# 32-bit RISC Microcontroller

# TXZ/TXZ+ Family

## Reference Manual
## Serial Peripheral Inteface
## (TSPI-C)

## Revision 3.2

**2021-10**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

# Contents

# List of Figures



# List of Tables

# Preface

**Related document**

| Document name |
|---|
| Clock Control and Operation Mode |
| Exception |
| Input/Output Ports |
| Product Information |

## Conventions

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

## 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 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. |
| | 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 transfer, 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.<br>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.<br>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.<br>1 x TSPIxSCK cycle to 16 x TSPIxSCK cycles |
| | | TSPIxCS0/1/2/3 deassertion delay | Time until TSPIxCS0/1/2/3 pin is deasserted from last data<br>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.<br>8 to 32bits (with no parity)<br>7 to 31bits (with parity) |
| | | Parity | Selection of with parity/without parity is possible.<br>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)<br>Reception: 8 steps (16bits), 4 steps (32bits) |
| | | Communication Operation mode | Full duplexes (transmission and reception), transmission, reception |
| | | Transfer mode | Single transfer (one burst transfer)<br>Burst transfer (2 to 255 times transfer)<br>Continuously transfer (No limit of transfer times specification) |
| | | Data sampling timing | Data is sampled with 2nd edge. |
| | | CS control | TSPIxCSIN<br>Polarity: Selection of positive logic/ negative logic is possible. |
| | Ganged Control | Interruption | Transmit interrupt (Transmit completion interrupt, Transmit FIFO interrupt)<br>Receive interrupt (Receive completion interrupt, Receive FIFO interrupt)<br>Error Interrupt (Vertical parity error interrupt, Overrun interrupt, Underrun interrupt) |
| | | Various status detection | TSPI modify status,<br>Transmit shift operation, Transmit completion,<br>Transmit FIFO fill level/ empty detection,<br>Receive operation, Receive completion, Receive FIFO fill level /full detection |
| | | DMA request | Transmit: Single DMA request, burst DMA request<br>Receive: Single DMA transfer, 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 is possible. |

**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.<br>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.<br>8 to 32bits (with no parity)<br>7 to 31bits (with parity) |
| | | Parity | Selection of with parity/without parity is possible.<br>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)<br>Reception: 8 steps (16bits), 4 steps (32bits) |
| | | Communication Operation mode | Full duplexes (transmission and reception), transmission, reception |
| | | Transfer mode | Single transfer (one burst transfer)<br>Burst transfer (2 to 255 times transfer)<br>Continuously transfer (No limit of transfer times specification) |
| | | Data sampling timing | Data is sampled with 1st edge.<br>Data is sampled with 2nd edge. |
| | Ganged Control | Interruption | Transmit interrupt (Transmit completion interrupt, Transmit FIFO interrupt)<br>Receive interrupt (Receive completion interrupt, Receive FIFO interrupt)<br>Error Interrupt (Vertical parity error interrupt, Trigger error interrupt) |
| | | Various status detection | TSPI modify status,<br>Transmit shift operation, Transmit completion,<br>Transmit FIFO fill level/ empty detection,<br>Receive operation, Receive completion, Receive FIFO fill level /full detection |
| | | DMA request | Transmit: Single DMA request, burst DMA request<br>Receive: Single DMA transfer, 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.<br>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.<br>8 to 32bits (with no parity)<br>7 to 31bits (with parity) |
| | | Parity | Selection of with parity/without parity is possible.<br>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)<br>Reception: 8 steps (16bits), 4 steps (32bits) |
| | | Communication Operation mode | Full duplexes (transmission and reception), transmission, reception |
| | | Transfer mode | Single transfer (one burst transfer)<br>Burst transfer (2 to 255 times transfer)<br>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)<br>Receive interrupt (Receive completion interrupt, Receive FIFO interrupt)<br>Error Interrupt (Vertical parity error interrupt, Overrun interrupt, Underrun interrupt) |
| | | Various status detection | TSPI modify status,<br>Transmit shift operation, Transmit completion,<br>Transmit FIFO fill level/ empty detection,<br>Receive operation, Receive completion, Receive FIFO fill level /full detection |
| | | DMA demand | Transmit: Single DMA request, burst DMA request<br>Receive: Single DMA transfer, burst DMA request |
| | Special Control | Final bit hold time of a TSPIxTXD pin | 2/fsys to 128/fsys |
| | | Output level of TSPIxTXD | Idle term : High, Low, a last bit data hold, Hi-z |
| | | | Underrun error occurrence : High, Low |
| | | Software reset | Reset by software is possible. |

# 2. Configuration

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

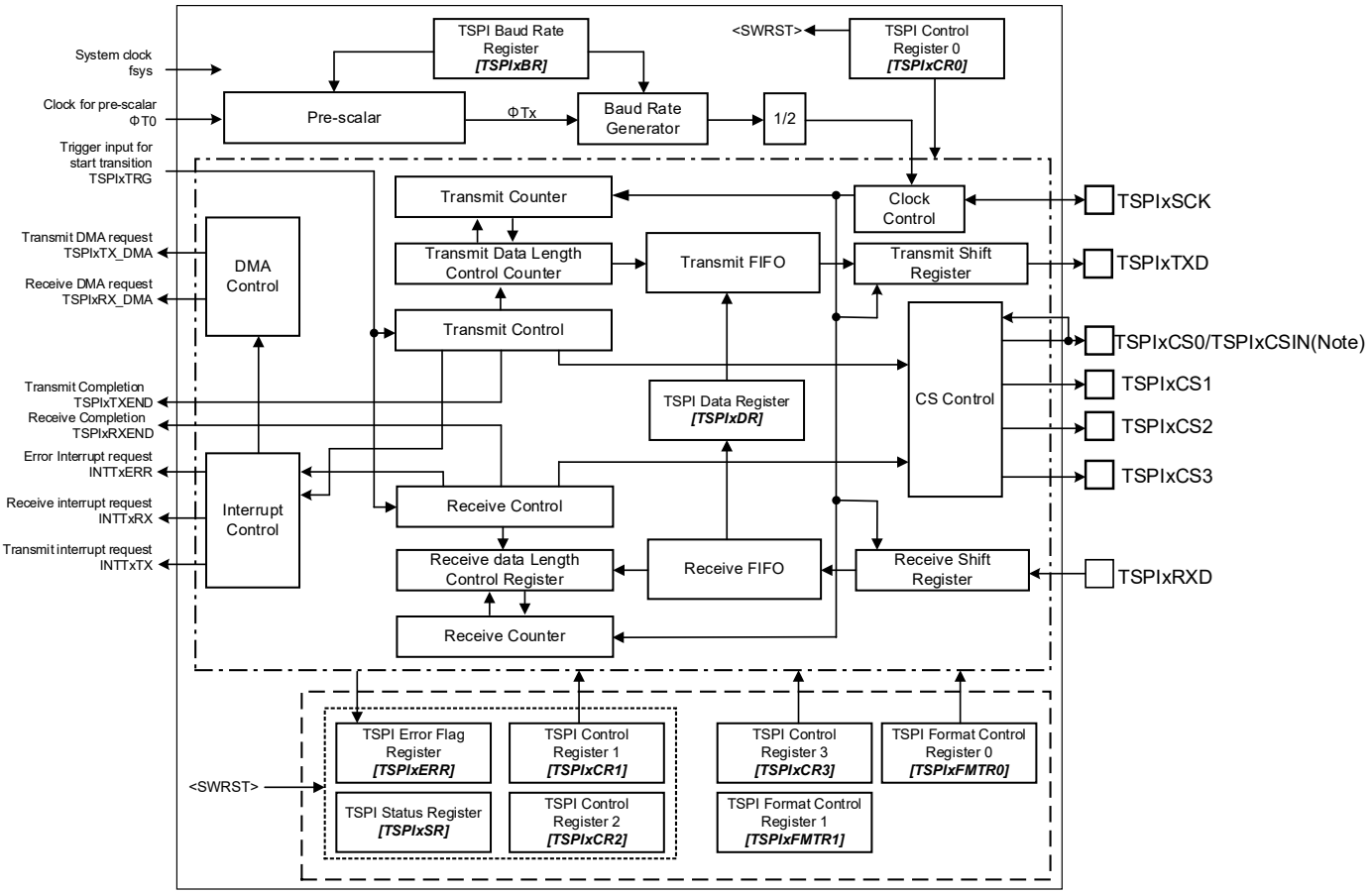


**Figure 2.1 Block diagram of TSPI**

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 | fsys | System clock | Input | Clock Control and Operation Mode |
| 2 | ΦT0 | Clock for prescaler | Input | Clock Control and Operation Mode |
| 3 | TSPIxSCK | Serial clock output/ Serial clock input | Input/Output | Input/Output Ports |
| 4 | TSPIxCS0 | Chip select 0 | Output | Input/Output Ports |
| 5 | TSPIxCS1 | Chip select 1 | Output | Input/Output Ports |
| 6 | TSPIxCS2 | Chip select 2 | Output | Input/Output Ports |
| 7 | TSPIxCS3 | Chip select 3 | Output | Input/Output Ports |
| 8 | TSPIxCSIN | Chip select input for slave operation | Input | Input/Output Ports |
| 9 | TSPIxTXD | Serial data of transmission | Output | Input/Output Ports |
| 10 | TSPIxRXD | Serial data of reception | Input | Input/Output Ports |
| 11 | INTTxTX | Transmit interrupt | Output | Exception |
| 12 | INTTxRX | Receive interrupt | Output | Exception |
| 13 | INTTxERR | Error interrupt | Output | Exception |
| 14 | TSPIxTRG | Trigger input for start communication | Input | Product Information |
| 15 | TSPIxTX_DMA | Transmit DMA request | Output | Product Information |
| 16 | TSPIxRX_DMA | Receive DMA request | Output | Product Information |
| 17 | TSPIxTXEND | Transmit Completion | Output | Product Information |
| 18 | TSPIxRXEND | Receive Completion | Output | Product Information |

# 3. Operation description

## 3.1. Basic operation

### 3.1.1. Clock supply

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

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

When stopping supply of a clock, please check that TSPI has stopped (***[TSPIxCR0]***<TSPIE>=0). 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.

