32-bit RISC Microcontroller

TXZ+ Family

Reference Manual
Serial Peripheral Interface
(TSPI-E)

Revision 1.3

2023-03

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

© 2020-2023
Toshiba Electronic Devices & Storage Corporation

Contents

List of Figures

List of Tables

Preface

Related Document

Document Name
Clock Control and Operation Mode
Exception
Input/Output Ports
Product Information

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1], [XYZ2], [XYZ3] → [XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C . . .
  Example: ***[ADACR0], [ADBCR0], [ADCCR0] → [ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 . . .
  Example: ***[T32A0RUNA], [T32A1RUNA], [T32A2RUNA] → [T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviation

Some of abbreviations used in this document are as follows:

ACK	Acknowledgement
DMA	Direct Memory Access
FIFO	First-In First-Out
LSB	Least Significant Bit
MSB	Most Significant Bit
SIO	Serial Input/Output
TSPI	Serial Peripheral Interface

1. Outline

TSPI (serial peripheral interface) has a total of eight communication modes by allowing for the following choices; SPI mode/SIO mode, clock master/clock slave, and frame mode/sector mode. It operates as a one-channel circuit for transmit and receive. The sector mode combines a maximum of four types of data length and allows for transmit and receive in one frame, improving the processing efficiency for each data type. The list of the functional outline is shown in Table 1.1 and the functional outline of each communication mode is shown in Table 1.2 to Table 1.9.

Table 1.1 Functional outline (frame mode, sector mode)

Function classification			Frame mode specification	Sector mode specification
Transfer clock	Prescaler		(Master) The input clock can be divided by 1, 2, 4 up to 512.	
	Baud rate generator		(Master) The prescaler output can be divided by 1, 2,3 up to 16.	
	Transfer clock generation		(Master) The baud rate generator output can be divided by 2.	
Transmit and Receive Control	Communication modes		SPI/SIO mode	
			Master/Slave	
			Frame mode	Sector mode
	Communication operation mode		Transmit and receive(Full-duplex communication)/Transmit/Receive	
	Transfer mode		Burst transfer/Continuous transfer	Continuous transfer
Data format	Sector count/Sector length		-	2 to 4 sectors / 1 to 32 bits
	Frame length		8 to 32 bits	8 to 128 bits (all sectors)
	Number of FIFO stages		16 bits × 8 stages / 32 bits x 4 stages	32 bits × 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.	
	Direction of data transfer		Selection of LSB first / MSB first is provided.	
Ganged control	Interruption	Transmission	Complete/FIFO	
		Reception	Complete/FIFO	
		Error	(Master) Vertical parity / Start Communication trigger	
			(Slave) Vertical parity / Underrun / Overrun	
	Status flag	Common	TSPI can be set up	
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty	
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full	
		Error	(Master) Vertical parity / Start Communication trigger	
			(Slave) Vertical parity / Underrun / Overrun	
	DMA request	Transmission	Single DMA request / Burst DMA request	
		Reception	Single DMA request / Burst DMA request	
	Trigger control (Note 1)	Trigger input	(Master) Start Communication trigger → Start communication	
		Trigger output	Transmission complete → Transmission Complete trigger, Reception complete → Reception Completed trigger	
Special control (Note 2)	SCK: Polarity selection		High, Low: (Level during the idle period)	
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z	
			(Slave) When an underrun error occurs: High, Low	
	TXD timing		(Slave) Last data hold time for the SIO mode: 2/fsys to 128/fsys	
	RXD timing		(Master) 1st edge / 2nd edge (timing for data sampling)	
			(Slave) 1st edge / 2nd edge (timing for data sampling)	(Slave) 2nd edge (timing for data sampling)
	CS polarity selection		High, Low: (Level of the asserted period)	
	CS timing		(Master) Cycle after CS assertion: 1/f_{SCK} to 16/f_{SCK}	
			(Master) Cycle before CS deassertion: 1/f_{SCK} to 16/f_{SCK}	
	Frame interval period		(Master) Burst transfer: 0/f_{SCK} to 15/f_{SCK}	-
	Idle period		(Master) Continuous transfer: 1/f_{SCK} to 15/f_{SCK}	
	Software reset		Reset by software	

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: fsys: System clock frequency f_{SCK}: Transfer clock frequency

Table 1.2 Function outline (SPI mode, Master, Frame)

Function classification			A Functional Description or the range
Transfer clock	Prescaler		The input clock can be divided by 1,2,4 up to 512
	Baud rate generator		The prescaler output can be divided by 1,2,3 up to 16
	Transfer clock generation		The baud rate generator output can be divided by 2
Transmit and Receive Control	Communication mode		SPI mode, master, frame mode
	Communication Operation mode		Transmit and receive(Full-duplex communication) / transmit / receive
	Transfer mode		Burst transfer/Continuous transfer
Data Format	Frame length		8 to 32 bits (adjustable in bit units))
	FIFO stage		16 bits × 8 stages / 32 bits × 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	The direction of data transfer		Selection of the LSB first / MSB first is possible.
Ganged Control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Start Communication trigger
	Status flag	Common	TSPI can be set up
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Start Communication trigger
	DMA request	Transmission	Single DMA request / burst DMA request
		Reception	Single DMA request / burst DMA request
	Trigger control (note1)	Trigger input	(Master) Start Communication trigger → Start communication
		Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception complete trigger
Special control (note2)	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
	RXD timing		1st edge / 2nd edge (timing for data sampling)
	CS polarity selection		High, Low: (Level of the asserted period)
	CS timing		Cycle after CS assertion: $1/f_{SCK}$ to $16/f_{SCK}$
			Cycle before CS deassertion: $1/f_{SCK}$ to $16/f_{SCK}$
	Frame interval period		Burst transfer: $0/f_{SCK}$ to $15/f_{SCK}$
	Idle period		Continuous transfer: $1/f_{SCK}$ to $15/f_{SCK}$
	Software reset		Reset by software.

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: f_{SCK}: Transfer Clock frequency

Table 1.3 Function outline (SPI mode, Slave, Frame)

Function classification			A Functional Description or the range
Transmit and Receive Control	Communication modes		SPI mode, slave mode, frame mode
	Communication Operation mode		Transmit and receive(Full-duplex communication), transmit, receive
	Transfer mode		Burst transfer/Continuous transfer
Data Format	Frame length		8 to 32 bits (adjustable in bit units))
	FIFO stage		16 bits × 8 stages / 32 bits × 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	The direction of data transfer		Selection of the LSB first / MSB first is possible.
Ganged Control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Underrun / Overrun
	Status flag	Common	TSPI Modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Underrun / Overrun
	DMA request	Transmit	Single DMA request / burst DMA request
		Receive	Single DMA request / burst DMA request
	Trigger control (Note)	Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception complete trigger
Special Control	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
			When an underrun error occurs: High, Low
	RXD timing		1st edge / 2nd edge (timing for data sampling)
	CS polarity selection		High, Low: (Level of the asserted period)
	Software reset		Reset by software

Note: For the synced peripheral, refer to reference manual "Product Information".

Table 1.4　Function outline (SIO mode, Master, Frame)

Function classification			A Functional Description or the range
Transfer clock	Prescaler		The input clock can be divided by 1,2,4 up to 512
	Baud rate generator		The prescaler output can be divided by 1,2,3 up to 16
	Transfer clock generation		The baud rate generator output can be divided by 2
Transmit and Receive Control	Communication modes		SIO mode, master mode, frame mode
	Communication Operation mode		Transmit and receive(Full-duplex communication), transmit, receive
	Transfer mode		Burst transfer / Continuous transfer
Data Format	Frame length		8 to 32 bits (adjustable in bit units)
	FIFO stage		16 bits × 8 stages / 32 bits × 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	The direction of data transfer		Selection of the LSB first / MSB first is possible.
Ganged Control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Start Communication trigger
	Status flag	Common	TSPI modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Start Communication trigger
	DMA request	Transmit	Single DMA request / burst DMA request
		Receive	Single DMA request / burst DMA request
	Trigger control (Note 1)	Trigger input	Start Communication trigger → Start communication
		Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception completed trigger
Special Control (Note2)	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
	RXD timing		1st edge / 2nd edge (timing for data sampling)
	Frame interval period		Burst transfer: 0/f_{SCK} to 15/f_{SCK}
	Idle period		Continuous transfer: 1/f_{SCK} to 15/f_{SCK}
	Software reset		Reset by software

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: f_{SCK}: Transfer Clock frequency

Table 1.5 Function outline (SIO mode, Slave, Frame)

Function classification			A Functional Description or the range
Transmit and Receive Control	Communication modes		SIO mode, slave mode, frame mode
	Communication Operation mode		Transmit and receive(Full-duplex communication), transmit, receive
	Transfer mode		Burst transfer / Continuous transfer
Data Format	Frame length		8 to 32 bits (adjustable in bit units)
	FIFO stage		16 bits × 8 stages / 32 bits × 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	The direction of data transfer		Selection of the LSB first / MSB first is possible.
Ganged Control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Underrun / Overrun
	Status flag	Common	TSPI modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Underrun / Overrun
	DMA request	Transmit	Single DMA request / burst DMA request
		Receive	Single DMA request / burst DMA request
	Trigger control (Note 1)	Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception completed trigger
Special Control	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
			When an underrun error occurs: High, Low
	TXD timing (Note 2)		Last data hold time: 2/fsys to 128/fsys
	RXD timing		1st edge / 2nd edge (timing for data sampling)
	Software reset		Reset by software

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: fsys: System Clock frequency

Table 1.6 Functions Outline (SPI mode, Master, Sector)

Function Classification			Function Description or the Range
Transfer clock	Prescaler		The input clock can be divided by 1,2,4 up to 512
	Baud rate generator		The prescaler output can be divided by 1,2,3 up to 16
	Transfer clock generation		The baud rate generator output can be divided by 2
Transmit and Receive Control	Communication modes		SPI mode, master mode, sector mode
	Communication operation mode		Transmit and receive(Full-duplex communication), Transmit, Receive
	Transfer mode		Continuous transfer
Data format	Sector count/Sector length		2 to 4 sectors / 1 to 32 bits (adjustable in bit units)
	Frame length		8 to 128 bits (all sectors)
	Number of FIFO stages		32 bits x 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	Direction of data transfer		Selection of LSB first / MSB first is provided.
Ganged control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Start Communication trigger
	Status flag	Common	TSPI modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Start Communication trigger
	DMA request	Transmission	Single DMA request / Burst DMA request
		Reception	Single DMA request / Burst DMA request
	Trigger control (Note 1)	Trigger input	Start Communication trigger → Start communication
		Trigger output	Transmission complete → Transmission Complete trigger, Reception complete → Reception Completed trigger
Special control (Note 2)	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
	RXD timing		1st edge / 2nd edge (timing for data sampling)
	CS polarity selection		High, Low: (Level of the asserted period)
	CS timing		Cycle after CS assertion: $1/f_{SCK}$ - $16/f_{SCK}$
			Cycle before CS deassertion: $1/f_{SCK}$ - $16/f_{SCK}$
	Idle period		Continuous transfer: $1/f_{SCK}$ - $15/f_{SCK}$
	Software reset		Reset by software

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: f_{SCK}: Transfer clock frequency

Table 1.7 Functions Overview (SPI mode, Slave, Sector)

Function Classification			Function Description or the Range
Transmit and Receive Control	Communication modes		SPI mode, slave mode, sector mode
	Communication operation mode		Transmit and receive(Full-duplex communication), Transmit, Receive
	Transfer mode		Continuous transfer
Data format	Sector count/Sector length		2 to 4 sectors / 2 to 32 bits (adjustable in bit units)
	Frame length		8 to 128 bits (all sectors)
	Number of FIFO stages		32 bits × 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	Direction of data transfer		Selection of LSB first / MSB first is provided.
Ganged control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Underrun / Overrun
	Status flag	Common	TSPI modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Underrun / Overrun
	DMA request	Transmission	Single DMA request / Burst DMA request
		Reception	Single DMA request / Burst DMA request
	Trigger control (Note)	Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception completed trigger
Special control	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
			When an underrun error occurs: High, Low
	RXD timing		2nd edge (timing for data sampling)
	CS polarity selection		High, Low: (Level of the asserted period)
	Software reset		Reset by software

Note: For the synced peripheral, refer to reference manual "Product Information".

Table 1.8 Functions Overview (SIO mode, Master, Sector)

Function Classification			Function Description or the Range
Transfer clock	Prescaler		The input clock can be divided by 1,2,4 up to 512
	Baud rate generator		The prescaler output can be divided by 1,2,3 up to 16
	Transfer clock generation		The baud rate generator output can be divided by 2
Transmit and Receive Control	Communication modes		SIO mode, master mode, sector mode
	Communication operation mode		Transmit and receive(Full-duplex communication), Transmit, Receive
	Transfer mode		Continuous transfer
Data format	Sector count/Sector length		2 to 4 sectors / 1 to 32 bits (adjustable in bit units)
	Frame length		8 to 128 bits (all sectors)
	Number of FIFO stages		32 bits x 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	Direction of data transfer		Selection of LSB first / MSB first is provided.
Ganged control	Interruption	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Start Communication trigger
	Status flag	Common	TSPI modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Start Communication trigger
	DMA request	Transmission	Single DMA request / Burst DMA request
		Reception	Single DMA request / Burst DMA request
	Trigger control (Note 1)	Trigger input	Start Communication trigger → Start communication
		Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception completed trigger
Special control	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
	RXD timing		1st edge / 2nd edge (timing for data sampling)
	Idle period (Note 2)		Continuous transfer: $1/f_{SCK}$ - $15/f_{SCK}$
	Software reset		Reset by software

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: f_{SCK}: Transfer clock frequency

Table 1.9 Functions Overview (SIO mode, Slave, Sector)

Function classification			Function Description or the Range
Transmit and Receive Control	Communication modes		SIO mode, slave mode, sector mode
	Communication operation mode		Transmit and receive(Full-duplex communication), Transmit, Receive
	Transfer mode		Continuous transfer
Data format	Sector count/Sector length		2 to 4 sectors / 2 to 32 bits (adjustable on a bit level)
	Frame length		8 to 128 bits (all sectors)
	Number of FIFO stages		32 bits x 4 stages
	Parity		Following selections are available: no parity, even parity / odd parity.
	Direction of data transfer		Selection of LSB first / MSB first is provided.
Ganged control	Interrupt	Transmission	Complete/FIFO
		Reception	Complete/FIFO
		Error	Vertical parity / Underrun / Overrun
	Status flag	Common	TSPI modify status
		Transmission	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO empty
		Reception	Operating / Complete / FIFO interrupt / FIFO fill level / FIFO full
		Error	Vertical parity / Underrun / Overrun
	DMA request	Transmission	Single DMA request / Burst DMA request
		Reception	Single DMA request / Burst DMA request
	Trigger control	Trigger output	Transmission complete → Transmission complete trigger, Reception complete → Reception completed trigger
Special control	SCK polarity selection		High, Low: (Level during the idle period)
	TXD level selection		During the idle period: High, Low, Last data, Hi-Z
			When an underrun error occurs: High, Low
	TXD timing		Last data hold time: 2/fsys to 128/fsys
	RXD timing		2nd edge (timing for data sampling)
	Software reset		Reset by software

Note 1: For the synced peripheral, refer to reference manual "Product Information".

Note 2: fsys: System clock frequency

2. Configuration

The block diagram of the TSPI and the signal list are shown.

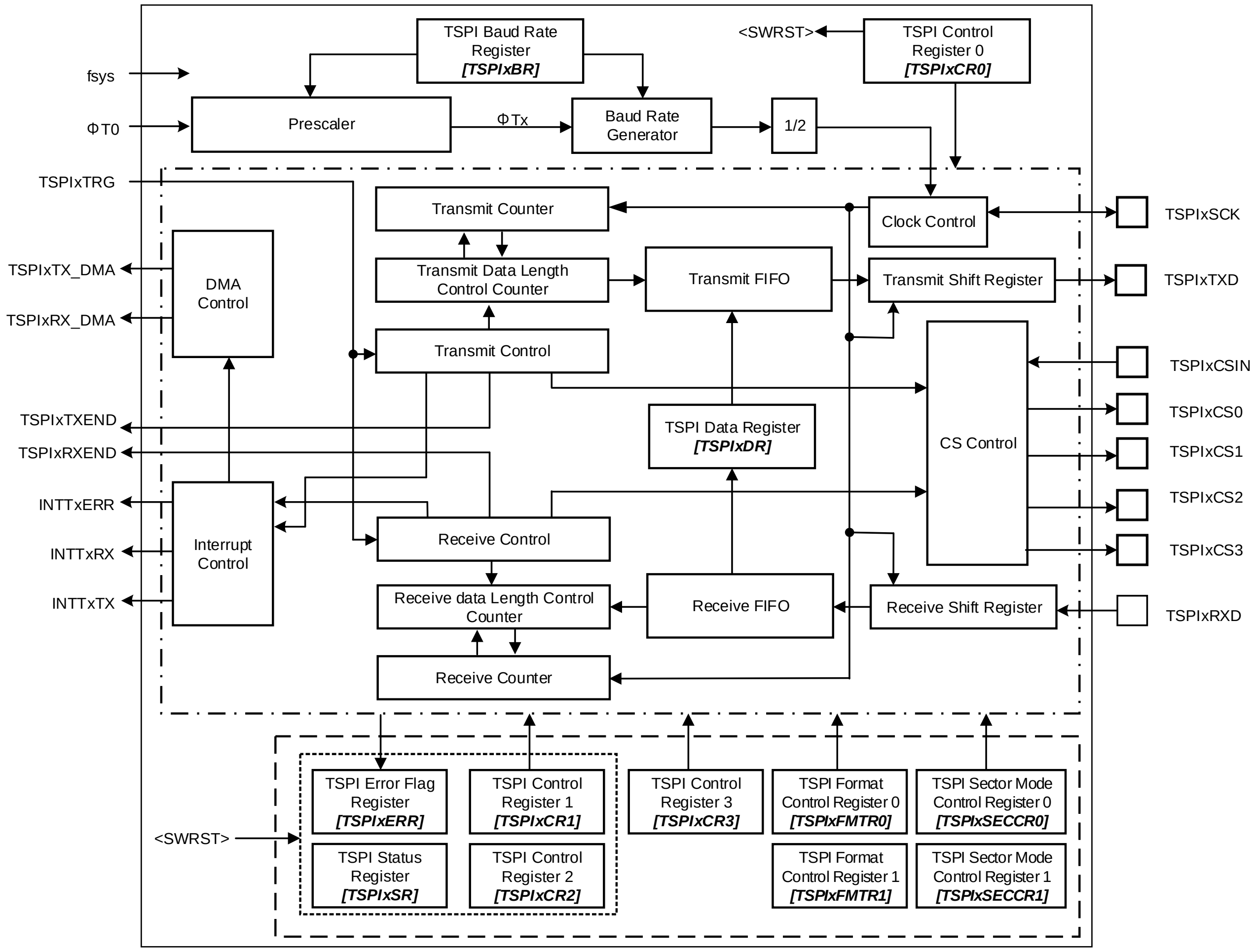

TSPI Baud Rate Register [TSPIxBR]
<SWRST>
TSPI Control Register 0 [TSPIxCR0]
fsys
ΦT0
Prescaler
ΦTx
Baud Rate Generator
1/2
TSPIxTRG
Transmit Counter
Clock Control
TSPIxSCK
TSPIxTX_DMA
TSPIxRX_DMA
DMA Control
Transmit Data Length Control Counter
Transmit FIFO
Transmit Shift Register
TSPIxTXD
Transmit Control
TSPIxCSIN
TSPIxTXEND
TSPI Data Register [TSPIxDR]
TSPIxCS0
TSPIxRXEND
CS Control
TSPIxCS1
INTTxERR
TSPIxCS2
INTTxRX
Interrupt Control
Receive Control
TSPIxCS3
INTTxTX
Receive data Length Control Counter
Receive FIFO
Receive Shift Register
TSPIxRXD
Receive Counter
TSPI Error Flag Register [TSPIxERR]
TSPI Control Register 1 [TSPIxCR1]
TSPI Control Register 3 [TSPIxCR3]
TSPI Format Control Register 0 [TSPIxFMTR0]
TSPI Sector Mode Control Register 0 [TSPIxSECCR0]
<SWRST>
TSPI Status Register [TSPIxSR]
TSPI Control Register 2 [TSPIxCR2]
TSPI Format Control Register 1 [TSPIxFMTR1]
TSPI Sector Mode Control Register 1 [TSPIxSECCR1]

Figure 2.1 Block diagram of TSPI

Table 2.1 List of Signals

No	Signal symbol	Signal name	I/O	Reference manual
1	fsys	System clock	Input	Clock Control and Operation Mode
2	ΦT0	Clock for prescaler	Input	Clock Control and Operation Mode
3	TSPIxSCK	Serial clock output / Serial clock input	Input/Output	Input/Output Ports
4	TSPIxCS0	Chip select 0	Output	Input/Output Ports
5	TSPIxCS1	Chip select 1	Output	Input/Output Ports
6	TSPIxCS2	Chip select 2	Output	Input/Output Ports
7	TSPIxCS3	Chip select 3	Output	Input/Output Ports
8	TSPIxCSIN	Chip select input for slave operation	Input	Input/Output Ports
9	TSPIxTXD	Serial data of transmit	Output	Input/Output Ports
10	TSPIxRXD	Serial data of receive	Input	Input/Output Ports
11	INTTxTX	Transmit interrupt	Output	Exception
12	INTTxRX	Receive interrupt	Output	Exception
13	INTTxERR	Error interrupt	Output	Exception
14	TSPIxTRG	Trigger input for start communication	Input	Product Information
15	TSPIxTX_DMA	Transmit DMA request	Output	Product Information
16	TSPIxRX_DMA	Receive DMA request	Output	Product Information
17	TSPIxTXEND	Transmit complete trigger	Output	Product Information
18	TSPIxRXEND	Receive complete trigger	Output	Product Information

3. Operation Description

3.1. Basic Operation

3.1.1. Clock Supply

When TSPI is used, the corresponding clock enable bits should be set to "1" (Clock supply) in fsys supply stop register A (***[CGFSYSENA]*** and ***[CGFSYSMENA]***), fsys supply stop register B (***[CGFSYSENB]*** and ***[CGFSYSMENB])***, fsys supply stop register C (***[CGFSYSMENC]***), and fc supply stop register (***[CGFCEN]***).

The corresponding registers and the bit locations depend on a product. Some products do not have all registers. For the details, refer to reference manual "Clock Control and Operation Mode".

When stopping the supply of a clock, please check that TSPI has stopped (***[TSPIxCR0]***<TSPIE>=0. Moreover, also when you change the operational mode to STOP1/STOP2, please check that TSPI has stopped.

3.1.2. Start and Stop Transfer

First, set ***[TSPIxCR0]***<TSPIE>(operation control register) to "1". After verifying that it has stopped, conduct the necessary configurations including the communication mode, the transmit mode, and the transfer format.

Also, when modifying various settings or enabling communications and the Start communication trigger, make sure that it has stopped before changing the settings.

Stopped status: ***[TSPIxSR]*** <TSPISUE>=0 and ***[TSPIxCR1]*** <TRGEN>=0, <TRXE>=0

There are two methods for a transfer start in the case of transmit and receive(full-duplex communication)full-duplex communication mode and transmitting mode.

1. Write Data to data register ***[TSPIxDR],*** after wrote "1" to ***[TSPIxCR1]*** <TRXE> to enable communication.

2. Write "1" to ***[TSPIxCR1]***<TRXE>, after wrote data to data register ***[TSPIxDR]***.

In the case of receiving mode, receive is started immediately after setting to ***[TSPIxCR1]***<TRXE>=1.

To stop the transfer, set ***[TSPIxCR1]***<TRXE>=0. For both burst transfer and continuous transfer, the transfer of a frame being transmitted is completed. For details, refer to "Table 4.2 Transfer state and setting enabled state flag at communication stop setting."

After stopped, TSPIxSCK, TSPIxCS0/1/2/3, and TSPIxTXD will be in an idle state. Refer to "3.10. Special Control".

When a transfer is enabled again after stopping a burst transfer, the operation starts from the first frame of the number of frames set in the ***[TSPIxCR1]***<FC[7:0]> (configuration register for the transfer frame count).

3.2. Transfer Clock

"Clock for prescaler" (ΦT0) is used as a general term that includes the clock for high speed prescaler (ΦT0h) and the clock for middle speed prescaler (ΦT0m).

A transfer clock used during master operation is generated from ΦT0 in a transfer clock generation circuit.

For more information on ΦT0 on different products and channels, refer to reference manual "Product Information".

For the maximum frequencies on f_{SCK} and f_{SCKi}, refer to the electrical characteristics in the data sheet of each product.

3.2.1. Transfer Clock Frequency

3.2.1.1. Master operation

The transfer clock generation circuit is shown in Figure 3.1.

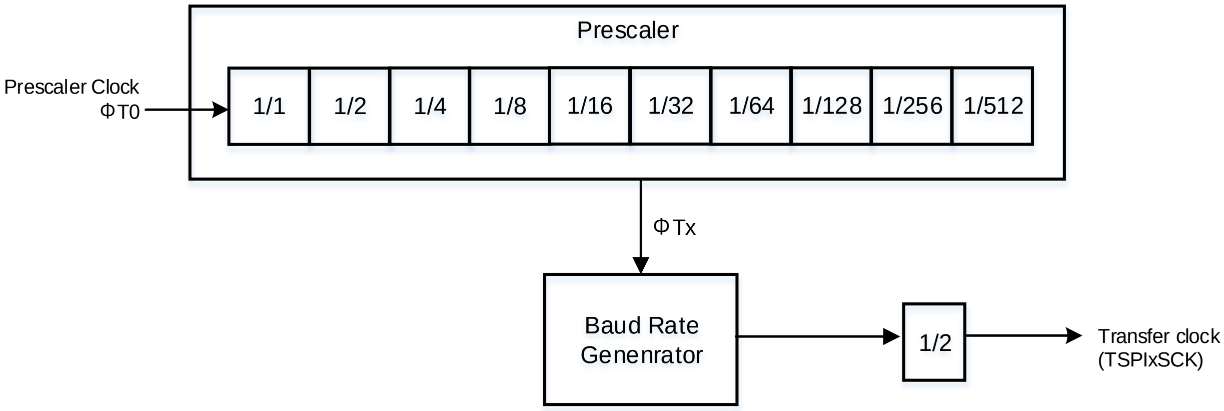


Figure 3.1 Transfer clock generation circuit

The prescaler divides ΦT0 by 1 to 512 (ΦT0 to ΦT256). The divided clock can be selected by ***[TSPIxBR]***<BRCK[3:0]>.

Baud rate generator divides ΦTx by 1 to 16. The division ratio is selected in ***[TSPIxBR]***<BRS[3:0]>.

The example of calculation of transfer clock frequency (hereinafter f_{SCK}) is shown below, Table 3.1 shows examples.

f_{SCK}=ΦT0 × ***[TSPIxBR]***<BRCK[3:0]> (1/x) ×***[TSPIxBR]***<BRS[3:0]> (1/N) × 1/2
(x=1, 2, 4, 8, 16 ... 256, 512 N=1, 2, 3, 4, ... 16)

Note: ΦT0 should be set less than or equal to fsys (i.e. fsys≥ΦT0).

