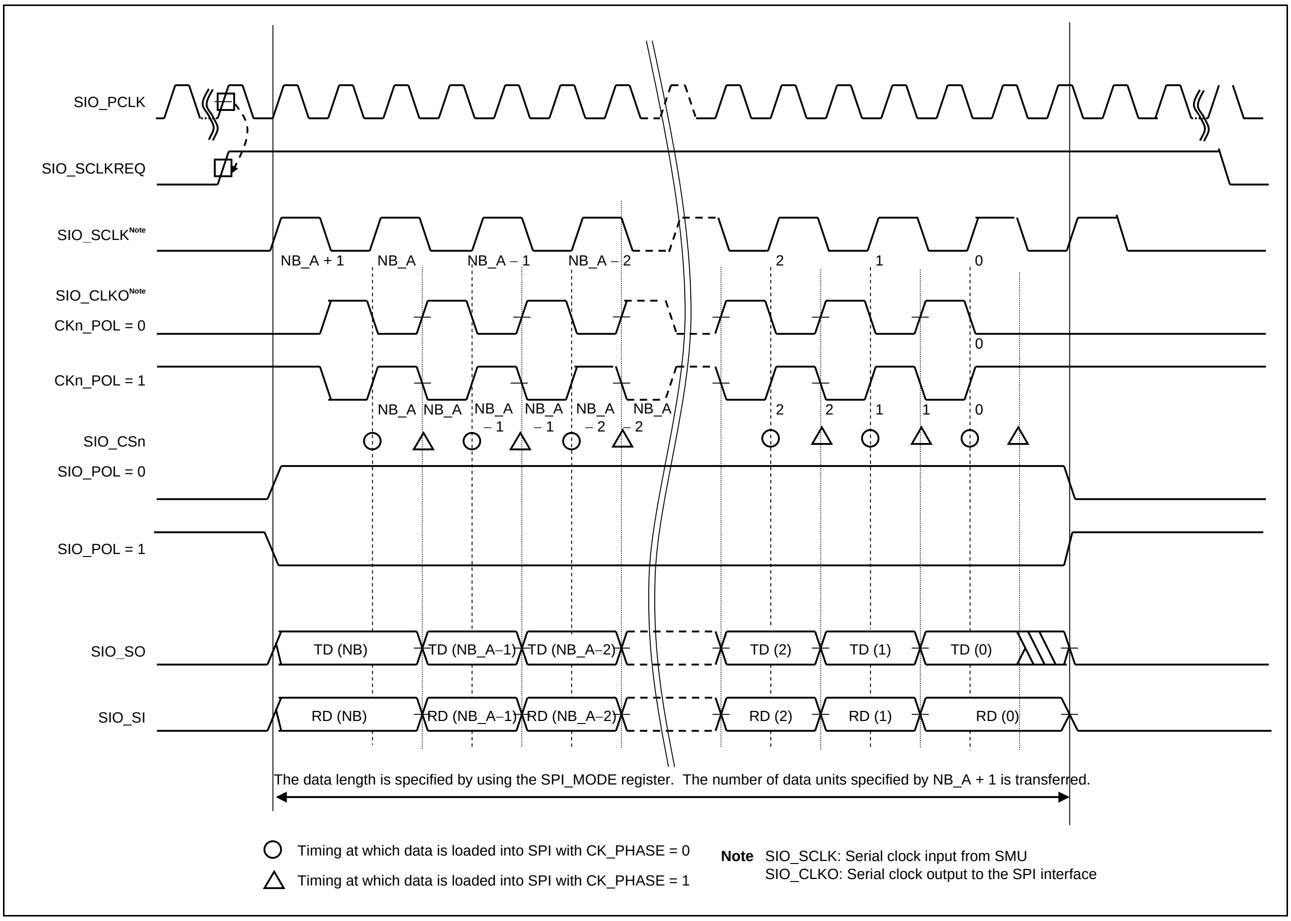

SIO_PCLK
SIO_SCLKREQ
SIO_SCLKNote
NB_A + 1
NB_A
NB_A – 1
NB_A – 2
2
1
0
SIO_CLKONote
CKn_POL = 0
CKn_POL = 1
NB_A
NB_A
NB_A – 1
NB_A – 1
NB_A – 2
NB_A – 2
SIO_CSn
SIO_POL = 0
SIO_POL = 1
SIO_SO
TD (NB)
TD (NB_A–1)
TD (NB_A–2)
TD (2)
TD (1)
TD (0)
SIO_SI
RD (NB)
RD (NB_A–1)
RD (NB_A–2)
RD (2)
RD (1)
RD (0)
The data length is specified by using the SPI_MODE register. The number of data units specified by NB_A + 1 is transferred.
○ Timing at which data is loaded into SPI with CK_PHASE = 0
△ Timing at which data is loaded into SPI with CK_PHASE = 1
Note SIO_SCLK: Serial clock input from SMU
SIO_CLKO: Serial clock output to the SPI interface

Figure 4-2 shows the timing of the SPI in master mode with the CKn_DLY bit[Note] of the SPI_POL register cleared to 0. A serial clock is output from SIO_CLKO, and a CS signal is output from SIO_CSn[Note] to an external device.

In master mode, SIO_SCLKREQ is set to 1 in synchronization with SIO_PCLK in cycle in which communication started, and this SIO_SCLKREQ triggers supply of SIO_SCLK from the SMU. Transmission/reception data is output in synchronization with SIO_SCLK when SIO_CSn is active[Note], and the received data is fetched by the SIO module in synchronization with SIO_SCLK.

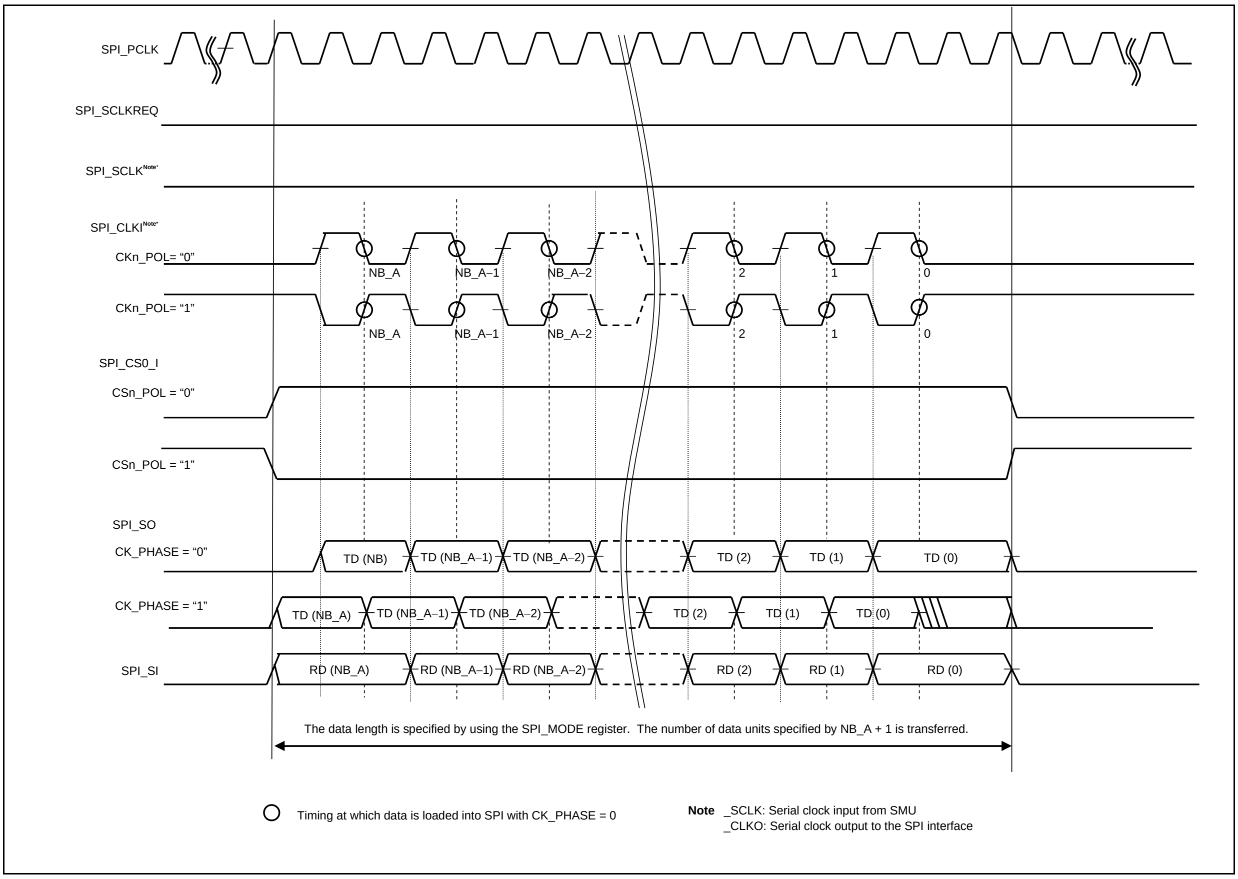
In master mode, when the reception data is fetched depends on the CK_PHASE setting.

Note CSn_POL, CKn_POL, and CKn_DLY show the polarity or delay setting of CSn specified by the CS_SELA bits of the SPI mode register (n = 0 to 5). Data is output to the selected CS during master operation, regardless of whether the mode is CPU mode or DMA mode. The CSn_POL bit is used to determine the active level of SIO_CSn (0: high, 1: low).

4.2.2 SPI interface timing 2 (CK0_DLY = 0 in slave mode)

Figure 4-3 shows SPI interface timing 2.

Figure 4-3. SPI Interface Timing 2



The data length is specified by using the SPI_MODE register. The number of data units specified by NB_A + 1 is transferred.

○ Timing at which data is loaded into SPI with CK_PHASE = 0

Note _SCLK: Serial clock input from SMU
_CLKO: Serial clock output to the SPI interface

Figure 4-3 shows the timing of the SPI in slave mode with the CK0_DLY bit[Note] of the SPI_POL register cleared to 0.

A serial clock is input from SIO_CLKI, and a CS signal is input to SIO_CSn_I from an external device.

In slave mode, SIO_SCLKREQ is not set to 1, and therefore a request for supply of SIO_SCLK is not issued.

Transmission/reception data is output in synchronization with SCLK (SIO_CLKI) when SIO_CS0_I is active[Note], and the received data is fetched by the SPI module in synchronization with SCLK (SPI_CLKI).

In slave mode, when the transmission data is output depends on the CK_PHASE setting.

4.2.3 SPI interface timing 3 (CKn_DLY = 1 in master mode)

Figure 4-4 shows SPI interface timing 3.

Figure 4-4. SPI Interface Timing 3

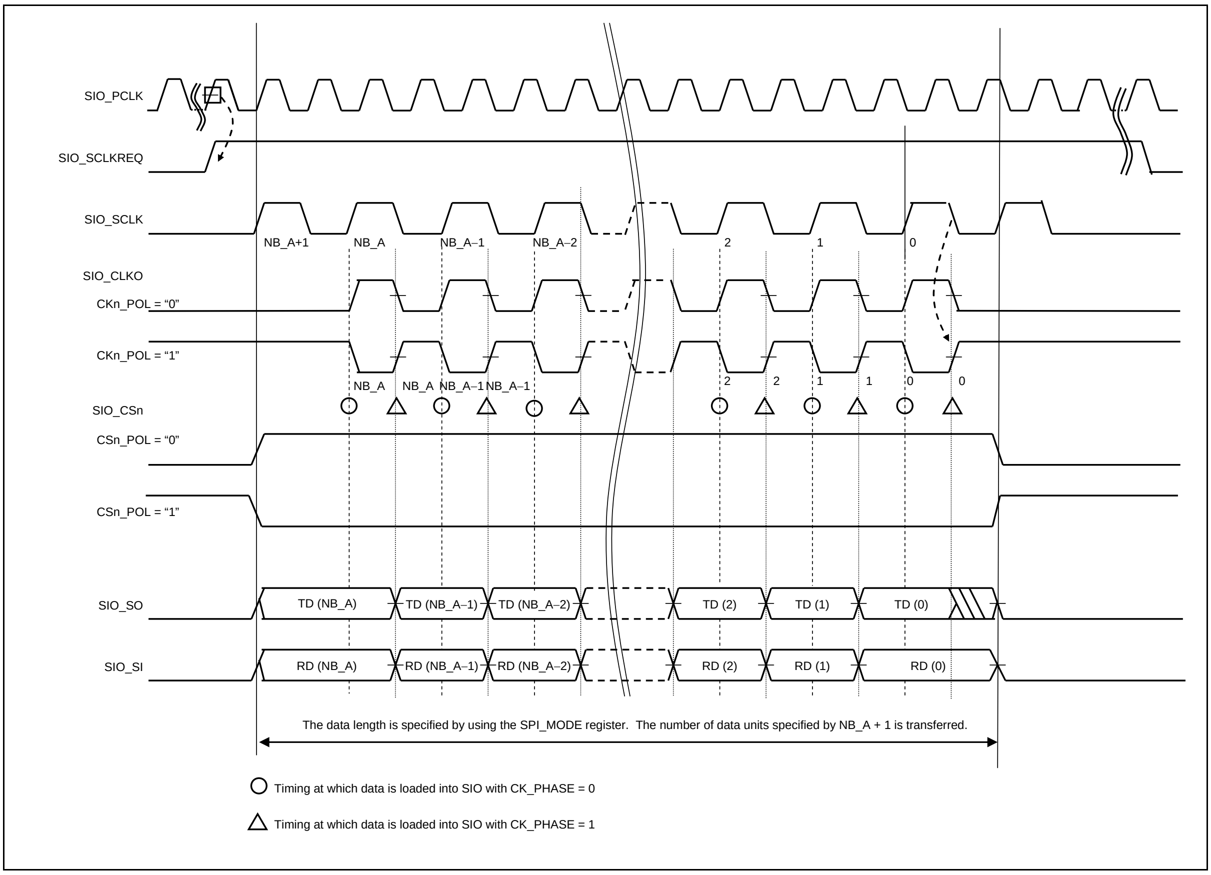


Figure 4-4 shows the timing of the SPI in master mode with the CKn_DLY bit[Note] of the SPI_POL register set to 1, and when SIO_CLKO is output half a clock cycle later than the output timing shown in Figure 4-4.

In master mode, when the received data is fetched depends on the CK_PHASE setting.

4.2.4 SPI interface timing 4 (CK0_DLY = 1 in slave mode)

Figure 4-5 shows SPI interface timing 4.

Figure 4-5. SPI Interface Timing 4

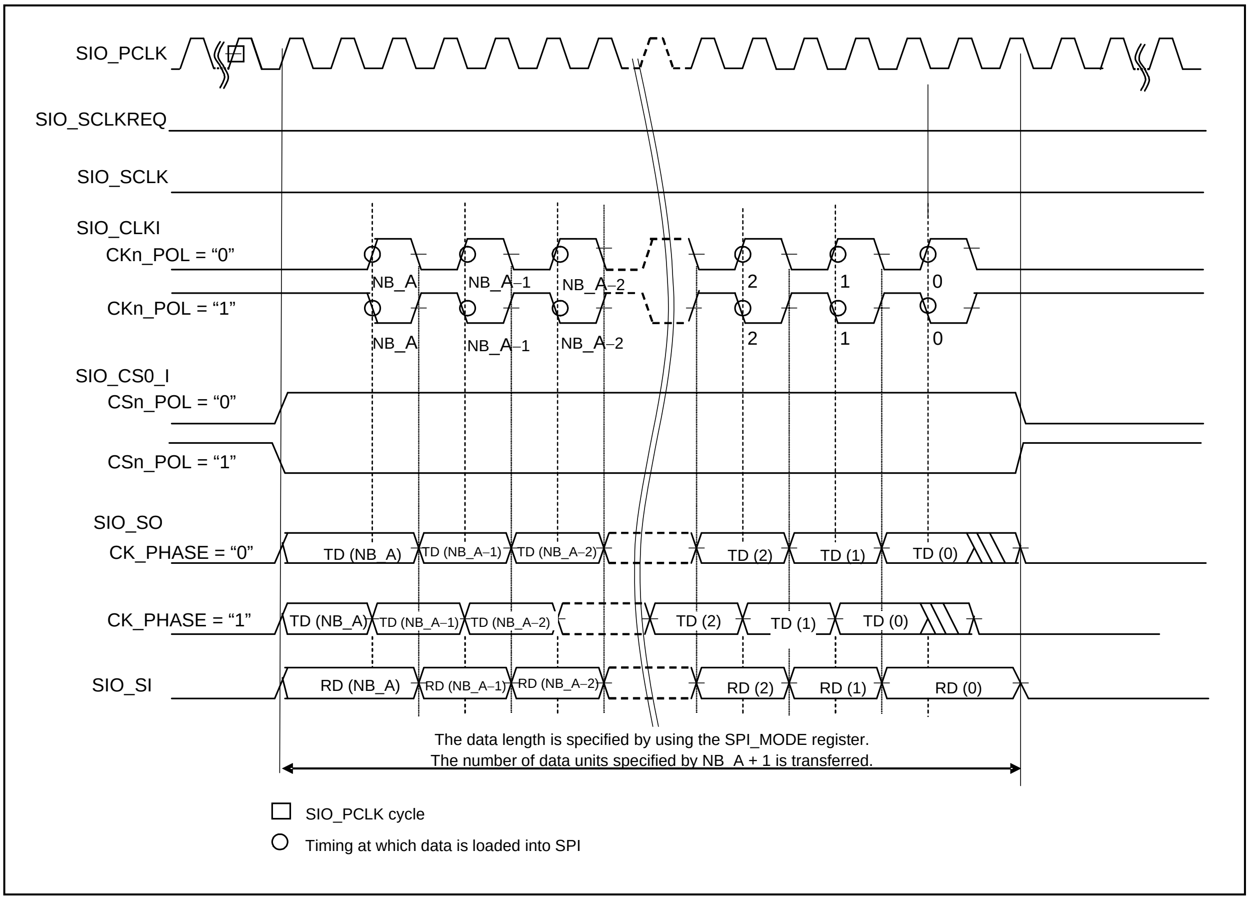
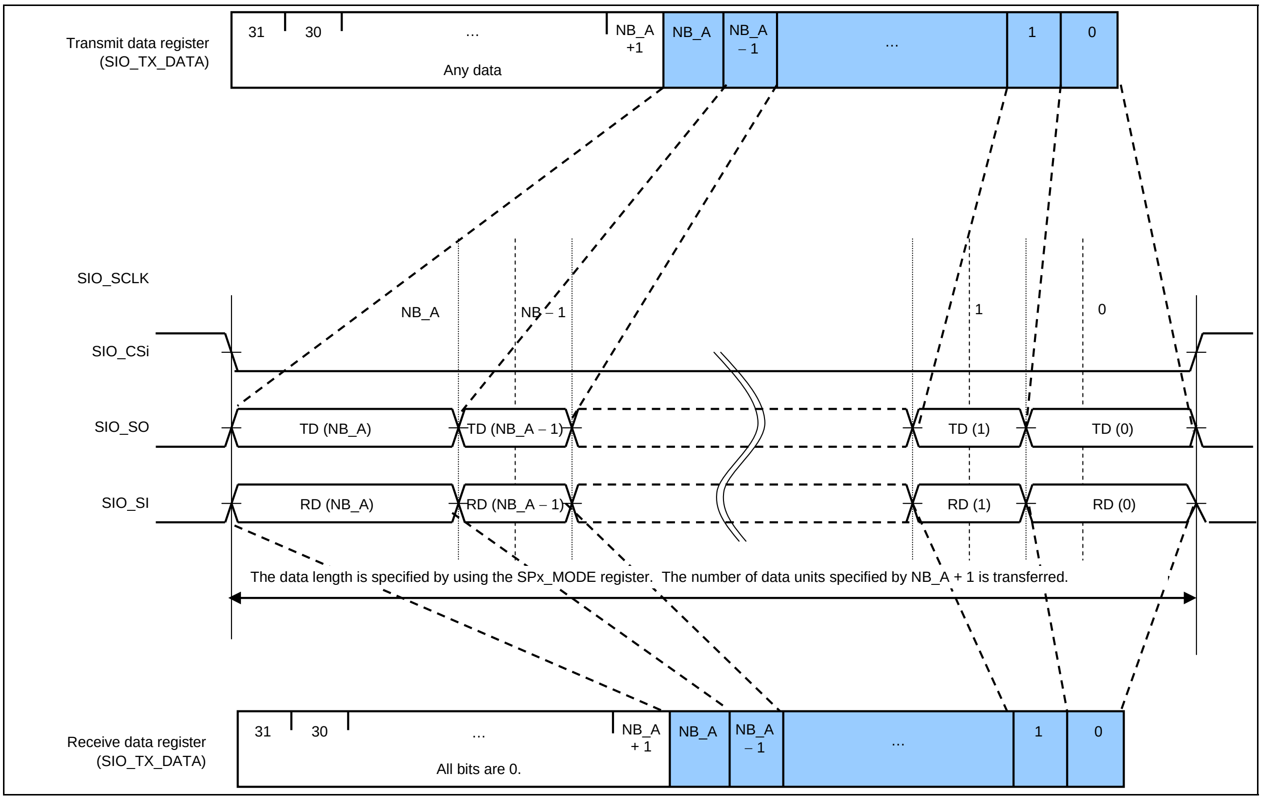


Figure 4-5 shows the timing of the SPI in slave mode with the CK0_DLY bit$^{\text{Note}}$ of the SPI_POL register set to 1, and when SIO_CLKI is output half a clock cycle later than the output timing shown in Figure 4-5.
In slave mode, when the transmission data is output depends on the CK_PHASE setting.