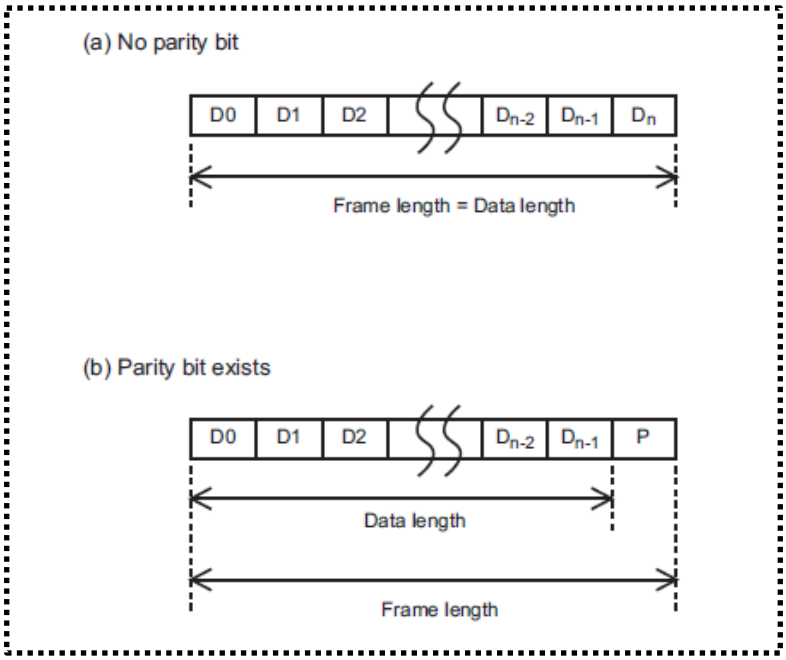


**Figure 3.1 Data format**

### 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 ***[TSPIxFMTR0]***<FL[5:0]>. 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.

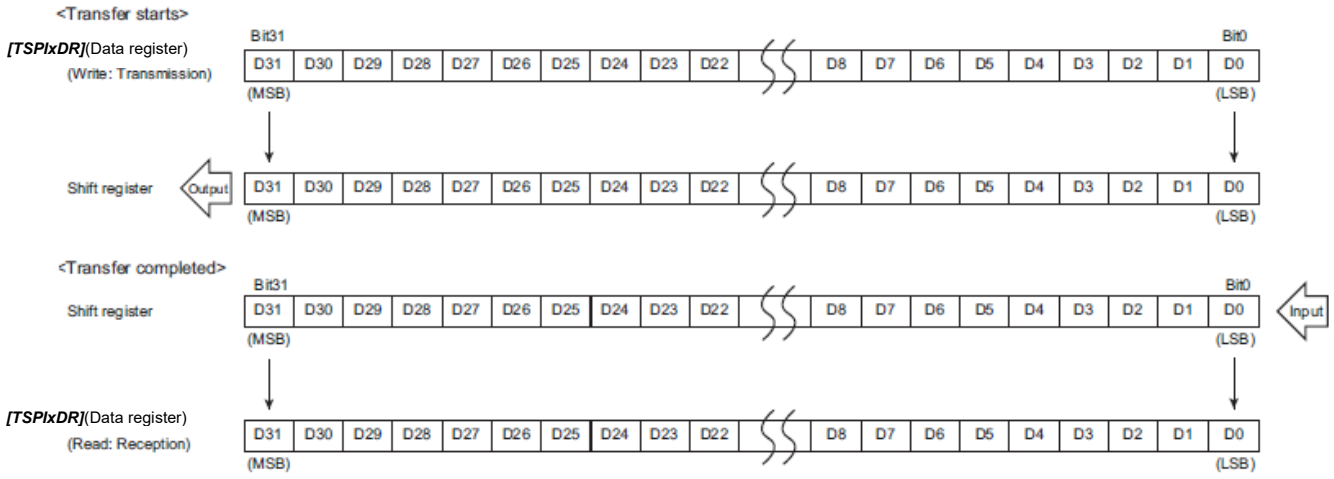


**Figure 3.2 MSB first (32-bit data without a parity bit)**

### (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.

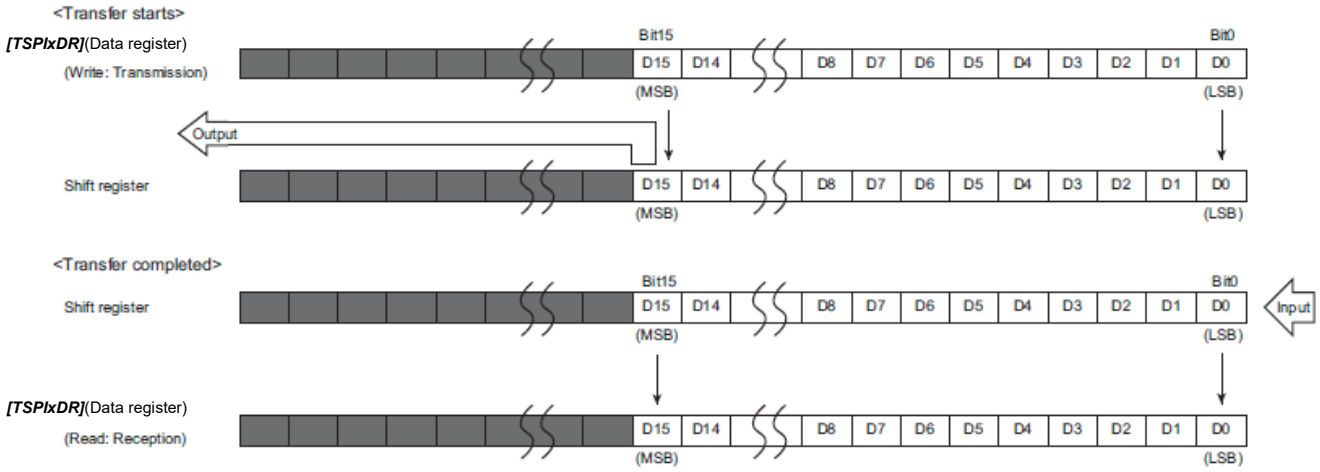


**Figure 3.3 MSB first (16-bit data without a parity bit)**

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

Figure 3.4shows 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.

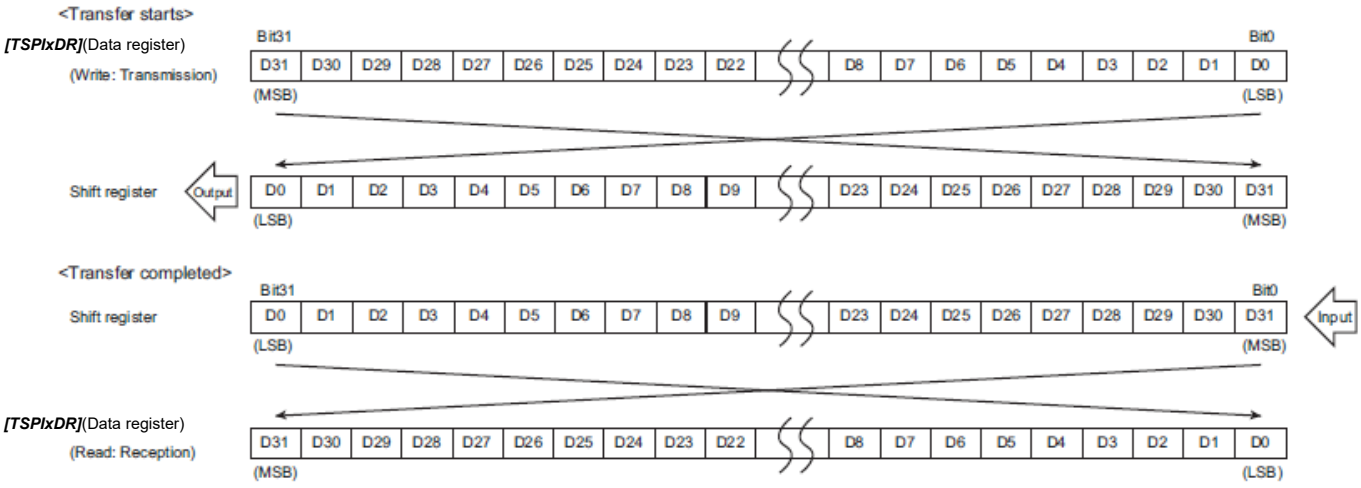


**Figure 3.4 LSB first (32-bit data without a parity bit)**

### (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.

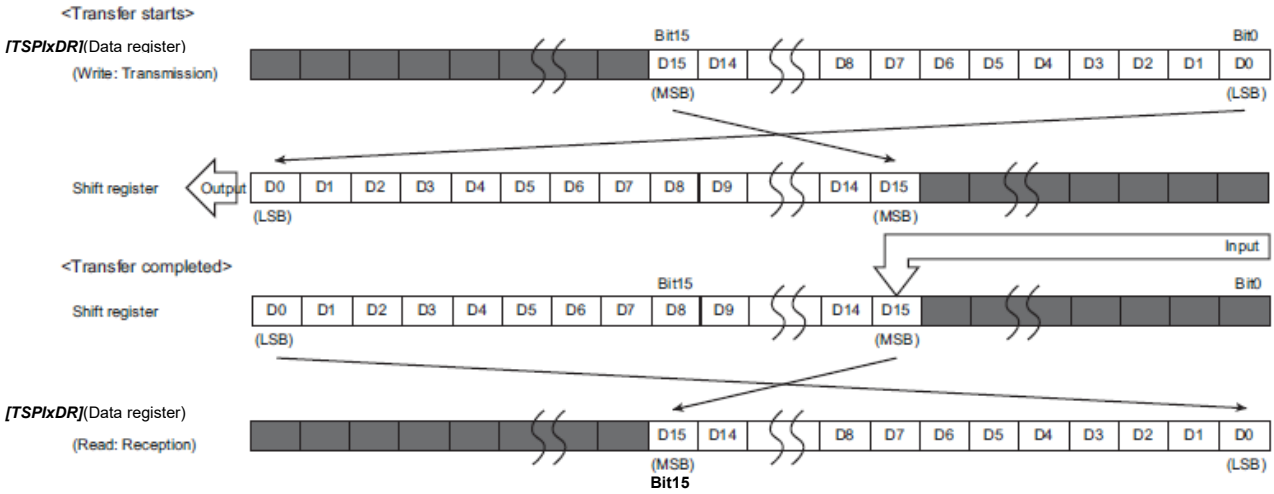


**Figure 3.5 LSB first (16-bit data without a parity bit)**

### 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 ***[TSPIxFMTR0]***<FL[5:0]>.

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.

