

**32-bit RISC Microcontroller**

# **TXZ/TXZ+ Family**

**Reference Manual  
Serial Peripheral Interface  
(TSPI-B)**

**Revision 3.1**

---

**2021-10**

**TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION**

## Table of Contents

Preface .....	5
Related document.....	5
<b>Conventions</b> .....	6
Terms and Abbreviation.....	8
1. Outline.....	9
2. Configuration.....	14
3. Operation description.....	16
3.1. Basic operation .....	16
3.1.1. Clock supply.....	16
3.1.2. Initial setting of TSPI.....	16
3.1.3. Start and stop transfer.....	16
3.2. Data Format.....	17
3.2.1. Data Format without Parity.....	18
3.2.2. Data Format with a Parity.....	22
3.3. Operation .....	26
3.3.1. Transfer clock.....	26
3.3.1.1. Master operation .....	26
3.3.1.2. Slave operation .....	27
3.3.2. Communication mode.....	28
3.3.2.1. SPI mode .....	28
3.3.2.2. SIO mode.....	29
3.3.2.3. Master / Slave selection .....	29
3.3.3. Buffer Structure .....	29
3.3.3.1. Data Length and FIFO Operation .....	30
3.3.4. Communication Operation mode.....	32
3.3.4.1. Full duplex communication mode.....	32
3.3.4.2. Transmit mode .....	34
3.3.4.3. Receive mode .....	35
3.3.5. Transfer mode .....	36
3.3.5.1. Single transfer .....	36
3.3.5.2. Burst transfer.....	36
3.3.5.3. Continuously transfer .....	36
3.3.6. Data sampling timing .....	37
3.3.7. Special control.....	40
3.3.7.1. Polarity of TSPIxCS0/1/2/3 signal and generation timing .....	40
3.3.7.2. Polarity of the Clock .....	41
3.3.7.3. TSPIxTXD Output during Idle.....	41
3.3.8. Communication control by trigger.....	43
3.3.9. Interrupt Request.....	44
3.3.9.1. Transmit Completion Interrupt/Receive Completion Interrupt.....	45
3.3.9.2. Transmit FIFO Interrupt/Receive FIFO Interrupt.....	45

3.3.9.3. Error Interruption .....	45
3.3.10. DMA request .....	47
3.3.10.1. Transmit DMA request .....	47
3.3.10.2. Receive DMA request .....	47
3.3.11. Coordinated movements by the completion of communication .....	47
3.3.12. Software reset .....	47
4. Registers .....	48
4.1. Register List .....	48
4.2. Detail of Register .....	49
4.2.1. [TSPIxCR0] (TSPI Control Register 0) .....	49
4.2.2. [TSPIxCR1] (TSPI Control Register 1) .....	50
4.2.3. [TSPIxCR2] (TSPI Control Register 2) .....	52
4.2.4. [TSPIxCR3] (TSPI Control Register 3) .....	54
4.2.5. [TSPIxBR] (TSPI Baud Rate Register) .....	54
4.2.6. [TSPIxFMTR0] (TSPI Format Control Register 0) .....	55
4.2.7. [TSPIxFMTR1] (TSPI Format Control Register 1) .....	57
4.2.8. [TSPIxDR] (TSPI Data Register) .....	57
4.2.9. [TSPIxSR] (TSPI Status Register) .....	58
4.2.10. [TSPIxERR] (TSPI Error Flag Register) .....	61
5. Example for use .....	62
6. Precautions .....	64
7. Revision History .....	65
RESTRICTIONS ON PRODUCT USE .....	68

## List of Figures

Figure 2.1	Block diagram of TSPI .....	14
Figure 3.1	Data format .....	17
Figure 3.2	MSB first (32-bit data without a parity bit).....	18
Figure 3.3	MSB first (16-bit data without a parity bit).....	19
Figure 3.4	LSB first (32-bit data without a parity bit).....	20
Figure 3.5	LSB first (16-bit data without a parity bit).....	21
Figure 3.6	MSB first (31-bit data with a parity) .....	22
Figure 3.7	MSB first (15-bit data with parity).....	23
Figure 3.8	LSB first (31-bit data with parity).....	24
Figure 3.9	LSB first (15-bit data with parity).....	25
Figure 3.10	Transfer clock generation circuit.....	26
Figure 3.11	Operation in 7 to 16-bit data length .....	30
Figure 3.12	Operation in 17 to 32-bit data length .....	31
Figure 3.13	Operation example of full duplex communication.....	32
Figure 3.14	Operation example of transmit mode .....	34
Figure 3.15	Operation example in receive mode.....	35
Figure 3.16	Data sampling timing of SPI mode (master).....	38
Figure 3.17	Data sampling timing of SPI mode (slave) .....	39
Figure 3.18	Data sampling timing of SIO mode (master) .....	39
Figure 3.19	Data sampling timing of SIO mode (slave).....	39
Figure 3.20	Transfer format and timing adjustment(Example for 2nd edge sampling) .....	40
Figure 3.21	Idle state in SPI mode and the transmit pin status.....	42
Figure 3.22	Idle state in SIO mode and the transmit pin status.....	42
Figure 3.23	Circuit of interrupt request .....	44
Figure 3.24	Overrun error and underrun error .....	46

## List of Tables

Table 1.1	Functional outline (SPI mode, master) .....	10
Table 1.2	Functional outline (SPI mode, slave).....	11
Table 1.3	Functional outline (SIO mode, master).....	12
Table 1.4	Functional outline (SIO mode, slave).....	13
Table 2.1	List of Signals .....	15
Table 3.1	Example of $f_{clk}/\Phi T0/\Phi Tx/$ transfer clock and usability .....	27
Table 3.2	Data format and settable fill level.....	29
Table 3.3	Usability of communication mode and data sampling timing.....	37
Table 3.4	Data capture timing .....	37
Table 3.5	TSPIxTXD output during idle state .....	41
Table 3.6	Interrupt events and requests .....	44
Table 4.1	Initialized registers by software reset.....	49
Table 4.2	The timing of write "0" to <TRXE>, and state of <TSPISUE> .....	60
Table 4.3	Current value of fill level depending on the range of <TLVL>/<RLVL>.....	60
Table 5.1	Transfer starting and stopping operation in each mode setting (Master).....	62
Table 7.1	Revision History .....	65

## 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 four operation mode as SPI/SIO mode, and the clock master/ clock slave mode. One channel/unit which is built-in TSPIxTXD,TSPIxRXD,TSPIxSCK,TSPIxCS0,TSPIxCS1, TSPIxCS2,TSPIxCS3, and TSPIxCSIN can be transmit and receive circuit. functions are as below.

**Table 1.1 Functional outline (SPI mode, master)**

Function classification		Function	A Functional Description or the range
SPI mode (Master)	Transmission Speed Control	Prescaler dividing selection	The clock inputted from the prescaler can be divided 1/1, 1/2, 1/4..up to 1/512.
		Baud rate generator	The input clock to baud rate generator. dividing $1/N \times 1/2$ (N= 1 to 16)
	Data Format	Data length	The data length can be setup in a 1-bit unit. 8 to 32bits (with no parity) 7 to 31bits (with parity)
		Parity	Selection of with parity/without parity is possible. Selection of even parity/odd parity is possible.
		The direction of data transfer	Selection of the LSB first/ MSB first is possible.
	Transmit and Receive Control	FIFO number of section	Transmission: 8 steps (16bits), 4 steps (32bits) Reception: 8 steps (16bits), 4 steps (32bits)
		Communication Operation mode	Full duplexes (transmission and reception), transmission, reception
		Transfer mode	Single transfer (one burst transfer) Burst transfer (2 to 255 times transfer) Continuously transfer (No limit of transfer times specification)
		Data sampling timing	Data is sampled with 1st edge. Data is sampled with 2nd edge.
		CS control	Selection of TSPIxCS0/1/2/3 is possible. Polarity: Selection of positive logic/ negative logic is possible.
	Ganged Control	Interruption	Transmit interrupt (Transmit completion interrupt, Transmit FIFO interrupt) Receive interrupt (Receive completion interrupt, Receive FIFO interrupt) Error Interrupt (Vertical parity error interrupt, Trigger error interrupt)
		Various status detection	TSPI modify status, Transmit shift operation, Transmit completion, Transmit FIFO fill level/ empty detection, Receive operation, Receive completion, Receive FIFO fill level /full detection
		DMA request	Transmit:Single DMA request, Burst DMA request Receive: Single DMA request, Burst DMA request
		Trigger communication control	Starting communication by trigger. Refer to the "Product Information" on a reference manual for a trigger source.
	Special Control	Output level of TSPIxTXD during an idle term	High, Low, a last bit data hold, Hi-Z
		Polarity of TSPIxSCK during an idle term	It is Low to during idle term. It is High to during idle term.
		Interval time between frames at the time of burst transfer.	0 x TSPIxSCK cycle to 15 x TSPIxSCK cycles
		Idle time at the time of continuously transfer	Period of changing TSPIxCS0/1/2/3 pin to asserted, deasserted, asserted. 1 x TSPIxSCK cycle to 15 x TSPIxSCK cycles
		TSPIxSCK delay of deassert	Time of delay between from TSPIxCS0/1/2/3 is asserted to TSPIxSCK is started. 1 x TSPIxSCK cycle - a 16 x TSPIxSCK cycles
		TSPIxCS0/1/2/3 deassertion delay	Time until TSPIxCS0/1/2/3 pin is deasserted from last data 1 x TSPIxSCK cycle to 16 x TSPIxSCK cycle
		Software reset	Reset by software

**Table 1.2 Functional outline (SPI mode, slave)**

Function classification	Function	A Functional Description or the range	
SPI mode (Slave)	Data Format	Data length	The data length can be setup in a 1-bit unit. 8 to 32bits (with no parity) 7 to 31bits (with parity)
		Parity	Selection of with parity/without parity is possible. Selection of even parity/odd parity is possible.
		The direction of data transfer	Selection of the LSB first/ MSB first is possible.
	Transceiver Control	FIFO number of section	Transmission: 8 steps (16bits), 4 steps (32bits) Reception: 8 steps (16bits), 4 steps (32bits)
		Communication Operation mode	Full duplexes (transmission and reception), transmission, reception
		Transfer mode	Single transfer (one burst transfer) Burst transfer (2 to 255 times transfer) Continuously transfer (No limit of transfer times specification)
		Data sampling timing	Data is sampled with 2nd edge.
		CS control	TSPIxCSIN Polarity: Selection of positive logic/ negative logic is possible.
	Ganged Control	Interruption	Transmit interrupt (Transmit completion interrupt, Transmit FIFO interrupt) Receive interrupt (Receive completion interrupt, Receive FIFO interrupt) Error Interrupt (Vertical parity error interrupt, Over run interrupt, Under run interrupt)
		Various status detection	TSPI modify status, Transmit shift operation, Transmit completion, Transmit FIFO fill level/ empty detection, Receive operation, Receive completion, Receive FIFO fill level /full detection
		DMA demand	Transmit: Single DMA request, Burst DMA request Receive: Single DMA request, Burst DMA request
	Special Control	Output level of TSPIxTXD during an idle term	High, Low, a last bit data hold, Hi-Z
		Output level of TSPIxTXD when underrun error occurred	High, Low
		Software reset	Reset by software.

**Table 1.3 Functional outline (SIO mode, master)**

Function classification		Function	A Functional Description or the range
SIO mode (Master)	Transmission speed Control	Prescaler dividing selection	The clock inputted from the prescaler can be divided 1/1, 1/2, 1/4...up to 1/512.
		Baud rate generator	The input clock to baud rate generator. dividing 1/N x 1/2 (N= 1 to 16)
	Data Format	Data length	The data length can be setup in a 1-bit unit. 8 to 32bits (with no parity) 7 to 31bits (with parity)
		Parity	Selection of with parity/without parity is possible. Selection of even parity/odd parity is possible.
		The direction of data transfer	Selection of the LSB first/ MSB first is possible.
	Transceiver Control	FIFO number of section	Transmission: 8 steps (16bits), 4 steps (32bits) Reception: 8 steps (16bits), 4 steps (32bits)
		Communication Operation mode	Full duplexes (transmission and reception), transmission, reception
		Transfer mode	Single transfer (one burst transfer) Burst transfer (2 to 255 times transfer) Continuously transfer (No limit of transfer times specification)
		Data sampling timing	Data is sampled with 2nd edge.
	Ganged Control	Interruption	Transmit interrupt (Transmit completion interrupt, Transmit FIFO interrupt) Receive interrupt (Receive completion interrupt, Receive FIFO interrupt) Error Interrupt (Vertical parity error interrupt, Trigger error interrupt)
		Various status detection	TSPI modify status, Transmit shift operation, Transmit completion, Transmit FIFO fill level/ empty detection, Receive operation, Receive completion, Receive FIFO fill level /full detection
		DMA demand	Transmit: Single DMA request, Burst DMA request Receive: Single DMA request, Burst DMA request
		Trigger communication control	Starting communication by trigger. Refer to the "Product Information" on a reference manual for a trigger source.
	Special Control	Output level of TSPIxTXD during an idle term	High, Low, a last bit data hold, Hi-Z
		Polarity of TSPIxSCK during an idle term	It is Low to during idle term. It is High to during idle term.
		Interval time between frames at the time of burst transfer.	0 x TSPIxSCK cycle to 15 x TSPIxSCK cycles
		Idle time at the time of continuously transfer	1 x TSPIxSCK cycle to 15 x TSPIxSCK cycles
		Software reset	Reset by software is possible.

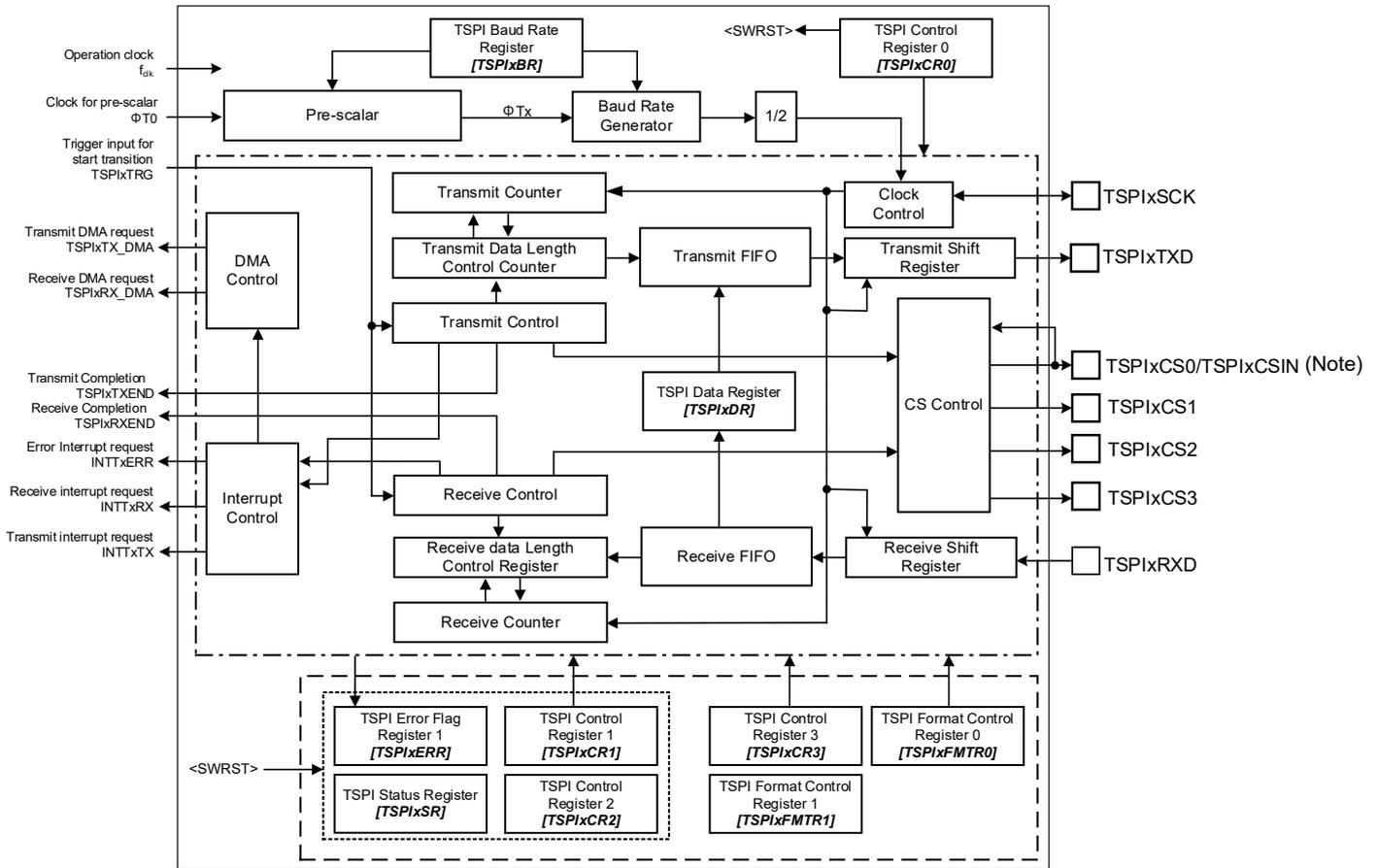
**Table 1.4 Functional outline (SIO mode, slave)**

Function classification		Function	A Functional Description or the range
SIO mode (Slave)	Data Format	Data length	The data length can be setup in a 1-bit unit. 8 to 32bits (with no parity) 7 to 31bits (with parity)
		Parity	Selection of with parity/without parity is possible. Selection of even parity/odd parity is possible.
		The direction of data transfer	Selection of the LSB first/ MSB first is possible.
	Transceiver Control	FIFO number of section	Transmission: 8 steps (16bits), 4 steps (32bits) Reception: 8 steps (16bits), 4 steps (32bits)
		Communication Operation mode	Full duplexes (transmission and reception), transmission, reception
		Transfer mode	Single transfer (one burst transfer) Burst transfer (2 to 255 times transfer) Continuously transfer (No limit of transfer times specification)
		Data sampling timing	Data is sampled with 2nd edge.
	Ganged Control	Interruption	Transmit interrupt (Transmit completion interrupt, Transmit FIFO interrupt) Receive interrupt (Receive completion interrupt, Receive FIFO interrupt) Error Interrupt (Vertical parity error interrupt, Over run interrupt, Under run interrupt)
		Various status detection	TSPI modify status, Transmit shift operation, Transmit completion, Transmit FIFO fill level/ empty detection, Receive operation, Receive completion, Receive FIFO fill level /full detection
		DMA demand	Transmit:Single DMA request, Burst DMA request Receive: Single DMA request, Burst DMA request
	Special Control	Final bit hold time of a TSPIxTXD pin	$2/f_{clk}$ to $128/f_{clk}$ (Note)
		Output level of TSPIxTXD during an idle term	High, Low, a last bit data hold, Hi-Z
		Output level of TSPIxTXD when underrun error occurred	High, Low
		Software reset	Reset by software is possible.