4.2.5 Correspondence between transmission data register, reception data register, and serial transfer data

Figure 4-6 shows the correspondence between the SPI transmission data register, the reception data register, and the serial transfer data.

Figure 4-6. Bit Correspondence



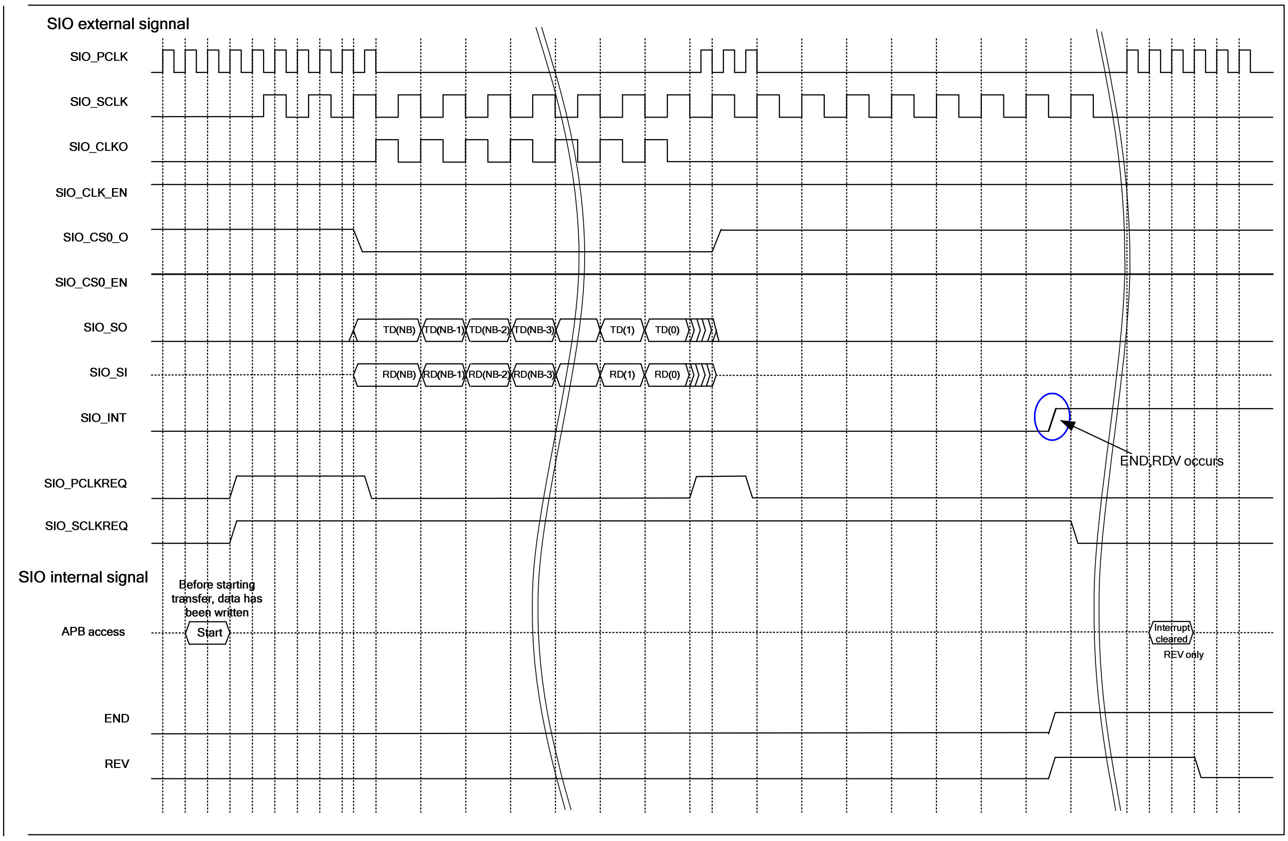
Serial data in the SPI transmission data register is transferred using SIO_SO starting from the MSB of the data specified by the NB_A bits of the SPI_MODE register.

Serial data received using SIO_SI is stored in the reception data register, starting from the MSB. The LSB of the reception data is stored in the LSB of the reception data register. Bits higher than the ones specified by the NB_A bits are filled with 0.

4.2.6 Master transfer in the CPU mode

Figure 4-7 shows the timing of master transfer in the CPU mode.

Figure 4-7. Master Transfer Timing in CPU Mode (CKn_DLY = 0)



Master transfer in the CPU mode is enabled by clearing the DMA and M_S bits to 0 in the SPI mode register (SPI_MODE). In master mode, the SPI interface is controlled by SIO_CLKO and SIO_CSn_O output from the SPI module.

If the RD bit of the control register (SPI_CONTROL) is set to 1, data is received serially starting from the MSB. If the WRT bit of the SPI_CONTROL register is set to 1, data stored in the SPI transmission data register (SPI_TX_DATA) is transmitted serially, starting from the MSB. The amount of data to transfer is the value of the NB_A bits specified in the SPI_MODE register + 1. SIO_SCLK is output for more cycles than the amount of data specified by NB_A, but SPI_CLKO is controlled by the SPI module and output to the SPI interface for number of cycles equivalent to the amount of data specified by NB_A + 1.

If the START bit of the SPI_CONTROL register is set to 1 while there is no data stored in the transmission FIFO buffer, transmission is held pending until data is written to the transmission FIFO buffer. Data is not received while the reception FIFO buffer is full.

The END flag of the SPI_STATUS register is set to 1 when frame transmission ends. For reception, the RDV flag is also set to 1. A TERR interrupt does not occur during master transfer in the CPU mode.

For transmission/reception, data is transferred according to the conditions for transmission.

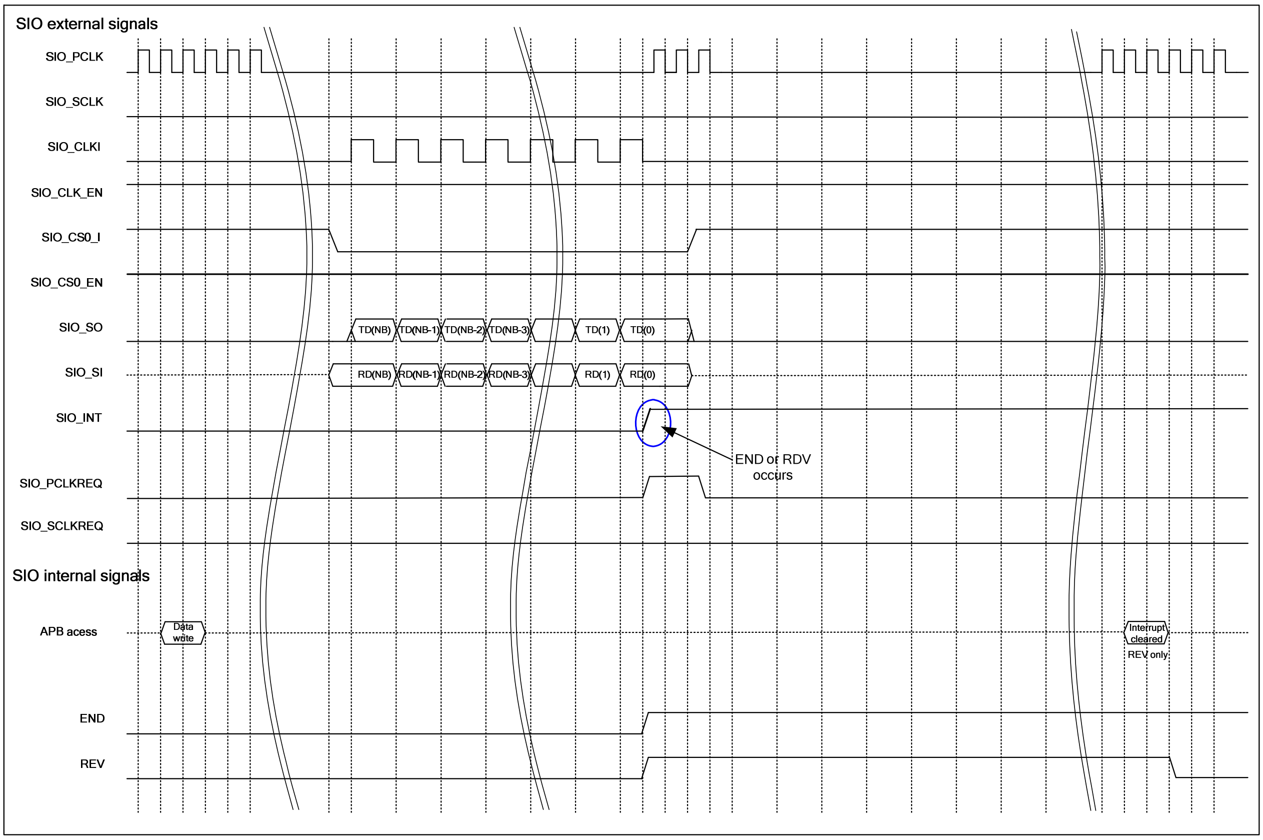
While the WRT bit is cleared to 0, SIO_SO outputs 0. In the same manner, if transmission is triggered by setting the WRT bit to 1 while the RD bit is 0, data sent using SPI_SI is ignored. At this time, an RDV interrupt request is not output. The START bit is reset to 0 in synchronization with SPI_PCLK when transfer of the frame for which transfer was started ends.

The RST bit of the SPI_CONTROL register resets the controller (registers) other than the SPI_MODE register and interrupts.

4.2.7 Slave transfer in the CPU mode

Figure 4-8 shows the timing of slave transfer in the CPU mode.

Figure 4-8. Slave Transfer Timing in CPU Mode (CKn_DLY = 0)



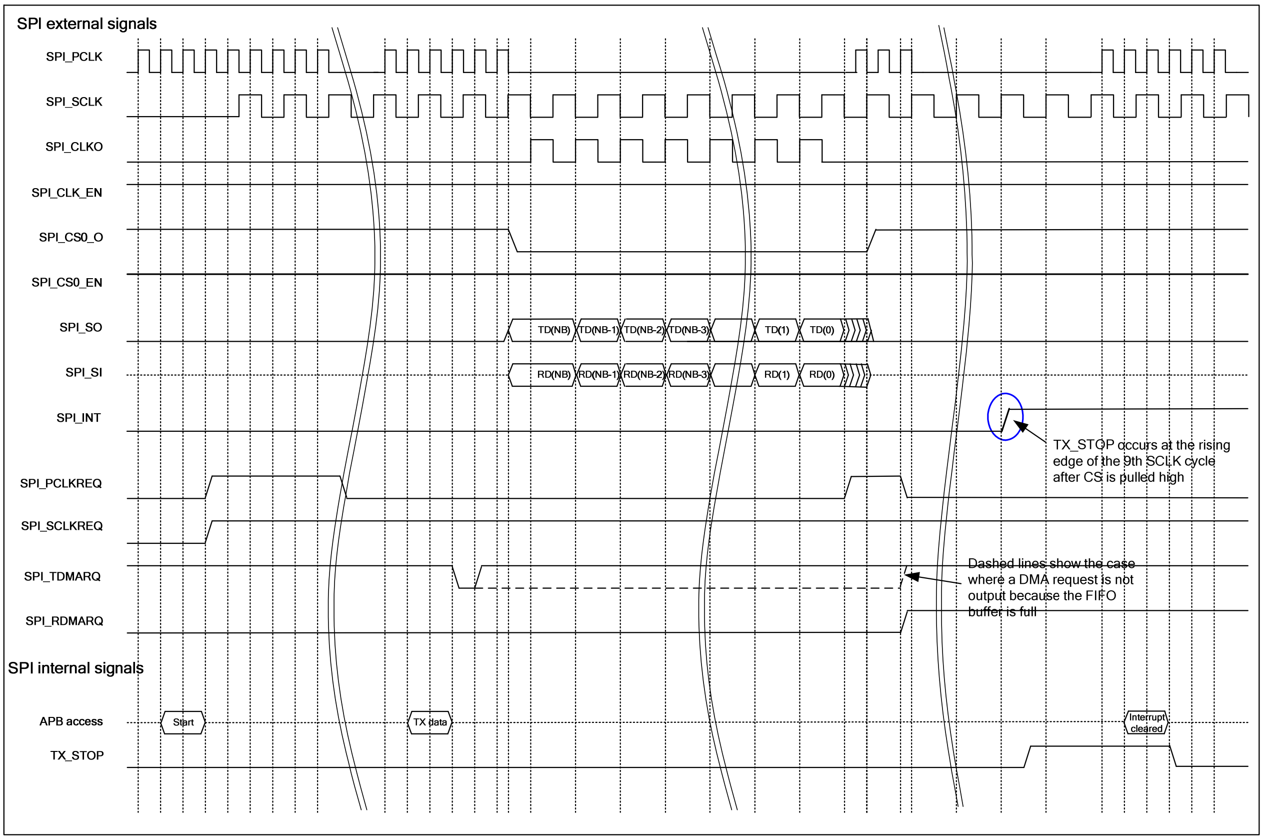
Slave transfer in the CPU mode is enabled by clearing the DMA bit to 0 and setting the M_S bit to 1 in the SPI mode register. In slave mode, the SPI is controlled by SIO_CLKI and SIO_CS0_I input from the SPI module. The internal operation is the same as that of master transfer in the CPU mode. However, a TERR interrupt occurs at the end of frame transmission.

Caution Set the RST bit when frame transfer is not being executed.

4.2.8 Master transfer in DMA mode (normal transfer)

Figure 4-9 shows the timing of normal master transfer in DMA mode.

Figure 4-9. Master Transfer Timing in DMA Mode (Normal Transfer) (CKn_DLY = 0)



TX_STOP occurs at the rising edge of the 9th SCLK cycle after CS is pulled high

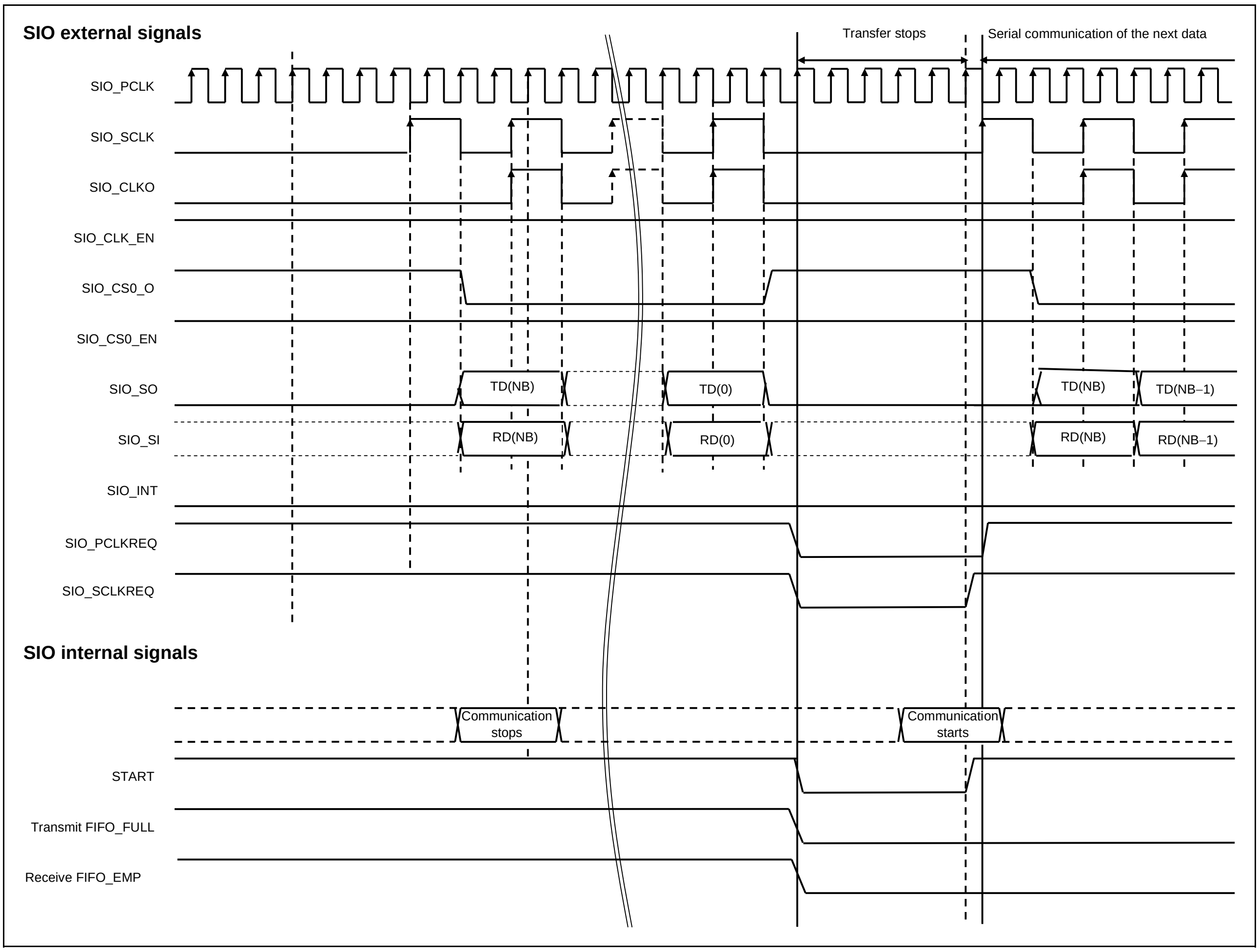
Dashed lines show the case where a DMA request is not output because the FIFO buffer is full

Master transfer in DMA mode is enabled by setting the DMA bit to 1 and clearing the M_S bit to 0 in the SPI mode register. The SPI interface operation is the same as that of master transfer in the CPU mode.
Transmission starts as soon as data is written to the transmission FIFO buffer by way of DMA.
Reception starts as soon as the RD and START bits are set to 1 in the SPI_CONTROL register. Data is not received if the reception FIFO buffer is full.
For transmission/reception, data is transferred according to the conditions for transmission.
When a serial transfer transaction ends, transfer of the next frame starts if the next data is stored in the transmission FIFO buffer and the reception FIFO buffer is not full. The interval for the next transfer can be specified from 1 to 16 SPI_SCLK cycles by using the CSW bits of the SPI polarity register.
The RDV, END, or TERR interrupt does not occur during master transfer in DMA mode, but the TX_STOP and RX_STOP interrupts do occur.