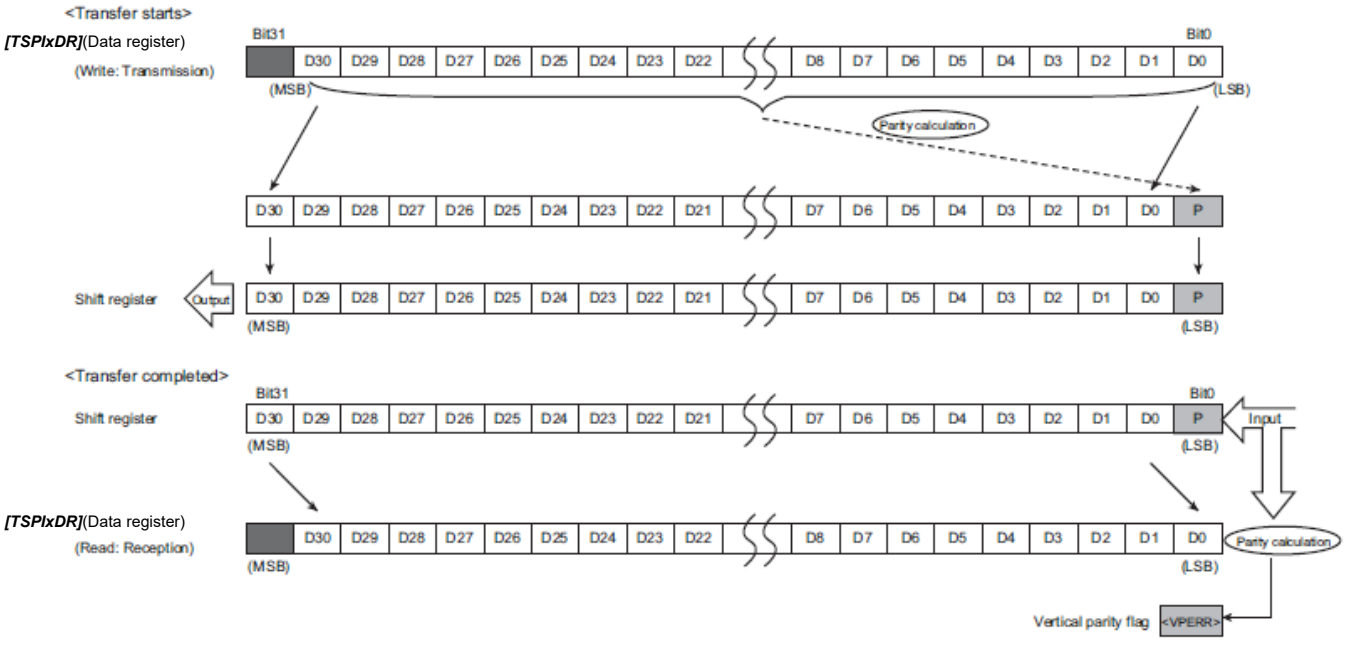


**Figure 3.6 MSB first (31-bit data with a parity)**

### (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.

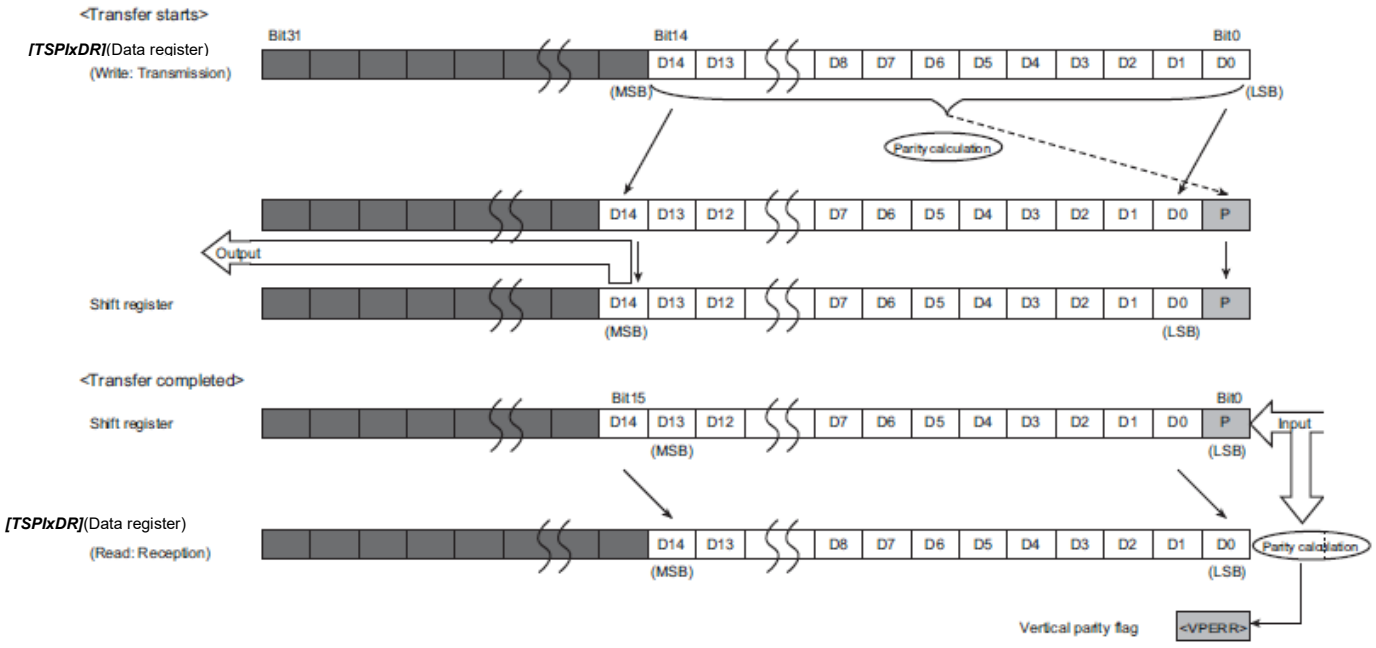


**Figure 3.7 MSB first (15-bit data with parity)**

### (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.

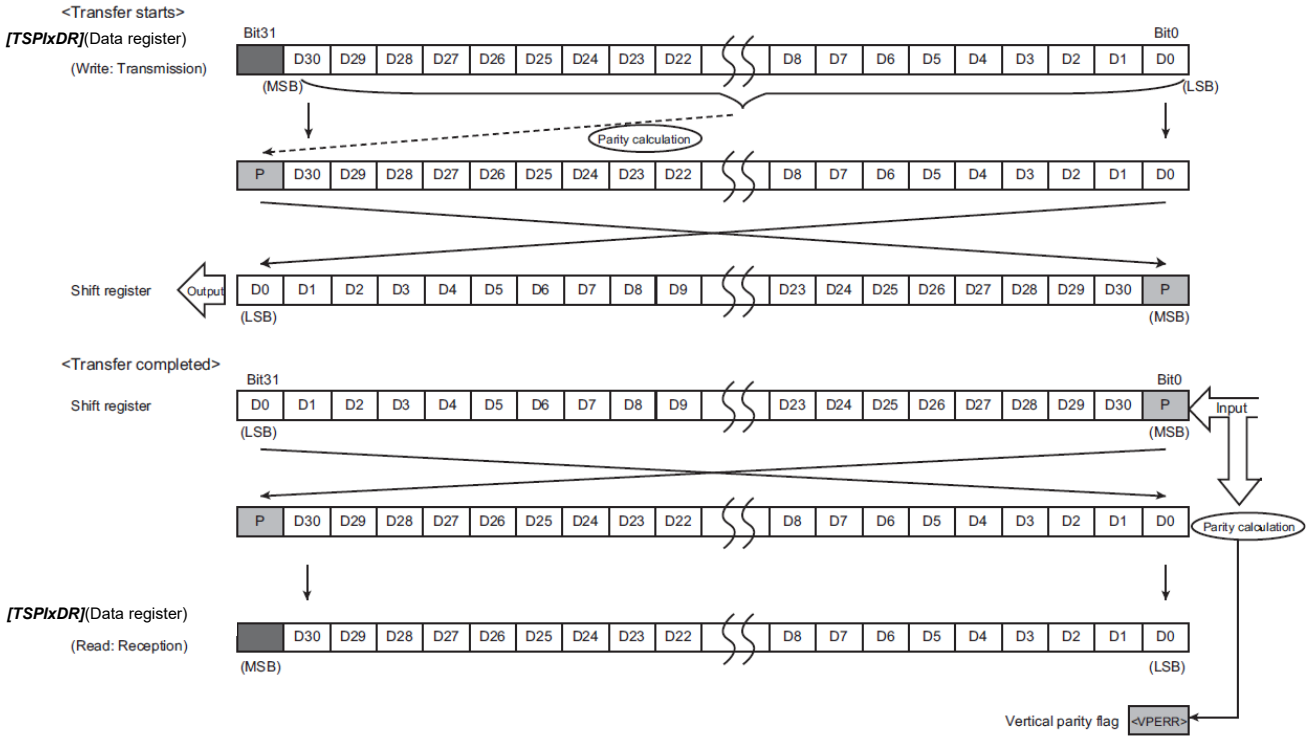


**Figure 3.8 LSB first (31-bit data with parity)**

### (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.

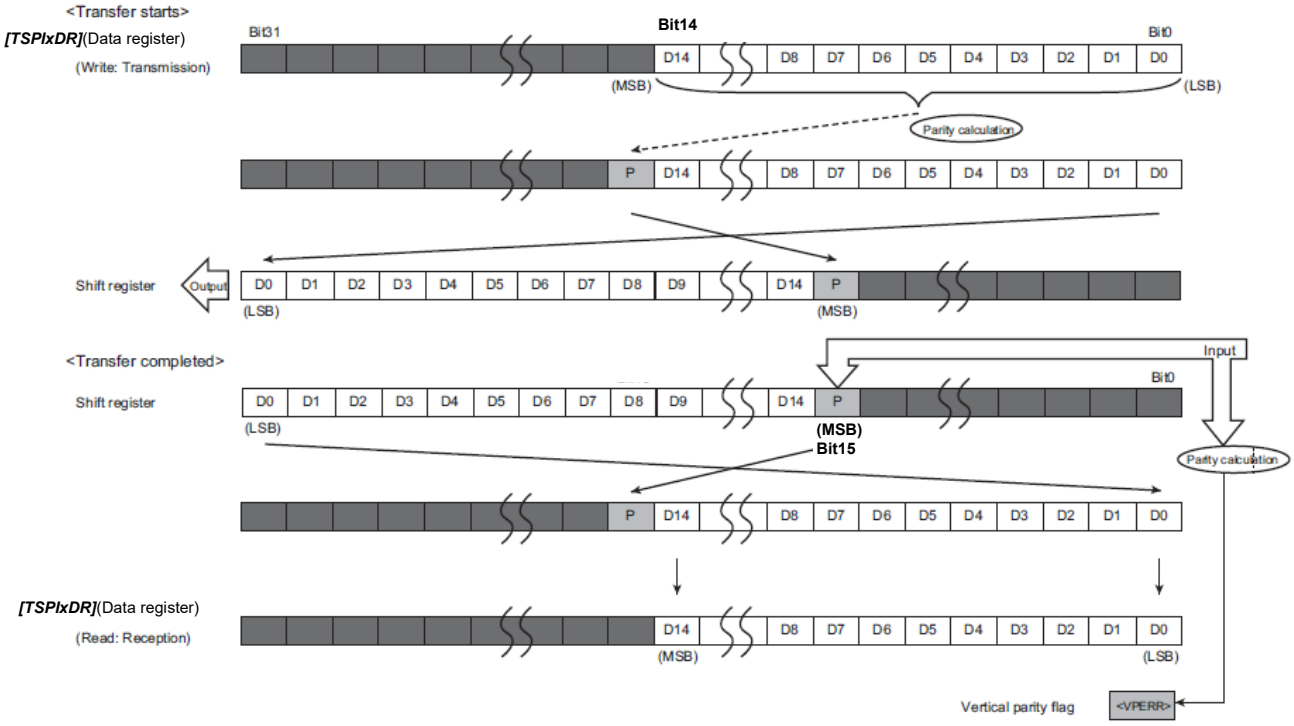


**Figure 3.9 LSB first (15-bit data with parity)**

## 3.3. Operation

### 3.3.1. Transfer clock

#### 3.3.1.1. Master Operation

The transfer clock generation circuit is shown Figure 3.10.

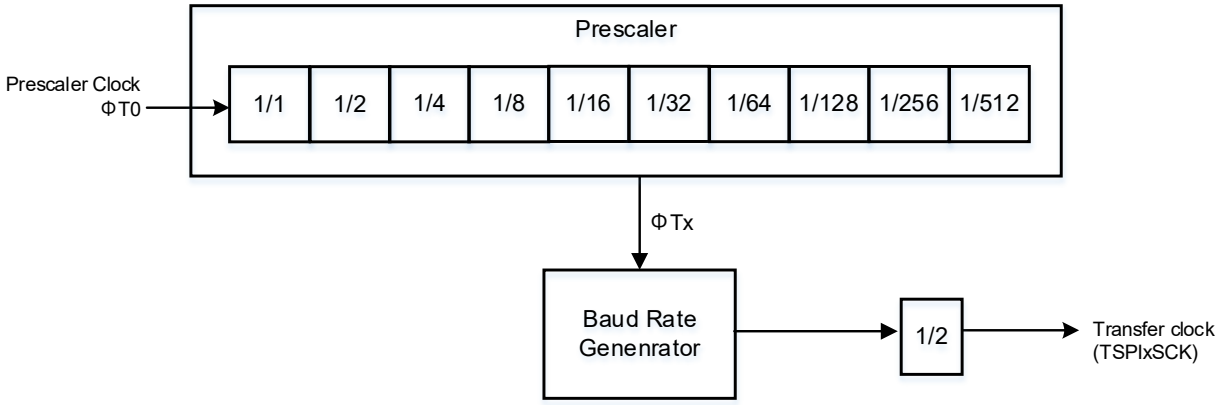


**Figure 3.10 Transfer clock generation circuit**

The prescaler dividing ΦT0 from 1/1 to 1/512 (ΦT0 to ΦT256). Divided clock can be selected by ***[TSPIxBR]***<BRCK>.

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

transfer clock = ΦT0 × ***[TSPIxBR]***<BRCK> (1/x) × ***[TSPIxBR]***<BRS> (1/N) × 1/2
(x= 1,2, 4,8,16 to 256,512 N= 1,2,3, 4, to16)

At this time, please keep below condition

The case of ***[TSPIxCR2]***<RXDLY>=0, 1/2 ×fsys = transfer clock
The case of ***[TSPIxCR2]*** <RXDLY>=1, 1/4 ×fsys ≥ transfer clock
and
fsys≥ΦT0

**Table 3.1 Example of Transfer clock**

| *[TSPIxCR2]*<RXDLY>(Note1) | 0 | 0 or 1 | | | | | |
|---|---|---|---|---|---|---|---|
| *[TSPIxBR]*<BRS[3:0]> | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Dividing / ΦT0[MHz] | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
| 160 | (80) | (40) | (26.7) | 20 | 16 | 13.3 | 11.4 |
| 140 | (70) | (35) | 23.3 | 17.5 | 14 | 11.7 | 10 |
| 120 | (60) | (30) | 20 | 15 | 12 | 10 | 8.6 |
| 100 | (50) | 25 | 16.7 | 12.5 | 10 | 8.3 | 7.1 |