Table 3.1 Transfer clock generation example

[TSPIxBR] <BRCK[3:0]>	*[TSPIxBR]* <BRS[3:0]>	ΦT0	Clock for prescaler ΦT0 [MHz]										
		Divide	40	50	60	80	100	120	140	150	160	180	200
0000	0001	2	20	25									
0000	0010	4	10	12.5	15	20	25						
0001	0001												
0000	0011	6	6.6	8.3	10	13.3	16.6	20	23.3	25			
0000	0100	8	5	6.2	7.5	10	12.5	15	17.5	18.7	20	22.5	25
0001	0010												
0010	0001												
0000	0101	10	4	5	6	8	10	12	14	15	16	18	20
0000	0110	12	3.3	4.1	5	6.6	8.3	10	11.6	12.6	13.3	15	16.6
0001	0011												
0000	0111	14	2.8	3.5	4.2	5.7	7.1	8.5	10	10.7	11.4	12.8	14.2
0000	1000	16	2.5	3.1	3.7	5	6.2	7.5	8.7	9.3	10	11.2	12.5
0001	0100												
0010	0010												
0011	0001												
0000	1001	18	2.2	2.7	3.3	4.4	5.5	6.6	7.7	8.3	8.8	10	11.1
0000	1010	20	2	2.5	3	4	5	6	7	7.5	8	9	10
0001	0101												
0000	1011	22	1.8	2.2	2.7	3.6	4.5	5.4	6.3	6.8	7.2	8.1	9.0
0000	1100	24	1.6	2.0	2.5	3.3	4.1	5	5.8	6.2	6.6	7.5	8.3
0001	0110												
0010	0011												
0000	1101	26	1,5	1.9	2.3	3.0	3.8	4.6	5.3	5.7	6.1	6.9	7.6
0000	1110	28	1.4	1.7	2.1	2.8	3.5	4.2	5	5.3	5.7	6.4	7.1
0001	0111												
0000	1111	30	1.3	1.6	2	2.6	3.3	4	4.6	5	5.3	6	6.6
0000	0000	32	1.2	1.5	1.8	2.5	3.1	3.7	4.3	4.6	5	5.6	6.2
0001	1000												
0010	0100												
0011	0010												
0100	0001												
0001	1001	36	1.1	1.3	1.6	2.2	2.7	3.3	3.8	4.1	4.4	5	5.5
0001	1010	40	1	1.2	1.5	2	2.5	3	3.5	3.7	4	4.5	5
0010	0101												
0001	1011	44	0.9	1.1	1.3	1.8	2.2	2.7	3.1	3.4	3.6	4.0	4.5
0001	1100	48	0.8	1.0	1.2	1.6	2.0	2.5	2.9	3.1	3.3	3.7	4.1
0010	0110												
0011	0011												
0001	1101	52	0.7	0.9	1.1	1.5	1.9	2.3	2.6	2.8	3.0	3.4	3.8
0001	1110	56	0.7	0.8	1.0	1.4	1.7	2.1	2.5	2.6	2.8	3.2	3.5
0010	0111												
0001	1111	60	0.6	0.8	1	1.3	1.6	2	2.3	2.5	2.6	3	3.3
0001	0000	64	0.6	0.7	0.9	1.2	1.5	1.8	2.1	2.3	2.5	2.8	3.1
0010	1000												
0011	0100												
0100	0010												
0101	0001												

Note: When setting conditions, the transfer clock cannot exceed 25 MHz (gray cells). It is also possible that some products have their maximum transfer clock set to less than 25 MHz. For more information on each product, refer to its datasheet and reference manual "Product Information".

3.2.1.2. Generating condition of the transfer clock

The condition for generating TSPI's transfer clock frequency (f_{SCK}) can be configured.

The condition for generating f_{SCK} includes the frequency ratio of the system clock (fsys) and the clock for prescaler (ΦT0).

The relationships between each clock and the configurable range for its register are as follows.

Condition for generating f_{SCK}:

fsys / f_{SCK} =2^L×2^M×2N

(fsys / f_{SCK}: Ratio between the system clock (fsys) and the transfer clock (f_{SCK}))

fsys / ΦT0 =2^L =1/16,1/8,1/4,1/2,1,2,4,8 to 512 L= -4 to 9

(2^L: Ratio between the system clock (fsys) and the clock for prescaler (ΦT0))

ΦT0 / ΦTx = 2^M =1,2,4,8,16,32,64,128,256,512 M = 0 to 9

(2^M: Division value for prescaler by the ***[TSPIxBR]***<BRCK[3:0]>)

ΦTx / f_{SCK} = 2N =2,4,6,8,10,12 to 22,24,26,28,30,32 N= 1 to 16

(N: Division value for baud rate generator by the ***[TSPIxBR]***<BRS[3:0]>)

The combinations of the configuration of each register and the condition for generating f_{SCK} is shown in Table 3.2.

Table 3.2 Condition for generating a transfer clock and the register configuration value

fsys / ΦT0	*[TSPIxBR]*<BRCK[3:0]>	*[TSPIxBR]*<BRS[3:0]>	fsys / f_{SCK}	*[TSPIxCR2]*<RXDLY[2:0]>
1/16 to 1/2 (Note)	-	-	-	-
1	0000	0001	2	000
1	0000	0010	4	000 to 001
1	0000	0011	6	000 to 010
1	0000	0100	8	000 to 011
1	0000	0101	10	000 to 100
1	0000	0110	12	000 to 101
1	0000	0111	14	000 to 110
1	0000	0000, 1000 to1111	16 or more	000 to 111
1	0001	0001	4	000 to 001
1	0001	0010	8	000 to 011
1	0001	0011	12	000 to 101
1	0001	0000, 0100 to 1111	16 or more	000 to 111
1	0010	0001	8	000 to 011
1	0010	0000, 0010 to 1111	16 or more	000 to 111
1	0011 to 1001	0000 to 1111	16	000 to 111
2	0000	0001	4	000 to 001
2	0000	0010	8	000 to 011
2	0000	0011	12	000 to 101
2	0000	0000, 0100 to 1111	16 or more	000 to 111
2	0001	0001	8	000 to 011
2	0001	0000, 0010 to 1111	16 or more	000 to 111
2	0010 to 1001	0000 to 1111	16 or more	000 to 111
4	0000	0001	8	000 to 011
4	0000	0000, 0010 to 1111	16 or more	000 to 111
4	0001 to 1001	0000 to 1111	16 or more	000 to 111
8 to 512	0000 to 1001	0000 to 1111	16 or more	000 to 111

Note: The use of the 1/16 to 1/2 in fsys / ΦT0 is prohibited.

3.2.1.3. Slave Operation

The input frequency (hereinafter f_{SCKi}) of the transfer clock should be set to the following conditions.
When using RXD sampling at the 1st edge, use under the condition of $4 \leq fsys / f_{SCKi}$.
When using RXD sampling at the 2nd edge, use under the conditions shown in Table 3.3.

Table 3.3 Conditions of a transfer clock for slave operation(2nd edge)

Condition for f_{SCKi} fsys / f_{SCKi} = N	Sector mode Sector length	Frame mode Frame length
N<2	-	-
2≤N	4 to 32 bits	All range (8 to 32 bits)
4≤N	2 to 32 bits	

3.3. Communication Mode

A communication mode is selected.

Specify the SPI or SIO mode in ***[TSPIxCR1]***<TSPIMS> (communication mode selection register)

Specify the master operation or slave operation in ***[TSPIxCR1]***<MSTR> (master or slave selection register)

Specify the frame or sector in ***[TSPIxSECTCR0]***<SECT> (sector mode selection register)

Communication modes along with their compatible operation specifications are shown in Table 3.4.

Table 3.4 Communication modes and the specifications for the compatible operations

Transmit and Receive Control					Data format				Special control	
Communication modes			Communication operation mode	Transfer mode	Frame length (bit)	FIFO Bit × stage	Common	Ganged control	Level selection	Timing selection
SPI	Master operation	Frame	Transmit and receive (Full-duplex) Transmit, Receive	Continuous burst	8 to 32	16×8 32×4	Parity -None -Even -Odd Transfer direction -MSB -LSB	Interrupt DMA Trigger -Start communication -Complete transmit -Complete receive	SCK polarity TXD -Idle period CS polarity	RXD sampling -1st edge -2nd edge CS timing Frame interval Idle time
		Sector		Continuous	8 to 128	32×4				
	Slave operation	Frame		Continuous burst	8 to 32	16×8 32×4		Interrupt DMA Trigger -Complete transmit -Complete receive	SCK polarity TXD -Idle period -Underrun error CS polarity	RXD sampling -1st edge -2nd edge
		Sector		Continuous	8 to 128	32×4				RXD sampling -2nd edge
SIO	Master operation	Frame		Continuous burst	8 to 32	16×8 32×4		Interrupt DMA Trigger -Start communication -Complete transmit -Complete receive	SCK polarity TXD -Idle period	RXD sampling -1st edge -2nd edge Frame interval Idle time
		Sector		Continuous	8 to 128	32×4				
	Slave operation	Frame		Continuous burst	8 to 32	16×8 32×4		Interrupt DMA Trigger -Complete transmit -Complete receive	SCK polarity TXD -Idle period -Underrun error	RXD sampling -1st edge -2nd edge TXD last data hold period
		Sector		Continuous	8 to 128	32×4				RXD sampling -2nd edge TXD last data hold period

3.3.1. SPI / SIO Mode Selection

The SPI mode or the SIO mode is selected. When ***[TSPIxCR1]***<TSPIMS> (communication mode selection register) is set to "0", the SPI mode is selected. When it is set to "1", the SIO mode is selected.

(1) SPI mode

In SPI mode, you can communicate as a master or slave by using the terminals of TSPIxSCK(clock input/output), TSPIxCS0/1/2/3(Chip select output), TSPIxCSIN(Chip select input), TSPIxTXD(data transmit), and TSPIxRXD(data receive).

In addition, it has four built-in chip select signal outputs (TSPIxCS0 / 1/2/3), and can communicate with four external slave devices as a master device. Moreover, it has one built-in chip select signal input (TSPIxCSIN) and can communicate with one master device as a slave device.

Note: The number of chip select outputs(TSPIxCS0/1/2/3) are different product by product. Refer to the datasheet and reference manual "Product Information".

(2) SIO mode

In SIO mode, you can use TSPIxSCK, TSPIxTXD, and TSPIxRXD to communicate as master or slave. Master and slave devices communicate one-to-one.

Note: When using SIO mode, do not select TSPIxCS0/1/2/3 and TSPIxCSIN in the port setting.

3.3.2. Master / Slave Selection

The master device operation (output clock) or the slave device operation (input clock) is selected.

When ***[TSPIxCR1]***<MSTR>(Master / Slave selection register) is set to "0", the TSPI operates as Slave. When ***[TSPIxCR1]***<MSTR> is set to "1", the TSPI operates as Master.

- Master device operation

  The TSPI outputs a clock from TSPIxSCK and operates synchronously with TSPIxSCK.

  In the SPI mode, the chip select signal is output from TSPIxCS0/1/2/3 to select the slave device.

- Slave device operation

  The TSPI receives a clock from TSPIxSCK and operates synchronously with TSPIxSCK.

  In the SPI mode, when not selected by the chip select signal input from TSPIxCSIN, the input to TSPIxSCK is ignored. However, from the start of TSPIxCSIN deassertion to the first edge of the serial clock during TSPIxCSIN deassertion, separate one cycle of the maximum speed of the TSPI slave mode used by the product.

3.3.3. Frame / Sector selection

The frame mode or the sector mode is selected. When ***[TSPIxSECTCR0]***<SECT> (sector mode control register 0) is set to "0" (initial value), the frame mode is selected. When it is set to "1", the sector mode is selected.

The sector mode is selected in case of a frame with 2 to 4 sectors with different lengths or a frame that exceeds 32 bits in length. For details of the sector mode, refer to "3.6.2 Sector mode".

3.4. Communication Operation Mode

3.4.1. Transmit and receive (Full-Duplex) Communication Mode

Figure 3.2 shows an example of operation for continuous transmit and receive (full-duplex) communication when master operation, frame length 32 bits, no parity, and one stage of FIFO is used.(***[TSPIxCR2]***<TIDLE[1:0]>=10)

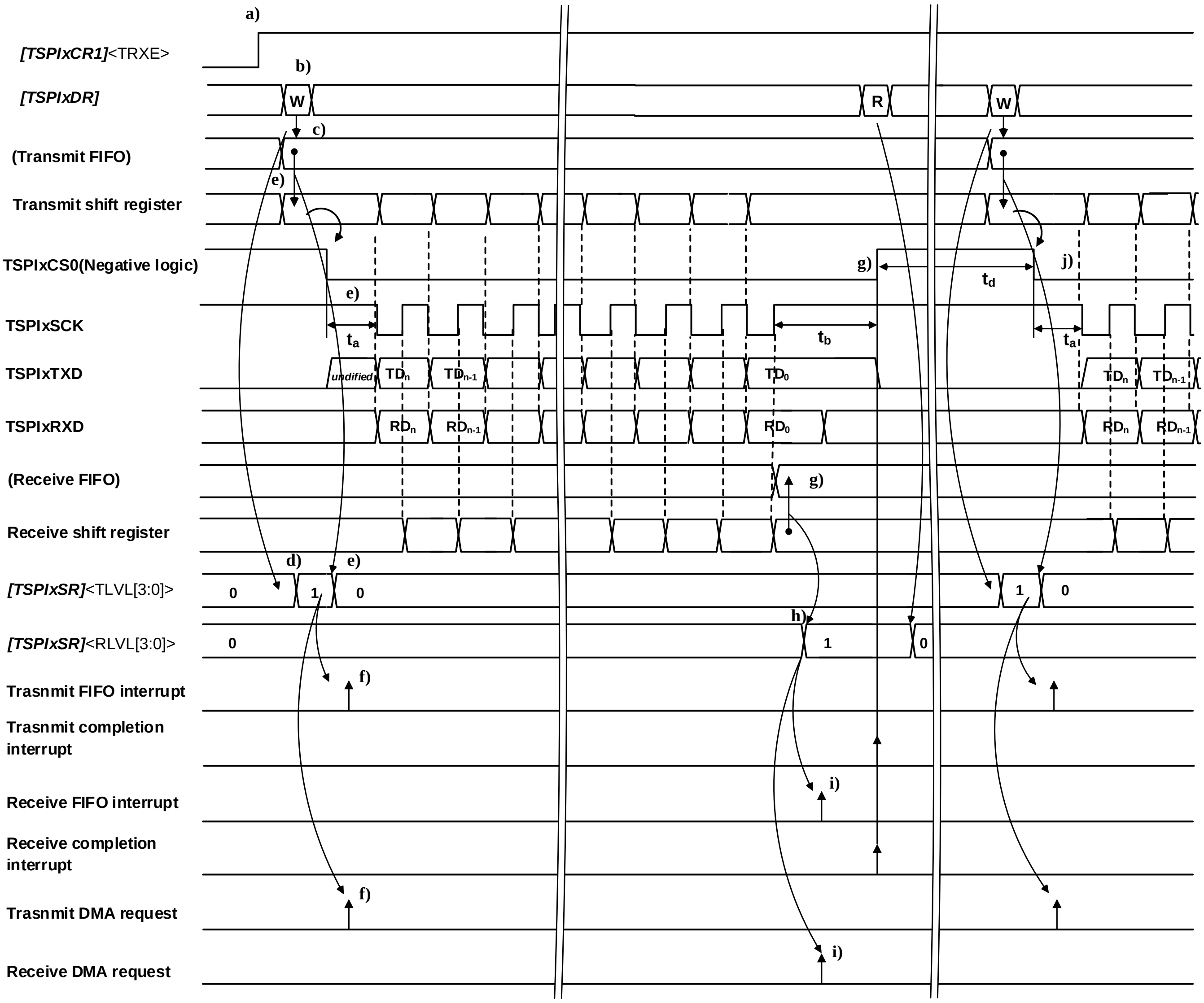


Figure 3.2 Operation example of transmit and receive(full-duplex) communication

a) Write "1" to ***[TSPIxCR1]***<TRXE> to enable communications.
b) Write data to ***[TSPIxDR]***.
c) When data is written to ***[TSPIxDR]***, it is buffered in order from the bottom of the internal transmit FIFO.
d) Since one stage of data is buffered in the transmit FIFO, ***[TSPIxSR]***<TLVL[3:0]> becomes "1".
e) The data buffered to the transmit FIFO is moved to the shift register and ***[TSPIxSR]***<TLVL[3:0]> is set to "0". After the cycle following the CS assertion configured with ***[TSPIxFMTR0]***<CSSCKDL[3:0]> (cycle register following the CS assertion) has elapsed (ta), TSPIxSCK starts outputting a serial clock.
f) Since ***[TSPIxSR]***<TLVL[3:0]> changes to "0" from "1", a transmit FIFO interrupt or transmit DMA request occurs.
g) All bits of the received data are loaded to the receive shift register and moved to the receive FIFO on the rising edge of the last serial clock. After the cycle preceding the CS assertion configured with ***[TSPIxFMTR0]***<SCKCSDL[3:0]> (cycle register preceding the CS deassertion) has elapsed (tb),

SPIxCS0 is deasserted so that a transmit completion interrupt and a receive completion interrupt occur.

h) ***[TSPIxSR]*** <RLVL [3: 0]> becomes "1" because the data for one stage is buffered in the receive FIFO.

i) Since ***[TSPIxSR]***<RLVL[3:0]> changes to "1" from "0", a receive FIFO interrupt or receive DMA request occurs.

j) Until the idle period (td) specified by ***[TSPIxFMTR0]***<CSINT[3:0]> has elapsed after TSPIxCS0 is deasserted, the serial transfer does not start and TSPIxCS0 remains deasserted. After the idle period (td) has elapsed, TSPIxCS0 is asserted and serial transfer starts.

3.4.2. Transmit Mode

Figure 3.3 shows an example of continuous transfer operation in transmit mode when master operation, frame length 32 bits, no parity, and one stage of FIFO is used. (***[TSPIxCR2]***<TIDLE[1:0]>=10)

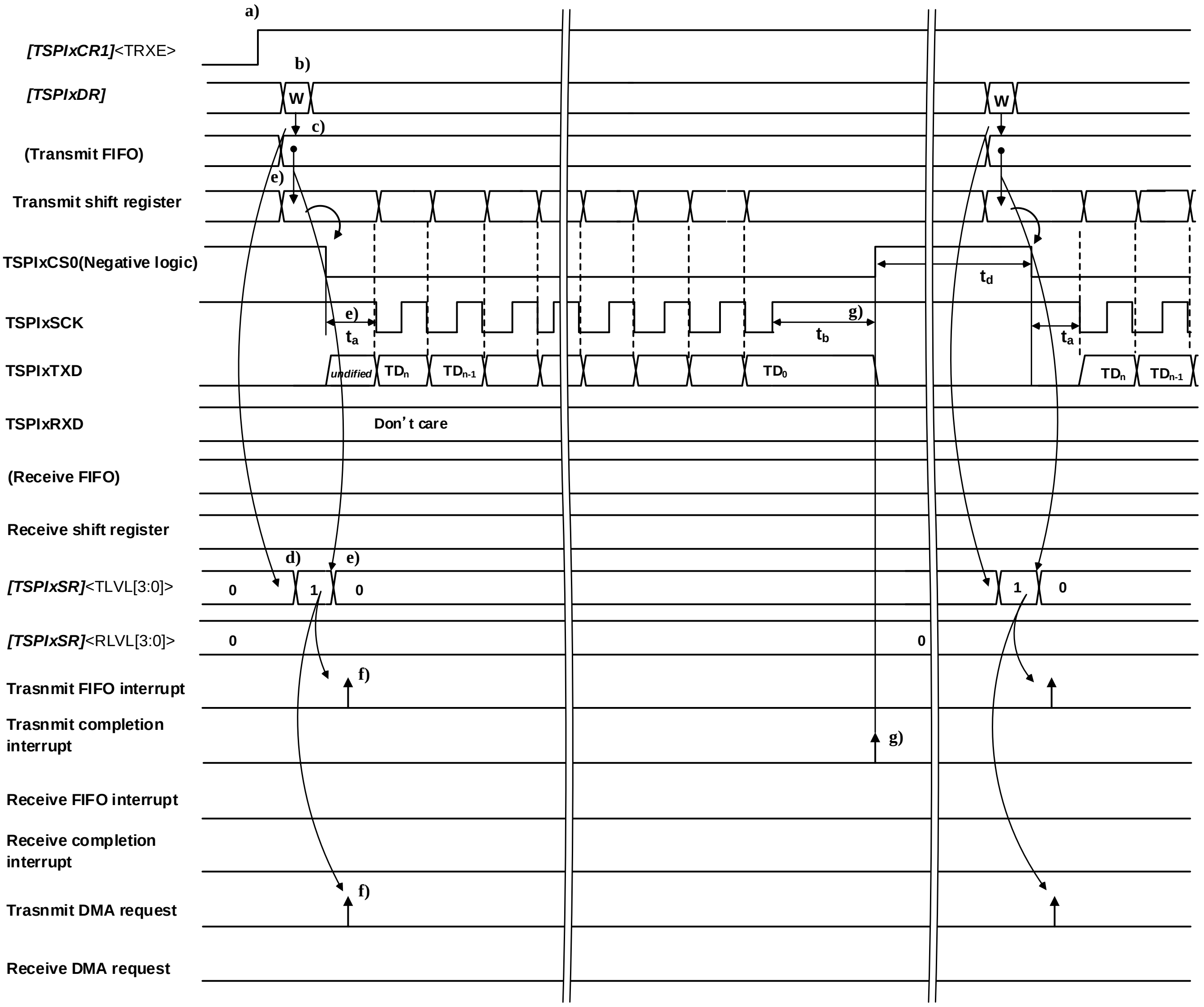


Figure 3.3 Operation example of transmit mode

a) Write "1" to ***[TSPIxCR1]***<TRXE> to enable the communications.
b) Write data to ***[TSPIxDR]***.
c) When the data is written to ***[TSPIxDR]***, the buffer starts from the lower stage of the internal transmit FIFO.
d) Since one stage of data is buffered to the transmit FIFO, ***[TSPIxSR]***<TLVL[3:0]> becomes "1".
e) The data buffered to the transmit FIFO is moved to the shift register, and ***[TSPIxSR]***<TLVL[3:0]> becomes "0". After the CS assert cycle (ta) set in ***[TSPIxFMTR0]***<CSSCKDL[3:0]> has elapsed, the serial clock output starts from TSPIxSCK.
f) Since ***[TSPIxSR]***<TLVL[3:0]> changed to "0" from "1", a transmit FIFO interrupt or transmit DMA request occurs.
g) Until the idle period (td) specified by ***[TSPIxFMTR0]***<CSINT[3:0]> has elapsed after TSPIxCS0 is deasserted, the serial transfer does not start and TSPIxCS0 remains deasserted. After the idle period (td) has elapsed, TSPIxCS0 is asserted and serial transfer starts.

3.4.3. Receive Mode

Figure 3.4 Figure 3.4 shows an example of continuous transfer operation in receive mode when master operation, frame length 32 bits, no parity, and one stage of FIFO is used. (***[TSPIxCR2]***<TIDLE[1:0]>=10).

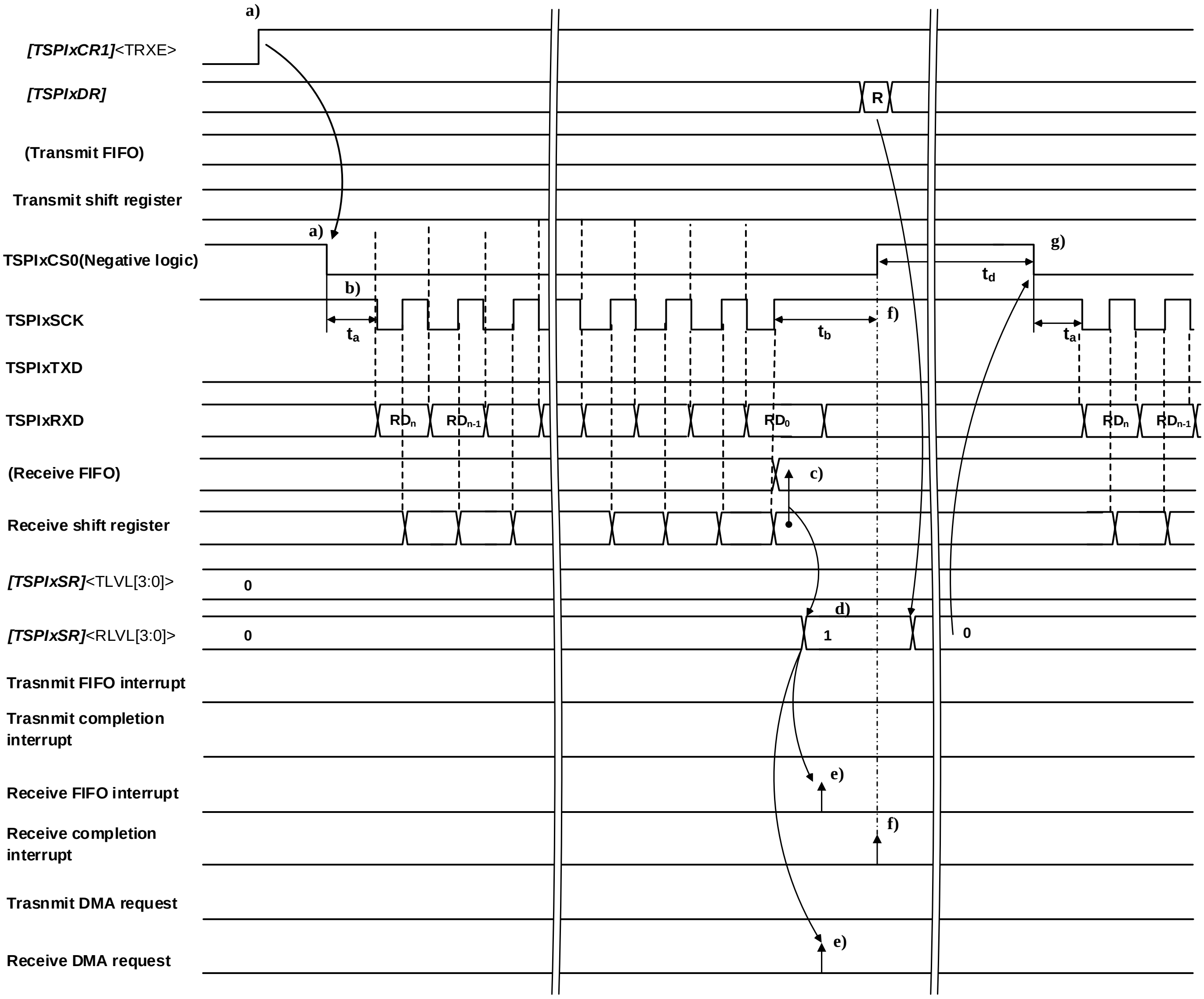


Figure 3.4 Operation example in receive mode

a) Write "1" to ***[TSPIxCR1]***<TRXE> to enable the communications. Since receive FIFO is not full, TSPIxCS0 is immediately asserted and serial transfer starts.
b) After the cycle following the CS assertion (ta) specified by ***[TSPIxFMTR0]***<CSSCKDL[3:0]> has elapsed, the serial clock starts outputting from TSPIxSCK.
c) On the last rising edge of the serial clock, all bits of receive data are captured in the receive shift register and moved to the receive FIFO.
d) Since one stage of data is buffered to the receive FIFO, ***[TSPIxSR]***<RLVL[3:0]> becomes "1".
e) Since ***[TSPIxSR]***<RLVL[3:0]> changed to "1" from "0", a receive FIFO interrupt or receive DMA request occurs.
f) After the cycle preceding the CS deassertion (tb) specified by ***[TSPIxFMTR0]***<SCKCSDL[3:0]> has elapsed after the last rising edge of the serial clock, TSPIxCS0 is deasserted and a receive completion interrupt occurs.
g) Until the idle period (td) specified by ***[TSPIxFMTR0]***<CSINT[3:0]> has elapsed after TSPIxCS0 is deasserted, the serial transfer does not start and TSPIxCS0 remains deasserted. After the idle period (td) has elapsed, TSPIxCS0 is asserted and serial transfer starts if the receive FIFO is not full.

3.5. Transfer Mode

There are two types of transfer mode; the burst transfer in which the transfer frame count (1 to 255) is specified for a transfer, and the continuous transfer where the transfer frame count is not specified.

The transfer mode and the frame count can be specified on ***[TSPIxCR1]***<FC[7:0]> (transfer frame count configuration register)

Table 3.5 Transfer mode compatibility

Frame / Sector selection	Transfer mode	
	Burst transfer	Continuous transfer
Frame mode	✓	✓
Sector mode	×	✓

3.5.1. Burst Transfer

Burst transfer is a transfer mode where the frame count from 1 to 255, infinite times of transfer is specified.

For master in SPI mode, TSPIxCS0/1/2/3 keeps asserted condition while specified frames are transferring. TSPIxCS0/1/2/3 is deasserted when the transfer of frame is completed.

Burst transfer can be used in the frame mode.

3.5.2. Continuously Transfer

It is the mode which repeats the burst transfer of one frame without specifying the number of transfer frames.

In SPI mode, in the case of a master, TSPIxCS0/1/2/3 are certainly deasserted for every one-frame transfer, and TSPIxCS0/1/2/3 are asserted at the time of transfer of the following frame.

Note1: At transmit in the Slave mode, when the serial transfer is performed with the setting data sampling on the 1st edge (***[TSPIxFMTR0]***<CKPHA>=0), the underrun flag is set to “1” (***[TSPIxERR]***<UDRERR>=1) just after the final data transmit.

Note2: At receive in the Slave mode, when the serial transfer is performed with the setting data sampling on the 1st edge (***[TSPIxFMTR0]***<CKPHA>=0), and the receive is continued in the FIFO full state (***[TSPIxSR]***<RFFLL>=1), the overrun flag is set to “1” (***[TSPIxERR]***<OVERR>=1) just after the final data receive.

The following 2 modes are available for a continuous transfer.

Frame mode

- Transfer mode where a transfer is conducted with a frame unit with the length of 8 to 32 bits.