4.2.9 Master transfer in DMA mode (STOP to START)

Figure 4-10 shows the timing of STOP to START master transfer in DMA mode.

Figure 4-10. Master Transfer Timing in DMA Mode (STOP to START)

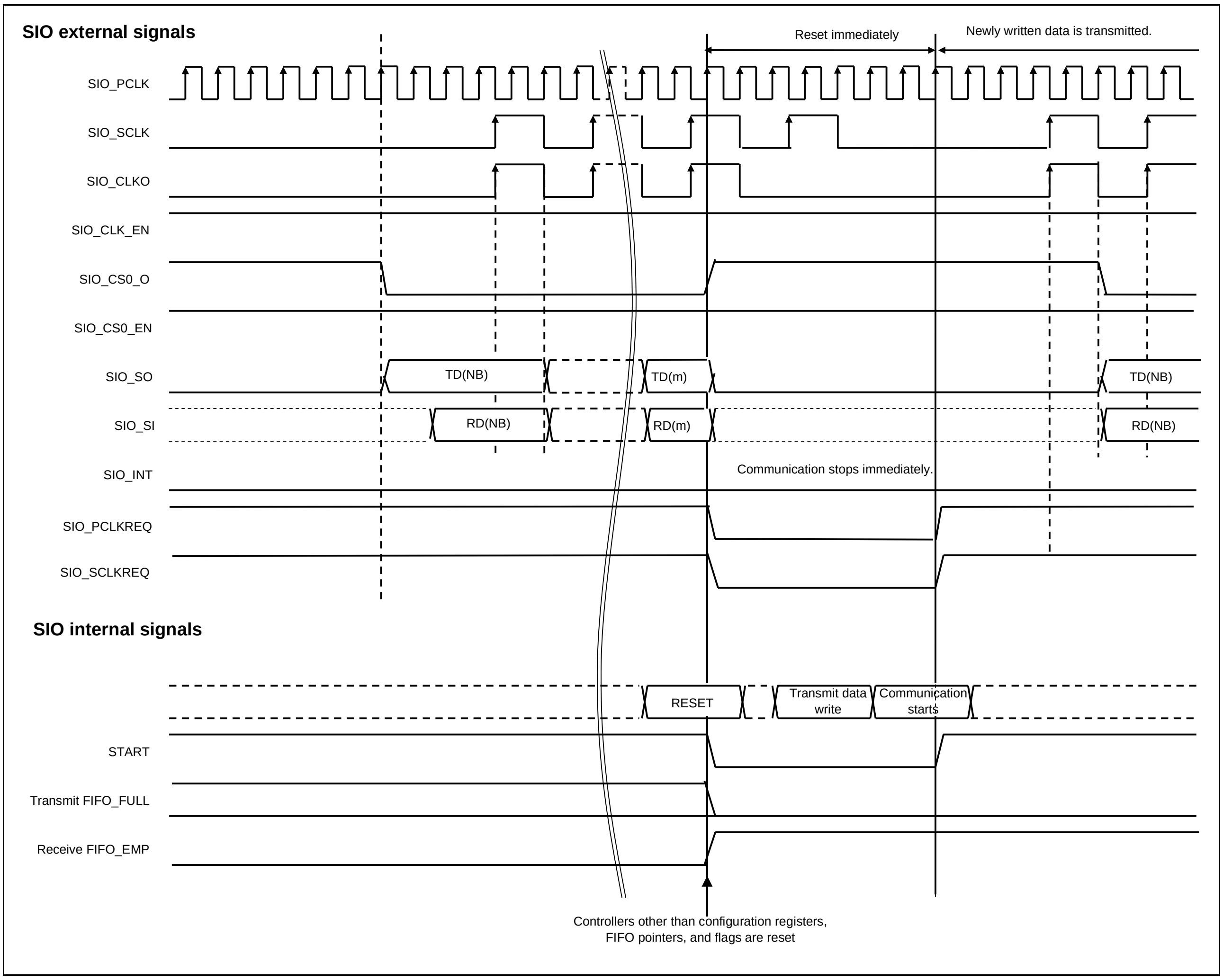


If the STOP bit of the SPI_CONTROL register is set to 1 while SPI transfer is being executed, transfer ends when transfer of the current frame is complete. The received data is stored in the reception FIFO buffer.
If the START bit of the SPI_CONTROL register is set to 1 immediately after the STOP bit was set to 1, transfer starts from the pointer to the position next to the position where the previous transfer was stopped.

4.2.10 Master transfer in DMA mode (RESET to START)

Figure 4-11 shows the timing of RESET to START master transfer in DMA mode.

Figure 4-11. Master Transfer Timing in DMA Mode (RESET to START)

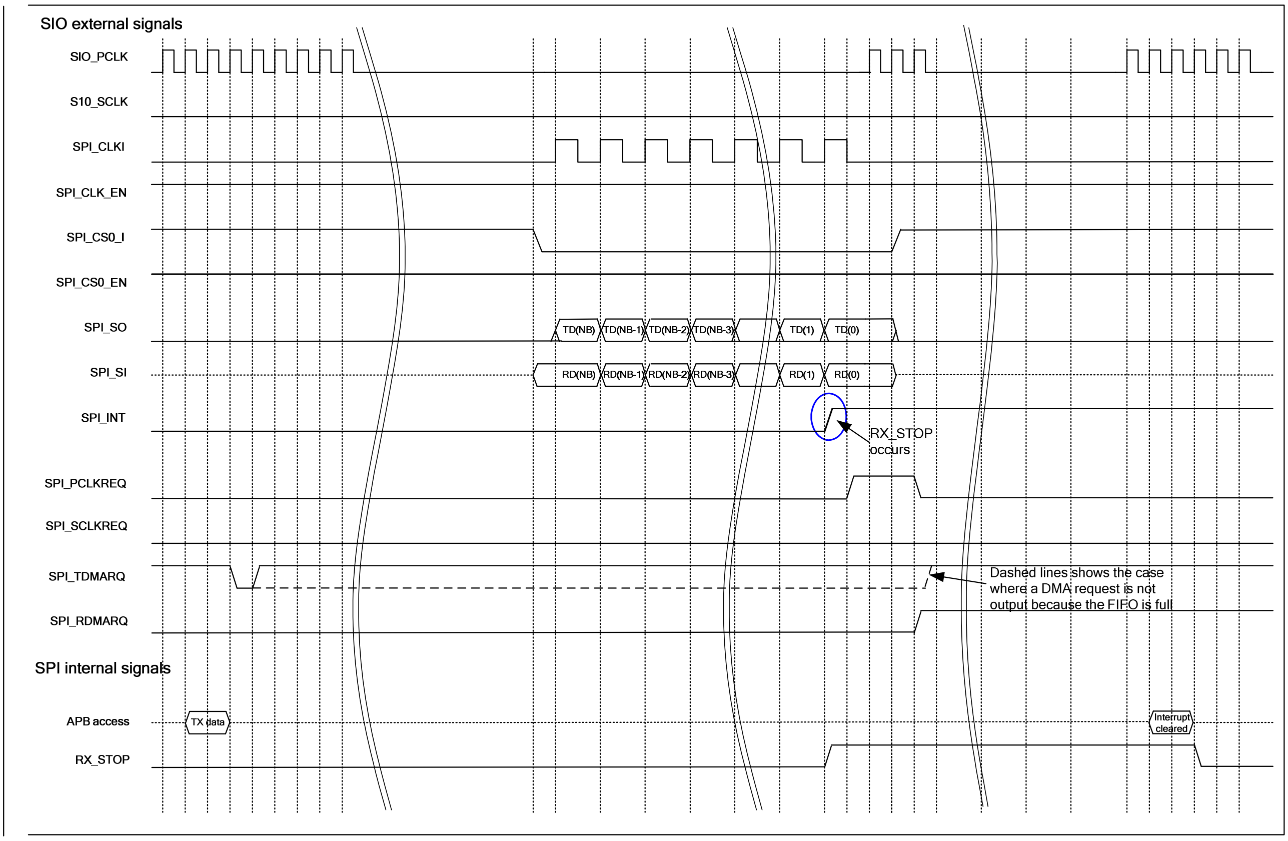


If the RST bit of the SPI_CONTROL register is set to 1 while SPI transfer is being executed, transfer ends immediately. At the same time, all the controllers (registers) other than configuration registers are reset.

4.2.11 Slave transfer in DMA mode (normal transfer)

Figure 4-12 shows the timing of normal slave transfer in DMA mode.

Figure 4-12. Slave Transfer Timing in DMA Mode (Normal Transfer) (CKn_DLY = 0)



Slave transfer in DMA mode is enabled by setting the DMA and M_S bits to 1 in the SPI mode register. In slave mode, the SPI is controlled by SIO_CLKI and SIO_CS0_I input from the SPI module. The operation of the SPI interface is the same as that of master transfer in DMA mode.

The RDV and END interrupts do not occur during slave transfer in DMA mode, but the TX_UDR, RX_OVR, and TERR interrupts occur.

4.3 Audio Mode

4.3.1 Mode

In audio mode, there are 7 sub-modes in whitch data can be transferred using the serial interface (modes 0 to 6). The mode to be used can be specified by setting bits 2 to 0 of the PCM operating mode setting register (PCM_FUNC_SEL).

The valid edges used for each mode are shown in the tables below.

The SIO module supports the I2S, and AC97 (mode5) standards.

Table 4-1. Valid Edges Used for Each Mode (PCM_FUNC_SEL. CLK_INV = 0)

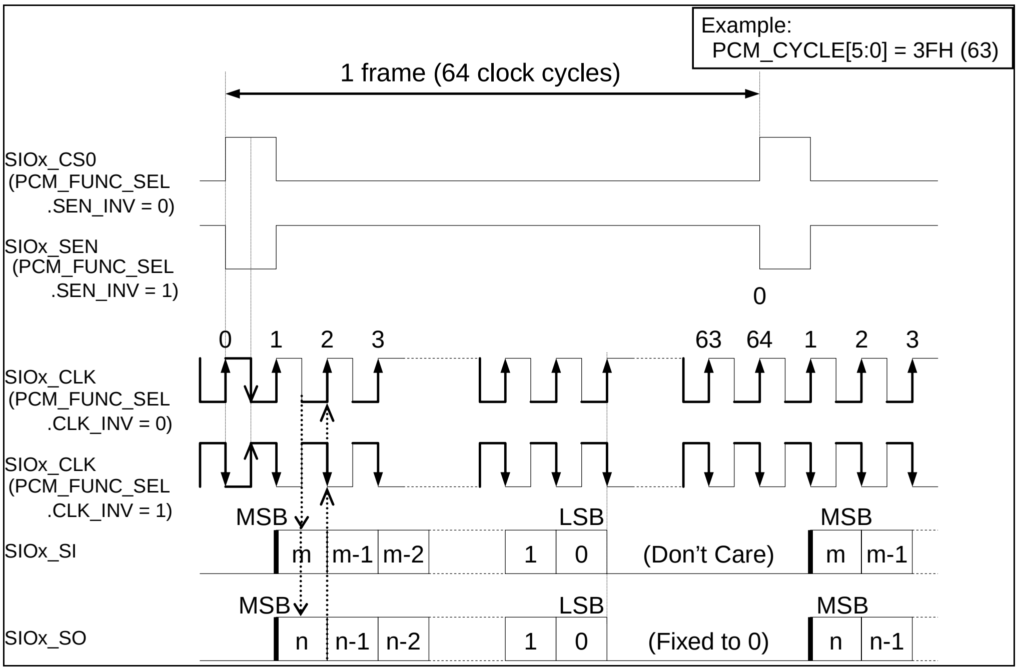
No.	Setting Mode	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
PCM_FUNC_SEL[2:0] Setting		000b	001b	010b	011b	100b	101b	110b
1	SEN output edge (master)	↑	↓	↓	↓	↓	↑	↓
	SEN input latch edge (slave)	↓	↑	↑	↑	↑	↓	↑
2	SI input latch edge	↓	↑	↑	↑	↑	↓	↑
3	SO output edge	↑	↑	↓	↓	↓	↑	↑

Table 4-2. Valid Edges Used for Each Mode (PCM_FUNC_SEL. CLK_INV = 1)

No.	Setting Mode	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
PCM_FUNC_SEL[2:0] Setting		000b	001b	010b	011b	100b	101b	110b
1	SEN output edge (master)	↓	↑	↑	↑	↑	↓	↑
	SEN input latch edge (slave)	↑	↓	↓	↓	↓	↑	↓
2	SI input latch edge	↑	↓	↓	↓	↓	↑	↓
3	SO output edge	↓	↓	↑	↑	↑	↓	↓

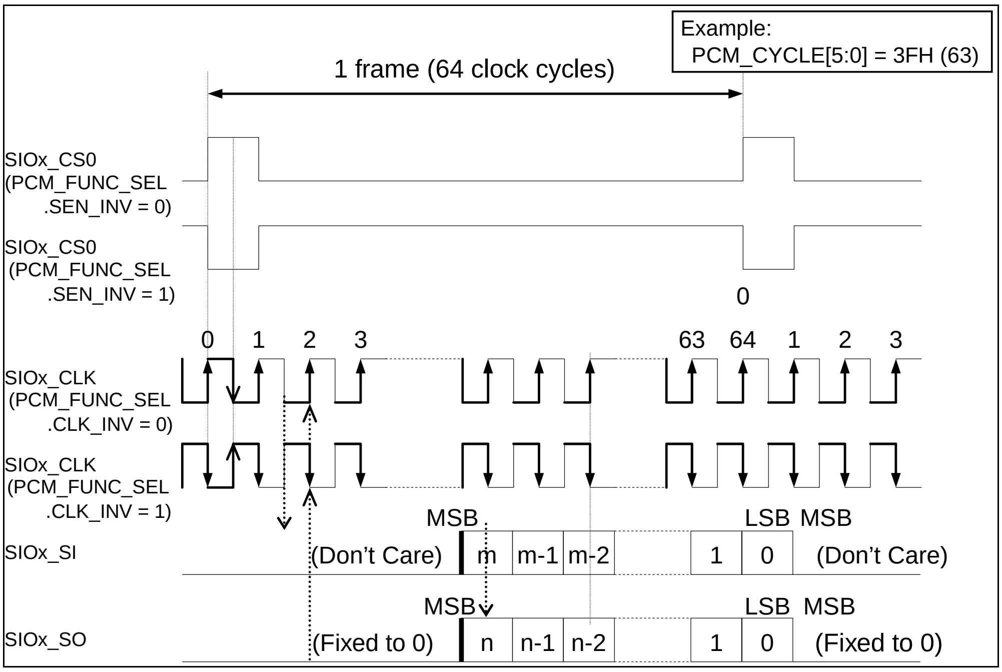
(1) Mode 0 (left-justified)

Figure 4-13. Operation Timing in Mode 0 (Left-Justified)



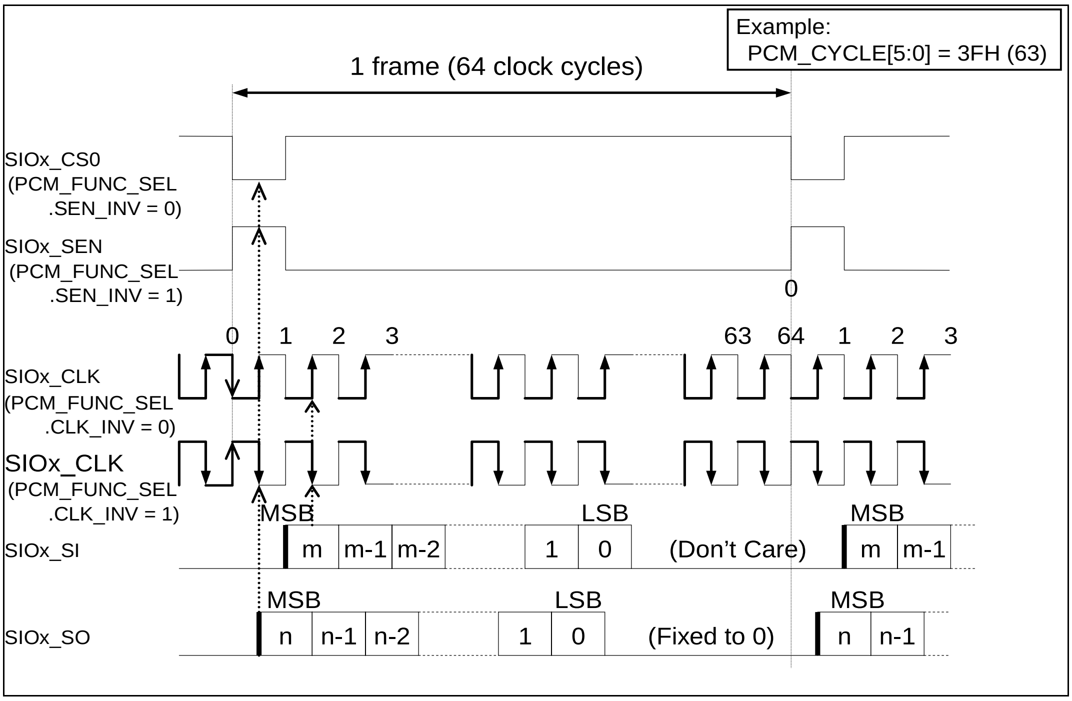
(2) Mode 0 (right-justified)

Figure 4-14. Operation Timing in Mode 0 (Right-Justified)



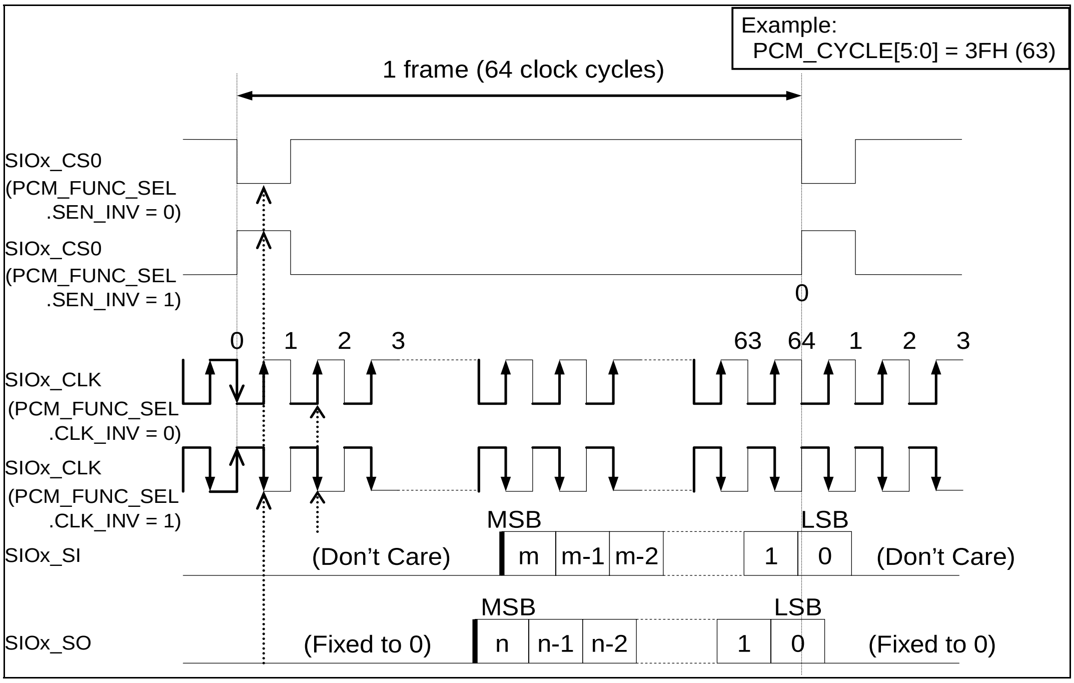
(3) Mode 1 (left-justified)