| 80 | (40) | 20 | 13.3 | 10 | 8 | 6.7 | 5.7 |
| 60 | (30) | 15 | 10 | 7.5 | 6 | 5 | 4.3 |
| 40 | 20 | 10 | 6.7 | 5 | 4 | 3.3 | 2.9 |
| 20 | 10 | 5 | 3.3 | 2.5 | 2 | 1.7 | 1.4 |

Note1: The setting value of this register varies depending on the product. For the setting value, refer to "Product Information" of the reference manual.

Note2: Transfer clock example for ***[TSPIxBR]***<BRCK[3:0]>=0000 and ΦT0=fsys.

Note3: Set the transfer clock to within maximum transfer clock which is determine depending on products, and 25MHz or less.

Note4: Combination of the dividing value and the ΦT0 within gray in Table3.1 cannot be selected. For the maximum frequency of the transfer clock depending of products, refer to the electrical characteristics of the data sheet.

**Table 3.2 Example of sys/ΦT0/ΦTx/ transfer clock and usability**

Condition: ***[TSPIxCR2]***<RXDLY>=1,fsys/Transfer clock ≥ 4
Transfer clock=ΦTx/2, Product Transfer clock ≤ 20MHz

| fsys(MHz) | ΦT0(MHz) | ΦTx(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>=0 ,fsys/ Transfer clock ≥ 2
Transfer clock=ΦTx/2, Product Transfer clock ≤ 20MHz

| fsys(MHz) | ΦT0(MHz) | ΦTx(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

#### 3.3.1.2. Slave Operation

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

1/2 × fsys ≥ transfer clock

### 3.3.2. Communication mode

The TSPI has two communication modes: SIO mode and SPI mode. Communication mode is specified by ***[TSPIxCR1]***<TSPIMS>.

#### 3.3.2.1. SPI mode

When write "0" to ***[TSPIxCR1]***<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 3.3 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.

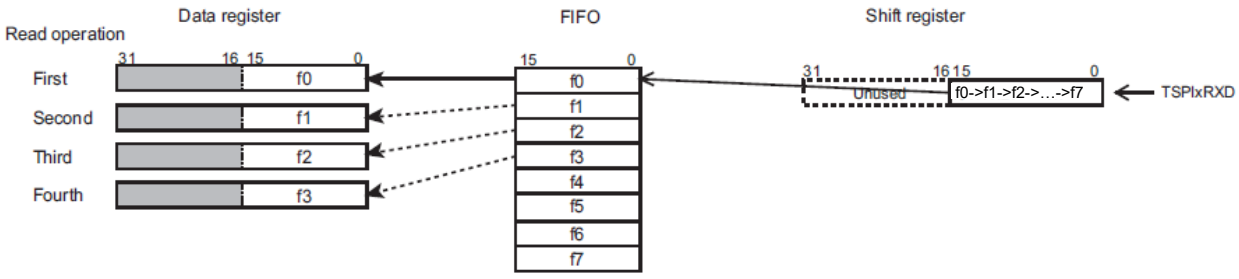


**Figure 3.11 Operation in 7 to 16-bit data length**

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.