Note:  $f_{clk}$  is either the system clock ( $f_{sys}$ ) or high speed clock ( $f_c$ ), depending on the product. For the details, refer to "Product Information" in Reference manual.

## 2. Configuration

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



**Figure 2.1 Block diagram of TSPI**

Note: Regarding to product, TSPIxCS0 and TSPIxCSIN may be assigned to a different pin. For details, refer to "Input/Output Ports" of reference manual.

**Table 2.1 List of Signals**

No	Signal symbol	Signal name	I/O	Reference manual
1	$f_{clk}$ (Note)	Operation clock	Input	Clock Control and Operation Mode Product Information
2	$\Phi T0$	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 transmission	Output	Input/Output Ports
10	TSPIxRXD	Serial data of reception	Input	Input/Output Ports
11	INTTxTX	Transmit interrupt request	Output	Exception
12	INTTxRX	Receive interrupt request	Output	Exception
13	INTTxERR	Error interrupt request	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 Completion	Output	Product Information
18	TSPIxRXEND	Receive Completion	Output	Product Information

Note:  $f_{clk}$  is either the system clock ( $f_{sys}$ ) or high speed clock ( $f_c$ ), depending on the product. For the details, refer to “Product Information” in Reference manual.

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

#### 3.1.2. Initial setting of TSPI

First, *[TSPIxCR0]*<TSPIE>(TSPI operation control) set as "1". Please perform needed setup, such as communicate mode, transfer mode, and a transfer format, after checking that *[TSPIxSR]*<TSPISUE>(TSPI modify status flag) is "0".

#### 3.1.3. Start and stop transfer

There are two methods for a transfer start in the case of full duplex communication mode and transmitting mode.

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

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

In order to stop transfer, please set "0" to *[TSPIxCR1]*<TRXE>. In single transfer burst transfer and continuously transfer, transfer is performed to the last of the frame under transfer.,

After stopped, TSPIxSCK, TSPIxCS0/1/2/3, and TSPIxTXD will be in an idle state. Please refer to "3.3.7. Special control".

If a transfer is enabled again after stopping transfer in a burst mode, operation will be started from the beginning of the stopped burst transfer.

## 3.2. Data Format

When specifying the transfer direction MSB/LSB first and frame length, set up  $[TSPIxFMTR0]$  (TSPI format control register 0). When specifying the enable parity and even/odd parity, set up  $[TSPIxFMTR1]$  (TSPI format control register 1).

Note: When the parity function is enabled, data length is 31 bits at maximum.

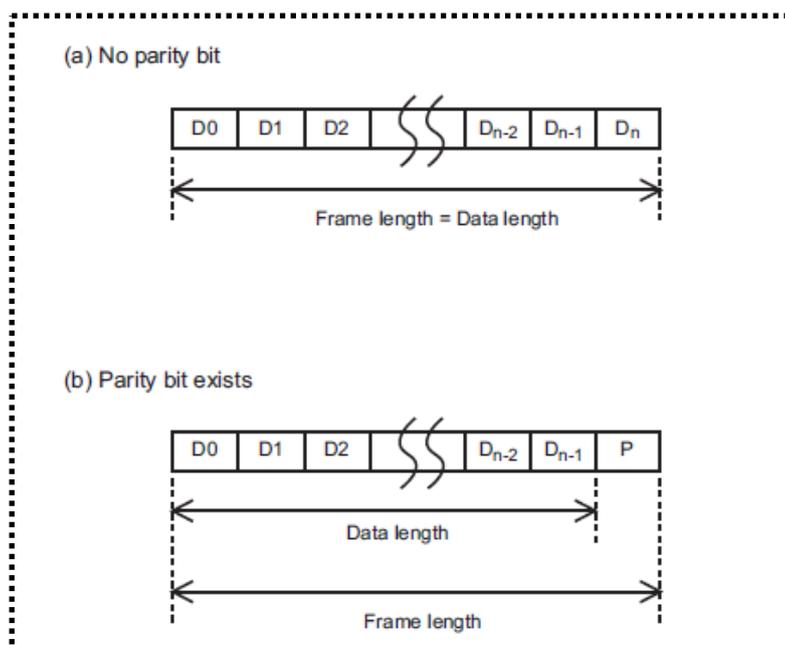


Figure 3.1 Data format

## 3.2.1. Data Format without Parity

If data format is without parity bit, the length of frame must be the same as the length of data. For example, when the length of data is 10-bit, set "001010" to  $[TSPIx\text{FMTR0}] \langle \text{FL}[5:0] \rangle$ . The data in the transmit FIFO is transferred remaining unchanged to the shift register.

### (1) MSB First Transfer (32-bit data, without parity, 32-bit frame length)

Figure 3.2 shows a transmit/receive operation (without parity, MSB first, 32-bit data length).

In the transmission, data in the transmit FIFO is copied to D31 through D0 in the shift register.

Transmit data copied to shift register is transferred sequentially from D31 through D0 on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is copied to the receive FIFO.

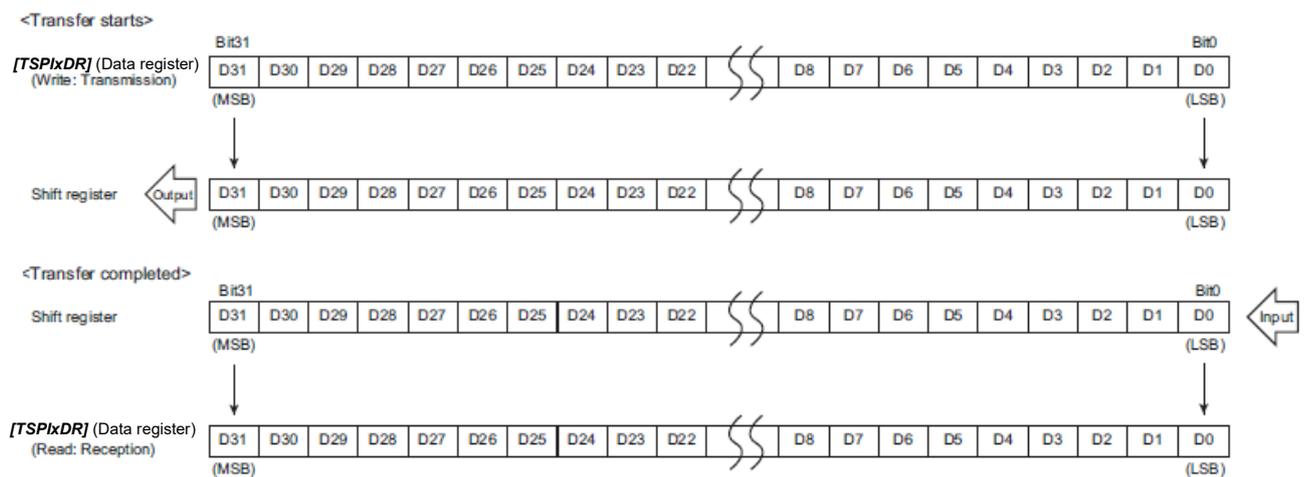


Figure 3.2 MSB first (32-bit data without a parity bit)

## (2) MSB First Transfer (16-bit data without a parity bit, 16-bit data frame length)

Figure 3.3 shows a transmit/receive operation (without parity, MSB first, 32-bit data length).

In the transmission, data in the transmit FIFO is copied to D15 through D0 in the shift register.

Transmit data copied to shift register is transferred sequentially from D15 through D0 on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 16-bit reception data, data is copied to the receive FIFO.

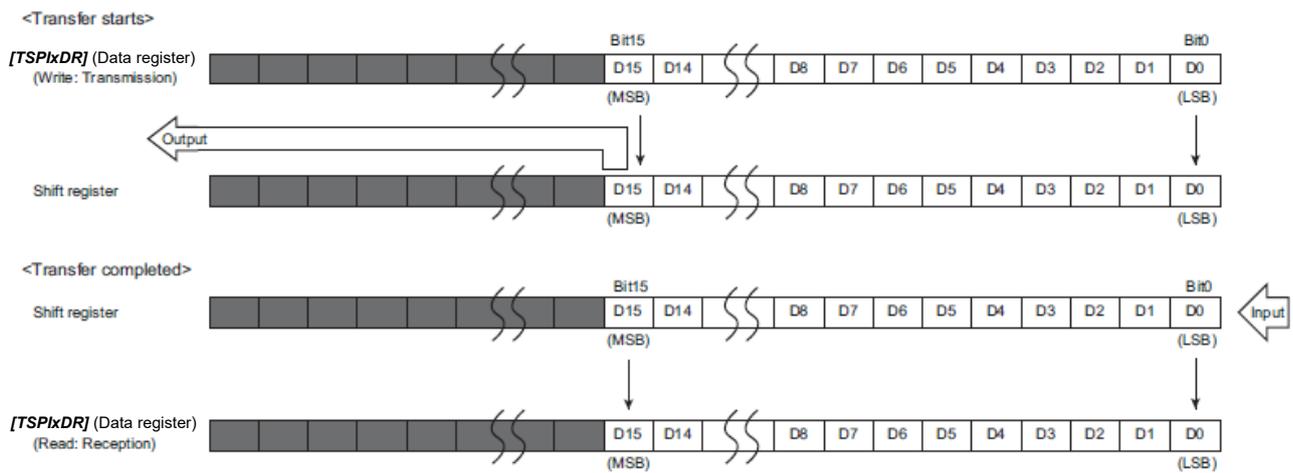


Figure 3.3 MSB first (16-bit data without a parity bit)

### (3) LSB First Transfer (32-bit data without a parity bit, 32-bit frame length)

Figure 3.4 shows a 32-bit data length transmit/receive operation when parity function is disabled.

In the transmission, data in the transmit FIFO is sorted bit by bit when the data is copied to the shift register. Transmit data copied to the shift register is transferred from D0 until reaching 32-bit shifted data on serial clock.

In the reception, receive data is stored in D31 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is sorted bit by bit and copied to the receive FIFO.

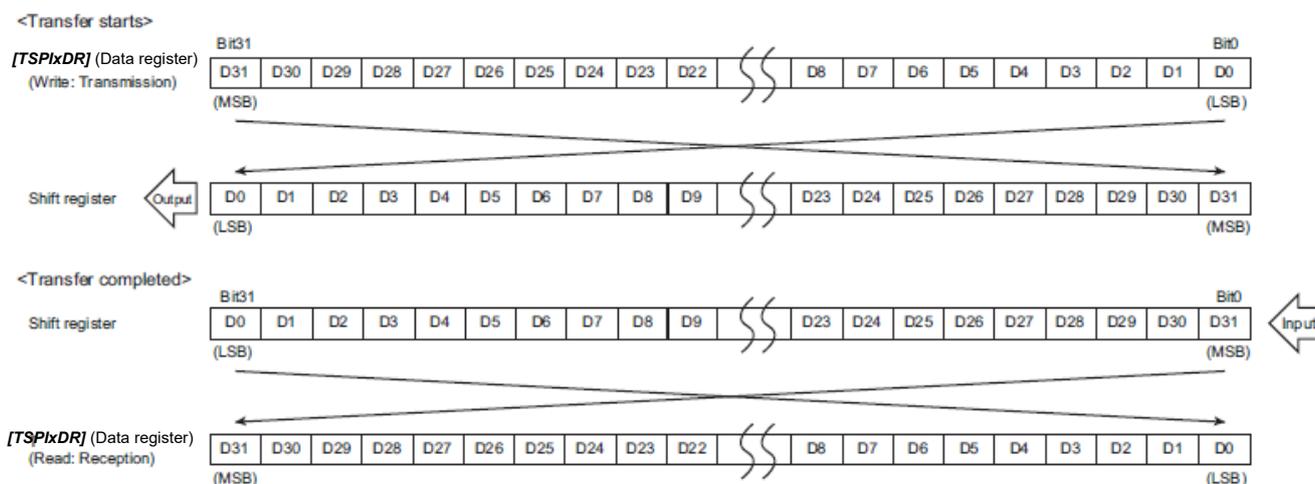


Figure 3.4 LSB first (32-bit data without a parity bit)

#### (4) LSB First Transfer (16-bit data without a parity bit, 16-bit frame length)

Figure 3.5 shows a 16-data length transmit/receive operation (without a parity bit, MSB first, 32-bit data length).

In the transmission, data in the transmit FIFO is sorted bit by bit when the data is copied to the shift register. Transmit data copied to shift register is transferred from D15 until reaching 16-bit shifted data on serial clock.

In the reception, receive data is stored in the D15 of the shift register. Shift operation repeats on serial clock. If the shift register stores 16-bit reception data, data is sorted bit by bit and copied to the receive FIFO.

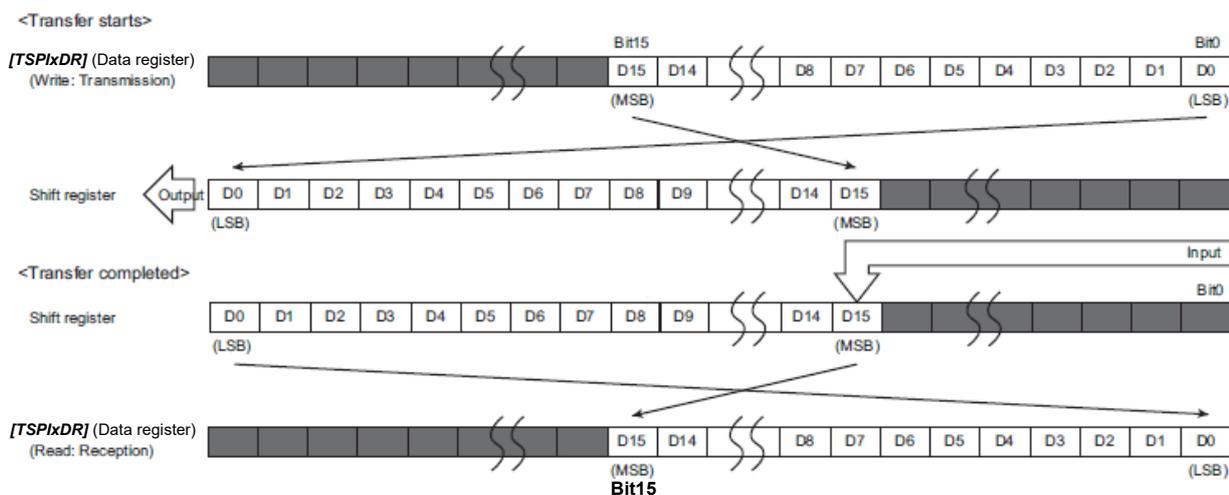


Figure 3.5 LSB first (16-bit data without a parity bit)

## 3.2.2. Data Format with a Parity

If data format is with parity bit, frame length is specified as a data length including a parity bit. For example, when data length is 10-bit, set "001011" to  $[TSPi x FMTR0] \langle FL[5:0] \rangle$ .

If data format is with parity bit, a parity bit is automatically added to data in the transmit FIFO and the data is copied to shift register. the parity bit is also automatically deleted from receive data in the shift register and the data is copied to the receive FIFO.

### (1) MSB first transfer (31-bit data with parity, 32-bit frame length)

Figure 3.6 shows a 31-bit data length transmit/receive operation (with a parity bit, MSB first, 31-bit data length). A frame length is 32-bit data length including a parity bit.

In the transmission, data D30 through D0 in the transmit FIFO are copied to D31 through D1 in the shift register. At the same time, a parity is calculated using data D31 through D1 in the shift register.

The result is stored in the D0 in the shift register.

Subsequently, transmit data in the shift register and parity data are sequentially transferred from D31 through D0 in the shift register on serial clock

In the reception, receive data is stored in D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 32-bit reception data, data is copied to the receive FIFO except a parity bit.

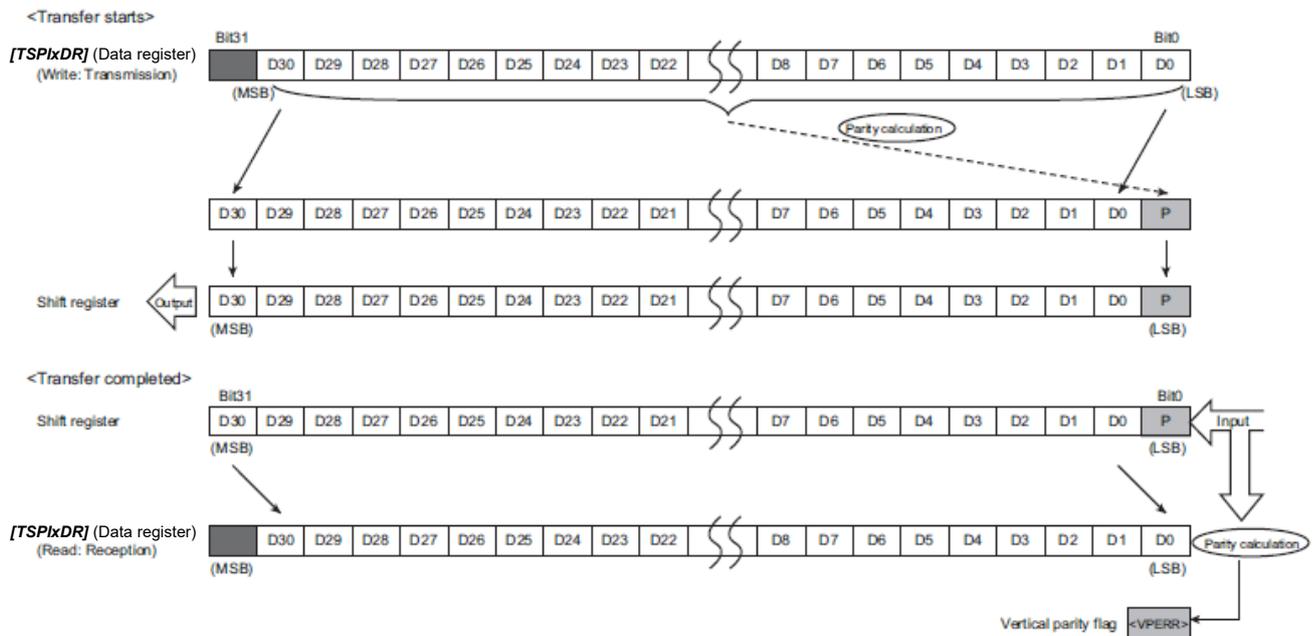


Figure 3.6 MSB first (31-bit data with a parity)

## (2) MSB first transfer (15-bit data with parity, frame length is 16-bit)

Figure 3.7 shows a 15-bit data length transmit/receive operation (with a parity bit, MSB first, 15-bit data length). A frame length is 16-bit data length including a parity bit.