Figure 4-15. Operation Timing in Mode 1 (Left-Justified)



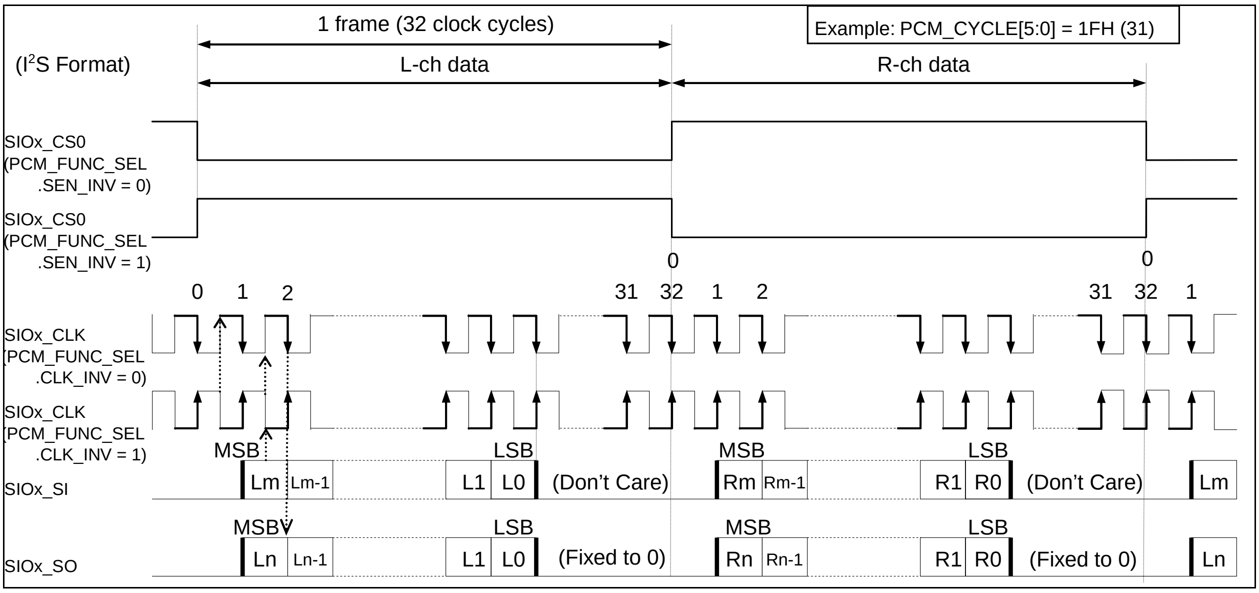
(4) Mode 1 (right-justified)

Figure 4-16. Operation Timing in Mode 1 (Right-Justified)



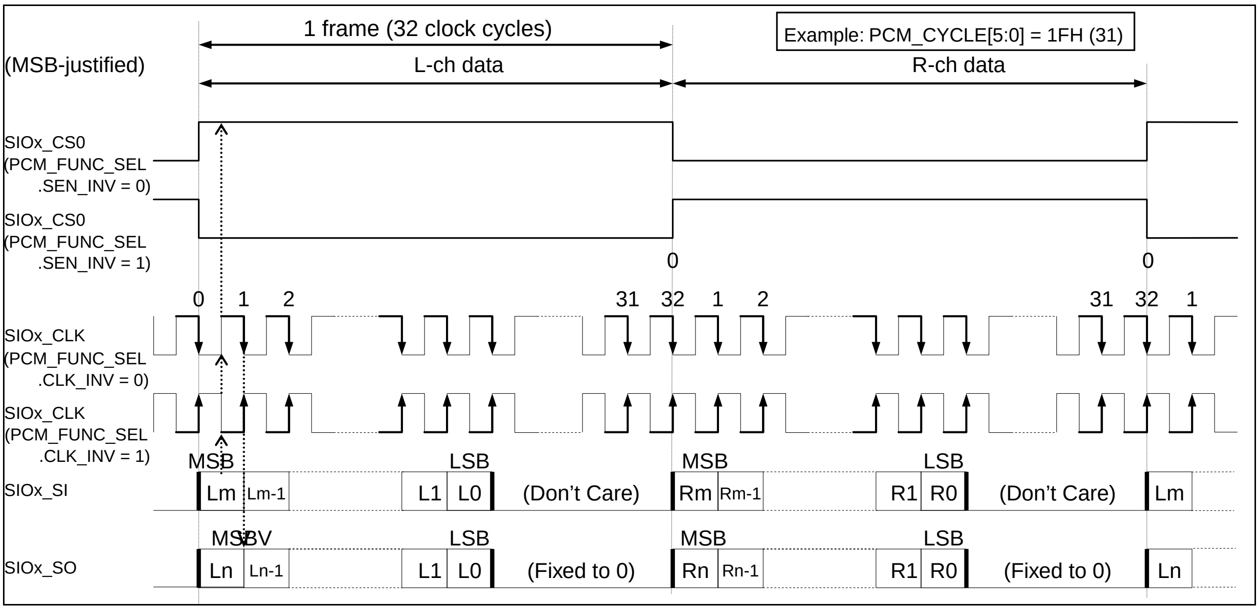
(5) Mode 2 (left-justified)

Figure 4-17. Operation Timing in Mode 2



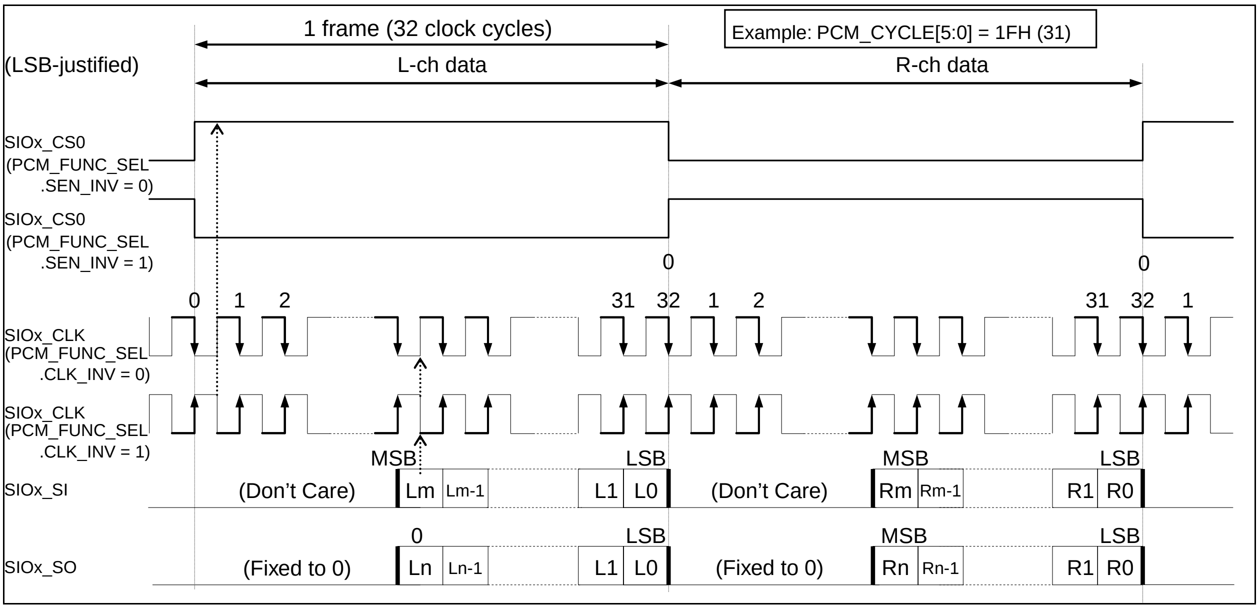
(6) Mode 3 (left-justified)

Figure 4-18. Operation Timing in Mode 3



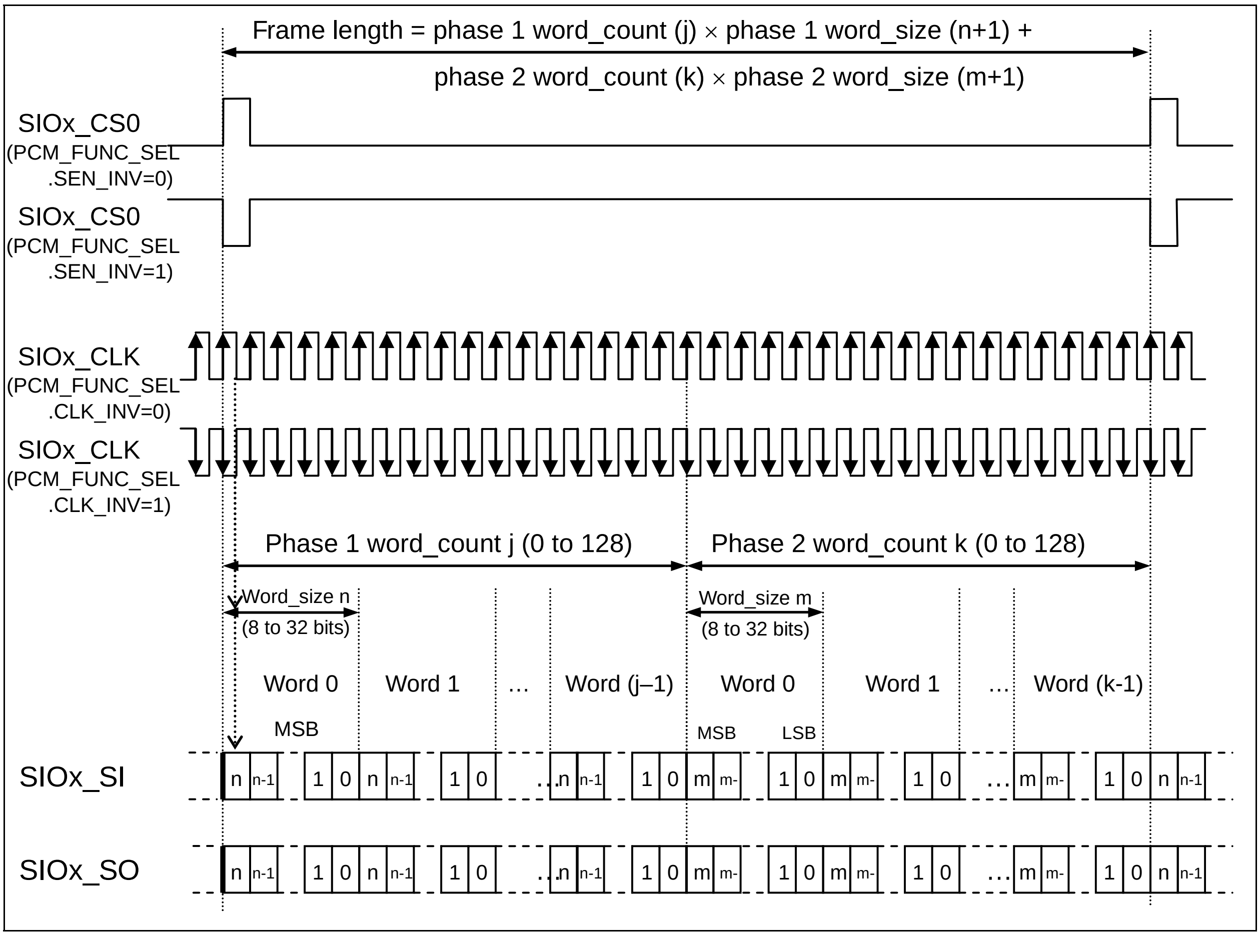
(7) Mode 4 (right-justified)

Figure 4-19. Operation Timing in Mode 4


1 frame (32 clock cycles)
Example: PCM_CYCLE[5:0] = 1FH (31)
(LSB-justified)
L-ch data
R-ch data
SIOx_CS0 (PCM_FUNC_SEL .SEN_INV = 0)
SIOx_CS0 (PCM_FUNC_SEL .SEN_INV = 1)
0 1 2 31 32 1 2 31 32 1
0
0
SIOx_CLK (PCM_FUNC_SEL .CLK_INV = 0)
SIOx_CLK (PCM_FUNC_SEL .CLK_INV = 1)
SIOx_SI
(Don't Care)
MSB
Lm
Lm-1
LSB
L1
L0
(Don't Care)
MSB
Rm
Rm-1
LSB
R1
R0
SIOx_SO
(Fixed to 0)
0
Ln
Ln-1
LSB
L1
L0
(Fixed to 0)
MSB
Rn
Rn-1
LSB
R1
R0

(8) Mode 5

Figure 4-20. Operation Timing in Mode 5



When word_count (number of words) is set to 0 for phase 1 or 2, a single-phase operation is performed.
It is prohibited to set word_count to 0 for both phase 1 and phase 2.
In simultaneous transmission/reception operation, specify the same number of bits and words for transmission and reception in each phase[Note 1].
Word_size for phase 1 might differ from that for phase 2, and word_count for phase 1 might differ from that for phase 2.

Notes 1. Be sure to specify SIB = SOB and SIB2 = SOB2.

2. SIB ≠ SIB2, SOB ≠ SOB2, and CYC_VAL ≠ CYC_VAL2 can be specified. Be sure to specify SIB = SOB and SIB2 = SOB2.

When using AC97 mode, set *j* to 1, *n* to 15, *k* to 12, and *m* to 19. Then set PCM_FUNC_SEL.AC97_SYNC to 1. The timing of operations in mode 5 of AC97 mode is as follows:

Figure 4-21. Operation Timing in Mode 5 (AC97)

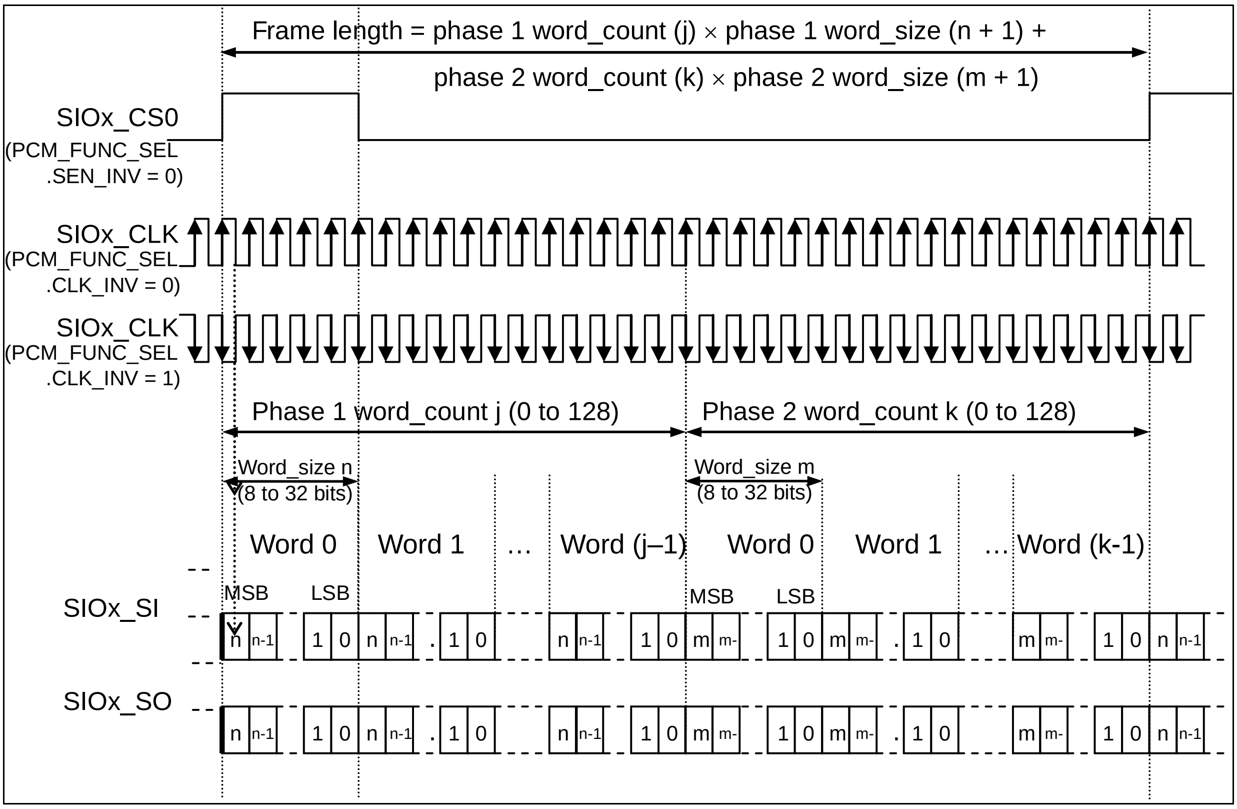


Figure 4-22. Operation Timing in Mode 5 (1 Word per Phase)