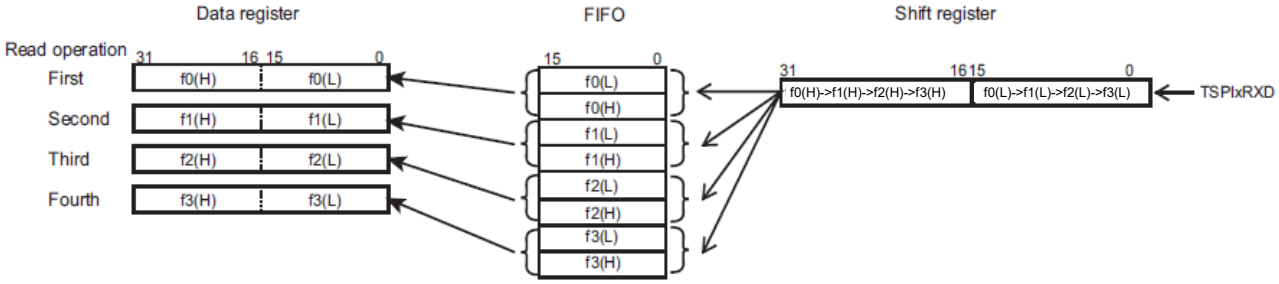


**Figure 3.12 Operation in 17 to 32-bit data length**

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)

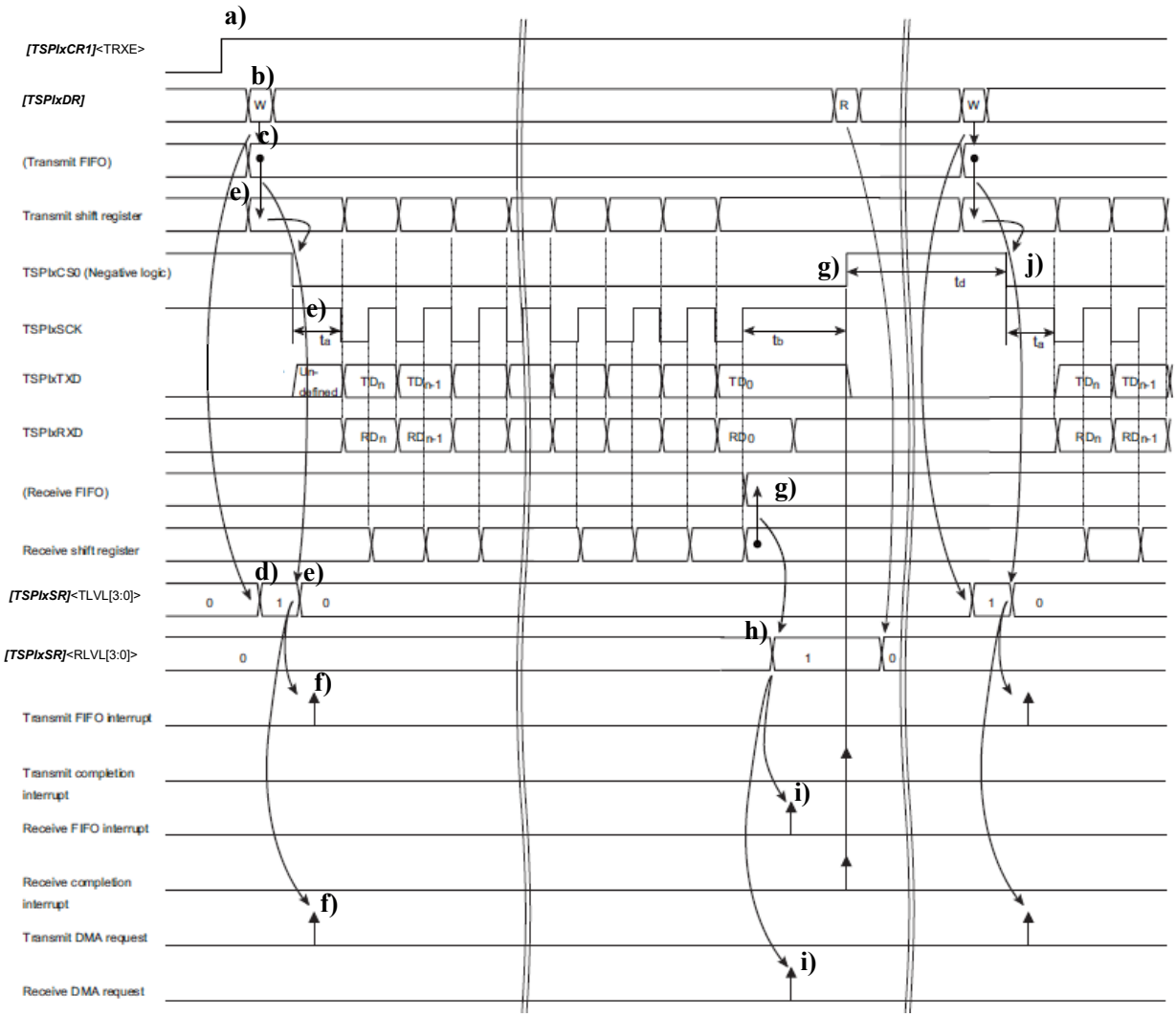


**Figure 3.13 Operation example of full duplex communication**

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, TSPIxSCK 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, TSPIxCS0 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)

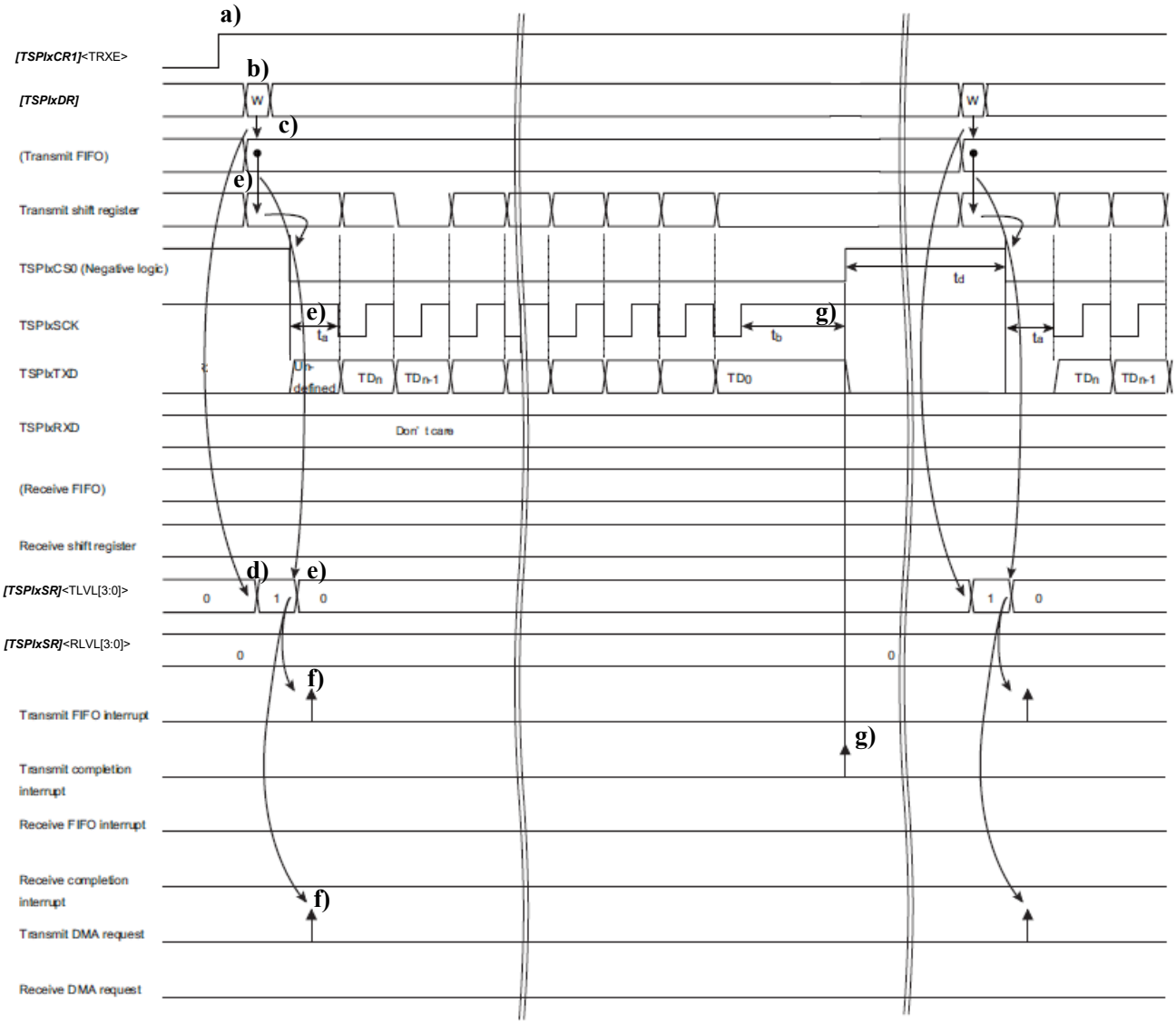


**Figure 3.14 Operation example of transmit mode**

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 ($t_d$) 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.