Sector mode

- Transfer mode where a transfer is conducted with a frame unit with the length of 8 to 128 bits.
- One frame consists of 2-4 sectors.
- The sector length for each sector can be configured in the following range.

Master operation:	1 to 32 bits

Slave operation:	2 to 32 bits (fsys/$f_{SCKi} \geq 4$)
	4 to 32 bits (fsys/$f_{SCKi} \geq 2$)

For details of the sector mode, refer to "3.6.2. Sector mode".

3.6. Data format

The frame mode and the sector mode are available as a data format. When choosing the sector mode, configure ***[TSPIxSECTCR0]*** (sector mode control register 0). To specify the sector length, configure ***[TSPIxSECTCR1]*** (sector mode control register 1). To choose the transfer direction (MSB first / LSB first), configure ***[TSPIxFMTR0]*** (format control register 0). To set the parity (with / no parity) and the type (even / odd), configure ***[TSPIxFMTR1]*** (format control register 1). Note that the data other than the received data are "0" when reading out the data register ***[TSPIxDR]*** (RD_31 to RD_0).

3.6.1. Frame mode

In the frame mode, the burst transfer or the continuous transfer is available when transferring a frame of length from 8 to 32 bits.

The data format for the frame mode is shown in Figure 3.5.

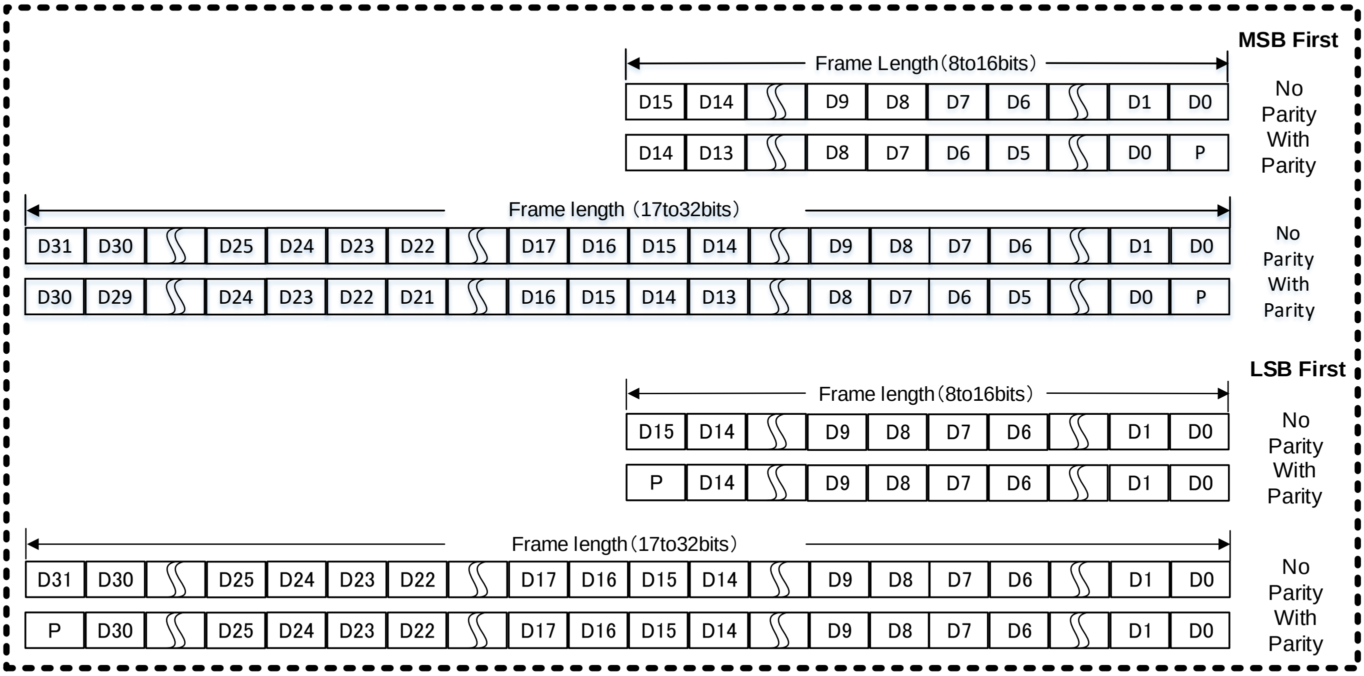

MSB First
Frame Length（8to16bits）
D15 D14 D9 D8 D7 D6 D1 D0
D14 D13 D8 D7 D6 D5 D0 P
No Parity
With Parity
Frame length（17to32bits）
D31 D30 D25 D24 D23 D22 D17 D16 D15 D14 D9 D8 D7 D6 D1 D0
D30 D29 D24 D23 D22 D21 D16 D15 D14 D13 D8 D7 D6 D5 D0 P
No Parity
With Parity
LSB First
Frame length（8to16bits）
D15 D14 D9 D8 D7 D6 D1 D0
P D14 D9 D8 D7 D6 D1 D0
No Parity
With Parity
Frame length（17to32bits）
D31 D30 D25 D24 D23 D22 D17 D16 D15 D14 D9 D8 D7 D6 D1 D0
P D30 D25 D24 D23 D22 D17 D16 D15 D14 D9 D8 D7 D6 D1 D0
No Parity
With Parity

Figure 3.5 Data format overview for the frame mode

No parity

The frame length should be set to the same value as the data length.
When the data length is 31 bits, configure ***[TSPIxFMTR0]***<FL[5:0]> = 011111.

With parity

The frame length should be set to the same value as the data length plus 1 bit, which is a bit length for the parity. Since the frame length is from 8 to 32 bits, the data length is from 7 to 31 bits. When the data length is 31 bits, configure ***[TSPIxFMTR0]***<FL[5:0]> = 100000. For transmit, the parity bit is added to the data in transmit FIFO and is moved to the transmit shift register. For receives, the parity bit is removed from the data in the receive shift register and is moved to the receive FIFO.

3.6.1.1. Buffer Structure and Operation

The transmit buffer and the receive buffer are independent of each other. Each buffer has a double buffer structure that consists of the FIFO and the shift register with the length of 32-bit and can be accessed in the data register ***[TSPIxDR]***.

There are the transmit FIFO and receive FIFO. Each FIFO is 16-bit width and 8-stage or 32-bit width and 4-stage. Settable FIFO level varies depending on the Frame length.

The ranges for the configurable fill levels corresponding to various frame lengths are shown in Table 3.6.

Table 3.6 Frame mode settable fill level

Frame length	Settable Fill level	
	Transmit FIFO ***[TSPIxCR2]***<TIL[3:0]>	Receive FIFO ***[TSPIxCR2]***<RIL[3:0]>
8 to 16 bits	0 to 7	1 to 8
17 to 32 bits	0 to 3	1 to 4

Note: Set a value within the settable fill level. If the value is outside of the settable fill level, the operation is not guaranteed.

Table 3.7 Data format examples for the frame mode

Condition	With / No parity	Transfer direction	Frame length
Frame mode operation 1	No parity	MSB first	32 bits
Frame mode operation 2		LSB first	8 bits
Frame mode operation 3	With parity	MSB first	16 bits
Frame mode operation 4		LSB first	24 bits

The conditions for the operation 1 to 4 in "Table 3.7 Data format examples for the frame mode" are described.

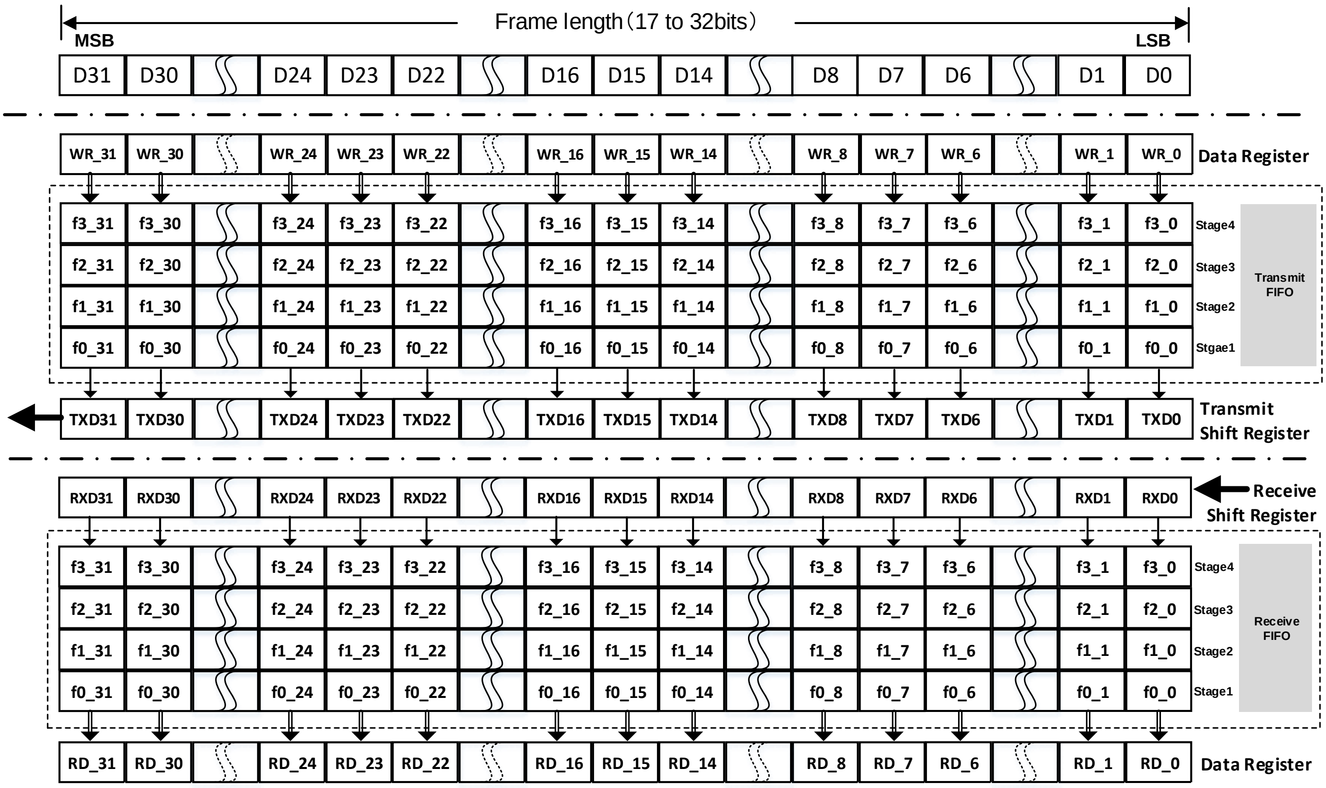
The transmit operations are the process from writing to the data register of the transmit data to output the transmit shift register. The receive operations are the process from inputting the receive data to the receive shift register to read from the data register.

In the figure, the notation of each data is WR_31 to WR_0 for the write data of the data register ***[TSPIxDR]***, RD_31 to RD_0 for the read data, TXD31 to TXD0 for the transmit shift register data, RXD31 to RXD0 for the receive shift register data, and FIFO.

The frame n data in the frame is represented by fn_31 to fn_0 (n = 0 to 3 for the FIFO fill level of 4, n =0 to 7 for the FIFO fill level of 8), and the parity bit is shown as P.

3.6.1.2. Frame mode operation 1 (No parity, MSB first)

The following figure shows an example of transmit no parity, MSB first, frame length of 17 to 32 bits, FIFO fill level of 4 stages.


Frame length（17 to 32bits）
MSB
LSB
D31 D30 D24 D23 D22 D16 D15 D14 D8 D7 D6 D1 D0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data Register
Stage4
Stage3
Stage2
Stgae1
Transmit FIFO
TXD31 TXD30 TXD24 TXD23 TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift Register
RXD31 RXD30 RXD24 RXD23 RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift Register
Receive FIFO
Stage4
Stage3
Stage2
Stage1
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data Register

f3_31	f3_30	f3_24	f3_23	f3_22	f3_16	f3_15	f3_14	f3_8	f3_7	f3_6	f3_1	f3_0	Stage4
f2_31	f2_30	f2_24	f2_23	f2_22	f2_16	f2_15	f2_14	f2_8	f2_7	f2_6	f2_1	f2_0	Stage3
f1_31	f1_30	f1_24	f1_23	f1_22	f1_16	f1_15	f1_14	f1_8	f1_7	f1_6	f1_1	f1_0	Stage2
f0_31	f0_30	f0_24	f0_23	f0_22	f0_16	f0_15	f0_14	f0_8	f0_7	f0_6	f0_1	f0_0	Stage1

Figure 3.6 Buffer operation in Frame mode (No Parity / MSB first)

Stop condition: ***[TSPIxSR]*** <TSPISUE> = 0 and ***[TSPIxCR1]*** <TRGEN> = 0, <TRXE> = 0

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and are stacked to the first level when the transmit FIFO is empty. The data are stacked to the second level when the first level is already filled with data.

Also, when the transmit buffer is empty, the transmit data written after the communication is enabled are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")

Figure 3.6 shows the transmit data from the frames f0 to f3 saved on the transmit FIFO level 1 to 4 stages.

When the operation is enabled to start communications after being stopped or the transmit of a frame has been completed, the data on the first level of the transmit FIFO are moved to the transmit shift register and are transmitted. At the same time, the data on level 2 to 4 stages are moved to level 1 to 3 stages.

The transmit data within the transmit shift register are transmitted for each serial clock, starting from MSB (f0_31) in the frame f0.

(2) Receive buffer operation

For each frame received during communications, the data within the receive shift register are stacked to the first level stage when the receive FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.

Figure 3.6 shows the receive data from the frames f0 to f3 saved on the receive FIFO level 1 to 4 stages.

The received data are saved in the receive shift register for each serial clock, starting from MSB (f0_31) in the frame f0, and are stacked to the lowest level of the receive FIFO whenever a frame is received.

From the data register ***[TSPIxDR]***, the CPU or DMAC reads out the data on the first level stage of the receive FIFO. At the same time, the data on level 2 to 4 stages are moved to level 1 to 3 stages.

3.6.1.3. Frame mode operation 2 (No parity, LSB first)

The following figure shows an example of transmit no parity, LSB first, frame length of 8 to 16 bits, FIFO fill level of 8 stages.

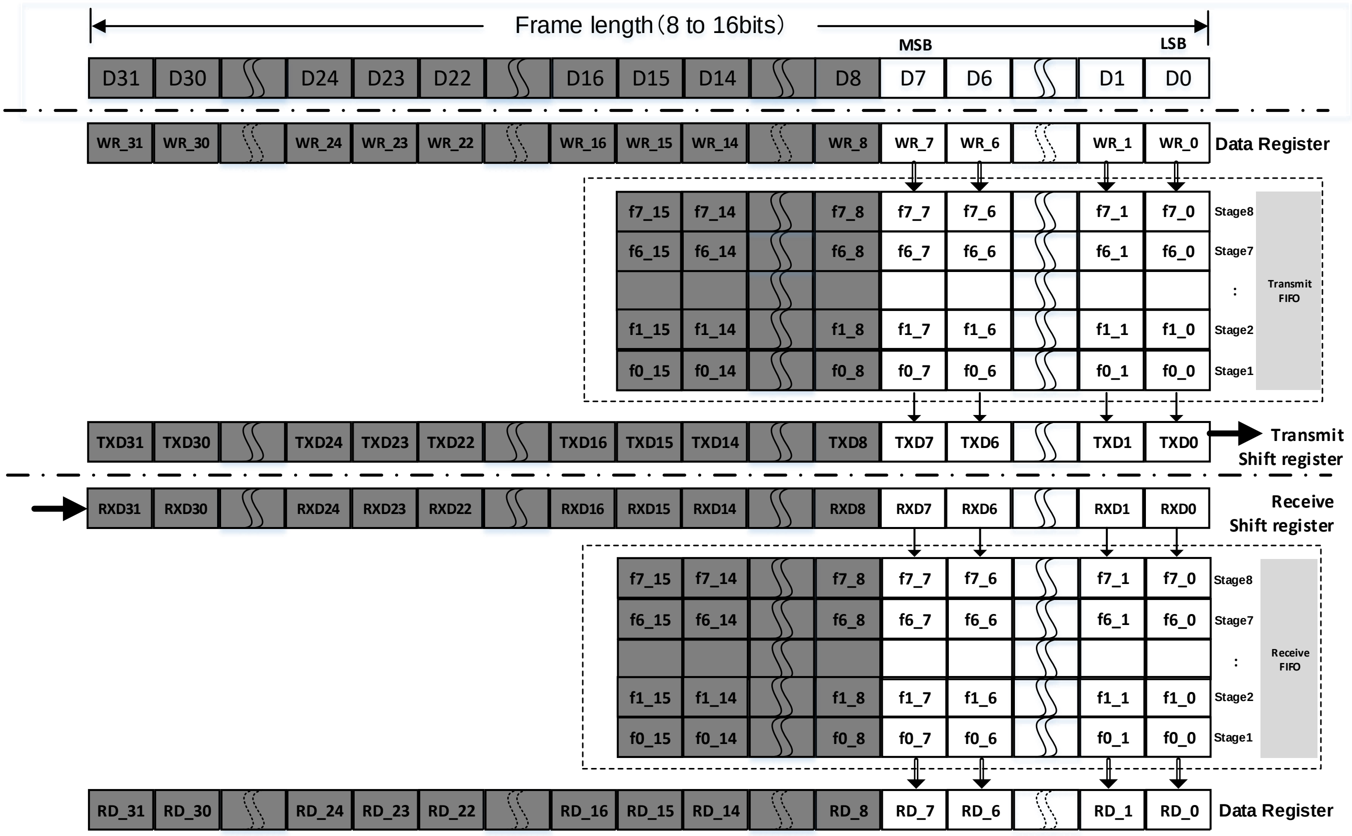

Frame length（8 to 16bits）
MSB
LSB
D31 D30 D24 D23 D22 D16 D15 D14 D8 D7 D6 D1 D0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data Register
f7_15 f7_14 f7_8 f7_7 f7_6 f7_1 f7_0 Stage8
f6_15 f6_14 f6_8 f6_7 f6_6 f6_1 f6_0 Stage7
:
f1_15 f1_14 f1_8 f1_7 f1_6 f1_1 f1_0 Stage2
f0_15 f0_14 f0_8 f0_7 f0_6 f0_1 f0_0 Stage1
Transmit FIFO
TXD31 TXD30 TXD24 TXD23 TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift register
RXD31 RXD30 RXD24 RXD23 RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift register
f7_15 f7_14 f7_8 f7_7 f7_6 f7_1 f7_0 Stage8
f6_15 f6_14 f6_8 f6_7 f6_6 f6_1 f6_0 Stage7
:
f1_15 f1_14 f1_8 f1_7 f1_6 f1_1 f1_0 Stage2
f0_15 f0_14 f0_8 f0_7 f0_6 f0_1 f0_0 Stage1
Receive FIFO
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data Register

Figure 3.7 Buffer Operation in Frame mode (No Parity / LSB First)

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and are stacked to the first level stage when the transmit FIFO is empty. The data are stacked to the second level stage when the first level is already filled with data.
Also, when the transmit buffer is empty, the transmit data written after the communication is enabled are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")
Figure 3.7 Shows the transmit data in the frames f0 to f7 saved on level 1 to 8 stages of the transmit FIFO.
When the operation is enabled to start communications after being stopped or the frames in the transmit have been transmitted, the data on the first level stage of the transmit FIFO are moved to the transmit shift register and are transmitted. At the same time, the data on level 2 to 4 are moved to level 1 to 3stages.
The transmit data in the transmit shift register are transmitted for each serial clock, starting from LSB (f0_0) in the frame f0.

(2) Receive buffer operation

For each frame received during communications, the data within the receive shift register are stacked to the first level stage when the receive FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Figure 3.7 Shows the received data in the frames f0 to f7 saved on the level 1 to 8 stages of the receive FIFO.
The received data are saved in the received shift register for each serial clock, starting from LSB (f0_0) in the frame f0, and are stacked to the lowest level stage of the received FIFO whenever a frame is received.
From the data register ***[TSPIxDR]***, the CPU or DMAC are read out the data on the first level stage of the receive FIFO. At the same time, the data on level 2 to 4 stages are moved to level 1 to 3 stages.

3.6.1.4. Frame mode operation 3 (with parity, MSB first)

The following figure shows an example of transmit with parity, MSB first, frame length 8 to 16 bits, FIFO fill level of 8 stages.

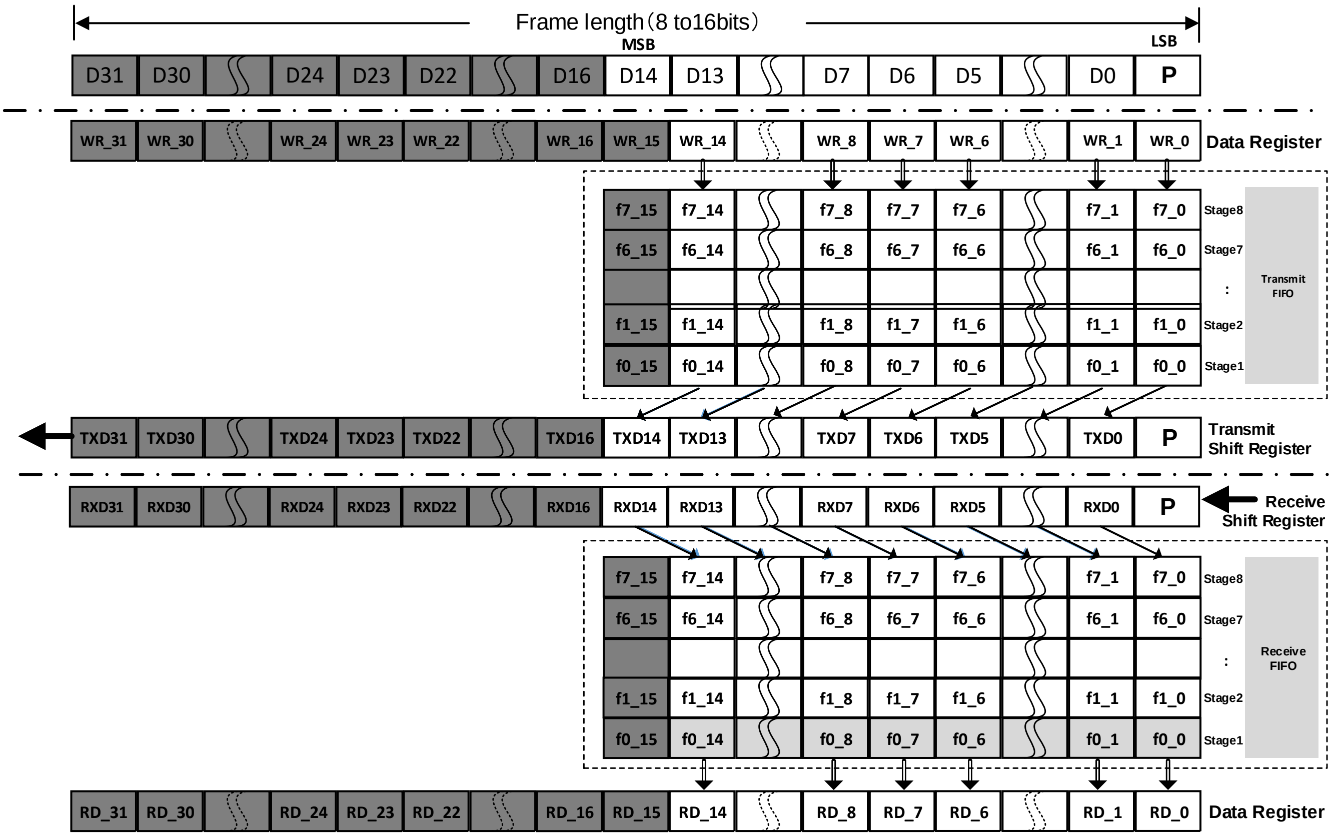


Figure 3.8 Buffer operation in frame mode (with parity, MSB first)

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation are stopped or during communication, and are stacked to the first level stage when the transmit FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Also, when the transmit buffer is empty, the transmit data written after the communication is enabled are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")
Figure 3.8 shows the transmit data in the frames f0 to f7 saved on level 1 to 8 stages of the transmit FIFO.
When the operation is enabled to start communications after being stopped or the frames in the transmit have been transmitted, the data on the first level stage of the transmit FIFO have the parity bit added to LSB, are moved to the transmit shift register and are transmitted. At the same time, the data on level 2 to 8 stages are moved to level 1 to 7stages.
The transmit data in the transmit shift register are transmitted for each serial clock, starting from MSB (f0_14) in the frame f0.

(2) Receive buffer operation

For each frame received during communications, the data in the receive shift register are stacked to the first level stage when the receive FIFO is empty. The data are stacked to the second level when the first level stage is already filled with data.
Figure 3.8 shows the received data in the frames f0 to f7 saved on level 1 to 8 stages of the receive FIFO.
The received data are saved in the receive shift register for each serial clock, starting from MSB (f0_14) in the frame f0. Every time a frame is received, the parity bit is removed from LSB and the received data are stacked to the lowest level stage of the receive FIFO.
From the data register ***[TSPIxDR]***, the CPU or DMAC read out the data on the first level stage of the receive FIFO. At the same time, the data on level 2 to 8 stages are moved to level 1 to 7 stages.

3.6.1.5. Frame mode operation 4 (with parity, LSB first)

The following figure shows an example of transmit with parity, LSB first, frame length of 17 to 32 bits, FIFO fill level of 4 stages.

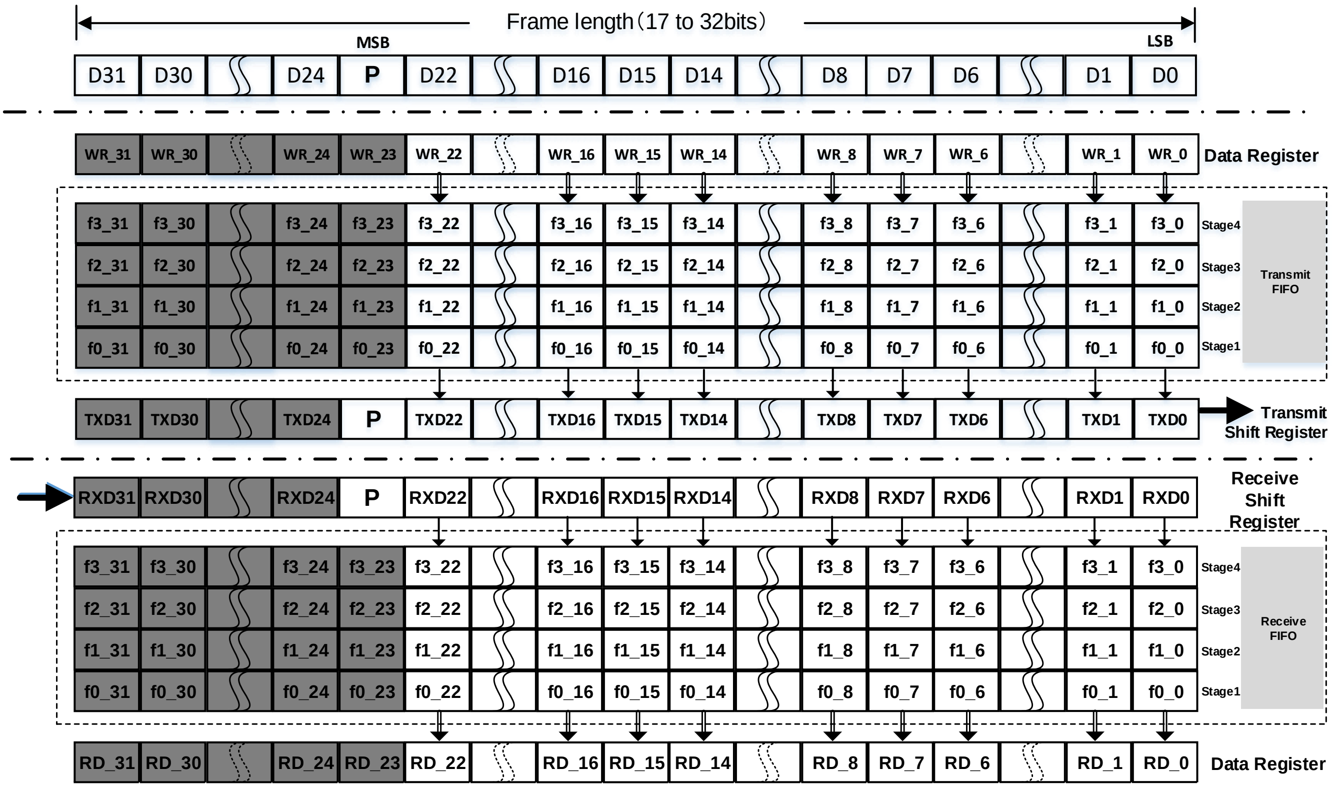

Frame length（17 to 32bits）
MSB
LSB
D31 D30 D24 P D22 D16 D15 D14 D8 D7 D6 D1 D0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data Register
f3_31 f3_30 f3_24 f3_23 f3_22 f3_16 f3_15 f3_14 f3_8 f3_7 f3_6 f3_1 f3_0 Stage4
f2_31 f2_30 f2_24 f2_23 f2_22 f2_16 f2_15 f2_14 f2_8 f2_7 f2_6 f2_1 f2_0 Stage3
f1_31 f1_30 f1_24 f1_23 f1_22 f1_16 f1_15 f1_14 f1_8 f1_7 f1_6 f1_1 f1_0 Stage2
f0_31 f0_30 f0_24 f0_23 f0_22 f0_16 f0_15 f0_14 f0_8 f0_7 f0_6 f0_1 f0_0 Stage1
Transmit FIFO
TXD31 TXD30 TXD24 P TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift Register
RXD31 RXD30 RXD24 P RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift Register
Receive FIFO
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data Register

Figure 3.9 Buffer operation of the frame mode (with parity, LSB first)

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and are stacked to the first level when the transmit FIFO is empty. The data are stacked to the second level when the first level is already filled with data.
Also, when the transmit buffer is empty, the transmit data written after the communication is enabled are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")
Figure 3.9 shows the transmit data from the frames f0 to f3 saved on the transmit FIFO level 1 to 4.
When the state is changed from stopped to enabled communication or when frame transmit is completed, the parity bit is added to MSB of the data in the first level of the transmit FIFO, before it is moved to the transmit shift register and the transmit starts. At the same time, the data on level 2 to 4 are moved to level 1 to 3.
The transmit data in the transmit shift register are transmitted for each serial clock, starting from LSB (f0_0) in the frame f0.