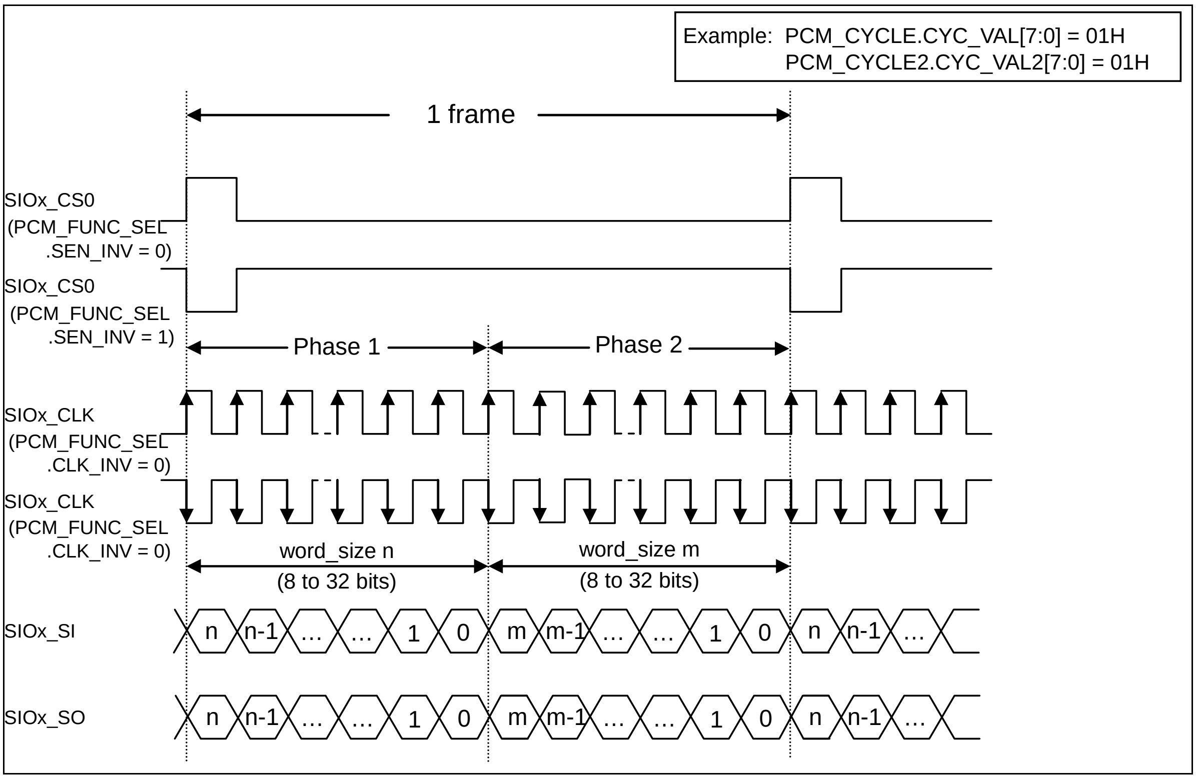
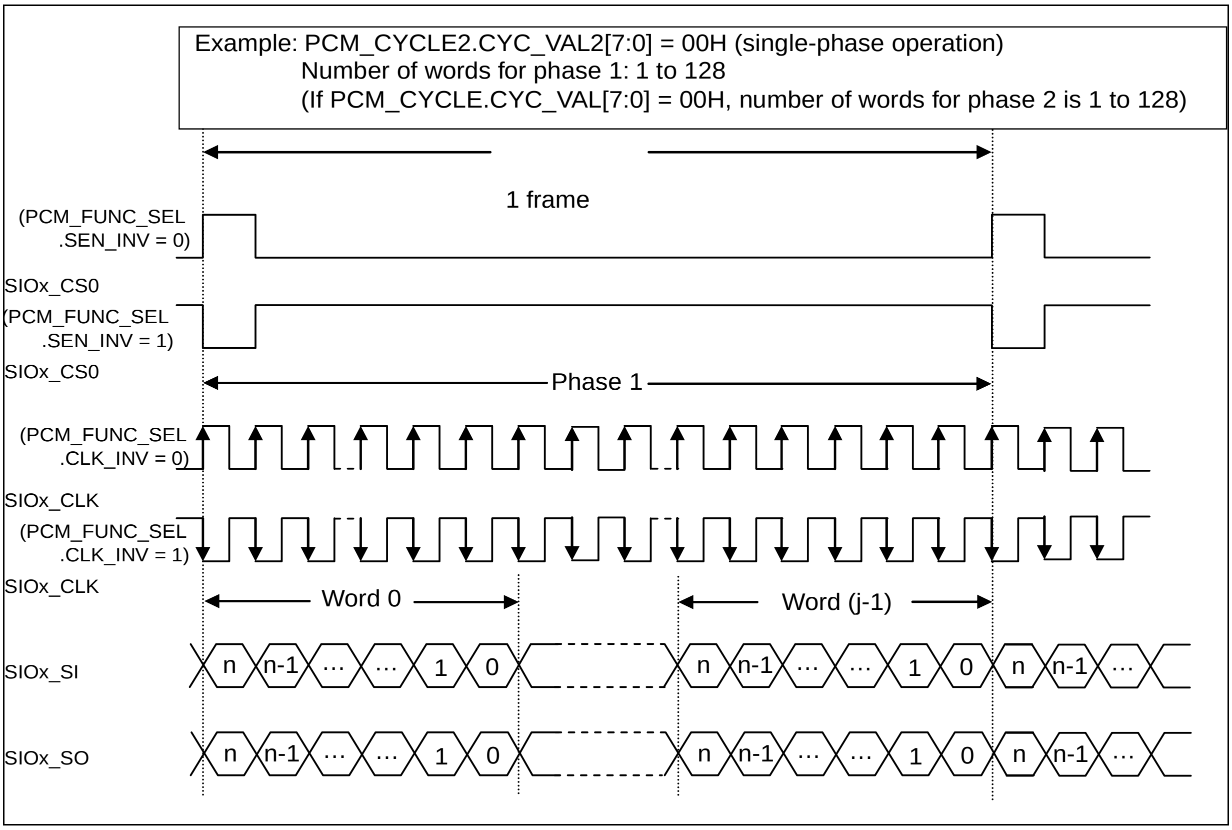
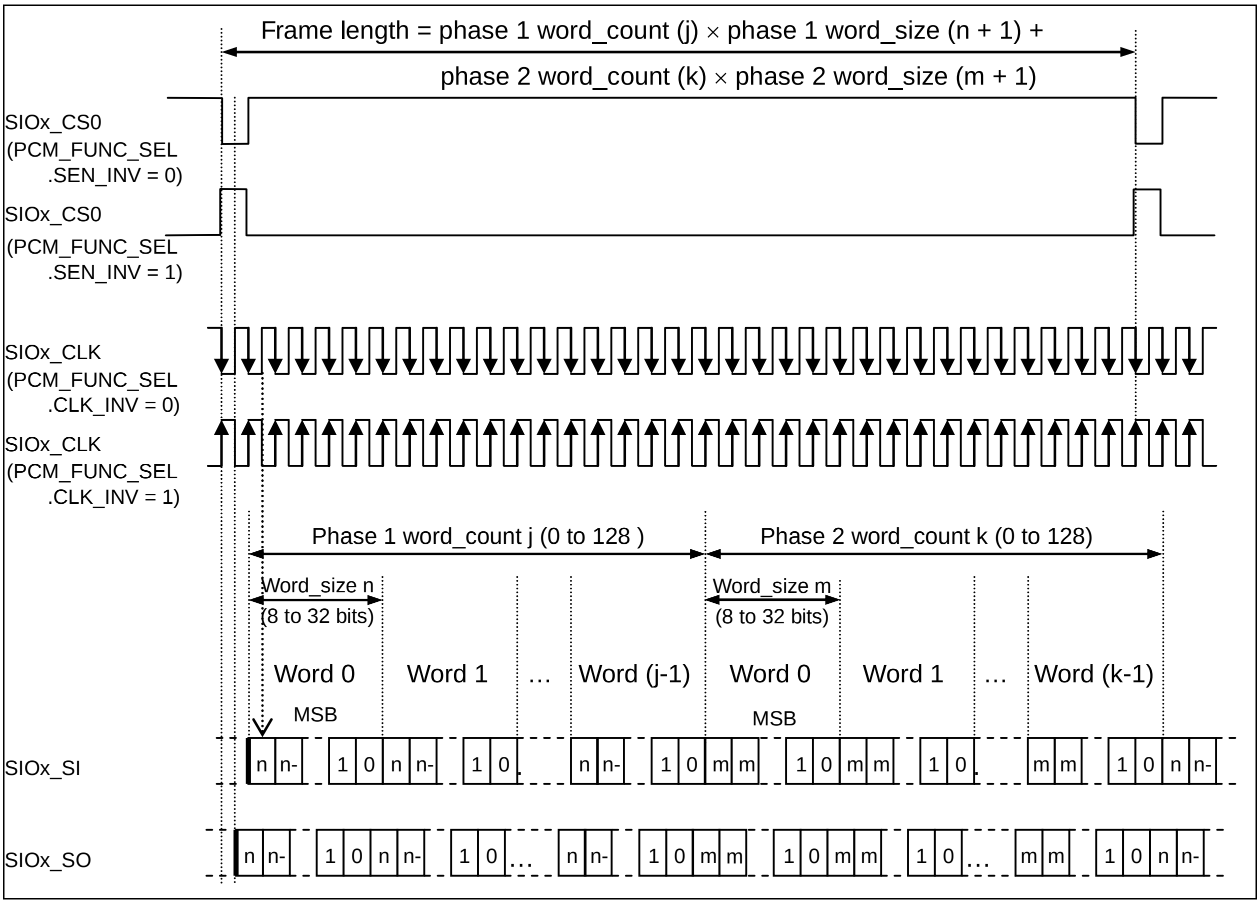


Figure 4-23. Operation Timing in Mode 5 (Single Phase)


Example: PCM_CYCLE2.CYC_VAL2[7:0] = 00H (single-phase operation)
Number of words for phase 1: 1 to 128
(If PCM_CYCLE.CYC_VAL[7:0] = 00H, number of words for phase 2 is 1 to 128)
1 frame
(PCM_FUNC_SEL
.SEN_INV = 0)
SIOx_CS0
(PCM_FUNC_SEL
.SEN_INV = 1)
SIOx_CS0
Phase 1
(PCM_FUNC_SEL
.CLK_INV = 0)
SIOx_CLK
(PCM_FUNC_SEL
.CLK_INV = 1)
SIOx_CLK
Word 0
Word (j-1)
SIOx_SI
SIOx_SO
n n-1 … … 1 0 n n-1 … … 1 0 n n-1 …

(9) Mode 6

Figure 4-24. Operation Timing in Mode 6



When word_count (number of words) is set to 0 for phase 1 or 2, a single-phase operation is performed.

It is prohibited to clear word_count to 0 for both phase 1 and phase 2.

In simultaneous transmission/reception operation, specify the same number of bits and words for transmission and reception in each phase[Note 1].

Word_size for phase 1 might differ from that for phase 2, and word_count for phase 1 might differ from that for phase 2.

Notes 1. Be sure to specify SIB = SOB and SIB2 = SOB2.

2. SIB ≠ SIB2, SOB ≠ SOB2, and CYC_VAL ≠ CYC_VAL2 can be specified. Be sure to specify SIB = SOB and SIB2 = SOB2.

Figure 4-25. Operation Timing in Mode 6 (1 Word per Phase)

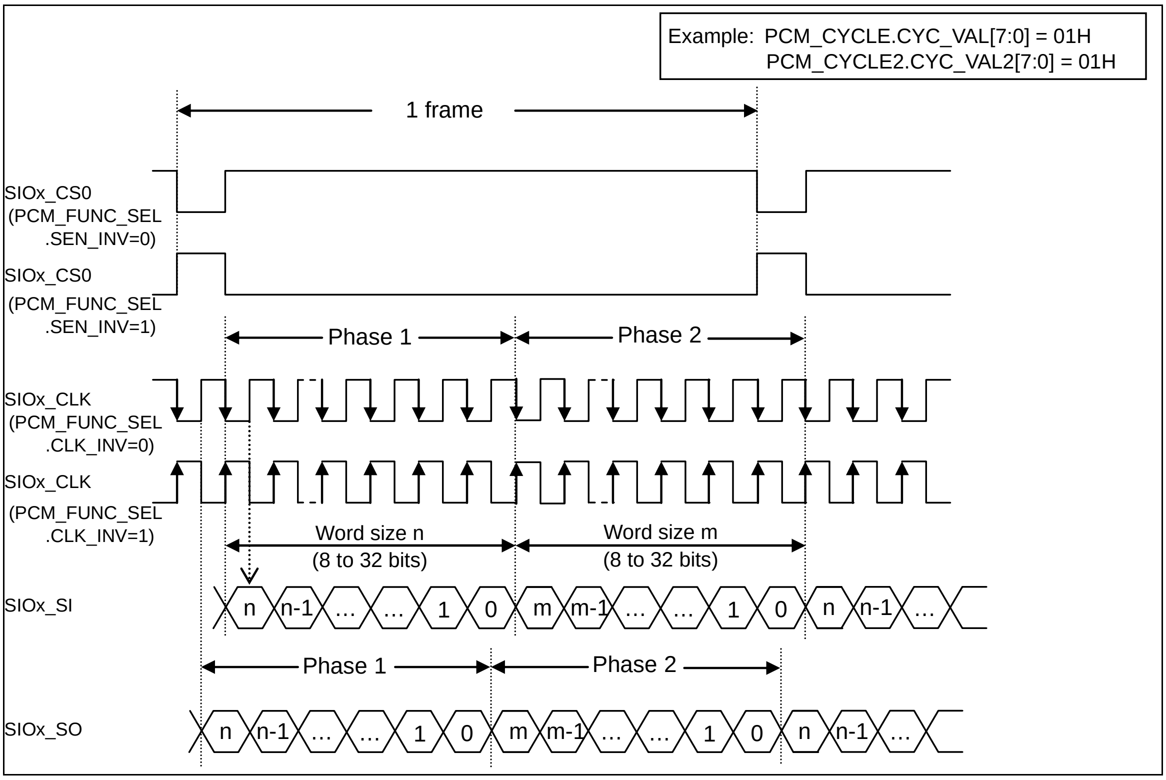
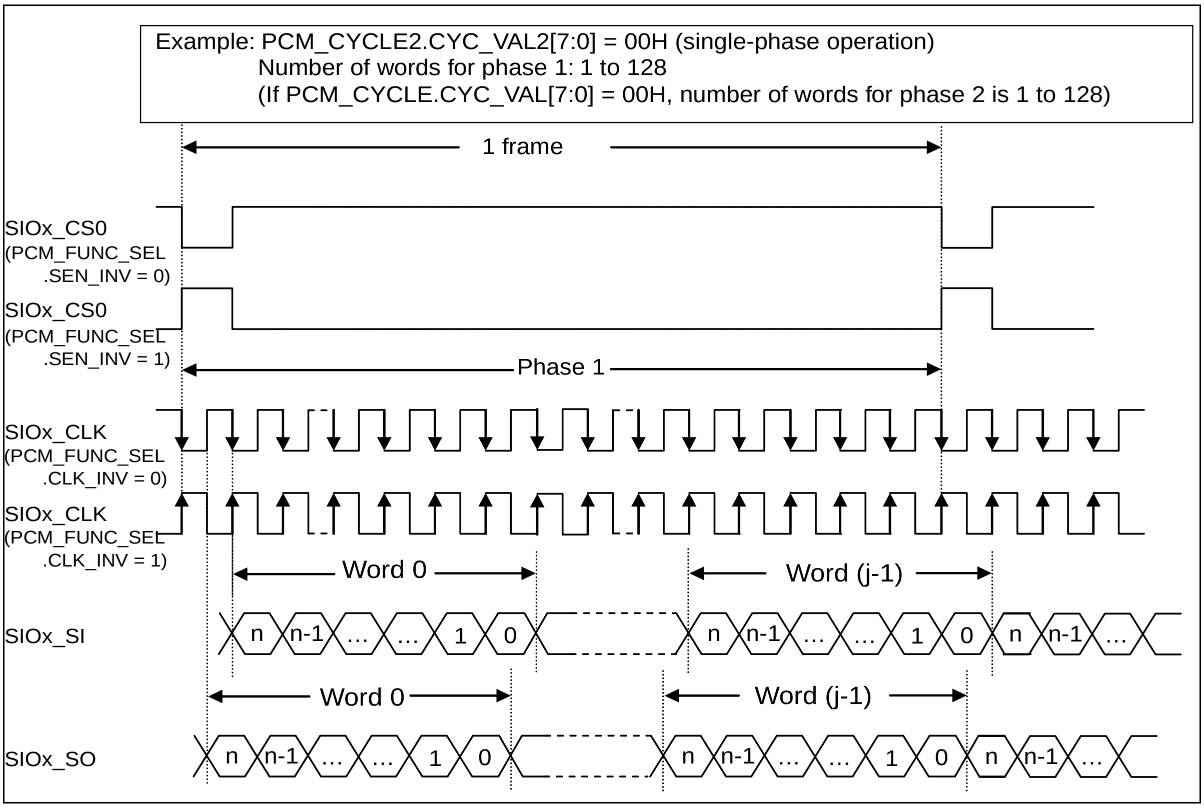

Example: PCM_CYCLE.CYC_VAL[7:0] = 01H
PCM_CYCLE2.CYC_VAL2[7:0] = 01H
1 frame
SIOx_CS0 (PCM_FUNC_SEL .SEN_INV=0)
SIOx_CS0 (PCM_FUNC_SEL .SEN_INV=1)
Phase 1
Phase 2
SIOx_CLK (PCM_FUNC_SEL .CLK_INV=0)
SIOx_CLK (PCM_FUNC_SEL .CLK_INV=1)
Word size n (8 to 32 bits)
Word size m (8 to 32 bits)
SIOx_SI
n n-1 … … 1 0 m m-1 … … 1 0 n n-1 …
SIOx_SO

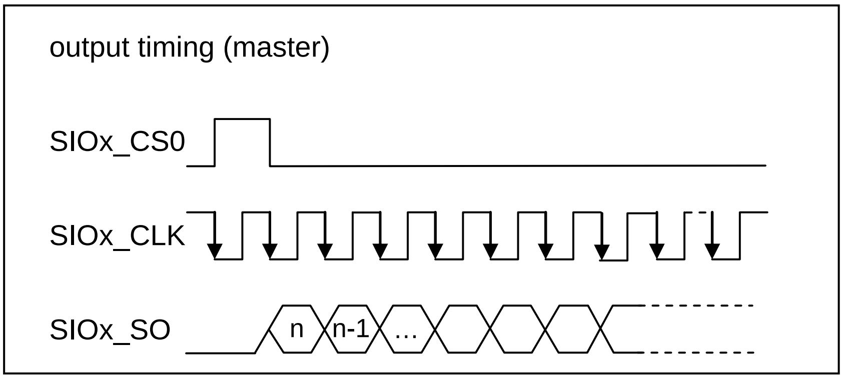
Figure 4-26. Operation Timing in Mode 6 (Single Phase)


Example: PCM_CYCLE2.CYC_VAL2[7:0] = 00H (single-phase operation)
Number of words for phase 1: 1 to 128
(If PCM_CYCLE.CYC_VAL[7:0] = 00H, number of words for phase 2 is 1 to 128)
1 frame
SIOx_CS0 (PCM_FUNC_SEL .SEN_INV = 0)
SIOx_CS0 (PCM_FUNC_SEL .SEN_INV = 1)
Phase 1
SIOx_CLK (PCM_FUNC_SEL .CLK_INV = 0)
SIOx_CLK (PCM_FUNC_SEL .CLK_INV = 1)
Word 0
Word (j-1)
SIOx_SI
n n-1 … … 1 0
SIOx_SO

(10) Mode 6

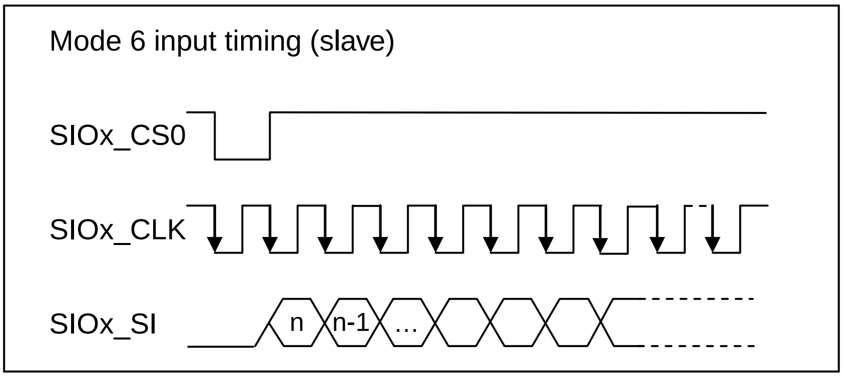
This macro can support transfers data as the master by setting PCM_FUNC_SEL.SEN_INV to 1.

Figure 4-27. Output Timing (in Master Mode)



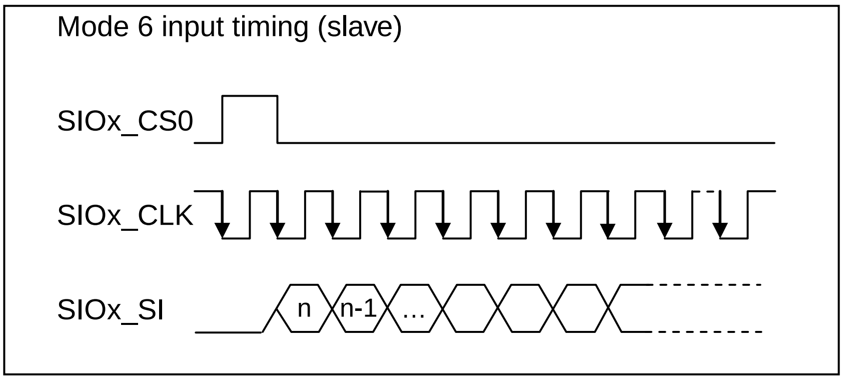
The (slave) format in mode 6 is shown below.