### 3.3.4.3. 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)

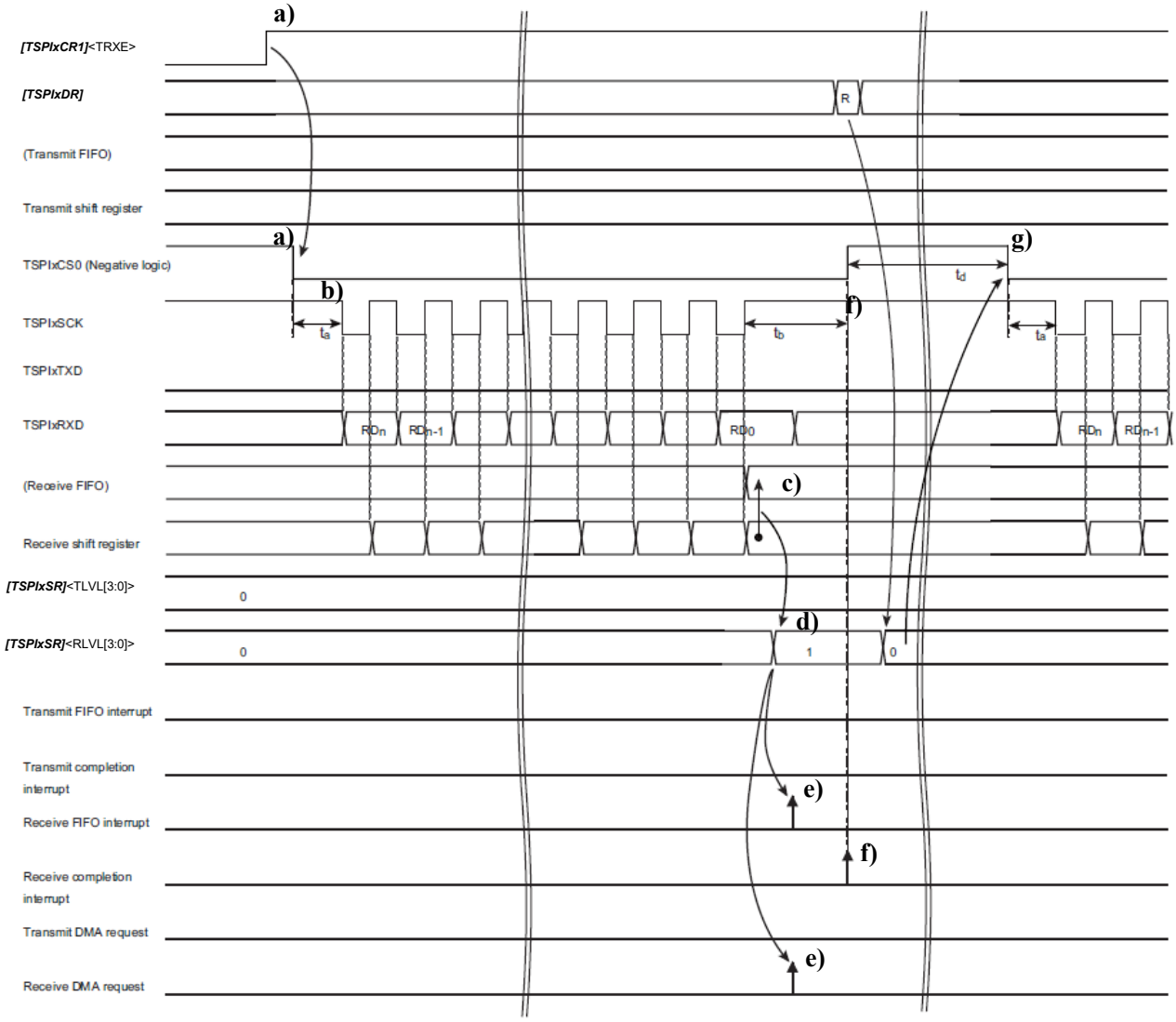


**Figure 3.15 Operation example in receive mode**

a) Write "1" to ***[TSPIxCR1]***<TRXE> to enable the communications. Since receive FIFO is not full, TSPIxCS0 is immediately asserted and serial clock transfer starts.
b) After a serial clock delay time ($t_a$) specified by ***[TSPIxFMTR0]***<CSSCKDL> has elapsed, serial clock starts outputting from TSPIxSCK.
c) 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.
d) Since one stage data is buffered to the receive FIFO, ***[TSPIxSR]***<RLVL> becomes "1".
e) Since ***[TSPIxSR]***<RLVL> changed to "1" from "0", a receive FIFO interrupt (or receive DMA request) occurs.
f) 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.
g) 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 ***[TSPIxCR1]***<FC[7:0]>.

#### 3.3.5.1. Single transfer

Single transfer is the mode that can transfer one frame. The one frame 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> = 1is set, data is sampled by the second edge. And when, ***[TSPIxFMTR0]***<CKPHA> = 0, by the first edge.

Table 3.4 is shown Usability of communication mode and data sampling timing. And Table 3.5 is shown Data capture timing.

**Table 3.4 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 3.5 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 |