(2) Receive buffer operation

For each frame received during communications, the data in the receive shift register are stacked to the first level when the receive FIFO is empty. The data are stacked to the second level when the first level is already filled with data.
Figure 3.9 shows the receive data from the frames f0 to f3 saved on the receive FIFO level 1 to 4.
The received data are saved in the receive shift register for each serial clock, starting from LSB (f0_0) in the frame f0. Every time a frame is received, the parity bit is removed from MSB and the received data are stacked to the lowest level of the receive FIFO.
From the data register ***[TSPIxDR]***, the CPU or DMAC read out the data on the first level of the receive FIFO. At the same time, the data on level 2 to 4 are moved to level 1 to 3.

3.6.1.6. Transfer cycle for master operations

The number of cycles for frame transfer for master operation is as follows.

Table 3.8 Example of transfer cycles for master operation (frame mode)

No.	Total cycles No.1,2: Burst transfer a+b+c×d+(d-1)×e No.3, 4: Continuous transfer a+b+c+g	Transfer cycle (equivalent of the CS assertion period in the SPI mode operation)					Continuous idle period for transfer g:1 to 15
		Cycle following the CS assertion a:1 to 16	Cycle preceding the CS deassertion b: 1 to 16	Frame length [bit] c:8 to 32	Burst transfer		
					Frame count d:1 to 255	Frame interval period e:0 to 15	
1	10 to 34	1	1	8 to 32	1	0	-
2	5882 to 12002	16	16	8 to 32	255	15	-
3	11 to 35	1	1	8 to 32	-	-	1
4	55 to 79	16	16	8 to 32	-	-	15

The total cycles for burst transfer cycle can be expressed with the following formula.

Total cycles = cycles following CS assertion (a) + cycles preceding CS deassertion (b) + frame length (c)×frame count (d) + (frame count (d) - 1) × frame interval period (e)

For the example in No. 1, when the frame length is 32 bits, there are 1+1+32×1+ (1-1) × 0 = 34 transfer cycles in total.

For the example in No. 2, when the frame length is 8 bits, there are 16+16+8×255+ (255-1) × 15 = 5882 transfer cycles in total.

The total cycles for each frame of a continuous transfer can be expressed with the following formula.

Total cycles = cycle following CS assertion (a) + cycle preceding CS deassertion (b) + frame length (c) + idle period (g)

For the example in No. 3, when the frame length is 32 bits, there are 1+1+32+1 = 35 transfer cycles in total.

For the example in No. 4, when the frame length is 8 bits, there are 16+16+8+15 = 55 transfer cycles in total.

Each transfer time can be expressed as follows.

Transfer time = transfer clock cycle ($1/f_{SCK}$) × total cycles

Also, the cycles following the CS assertion and the ones preceding the CS deassertion are configured in ***[TSPIxFMTR0]***<CSSCKDL[3:0]> <SCKCSDL[3:0]> and the frame interval period for burst transfers with 2 or more frames and the idle period for continuous transfer are specified with ***[TSPIxFMTR0]*** <FINT[3:0]> <CSINT[3:0]>. The period for the generation of each cycle is valid for both the SPI and SIO modes.

3.6.2. Sector mode

In the sector mode, the transfer of a frame that consists of at most 4 sectors are conducted in the continuous transfer mode.

The sectors are called Sector 0 (S0), Sector 1 (S1), Sector 2 (S2), and Sector 3 (S3) in order of the data transfer. In an MSB-first transfer, the MSB side becomes S0. In an LSB-first transfer, the LSB side becomes S0.

The length of each sector is specified with ***[TSPIxSECTCR1]*** <SECTL0[5:0]>/<SECTL1[5:0]>/<SECTL2[5:0]> /<SECTL3[5:0]>.

The data format overview for the sector mode is shown in Figure 3.10.

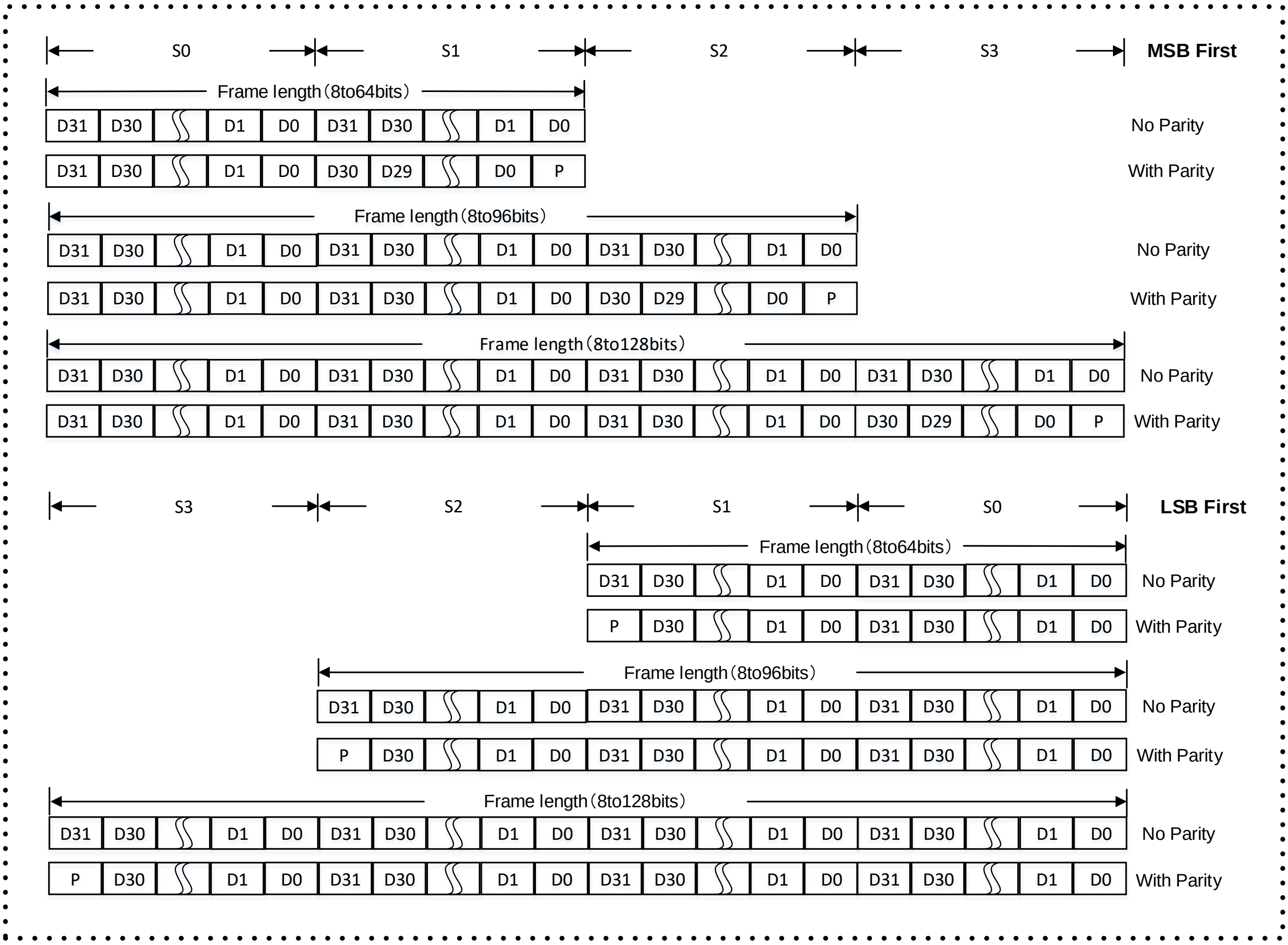

S0
S1
S2
S3
MSB First
Frame length (8to64bits)
D31 D30 D1 D0 D31 D30 D1 D0
No Parity
D31 D30 D1 D0 D30 D29 D0 P
With Parity
Frame length (8to96bits)
D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0
No Parity
D31 D30 D1 D0 D31 D30 D1 D0 D30 D29 D0 P
With Parity
Frame length (8to128bits)
D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0
No Parity
D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0 D30 D29 D0 P
With Parity
S3
S2
S1
S0
LSB First
Frame length (8to64bits)
D31 D30 D1 D0 D31 D30 D1 D0
No Parity
P D30 D1 D0 D31 D30 D1 D0
With Parity
Frame length (8to96bits)
D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0
No Parity
P D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0
With Parity
Frame length (8to128bits)
D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0
No Parity
P D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0 D31 D30 D1 D0
With Parity

Figure 3.10 Data Format Overview for the Sector Mode

No parity

The length of each sector is set to the same value as the sector's data length (1 to 32 bits). when the data length of the sector 0 is 31 bits, the sector length configuration register for Sector 0 is set as ***[TSPIxSECTCR1]***<SECTL0[5:0]>= 011111.

With parity

The sector length of the sectors except the final sector are specified with same values as the data length from 1 to 32 bits. The sector length of the final sector is specified with the sector length from 2 to 32 bits (the data length from 1 to 31 bits and added 1 parity bit). Setting only one bit (only parity bit) of sector length is inhibited. For example, when the data length of the final sector 2 is 31 bits, ***[TSPIxSECTCR1]***<SECTL2[5:0]>=100000 is set to the sector length setting register. Then, set the sector length setting register for unused sector 3 ***[TSPIxSECTCR1]***<SECTL3[5:0]> to 000000.

For transmit, the parity bit is added to the data in transmit FIFO and is moved to the transmit shift register. For receives, the parity bit is removed from the data in the receive shift register and is moved to the receive FIFO.

3.6.2.1. Buffer Structure and Operation

The transmit buffer and the receive buffer are independent of each other. Each buffer has a double buffer structure that consists of the FIFO that is 32-bit wide and has 4 level stages and the shift register with the length of 32 bits and can be accessed in the data register ***[TSPIxDR]***.

In the sector mode, the data are saved and moved with the maximum sector unit of 32 bits.

Table 3.9 shows the configurable ranges of the fill levels in the sector mode.

Table 3.9 Settable fill level in Sector mode

Sector length (S0/S1/S2/S3)	Settable Fill level	
	Transmit FIFO *[TSPIxCR2]*<TIL[3:0]>	Receive FIFO *[TSPIxCR2]*<RIL[3:0]>
1 to 32bit	0 to 3	1 to 4

Note1: Set a value within the settable fill level. If the value is outside of the settable fill level, the operation is not guaranteed.

Note2: Changing the sector length is inhibited when the receive FIFO is full.

Table 3.10 Data format examples for the Sector mode

Condition	With / No parity	Transfer direction	Sector count / Sector length (S0/S1/S2/S3)
Sector mode operation 1	No parity	MSB first	4 sectors (32/32/32/32)
Sector mode operation 2		LSB first	2 sectors (24/16/0/0)
Sector mode operation 3	With parity	MSB first	3 sectors (8/16/32/0)
Sector mode operation 4		LSB first	4 sectors (32/24/16/8)

The conditions for the operation 1 to 4 in "Table 3.10 Data format examples for the Sector mode" are described.

For transmit operations, the process from writing to the data register of the transmit data to outputting the transmit shift register is described. For receive operations, the process from inputting the receive data to the receive shift register to reading out the data register is described.

In the figure, the notation of each data is WR_31 to WR_0 for the write data of the data register ***[TSPIxDR]***, RD_31 to RD_0 for the read data, TXD31 to TXD0 for the transmit shift register data, RXD31 to RXD0 for the receive shift register data, and the frame. The configuration example and sector n data in the FIFO are represented as Sn_31 to Sn_0 (n= 0 to 3 and the maximum sector count is 4), and the parity bit is represented as P.

3.6.2.2. Sector mode operation 1 (No parity, MSB first)

The following figure shows a buffer operation for a transmit No parity, MSB first, 4 sectors (128 bits / frame).

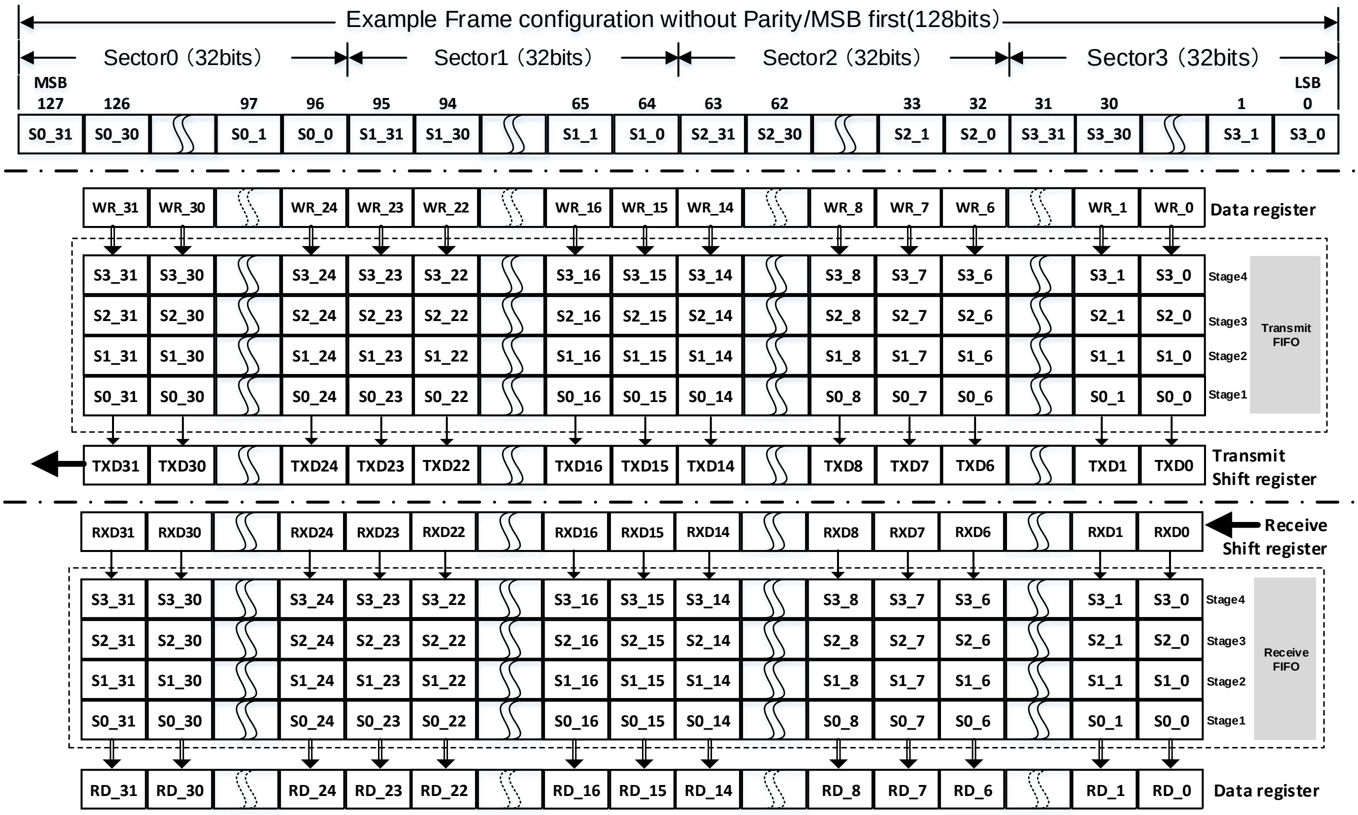

Example Frame configuration without Parity/MSB first(128bits)
Sector0 (32bits)
Sector1 (32bits)
Sector2 (32bits)
Sector3 (32bits)
MSB
LSB
127 126 97 96 95 94 65 64 63 62 33 32 31 30 1 0
S0_31 S0_30 S0_1 S0_0 S1_31 S1_30 S1_1 S1_0 S2_31 S2_30 S2_1 S2_0 S3_31 S3_30 S3_1 S3_0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data register
Stage4
Stage3
Stage2
Stage1
Transmit FIFO
TXD31 TXD30 TXD24 TXD23 TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift register
RXD31 RXD30 RXD24 RXD23 RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift register
Receive FIFO
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data register

Figure 3.11 Buffer operation of the sector mode (No parity, MSB first)

Stop condition: ***[TSPIxSR]*** <TSPISUE>= 0 and ***[TSPIxCR1]*** <TRGEN>= 0, <TRXE>= 0

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and are stacked to the first level stage when the transmit FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.

Also, when the transmit buffer is empty, the transmit data written after the communication is enabled are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")

Figure 3.11 shows the transmit data in the sector 0 to 3 saved on level stages 1 to 4 on the transmit FIFO.

When the operation is enabled to start communications after being stopped or the transmit of a frame has been completed, the data on the first level stage of the transmit FIFO is moved to the transmit shift register and are transmitted. At the same time, the data on level stages 2 to 4 are moved to level stages 1 to 3.

The transmit data in the transmit shift register are transmitted for each serial clock, starting from MSB (S0_31) in the sector 0.

(2) Receive buffer operation

Each time a sector is received during communications, the data in the receive shift register are stacked to the first level stage when the receive FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.

Figure 3.11 shows the data in the sector 0 to 3 saved on level stages 1 to 4 on the receive FIFO.

For each serial clock, the received data are saved in the receive shift register starting from MSB (S0_31) in the sector 0 and are stacked to the lowest level stage on the receive FIFO each time a sector is received.

From the data register ***[TSPIxDR]***, the CPU or DMAC read out the data on the first level stage of the receive FIFO. At the same time, the data on level stages 2 to 4 are moved to level stages 1 to 3.

3.6.2.3. Sector mode operation 2 (No parity, LSB first)

The following figure shows a buffer operation for a transmit No parity, LSB first, 2 sectors (40 bits / frame).

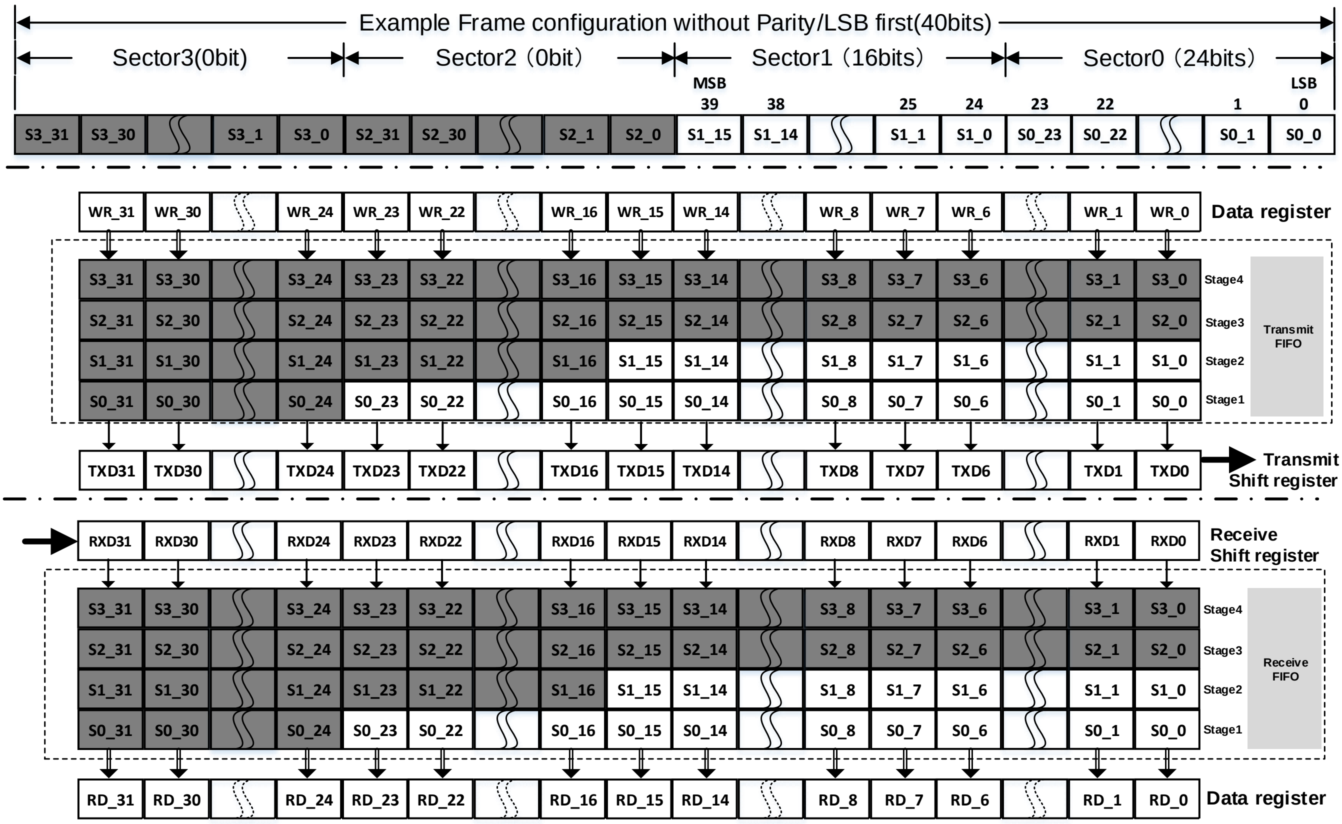

Example Frame configuration without Parity/LSB first(40bits)
Sector3(0bit)
Sector2（0bit）
Sector1（16bits）
Sector0（24bits）
MSB
39
38
25
24
23
22
1
LSB
0
S3_31 S3_30 S3_1 S3_0 S2_31 S2_30 S2_1 S2_0 S1_15 S1_14 S1_1 S1_0 S0_23 S0_22 S0_1 S0_0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data register
S3_31 S3_30 S3_24 S3_23 S3_22 S3_16 S3_15 S3_14 S3_8 S3_7 S3_6 S3_1 S3_0
S2_31 S2_30 S2_24 S2_23 S2_22 S2_16 S2_15 S2_14 S2_8 S2_7 S2_6 S2_1 S2_0
S1_31 S1_30 S1_24 S1_23 S1_22 S1_16 S1_15 S1_14 S1_8 S1_7 S1_6 S1_1 S1_0
S0_31 S0_30 S0_24 S0_23 S0_22 S0_16 S0_15 S0_14 S0_8 S0_7 S0_6 S0_1 S0_0
Stage4
Stage3
Stage2
Stage1
Transmit FIFO
TXD31 TXD30 TXD24 TXD23 TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift register
RXD31 RXD30 RXD24 RXD23 RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift register
S3_31 S3_30 S3_24 S3_23 S3_22 S3_16 S3_15 S3_14 S3_8 S3_7 S3_6 S3_1 S3_0
S2_31 S2_30 S2_24 S2_23 S2_22 S2_16 S2_15 S2_14 S2_8 S2_7 S2_6 S2_1 S2_0
S1_31 S1_30 S1_24 S1_23 S1_22 S1_16 S1_15 S1_14 S1_8 S1_7 S1_6 S1_1 S1_0
S0_31 S0_30 S0_24 S0_23 S0_22 S0_16 S0_15 S0_14 S0_8 S0_7 S0_6 S0_1 S0_0
Stage4
Stage3
Stage2
Stage1
Receive FIFO
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data register

Figure 3.12 Buffer operation of the sector mode (No parity, LSB first)

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and are stacked to the first level stage when the transmit FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Also, when the transmit buffer is empty, the transmit data written after the communication is allowed are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")
Figure 3.12 shows the transmit data in the sector 0 to 1 saved on level stages 1 to 2 on the transmit FIFO.
When the operation is allowed to start communications after being stopped or the transmit of a frame has been completed, the data on the first level stage of the transmit FIFO is moved to the transmit shift register and are transmitted. At the same time, the data on level stage 2 is moved to level stage 1.
The transmit data within the transmit shift register are transmitted for each serial clock, starting from LSB (S0_0) in the sector 0.

(2) Receive buffer operation

Each time a sector is received during communications, the data within the receive shift register are stacked to the first level stage if the receive FIFO is empty. The data are stacked to the second level stage if the first level stage is already filled with data.
Figure 3.12 shows the data within the sectors 0 to 1 saved on level stages 1 to 2 on the receive FIFO.
For each serial clock, the received data are saved within the receive shift register, starting from LSB (S0_0) in the sector 0 and are stacked to the lowest level stage on the receive FIFO each time a sector is received.
From the data register ***[TSPIxDR],*** the CPU or DMAC read out the data on the first level stage of the receive FIFO. At the same time, the data on level stage 2 is moved to level stage 1.

3.6.2.4. Sector mode operation 3 (with parity, MSB first)

The following figure shows a buffer operation for a transmit with parity, MSB first, 3 sectors (56 bits / frame).