Figure 4-28. Input Timing in PCM Mode 6 (in Slave Mode)



The mode 6 format is supported by setting PCM_FUNC_SEL.SEN_INV to 1 and inverting the polarity of PCM_SEN. The format is shown below.

Figure 4-29. Input Timing in Mode 6 (in Slave Mode)



4.3.2 PCM macro operation

The settings of the PCM macro are as follows after the macro is reset:

PCM_TXRX_EN:	Cleared to 0.	Transmission and reception are disabled.
PCM_RAW:	Cleared to 0.	All the interrupt sources are cleared.
PCM_ENSET:	Cleared to 0.	All the interrupt sources are disabled.
Other FIFO pointers:	Cleared.	

(1) Operation in transmission block

The DMA request signal is asserted when the state is *waiting for transmission* or *transmitting* and the reception FIFO buffer has space for data (space for 8 words or more in the case of 8-burst transfer). The request signal is deasserted when transmission stops or is disabled, or the transmission FIFO buffer becomes full. The request signal is not deasserted each time the AHB is accessed.

Table 4-3. Transmission Pin Operation Sources

Operation Source	Description	State		
		Transmission Disabled	Transmission Enabled	Transmission Stopped
SIOx_DMA_TDMARQ pin	DMA transmission request	Fixed to off	On or off	Fixed to off
SIOx_SO pin	Serial data transmission	Fixed to 0	Data transmitted	Fixed to 0
SIOx_INT pin	INT interrupt request	**Note 1**	With assertion	**Note 2**
SIOx_CS0 pin (in master mode)	Serial data synchronization signal	Fixed to 0	Data transmitted	Data transmitted
TX_STR flag	Transmission data write enable flag	0 (off)	On or off	On or off retained
TX_ORE flag	Transmission overrun error detected	0 (off)	On or off	On or off retained
TX_URE flag	Transmission underrun error detected	0 (off)	On or off	On or off retained
TX_FRE flag	Transmission synchronization error detected	0 (off)	On or off	On or off
Transmission FIFO	Transmission FIFO buffer	Write disabled	Write enabled	Write disabled^{Note 3}
TX_W_CONT	Write counter of transmission FIFO buffer	0	Counter value incremented	Counter value retained
TX_R_CONT	Read counter of transmission FIFO buffer	0	Counter value incremented	Counter value retained
TX_WP_NUM^{Note 4}	Word number corresponding to FIFO buffer pointed to by write pointer in transmission FIFO buffer	0	Shows word number	Word number retained
TX_PHASE^{Note 4}	Write pointer in transmission FIFO buffer points to either phase 1 or 2	0	Shows phase	Phase retained

Notes **1.** The SIOx_INT pin status depends on the reception interrupt source.

2. Asserted if the transmission interrupt source is unmasked.

3. Writing to the transmission FIFO buffer while transmission is disabled might cause a transmission overrun error.

4. Fixed to 0 in modes other than modes 5 and 6.

State transitions

1) *Transmission stopped* to *waiting for transmission*

This transition occurs when the TX_EN bit of the PCM_TXRX_EN register is set to 1.

2) *Waiting for transmission* to *transmitting*

This transition occurs when the amount of data accumulated in the transmission FIFO buffer reaches the serial transmission start threshold (specified using the TX_TIM bit of the PCM_FUNC_SEL register), and then a rise of the SEN signal is detected.

3) Any state to *transmission disabled*

This transition occurs when the TX_ENCLR bit of the PCM_TXRX_DIS register is set to 1.

If a transmission error occurs, the state shifts to transmission stopped, and serial transmission stops. In the transmission stopped state, all internal statuses are held, so transmission can be resumed by using the following procedure:

1) Check the internal statuses

2) Set the TX_ENCLR bit of the PCM_TXRX_DIS register to 1 to disable transmission.

3) Set the TX_EN bit of the PCM_TXRX_EN register to 1 to shift to the waiting for transmission state.

(2) Operation in reception block

The DMA request signal is asserted when the state is *waiting for reception* or *receiving* and the reception FIFO buffer is not full. The request signal is deasserted when reception stops or is disabled, or the reception FIFO buffer becomes empty. The request signal is not deasserted each time the AHB is accessed.

2SCLK is needed before it's stocked in a FIFO after the last data is received.

Table 4-4. Reception Pin Operation Sources

Operation Source	Description	State		
		Reception Disabled	Reception Enabled	Reception Stopped
SIOx_DMA_RDMARQ pin	DMA reception request	Fixed to off	With assertion	Fixed to OFF
SIOx_SI pin	Serial data reception	Data not received	Data received	Data not received
SIOx_INT pin	INT interrupt request	**Note 1**	With assertion	**Note 2**
RX_REN flag	Reception data write enable flag	0 (off)	On or off	On/off retained
RX_ORE flag	Reception overrun error detected	0 (off)	On or off	On/off retained
RX_URE flag	Reception underrun error detected	0 (off)	On or off	On/off retained
RX_FRE flag	Reception synchronization error detected	0 (off)	On or off	On/off retained
RXFIFO	Reception FIFO buffer	Read disabled	Read enabled	Read disabled**Note 3**
RX_W_CONT	Reception FIFO buffer write counter	0	Counter value incremented	Counter value retained
RX_R_CONT	Reception FIFO buffer read counter	0	Counter value incremented	Counter value retained
RX_RP_NUM**Note 4**	Word number corresponding to FIFO buffer pointed to by read pointer in the reception FIFO buffer	0	Shows word number	Word number retained
RX_PHASE**Note 4**	Read pointer in reception FIFO buffer points to either phase 1 or 2	0	Shows phase	Phase retained

Notes **1.** The SIOx_INT pin status depends on the transmission interrupt source.

2. Asserted if the reception interrupt source is disabled.

3. Reading from the reception FIFO buffer while reception is disabled might cause a reception underrun error.

4. Fixed to 0 in modes other than modes 5 and 6.

State transitions

1) Reception disabled to waiting for reception

This transition occurs when the RX_EN bit of the PCM_TXRX_EN register is set to 1.

2) Waiting for reception to receiving

This transition occurs when a rise of the SEN signal is detected.

3) Receiving to reception stopped

This transition occurs when a reception error is detected.

4) Any state to reception disabled

This transition occurs when the RX_ENCLR bit of the PCM_TXRX_DIS register is set to 1.

If a reception error occurs, the state shifts to reception stopped, and serial reception stops. In the reception stopped state, all internal statuses are held, so reception can be resumed by using the following procedure:

1) Check the internal statuses

2) Set the RX_ENCLR bit of the PCM_TXRX_DIS register is set to 1 to disable transmission.

3) Set the RX_EN bit of the PCM_TXRX_EN register to 1 to shift to the waiting for reception state.

(3) SIOx_CLK operation

For this macro, SPI and PCM can be switched during use. The PCM SEN/SYNC signal shares a macro pin with the SPI CSO signal, so the macro pin name is CSO.

(a) SIOx_CS0 pin

Table 4-5. SEN Pin Operations

Master or Slave Setting	Reception	Transmission	SIOx_CS0_EN	SIOx_CS0_I	SIOx_CS0_O	Remark
Initial setting	×	×	0	Ignored	Fixed to 0	
Slave	×	×	0	Received	Fixed to 0	
Master	Disabled	Disabled	1	Ignored	Fixed to 0	
	Disabled	Enabled			Depends on the amount accumulated in the FIFO buffer	**Note 1**
	Enabled	Disabled			Normal output	**Note 2**
	Enabled	Enabled			Programming procedure dependence	**Note 3**

Master or slave setting: Bits 4 and 3 of the PCM_FUNC_SEL register

Setting for enabling or disabling reception: Bits 1 and 0 of PCM_TXRX_EN and PCM_TXRX_DIS

Notes **1.** If the master mode, reception disabling, and transmission enabling settings are specified, after SIOx_SEN_O starts (enabling transmission), the system waits until the amount of data accumulated in the transmission FIFO buffer reaches the transmission start threshold (specified using the TX_TIM bit of the PCM_FUNC_SEL register), and then starts transmission.

2. While the MASTER and reception enabling settings are specified, SIOx_CS0_O is output (in the format indicated by modes 0 to 6). This signal must be supplied to enable partner transmission.

3. It'll be the following one of movement by a programming procedure.

When it was established by an order of the transmission permission -> reception permission.

: It's sent out from data accumulation later to a transmission FIFO.

(The same operation as Note1.)

When it was established by an order of the reception permission -> transmission permission.

: Anytime output (The same operation as Note2.)

(b) SIOx_CLK pins

SIOx_CLK_EN, SIOx_CLKI, and SIOx_CLKO are switched according to the master/slave setting (specified for bits 4 and 3 of the PCM_FUNC_SEL register). In the AC97 mode (bit 14 of PCM_FUNC_SEL = 1), these pins are only used in the master mode (when SYNC is output), but the direction of the clock is switched according to the clock direction setting register (bit 15 of PCM_FUNC_SEL). Whenever a clock in the AC97 mode is referred to below, assume that the master is output and the slave is input.

Table 4-6. CLK Pin Operations

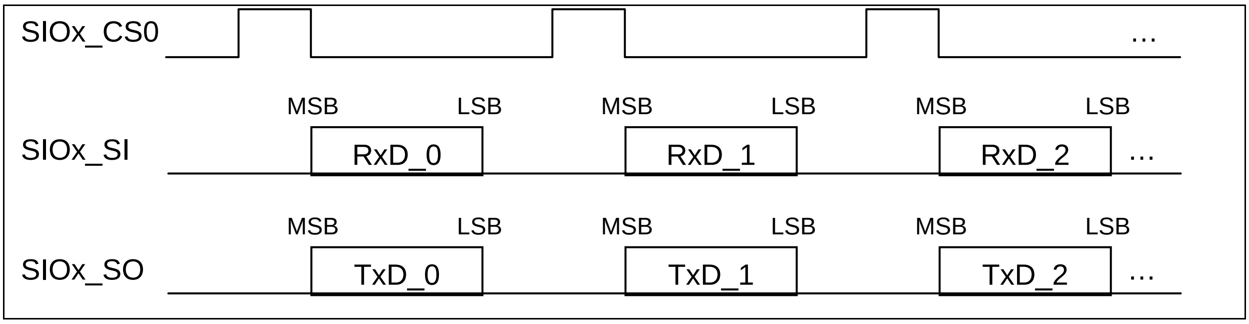
Master/Slave Setting	SIOx_CLK_EN	SIOx_CLKI	SIOx_CLKO
Initial setting	0	Ignored	Fixed to 0
Slave	0	Received	Fixed to 0
Master	1	Ignored	Output[Note]

Note In the master mode, SIOx_SCLK is output to SIOx_CLKO. The SIOx_SCLK clock request signal is asserted during serial transmission or reception, as well as when the setting for enabling transmission or reception is changed, but a mask is applied such that a clock is only output to SIO_CLKO during serial transmission or reception.

(4) FIFO operation

(a) Modes 0 and 1

Figure 4-30. Transmitting Data in Modes 0 and 1



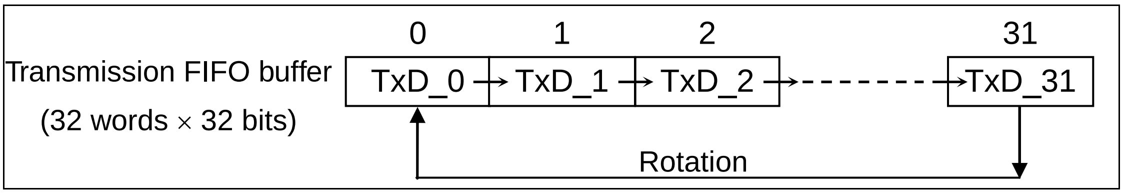
i) Transmission FIFO buffer

1) Transmission data is stored in the transmission FIFO buffer in the order corresponding to the data written by way of the AHB bus or by writing the data to the PCM transmission data register (PCM_TXQ) by using the CPU.

2) One word of data is extracted from the transmission FIFO buffer according to the transmission FIFO buffer counter. Based on the value *n* specified for SOB[4:0] in the PCM data transfer cycle setting register (PCM_CYCLE), serial data is output from the SIOx_SO pin in order from bit *n* (the MSB) to bit 0 (the LSB).

3) After the data is output up to bit 0 (the LSB), the transmission FIFO buffer counter is incremented. The same processing as 2) is repeated when operation starts for the next frame. The range for the counter is from 0 to 31 (1FH), and, if the counter is incremented while at 31, it is reset to 0.

Figure 4-31. Storing Data to Transmission FIFO Buffer



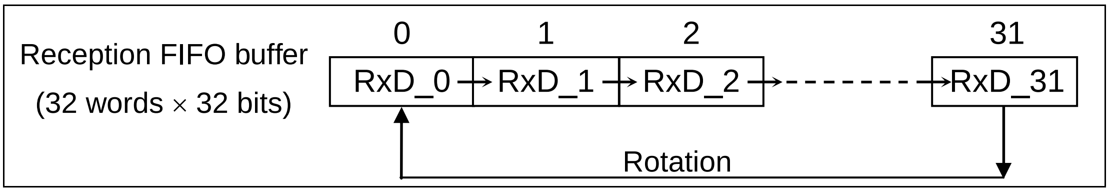
ii) Reception FIFO buffer

1) The reception FIFO buffer receives serial data from the SIOx_SI pin.

2)Based on the value *m* specified for SIB[4:0] in the PCM data transfer cycle setting register (PCM_CYCLE), the received bits are stored in the reception FIFO buffer in word units in order from bit (the MSB) to bit 0 (the LSB). *0 is used as padding in bits higher than bit m.*

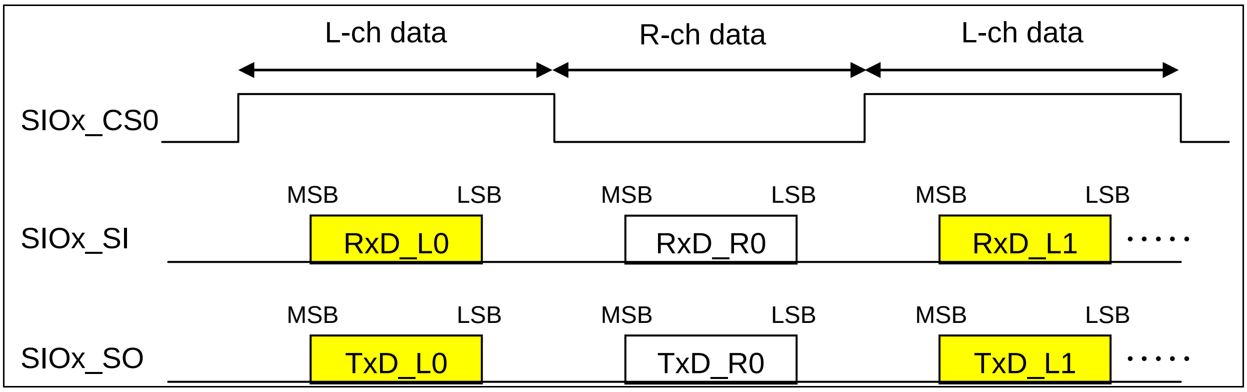
3)After the data is stored up to bit 0 (the LSB), the reception FIFO buffer counter is incremented. The same processing as 2) is repeated when the reception FIFO buffer receives the next frame. The range for the counter is from 0 to 31 (1FH), and, if the counter is incremented while at 31, it is reset to 0.

Figure 4-32. Storing Data to Reception FIFO Buffer



(b) Modes 2, 3, and 4

Figure 4-33. Transmitting/Receiving Data in Modes 2, 3, and 4



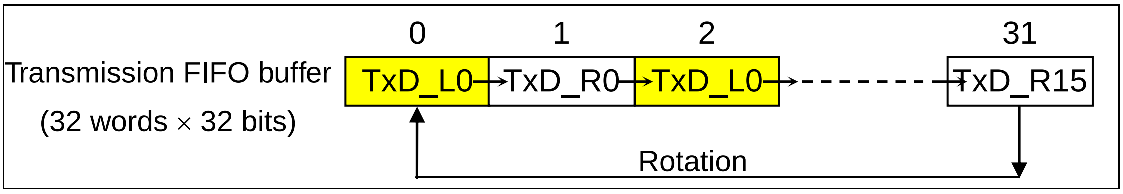
In modes 2 to 4, L-ch data and R-ch data are alternately stored in the reception and transmission FIFO buffers because the L-ch data and the R-ch data are alternately transmitted and received.

Therefore, the L-ch data and the R-ch data need to be alternately written for the transmission data. The received data is read from the L-ch data first, then the R-ch data.

For any other than the above, the FIFO operation is the same as that in modes 0 and 1.

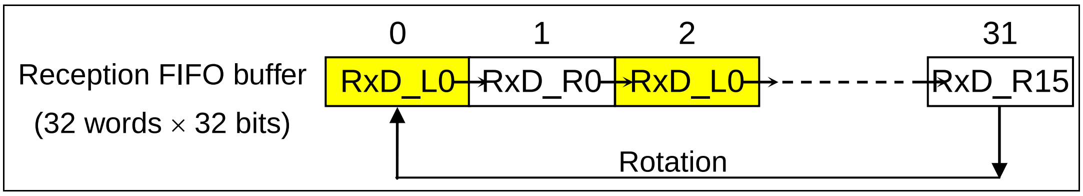
i) Transmission FIFO buffer

Figure 4-34. Storing Data to Transmission FIFO buffer



ii) Reception FIFO buffer

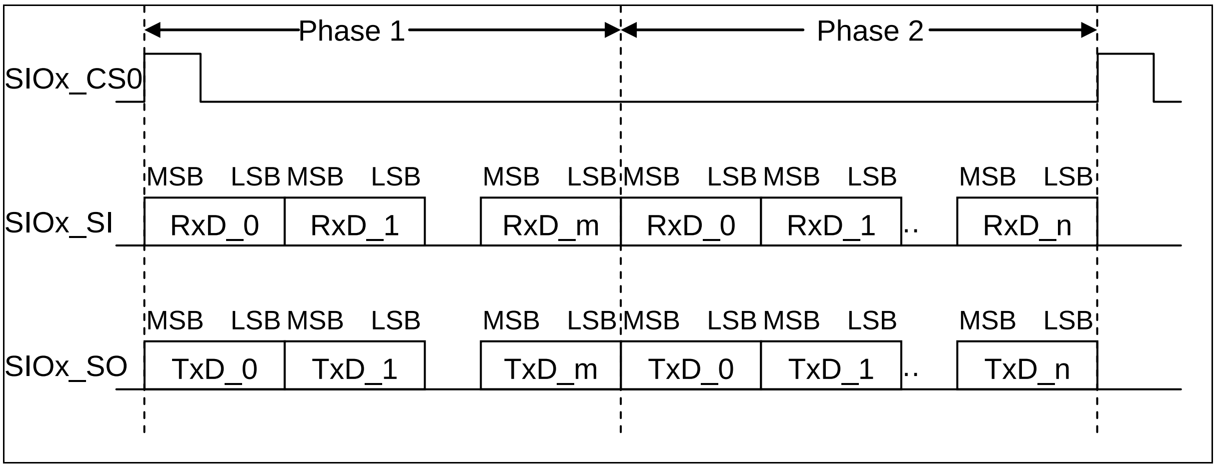
Figure 4-35. Storing Data to Reception FIFO Buffer



For details about storing data to FIFO buffers with data padding, see **4.3.2 (5) Data padding**.