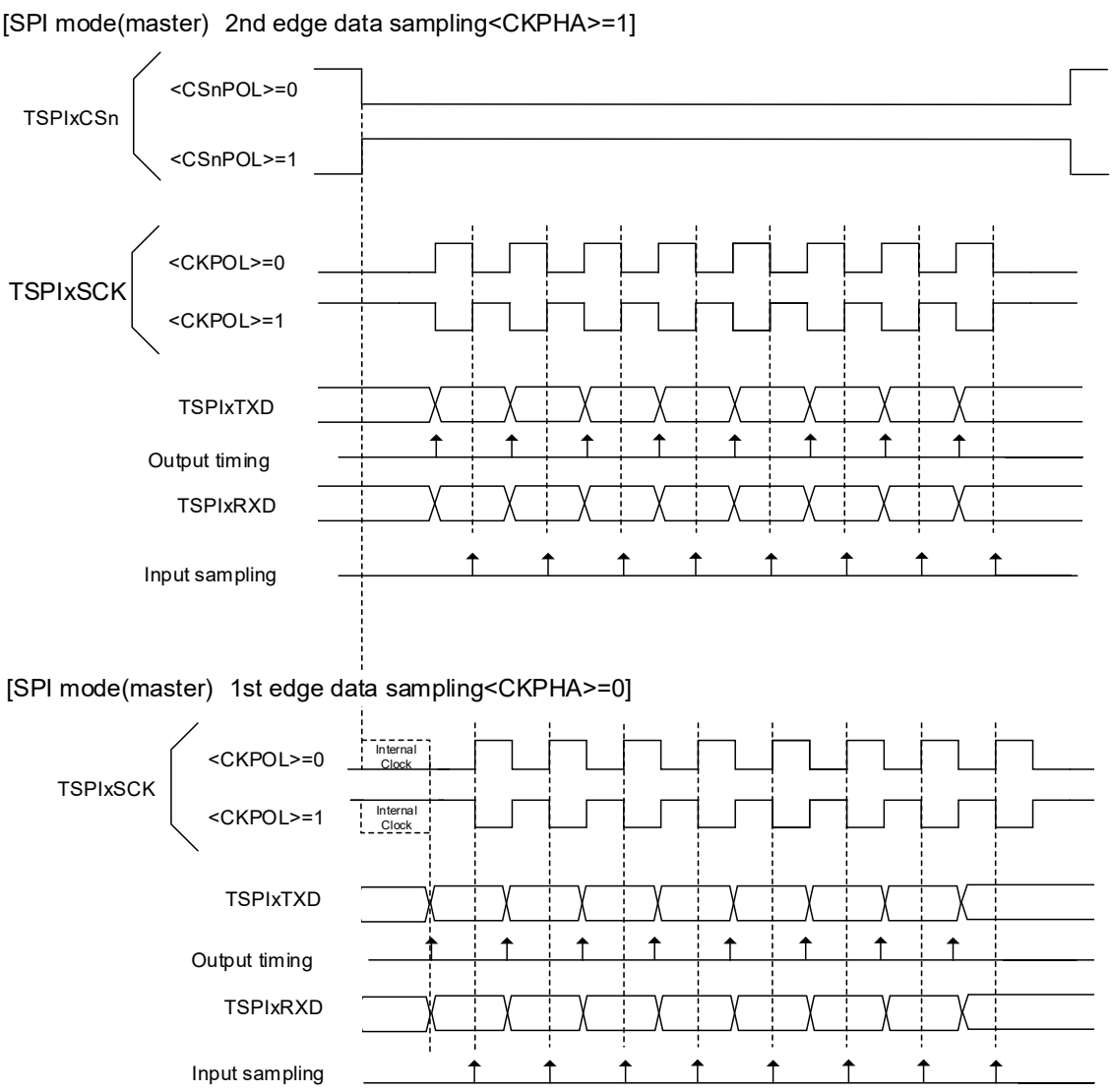


**Figure 3.16 Data sampling timing of SPI mode (master)**

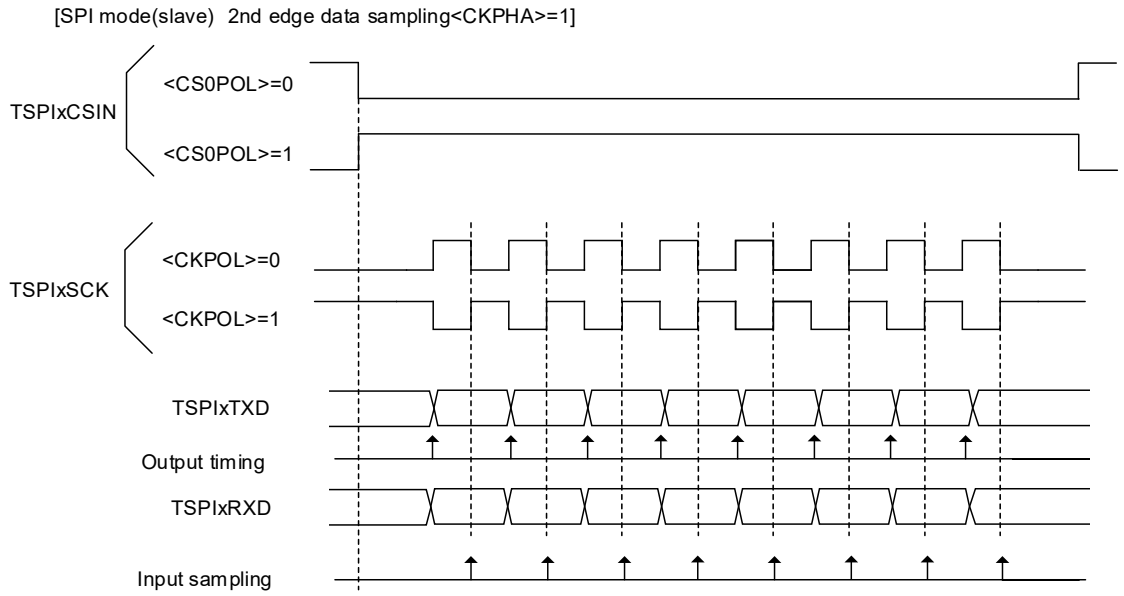


**Figure 3.17 Data sampling timing of SPI mode (slave)**

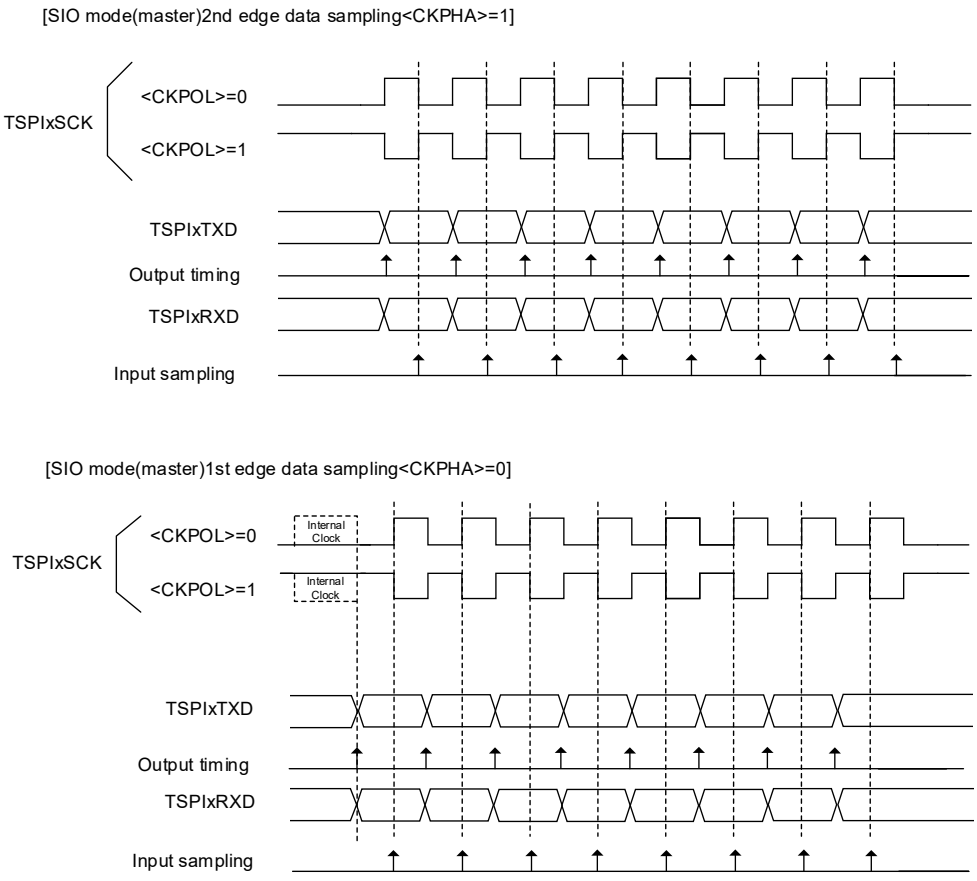


**Figure 3.18 Data sampling timing of SIO mode (master)**

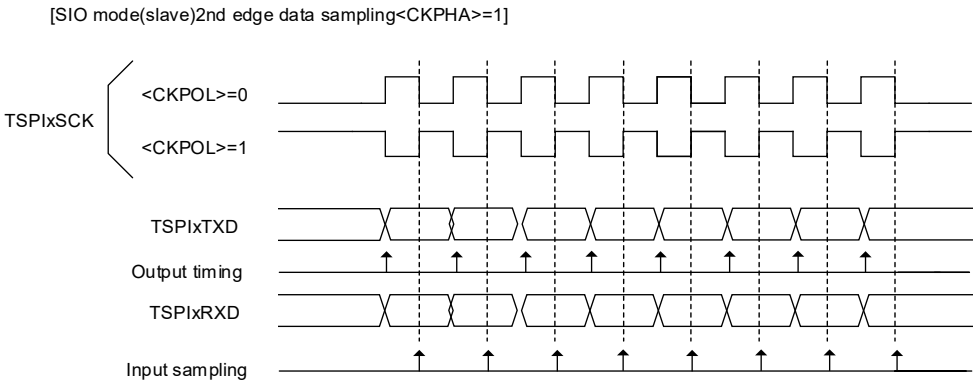


**Figure 3.19 Data sampling timing of SIO mode (slave)**

## 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>.

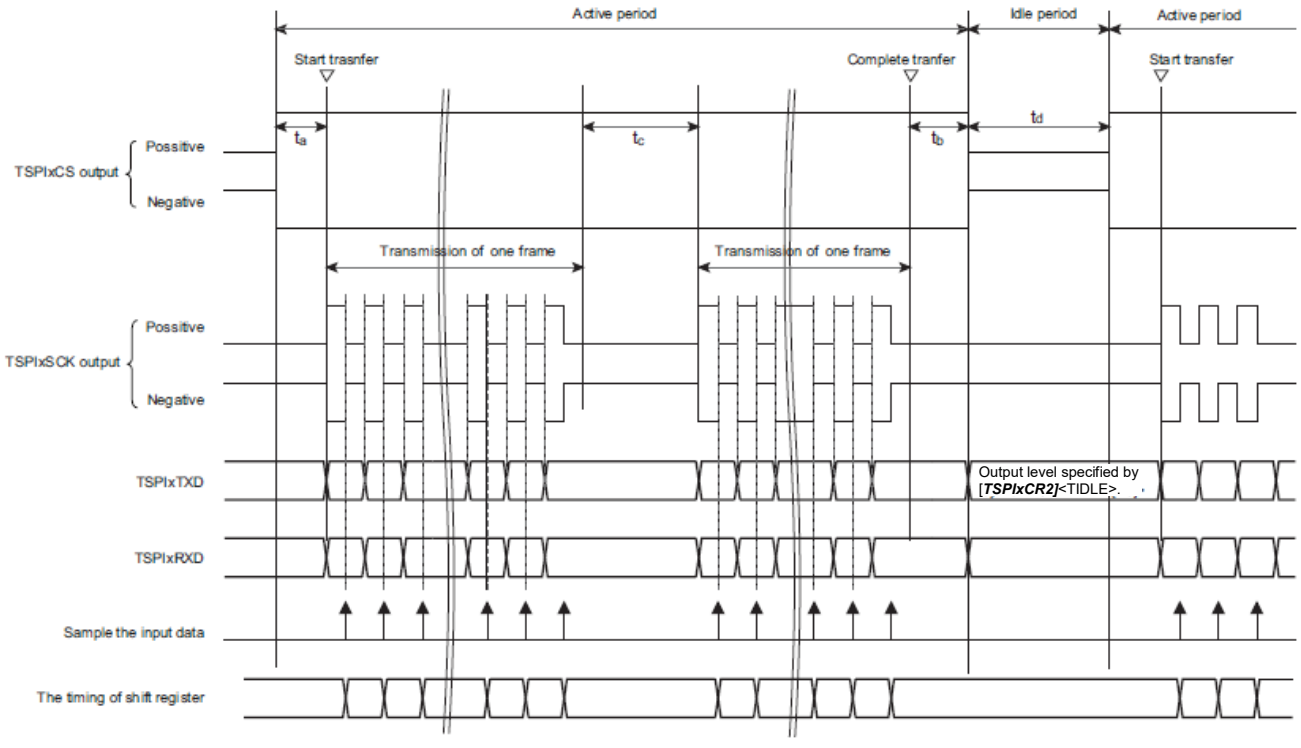


**Figure 3.20 Transfer format and timing adjustment (Example for 2nd edge sampling)**

### 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 a 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 3.6 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