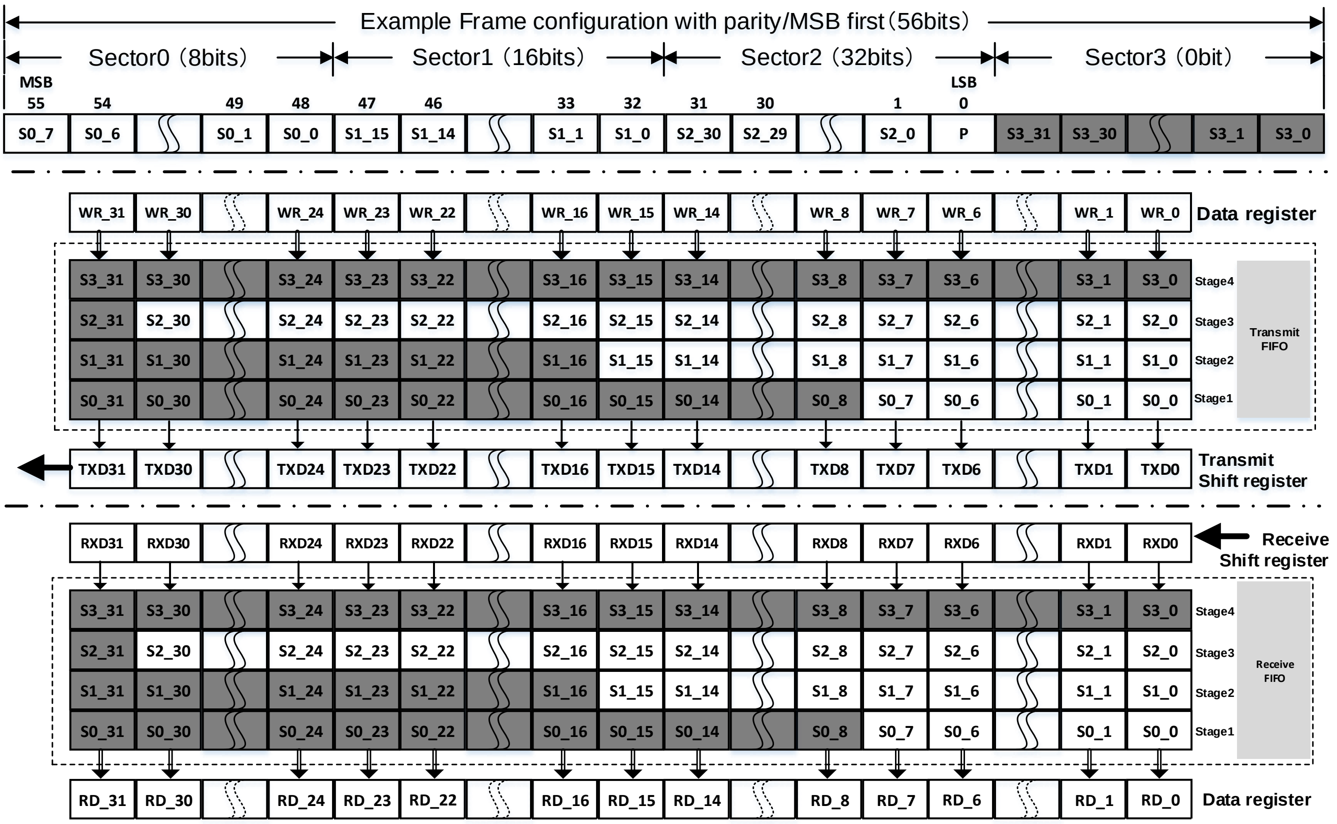

Example Frame configuration with parity/MSB first（56bits）
Sector0（8bits）
Sector1（16bits）
Sector2（32bits）
Sector3（0bit）
MSB
LSB
55 54 49 48 47 46 33 32 31 30 1 0
S0_7 S0_6 S0_1 S0_0 S1_15 S1_14 S1_1 S1_0 S2_30 S2_29 S2_0 P S3_31 S3_30 S3_1 S3_0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data register
S3_31 S3_30 S3_24 S3_23 S3_22 S3_16 S3_15 S3_14 S3_8 S3_7 S3_6 S3_1 S3_0 Stage4
S2_31 S2_30 S2_24 S2_23 S2_22 S2_16 S2_15 S2_14 S2_8 S2_7 S2_6 S2_1 S2_0 Stage3
S1_31 S1_30 S1_24 S1_23 S1_22 S1_16 S1_15 S1_14 S1_8 S1_7 S1_6 S1_1 S1_0 Stage2
S0_31 S0_30 S0_24 S0_23 S0_22 S0_16 S0_15 S0_14 S0_8 S0_7 S0_6 S0_1 S0_0 Stage1
Transmit FIFO
TXD31 TXD30 TXD24 TXD23 TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift register
RXD31 RXD30 RXD24 RXD23 RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift register
S3_31 S3_30 S3_24 S3_23 S3_22 S3_16 S3_15 S3_14 S3_8 S3_7 S3_6 S3_1 S3_0 Stage4
S2_31 S2_30 S2_24 S2_23 S2_22 S2_16 S2_15 S2_14 S2_8 S2_7 S2_6 S2_1 S2_0 Stage3
S1_31 S1_30 S1_24 S1_23 S1_22 S1_16 S1_15 S1_14 S1_8 S1_7 S1_6 S1_1 S1_0 Stage2
S0_31 S0_30 S0_24 S0_23 S0_22 S0_16 S0_15 S0_14 S0_8 S0_7 S0_6 S0_1 S0_0 Stage1
Receive FIFO
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data register

Figure 3.13 Buffer operation of the sector mode (with parity, MSB first)

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and are stacked to the first level stage when the transmit FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Also, when the transmit buffer is empty, the transmit data written after the communication is allowed are immediately transferred to the transmit shift register, and the transmit starts. For slave operations, make sure that an underrun error does not occur. (Refer to "3.7.3. Error Interrupt")
Figure 3.13 shows the transmit data within the sector 0 to 2 saved on level stages 1 to 3 on the transmit FIFO. When the operation is allowed to start communications after being stopped or the transmit of a frame has been completed, the data on the first level stage of the transmit FIFO is moved to the transmit shift register and are transmitted. At the same time, the data on level stages 2 to 3 are moved to level stages 1 to 2. If the first level stage of the transmit FIFO corresponds to the last sector 2, the parity bit is attached to LSB before the data are moved to the transmit shift register and transmitted.
The transmit data within the transmit shift register are transmitted for each serial clock, starting from MSB (S0_7) in the sector 0.

(2) Receive buffer operation

Each time a sector is received during communications, the data within the receive shift register are stacked to the first level stage when the receive FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Figure 3.13 shows the data within the sector 0 to 2 saved on level stages 1 to 3 on the receive FIFO.
For each serial clock, the received data are saved within the receive shift register starting from MSB (S0_7) in the sector 0 and are stacked to the lowest level stage on the receive FIFO each time a sector is received. When the last sector 2 is received, the parity bit is removed from LSB before the data are moved to the receive FIFO.
From the data register ***[TSPIxDR]***, the CPU or DMAC read out the data on the first level stage of the receive FIFO. At the same time, the data on level stages 2 to 3 is moved to level stages 1 to 2.

3.6.2.5. Sector mode operation 4 (with parity, LSB first)

The figure below shows a buffer operation for a transmit with parity, LSB first, 4 sectors (80 bits / frame).

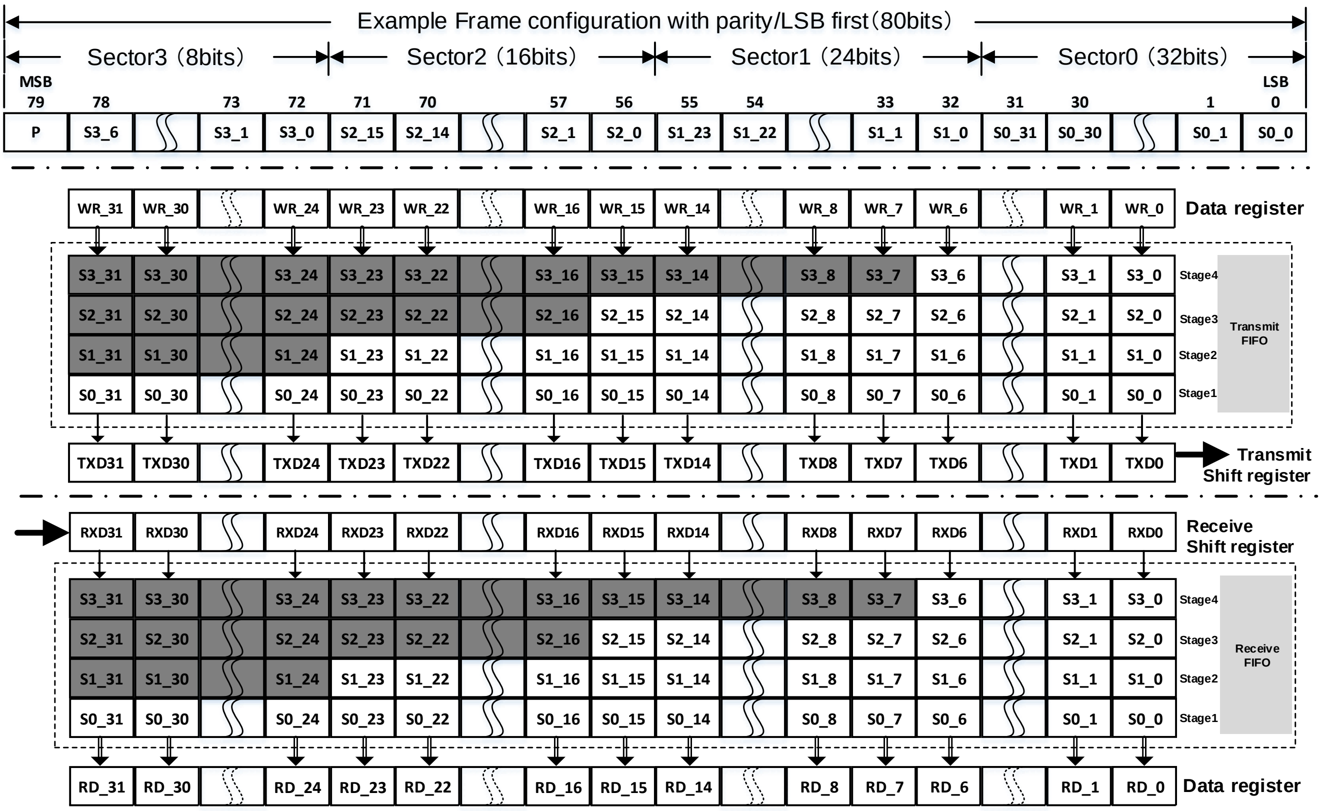

Example Frame configuration with parity/LSB first（80bits）
Sector3（8bits）
Sector2（16bits）
Sector1（24bits）
Sector0（32bits）
MSB
LSB
79 78 73 72 71 70 57 56 55 54 33 32 31 30 1 0
P S3_6 S3_1 S3_0 S2_15 S2_14 S2_1 S2_0 S1_23 S1_22 S1_1 S1_0 S0_31 S0_30 S0_1 S0_0
WR_31 WR_30 WR_24 WR_23 WR_22 WR_16 WR_15 WR_14 WR_8 WR_7 WR_6 WR_1 WR_0
Data register
S3_31 S3_30 S3_24 S3_23 S3_22 S3_16 S3_15 S3_14 S3_8 S3_7 S3_6 S3_1 S3_0 Stage4
S2_31 S2_30 S2_24 S2_23 S2_22 S2_16 S2_15 S2_14 S2_8 S2_7 S2_6 S2_1 S2_0 Stage3
S1_31 S1_30 S1_24 S1_23 S1_22 S1_16 S1_15 S1_14 S1_8 S1_7 S1_6 S1_1 S1_0 Stage2
S0_31 S0_30 S0_24 S0_23 S0_22 S0_16 S0_15 S0_14 S0_8 S0_7 S0_6 S0_1 S0_0 Stage1
Transmit FIFO
TXD31 TXD30 TXD24 TXD23 TXD22 TXD16 TXD15 TXD14 TXD8 TXD7 TXD6 TXD1 TXD0
Transmit Shift register
RXD31 RXD30 RXD24 RXD23 RXD22 RXD16 RXD15 RXD14 RXD8 RXD7 RXD6 RXD1 RXD0
Receive Shift register
S3_31 S3_30 S3_24 S3_23 S3_22 S3_16 S3_15 S3_14 S3_8 S3_7 S3_6 S3_1 S3_0 Stage4
S2_31 S2_30 S2_24 S2_23 S2_22 S2_16 S2_15 S2_14 S2_8 S2_7 S2_6 S2_1 S2_0 Stage3
S1_31 S1_30 S1_24 S1_23 S1_22 S1_16 S1_15 S1_14 S1_8 S1_7 S1_6 S1_1 S1_0 Stage2
S0_31 S0_30 S0_24 S0_23 S0_22 S0_16 S0_15 S0_14 S0_8 S0_7 S0_6 S0_1 S0_0 Stage1
Receive FIFO
RD_31 RD_30 RD_24 RD_23 RD_22 RD_16 RD_15 RD_14 RD_8 RD_7 RD_6 RD_1 RD_0
Data register

Figure 3.14 Buffer operation of the sector mode (with parity, LSB first)

(1) Transmit buffer operation

The transmit data are written by the CPU or DMAC to the data register ***[TSPIxDR]*** when the operation is stopped or during communication, and is stacked to the first level stage when the transmit FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Also, when the transmit buffer is empty, the transmit data written after the communication is allowed are immediately transferred to the transmit shift register, and the transmit will start. Make sure that no underrun errors occur during a slave operation. (Refer to "3.7.3. Error Interrupt")
Figure 3.14 shows the transmit data in the sector 0 to 3 saved on level stages 1 to 4 on the transmit FIFO.
When the operation is allowed to start communications after being stopped or the transmit of a frame has been completed, the data on the first level stage of the transmit FIFO is moved to the transmit shift register and are transmitted. When the first level stage of the transmit FIFO corresponds to the last sector 3, the parity bit is attached to MSB before the data are moved to the transmit shift register and transmitted. At the same time, the data on level stages 2 to 4 is moved to level stages 1 to 3.
The transmit data within the transmit shift register are transmitted for each serial clock, starting from LSB (S0_0) in the sector 0.

(2) Receive buffer operation

Each time a sector is received during communications, the data within the receive shift register are stacked to the first level stage when the receive FIFO is empty. The data are stacked to the second level stage when the first level stage is already filled with data.
Figure 3.14 shows the data within the sector 0 to 3 saved on level stages 1 to 4 on the receive FIFO.
For each serial clock, the received data are saved within the receive shift register, starting from LSB (S0_0) in the sector 0 and are stacked to the lowest level stage on the receive FIFO each time a sector is received. When the last sector 3 is received, the parity bit is removed from MSB before the data are moved to the receive FIFO.
From the data register ***[TSPIxDR]***, the CPU or DMAC will read out the data on the first level of the receive FIFO. At the same time, the data on level stages 2 to 4 is moved to level stages 1 to 3.

3.6.2.6. Transfer cycle for master operations

The operation specification following sector transfer during a master operation is described.

When the sector length of the sector except the final sector is 1 bit, a cycle between sectors occurs after the sector transfer.

Table 3.11 Operation specification following sector transfer during a master operation

Order of sector transfer	Sector (note) [bit]	Operation specification following sector transfer	
Sectors other than the last sector	1	The cycle is generated between sectors and the next sector is transferred after one cycle of the transfer clock ($1/f_{SCK}$) has passed.	
	2 to 32	No cycle is generated between sectors and the next sector is transferred.	
Last sector	1 to 32	Frame transfer complete (Parity bit included if available)	"Parity bit only" is prohibited.
	5 to 32		-
	6 to 32		
	7 to 32		

Note: Corresponds to the bit length of the sector 0/1/2/3 in the data transfer cycle in Table 3.12.

Figure 3.15 shows an example of when a cycle is generated between sectors (sector 1, sector length = 1).
One cycle corresponds to the cycle of a transfer clock TSPIxSCK ($1/f_{SCK}$).

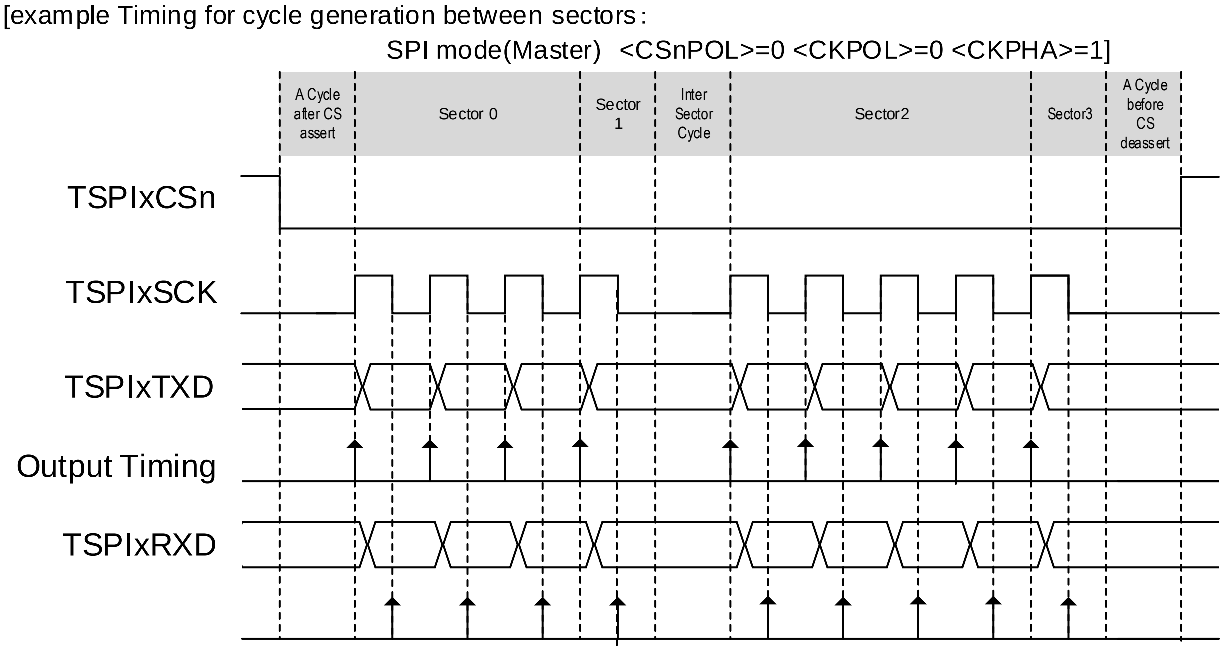

[example Timing for cycle generation between sectors :
SPI mode(Master) <CSnPOL>=0 <CKPOL>=0 <CKPHA>=1]
A Cycle after CS assert
Sector 0
Sector 1
Inter Sector Cycle
Sector2
Sector3
A Cycle before CS deassert
TSPIxCSn
TSPIxSCK
TSPIxTXD
Output Timing
TSPIxRXD

Figure 3.15 Timing for cycle generation between sectors (sector mode, master operation)

The above figure shows the cycle generation timing where the polarity of TSPIxCSn is negative ***[TSPIxFMTR0]***<CSnPOL>= 0, the polarity of TSPIxSCK (levels during the idle period) is "Low" <CKPOL>= 0, the RXD sampling timing is configured to the 2nd edge <CKPHA>= 1, the cycle following the CS assertion and the cycle preceding the CS deassertion cycle is 1, the each sector length is 3 bits (Sector 0), 1 bit (Sector 1), 4 bits (Sector 2), and 1 bit (Sector 3).

Inter-sector cycle is generated after Sector 1 with the sector length of 1 bit is transferred. Although Sector 3 is 1 bit length, inter-sector cycle is not generated because it is the last sector.

TSPIxSCK will not be changed during inter-sector cycles and TSPIxTXD will continue to maintain the data from Sector 1.

(TSPIxRXD simulates the data received from TSPIxTXD in the slave mode)

The transfer cycle count for each condition is shown along with sector counts and sector lengths.

Table 3.12 Example of transfer cycles (sector mode, master operation)

No.	Total cycles Continuous transfer: a+b+c+f+g	Transfer cycle (corresponding to the CS assertion period in the SPI mode operation)								Continuous idle period for transfer g:1 to g15
		Cycle following the CS assertion a:1 to 16	Cycle preceding the CS deassertion b: 1 to 16	Frame length c: 8 to 128					Inter-sector cycle f:0 to f3	
				Data transfer cycle [bit]						
				Sector 0 1 to 32	Sector 1 1 to 32	Sector 2 1 to 32	Sector 3 1 to 32			
1	10 to 130	1	1	2 to 32	2 to 32	2 to 32	1 to 32	8 to 128	0	-
2	11 to 100	1	1	2 to 32	2 to 32	1	1 to 32	8 to 97	1	-
				2 to 32	1	2 to 32	1 to 32			
				1	2 to 32	2 to 32	1 to 32			
3	12 to 70	1	1	2 to 32	1	1	1 to 32	8 to 66	2	-
				1	2 to 32	1	1 to 32			
				1	1	2 to 32	1 to 32			
4	43 to 70	16	16	1	1	1	5 to 32	8 to 35	3	-
5	10 to 98	1	1	2 to 32	2 to 32	1 to 32	0	8 to 96	0	-
6	11 to 68	1	1	2 to 32	1	1 to 32	0	8 to 65	1	-
				1	2 to 32	1 to 32	0			
7	12 to 38	1	1	1	1	6 to 32	0	8 to 34	2	-
8	10 to 66	1	1	2 to 32	1 to 32	0	0	8 to 64	0	-
9	11 to 36	1	1	1	7 to 32	0	0	8 to 33	1	-
10	11 to 67	1	1	2 to 32	1 to 32	0	0	8 to 64	0	1
11	58 to 85	16	16	1	1	1	5 to 32	8 to 35	3	15

Upon launching the software, the total cycles that occur from when the transfer is conducted to when the next frame starts can be expressed with the following formula.

Total cycles = cycles following CS assertion (a) + cycles preceding CS deassertion (b) + frame length (c) + inter-sector cycles (f) + idle period (g)

For the example in No. 10, when the frame length is 8 bits, there are 1+1+8+0+1 = 11 transfer cycles in total.

For the example in No. 11, when the frame length is 35 bits, there are 16+16+35+3+15 = 85 transfer cycles in total.

Also, a frame can be transferred by continuously writing in Start <TRXE> = 1 and Stop <TRXE> = 0 within the register.

The total cycle for a single frame transfer is the same as when the Start Communication trigger is activated.

When the Start Communication trigger is activated, the total cycle for transferring one frame can be expressed with the following formula.

Total cycles = cycles following CS assertion (a) + cycles preceding CS deassertion (b) + frame length (c) + inter-sector cycles (f)

For the example in No. 8, when the frame length is 64 bits, there are 1+1+64+0 = 66 transfer cycles in total.

For the example in No. 4, when the frame length is 8 bits, there are 16+16+8+3 = 43 transfer cycles in total.

Each transfer time can be expressed as follows.

Transfer time = transfer clock cycle (1/f_{SCK}) × total cycles

In addition, the cycles following the CS assertion and the ones preceding the CS deassertion are configured in ***[TSPIxFMTR0]***<CSSCKDL[3:0]> <SCKCSDL[3:0]> and the idle period for continuous transfers of 2 or more frames is set in ***[TSPIxFMTR0]***<CSINT[3:0]>. The period for the generation of each cycle is valid for both the SPI and SIO modes.

3.7. Interrupt Request

The TSPI has three types of interrupts: receive interrupt, transmit interrupt and error interrupt.

Each interrupt is an output consisting of some signals related to interrupts. Transmit interrupts and receive interrupts are enabled / disabled for each interrupt factor, and error interrupts can be set to be enabled / disabled at once.

Table 3.13 Interrupt factors and requests

Interrupt request	Interrupt factor	Enable register
Transmit interrupt	Transmit completion interrupt	***[TSPIxCR2]***<INTTXWE>
	Transmit FIFO interrupt	***[TSPIxCR2]***<INTTXFE>
Receive interrupt	Receive completion interrupt	***[TSPIxCR2]***<INTRXWE>
	Receive FIFO interrupt	***[TSPIxCR2]***<INTRXFE>
Error interrupt	Vertical parity error interrupt	***[TSPIxCR2]***<INTERR>
	Overrun error interrupt	
	Underrun error interrupt	
	Communication start trigger error interrupt	

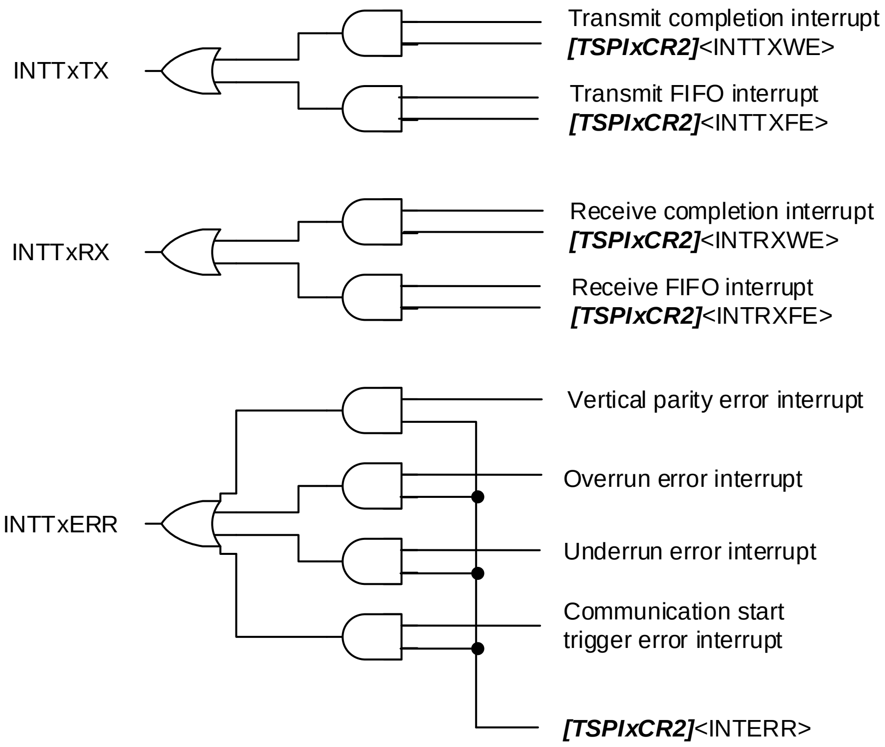


Figure 3.16 Interrupt request Circuit

3.7.1. Transmit Completion Interrupt / Receive Completion Interrupt

- Master device operation

The transmit completion interrupt is generated with the burst transfer and continuous transfer at the timing of which TSPIxCS0/1/2/3 are deasserted at transmit or transmit and receive(full-duplex) communication.
As well, the receive completion interrupt is generated with the burst transfer and continuous transfer at the timing of which TSPIxCS0/1/2/3 are deasserted at receive or transmit and receive(full-duplex) communication.

- Slave device operation

The transmit completion interrupt with the burst transfer is generated at the timing of the final frame transfer end at transmit or transmit and receive(full-duplex) communication.
And in the case of the continuous transfer, that is generated at the timing of one frame transfer end.
As well, the receive end interrupt with the burst transfer is generated at the timing of the final frame transfer end at receive or transmit and receive(full-duplex) communication.
And in the case of the continuous transfer, that is generated at the timing of one frame transfer end.

3.7.2. Transmit FIFO Interrupt / Receive FIFO Interrupt

A transmit FIFO interrupt occurs when the following conditions are met.

a) The data is transferred to the transmit shift register when ***[TSPIxSR]***<TLVL[3:0]>(transmit FIFO fill level status register) is added by 1 to the transmit FIFO interrupt generation condition specified in ***[TSPIxCR2]***<TIL[3:0]>(transmit FIFO Fill level setting register). Then, the transmit FIFO interrupt is generated when the fill level of transmit FIFO is reduced by 1, and <TLVL[3:0]> is changed to the same value of the transmit interrupt generation condition <TIL[3:0]>.

b) In case of <TLVL[3:0]> = <TIL[3:0]>+1
When the data in the FIFO is read into the shift register before transmit starts, then interrupt occurs.

Receive FIFO interrupt occurs when the following conditions are met.

a) The data is transferred to the receive shift register when ***[TSPIxSR]***<RLVL[3:0]>(receive FIFO fill level status register) is reduced by 1 to the receive FIFO interrupt generation condition specified in ***[TSPIxCR2]***<RIL[3:0]>(receive FIFO fill level setting register). Then, the receive FIFO interrupt is generated when the fill level of receive FIFO is added by 1, and <RLVL[3:0]> is changed to the same value of the receive interrupt generation condition <RIL[3:0]>.

3.7.3. Error Interrupt

The following error interrupt is generated. In case, please process appropriately.

(1) Parity error interrupt

The parity error interrupt is generated when the parity error occurs. When the parity is enabled, the parity is calculated automatically by the data received one bit before the final bit of the frame length.

The parity error interrupt is generated when the parity bit received as the final bit of the frame length is compared and not matched. Also, the interrupt generation timing is the time when the receive frame data are stored to the receive FIFO.

(2) Underrun error interrupt and overrun error interrupt

An underrun and overrun errors occur in slave mode.

An underrun error occurs if, after transmit the data in the transmit shift register, the next transfer clock is inputted while the transmit buffer is empty. For the transmitted data for frames where an underrun error has occurred, the output level configured on ***[TSPIxCR2]*** <TXDEMP> (output value control for underrun errors in TSPIxTXD) register is maintained until the end of the frames.

An overrun error occurs if, after receiving the data in the receive shift register, the next transfer clock is inputted while the receive buffer is full. Data in the frame where an overrun occurs is not received. Thus, the contents of the receive FIFO and receive shift register are not updated.