In the transmission, data D14 through D0 in the transmit FIFO is copied to D15 through D1 in the shift register. At the same time, a parity is calculated using data D14 through D0. The result is stored in D0 in the shift register.

Subsequently, transmit data in the shift register and parity data are sequentially transferred from D15 through D0 in the shift register on serial clock

In the reception, receive data is stored in D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 16-bit reception data, data is copied to the receive FIFO except a parity bit.

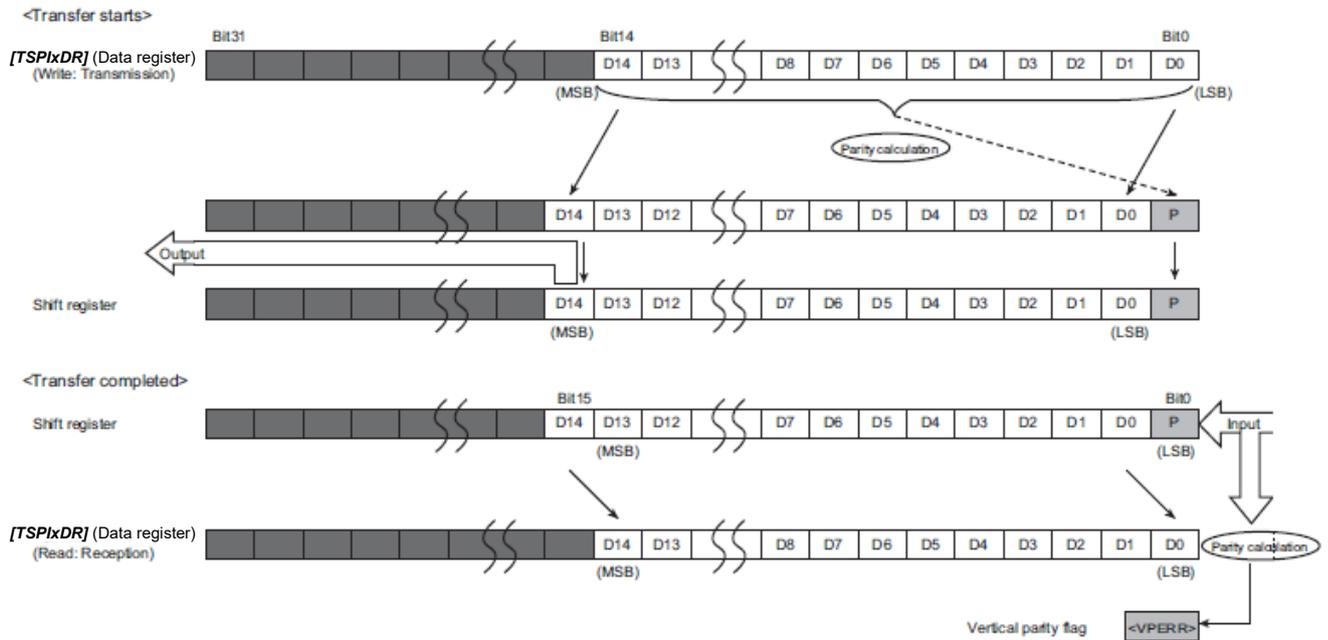


Figure 3.7 MSB first (15-bit data with parity)

### (3) LSB first transfer (31-bit data with parity; frame length is 32-bit)

Figure 3.8 shows a 31-bit data length transmit/receive operation (with a parity bit, LSB first, 31-bit data length).

In the transmission, data D30 through D0 in the transmit FIFO is sorted bit by bit and the data is copied to bit 31 through bit 1 in the shift register. At the same time, a parity is calculated using data D30 through D0. The result is stored in the D0 in the shift register.

Consequently, transmit data in the shift register and a parity data are sequentially transferred from D31 to D0 in the shift register on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 31-bit reception data, only data excluding a parity bit is copied to the receive FIFO.

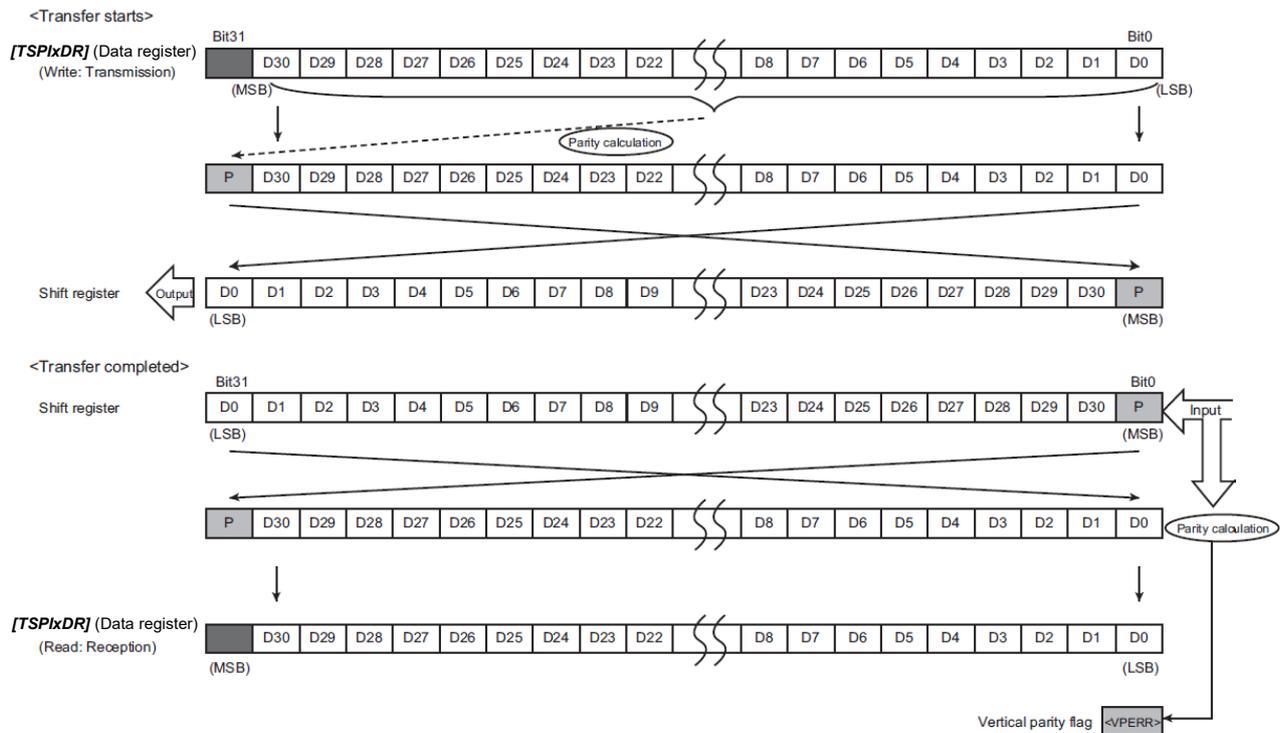


Figure 3.8 LSB first (31-bit data with parity)

## (4) LSB first transfer (15-bit data with a parity bit, 16-bit frame length)

Figure 3.9 shows a 15-bit data length transmit/receive operation (with a parity bit, LSB first, 15-bit data length). A frame length is 16-bit data length including a parity bit.

In the transmission, data D14 through D0 in the transmit FIFO are sorted bit by bit and the data is copied to D17 from D31 in the shift register. At the same time, a parity is calculated using data D14 through D0. The result is stored in the D16 in the shift register.

Subsequently, transmit data in the shift register and parity data are sequentially transferred from D31 through D16 in the shift register on serial clock.

In the reception, receive data is stored in the D0 of the shift register. Shift operation repeats on serial clock. If the shift register stores 16-bit reception data, only data excluding a parity bit is copied to the receive FIFO.

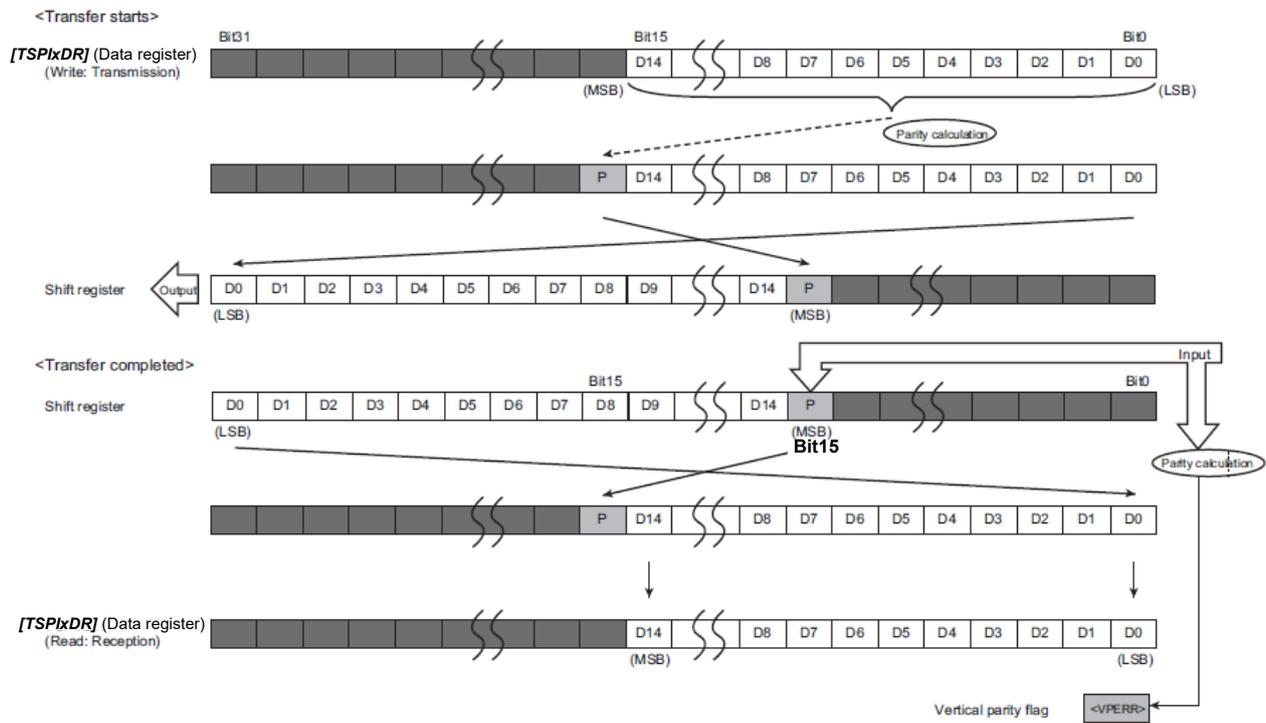


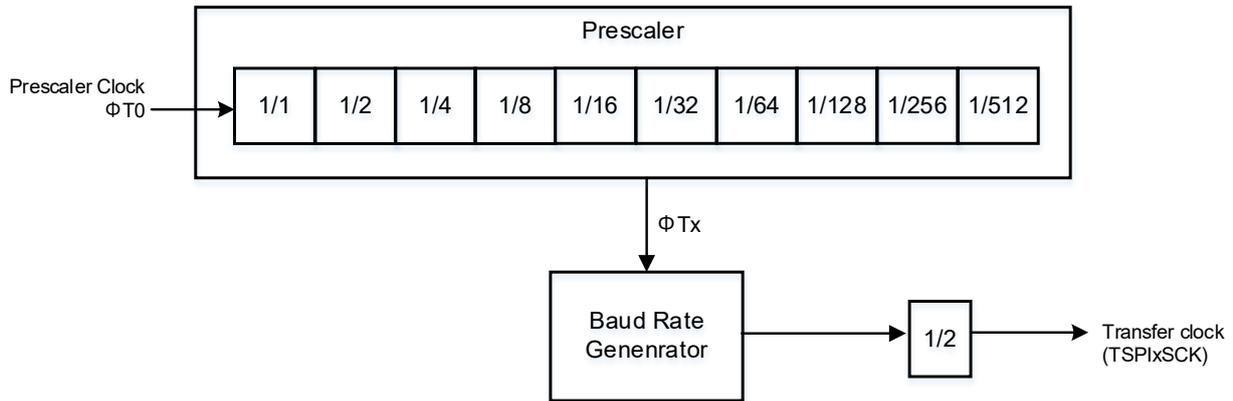
Figure 3.9 LSB first (15-bit data with parity)

### 3.3. Operation

#### 3.3.1. Transfer clock

##### 3.3.1.1. Master operation

The transfer clock generation circuit is shown Figure 3.10.



**Figure 3.10 Transfer clock generation circuit**

The prescaler dividing  $\Phi T0$  from 1/1 to 1/512 ( $\Phi T0$  to  $\Phi T256$ ). Divided clock can be selected by  $[TSPIxBR]<BRCK>$ .

The example of calculation of transfer clock frequency (transfer clock) is shown below.

$$\text{transfer clock} = \Phi T0 \times [TSPIxBR]<BRCK> (1/x) \times [TSPIxBR]<BRS> (1/N) \times 1/2$$

(x= 1,2,4,8,16 to 256, 512, N= 1,2,3,4, to 16)

At this time, please keep below condition

The case of  $[TSPIxCR2]<RXDLY>=0$ ,  $1/2 \times f_{clk} = \text{transfer clock}$   
 The case of  $[TSPIxCR2] <RXDLY>=1$ ,  $1/4 \times f_{clk} \geq \text{transfer clock}$   
 and  
 $f_{clk} \geq \Phi T0$

**Table 3.1 Example of  $f_{clk}/\Phi T_0/\Phi T_x$ / transfer clock and usability**

Condition:  $[TSPIxCR2] < RXDLY \geq 1$ ,  $f_{clk} / \text{transfer clock} \geq 4$ ,  
transfer clock =  $\Phi T_x / 2$  product transfer clock  $\leq 20$  MHz

$f_{clk}$ (MHz)	$\Phi T_0$ (MHz)	$\Phi T_x$ (MHz)	transfer clock TSPIxSCK(MHz)	Usability
160	160	80	40	-
160	160	40	20	✓
160	80	40	20	✓
160	40	40	20	✓
160	20	20	20	-
160	20	20	10	✓
120	120	60	30	-
120	60	60	30	-
120	120	30	15	✓
120	30	30	15	✓
100	100	50	25	-
100	100	25	12.5	✓
100	50	25	25	-
100	50	25	12.5	✓
80	80	40	20	✓
80	40	40	20	✓
80	20	20	20	-
80	20	20	10	✓

✓:Can be used, -:Cannot be used

Condition:  $[TSPIxCR2] < RXDLY \geq 0$ ,  $f_{clk}/TSPIxSCK \geq 2$ ,  
transfer clock =  $\Phi T_x / 2$ , product transfer clock  $\leq 20$  MHz

$f_{clk}$ (MHz)	$\Phi T_0$ (MHz)	$\Phi T_x$ (MHz)	transfer clock TSPIxSCK(MHz)	Usability
40	40	40	20	✓
40	20	20	20	-
40	20	20	10	✓
40	10	10	10	-
20	20	10	10	-
20	10	10	5	✓

✓:Can be used, -:Cannot be used

Note1: Regarding to maximum operation frequency and maximum transfer clock, please refer to “Electric Characteristics” of datasheet.

Note2:  $f_{clk}$  is either the system clock ( $f_{sys}$ ) or high speed clock ( $f_c$ ), depending on the product. For the details, refer to “Product Information” in Reference manual.

### 3.3.1.2. Slave operation

Set the transfer clock frequency so that the following condition is satisfied.

$$1/2 \times f_{clk} \geq \text{transfer clock}$$

Note:  $f_{clk}$  is either the system clock ( $f_{sys}$ ) or high speed clock ( $f_c$ ), depending on the product. For the details, refer to “Product Information” in Reference manual.

### 3.3.2. Communication mode

The TSPI has two communication modes: SIO mode and SPI mode. Communication mode is specified by *[TSPIxCRI]<TSPIMS>*.

#### 3.3.2.1. SPI mode

When write "0" to *[TSPIxCRI]<TSPIMS>*(Communication mode selection), the TSPI operates in SPI mode.

In SPI mode, one master device can be connected with four slave devices via TSPIxSCK(clock input/output), TSPIxCS0/1/2/3(Chip select output), TSPIxCSIN(Chip select input), TSPIxTXD(data transmission), and TSPIxRXD(data reception).

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

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

- Master operation

The TSPI outputs a chip select signal from TSPIxCS0/1/2/3 and outputs a clock from TSPIxSCK. The TSPI operates synchronously with TSPIxSCK.

- Slave operation

The TSPI receives a chip select signal input via TSPIxCSIN pin and operates synchronously with TSPIxSCK. When a chip select signal is invalid, TSPIxSCK is ignored.

### 3.3.2.2. SIO mode

When write "1" to  $[TSPIxCR1]<TSPIMS>$ (Communication mode selection), the TSPI operates in SIO mode.

In SIO mode, one master device can be connected with one slave device via TSPIxSCK(clock input/output), TSPIxTXD(data transmission), TSPIxRXD(data reception).

- Master operation

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

- Slave operation

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

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

### 3.3.2.3. Master / Slave selection

The TSPI operates as Master (the device outputs transfer clock) or Slave (transfer clock is input to the device).

When "0" is set to  $[TSPIxCR1]<MSTR>$ , the TSPI operates as Slave.

When "1" is set to  $[TSPIxCR1]<MSTR>$ , the TSPI operates as Master.