(c) Modes 5 and 6

Figure 4-36. Transmitting Data in Modes 5 and 6



i) Transmission FIFO buffer

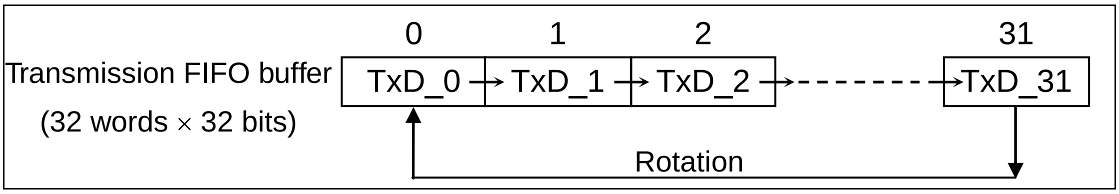
1)Transmission data is stored in the transmission FIFO buffer in the order corresponding to the data written by way of the AHB bus or by writing the data to the PCM transmission data register (PCM_TXQ) using the CPU.

2)Based on the transmission FIFO buffer counter, 1 word of data is extracted from the transmission FIFO buffer.

Based on the values *n* and *m* specified for SOB[4:0] and SOB2[4:0] in the PCM data transfer cycle setting registers (PCM_CYCLE and PCM_CYCLE2), serial data is output as the phase 1 transmission data corresponding to the words specified for CYC_VAL[7:0] in the PCM data transfer cycle setting register in order from bit *m* (the MSB) to bit 0 (the LSB), and then serial data is output as the phase 2 transmission data corresponding to the words specified for CYC_VAL2[7:0] in the PCM data transfer cycle setting register in order from bit *n* (the MSB) to bit 0 (the LSB).

3)When one word of data is output up to bit 0 (the LSB), the transmission FIFO buffer counter is incremented. Next, the data corresponding to the next word is output up to bit 0 (the LSB) and the transmission FIFO buffer counter is incremented. The same processing is repeated after this. The range for the counter is from 0 to 31 (1FH), and, if the counter is incremented while at 31, it is reset to 0.

Figure 4-37. Storing Data to Transmission FIFO Buffer



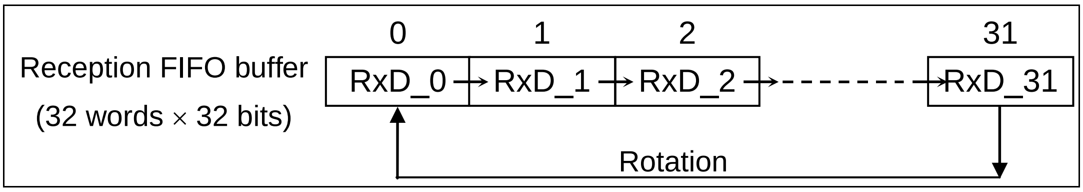
ii) Reception FIFO buffer

1) The reception FIFO buffer receives serial data from the SIOx_SI pin.

2) Based on the values *m* and *n* specified for SIB[4:0] and SOB2[4:0] in the PCM data transfer cycle setting registers (PCM_CYCLE and PCM_CYCLE2), the phase 1 received data corresponding to the words specified for CYC_VAL[7:0] in the PCM data transfer cycle setting register is stored in the reception FIFO buffer in word units in order from bit *m* (the MSB) to bit 0 (the LSB), and then the phase 2 received data corresponding to the words specified for CYC_VAL2[7:0] in the PCM data transfer cycle setting register is stored in the reception FIFO buffer in word units in order from bit *n* (the MSB) to bit 0 (the LSB). *0 is used as padding in bits higher than m and n if the received data is smaller than 32 bits.*

3) When the data corresponding to one word is stored up to bit 0 (the LSB), the reception FIFO buffer counter is incremented. Then the data corresponding to the next word is received up to bit 0 (the LSB) and the reception FIFO buffer counter is incremented. The same processing is repeated after this. The range for the counter is from 0 to 31 (1FH), and, if the counter is incremented while at 31, it is reset to 0.

Figure 4-38. Storing Data to Reception FIFO Buffer



(5) Data padding

Data padding can be performed when the data bit length is 8 or 16 bits. Data padding refers to an operation by which four 8-bit units or two 16-bit units are padded into one word (32 bits) because the FIFO buffer bit width is 32 bits.

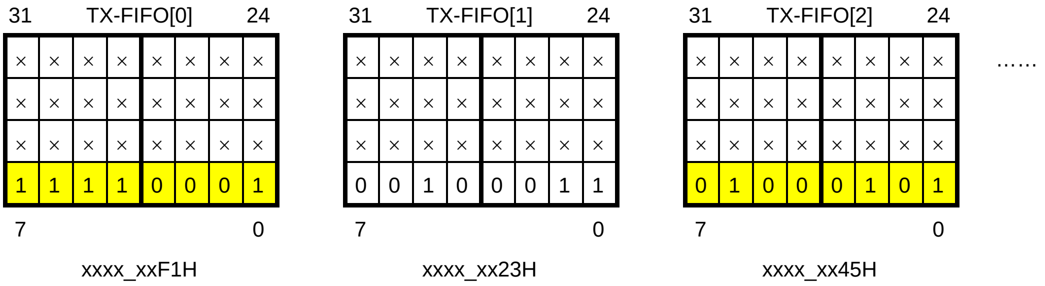
Data padding can be enabled or disabled by using the PCM_CYCLE register.

Data padding is prohibited in mode 5 or 6.

(1) Data padding on the transmission side

(a) Without data padding

Example 1	Transmission data bit length = 8 bits (SOB = 07H)	Bits 20 to 16 of PCM_CYCLE
	Data padding on the transmission side is disabled	Bit 23 of PCM_CYCLE = 0
	xxxx_xxF1H → Write data (PCM_TXQ)	
	xxxx_xx23H → Write data (PCM_TXQ)	
	xxxx_xx45H → Write data (PCM_TXQ)	
	xxxx_xx67H → Write data (PCM_TXQ)	
	...	



TX-FIFO[0] (31–24 … 7–0), xxxx_xxF1H

×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
1	1	1	1	0	0	0	1

TX-FIFO[1] (31–24 … 7–0), xxxx_xx23H

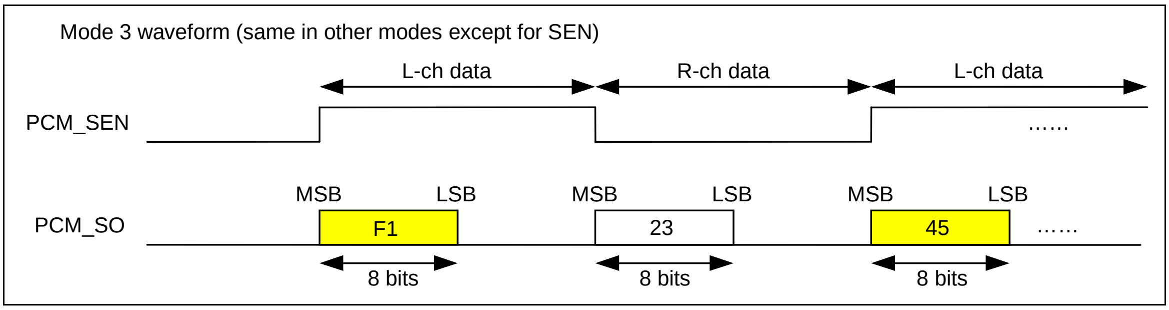
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
0	0	1	0	0	0	1	1

TX-FIFO[2] (31–24 … 7–0), xxxx_xx45H

×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
0	1	0	0	0	1	0	1

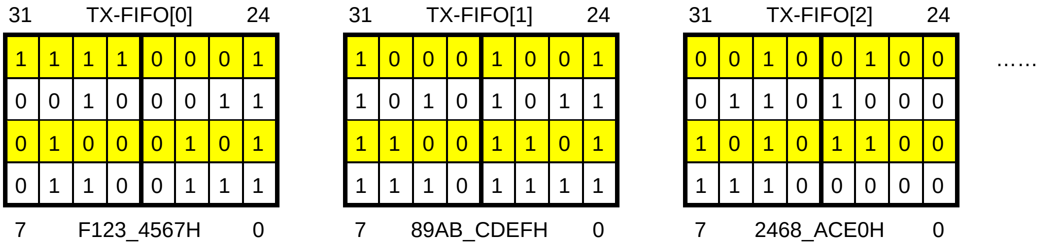
Data longer than the transmission data bit length is ignored (×: Don't care).

Figure 4-39. Transmission Waveform in Mode 3



(b) With data padding

Example 2 Transmission data bit length = 8 bits (SOB = 07H) Bits 20 to 16 of PCM_CYCLE
Data padding on the transmission side is enabled Bit 23 of PCM_CYCLE = 1
F123_4567H → Write data (PCM_TXQ)
89AB_CDEFH → Write data (PCM_TXQ)
2468_ACE0H → Write data (PCM_TXQ)



31 TX-FIFO[0] 24

1	1	1	1	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1

7 F123_4567H 0

31 TX-FIFO[1] 24

1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

7 89AB_CDEFH 0

31 TX-FIFO[2] 24

0	0	1	0	0	1	0	0
0	1	1	0	1	0	0	0
1	0	1	0	1	1	0	0
1	1	1	0	0	0	0	0

7 2468_ACE0H 0

……

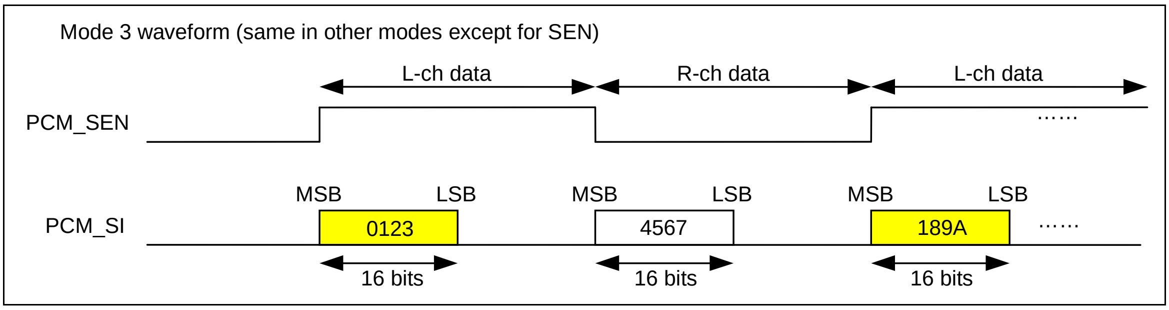
Remark The signal waveform is the same as in Figure 4-39 Transmission Waveform in Mode 3.

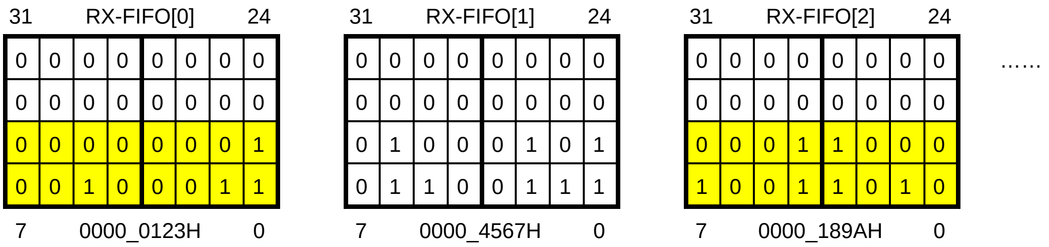
(2) Data padding on the reception side

(a) Without data padding

Example 3 Reception data bit length = 16 bits (SIB = 0FH) Bits 12 to 8 of PCM_CYCLE
Data padding on the reception side is disabled Bit 15 of PCM_CYCLE = 0

Figure 4-40. Reception Waveform in Mode 3





31 RX-FIFO[0] 24

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1

7 0000_0123H 0

31 RX-FIFO[1] 24

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1

7 0000_4567H 0

31 RX-FIFO[2] 24

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
1	0	0	1	1	0	1	0

7 0000_189AH 0

……

Data exceeding the reception data bit length is padded with zeros.

0000_0123H (RXFIFO[0]) → Read data (PCM_RXQ)
0000_4567H (RXFIFO[1]) → Read data (PCM_RXQ)
0000_189AH (RXFIFO[2]) → Read data (PCM_RXQ)
...

(b) With data padding

Example 4 Reception data bit length = 16 bits (SIB = 0FH) Bits 12 to 8 of PCM_CYCLE
Data padding on the reception side is enabled Bit 15 of PCM_CYCLE = 1

31 RX-FIFO[0] 24

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1

7 0123_4567H 0

31 RX-FIFO[1] 24

0	0	0	1	1	0	0	0
1	0	0	1	1	0	1	0
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	0

7 189A_BCDEH 0

31 RX-FIFO[2] 24

0	0	1	0	1	1	1	1
0	0	1	0	0	1	0	0
0	1	1	0	1	0	0	0
1	0	1	0	1	1	0	0

7 2F24_68ACH 0

……

0123_4567H (RXFIFO[0]) → Read data (PCM_RXQ)
189A_BCDEH (RXFIFO[1]) → Read data (PCM_RXQ)
2F24_68ACH (RXFIFO[2]) → Read data (PCM_RXQ)
...

Caution If 32 bits of the data received from the PCM_SI pin does not amount to 32 bit in the reception FIFO buffer, the data is not considered valid and the last data cannot be read.

Remark The signal waveform is the same as in **Figure 4-40 Reception Waveform in Mode 3**.

(6) Cautions on using the PCM macro

- Caution on specifying the slave mode

In the slave mode, the SIOx_CS0 and SIOx_CLK signals are supplied from the opposite side of communication. If transmission or reception is enabled while these signals are not stable, a synchronous error might occur or data might be corrupted.

- Caution on switching master and slave mode

Perform the procedure below to switch the master and slave mode. Because this procedure switches the direction of the SIOx_CS0 and SIOx_CLK pins, thoroughly exercise care so that no bus conflicts occur on the board.

1) Disabling transmission and reception: Set bits 1 and 0 of the PCM_TXRX_DIS register to 1.
2) Specifying the master or slave mode: Clear bits 4 and 3 of the PCM_FUNC_SEL register to 0 (to initialize the status)
3) Specifying the master or slave mode: Respectively set bits 4 and 3 of the PCM_FUNC_SEL register to 0 and 1 (for the master mode) or 1 and 0 (for the slave mode).
4) Enable transmission and reception before starting communication by using the PCM_TXRX_EN register.

- Caution on changing communication setup register settings

Be sure to disable transmission and reception before changing the settings of a communication setup register. Otherwise, the operation is not guaranteed.

Communication setup registers:
Operating mode setting register (PCM_FUNC_SEL)
Data transfer cycle setting register (PCM_CYCLE)
Data transfer cycle setting register 2 (PCM_CYCLE2)

- Cautions on reception

While reception is enabled, even if no data is input to the SIOx_SI pin (0 continues), the reception FIFO buffer is filled with zeros. At this time, the reception FIFO buffer counter is incremented, so, if the reception FIFO buffer is not read, an overrun error occurs.

(7) Communication where frame length equals bit length

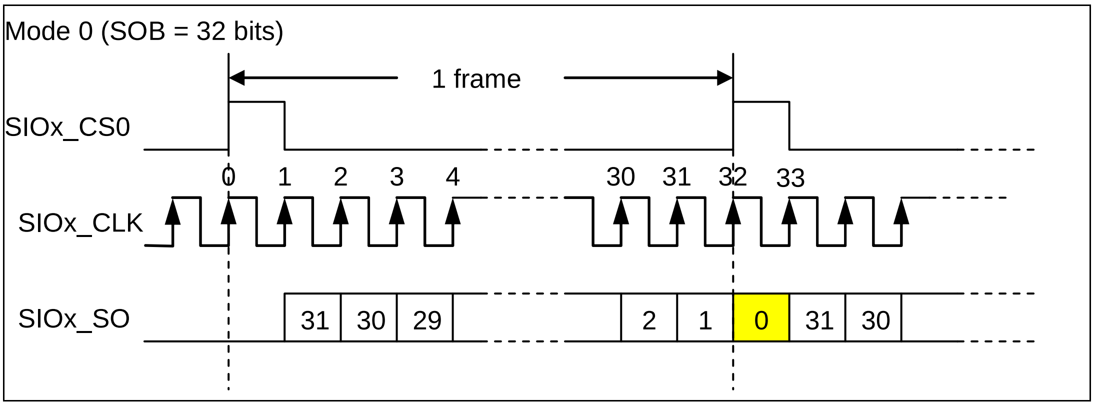
For SSP (in mode 0 or 1), if the frame length and bit length are the same, data bits are aligned as shown below.

- PCM_CYCLE register

1) CYC_VAL[5:0] = 1FH (The number of clock cycles per frame is 32.)

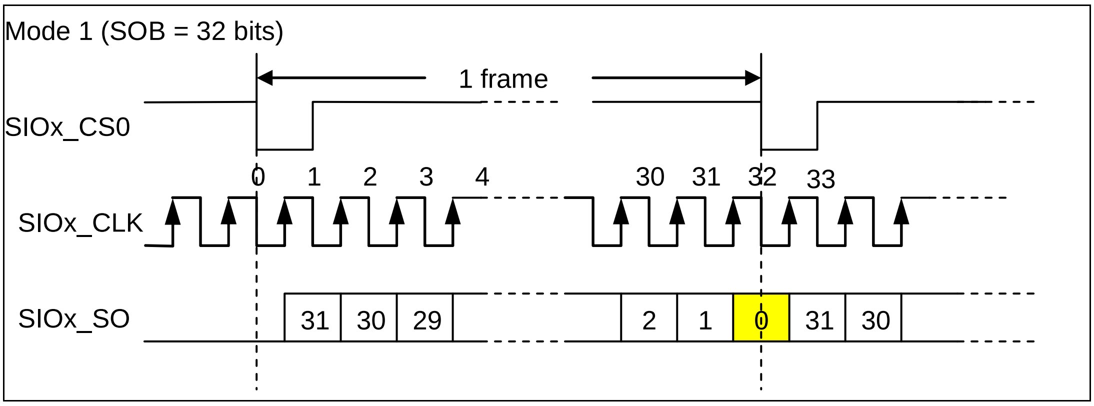
2) SOB[4:0] = 1FH (The length of SIOx_SO data bits per frame is 32.)

Figure 4-41. When Frame Length = Bit Length Is Specified for SSP (in Mode 0)



Mode 0: Because data is output one clock cycle after the frame signal changes, the last bit is sent at the beginning of the next frame.

Figure 4-42. When Frame Length = Bit Length Is Specified for SSP (in Mode 1)



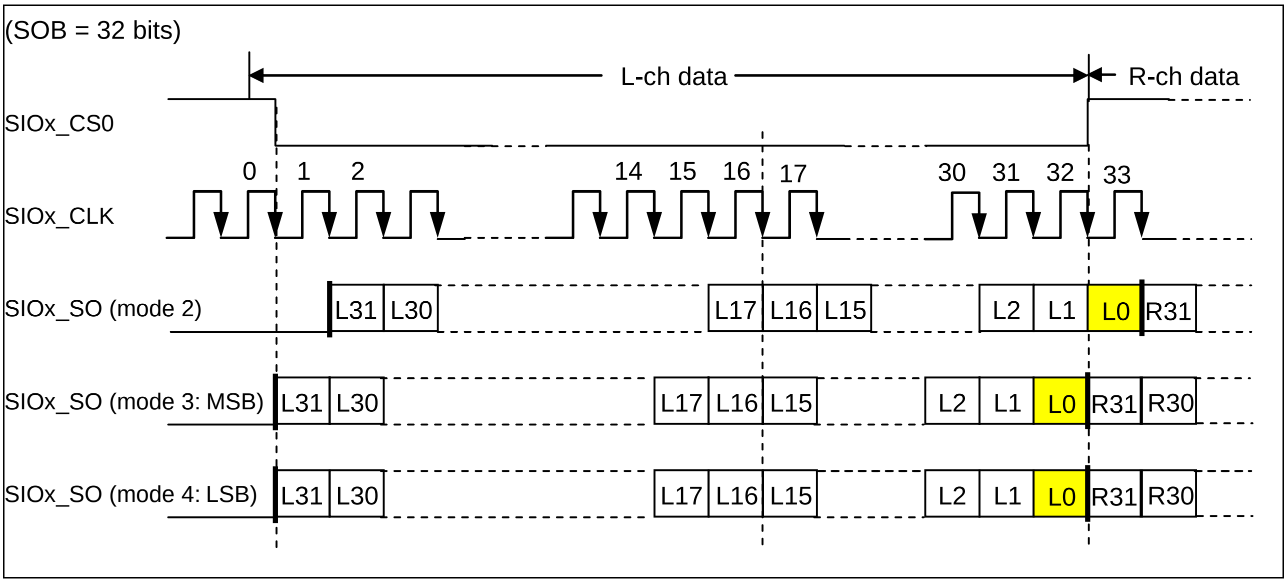
For I^{2}S (in mode 2, 3, or 4), if the frame length and bit length are the same, data bits are aligned as shown below.