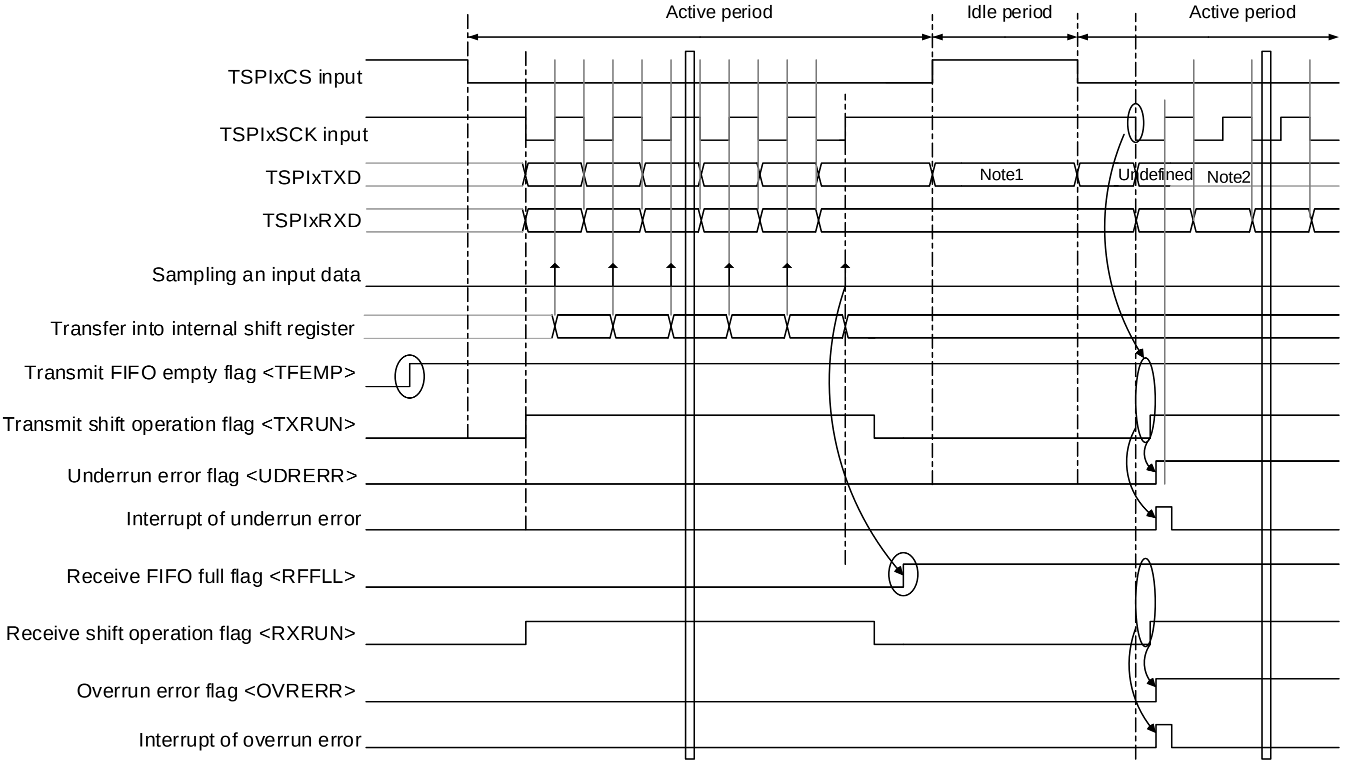

Active period
Idle period
Active period
TSPIxCS input
TSPIxSCK input
TSPIxTXD
Note1
Undefined
Note2
TSPIxRXD
Sampling an input data
Transfer into internal shift register
Transmit FIFO empty flag <TFEMP>
Transmit shift operation flag <TXRUN>
Underrun error flag <UDRERR>
Interrupt of underrun error
Receive FIFO full flag <RFFLL>
Receive shift operation flag <RXRUN>
Overrun error flag <OVRERR>
Interrupt of overrun error

Figure 3.17 Overrun error and underrun error

Note1: It depends on ***[TSPIxCR2]***<TIDLE[1:0]>.

Note2: It depends on ***[TSPIxCR2]***<TXDEMP>.

(3) Communication start trigger error interrupt

Communication start trigger error interrupt is generated in master mode if communication cannot be started with trigger while the start communication trigger is enabled (***[TSPIxCR1]***<TRGEN>= 1). For more information, refer to 3.9.1 " Communication Start Trigger".

3.8. DMA Request

The DMA request has the transmit and receive request. These requests have a single and burst request.

Supported DMA requests depend on the product. Refer to reference manual "Product Information" for details.

3.8.1. Transmit

The single DMA request of transmit and a burst DMA request of transmit will be enabled when ***[TSPIxCR2]***<DMATE> is set to "1".

When FIFO has one or more empty stages, a single request occurs.

A burst transmit DMA request occurs when a value of ***[TSPIxSR]***<TLVL[3:0]> indicating the current value of fill level is equal to or less than transmit interrupt generation condition (fill level) specified in ***[TSPIxCR2]***<TIL[3:0]>. If ***[TSPIxSR]***<TLVL[3:0]> is still equal to or less than the fill level after completion of DMA transfer, a burst transmit DMA request occurs again.

3.8.2. Receive

The single DMA request of receive and a burst DMA request of receiving will be enabled when ***[TSPIxCR2]*** <DMARE> is set to "1".

When FIFO has one or more data, a single request occurs.

A burst receive DMA request occurs when a value of ***[TSPIxSR]***<RLVL[3:0]> indicating the current value of fill level is equal to or greater than receive interrupt generation condition (fill level) specified in ***[TSPIxCR2]***<RIL[3:0]>. If ***[TSPIxSR]***<RLVL[3:0]> is still equal to or greater than ***[TSPIxCR2]***<RIL[3:0]> after completion of DMA transfer, a burst receive DMA request occurs again.

3.9. Trigger Control

3.9.1. Communication Start Trigger

The table below shows the configurable modes that allow for starting communication by activating triggers on peripherals.

Table 3.14 Configurable modes for communication start trigger

Configurable mode			Operation specification for when communication start triggers are provided
Master operation	Frame mode	Burst transfer	Starts transfer, ends after transferring 1 to 255 frame(s)
		Continuous transfer	- (Not available)
	Sector mode		Starts transfer, ends after transferring 1 frame
Slave operation			- (Not available)

Stop condition: ***[TSPIxSR]***<TSPISUE>= 0 and ***[TSPIxCR1]***<TRGEN>= 0, <TRXE>= 0

Waiting for the communication start trigger: ***[TSPIxSR]***<TSPISUE>= 0 and ***[TSPIxCR1]***<TRGEN>= 1, <TRXE>= 0

Communication start trigger activating: ***[TSPIxSR]***<TSPISUE>= 1 and ***[TSPIxCR1]***<TRGEN>= 1, <TRXE>= 1

The peripherals that output Communication start trigger vary depending on the product to be used. For more information, refer to reference manual "Product Information".

The transmit and receive operations in the trigger control are conducted as follows.

(1) Receiving trigger
When the stop condition is confirmed and the condition ***[TSPIxCR1]***<TRGEN>= 1 is configured, the state enters to the "Waiting for the Communication start trigger" state. When the trigger is input and communication is started, the state enters to the "Communication start trigger activating" state. Once the communication is terminated, the state is changed to "Waiting for the communication start trigger." Triggers that are input under the "Communication start trigger activating" state is ignored. Enter the next trigger after the communication is completed.

(2) Transmit operation
When a trigger is provided while the transmit FIFO contains data, the clock and the data are output, and the transmit starts.
In the frame mode, the clock output is stopped, and the transmit is terminated once the number of transfers specified in ***[TSPIxCR1]***<FC[7:0]> has been conducted.
In the sector mode, the clock output is stopped, and the transmit is terminated for each frame that has been transmitted.

(3) Receive operation
When a trigger is provided while the receive FIFO is not full, the clock is output and the receive starts.
In the frame mode, the clock output is stopped, and the receive is terminated once the number of transfers specified in ***[TSPIxCR1]***<FC[7:0]> has been conducted.
In the sector mode, the clock output is stopped, and the receive is terminated for each frame that has been received.

(4) Error for the communication start trigger
When a trigger is provided while the transmit FIFO is empty, the trigger is ignored, the error interrupt is output, and the flag for the error on the Communication start trigger (***[TSPIxERR]***<TRGERR>) is set.
When a trigger is provided while the receive FIFO is full, the trigger is ignored, the error interrupt is output, and the flag for the error on the Communication start trigger is set.
Triggers that are provided while the flag for the error on the Communication start trigger is set are ignored.

3.9.2. Communication Complete Trigger

Using the outputs of the Transmit Complete trigger (TSPIxTXEND) and the Receive Complete trigger (TSPIxRXEND) allows for syncing with other peripherals, such as triggering the timer counter. The synced peripherals vary depending on the product to be used. For more information, refer to reference manual "Product Information".

3.10. Special Control

3.10.1. Polarity of Serial Clock

The polarity of the serial clock TSPIxSCK (SCK) can be selected on ***[TSPIxFMTR0]***<CKPOL> (the polarity register of the idle period for a serial clock).

When ***[TSPIxFMTR0]***<CKPOL>= 0, TSPIxSCK outputs "Low" level signal during the idle period and the first clock edge is a rising edge.

When ***[TSPIxFMTR0]***<CKPOL>= 1, TSPIxSCK outputs "High" level signal during the idle period and the first clock edge is a falling edge.

3.10.2. TSPIxTXD Output during Idle Period

The output level during idle period of the transmit data TSPIxTXD (hereinafter referred as TXD) is selected in ***[TSPIxCR2]***<TIDLE[1:0]>.

Table 3.15 TSPIxTXD Output during Idle Period

[TSPIxCR2]<TIDLE[1:0]>	Output
00	Hi-Z
01	Final transmit data
10	Low
11	High

The TXD output changes to the state shown in Table 3.15 at the timing of ***[TSPIxCR2]***<TIDLE[1:0]> setting. The different conditions from the state shown in Table 3.15 are as follows.

- When the “final transmit data” is set before the first transmit execution, “High” is output.
- At Master operation, even if “final transmit data setting state” is changed from “High” setting or “Low“ setting, each state of “High” or “Low” is held. And it moves to the idle period while the final data is held after the transmit completion.
- At Slave operation, when an underrun error occurs and the value specified in ***[TSPIxCR2]***<TXDEMP> is output, the state is changed to the state shown in Table 3.15 after the transmit completion.
- The final transmit data are held in the frame interval period during the burst transfer.
- At Master operation in SIO mode, an unfixed value is output before one clock of SCK clock cycle of the transmit start.

[TSPIxCR2]<TIDLE[1:0]>= 10 ,***[TSPIxFMTR0]***<CKPHA>= 1: Example of outputting low at idle

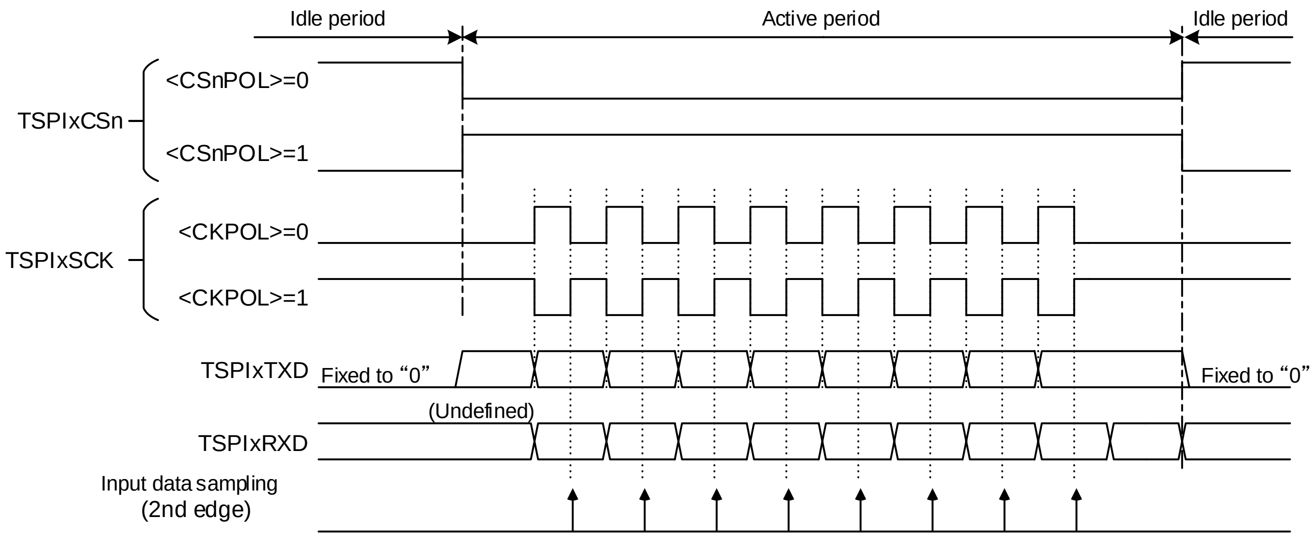


Figure 3.18 Idle period in SPI mode and the transmit pin status

[TSPIxCR2]<TIDLE[1:0]>= 10 , ***[TSPIxFMTR0]***<CKPHA>= 1: Example of outputting Low at idle

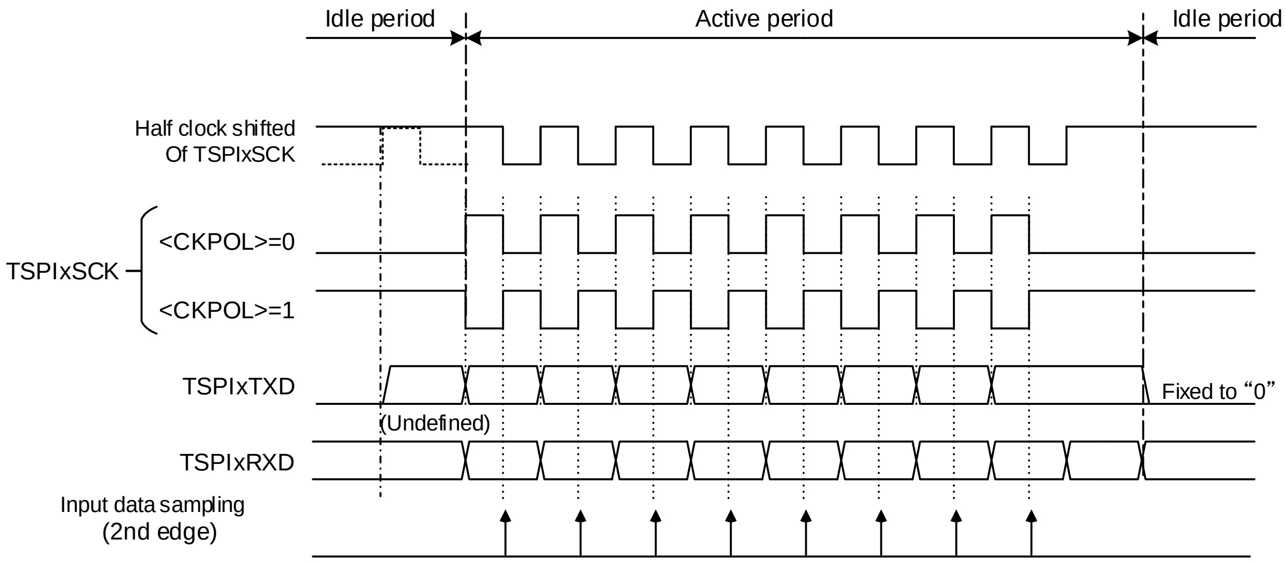


Figure 3.19 Idle period in SIO mode and the transmit pin status

3.10.3. TXD last data hold period

During a slave operation in the SIO mode, the last data hold period for the transmit data TSPIxTXD (TXD) is set to the system clock fsys cycle times 2^n (n=1 to 7) in ***[TSPIxFMTR1]*** <EHOLD[2:0]>.

3.10.4. RXD sampling timing

The sampling timing for the received data TSPIxRXD (RXD) can be switched in ***[TSPIxFMTR0]***<CKPHA>. The sampling is conducted in the 2nd edge and the 1st edge when ***[TSPIxFMTR0]***<CKPHA>= 1 and ***[TSPIxFMTR0]***<CKPHA>= 0, respectively.

Table 3.16 shows various communication modes along with the availabilities of RDX sampling timing.

When the Slave operation / sector mode is selected, the sampling timing of 1st edge is inhibited using.

Table 3.16 Communication modes and the availabilities of the RXD sampling timing

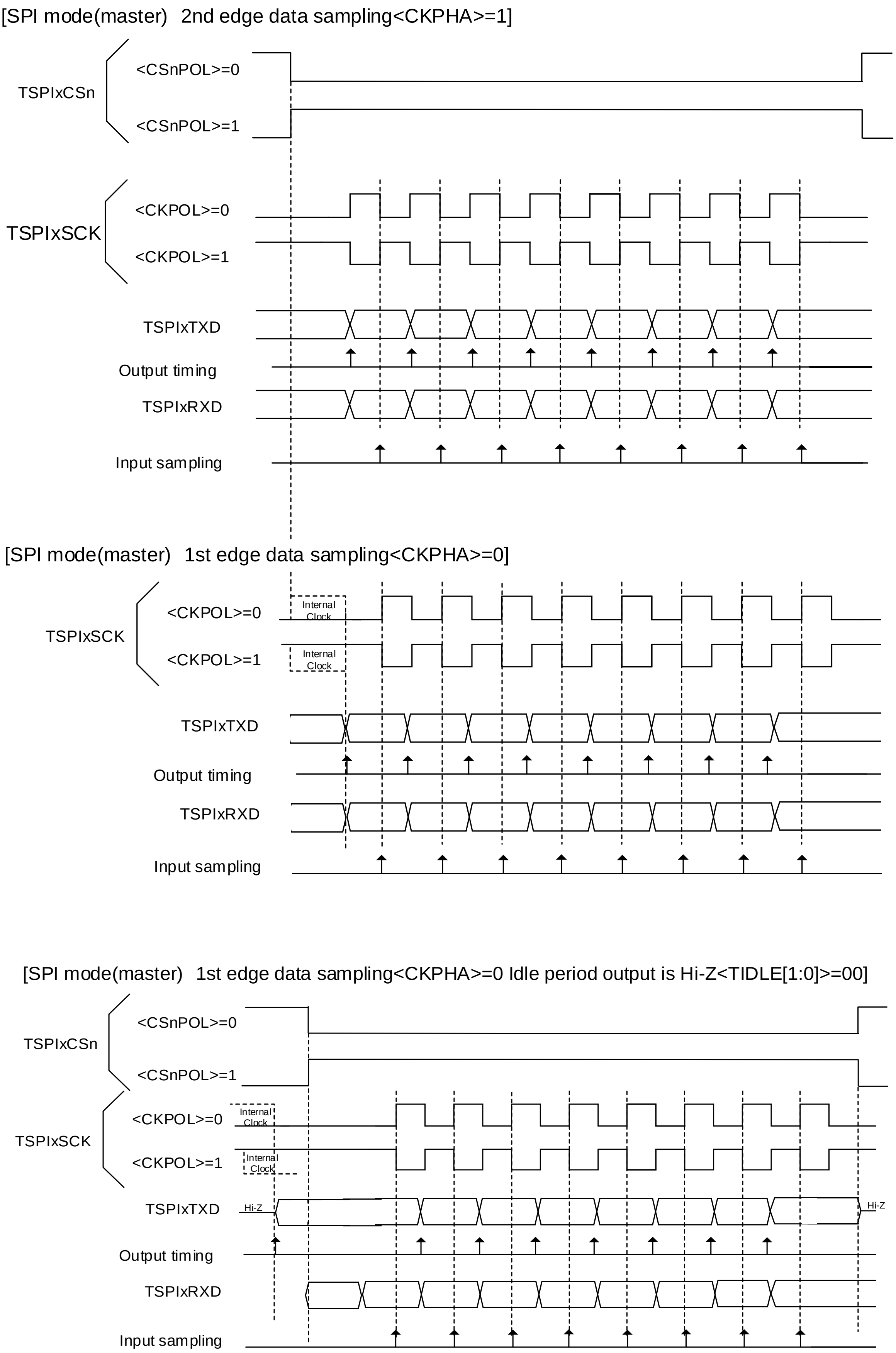
Communication modes			RXD sampling timing	
			1st edge	2nd edge
SPI mode	Master operation	Frame	✓	✓
		Sector	✓	✓
	Slave operation	Frame	✓	✓
		Sector	×	✓
SIO mode	Master operation	Frame	✓	✓
		Sector	✓	✓
	Slave operation	Frame	✓	✓
		Sector	×	✓

✓: Can be used ×: Can not be used

Table 3.17 shows the data capture timing

Table 3.17 Data capture timing

Polarity of idle period of TSPIxSCK *[TSPIxFMTR0]*<CKPOL>	Data capture timing *[TSPIxFMTR0]*<CKPHA>	
	0 (1st edge sampling)	1 (2nd edge sampling)
0 (Polarity of idle period is "Low")	Rising edge	Falling edge
1 (Polarity of idle period is "High")	Falling edge	Rising edge



[SPI mode(master) 2nd edge data sampling<CKPHA>=1]

[SPI mode(master) 1st edge data sampling<CKPHA>=0]

[SPI mode(master) 1st edge data sampling<CKPHA>=0 Idle period output is Hi-Z<TIDLE[1:0]>=00]

Figure 3.20 Data sampling timing of SPI mode (master)

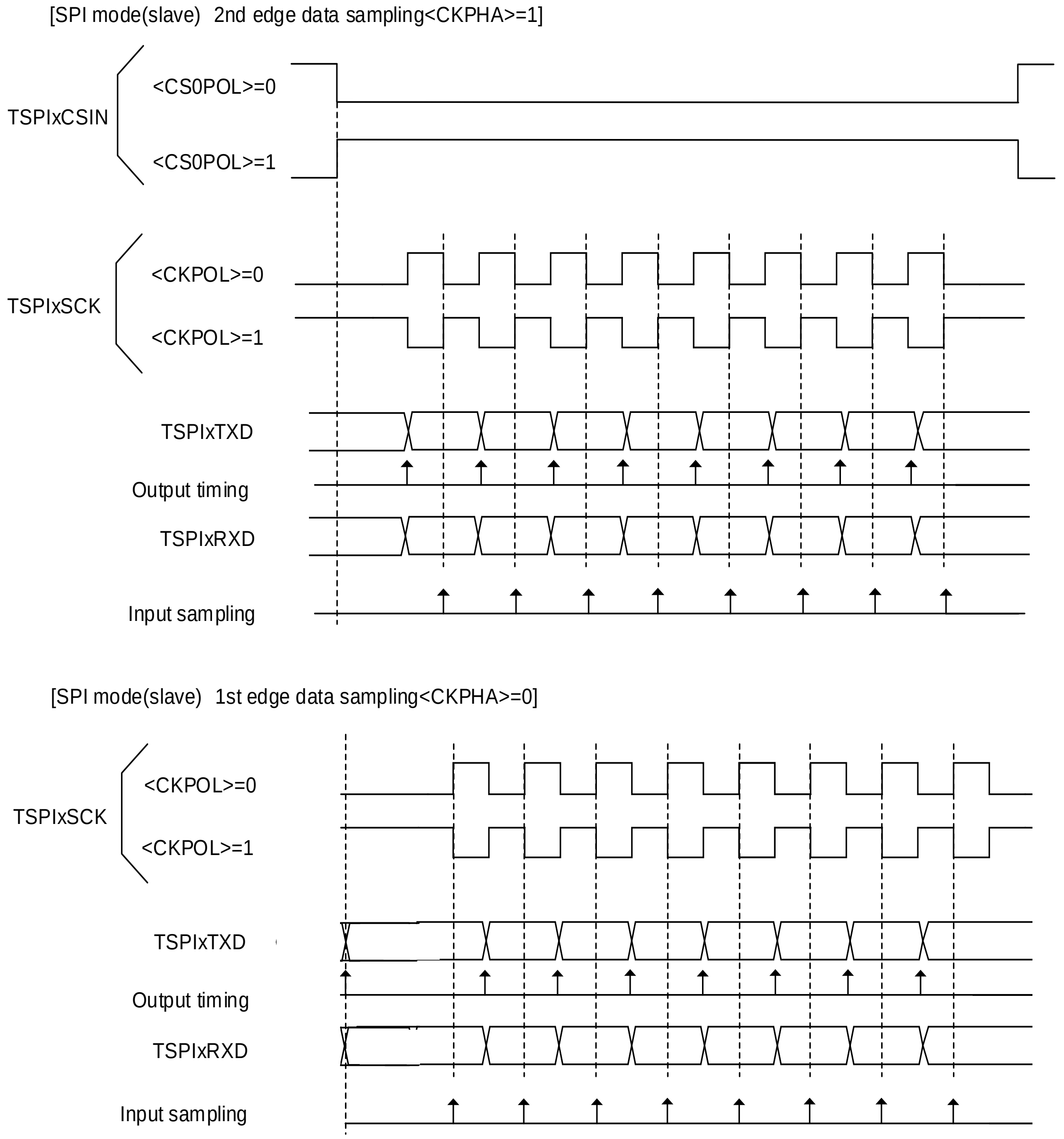
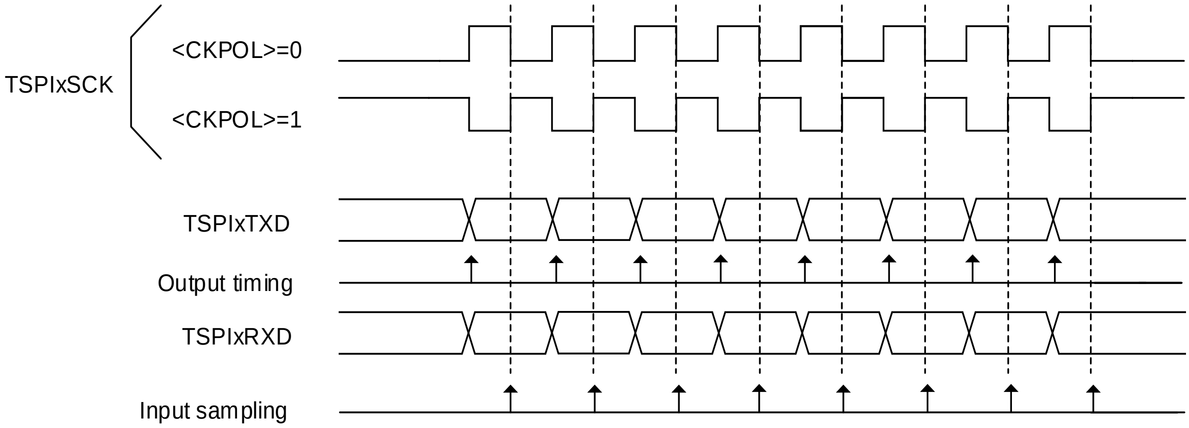


Figure 3.21 Data sampling timing SPI mode (slave)

[SIO mode(master)2nd edge data sampling<CKPHA>=1]



[SIO mode(master)1st edge data sampling<CKPHA>=0]

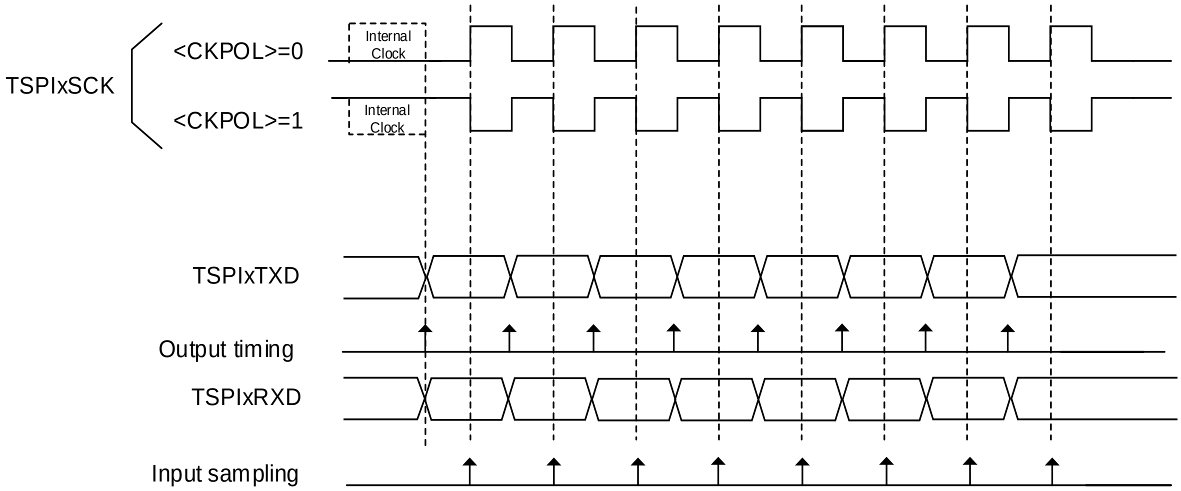


Figure 3.22 Data sampling timing of SIO mode (master)

[SIO mode(slave)2nd edge data sampling<CKPHA>=1]

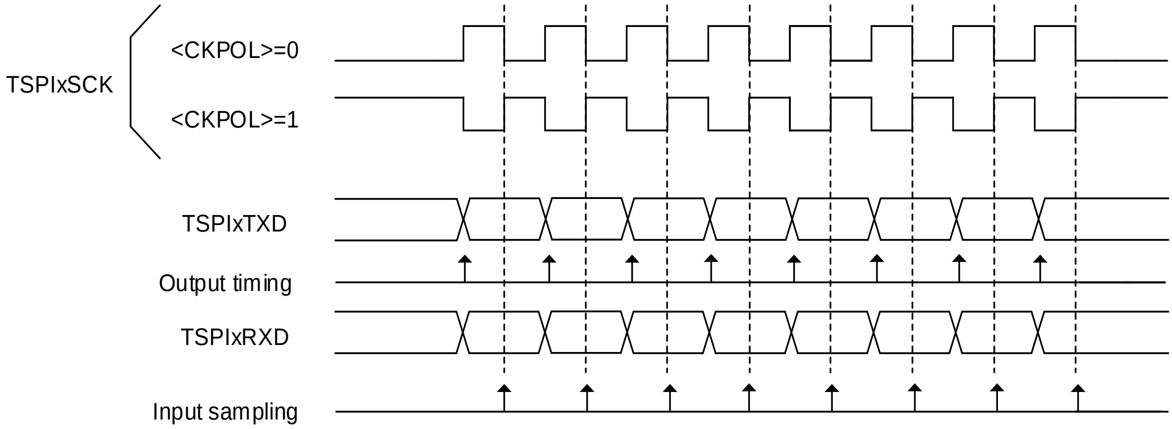


Figure 3.23 Data sampling timing of SIO mode (Slave)

3.10.5. CS timing and polarity

The polarities of the chip select TSPIxCS0/1/2/3 (CS) is selectable with From ***[TSPIxFMTR0]***<CSnPOL> for each pin.