### 3.3.3. Buffer Structure

The transmit buffer and receive buffer are independent respectively. Each buffer has a double-buffering structure consisting of the FIFO and 32-bit width shift register.

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

**Table 3.2 Data format and settable fill level**

Data	Settable fill level	
	Transmit FIFO $[TSPIxCR2]<TIL[3:0]>$	Receive FIFO $[TSPIxCR2]<RIL[3:0]>$
7 to 16bit	0 to 7	1 to 8
17 to 32bit	0 to 3	1 to 4

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

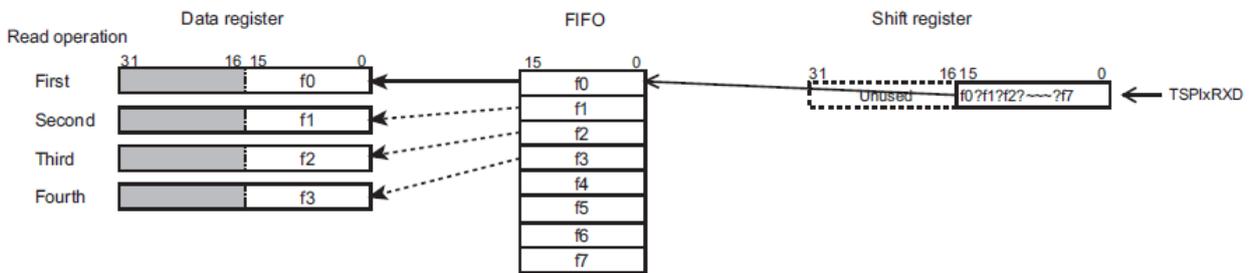
**3.3.3.1. Data Length and FIFO Operation**

Data register is 32-bit width. The FIFO of the TSPI adjusts data to 32-bit width for most efficient DMA transfer. The following description is an example of receive FIFO operation. Transmission is the same operation except data direction.

Received data are indicated as f0 and f1 in each frame. f0 indicates the first frame; f1 indicates the second frame. Also, the upper byte and lower byte in one frame indicate as f1 (H) and f1 (L) respectively.

**(1) Data length 7-bit to 16-bit**

If 7-bit to 16-bit data length are used, one stage of the FIFO is used to store one frame data. The FIFO is 8 stages, so that it can store data up to 8 stage levels.



**Figure 3.11 Operation in 7 to 16-bit data length**

Input data to TSPIxRXD is captured by the shift register.

When a certain frame length is transferred, if the FIFO has space, received data in the shift register is copied to the FIFO. Data is stacked in the FIFO in the order starting from f0, f1, f2, f3, f4, f5, f6 and f7.

If the DMAC or CPU reads data register, the contents of the stage in the receive FIFO directed by the receive FIFO pointer are read.

On the first read operation, the contents f0 of the first stage of the FIFO is copied to the lower 16 bits in the data register. The upper 16 bits in the data register become undefined.

The receive FIFO pointer is incremented by one and directs the second stage of the FIFO.

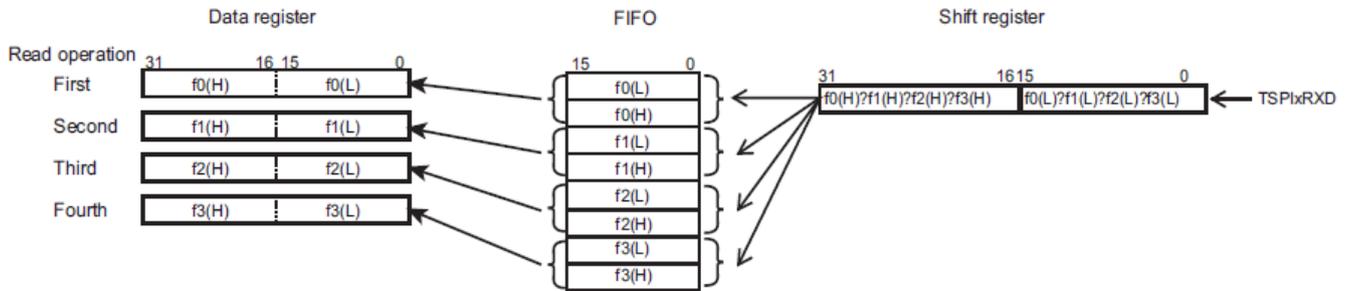
On the second read operation, the contents f1 is copied to the lower 16 bits in the data register.

The receive FIFO pointer is incremented by one and directs the third stage of the FIFO.

In the subsequent frames, the receive pointer is incremented on reading operations, the contents are copied to the lower 16 bits in the data register.

## (2) Data length 17 to 32-bit

If 17 to 32-bit data length are used, two stages of the FIFO are used for one frame. The FIFO has 8 stages, so that it can store four frame data up to four stage levels.



**Figure 3.12 Operation in 17 to 32-bit data length**

Input data to TSPIxRXD is captured in the shift register.

When a certain frame length is transferred, if the FIFO has a space, received data in shift register is copied to the FIFO. Data is stacked in the FIFO in the order starting from f0 (L), f0 (H). In the next frame, data is copied and stacked in the FIFO in the order starting from f1 (L), f1 (H), f2 (L), f2 (H), f3 (L), and f3 (H).

If the DMAC or CPU reads data register, contents of the stage in receive FIFO directed by receive FIFO pointer is read.

On the first read operation, the first stage f0 (L) of the FIFO is copied to the lower 16 bits in the data register. The contents in the second stage f0 (H) of the FIFO is copied to the upper 16 bits in data register.

The receive FIFO pointer is incremented by two and directs the third stage of the FIFO.

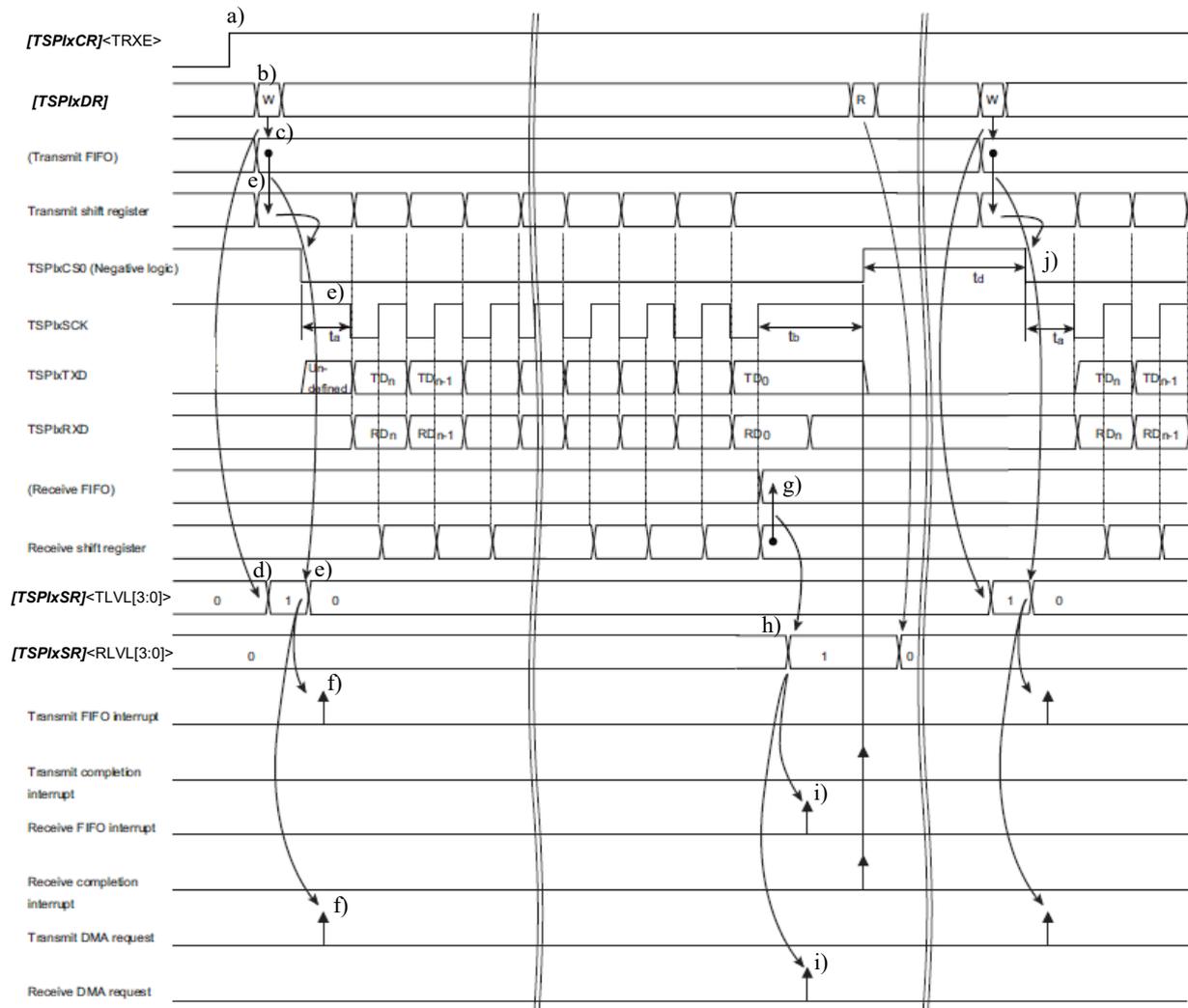
On the second read operation, f1 is read in the same manner as f0. The receive FIFO pointer directs fifth stage of the FIFO.

In the subsequent frames, the receive pointer is incremented by two on reading operations, and the contents fm (L) are copied to the lower 16 bits in the data register; the contents fm (H) are copied to the upper 16 bits in the data register.

### 3.3.4. Communication Operation mode

#### 3.3.4.1. Full duplex communication mode

Figure 3.13 shows an operation example of full duplex communication in continuously transfer (32-bit frame length, no parity, one-stage of FIFO). ( $[TSPIxCR2]<TIDLE[1:0]>=10$ )



**Figure 3.13 Operation example of full duplex communication**

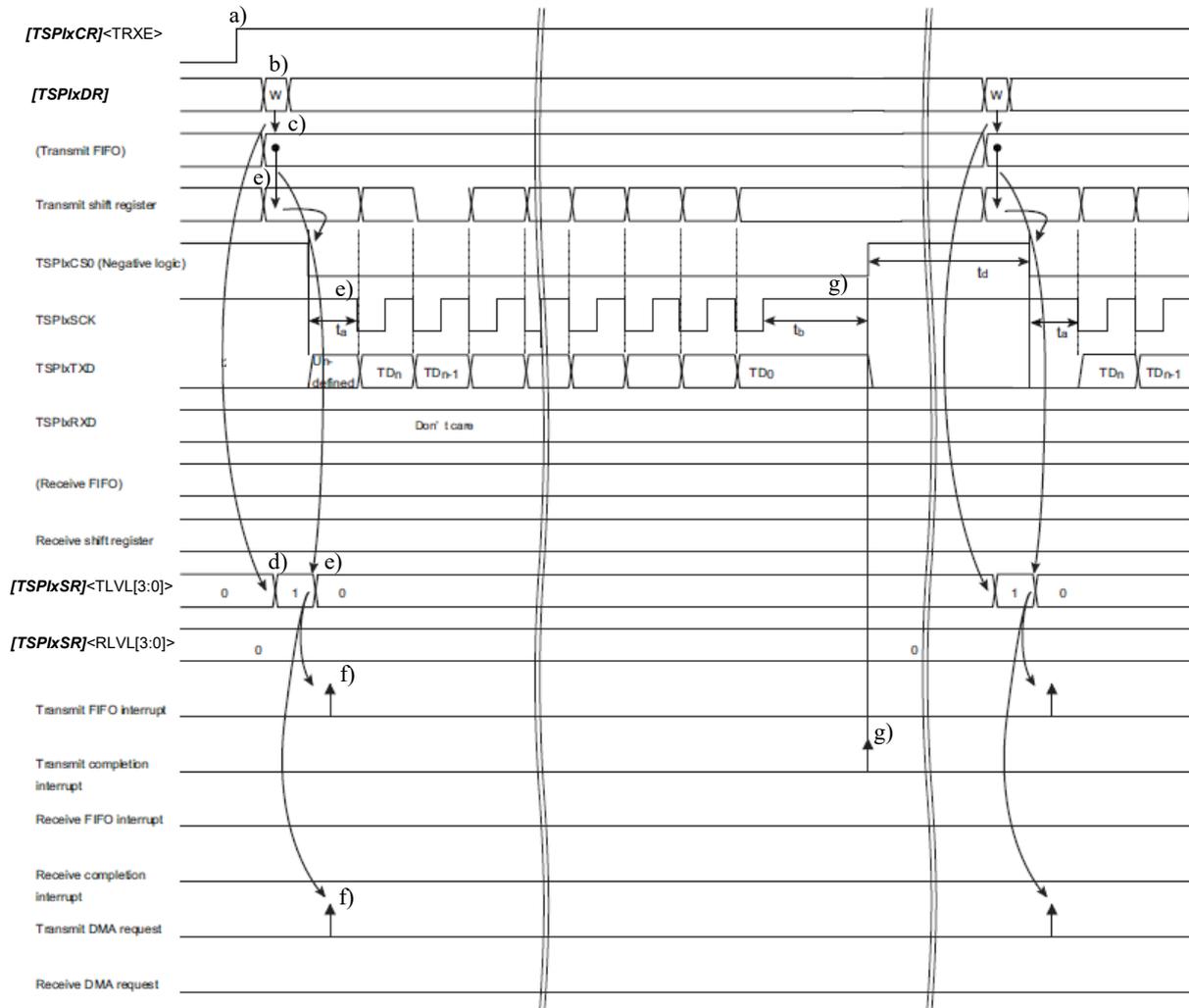
- a) Write "1" to  $[TSPIxCR1]<TRXE>$  to enable communications,
- b) Write data to  $[TSPIxDR]$ .
- c) If data is written to  $[TSPIxDR]$ , the data is written to the stage of FIFO directed by internal transmit FIFO pointer.
- d) Since one stage of data is buffered in the transmit FIFO,  $[TSPIxSR]<TLVL>$  becomes "1".
- e) Buffered data in the transmit FIFO is copied to the shift register, so that  $[TSPIxSR]<TLVL>$  becomes "0". After a serial clock delay time ( $t_a$ ) specified by  $[TSPIxFMTR0]<CSSCKDL>$  has elapsed, TSPiSCK starts outputting serial clock.
- f) Since  $[TSPIxSR]<TLVL>$  changes to "0" from "1", a transmit FIFO interrupt (or transmit DMA request) occurs.
- g) On the last rising edge of serial clock, all bits of receive data are captured by the receive shift register and copied to the receive FIFO. After a CS deassert delay time ( $t_b$ ) specified by  $[TSPIxFMTR0]<SCKCSDL>$  has elapsed after the last rising edge of serial clock, TSPiCS0 is deasserted so that a transmit completion

interrupt and a receive completion interrupt occur.

- h) Since one stage of the receive FIFO is buffered,  $[TSPIxSR]<RLVL>$  becomes "1".
- i) Since  $[TSPIxSR]<RLVL>$  changes to "1" from "0", a receive FIFO interrupt (or receive DMA request) occurs.
- j) Until the minimum idle time ( $t_d$ ) specified by  $[TSPIxFMTR0]<CSINT>$  has elapsed after TSPIxCS0 is deasserted, serial transfer does not start and TSPIxCS0 remains deasserted. After the minimum idle time ( $t_d$ ) has elapsed, TSPIxCS0 is asserted and serial transfer starts.

### 3.3.4.2. Transmit mode

Figure 3.14 shows an operation example of continuously transfer (32-bit frame length, no parity, one-stage of FIFO) in the transmit mode. ( $[TSPIxCR2]<TIDLE[1:0]>=10$ )

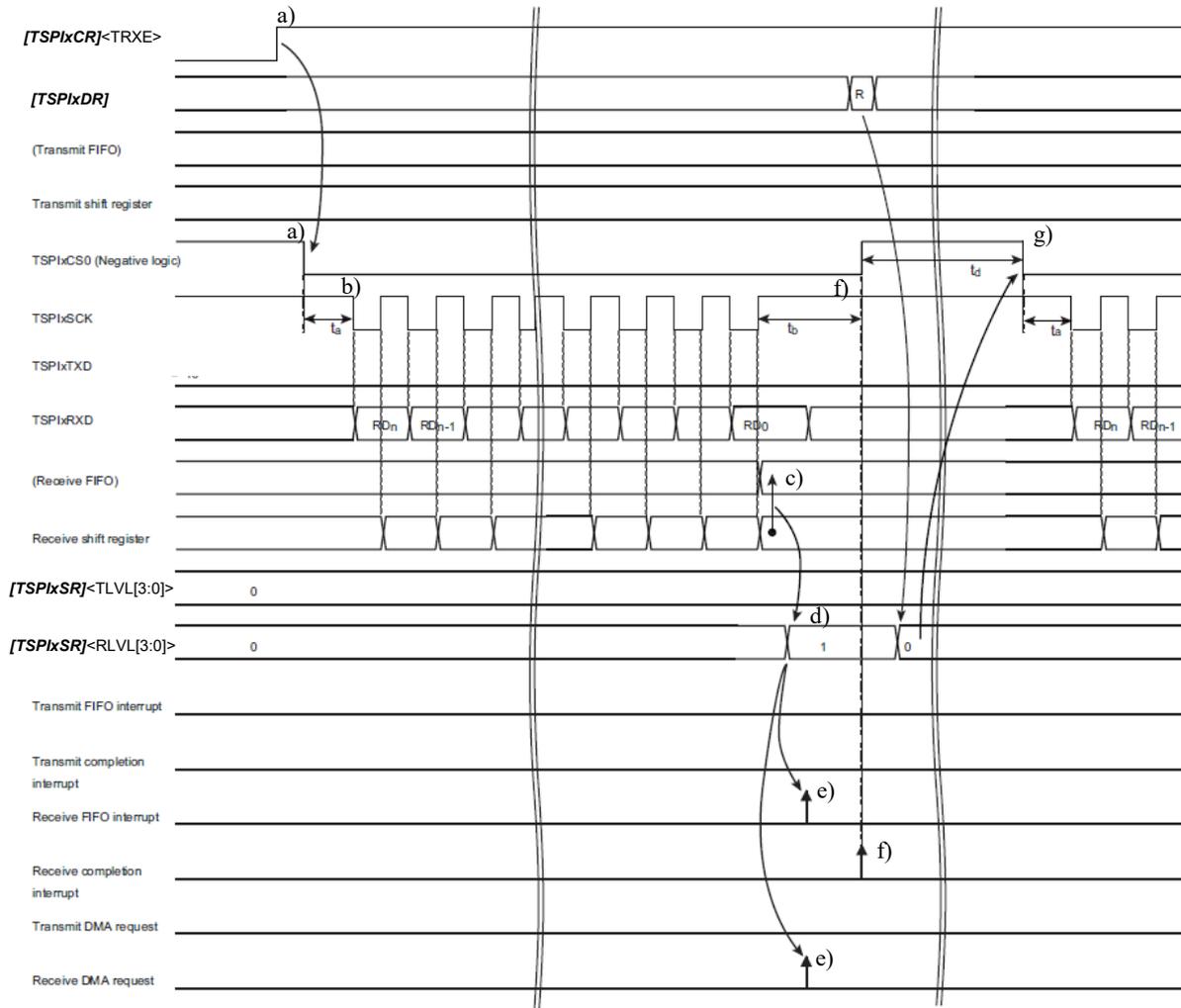


**Figure 3.14 Operation example of transmit mode**

- a) Write "1" to  $[TSPIxCR1]<TRXE>$  to enable the communications
- b) Write data to  $[TSPIxDR]$ .
- c) If data is written to  $[TSPIxDR]$ , data is written to a stage of FIFO directed by internal transmit FIFO pointer.
- d) Since one stage of data is buffered to the transmit FIFO,  $[TSPIxSR]<TLVL>$  becomes "1".
- e) Since buffered data in the transmit FIFO is copied to the shift register,  $[TSPIxSR]<TLVL>$  becomes "0". After a serial clock delay time ( $t_a$ ) specified by  $[TSPIxFMTR0]<CSSCKDL>$  has elapsed, TSPIxSCK starts outputting serial clock.
- f) Since  $[TSPIxSR]<TLVL>$  changed to "0" from "1", a transmit FIFO interrupt (or transmit DMA request) occurs.
- g) Until the minimum idle time ( $t_a$ ) specified by  $[TSPIxFMTR0]<CSINT>$  has elapsed after TSPIxCS0 is deasserted, serial transfer does not start and TSPIxCS0 remains deasserted. After the minimum idle time ( $t_a$ ) has elapsed, TSPIxCS0 is asserted and serial transfer starts.