- PCM_CYCLE register

1) CYC_VAL[5:0] = 1FH (The number of clock cycles per frame is 32.)

2) SOB[4:0] = 1FH (The length of SIOx_SO data bits per frame is 32.)

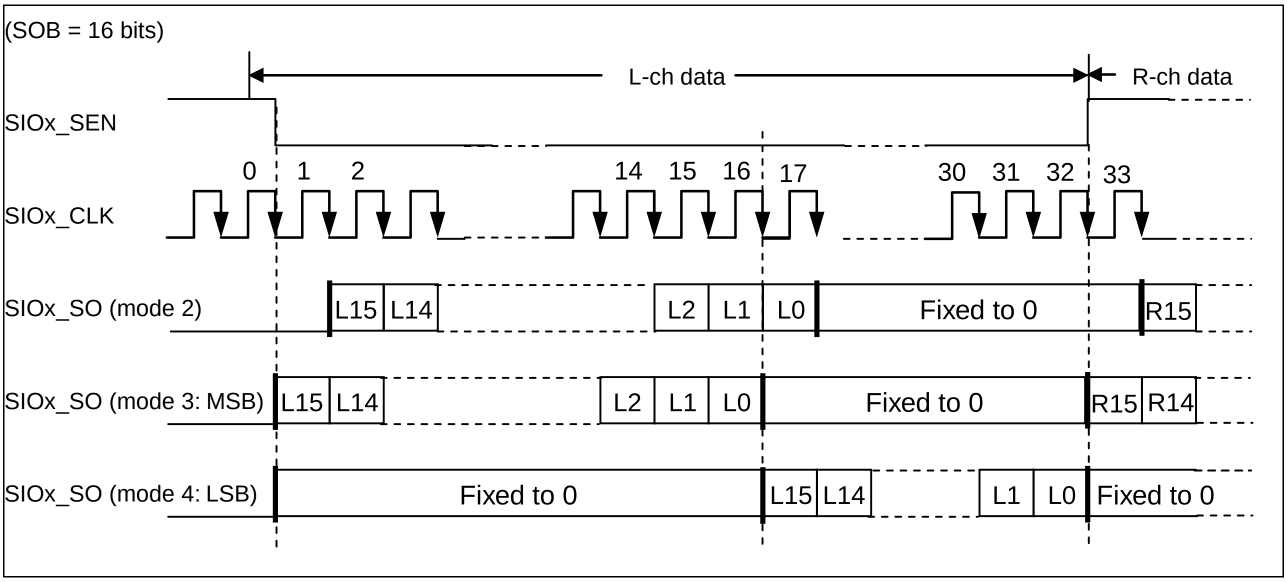
Figure 4-43. When Frame Length = Bit Length Is Specified for I^{2}S (in Mode 2, 3, or 4)



Mode 2: Because data is output one clock cycle after the frame signal changes, the last bit is sent at the beginning of the next frame. The bit alignment result in mode 3 (MSB-aligned) and mode 4 (LSB-aligned) is the same.

Remark For normal settings (frame length = 32 clock cycles, bit length = 16), data bits are aligned as shown below.

Figure 4-44. When Frame Length = 32 and Bit Length = 16 Are Specified for I^{2}S (in Mode 2, 3, or 4)



For a multi-channel mode (mode 5 or 6), if the frame length and bit length are the same and the single phase is specified, data bits are aligned as shown below.

- PCM_CYCLE register

1) CYC_VAL[7:0] = 01H (The number of words at phase 1 is 1)
2) SOB[4:0] = 1FH (The length of SIOx_SO data bits at phase 1 is 32
3) CYC_VAL2[7:0] = 00H (The number of words at phase 2 is 0)

If CYC_VAL[7:0] = 00H and CYC_VAL2[7:0] = 01H, the setting of SOB2[4:0] becomes valid.

Figure 4-45. When Frame Length = Bit Length and Single Phase Are Specified for a Multi Channel Mode (Mode 5)

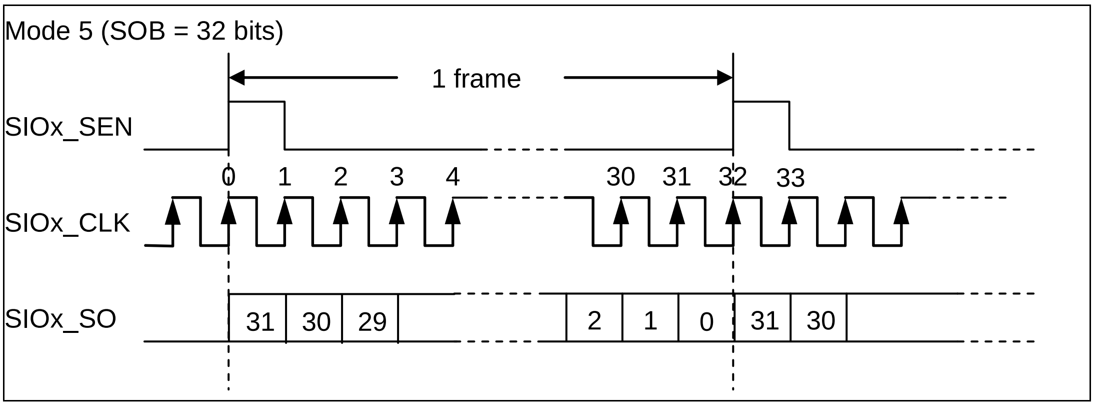
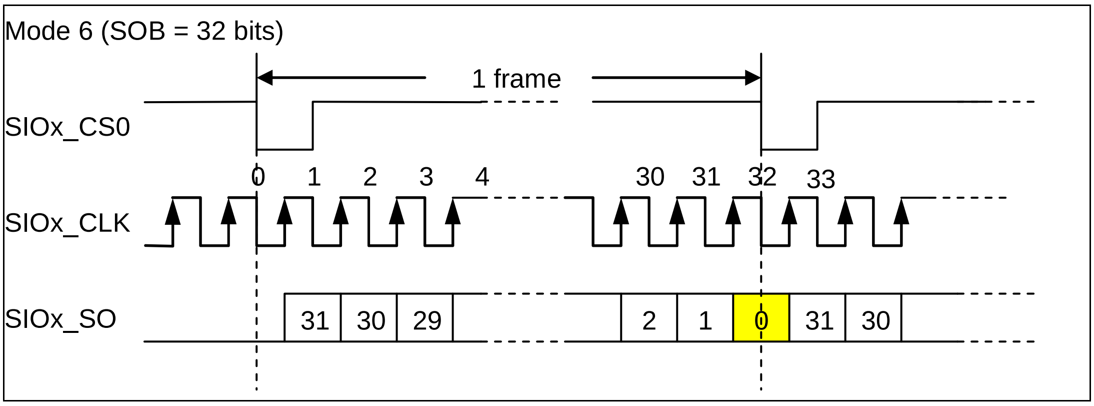


Figure 4-46. When Frame Length = Bit Length and Single Phase Are Specified for a Multi Channel Mode (Mode 6)



4.4 DMA Request Timing

The table below shows the conditions for issuing a DMA request in each operating mode. The DMA request signal is asserted when the conditions in the table are satisfied and then deasserted when they are no longer satisfied.

Operating Mode	Direction	Burst Transfer	DMA Request Signal Assertion Condition
SPI	TX	No	The DMA mode is specified, transmission is enabled, and the transmission FIFO buffer is not full.
		Yes	The DMA mode is specified, transmission is enabled, and the transmission FIFO buffer has at least 8 words of open space.
	RX	No	The DMA mode is specified, reception is enabled, and the reception FIFO buffer is not empty.
Audio	TX	No	The DMA mode is specified, transmission is enabled, transmission is not stopped, and the transmission FIFO buffer is not full.
		Yes	The DMA mode is specified, transmission is enabled, transmission is not stopped, and the transmission FIFO buffer has at least 8 words of open space.
	RX	No	The DMA mode is specified, reception is enabled, reception is not stopped, and the reception FIFO buffer is not empty.

Remark If transmission or reception is stopped, the DMA request signal is not asserted, but there are no restrictions on accessing the FIFO buffers.

The timing at which DMARQ is deasserted according to whether FIFO buffers are full or empty is shown below.

Figure 4-47. Timing at Which the Reception DMARQ Signal Is Deasserted When the RX_FIFO Buffer Is Empty

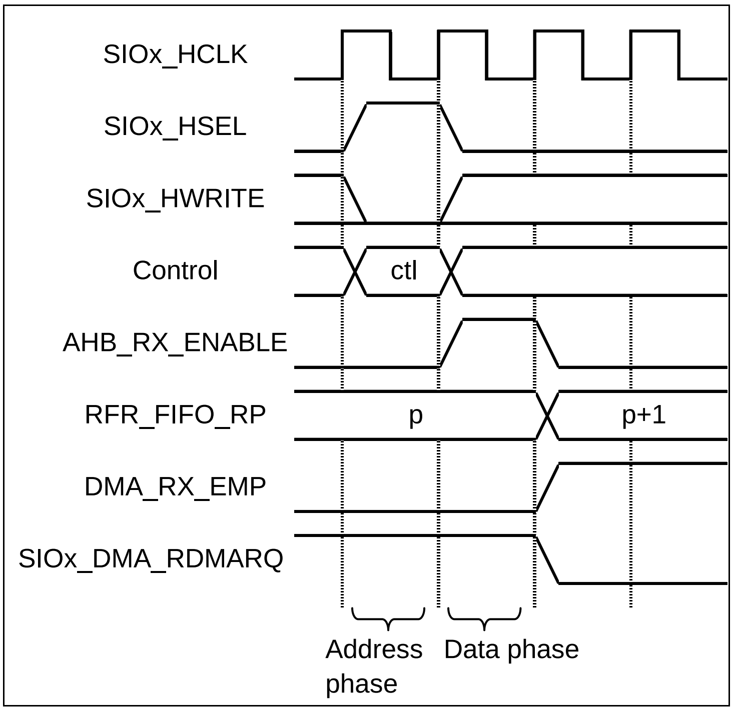
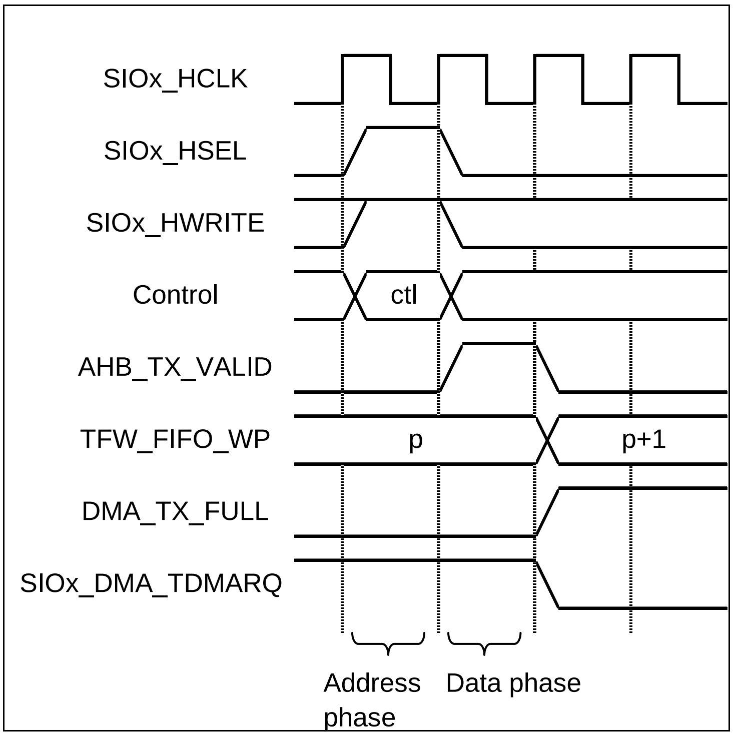


Figure 4-48. Timing at Which the Transmission DMARQ Signal Is Deasserted When the TX_FIFO Buffer Is Full



Remark The above figures assume single-burst transfers. For 8-burst transfers, DMARQ is deasserted when there are 8 or fewer open spaces in the transmission FIFO buffer, not when the transmission FIFO buffer becomes full.

5. Usage

5.1 SPI Mode

Before starting operations, perform the following procedure:

(1) Set up the operating mode (common for SPI and audio modes)

Bit 0 of SIO_SWITCH_EN = 1: Enable changing the SIO_MODE_SWITCH register settings.

Bit 4 of SIO_MODE_SWITCH: Specify the type of AHB burst transfer. (0: Single-burst transfer, 1: 8-burst transfer)

Bits 3 to 0 of SIO_MODE_SWITCH = 0001: SPI mode

Bit 0 of SIO_SWITCH_EN = 0: Disable changing the SIO_MODE_SWITCH register settings.

(2) Specify settings for communication setup registers

Specify settings for the following communication setup registers according to the mode and parameters used:

SPI mode register (SPI_MODE)

SPI polarity register (SPI_POL)

SPI CS fixed value setting register (SPI_TIECS)

(3) Enable interrupts

Enable the interrupts to be used.

Write 1 to the corresponding bits in the SPI_ENSET register.

(4) Start transmission/reception

Bits 3 and 2 of SPI_CONTROL: 01: reception, 10: transmission, 11: reception/transmission

Bit 0 of SPI_CONTROL = 1: Start a transfer. (Valid only in master mode)

○ Resuming a transfer after an error

To resume a transfer after an error, perform the following procedure:

(1) Execute a software reset

Set bit 8 of SPI_CONTROL to 1.

(2) Cancel the software reset

Clear bit 8 of SPI_CONTROL to 0.

Caution Perform this step 5 or more HCLK or PCLK cycles after the software reset is executed.

(3) Start transmission/reception

Bits 3 and 2 of SPI_CONTROL: 01: reception, 10: transmission, 11: reception/transmission

Bit 0 of SPI_CONTROL = 1: Start transfer. (Valid only in master mode)

Remark It is not necessary to set up the communication setup registers or disable interrupts again because these settings are retained after the software reset.

5.2 Audio Mode

Before starting operations, perform the following procedure:

(1) Set up the operating mode (common for SPI and audio modes)

Bit 0 of SIO_SWITCH_EN = 1: Enable changing the SIO_MODE_SWITCH register settings.

Bit 4 of SIO_MODE_SWITCH: Specify the type of AHB burst transfer. (0: Single-burst transfer, 1: 8-burst transfer)

Bits 3 to 0 of SIO_MODE_SWITCH = 0010: Audio mode

Bit 0 of SIO_SWITCH_EN = 0: Disable changing the SIO_MODE_SWITCH register settings.

(2) Specify settings for communication setup registers

Specify settings for the following communication setup registers according to the mode and parameters used:

Operating mode setting register (PCM_FUNC_SEL)

Data transfer cycle setting register (PCM_CYCLE)

Data transfer cycle setting register 2 (PCM_CYCLE2)

(3) Enable interrupts

Enable the interrupts to be used.

(4) All "0" is written in PCM_TXQ

It's necessary for transmission permission (The implementation of the professional order is unnecessary for only reception in case of permission.)

These all "0" data is preparations before transmission permission, and is broken (it isn't sent actually.)

(5)Enable transmission and/or reception (PCM_TXRX_EN)

Enable transmission and/or reception.

The occasion and the thing which pays attention to the difference in the operation by a programming procedure which permit send and receive both of them (table 4-5 referring of 4.3.2 (3)).

(6) All "0" for 1 frame is written in PCM_TXQ

It's necessary for transmission permission (The implementation of the professional order is unnecessary for only reception in case of permission.)

These all "0" data is sent to 1 frame at the beginning of the send data certainly.

(7) Data transmission starting (in a transmission DMA start or a PCM_TXQ, send data writing in). But only transmission permission.

Example of steps (2) and (3)

2-1) PCM_FUNC_SEL register (00H): Specify the operating mode.

MODE_SEL[2:0] = 011b: Mode 3

M_S[1:0] = 10b: Slave mode

TX_TIM[1:0] = 000b: Transmission starts when 30 words of data is stored in the transmission FIFO buffer.

2-2) PCM_CYCLE register (0CH): Specify the number of clock cycles and the length of data bits to transmit or receive.

CYC_VAL[5:0] = 01_1111b: The number of clock cycles per frame is 32.

SIB[4:0] = 1_0111b: The length of PCM_SI data is 24 bits.

RX_PD = 0b: Received data is not padded.

SOB[4:0] = 1_0111b: The length of PCM_SO data is 24 bits.

TX_PD = 0b: Transmitted data is not padded.

3) PCM_ENSET register (18H): Enable interrupts.

RX_REN_EN = 0
RX_ORE_EN = 1
RX_URE_EN = 1
RX_FRE_EN = 1
TX_STRENB = 0
TX_ORE_EN = 1
TX_URE_EN = 1
TX_FRE_EN = 1

The resumption procedure when an error occurred

When a send error occurred, it transfer in transmission abeyance and suspend serial transmission automatically. (The inner status is maintained by transmission abeyance.)

It resumes in following procedure.

1. Confirmation of the inner status
2. The transmission permission clear bit is set as '1' (PCM_TXRX_ENCLR [0] = '1') and it's made the transmission prohibited state.
3. Transmission permission bit is set as '1' (PCM_TXRX_EN [0] = '1') and it's made the transmission waiting state.

When a receive error occurred, it transfer in reception abeyance and suspend serial reception automatically. (The inner status is maintained by reception abeyance.)

It resumes in following procedure.

1. Confirmation of the inner status
2. The reception permission clear bit is set as '1' (PCM_TXRX_ENCLR [1] = '1') and it's made the reception prohibited state.
3. Reception permission bit is set as '1' (PCM_TXRX_EN [1] = '1') and it's made the reception waiting state.

REVISION HISTORY	EMMA Mobile EV2 User’s Manual: Unified Serial Interface

Rev.	**Date**	**Description**	
		Page	**Summary**
1.00	Feb 1, 2010	—	1st revision release
2.00	Jun 7, 2010	—	Incremental update from comments to the 1.0.
3.00	Sep 30, 2010	—	Incremental update from comments to the 2.0. (A change part from the old revision is “★” marked in the page left end.)
4.00	May 31, 2011	—	Incremental update from comments to the 3.0.
5.00	Sep 30, 2011	—	Incremental update from comments to the 4.0.
		3	Figure 1-2 added.
		4	Figure 1-3 added.
6.00	Dec 21, 2011	—	Incremental update from comments to the 5.0.
		4	Chapter 1.2 corrected. (SIO0, SIO2 CS number and Interrupt number)

EMMA Mobile EV2 User’s Manual: Unified Serial Interface

Publication Date: Rev.1.00 Feb 1, 2010
Rev.6.00 Dec 21, 2011

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2011 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0

Unified Serial Interface

EMMA Mobile EV2

Renesas Electronics Corporation