In the case of ***[TSPIxFMTR0]***<CSnPOL>= 0, it becomes negative logic and ***[TSPIxFMTR0]***<CSnPOL>= 1, it becomes positive logic.

Moreover, the generating timing of CS in master mode can be set up as follows.

1. Cycle following the CS assertion (common across Frame / Sector mode and burst / continuous transfer)
   In the SPI mode operation, set the time interval from the CS assertion to the SCK 1st edge (ta) to n (n=1 to 16) times the SCK cycle in ***[TSPIxFMTR0]***<CSSCKDL[3:0]>.

2. Cycle preceding the CS deassertion (Common across frame / sector and burst / continuous transfer)
   In the SPI mode operation, set the time interval from the end of the SCK cycle in the last data to the CS deassertion (tb) to n (n=1 to 16) times the SCK cycle in ***[TSPIxFMTR0]***<SCKCSDL[3:0]>.

3. Frame interval period (common across the SPI / SIO mode)
   In frame mode / burst transfers, set the frame interval period (tc) to n (n=1 to 15) times the SCK cycle in ***[TSPIxFMTR0]***<FINT[3:0]>.

4. Idle period (common across SPI / SIO mode, Frame / Sector mode)
   In continuous transfers, set the idle period (td) to n (n=1 to 15) times the SCK cycle in ***[TSPIxFMTR0]***<CSINT[3:0]>. The idle period corresponds to the CS deassertion period in the SPI mode operation and can be configured in the SIO mode operation.

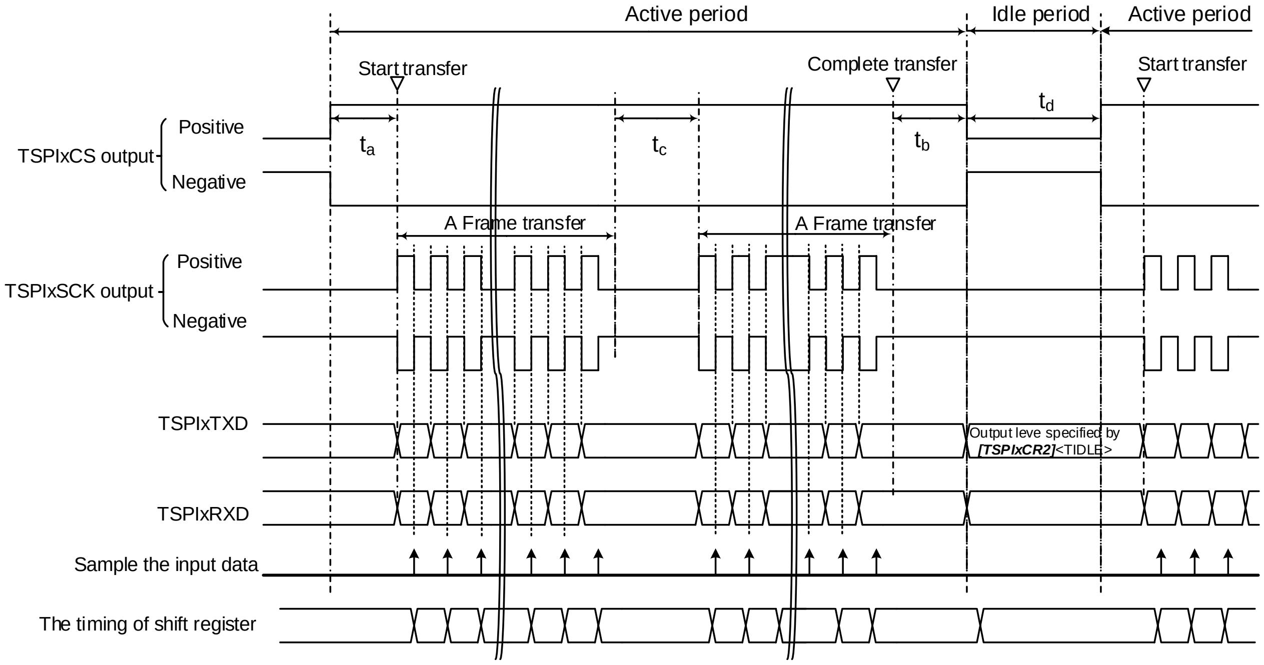


Figure 3.24 Transfer format and timing adjustment (Example for 2nd edge sampling)

3.10.6. Software Reset

TSPI can be reset by software and initialized. Refer to "4.2.1 ***[TSPIxCR0]*** (TSPI Control Register 0)" for details.

4. Registers

4.1. Register List

The following tables show the control registers and addresses.

Peripheral Function		Channel / Unit	Base address		
			TYPE 1	TYPE 2	TYPE 3
Serial Peripheral Interface	TSPI	ch0	0x40098000	0x400CA000	0x4006A000
		ch1	0x40099000	0x400CA400	0x4006A400
		ch2	0x4009A000	0x400CA800	0x4006A800
		ch3	0x4009B000	0x400CAC00	0x4006AC00
		ch4	0x4009C000	0x400CB000	0x4006B000
		ch5	0x4009D000	0x400CB400	0x4006B400
		ch6	0x4009E000	0x400CB800	0x4006B800
		ch7	0x4009F000	0x400CBC00	0x4006BC00
		ch8	0x40096000	0x400CC000	0x4006C000
		ch9	0x40097000	0x400CC400	0x4006C400
		ch10	-	0x400CC800	0x4006C800
		ch11	-	0x400CCC00	0x4006CC00

Note: The channel / unit and base address type are different by products. Refer to reference manual "Product Information" for details.

Register name		Address(Base+)
TSPI Control Register 0	***[TSPIxCR0]***	0x0000
TSPI Control Register 1	***[TSPIxCR1]***	0x0004
TSPI Control Register 2	***[TSPIxCR2]***	0x0008
TSPI Control Register 3	***[TSPIxCR3]***	0x000C
TSPI Baud Rate Register	***[TSPIxBR]***	0x0010
TSPI Format Control Register 0	***[TSPIxFMTR0]***	0x0014
TSPI Format Control Register 1	***[TSPIxFMTR1]***	0x0018
TSPI Sector Mode Control Register 0	***[TSPIxSECTCR0]***	0x001C
TSPI Sector Mode Control Register 1	***[TSPIxSECTCR1]***	0x0020
TSPI Data Register	***[TSPIxDR]***	0x0100
TSPI Status Register	***[TSPIxSR]***	0x0200
TSPI Error Flag Register	***[TSPIxERR]***	0x0204

Note: Registers except ***[TSPIxCR0]***<SWRST[1:0]>, ***[TSPIxCR1]***<TRXE>, ***[TSPIxDR]*** and ***[TSPIxSR]*** cannot be set when ***[TSPIxSR]***<TSPISUE> is "1".

4.2. Detail of Register

4.2.1. *[TSPIxCR0]* (TSPI Control Register 0)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as "0".
7:6	SWRST[1:0]	00	W	TSPI software reset (Note) Software reset occurs by writing "10" and then "01". By software reset, the transfer operation under execution is forcibly terminated and the value of the control register other than the transfer setting is initialized. (Table 4.1)
5:1	-	0	R	Read as "0".
0	TSPIE	0	R/W	TSPI operation control 0: Stop 1: Operation <TSPIE> controls a whole operation of TSPI to start/stop (clock shutdown). When <TSPIE>=0 (stop) is set, a clock is not fed into the internal of the TSPI. Set <TSPIE>=1 (operation) to start operation first. Then perform initialization and communications. <TSPIE> is not initialized by software reset.

Note: Completion of software reset takes two clocks after an instruction is executed. When TSPI setting is stopped (<TSPIE>= 0), software reset is not applied.

To perform a software reset, consecutively write "10" and then "01" to ***[TSPIxCR0]***<SWRST[1:0]>(Software reset register). Software reset will become invalid if other TSPI control registers are accessed in between "10" and "01". Please redo from writing "10".

"Table 4.1 Initialized registers by software reset" shows the initialized register by software reset.

Table 4.1 Initialized registers by software reset

Register name	Symbol name
[TSPIxCR0]	No registers
[TSPIxCR1]	<TRXE>
[TSPIxCR2]	<TIL[3:0]><RIL[3:0]><INTTXFE><INTTXWE><INTRXFE> <INTRXWE><INTERR><DMATE><DMARE>
[TSPIxCR3]	No registers
[TSPIxBR]	No registers
[TSPIxFMTR0]	No registers
[TSPIxFMTR1]	No registers
[TSPIxSECTCR0]	No registers
[TSPIxSECTCR1]	No registers
[TSPIxDR]	No registers
[TSPIxSR]	<TSPISUE><TXRUN><TXEND><INTTXWF><TFEMP> <TLVL[3:0]><RXRUN><RXEND><INTRXFF><RFFLL><RLVL[3:0]>
[TSPIxERR]	<TRGERR><UDRERR><OVRERR><PERR>

4.2.2. *[TSPIxCR1]* (TSPI Control Register 1)

Bit	Bit Symbol	After reset	Type	Function
31:17	-	0	R	Read as "0".
16	INF	0	R/W	Infinite times of transfer control for the burst transfer frames (Note 1) 0: Infinite times of transfer control is disabled. 1: Infinite times of transfer control is enabled.
15	TRGEN	0	R/W	Trigger control (valid only in Master operation) 0: Not used 1: A trigger is valid
14	TRXE	0	R/W	Communication control (Note2)(Note3)(Note4) (Note5)(Note6) (Note7) 0: Communication stops 1: Communication is enabled Set TRXE to "1" after all the settings have been completed.
13	TSPIMS	0	R/W	Communication mode selection 0: SPI mode 1: SIO mode
12	MSTR	1	R/W	Master / slave selection 0: Slave operation 1: Master operation
11:10	TMMD[1:0]	11	R/W	Transfer mode selection 00: Reserved 01: Transmit only 10: Receive only 11: Full-duplex mode (Transmit / receive) When the mode "transmit only" is selected, the processing circuit for TSPIxRXD stops. When the mode "receive only" is selected, the processing circuit for TSPIxTXD stops.
9:8	CSSEL	0	R/W	Selection of TSPIxCS0/1/2/3 (Note8) 00: TSPIxCS0 is valid 01: TSPIxCS1 is valid 10: TSPIxCS2 is valid 11: TSPIxCS3 is valid
7:0	FC[7:0]	0x01	R/W	Sets the number of transfer frames 0: continuous transfer (No limit of transfer times specification) 1 to 255: Burst transfer (1 to 255 times transfer(s))

Note 1: When INF=1, the infinite times of transfer frames for the burst transfer is enabled regardless of FC value. When the serial transfer and the burst transfer (frame count 1-255) are used, INF=0 should be set.
* In the sector mode, (SECT=1), the INF setting has no meaning.

Note 2: In Transmit and receive(Full-duplex communication) / transmit mode, a transmit can be started either by configuring ***[TSPIxCR1]***<TRXE>= 1 (enable communication) while the transmit shift register contains data or by writing the transmit data with <TRXE>= 1. When the configuration of <TRXE>= 0 (stop communication) is conducted during transmit, the configuration of <TRXE>= 0 is applied after the transmit of the frame has been completed.
In receive mode, the receive will start immediately when the configuration of <TRXE>= 1 is conducted. When the configuration of <TRXE>= 0 is conducted during the receive, the configuration is applied after the receive of the frame has been completed.

Note 3: In a burst transmit, the configuration of <TRXE>= 0 is applied after the transfer is completed. When conducting another burst transmit, configure <TRXE>= 1 after verifying that the flag that signifies the configurable state of TSPI has been set as follows: ***[TSPIxSR]***<TSPISUE>= 0 (configurable state).
In continuous transmits, the configuration <TRXE>= 0 is not applied unless it is explicitly set.

Note 4: During slave operation, to stop communication in such a way that the communications on the master side do not start after the configuration of <TRXE>= 1 has been conducted, configure the TSPI software reset option as follows: ***[TSPIxCR0]***<SWRST[1:0]>= 10 → 01 (reset).

Note 5: During slave operation, when the configuration of <TRXE>= 0 has been conducted while the transmit FIFO contained data during the transmit of the data within the transmit shift register, clear the transmit buffer per ***[TSPIxCR3]***<TFEMPCLR>= 1 (clear) or conduct the configuration again after setting <SWRST[1:0]>= 10 → 01.

Note 6: The FIFO data should be written before starting to transmit (<TRXE>=1) in SIO mode slave 1st edge setting. When <TRXE> = 1 is set without writing FIFO data, an underrun will occur.

Note 7: If the transfer ends 1st edge transmit of slave transmit in the SIO mode by setting <TRXE>=0 during transfer, the reset should be performed after the software reset in ***[TSPIxCR0]***<SWRST[1:0]>.

Note 8: Available during master operation in SPI mode. For the SIO mode or during slave operation, do not choose TSPIxCS0/1/2/3 in the settings of the port's functional terminals.

Note 9: When stop communication in slave 1st edge transmit operation, stop communication while the serial clock is inputting from the master device. If stop communication when no clock input from master device, execute software reset.

4.2.3. *[TSPIxCR2]* (TSPI Control Register 2)

Bit	Bit Symbol	After reset	Type	Function
31:24	-	0	R	Read as "0".
23:22	TIDLE[1:0]	11	R/W	Fixed output value function control when TSPIxTXD idles. (Note 1) 00: Hi-Z 01: Last data in the previous transfer 10: Fixed to low 11: Fixed to high
21	TXDEMP	1	R/W	Fixed output value function control when TSPIxTXD underruns (Slave operation). 0: Fixed to low 1: Fixed to high
20:19	-	0	R	Read as "0".
18:16	RXDLY[2:0]	001	R/W	Conditions for transfer clock frequencies f_{SCK} (master operation) 000: fsys / f_{SCK} ≥ 2 100: fsys / f_{SCK} ≥ 10 001: fsys / f_{SCK} ≥ 4 101: fsys / f_{SCK} ≥ 12 010: fsys / f_{SCK} ≥ 6 110: fsys / f_{SCK} ≥ 14 011: fsys / f_{SCK} ≥ 8 111: fsys / f_{SCK} ≥ 16 For more information, refer to "Table 3.2." (Note 2)
15:12	TIL[3:0]	0000	R/W	Transmit fill level setting Transmit FIFO interrupt occurrence condition (Note3)
11:8	RIL[3:0]	0001	R/W	Receive fill level setting Receive FIFO interrupt occurrence condition (Note3)
7	INTTXFE	0	R/W	Transmit FIFO interrupt control 0: Disabled 1: Enabled This is enable bit for generating fill level interrupt of transmit FIFO. Fill level setting is by <TIL[3:0]>.
6	INTTXWE	0	R/W	Transmit completion interrupt control 0: Disabled 1: Enabled An interrupt is generated at the deassertion timing of TSPIxCS 0/1/2/3 when one frame ends during continuous transfer and when burst transfer ends during burst transfer.
5	INTRXFE	0	R/W	Receive FIFO interrupt control 0: Disabled 1: Enabled This is enable bit for generating fill level interrupt of receive FIFO. Fill level setting is by <RIL[3:0]>.

4	INTRXWE	0	R/W	Receive completion interrupt control 0: Disabled 1: Enabled An interrupt is generated at the deassertion timing of TSPIxCS 0/1/2/3 when one frame ends during continuous transfer and when burst transfer ends during burst transfer.
3	-	0	R	Read as "0".
2	INTERR	0	R/W	Error Interrupt control 0: Disabled 1: Enabled Enables interrupt for a receive parity error, a trigger communication error in master operation, and an overrun error and an underrun error in slave operation.
1	DMATE	0	R/W	Transmit DMA control 0: Disabled 1: Enabled When <DMATE> is set to "0" while the transmit DMA request signal is asserted, the request signal is deasserted. It is reasserted if it satisfies the transmit DMA request signal generation requirement when <DMATE> is set to enable again.
0	DMARE	0	R/W	Receive DMA control 0: Disabled 1: Enabled When <DMARE> is set to "0" while the receive DMA request signal is asserted, the request signal is deasserted. It is reasserted if it satisfies the receive DMA request signal generation requirement when <DMARE> is set to enable again.

Note1: When the continuous transfer is performed in the SPI mode Slave operation 1st edge data sampling, each setting is disabled. An unfixed value is output.

Note2: Recommended setting values may be determined depending on the product. Refer to the Serial Peripheral Interface chapter in the Reference Manual "Product Specific Information" and the AC Electrical Characteristics chapter for the Serial Peripheral Interface in the “Data Sheet”.

Note3: Fill level: Refer to Table 3.6 and Table 3.9 for the frame mode and the sector mode, respectively.

4.2.4. *[TSPIxCR3]* (TSPI Control Register 3)

Bit	Bit Symbol	After reset	Type	Function
31:2	-	0	R	Read as "0".
1	TFEMPCLR	0	W	Clears transmit buffer 0: Invalid 1: Clear The transmit FIFO and the transmit shift register are reset to the default settings.
0	RFFLLCLR	0	W	Clears receive buffer 0: Invalid 1: Clear The receive FIFO and the receive shift register are reset to the default settings.

4.2.5. *[TSPIxBR]* (TSPI Baud Rate Register)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as "0".
7:4	BRCK[3:0]	0000	R/W	Input clock selection for baud rate generator. 0000: ΦT0 0101: ΦT16(1/32 ΦT0) 0001: ΦT1(1/2 ΦT0) 0110: ΦT32(1/64 ΦT0) 0010: ΦT2(1/4 ΦT0) 0111: ΦT64(1/128 ΦT0) 0011: ΦT4(1/8 ΦT0) 1000: ΦT128(1/256 ΦT0) 0100: ΦT8(1/16 ΦT0) 1001: ΦT256(1/512 ΦT0) 1010 to 1111: Prohibited
3:0	BRS[3:0]	0000	R/W	Sets a division ratio "N" of baud rate generator. 0000: 16 0110: 6 1100: 12 0001: 1 0111: 7 1101: 13 0010: 2 1000: 8 1110: 14 0011: 3 1001: 9 1111: 15 0100: 4 1010: 10 0101: 5 1011: 11

4.2.6. *[TSPIxFMTR0]* (TSPI Format Control Register 0)

Bit	Bit Symbol	After reset	Type	Function
31	DIR	1	R/W	Transfer direction 0: LSB first 1: MSB first
30	-	0	R	Read as "0".
29:24	FL[5:0]	001000	R/W	Sets a frame length. (Note1) Sets a data length of one frame including a parity bit. 001000: 8 bits 001001: 9 bits : 011111: 31 bits 100000: 32 bits Available on the frame mode operation. The parity bit is included in the frame length. Frame length must be set to a value from 8 to 32 (bit).
23:20	FINT[3:0]	0000	R/W	Interval time between frames in the burst transfer. 0000: 0 (No interval) 0001: 1 x TSPIxSCK cycle 0010: 2 x TSPIxSCK cycles : 1110: 14 x TSPIxSCK cycles 1111: 15 x TSPIxSCK cycles Settings of the frame interval period for a burst transfer during master operation. The frame interval period specified in <FINT> will occur on the SIO mode operation. This setting is disabled at the serial transfer and slave operation.
19	CS3POL	0	R/W	Polarity of TSPIxCS3(Master operation) 0: Negative logic 1: Positive logic
18	CS2POL	0	R/W	Polarity of TSPIxCS2(Master operation) 0: Negative logic 1: Positive logic

17	CS1POL	0	R/W	Polarity of TSPIxCS1(Master operation) 0: Negative logic 1: Positive logic
16	CS0POL	0	R/W	Polarity of TSPIxCS0(Master operation) Polarity of TSPIxCSIN(Slave operation) 0: Negative logic 1: Positive logic
15	CKPHA	1	R/W	Timing of data capture for serial clock 0: Data is sampled on the first edge. 1: Data is sampled on the second edge.
14	CKPOL	1	R/W	Polarity of idle period of serial clock (Note 2) (Note3) 0: TSPIxSCK is "Low" level at idle. 1: TSPIxSCK is "High" level at idle.
13:10	CSINT[3:0]	0001	R/W	Idle time (Note4) TSPIxCS0/1/2/3 invalid→TSPIxCS0/1/2/3 valid time 0000: Prohibited 0001: 1 × TSPIxSCK cycle 0010: 2 × TSPIxSCK cycles : 1110: 14 × TSPIxSCK cycles 1111: 15 × TSPIxSCK cycles Settings of the CS deassertion period for a continuous transfer during master operation. The idle period specified in <CSINT> will occur in the SIO mode operation.
9:8	-	0	R	Read as "0"
7:4	CSSCKDL[3:0]	0000	R/W	Cycle following the CS assertion TSPIxCS0/1/2/3 valid→TSPIxSCK valid time 0000: 1 × TSPIxSCK 1000: 9 × TSPIxSCK 0001: 2 × TSPIxSCK 1001: 10 × TSPIxSCK 0010: 3 × TSPIxSCK 1010: 11 × TSPIxSCK 0011: 4 × TSPIxSCK 1011: 12 × TSPIxSCK 0100: 5 × TSPIxSCK 1100: 13 × TSPIxSCK 0101: 6 × TSPIxSCK 1101: 14 × TSPIxSCK 0110: 7 × TSPIxSCK 1110: 15 × TSPIxSCK 0111: 8 × TSPIxSCK 1111: 16 × TSPIxSCK In master operations, the time period between the assertion of the TSPIxCS0/1/2/3 terminals and the change in the TSPIxSCK terminal is set with the serial clock cycle used as a unit. In the SIO mode operation, <CSSCKDL[3:0]> should be set to ’0000’.
3:0	SCKCSDL[3:0]	0000	R/W	Cycle preceding the CS deassertion 0000: 1 × TSPIxSCK 1000: 9 × TSPIxSCK 0001: 2 × TSPIxSCK 1001: 10 × TSPIxSCK 0010: 3 × TSPIxSCK 1010: 11 × TSPIxSCK 0011: 4 × TSPIxSCK 1011: 12 × TSPIxSCK 0100: 5 × TSPIxSCK 1100: 13 × TSPIxSCK 0101: 6 × TSPIxSCK 1101: 14 × TSPIxSCK 0110: 7 × TSPIxSCK 1110: 15 × TSPIxSCK 0111: 8 × TSPIxSCK 1111: 16 × TSPIxSCK In master operations, the time period required for the deassertion of the TSPIxCS0/1/2/3 terminals from the location of the last data is set with the serial clock cycle used as a unit. In the SIO mode operation, <CSSCKDL[3:0]> should be set to ’0000’.

Note1: All data in the FIFO are discarded if <FL[5:0]> is changed remaining data in the FIFO even if ***[TSPIxSR]***<TSPISUE> is "0".

Note2: Please perform a <CKPOL> setup when transmit / receive is disabled (***[TSPIxCR2]***<TRXE>= 0) at the time of slave operation.

Note3: At the Slave operation in the SIO mode, in the case of using at 1st edge, CS0POL=1 should be set surely.

Note4: When use 1st edge data sampling of master operation,
"Setting value (Integer multiple of TSPIxSCK) + 0.5 × TSPIxSCK".

4.2.7. *[TSPIxFMTR1]* (TSPI Format Control Register 1)

Bit	Bit Symbol	After reset	Type	Function
31:7	-	0	R	Read as "0".
6:4	EHOLD[2:0]	000	R/W	Sets a last bit holding time of TSPIxTXD pin in SIO mode slave operation.(Note1) 000: 2/fc 001: 4/fc 010: 8/fc 011: 16/fc 100: 32/fc 101: 64/fc 110: 128/fc 111: Reserved
3:2	-	0	R	Read as "0".
1	VPE	0	R/W	Vertical parity function (Note2) 0: Disabled 1: Enabled
0	VPM	0	R/W	Vertical parity selection (Note2) 0: Even parity 1: Odd parity

Note1: When the transmit is performed in the SIO mode Slave operation 1st clock edge sampling, and next data is written to the transmit buffer within the final bit hold time, the written data is output to TSPIxTXD.

Note2: Do not write <VPE> and <VPM> when transfer data remain in the shift register.

4.2.8. *[TSPIxSECTCR0]* (TSPI Sector Mode Control Register 0)

Bit	Bit Symbol	After reset	Type	Function
31:1	-	0	R	Read as "0".
0	SECT	0	R/W	Sector / Frame mode selection 0: Frame mode 1: Sector mode Select the sector mode under one of the following conditions. - Frame length of 33 to 128 bits - Frame consisting of 2 to 4 sectors

4.2.9. *[TSPIxSECTCR1]* (TSPI Sector Mode Control Register 1)

Bit	Bit Symbol	After reset	Type	Function
31:30	-	0	R	Read as "0".
29:24	SECTL3[5:0]	000000	R/W	Sector length configuration for sector 3 (Setting prohibited except 0 to 32.) 000000: sector 3 not used 000001 1 bit : 011111: 31 bits 100000: 32 bits when the sector 3 is the last sector and there is a parity bit, add 1 to the data length.
23:22	-	0	R	Read as "0".
21:16	SECTL2[5:0]	000000	R/W	Sector length configuration for sector 2 (Setting prohibited except 0 to 32.) 000000: sector 2 and sector 3 not used (Note 1) 000001: 1 bit : 011111: 31 bits 100000: 32 bits when the sector 2 is the last sector and there is a parity bit, add 1 to the data length.
15:14	-	0	R	Read as "0".
13:8	SECTL1[5:0]	000001	R/W	Sector length configuration for sector 1 (Setting prohibited except 1 to 32.) 000001: 1 bit : 011111: 31 bits 100000: 32 bits when the sector 1 is the last sector and there is a parity bit, add 1 to the data length.
7:6	-	0	R	Read as "0".
5:0	SECTL0[5:0]	000001	R/W	Sector length configuration for sector 0 (Setting prohibited except 1 to 32.) 000001: 1 bit : 011111: 31 bits 100000: 32 bits

Note1: The configuration of <SECTL3[5:0]> is not available,

Note2: Should be used with the frame length (the total sector length) in the range of 8 to 128 bits.

Note3: 1-bit sector length communication in slave mode cannot be set.

Note4: Change the sector length after reading all the received data.

4.2.10. *[TSPIxDR]* (TSPI Data Register)

Bit	Bit Symbol	After reset	Type	Function
31:0	TSPIDR[31:0]	0x00000000	R	Read from the receive FIFO
			W	Write to the transmit FIFO

Note1: Do not write data to this register when the transmit FIFO is full.

Note2: Do not read data from this register when the receive FIFO is empty.

4.2.11. *[TSPIxSR]* (TSPI Status Register)

Bit	Bit Symbol	After reset	Type	Function
31	TSPISUE	0	R	TSPI modify status flag 0: Modification is enabled. 1: Modification is disabled. When <TSPISUE> is "0", the TSPI is not transmitting or receiving, thus the register setting can be modified. <TSPISUE> is "0" in the following conditions: 1. Reset is input. 2. Software reset occurs. 3. In the continuous transfer mode operation, the communication stop setting (***[TSPIxCR1]***<TRXE>=0) is set. And when there is a frame under the transfer, the frame transfer ends. 4. In the burst mode operation, the timing when the specified number of transfers are finished. 5. The current transferring frame is finished by setting ***[TSPIxCR1]***<TRXE>=0 during burst transfer. (refer to the Table 4.2) However, even if the above conditions are satisfied, <TSPISUE> does not become "0" when the receive FIFO or receive shift register is full. To set <TSPISUE>=0, read the receive FIFO and transfer a receive value in the receive shift register to the receive FIFO.
30:24	-	0	R	Read as "0".
23	TXRUN	0	R	Transmit operation flag 0: Stop 1: Operation A status flag indicates the transmit operation is ongoing. <TXRUN> is set when data exists in the transmit shift register even if data does not exist in the transmit FIFO. The state associated with <TXEND><TFEMP> is shown in "Table 4.4."
22	TXEND	0	R	Transmit completion flag A flag that is set at the time when the transmit is complete. 0: - 1: Transmit is complete. TSPIxCS0/1/2/3 of the last frame transfer during continuous transfer and burst transfer is set when it is deasserted. The state associated with <TXRUN><TFEMP> is shown in Table 4.4.
			W	Flag can be cleared by writing "1". 0: Don't care 1: Flag is cleared. When the setting by transmit completion and clearing by writing "1" occur simultaneously, the setting by transmit has higher priority.
21	INTTXWF	0	R	Transmit FIFO interrupt flag This bit is set when remaining data in the transmit FIFO reaches a TIL value from a fill level setting value (TIL) + 1. 0: No interrupt 1: Interrupt occurs
			W	Flag can be cleared by writing "1". 0: Don't care 1: Flag is cleared.