### 3.3.4.3. Receive mode

Figure 3.15 shows an operation example of continuously transfer (32-bit length, no parity, one stage of FIFO) in receive mode. ( $[TSPIxCR2]<TIDLE[1:0]>=10$ )



**Figure 3.15 Operation example in receive mode**

- Write "1" to  $[TSPIxCR1]<TRXE>$  to enable the communications. Since receive FIFO is not full,  $TSPIxCS0$  is immediately asserted and serial clock transfer starts.
- After a serial clock delay time ( $t_a$ ) specified by  $[TSPIxFMTR0]<CSSCKDL>$  has elapsed, serial clock starts outputting from  $TSPIxSCK$ .
- On the last rising edge of serial clock, all bits of receive data are captured in the receive shift register and the data is copied to the receive FIFO.
- Since one stage data is buffered to the receive FIFO,  $[TSPIxSR]<RLVL[3:0]>$  becomes "1".
- Since  $[TSPIxSR]<RLVL[3:0]>$  changed to "1" from "0", a receive FIFO interrupt (or receive DMA request) occurs.
- After a CS deasserted delay time ( $t_b$ ) specified by  $[TSPIxFMTR0]<SCKCSDL>$  has elapsed after the last rising edge of serial clock,  $TSPIxCS0$  is deasserted and a receive completion interrupt occurs.
- Until the minimum idle time ( $t_d$ ) specified by  $[TSPIxFMTR0]<CSINT>$  has elapsed after  $TSPIxCS0$  is deasserted, serial transfer does not start and  $TSPIxCS0$  remains deasserted. After the minimum idle time ( $t_d$ ) has elapsed,  $TSPIxCS0$  is asserted and serial transfer starts if the receive FIFO is not full.

### 3.3.5. Transfer mode

A transfer mode consists of three modes: single transfer, burst transfer and continuously transfer. Single transfer can transfer one frame of data; burst transfer can transfer multiple frames of data; continuously transfer can transfer without specifying the number of transfer frames.

Transfer mode is set by the frame specified in *[TSPIxCRI]<FC[7:0]>*.

#### 3.3.5.1. Single transfer

Single transfer is the mode that can transfer one frame. The one frame transfer of burst transfer is called single transfer. In the case of the master in SPI mode, TSPIxCS0/1/2/3 is asserted during transfer of one frame, and TSPIxCS0/1/2/3 is deasserted when the transfer is completed.

#### 3.3.5.2. Burst transfer

Burst transfer is the transfer mode that can consecutively transfer multiple frames.

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.

#### 3.3.5.3. 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.

### 3.3.6. Data sampling timing

The data sampling timing can be set by  $[TSPIxFMTR0]<CKPHA>$  (Edge selection register for the serial clock). When  $[TSPIxFMTR0]<CKPHA> = 1$  is set, data is sampled by the second edge. And when,  $[TSPIxFMTR0]<CKPHA>=0$ , by the first edge.

Table 3.3 is shown Usability of communication mode and data sampling timing. And Table 3.4 is shown Data capture timing.

**Table 3.3 Usability of communication mode and data sampling timing**

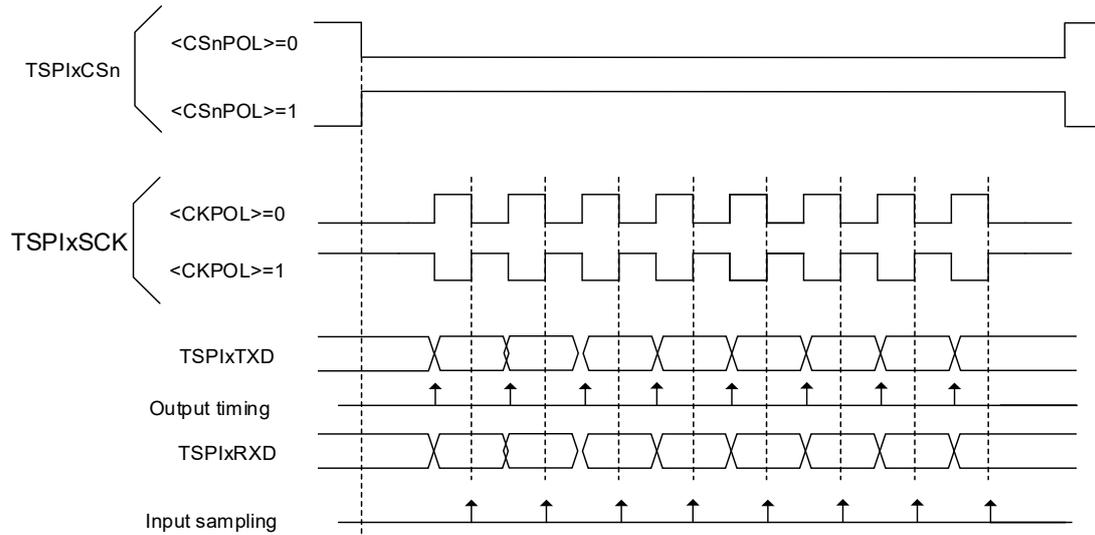
Data Sampling Timing	SPI mode		SIO mode	
	Master Operation	Slave Operation	Master Operation	Slave Operation
2nd edge	✓	✓	✓	✓
1st edge	✓	-	-	-

✓: Can be used, -: Cannot be used

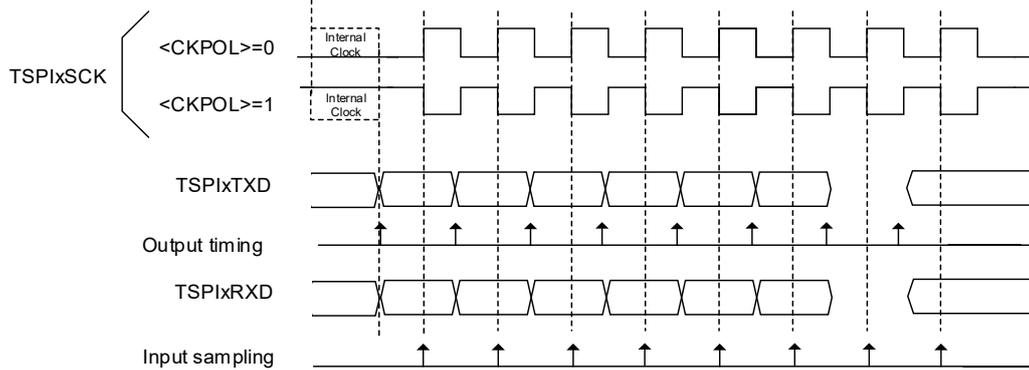
**Table 3.4 Data capture timing**

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

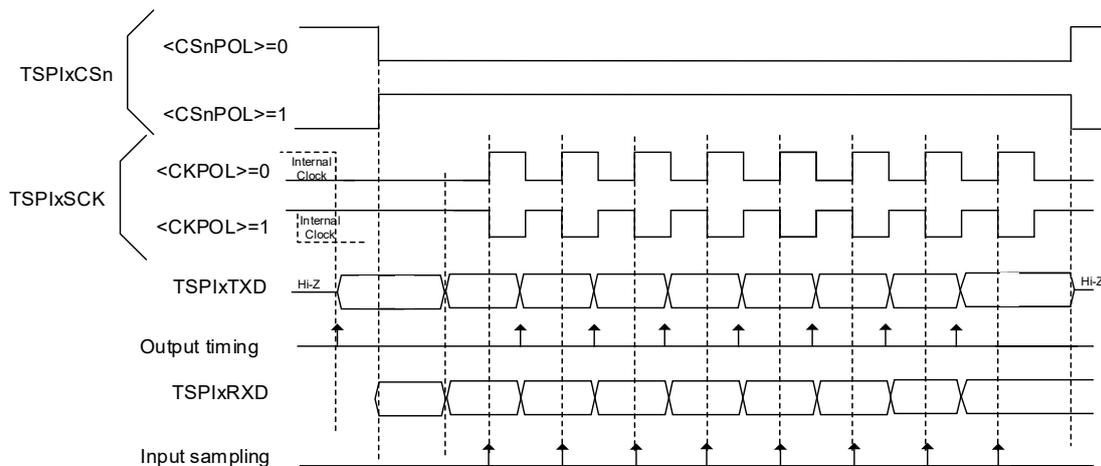
**[SPI mode(master) 2nd edge data sampling<CKPHA>=1]**



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

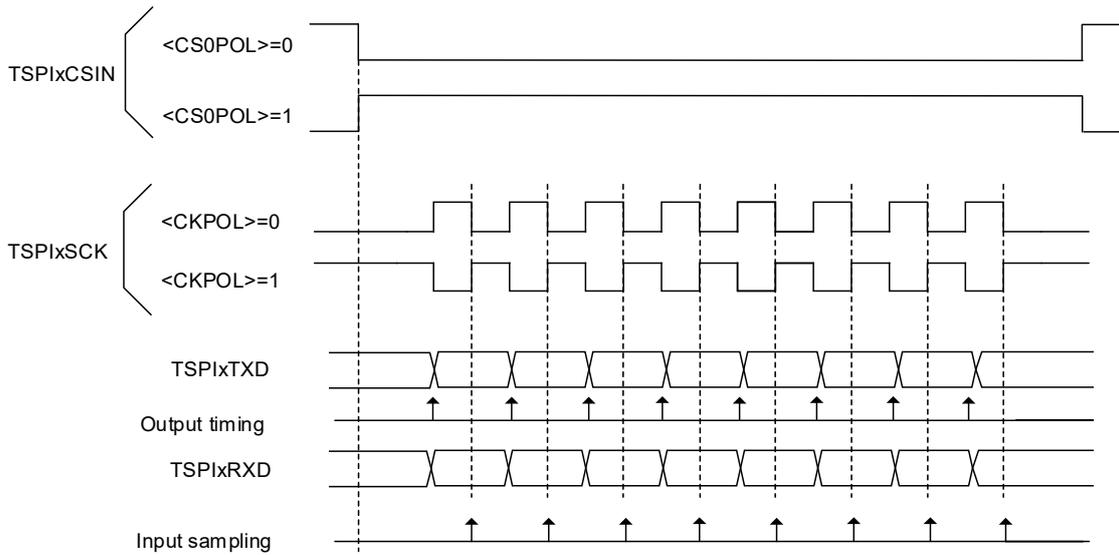


**[SPI mode(master) 1st edge data sampling<CKPHA>=0 Idle period output is Hi-Z<TIDLE>=00]**



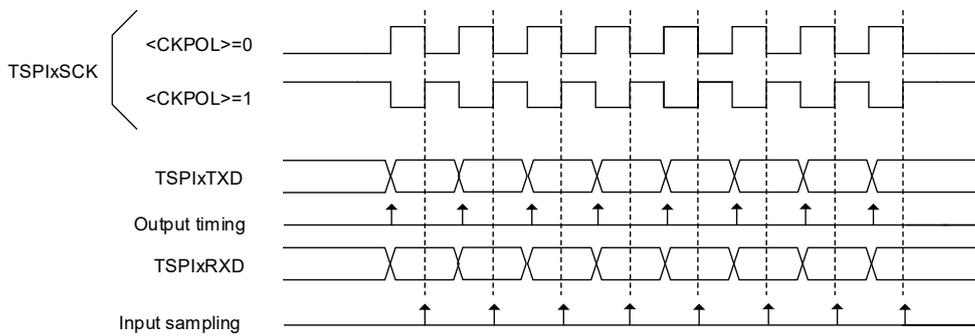
**Figure 3.16 Data sampling timing of SPI mode (master)**

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



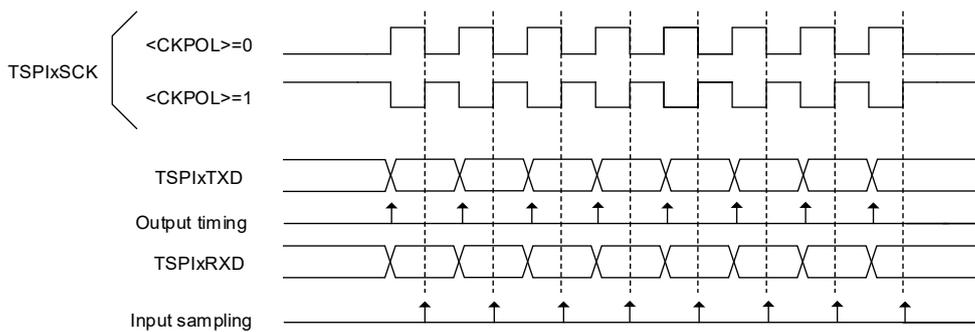
**Figure 3.17 Data sampling timing of SPI mode (slave)**

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



**Figure 3.18 Data sampling timing of SIO mode (master)**

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



**Figure 3.19 Data sampling timing of SIO mode (slave)**

**3.3.7. Special control**

**3.3.7.1. Polarity of TSPIxCS0/1/2/3 signal and generation timing**

The polarity of TSPIxCS0/1/2/3 can be individually selected by  $[TSPIxFMTR0]<CSnPOL>$  (polarity register of TSPIxCS0/1/2/3). In the case of  $[TSPIxFMTR0]<CSnPOL>=0$ , it becomes negative logic and  $[TSPIxFMTR0]<CSnPOL>=1$ , it becomes positive logic.

Moreover, the generating timing of TSPIxCS0/1/2/3 can be set up as follows.

(1) Serial clock delay

“ $t_a$ ” is a delay time from the time when TSPIxCS0/1/2/3 is asserted until the transmit clock (TSPIxSCK) changes. To set a serial clock delay time, set  $[TSPIxFMTR0]<CSSCKDL>$ .

(2) TSPIxCS0/1/2/3/ deassert delay

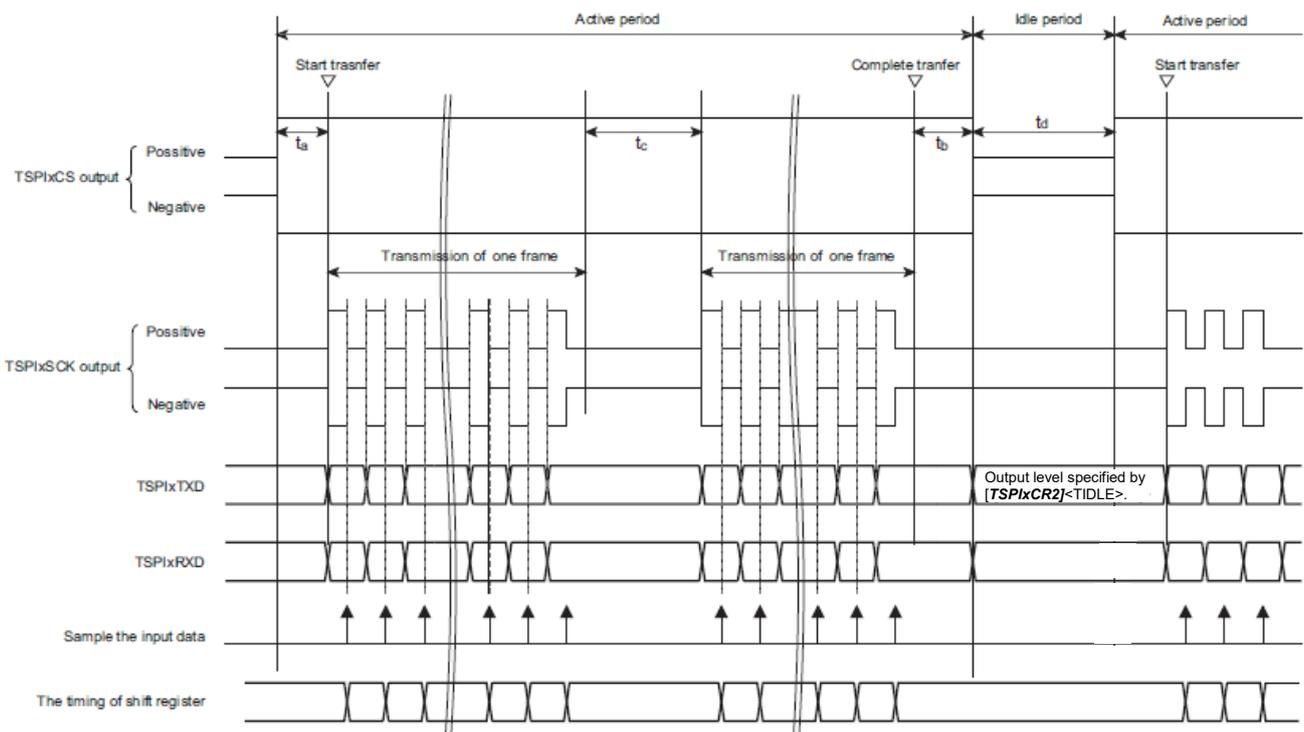
“ $t_b$ ” is a delay time from the time when TSPIxCS0/1/2/3 is deasserted after serial transfer completion. To set a TSPIxCS0/1/2/3 deassert delay time, set  $[TSPIxFMTR0]<SCKCSDL>$ .