20	TFEMP	1	R	Transmit FIFO empty flag 0: Data exists in the FIFO 1: Empty When the transmit FIFO is empty, this bit is set to "1". This is cleared to "0" once the transmit data is written to the data register. The state associated with <TXRUN><TXEND> is shown in "Table 4.4".
19:16	TLVL[3:0]	0000	R	Transmit FIFO fill level status Indicates the current value of the transmit FIFO fill level (number of data). Stages of the FIFO vary depending on the length of frame. "Table 4.3" shows the display range.
15:8	-	0	R	Read as "0".
7	RXRUN	0	R	Receive operation flag 0: Stop 1: Operation The state associated with <RXEND><RFFLL> is shown in "Table 4.5".
6	RXEND	0	R	Receive completion flag A flag that is set at the time when receive is complete. 0: - 1: Receiving is complete. TSPIxCS0/1/2/3 of the last frame transfer during continuous transfer and burst transfer is set when it is deasserted. The state associated with <RXRUN><RFFLL> is shown in "Table 4.5."
			W	Flag can be cleared by writing "1". 0: Don’t care 1: Flag is cleared. When setting by receive completion and clearing by writing "1" occur simultaneously, setting has higher priority.
5	INTRXFF	0	R	Receive FIFO interrupt flag 0: No interrupt 1: Interrupt occurs This bit is set when remaining data in the receive FIFO reaches a <RIL[3:0]> value from a fill level setting value <RIL[3:0]> - 1.
			W	0: Don’t care 1: Flag is cleared. This bit is cleared by writing "1".
4	RFFLL	0	R	Receive FIFO full flag 0: Not Full 1: Full Indicates that the receive FIFO is full. It is automatically cleared once the data is read out from the data register. (Note) The state associated with <RXRUN><RXEND> is shown in "Table 4.5."
3:0	RLVL[3:0]	0000	R	Receive FIFO fill level status Indicates that the current value of the receive FIFO fill level (number of data). Stages of the FIFO varies depending on the length of the frame. "Table 4.3" shows the display range on <RLVL[3:0]>.

Note: When there is data in the receive shift register, it will be cleared once the data register is read twice.

Table 4.2 Transfer state and setting enabled state flag at communication stop setting

Transfer state of communication stop *[TSPIxCR1]*<TRXE>=0 setting		TSPI setting enabled state flag *[TSPIxSR]*<TSPISUE>	
		Master operation	Slave operation
Before transfer		"0" Setting enabled state	"1" Setting disabled state (stop instruction is disabled.) Software reset should be performed.
During transfer		After the frame during transfer ends, "0" is set.	
Transfer break (Note)	Frame mode	"0" Setting enabled state	"1" Setting disabled state (transfer break state continues.) After the frame ends during transfer, "0" is set.
	Sector mode	"1" Setting disabled state (transfer break continues). After the frame ends during transfer, "0" is set.	
After transfer		"0" Setting enabled state (communication stop state, stop instruction is not needed.)	

Note: Transfer break due to unprocessed transmit / receive data in Master operation, and due to Master in Slave operation.

Table 4.3 Display range of Fill level status

Mode	Frame length / Sector length	FIFO stage	FIFO fill level status range	
			For receive <RLVL[3:0]>	For transmit <TLVL[3:0]>
Frame	8 to 16bits	8 stages	0 to 8	0 to 8
	17 to 32bits	4 stages	0 to 4	0 to 4
Sector	1 to 32bits	4 stages	0 to 4	0 to 4

Table 4.4 Transmit FIFO and the transmit status

<TXRUN>	<TXEND>	<TFEMP>	Status	
0	0	1	(1)	The initial status or the transmit completion flag in (3) cleared
		0	(2)	The transmit data are written, or the transmit completion flag in (4) cleared
	1	1	(3)	Stopped, transmit completed, no transmit FIFO data available (separate shift register available)
		0	(4)	Stopped, transmit completed, transmit FIFO data available
1	0	1	(5)	Transmit in progress, no transmit FIFO data available, transmit of the transmit shift register in progress or, transmit buffer paused due to its empty during master operation
		0	(6)	Transmit in progress, transmit FIFO data available
	1	1/0	-	Transmit completion flag not cleared (not used properly)

Table 4.5 Receive FIFO and the receive status

<RXRUN>	<RXEND>	<RFFLL>	Status	
0	0	0	(7)	The initial status or the receive completion flag in (9) cleared
		1	(8)	The receive completion flag in (10) cleared
	1	0	(9)	Stopped, receive completed, space in receive FIFO available
		1	(10)	Stopped, receive completed, space in receive FIFO unavailable
1	0	0	(11)	Receive in progress, space in receive FIFO available
		1	(12)	Receive in progress, space in receive FIFO unavailable, receive shift register being received Or, receive buffer paused due to its full during master operation (Note)
	1	0/1	-	Receive completion flag not cleared (not used properly)

Note: While the operation is paused because of the receive buffer is full, it can be resumed by reading the data register twice.

4.2.12. *[TSPIxERR]* (TSPI Error Flag Register)

Bit	Bit Symbol	After reset	Type	Function
31:4	-	0	R	Read as "0".
3	TRGERR	0	R	Communication Start trigger error Flag (master operation) This bit is set when communication cannot be started by the trigger input. 0: No error 1: Error
			W	Flag can be cleared by writing "1". 0: Don't care 1: Flag is cleared
2	UDRERR	0	R	Underrun error Flag (Slave operation) This bit is set when an underrun error occurs. 0: No error 1: Error ("Table 4.6" shows the error processing procedure)
			W	Flag can be cleared by writing "1". 0: Don't care 1: Flag is cleared
1	OVRERR	0	R	Overrun error Flag (Slave operation) This bit is set when an overrun error occurs. 0: No error 1: Error ("Table 4.7" shows the error processing procedure)
			W	Flag can be cleared by writing "1". 0: Don't care 1: Flag is cleared
0	PERR	0	R	Vertical parity error flag This bit is set when a vertical parity error occurs. 0: No error 1: Error
			W	Flag can be cleared by writing "1". 0: Don't care 1: Flag is cleared

Note: during communication, do not clear all flag bits.

Table 4.6 Processing an underrun error

Mode	Processing an underrun error
Frame	To resume transmit after an error has occurred, conduct a reconfiguration and re-transmit after the frame with invalid data has been transmitted and the error has been processed. When the data written in the data register after the error has occurred is not used, clear the transmit buffer.
Sector	To resume transmit after an error has occurred, clear the transmit buffer, reconfigure and re-transmit from the frame's sector 0 after the frame with invalid data has been transmitted and the error has been processed. When the master operation stops and the frame transmit cannot be completed after the error, conduct a software reset, before proceeding with reconfiguration and transmit.

Note: When in slave 1st edge operation, stop TSPI after error recovery processing. After that, re-setting and re-start to transmit or receive.

Table 4.7 Processing an overrun error

Mode	Processing an overrun error
Frame	To resume a receive after an error has occurred, conduct a reconfiguration and re-receive after the frame with invalid data has been received and the error has been processed. Note that the error processing includes reading out the data register (2 or more frames need to be read out to resume the receive). To reset the status of the receive FIFO and the receive shift register, clear the receive buffer.
Sector	To resume a receive after an error has occurred, clear the receive buffer, reconfigure and receive data from the frame's sector 0 after the frame with invalid data has been received and the error has been processed. The error processing includes reading out the data register. If the master operation stops and the frame receive cannot be completed after the error, conduct a software reset before proceeding with reconfiguration and receive.

Note: When in slave 1st edge operation, stop TSPI after error recovery processing. After that, re-setting and re-start to transmit or receive.

5. Example for use

5.1. List of Available Operations for Each Mode

The following table shows the list of operations available for each mode.

Also, the reference table of the specifications on starting / stopping transfer for each mode is shown on the "Operation specification" column.

Table 5.1 List of available operations for each mode (Activation method, RXD data sampling)

Communication modes			Communication operation mode	Transfer mode	Activation method	RXD data sampling	Operation specification
SPI SIO	Master operation	Frame	Transmit and receive(full-duplex communication), Transmit, Receive	Continuous	Activate by Software	1st edge 2nd edge	Table 5.3
				Burst			
					Activate by the Start Communication trigger		Table 5.2
		Sector		Continuous			Table 5.5
					Activate by Software		Table 5.6
	Slave operation	Frame		Continuous		1st edge 2nd edge	Table 5.4
				Burst			
		Sector		Continuous		2nd edge	Table 5.7

Note: The following notations are used when describing the operation specifications starting / stopping transfer on the frame mode operation and the sector mode operation.

Stopped: ***[TSPIxSR]*** <TSPISUE>= 0 and ***[TSPIxCR1]*** <TRGEN>= 0, <TRXE>= 0

Software activation: ***[TSPIxSR]*** <TSPISUE>= 1 and ***[TSPIxCR1]*** <TRGEN>= 0, <TRXE>= 1

Waiting for the Start Communication trigger:

[TSPIxSR] <TSPISUE>= 0 and ***[TSPIxCR1]*** <TRGEN>= 1, <TRXE>= 0

The Start Communication trigger activated:

[TSPIxSR] <TSPISUE>= 1 and ***[TSPIxCR1]*** <TRGEN>= 1, <TRXE>= 1

(Master operation)	SCK output: TSPIxSCK output	CS output: TSPIxCS0/1/2/3 output
(Slave operation)	SCK input: TSPIxSCK input	CS input: TSPIxCSIinput

Transmit buffer: Transmit shift register + Transmit FIFO

Receive buffer: Receive shift register + Receive FIFO

When the state is in "Modification is disabled (<TSPISUE>=1)", a read of each register and register setting in the following is possible. The configurations of other registers need to be conducted in "Modification is enabled (<TSPISUE>=0)".

[TSPIxCR0] <SWRST[1:0]> (software reset): Writing "10" and then "01".

[TSPIxCR1] <TRXE> (communication control register):

Writing "0" (stop communication) after reading out "1" (verify if the communication is enabled)

[TSPIxDR] <TSPIDR[31:0]> (data register): Writing transmit data

5.2. Starting / Stopping Transfer in Flame Mode

The operation specifications for starting / stopping transfer in the frame mode operation are as follows.

5.2.1. Activating the Start Communication trigger in master operation

Table 5.2 Starting / Stopping transfer in Frame mode (master operation, activating the Start Communication trigger)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Burst transfer	Transmit and receive(Full-duplex communication) mode	With the configuration (***[TSPIxCR1]***<TRGEN>=1) for the stopped state, the operation moves to the Waiting for the Start Communication Trigger state. When the transmit FIFO has data with 1 or more frames and the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS output and the SCK output with the trigger, and the transfer starts. (Activate the Start Communication trigger) – When the transmit buffer becomes empty after completing a frame transfer, the transfer of the next frame does not start, and the assert state is maintained in the CS output. The transfer is resumed once the data register is written in. – When the receive buffer becomes full after completing a frame transfer, the transfer of the next frame does not start, and the assert state is maintained in the CS output. The transfer is resumed once 2 frames from the data register are read out. The operation moves to the waiting for the Start Communication Trigger state once the configured number of frames is transferred.
	Transmit mode	With the configuration (***[TSPIxCR1]***<TRGEN>=1) for the stopped state, the operation moves to the Waiting for the Start Communication Trigger state. When the transmit FIFO has data with 1 or more frames, the operation is synchronized to the CS output and the SCK output with the trigger, and the transmit starts. (Activate the Start Communication trigger) – When the transmit buffer becomes empty after completing a frame transmit, the transmit of the next frame does not start, and the assert state is maintained in the CS output. The transmit is resumed once the data register is written in. The operation will move to the Waiting for the Start Communication Trigger state once the configured number of frames is transmitted.
	Receive mode	With the configuration (***[TSPIxCR1]***<TRGEN>=1) for the stopped state, the operation moves to the Waiting for the Start Communication Trigger state. When the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS output and the SCK output with the trigger and the receive starts. (Activate the Start Communication trigger) – When the receive buffer becomes full after completing a frame receive, the receive of the next frame does not start, and the assert state is maintained in the CS output. The receive is resumed once 2 frames from the data register are read out. The operation moves to the Waiting for the Start Communication Trigger state once the configured number of frames is received.
Common	Stop	The operation will move to the Waiting for the Start Communication Trigger state once the currently transferred frame is completed with the Start Communication trigger activated (***[TSPIxCR1]***<TRXE>=0). With <TRGEN>=0, the Waiting for the Start Communication Trigger state is released and the state changes to Stopped. – When configuring <TRGEN>=0, make sure that the state is in Waiting for the Start Communication trigger (i.e. Before providing the next trigger following a completed transfer, or after stopping the features for activating triggers and providing the external trigger).
	Resume	Providing the next trigger is invalid (communication cannot be started) unless the configured number of frames have been transferred. The trigger activated after completing the transfer (Stopped state) can be used as a trigger to enable for starting communication.

5.2.2. Software activation in master operation

Table 5.3 Start/Stop transfer in the frame mode (master operation, using software activation)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Burst transfer	Transmit and receive(Full-duplex communication) mode	When the transmit FIFO has data with 1 or more frames and the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state, and the transfer starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transfer, the transfer of the next frame does not start, and the assert state is maintained in the CS output. The transfer is resumed once the data register is written in. – When the receive buffer becomes full after completing a frame transfer, the transfer of the next frame does not start, and the assert state is maintained in the CS output. The transfer is resumed once 2 frames from the data register are read out. The state changes to Stopped once the configured number of frames has been transferred.
	Transmit mode	When the transmit FIFO has data with 1 or more frames, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state, and the transmit starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transmit, the transmit of the next frame does not start, and the assert state is maintained in the CS output. The transmit is resumed once the data register is written in. The state changes to Stopped once the configured number of frames has been transmitted.
	Receive mode	When the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state and the receive starts. (Software activation) – When the receive buffer becomes full after completing a frame receive, the receive of the next frame does not start, and the assert state is maintained in the CS output. The receive is resumed once 2 frames from the data register are read out. The state changes to Stopped once the configured number of frames has been received.
Continuous transfer	Transmit and receive(Full-duplex communication) mode	When the transmit FIFO has data with 1 or more frames and the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state and the transfer starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transfer, the transfer of the next frame does not start, and the deassert state is maintained in the CS output. The transfer is resumed once the data register is written in. – When the receive buffer becomes full after completing a frame transfer, the transfer of the next frame does not start, and the deassert state is maintained in the CS output. The transfer is resumed once 2 frames from the data register are read out.
	Transmit mode	When the transmit FIFO has data with 1 or more frames, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state, and the transmit starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transmit, the transfer of the next frame does not start, and the deassert state is maintained in the CS output. The transmit is resumed once the data register is written in.
	Receive mode	When the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state and the receive starts. (Software activation) – When the receive buffer becomes full after completing a frame receive, the receive of the next frame does not start, and the deassert state is maintained in the CS output. The receive is resumed once 2 frames from the data register are read out.
Common	Stop	The state changes to Stopped once the frame being transmitted has been completed after setting ***[TSPIxCR1]***<TRXE> to "0" at the Software Activation state.

Apart from the methods described in the above tables, the following conditions are also available when starting a transfer.

- Writing to the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the transmit buffer is empty.
- Reading out 2 frames from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive buffer is full.
- Reading out one frame from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive FIFO is full (the shift register is empty).

5.2.3. Software activation in slave operation

Table 5.4 Start/Stop transfer in the frame mode (slave operation, using software activation) (1)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Burst transfer	Transmit and receive(Full-duplex communication) mode	When the transmit FIFO has data with 1 or more frames and the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS input and the SCK input after setting ***[TSPIxCR1]***<TRXE> to "1" at the stopped state, and the transfer starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transfer, starting the transfer of the next frame will result in an underrun error and the configured output (Low/High) is maintained in TXD. The data written to the data register after the transfer of the next frame has been started are transmitted when the frame following the next frame is transferred. – When the receive buffer becomes full after completing a frame transfer, starting the transfer of the next frame will result in an overrun error, and the received data become invalid. When the data register is read out after starting the transfer of the next frame, the data are received in the frame following the next frame. The state changes to Stopped once the configured number of frames has been transferred.
	Transmit mode	When the transmit FIFO has data with 1 or more frames, the operation is synchronized to the CS input and the SCK input after setting ***[TSPIxCR1]***<TRXE> to "1" at the stopped state, and the transmit starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transmit, starting the transmit of the next frame will result in an underrun error, and the configured output (Low/High) is maintained in TXD. The data written to the data register after the transmit of the next frame has been started are transmitted when the frame following the next frame is transmitted. The state changes to Stopped once the configured number of frames has been transmitted.
	Receive mode	When the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS input and the SCK input after setting ***[TSPIxCR1]***<TRXE> to "1" at the stopped state and the receive starts. (Software activation) – When the receive buffer becomes full after completing a frame receive, starting the receive of the next frame results in an overrun error, and the received data become invalid. When the data register is read out after starting the receive of the next frame, the data are received in the frame following the next frame. The state changes to Stopped once the configured number of frames has been received.

Table 5.5 Start/Stop transfer in the frame mode (slave operation, using software activation) (2)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Continuous transfer	Transmit and receive(Full-duplex communication) mode	When the transmit FIFO has data with 1 or more frames and the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS input and the SCK input after setting ***[TSPIxCR1]***<TRXE> to "1" at the stopped state, and the transfer starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transfer, starting the transfer of the next frame will result in an underrun error, and the configured output (Low/High) is maintained in TXD. The data written to the data register after the transfer of the next frame has been started is transmitted when the frame following the next frame is transferred. – When the receive buffer becomes full after completing a frame transfer, starting the transfer of the next frame will result in an overrun error, and the received data becomes invalid. When the data register is read out after starting the transfer of the next frame, the data is received in the frame following the next frame.
	Transmit mode	When the transmit FIFO has data with 1 or more frames, the operation is synchronized to the CS input and the SCK input after setting ***[TSPIxCR1]***<TRXE> to "1" at the stopped state, and the transmit starts. (Software activation) – When the transmit buffer becomes empty after completing a frame transmit, starting the transmit of the next frame will result in an underrun error, and the configured output (Low/High) is maintained in TXD. The data written to the data register after the transmit of the next frame has been started is transmitted when the frame following the next frame is transmitted.
	Receive mode	When the receive FIFO has an empty space with 1 or more frames, the operation is synchronized to the CS input and the SCK input after setting ***[TSPIxCR1]***<TRXE> to "1" at the stopped state, and the receive starts. (Software activation) – When the receive buffer becomes full after completing a frame receive, starting the receive of the next frame will result in an overrun error, and the received data becomes invalid. When the data register is read out after starting the receive of the next frame, the data is received in the frame following the next frame.
Common	Stop	The state changes to Stopped once the frame being transmitted has been completed after setting ***[TSPIxCR1]***<TRXE> to "0" at the Software Activation state.

Apart from the methods described in the above tables, the following conditions are also available when starting a transfer. However, care should be taken when using the following methods as they could result in an underrun / overrun error.

- Writing to the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the transmit buffer is empty.
- Reading out 2 frames from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive buffer is full.
- Reading out one frame from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive FIFO is full (the shift register is empty).

5.3. Starting / Stopping Transfer in Sector Mode

This section describes the operation specifications for starting / stopping a transfer in the sector mode operation.

5.3.1. Activating the Start Communication trigger in master operation

Table 5.6 Starting / Stopping transfer in the sector mode (master operation, activate the Start Communication trigger)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Continuous transfer	Transmit and receive(Full-duplex communication) mode	With the configuration for setting ***[TSPIxCR1]***<TRGEN> to "1" at the stopped state, the operation moves to the Waiting for the Start Communication Trigger state. When the transmit FIFO has data with 1 or more sectors and the receive FIFO has an empty space, the operation is synchronized to the CS output and the SCK output with the trigger, and the transfer starts. (Activate the Start Communication trigger) – When the transmit buffer becomes empty after completing the transfer of the sectors other than the last sector, the transfer of the next sector does not start, and the assert state is maintained in the CS output. The transfer is resumed once data is written to the data register. – When the receive buffer becomes full after completing the transfer of the sectors other than the last sector, the transfer of the next sector does not start, and the assert state is maintained in the CS output. The transfer is resumed once data is read from the data register. The state changes to Waiting for the Start Communication Trigger once one frame is transferred.
	Transmit mode	The state changes to Waiting for the Start Communication Trigger with ***[TSPIxCR1]***<TRGEN>=1 at the Stopped state. When the transmit FIFO has data with 1 or more sectors, the operation is synchronized to the CS output and the SCK output with the trigger, and the transfer starts. (Activate the Start Communication trigger) – When the transmit buffer becomes empty after completing the transmit of the sectors other than the last sector, the transmit of the next sector does not start, and the assert state is maintained in the CS output. The transmit is resumed once data is written to the data register. The state changes to Waiting for the Start Communication Trigger once one frame is transmitted.
	Receive mode	The state changes to Waiting for the Start Communication Trigger with ***[TSPIxCR1]***<TRGEN>=1 at the Stopped state. When the receive FIFO has an empty space, the operation is synchronized to the CS output and the SCK output with the trigger, and the transmit starts. (Activate the Start Communication trigger) – When the receive buffer becomes full after completing the receive of the sectors other than the last sector, the receive of the next sector does not start, and the assert state is maintained in the CS output. The receive is resumed once data is read from the data register. The state changes to Waiting for the Start Communication Trigger once one frame is received.
Common	Stop	The state changes to Waiting for the Start Communication Trigger once one frame for activating the Start Communication trigger has been transferred. It does not make sense to stop while the Start Communication trigger is activated for a transfer of only one frame. With the configuration of ***[TSPIxCR1]***<TRGEN>=0, the Waiting for the Start Communication Trigger state is released and the state changes to Stopped. – When configuring ***[TSPIxCR1]***<TRGEN>=0, make sure that the state is in waiting for the Start Communication trigger (Before providing the next trigger following a completed transfer, or after stopping the function for activating triggers and providing the external trigger).
	Resume	Providing the next trigger is invalid (communication cannot be started) before a transfer is completed (while a frame is transferred). The trigger activated after completing the transfer (Stopped state) can be used as a trigger to enable for starting communication.

5.3.2. Software activation in master operation

Table 5.7 Starting / Stopping transfer in the sector mode (master operation, using software activation)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Continuous transfer	Transmit and receive(Full-duplex communication) mode	When the transmit FIFO has data with 1 or more sectors, and the receive FIFO has an empty space with 1 or more sector, the operation is synchronized to the CS output and the SCK output after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state, and the transfer starts. (Software activation) – When the transmit buffer becomes empty after completing the transfer of the sectors other than the last sector, the transfer of the next sector does not start, and the assert state is maintained in the CS output. The transfer is resumed once data is written to the data register. – When the transmit buffer becomes empty after completing the transfer of the sectors of the last sector, the transfer of the next frame does not start, and the assert state is maintained in the CS output. The transfer is resumed once data is written to the data register. – When the receive buffer becomes full after completing the transfer of the sectors other than the last sector, the transfer of the next sector does not start, and the assert state is maintained in the CS output. The transfer is resumed once data is read from the data register. – When the receive buffer becomes full after completing the transfer of the last sector, the transfer of the next frame does not start, and the assert state is maintained in the CS output. The transfer is resumed once data is read from the data register.
	Transmit mode	When the transmit FIFO has data with 1 or more sectors, the operation is synchronized to the CS output and the SCK output with ***[TSPIxCR1]***<TRXE>=1 at the Stopped state, and the transmit starts. (Software activation) – When the transmit buffer becomes empty after completing the transmit of the sectors other than the last sector, the transmit of the next sector does not start, and the assert state is maintained in the CS output. The transmit is resumed once data is written to the data register. – When the transmit buffer becomes empty after completing the transmit of the last sector, the transmit of the next frame does not start, and the deassert state is maintained in the CS output. The transmit is resumed once data is written to the data register.
	Receive mode	When the receive FIFO has an empty space with 1 or more sectors, the operation is synchronized to the CS output and the SCK output with ***[TSPIxCR1]***<TRXE>=1 at the Stopped state, and the receive starts. (Software activation) – When the receive buffer becomes full after completing the receive of the sectors other than the last sector, the receive of the next sector does not start, and the assert state is maintained in the CS output. The receive is resumed once data is read from the data register. – When the receive buffer becomes full after completing the receive of the last sector, the receive of the next frame does not start, and the deassert state is maintained in the CS output. The receive is resumed once data is read from the data register.
Common	Stop	The state changes to Stopped once the frame being transmitted has been completed after setting ***[TSPIxCR1]***<TRXE> to "0" at the Software Activation state.

Apart from the methods described in the above tables, the following conditions are also available when starting a transfer.

- Writing to the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the transmit buffer is empty.
- Reading out 2 sectors from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive buffer is full.
- Reading out one sector from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive FIFO is full (the shift register is empty).

5.3.3. Software activation in slave operation

Table 5.8 Starting / Stopping transfer in the sector mode (slave operation, using software activation)

Transfer mode	Communication operations	Timing for starting / stopping transfer
Continuous transfer	Transmit and receive(Full-duplex communication) mode	When the transmit FIFO has data with 1 or more sectors and the receive FIFO has an empty space with 1 or more sectors, the operation is synchronized to the CS input and the SCK input with ***[TSPIxCR1]***<TRXE>=1 at the Stopped state, and the transfer starts. (Software activation) – When the transmit buffer becomes empty after completing a sector transfer, starting the transfer of the next sector will result in an underrun error and the configured output (Low/High) is maintained in TXD until the transfer of the frame is completed. The data written to the data register after an underrun error has occurred is invalid. Please clear the FIFO after the frame transfer is completed. – When the receive buffer becomes full after completing a sector transfer, starting the transfer of the next sector will result in an overrun error, and the received data becomes invalid. When the data register is read after the transfer of the next sector has begun, the data is received in the sector following the next sector.
	Transmit mode	When the transmit FIFO has data with 1 or more sectors, the operation is synchronized to the CS input and the SCK input with ***[TSPIxCR1]***<TRXE>=1 at the Stopped state, and the transmit starts. (Software activation) – When the transmit buffer becomes empty after completing a sector transfer, starting the transfer of the next sector will result in an underrun error, and the configured output (Low/High) is maintained in TXD. The data written to the data register after the transfer of the next sector has started is transmitted in the sector following the next sector.
	Receive mode	When the receive FIFO has an empty space with 1 or more sectors, the operation is synchronized to the CS input and the SCK input with ***[TSPIxCR1]***<TRXE>=1 at the Stopped state and the receive starts. (Software activation) – When the receive buffer becomes full after completing a sector transfer, starting the transfer of the next sector will result in an overrun error, and the received data becomes invalid. When the data register is read after the transfer of the next sector has begun, the data is received in the sector following the next sector.
Common	Stop	The state changes to Stopped once the frame being transmitted has been completed after setting ***[TSPIxCR1]***<TRXE> to "0" at the Software Activation state.

Apart from the methods described in the above tables, the following conditions are also available when starting a transfer. However, care should be taken when using the following methods as they could result in an underrun / overrun error.

- Writing to the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the transmit buffer is empty.
- Reading out 2 sectors from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive buffer is full.
- Reading out one sector from the data register after setting ***[TSPIxCR1]***<TRXE> to "1" at the Stopped state while the receive FIFO is full (the shift register is empty).

6. Precautions

- In case of the product which does not have TSPIxCS0/1/2/3 pins or TSPIxCSIN pin, please use SIO mode that is not using these pins.

- Do not access the address that is not assigned register.

7. Revision History

Table 7.1 Revision History

Revision	Date	Description
1.0	2020-11-16	First release
1.1	2021-07-16	Corrected book marks.
1.2	2021-10-15	- 3.2.1.3. Slave Operation Added description. - 3.9.1. Communication Start Trigger (1) Receiving trigger Added description. - 4.2.2. Changed explanations in Note7 and Note8 of ***[TSPIxCR1]*** . - 4.2.6. In function description of ***[TSPIxFMTR0]*** FINT[3:0] and CSINT[3:0], corrected typos.
1.3	2023-03-31	- 3.7.2. Transmit FIFO Interrupt / Receive FIFO Interrupt Changed description

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION
https://toshiba.semicon-storage.com/