(3) Interval time between frames in the burst transfer

“ $t_c$ ” is an interval time between frames in the burst transfer. To set an interval time between frames, set  $[TSPIxFMTR0]<FINT>$ .

(4) Minimum idle time

“ $t_d$ ” is a minimum wait time from the time when TSPIxCS0/1/2/3 is deasserted and then until TSPIxCS0/1/2/3 is asserted again. To set the minimum idle time, set  $[TSPIxFMTR0]<CSINT>$ .



**Figure 3.20 Transfer format and timing adjustment(Example for 2nd edge sampling)**

### 3.3.7.2. Polarity of the Clock

To select a polarity of the clock, set  $[TSPIxFMTR0]<CKPOL>$ .

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

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

### 3.3.7.3. TSPIxTXD Output during Idle

The level of TSPIxTXD output during idle state can be selected by  $[TSPIxCR2]<TIDLE[1:0]>$  (output value fixed functional control register at the time of an idle).

The output value of TSPIxTXD which is according to a set up data at set up timing of  $[TSPIxCR2]<TIDLE[1:0]>$ . However, in master operation, when "Fix to Low" ( $<TIDLE[1:0]>=10$ ) or "Fix to High" ( $<TIDLE[1:0]>=11$ ) is once selected. And re-select to "Last data in previous transmission" ( $<TIDLE[1:0]>=01$ ) immediately after. Then TSPIxTXD is kept previous setting until starts next transmission.

When an underrun error occurs with a final data output at the time of slave operation, the value specified in the  $[TSPIxCR2]<TXDEMP>$  is outputted during frame transmission, and it changes to the data output value performed at the end by the end of transmission.

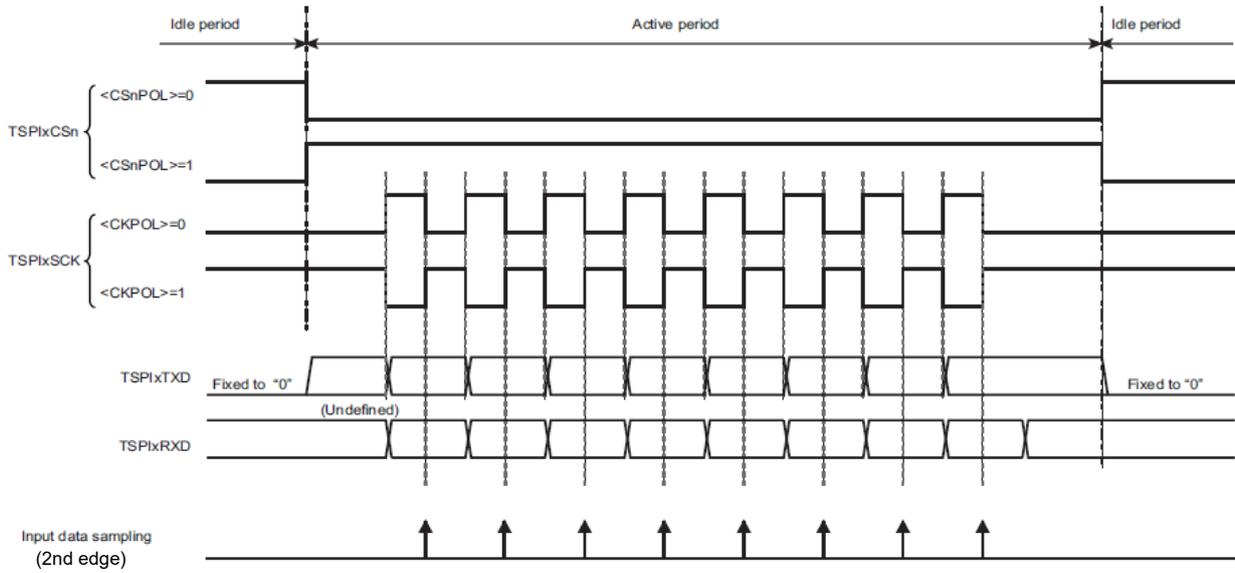
The frame interval period during burst transfer keep the last data of previous transfer. "High" is output if there is no data equivalent to the last data of the previous transfer, such as immediately after reset release.

**Table 3.5 TSPIxTXD output during idle state**

$[TSPIxCR2]<TIDLE[1:0]>$	Output
00	Hi-Z
01	Last data of the previous transfer
10	Low
11	High

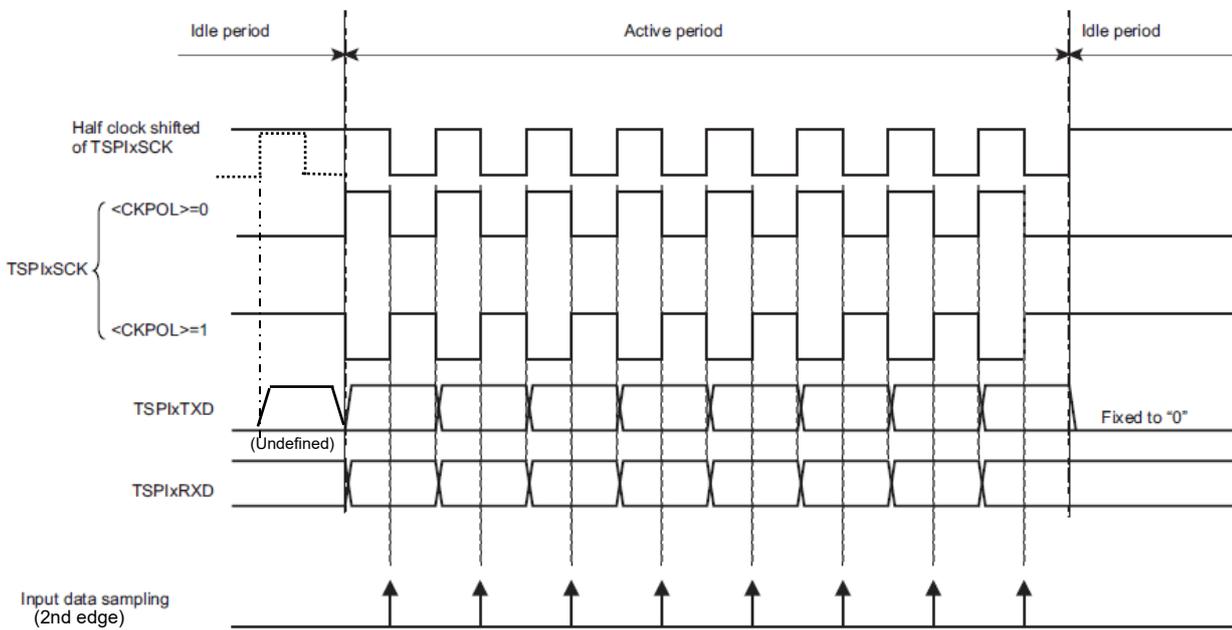
Note: When transmitting by master operation of SIO mode, undefined value is output one clock (TSPIxSCK) early before transmission starts.

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



**Figure 3.21 Idle state in SPI mode and the transmit pin status**

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



**Figure 3.22 Idle state in SIO mode and the transmit pin status**

### 3.3.8. Communication control by trigger

In the master operation, when  $[TXPIxCRI]<TRGEN>$  is set to "1", the function of the starting communication by generating of a trigger can be used.

If "1" is set as  $[TSPIxCRI]<TRGEN>$ , when communication is not performed, it will be waiting for a trigger. If a trigger is inputted, it will be automatically set to  $[TSPIxCRI]<TRXE>=1$ , and communication will be started. After communication is completed, it will be automatically set to  $[TSPIxCRI]<TRXE>=0$  and will be waiting for a trigger.

When you use a trigger start function, please set it as  $[TSPIxCRI]<TRXE>=0$ . Moreover,  $[TSPIxCRI]<FC>=0$  (continuously transferring) cannot use.

When using Trigger transfer control, please refer to Reference manual of "Product Information" to confirm which trigger can use.

The operation of each transmission and reception is as follows.

#### (1) Reception operation

A trigger is inputted, and if FIFO is not full, the receiving start and the clock will be outputted.

When FIFO is not full, reception is continued. And when FIFO is full clock output will be stopped. After the number of transferring which set as  $[TSPIxCRI]<FC>$  is completed, a clock output is stopped and reception is terminated.

A trigger will be ignored if a trigger is inputted when FIFO is full. Moreover, error interruption is outputted and a flag ( $[TSPIxERR]<TRGERR>$ ) is set.

Please refer to "3.3.9.3. Error Interruption" for details.

#### (2) Transmission operation

When the data is in FIFO, a trigger is inputted, the transmission start and the clock will be outputted.

When the data is in FIFO, transmission is continued. And when FIFO is empty clock output will be stopped. After the number of transferring which set as  $[TSPIxCRI]<FC>$  is completed, a clock output is stopped and transmission is terminated.

A trigger will be ignored if a trigger is inputted when FIFO is empty. Moreover, error interruption is outputted and a flag ( $[TSPIxERR]<TRGERR>$ ) is set. Please refer to "3.3.9.3. Error Interruption" for details.

#### (3) Receiving trigger

It receives when communication is not performed. And the trigger input under communicating is ignored. Enter the next trigger after the communication is completed.

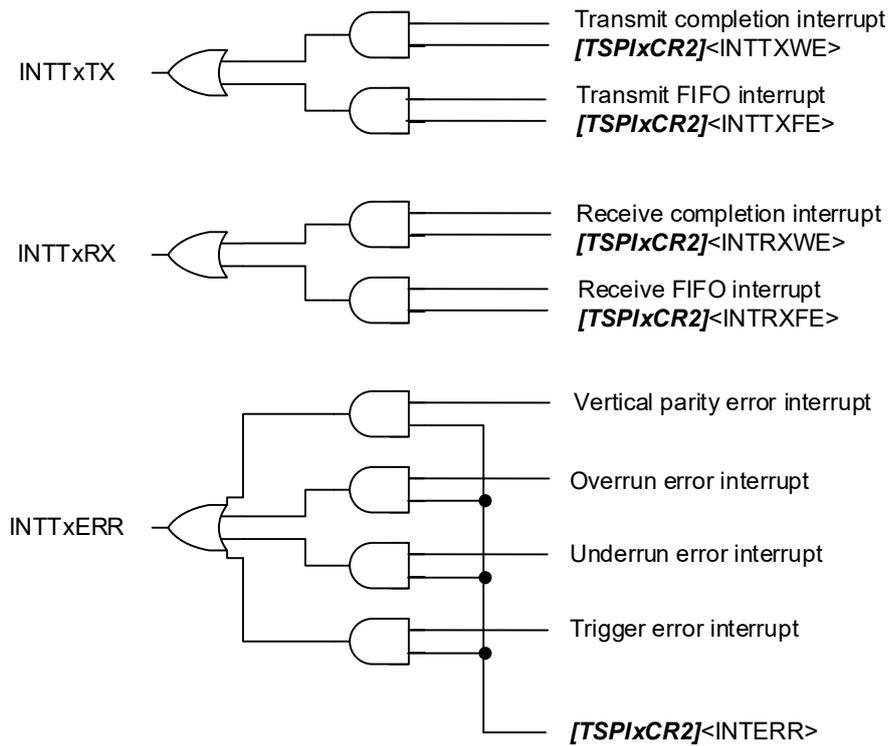
### 3.3.9. Interrupt Request

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. They are enabled/disabled respectively.

**Table 3.6 Interrupt events and requests**

Interrupt request	Interrupt event	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	
	Trigger error interrupt	



**Figure 3.23 Circuit of interrupt request**

### 3.3.9.1. Transmit Completion Interrupt/Receive Completion Interrupt

#### (1) Master operation

Each single transfer, burst transfer and continuously transfer, transmit completion interrupt occurs (Note) when  $TSPIxCS0/1/2/3$  is deasserted in transmission or full duplex communications.

Each single transfer, burst transfer and continuously transfer, receive completion interrupt occurs (Note) when  $TSPIxCS0/1/2/3$  is deasserted in reception or full duplex communications.

Note: When use SIO mode, it cannot confirm the  $TSPIxCS0/1/2/3$  deassertion. The timing of interrupt occurrence is depends on value of  $[TSPIxFRMR0]<SCKCSDL>$ .

#### (2) Slave operation

A transmit completion interrupt occurs in transmission or full-duplex communication. A transmit completion interrupt occurs at the timing of one frame transfer completion in single transfer or continuous transfer, or at the end of the last frame transfer in burst transfer. A reception completion interrupt occurs in receive or full-duplex communication. The receive completion interrupt occurs at the timing of one frame transfer completion in the case of single transfer or continuous transfer, or at the end of the last frame transfer in the case of burst transfer.

### 3.3.9.2. Transmit FIFO Interrupt/Receive FIFO Interrupt

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

- $[TSPIxSR]<TLVL[3:0]>$  is greater one than the transmit FIFO interrupt condition (fill level) specified in  $[TSPIxCR2]<TIL[3:0]>$ .
- Data is transferred from the transmit FIFO to the transmit shift register. fill level of the transmit FIFO is decreased by one, and the level is changed to the same value of transmit interrupt generation condition (fill level).

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

- $[TSPIxSR]<RLVL[3:0]>$  is less one than receive interrupt generation condition (fill level) specified in  $[TSPIxCR2]<RIL[3:0]>$ .
- Data is transferred from the receive shift register to the receive FIFO. fill level of the receive FIFO is increased by one, and the level is changed to the same value of receive interrupt generation condition (fill level).

### 3.3.9.3. Error Interruption

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

#### (1) Parity Error Interrupt

When a parity error is detected, a parity error interrupt is generated.

If a parity is enabled, a parity is calculated using received data that is received one bit previous to the last data of the frame.

The calculated parity is compared with the received parity that is the last bit of frame. If they do not match, a parity error interrupt occurs.

An interrupt generates when receive frame data is stored to the receive FIFO.

## (2) Overrun error interrupt and underrun error interrupt

An underrun and overrun errors occur in slave mode.

An underrun error occurs when data does not exist in the transmit FIFO after data in the shift register is transferred completely if the next transfer clock is input.

An overrun error occurs when the receive FIFO is full and receive shift register contains data if the next transfer clock is input.

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.

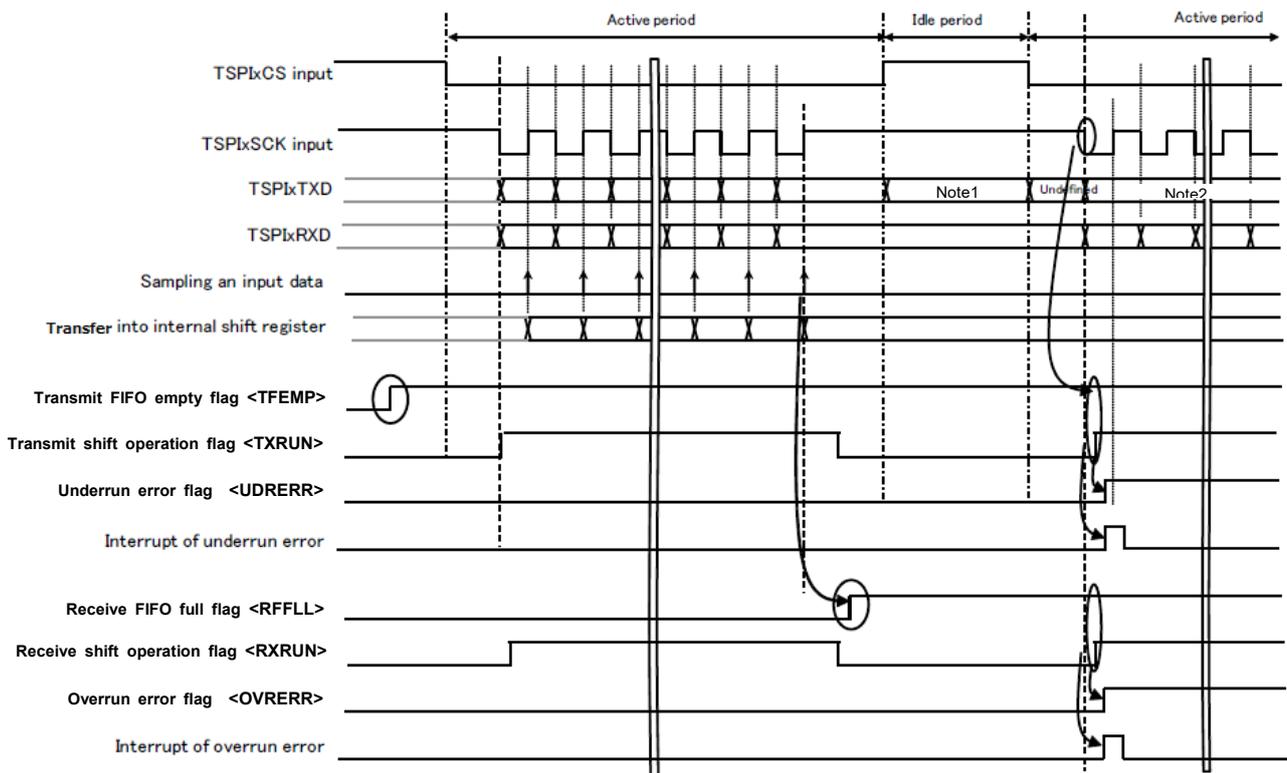


Figure 3.24 Overrun error and underrun error

Note1: It depends on  $[TSPIxCR2]<TIDLE[1:0]>$  of settings.

Note2: It depends on  $[TSPIxCR2]<TXDEMP>$  of settings.

## (3) Trigger error interrupt

In the master operation, it is set when communication by a trigger input is not able to be started in the trigger communication control is enable ( $[TSPIxCR1]<TRGEN>=1$ ) state.

### 3.3.10. DMA request

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

Supported DMA requests depend on the product. Please refer to reference manual of “Product Information” for details.

#### 3.3.10.1. Transmit DMA request

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

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

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

#### 3.3.10.2. Receive DMA request

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

### 3.3.11. Coordinated movements by the completion of communication

It can cooperate with other functions, such as starting of a timer counter, with the signal of the completion of the completion of transmitting/ reception. Please refer to the reference manual "Product Information" for details.

### 3.3.12. Software reset

TSPI can be initialized, being able to apply reset by software. Please refer to “4.2.1 *[TSPIxCR0]* (TSPI control register 0)” for details.

## 4. Registers

### 4.1. Register List

The following table lists the control registers and addresses.

Peripheral	Function name	Channel/Unit	Base address		
			TYPE1	TYPE2	TYPE3
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. Please refer to "Product Information" of the reference manual for the details.

Register name (x= Channel number)		Address(Base+)
TSPI Control Register 0	<b>[TSPIxCR0]</b>	0x0000
TSPI Control Register 1	<b>[TSPIxCR1]</b>	0x0004
TSPI Control Register 2	<b>[TSPIxCR2]</b>	0x0008
TSPI Control Register 3	<b>[TSPIxCR3]</b>	0x000C
TSPI Baud Rate Register	<b>[TSPIxBR]</b>	0x0010
TSPI Format Control Register 0	<b>[TSPIxFMTR0]</b>	0x0014
TSPI Format Control Register 1	<b>[TSPIxFMTR1]</b>	0x0018
TSPI Data Register	<b>[TSPIxDR]</b>	0x0100
TSPI Status Register	<b>[TSPIxSR]</b>	0x0200
TSPI Error Flag Register	<b>[TSPIxERR]</b>	0x0204

Note: Registers except **[TSPIxCR0]**<SWRST>, **[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. (Table4.1)
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 write "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><RIL><INTTXFE><INTTXWE><INTRXFE> <INTRXWE><INTERR><DMATE><DMARE>
[TSPIxCR3]	No registers
[TSPIxBR]	No registers
[TSPIxFMTR0]	No registers
[TSPIxFMTR1]	No registers
[TSPIxDR]	No registers
[TSPIxSR]	<TSPISUE><TXRUN><TXEND><INTTXWF><TFEMP> <TLVL><RXRUN><RXEND><INTRXFF><RFFLL><RLVL>
[TSPIxERR]	<TRGERR><UDRERR><OVRERR><PERR>

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

Bit	Bit Symbol	After reset	Type	Function
31:16	-	0	R	Read as "0".
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 (Note1)(Note2)(Note3)(Note4) 0: Communication stops 1: Communication is enabled  Full duplex mode/transmission mode If valid data exists in the transmit FIFO or shift register, transmission starts. If valid data does not exist in the transmit FIFO or shift register, transmission does not start. To start communications, write data to transmit FIFO or write transmit data when communications are enabled. If this bit is set as disable during transmission, the transmission will stop after the ongoing frame will complete and the setting will disable.  Receive mode: Once this bit is enabled, reception immediately starts. If this bit is set to disable during reception, reception will stop after the ongoing frame is complete and the setting will disable.
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)  If the mode "transmit only" is selected, the process circuit for TSPIxRXD stops. If the mode "receive only" is selected, the process circuit for TSPIxTXD stops.
9:8	CSSEL	0	R/W	Selection of TSPIxCS0/1/2/3 00: TSPIxCS0 is valid 01: TSPIxCS1 is valid 10: TSPIxCS2 is valid 11: TSPIxCS3 is valid  When slave operation and selected SIO mode, TSPIxCS0/1/2/3 cannot be used. (Note5)
7:0	FC[7:0]	0x01	R/W	Sets the number of transfer frames 0: Continuously transfer (No limit of transfer times specification) (Note6) 1: Single transfer (one burst transfer) 2 to 255: Burst transfer (2 to 255 times transfer)

Note1: <TRXE> must set to "1" after all the setting.

Note2: <TRXE> is not cleared to "0" unless the CPU writes "0" to <TRXE> in continuously transfer. However, in the case of single and burst transfer, <TRXE> is automatically cleared to "0" after the specified numbers of transfers are complete. If single and burst transfer is executed again, check whether [TSPIxSR]<TSPISUE> bit returns to "0", and then write "1" to <TRXE>.

Note3: Even if it rewrites <TRXE> to "0" (Communication stops) when actual communication (master side) is not started by <TRXE> after the setup to "1" (Communication is enable), when slave operation, state flag *[TSPIxSR]*<TSPISIUE> is not set to "0" (Modification is enabled). In spite of enable communication in slave device when communication of a master device is not started, please perform re-set it up , after the slave device performs software reset by *[TSPIxCR0]*<SWRST>.

Note4: In the slave operation, if <TRXE> is rewritten to "0" (communication prohibited) when data remains in the transmission buffer (FIFO) during transmission, transmission buffer (FIFO) should be cleared by *[TSPIxCR3]*<TFEMPCLR>, or should be performed software reset by *[TSPIxCR0]*<SWRST>, then re-set it up.

Note5: When slave operation or SIO mode is selected, do not select TSPIxCS0/1/2/3 as the port pin setting.

Note6: It cannot set up at the time of trigger transmission

### 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. 00: Hi-Z 01: Last data in previous transfer 10: Fixed to low 11: Fixed to high
21	TXDEMP	1	R/W	Fixed output value function control when TSPIxTXD underruns. 0: Fixed to low 1: Fixed to high
20:17	-	0	R	Read as "0".
16	RXDLY	1	R/W	Valid only in master mode (Note1)(Note2) 0: $f_{clk} \leq 40\text{MHz}$ 1: $f_{clk} > 40\text{MHz}$
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>.
6	INTTXWE	0	R/W	Transmit completion interrupt control 0: Disabled 1: Enabled  When continuously transfer is completed one frame transfer, single transfer is completed, and during burst transfer, an interrupt is generated at the deassertion timing of TSPIxCS0/1/2/3 when burst transfer is completed.
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>.
4	INTRXWE	0	R/W	Receive completion interrupt control 0: Disabled 1: Enabled  When continuously transfer is completed one frame transfer, single transfer is completed, and during burst transfer, an interrupt is generated at the deassertion timing of TSPIxCS0/1/2/3 when burst transfer is completed.
3	-	0	R	Read as "0".
2	INTERR	0	R/W	Error interrupt control 0: Disabled 1: Enabled  This is enable bit for the transmit parity error, trigger error at master device, or interrupt of overrun error and underrun error at slave device operation.

1	DMATE	0	R/W	<p>Transmit DMA control 0: Disabled 1: Enabled</p> <p>When &lt;DMATE&gt; is enabled The DMA request is output when the transmit FIFO is less than the level set in &lt;TIL&gt;. It does not depend on the state of <b>[TSPIxCR1]</b> &lt;TRXE&gt;. If &lt;DMATE&gt; is set to "0" while the transmit DMA request signal is asserted, the request signal is deasserted. It is reasserted if it satisfies the transmission DMA request signal generation requirement when set again to enable.</p>
0	DMARE	0	R/W	<p>Receive DMA control 0: Disabled 1: Enabled</p> <p>When &lt;DMARE&gt; is enabled The DMA request is output when the receive FIFO is less than the level set in &lt;RIL&gt;. It does not depend on the state of <b>[TSPIxCR1]</b> &lt;TRXE&gt;. If &lt;DMARE&gt; 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 set again to enable.</p>

Note1: Depending on the product, the set value is fixed. For details, refer to "Product information" of reference manual.

Note2:  $f_{clk}$  is either the system clock (fsys) or high speed clock (fc), depending on the product. For the details, refer to "Product Information" in Reference manual.

Note3: Set the fill level within available values shown in "Table 3.2 Data format and settable fill level".

## 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	<p>Clears transmit buffer 0: Invalid 1: Clear</p> <p>By writing "1" to &lt;TFEMPCLR&gt;, the internal pointer of the transmit FIFO and pointer of transmit shift register are initialized. Since the contents of the transmit FIFO and transmit shift register are not affected on the initialization, data remains the previous condition in which transmit buffer is cleared.</p>
0	RFFLLCLR	0	W	<p>Clears receive buffer 0: Invalid 1: Clear</p> <p>By writing "1" to &lt;RFFLLCLR&gt;, the internal pointer of the receive FIFO becomes empty and the internal pointer of receive shift register is initialized. Since the contents of the transmit FIFO and transmit shift register are not affected on the initialization, data remains the previous condition in which transmit buffer is cleared.</p>

## 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	<p>Input clock selection for baud rate generator.            0000: <math>\Phi T0</math>      0101: <math>\Phi T16(1/32 \Phi T0)</math>            0001: <math>\Phi T1(1/2 \Phi T0)</math>    0110: <math>\Phi T32(1/64 \Phi T0)</math>            0010: <math>\Phi T2(1/4 \Phi T0)</math>    0111: <math>\Phi T64(1/128 \Phi T0)</math>            0011: <math>\Phi T4(1/8 \Phi T0)</math>    1000: <math>\Phi T128(1/256 \Phi T0)</math>            0100: <math>\Phi T8(1/16 \Phi T0)</math>    1001: <math>\Phi T256(1/512 \Phi T0)</math></p> <p>1010 to 1111: Prohibited</p>
3:0	BRS[3:0]	0000	R/W	<p>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</p>

## 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 Other than the above is prohibited.
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  This setup is invalid in continuously transfer and slave operation. In SIO mode, a interval time between frames equivalent of <FINT> occurs.
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	Polarity of serial clock 0: Data is sampled on the 1st edge.(Master operation) 1: Data is sampled on the 2nd edge.
14	CKPOL	1	R/W	Polarity of idle period of serial clock (Note2) 0: TSPIxSCK is "Low" level at idle. 1: TSPIxSCK is "High" level at idle.

13:10	CSINT[3:0]	0001	R/W	<p>Idle time (Note3) TSPIxCS0/1/2/3 invalid→TSPIxCS0/1/2/3 valid time</p> <p>0000: Prohibited 0001: 1 x TSPIxSCK cycle 0010: 2 x TSPIxSCK cycles : 1110: 14 x TSPIxSCK cycles 1111: 15 x TSPIxSCK cycles</p> <p>CS deassertion period until the start of next frame of continuously transfer. Even if selecting SIO mode, transfer waits &lt;CSINT&gt; setting value time equivalent. The setting is valid only in master mode.</p>																
9:8	-	0	R	Read as "0".																
7:4	CSSCKDL[3:0]	0000	R/W	<p>Serial clock delay TSPIxCS0/1/2/3 valid→TSPIxSCK valid time</p> <table border="0"> <tr> <td>0000: 1 x TSPIxSCK</td> <td>1000: 9 x TSPIxSCK</td> </tr> <tr> <td>0001: 2 x TSPIxSCK</td> <td>1001: 10 x TSPIxSCK</td> </tr> <tr> <td>0010: 3 x TSPIxSCK</td> <td>1010: 11 x TSPIxSCK</td> </tr> <tr> <td>0011: 4 x TSPIxSCK</td> <td>1011: 12 x TSPIxSCK</td> </tr> <tr> <td>0100: 5 x TSPIxSCK</td> <td>1100: 13 x TSPIxSCK</td> </tr> <tr> <td>0101: 6 x TSPIxSCK</td> <td>1101: 14 x TSPIxSCK</td> </tr> <tr> <td>0110: 7 x TSPIxSCK</td> <td>1110: 15 x TSPIxSCK</td> </tr> <tr> <td>0111: 8 x TSPIxSCK</td> <td>1111: 16 x TSPIxSCK</td> </tr> </table> <p>Set the time from the assertion of the TSPIxCS0/1/2/3 pin until the TSPIxSCK pin changes in units of the serial clock cycle. The setting is valid only in master mode.</p>	0000: 1 x TSPIxSCK	1000: 9 x TSPIxSCK	0001: 2 x TSPIxSCK	1001: 10 x TSPIxSCK	0010: 3 x TSPIxSCK	1010: 11 x TSPIxSCK	0011: 4 x TSPIxSCK	1011: 12 x TSPIxSCK	0100: 5 x TSPIxSCK	1100: 13 x TSPIxSCK	0101: 6 x TSPIxSCK	1101: 14 x TSPIxSCK	0110: 7 x TSPIxSCK	1110: 15 x TSPIxSCK	0111: 8 x TSPIxSCK	1111: 16 x TSPIxSCK
0000: 1 x TSPIxSCK	1000: 9 x TSPIxSCK																			
0001: 2 x TSPIxSCK	1001: 10 x TSPIxSCK																			
0010: 3 x TSPIxSCK	1010: 11 x TSPIxSCK																			
0011: 4 x TSPIxSCK	1011: 12 x TSPIxSCK																			
0100: 5 x TSPIxSCK	1100: 13 x TSPIxSCK																			
0101: 6 x TSPIxSCK	1101: 14 x TSPIxSCK																			
0110: 7 x TSPIxSCK	1110: 15 x TSPIxSCK																			
0111: 8 x TSPIxSCK	1111: 16 x TSPIxSCK																			
3:0	SCKCSDL[3:0]	0000	R/W	<p>TSPIxCS0/1/2/3 deassertion delay Last data →TSPIxCS0/1/2/3 invalid time</p> <table border="0"> <tr> <td>0000: 1 x TSPIxSCK</td> <td>1000: 9 x TSPIxSCK</td> </tr> <tr> <td>0001: 2 x TSPIxSCK</td> <td>1001: 10 x TSPIxSCK</td> </tr> <tr> <td>0010: 3 x TSPIxSCK</td> <td>1010: 11 x TSPIxSCK</td> </tr> <tr> <td>0011: 4 x TSPIxSCK</td> <td>1011: 12 x TSPIxSCK</td> </tr> <tr> <td>0100: 5 x TSPIxSCK</td> <td>1100: 13 x TSPIxSCK</td> </tr> <tr> <td>0101: 6 x TSPIxSCK</td> <td>1101: 14 x TSPIxSCK</td> </tr> <tr> <td>0110: 7 x TSPIxSCK</td> <td>1110: 15 x TSPIxSCK</td> </tr> <tr> <td>0111: 8 x TSPIxSCK</td> <td>1111: 16 x TSPIxSCK</td> </tr> </table> <p>Set the time from the position of the last data until the TSPIxCS0/1/2/3 pin is deasserted in units of the serial clock cycle. The setting is valid only in master mode.</p>	0000: 1 x TSPIxSCK	1000: 9 x TSPIxSCK	0001: 2 x TSPIxSCK	1001: 10 x TSPIxSCK	0010: 3 x TSPIxSCK	1010: 11 x TSPIxSCK	0011: 4 x TSPIxSCK	1011: 12 x TSPIxSCK	0100: 5 x TSPIxSCK	1100: 13 x TSPIxSCK	0101: 6 x TSPIxSCK	1101: 14 x TSPIxSCK	0110: 7 x TSPIxSCK	1110: 15 x TSPIxSCK	0111: 8 x TSPIxSCK	1111: 16 x TSPIxSCK
0000: 1 x TSPIxSCK	1000: 9 x TSPIxSCK																			
0001: 2 x TSPIxSCK	1001: 10 x TSPIxSCK																			
0010: 3 x TSPIxSCK	1010: 11 x TSPIxSCK																			
0011: 4 x TSPIxSCK	1011: 12 x TSPIxSCK																			
0100: 5 x TSPIxSCK	1100: 13 x TSPIxSCK																			
0101: 6 x TSPIxSCK	1101: 14 x TSPIxSCK																			
0110: 7 x TSPIxSCK	1110: 15 x TSPIxSCK																			
0111: 8 x TSPIxSCK	1111: 16 x TSPIxSCK																			

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 transmission/reception is disabled (**[TSPIxCR2]<TRXE>**=0) at the time of slave operation.

Note3: 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 slave mode.(Note1) 000: 2/f <sub>clk</sub> 001: 4/f <sub>clk</sub> 010: 8/f <sub>clk</sub> 011: 16/f <sub>clk</sub> 100: 32/f <sub>clk</sub> 101: 64/f <sub>clk</sub> 110: 128/f <sub>clk</sub> 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 mode selection(Note2) 0: Even parity 1: Odd parity

Note1: f<sub>clk</sub> is either the system clock (f<sub>sys</sub>) or high speed clock (f<sub>c</sub>), depending on the product. For the details, refer to "Product Information" in Reference manual.

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

## 4.2.8. [TSPIxDR] (TSPI Data Register)

Bit	Bit Symbol	After reset	Type	Function
31:0	TSPIxDR[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.

## 4.2.9. [TSPIxSR] (TSPI Status Register)

Bit	Bit Symbol	After reset	Type	Function											
31	TSPISUE	0	R	<p>TSPI modify status flag 0: Modification is enabled. 1: Modification is disabled.</p> <p>If &lt;TSPISUE&gt; is "0", the TSPI is not transmitting or receiving, thus the register setting can be modified. &lt;TSPISUE&gt; is "0" in the following conditions. (Please refer to Table 4.2):</p> <ol style="list-style-type: none"> <li>1. Reset is input.</li> <li>2. Software reset occurs.</li> <li>3. In the continuously transfer mode, the time when current transferring frame is finished when [TSPIxCR1]&lt;TRXE&gt;=0 is set.</li> <li>4. In the burst mode, the timing when specified number of transfers is finished.</li> <li>5. The time when current transferring frame is finished when [TSPIxCR1]&lt;TRXE&gt;=0 is set during burst transfer.</li> </ol> <p>However, even if above conditions are satisfied, &lt;TSPISUE&gt; does not become "0" when the transmit FIFO or receive shift register is full. To set &lt;TSPISUE&gt;=0, read the receive FIFO and transfer a receive value in the receive shift register to the receive FIFO.</p>											
30:24	-	0	R	Read as "0".											
23	TXRUN	0	R	<p>Transmit shift operation flag 0: Stop 1: Operation</p> <p>A status flag indicates the transmit shift operation is ongoing. Combination of &lt;TXRUN&gt; and &lt;TFEMP&gt; bits indicates the following status:</p> <table border="1"> <thead> <tr> <th>&lt;TXRUN&gt;</th> <th>&lt;TFEMP&gt;</th> <th>Conditions</th> </tr> </thead> <tbody> <tr> <td rowspan="2">0</td> <td>0</td> <td>Stop or wait for the next transmission</td> </tr> <tr> <td>1</td> <td>Completed transmission and the transmit FIFO is empty.</td> </tr> <tr> <td>1</td> <td>-</td> <td>In transmission</td> </tr> </tbody> </table> <p>&lt;TXRUN&gt; is set when data exists in the transmit shift register even if data does not exist in the transmit FIFO.</p>	<TXRUN>	<TFEMP>	Conditions	0	0	Stop or wait for the next transmission	1	Completed transmission and the transmit FIFO is empty.	1	-	In transmission
<TXRUN>	<TFEMP>	Conditions													
0	0	Stop or wait for the next transmission													
	1	Completed transmission and the transmit FIFO is empty.													
1	-	In transmission													
22	TXEND	0	R	<p>Transmit completion flag 0: - 1: Transmit is complete.</p> <p>A flag that is set at the time when transmission is complete. This flag is set at the last frame (TSPIxCS0/1/2/3 is deasserted) in the single transfer, burst transfer or continuously transfer after one frame transfer.</p>											
			W	<p>This bit is cleared by writing "1". 0: Don't care 1: Flag is cleared.</p> <p>When the setting by transmission completion and clearing by writing "1" occur simultaneously, the setting by transmission receives a higher priority.</p>											

21	INTTXWF	0	R	Transmit FIFO interrupt flag 0: No interrupt 1: Interrupt occurs  This bit is set when remaining data in the transmit FIFO reaches a TIL value from a fill level setting value (TIL)+1.											
			W	0: Don't care 1: Flag is cleared.  This bit is cleared by writing "1".											
20	TFEMP	1	R	Transmit FIFO empty flag 0: Data exists in the FIFO. 1: Empty  When the transmit FIFO is empty, "1" is set. If transmit data is written to the transmit FIFO, this bit is automatically cleared to "0".											
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 varies 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  A status flag indicates the receive shift operation is ongoing. Combination of <RXRUN> and <RFLL> indicate the following status.											
				<table border="1"> <thead> <tr> <th>&lt;RXRUN&gt;</th> <th>&lt;RFLL&gt;</th> <th>Conditions</th> </tr> </thead> <tbody> <tr> <td rowspan="2">0</td> <td>0</td> <td>Stop or wait for next reception</td> </tr> <tr> <td>1</td> <td>The receive FIFO is FULL and reception is complete.</td> </tr> <tr> <td>1</td> <td>-</td> <td>Receiving is ongoing.</td> </tr> </tbody> </table>	<RXRUN>	<RFLL>	Conditions	0	0	Stop or wait for next reception	1	The receive FIFO is FULL and reception is complete.	1	-	Receiving is ongoing.
				<RXRUN>	<RFLL>	Conditions									
				0	0	Stop or wait for next reception									
1	The receive FIFO is FULL and reception is complete.														
1	-	Receiving is ongoing.													
6	RXEND	0	R	Receive completion flag 0: - 1: Receiving is complete.  A flag that is set at the time when reception is complete. This flag is set at the last frame (TSPIxCS0/1/2/3 is deasserted) in the single transfer, burst transfer and continuously transfer after one frame transfer.											
			W	This bit is cleared by writing "1". 0: Don't care 1: Flag is cleared.  When setting by reception completion and clearing by writing "1" occur simultaneously, setting receives a 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 an RIL value from a fill level setting value (RIL)-1.											
			W	0: Don't care 1: Flag is cleared.  This bit is cleared by writing "1".											

4	RFFLL	0	R	<p>Receive FIFO full flag 0: A space exists in the FIFO 1: Full</p> <p>Indicates that the receive FIFO is full. This bit is automatically cleared if data is read from the receive FIFO.</p>
3:0	RLVL[3:0]	0000	R	<p>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 frame. Table 4.3 shows the display range on &lt;TLVL&gt;.</p>

**Table 4.2 The timing of write "0" to <TRXE>, and state of <TSPISUE>**

Timing of write "0" to [TSPixCR1]<TRXE>	State of <TSPISUE>	
	Master operation	Slave operation
Before start transferring	Clear to "0"	"1" (Please performs software reset by [TSPixCR0]<SWRST>)
During transferring	Clear to "0" after finish current frame	Clear to "0" after finish current frame
After stop transferring	Clear to "0"	Clear to "0"

**Table 4.3 Current value of fill level depending on the range of <TLVL>/<RLVL>**

Frame length	FIFO configuration		
	FIFO stage	Range of <RLVL> when receiving	Range of <TLVL> when transmitting
8 to 16bits	8 stages	0 to 8	0 to 8
17 to 32bits	4 stages	0 to 4	0 to 4

## 4.2.10. [TSPIxERR] (TSPI Error Flag Register)

Bit	Bit Symbol	After reset	Type	Function
31:4	-	0	R	Read as "0".
3	TRGERR	0	R	Trigger communication error flag 0: No error 1: Error exists  When a trigger is input, this bit is set if communication has not been started.
			W	0: Don't care 1: Flag is cleared  This bit is cleared by writing "1". Do not clear the bit during transmission/reception.
2	UDRERR	0	R	Underrun error flag 0: No error 1: Error exists  This bit is set when an underrun error occurs.
			W	0: Don't care 1: Flag is cleared  This bit is cleared by writing "1". Do not clear the bit during transmission/reception.
1	OVRERR	0	R	Overrun error flag 0: No error 1: Error exists  This bit is set when an overrun error occurs.
			W	0: Don't care 1: Flag is cleared  This bit is cleared by writing "1". Do not clear the bit during transmission/reception.
0	PERR	0	R	Vertical parity error flag 0: No error 1: Error exists  This bit is set when a vertical parity error occurs.
			W	0: Don't care 1: Flag is cleared  This bit is cleared by writing "1". Do not clear the bit during transmission/reception.

## 5. Example for use

For example, the details of starting / stopping operation for every communicate mode and transfer mode are shown in Table 5.1.

**Table 5.1 Transfer starting and stopping operation in each mode setting (Master)**

communication mode	Transfer mode	Communication start timing	Transfer stop timing
Full duplex	Continuously transfer	A transfer starts if the $[TSPIxCR1]<TRXE>$ bit is "1" and valid data exists in the transmit FIFO. Either condition can start transferring. When the receive buffer (receive FIFO or receive shift register) is full, next frame cannot be transferred. When reading data in the receive FIFO, if data in the receive shift register is automatically transferred to the FIFO, the shift register is determined as not full and a transfer is automatically restarted.	In transmission/reception, if $<TRXE>$ is set to stop, a transfer is stopped after a frame in progress is complete.
	Burst transfer (Include single transfer)	A transfer starts if $[TSPIxCR1]<TRXE>$ bit is "1" and valid data exists in the transmit FIFO. Either condition can start transferring. In burst transfer mode, if specified number of burst transfers are complete, the $[TSPIxCR1]<TRXE>$ bit returns to "0". If a burst transfer is attempted again, set "1" to $[TSPIxCR1]<TRXE>$ after the confirmation that $[TSPIxSR]<TSPIxSR>$ bit was returned to "0". If data in the transmit FIFO runs out during specified number of burst transfers, $TSPIxCS0/1/2/3$ stays asserted. A transfer automatically will restart when valid data is written to the transmit FIFO. If receive buffer (receive FIFO or receive shift register) is full, next frame cannot be transferred. At this time, $TSPIxCS0/1/2/3$ stays asserted. When reading the receive FIFO data, if data in the shift register is automatically transferred to the FIFO, the shift register is determined as not full and a transfer is automatically restarted.	In transmission/reception, if $<TRXE>$ is set to stop, a transfer is stopped after a frame in progress is complete
Transmission mode	Continuously transfer	A transfer starts if $[TSPIxCR1]<TRXE>$ bit is "1" and valid data exists in the FIFO. Either condition can start transferring.	In transmission/reception, if $<TRXE>$ is set to stop, a transfer is stopped after a frame in progress is complete
	Burst transfer (Include single transfer)	A transfer starts if $[TSPIxCR1]<TRXE>$ bit is "1" and valid data exists in the transmit FIFO. Either condition can start transferring. If a burst transfer is attempted again, set "1" to $<TRXE>$ after the confirmation that $[TSPIxSR]<TSPIxSR>$ bit was returned to "0". If data in the transmit FIFO runs out during specified number of burst transfers, $TSPIxCS0/1/2/3$ stays asserted. A transfer automatically will restart when valid data is written to the transmit FIFO.	In reception, if $t<TRXE>$ is set to stop, a transfer is stopped after a frame in progress is complete.
Receive mode	Continuously transfer	A reception starts if the receive buffer (receive FIFO or receive shift register) is not full. If the receive buffer is full, serial clock stops and next frame cannot be transferred. When reading the receive FIFO data, if data in the shift register is automatically transferred to FIFO, the shift register is determined as not full and a transfer is automatically restarted.	In reception, if $<TRXE>$ is set to stop, a transfer is stopped after a frame in progress is complete.

	<p>Burst transfer (Include single transfer)</p>	<p>A reception starts if receive buffer (receive FIFO or receive shift register) is not full. If the receive buffer is full, next frame cannot be transferred. When reading the receive FIFO data, if data in the shift register is automatically transferred to the FIFO, the shift register is determined as not full and a transfer is automatically restarted. TSPiXCS0/1/2/3 stays asserted until a transfer starts again. If a burst transfer is attempted again, set "1" to &lt;TRXE&gt; after <b>[TSPiXSR]</b>&lt;TSPISUE&gt; bit was returned to "0".</p>	<p>In reception, if &lt;TRXE&gt; is set to stop, a transfer is stopped after a frame in progress is complete.</p>
--	---	--	---

## 6. Precautions

- In case of the product which does not have TSPIxCS0/1/2/3 terminals or TSPIxCSIN terminal, please use SIO mode that is not using these terminals.
- Do not access the address that is not assigned register.

## 7. Revision History

**Table 7.1 Revision History**

Revision	Date	Description
1.0	2017-09-11	First release
2.0	2018-03-20	<p>-1.Outline: Corrected "(synchronous serial communication)" to "(Serial Peripheral Interface)". Modified description in DMA request in Table1.1/1.2/1.3/1.4. Modified "Minimum idle time..." to "Idle time..." in Table1.1/1.3 Modified TSPIxCS0/1/2/3 deassertion delay "final"--&gt;"last" in Table1.1.</p> <p>-2.Configuration: Added description "The block diagram...are shown". Modified "...TSPIx" to "...TSPI" in name of Figure2.1. Modified connection of TSPIxCS0/TSPIxCSIN in Figure2.1, Added Note.</p> <p>-3.1.3. Start and stop transfer: Added "continuously transfer".</p> <p>-3.3.1.1. Transmission clock: Deleted "transfer clock...clock(Note).", Corrected "refer" to "refer to" in Note, Corrected condition of Table 3.1.</p> <p>-3.3.4.1. Full duplex communication mode:Corrected "single" to "continuously"</p> <p>-3.3.4.2. Transmit mode: Corrected "single" to "continuously".</p> <p>-3.3.4.3. Receive mode: Corrected "single" to "continuously".</p> <p>-3.3.5. Transfer mode: Corrected "two modes" to "three modes", Added "continuously transfer". Added "continuously transfer can ...the number of transfer frames".</p> <p>-3.3.5.1.Single transfer: Added "The one frame ...single transfer",Modified "if..." to "when...".</p> <p>-3.3.5.2. Burst transfer: Added "TSPIxCS0/1/2/3 is...frame is completed".</p> <p>-3.3.6. Data sampling timing: Modified and added Figure3.16, added figure3.17/3.18/3.19.</p> <p>-3.3.7.1. Polarity of TSPIxCS0/1/2/3 signal and generation timing: Added "(Example for 2nd edge sampling)" to title of Figure 3.20.</p> <p>-3.3.7.3. TSPIxTXD Output during Idle: Added Note under Table3.3, Added "(2nd edge)" in Figure 3.21/3.22. Added condition in Figure3.22 Modified Figure 3.22.</p> <p>-3.3.8. Communication control by trigger: Added description.</p> <p>-3.3.9. Interrupt Request: Corrected signal name in Figure3.23.</p> <p>-3.3.9.1. Transmit Completion Interrupt/Receive Completion Interrupt: Modified "Both in the"--&gt;"Each", Added "continuously transfer", Deleted "when".</p> <p>-3.3.10. DMA request: Added description.</p> <p>-3.3.10.1. Transmit DMA request: Modified "transmit DMA" to "burst transmit DMA".</p> <p>-3.3.10.2.Receive DMA request: Modified "receive DMA" to "burst receive DMA".</p> <p>-4.1.Register List: Added ch10,ch11, and Type3 addresses. Delete "Reserved" descriptions and Note1.</p> <p>-4.2.1.[<b>TSPIxCR0</b>]: Modified "If" to "When", "software...initialized" to "By software ....initialized. (Table4.1) ", and deleted "The control...register)." in &lt;SWRST&gt;. Corrected "&lt;VPERR&gt;" to "&lt;PERR&gt;" in Tabel 4.1.</p> <p>-4.2.2.[<b>TSPIxCR1</b>]: Added "(Note4)", modified "will become disable" to "will disable" and "is become disable" to "will disable" in &lt;TRXE&gt;. Added "(Note5)" in &lt;CSSEL&gt;, Corrected Note2, Added Note4,5.</p> <p>-4.2.3.[<b>TSPIxCR2</b>]: Added "(Note1)" in &lt;TIDLE&gt;, Added "(Note2)"in &lt;RXDLY&gt;. Modified "(Note)" to "(Note3)" in &lt;TIL&gt;&lt;RIL&gt;. Corrected "recieve" to "receive" in &lt;INTRXFE&gt;. Corrected "Receive" to "Error", added description in &lt;INTERR&gt;. Corrected to "When continuously...completed." in &lt;INTTXWE&gt;.</p>

		<p>Corrected to "When continuously...completed." in &lt;INTRXWE&gt;.          Added Note1, Note2.          -4.2.5. <b>[TSPIxBR]</b>: Added Note.          -4.2.6. <b>[TSPIxFMTR0]</b>:          Modified to "This setup...of &lt;FINT&gt; occurs." in &lt;FINT&gt;.          Corrected to "Polarity of serial clock" to "Polarity of idle period of serial clock (Note2)" in &lt;CKPOL&gt;.          Modified "Minimum idle time" to "Idle time" and to "CS deassertion...in master mode." in &lt;CSINT&gt;.          Deleted "Valid...mode.", modified description in &lt;CSSCKDL&gt;/&lt;SCLCSDL&gt;.          -4.2.9. <b>[TSPIxSR]</b>:          Corrected "3. In the single..." to "3. In the continuously..." modified/ added description in &lt;TSPISUE&gt;.          Modified description in &lt;TXEND&gt;, modified description in &lt;RXEND&gt;.          -4.2.10. <b>[TSPIxERR]</b>: Corrected "&lt;VPERR&gt;" to "&lt;PERR&gt;"          - 5.Example for use:          Corrected "Single" to "Burst", "Burst transfer" to "Burst transfer (Include single transfer)".          -6.Precautions: Deleted "●When using ...to confirm which trigger can use."</p>
3.0	2019-02-28	<p>-Trademark          Modified description          - 1, Outline          Deleted "Data is sampled with 1st edge" from Data sampling timing in Table1.2          Deleted "Trigger error interrupt" from Interruption in Table 1.2 and Table 1.4          Added "Output level of TSPIxTXD when underrun error occurred" in Table1.4          Modified fsys to f<sub>clk</sub> of Final bit hold time of a TSPIxTXD pin in Table1.4.          Added Note in Table1.4.          - 2. Configuration          Modified fsys to f<sub>clk</sub> (Note) and signal name of f<sub>clk</sub> in Table2.1          Added reference manual of f<sub>clk</sub> and Note in Table2.1.          - 3.3.1 Transfer clock          Tile was modified from Transmission clock to Transfer clock          - 3.3.1.1 Master operation          Modified parameter of (1/x),          Modified the formula in case of <b>[TSPIxCR2]</b>&lt;RXDLY&gt;=0          Modified fsys to f<sub>clk</sub> of formula          Added Note2 in Table3.1 And modified Note to Note2          - 3.3.1.2 Slave operation          Added the formula of slave operation          - 3.3.2.1 SPI mode          Modified "Master device operation" to "Master operation"          - 3.3.2.2 SIO mode          Modified "Master device operation" to "Master operation", Added Note          - 3.3.5.3 Continuously transfer          Deleted Note1 and Note2          - 3.3.6 Data sampling timing          Added Table 3.3, Table 3.4          Deleted Data sampling timing of SPI mode(slave) 1st edge from Figure 3.17          Modified Data sampling timing of SIO mode(master) 2nd edge from Figure 3.18          Modified Data sampling timing of SIO mode(slave) 2nd edge from Figure 3.19          3.3.9.1 Transmit Completion Interrupt/Receive Completion Interrupt          Added explanation of slave device operation          4.2.3 <b>[TSPIxCR2]</b>          Deleted Note1) of function explanation in &lt;TIDLE[1:0]&gt;          Added Note1) of function explanation in &lt;RXDLY&gt;          Modified &lt;INTERR&gt; of function explanation          Deleted old Note1), Shifted old Note2) to Note1), Added new Note2)          4.2.5 <b>[TSPIxBR]</b> (TSPI Baud Rate Register)          Deleted Note          4.2.6 <b>[TSPIxFMTR0]</b>          Added (Master operation) in &lt;CKPHA&gt; function explanation field,          4.2.7 <b>[TSPIxFMTR1]</b></p>

		Modified <EHOLD> of function explanation. Added Note1, Shifted old Note1 to Note2. 4.2.9 <b>[TSPIxSR]</b> (TSPI Status Register) Modified headline of Table 4.2
3.1	2021-10-15	3.3.8. Communication control by trigger (3) Receiving trigger Added description. 4.2.6. In function description of <b>[TSPIxFMTR0]</b> FINT[3:0] and CSINT[3:0], corrected typos.

## 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.**