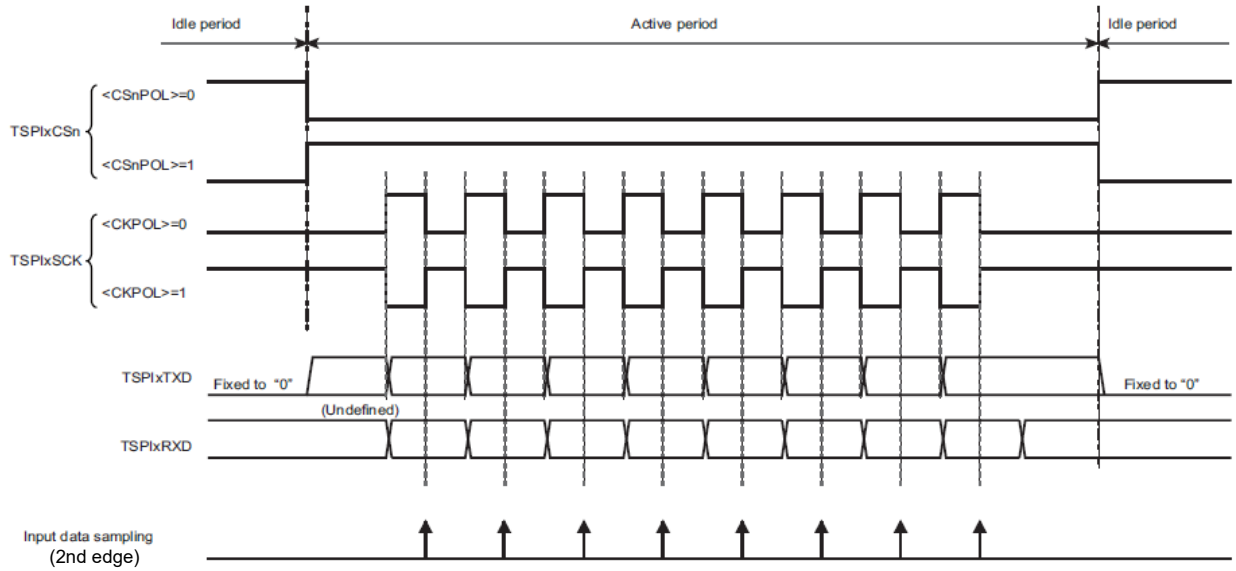


**Figure 3.21 Idle state in SPI mode and the transmit pin status**

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

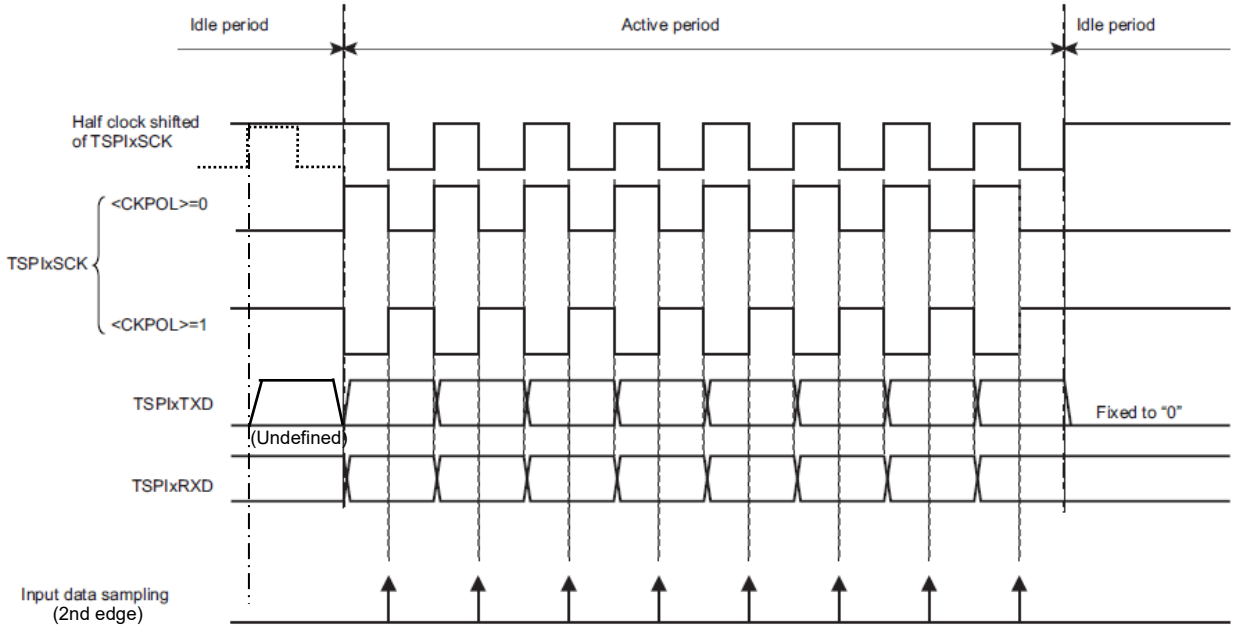


**Figure 3.22 Idle state in SIO mode and the transmit pin status**

## 3.3.8. Communication control by trigger

In the master operation, when ***[TXPIxCR1]***<TRGEN> is set to “1”, the function of the starting communication by generating of a trigger can be used.

If “1” is set as ***[TSPIxCR1]***<TRGEN>, when communication is not performed, it will be waiting for a trigger. If a trigger is inputted, it will be automatically set to ***[TSPIxCR1]***<TRXE>=1, and communication will be started. After communication is completed, it will be automatically set to ***[TSPIxCR1]***<TRXE>=0 and will be waiting for a trigger.

When you use a trigger start function, please set it as ***[TSPIxCR1]***<TRXE>=0.Moreover, ***[TSPIxCR1]***<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 ***[TSPIxCR1]***<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 "3.3.9.3Error 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 ***[TSPIxCR1]***<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 "3.3.9.3Error 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 3.7 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 | |

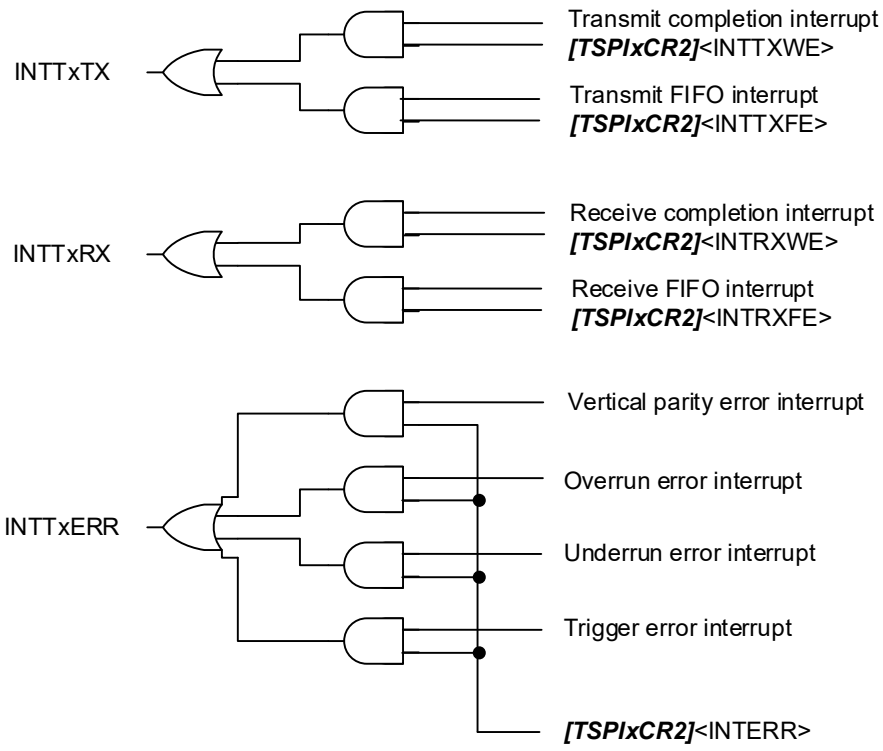


**Figure 3.23 Circuit of interrupt request**

### 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 transmit 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.

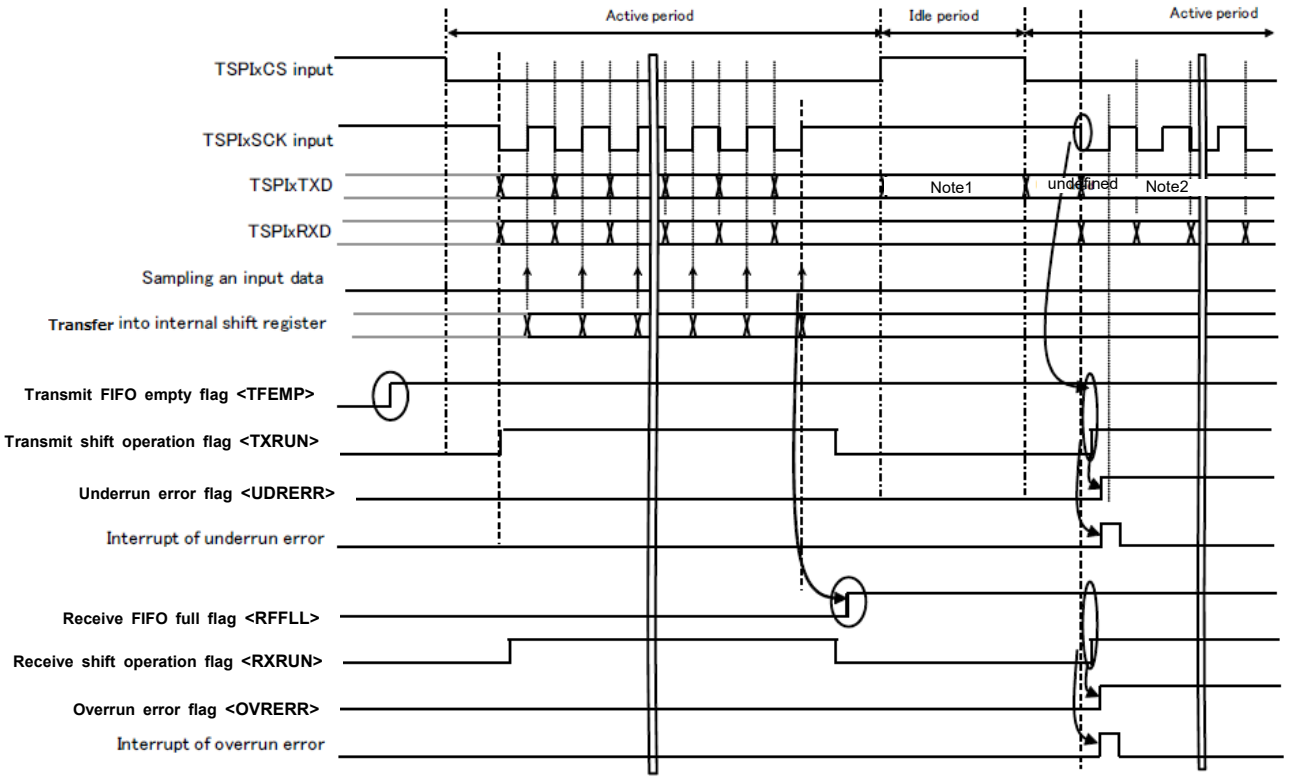


**Figure 3.24 Overrun error and underrun error**

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

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

| Register name | | Address(Base+) |
|---|---|---|
| TSPI Control Register 0 | ***[TSPIxCR0]*** | 0x0000 |
| TSPI Control Register 1 | ***[TSPIxCR1]*** | 0x0004 |
| TSPI Control Register 2 | ***[TSPIxCR2]*** | 0x0008 |
| TSPI Control Register 3 | ***[TSPIxCR3]*** | 0x000C |
| TSPI Baud Rate Register | ***[TSPIxBR]*** | 0x0010 |
| TSPI Format Control Register 0 | ***[TSPIxFMTR0]*** | 0x0014 |
| TSPI Format Control Register 1 | ***[TSPIxFMTR1]*** | 0x0018 |
| TSPI Data Register | ***[TSPIxDR]*** | 0x0100 |
| TSPI Status Register | ***[TSPIxSR]*** | 0x0200 |
| TSPI Error Flag Register | ***[TSPIxERR]*** | 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)<br>Software reset occurs by writing "10" and then "01".<br><br>By software reset, the transfer operation under execution is forcibly terminated and the value of the control register other than the transfer setting is initialized. (Table 4.1) |
| 5:1 | - | 0 | R | Read as "0". |
| 0 | TSPIE | 0 | R/W | TSPI operation control<br>0: Stop<br>1: Operation<br><br><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.<br><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><br><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><br><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)<br>0: Not used<br>1: A trigger is valid |
| 14 | TRXE | 0 | R/W | Communication control (Note1)(Note2)(Note3)(Note4)<br>0: Communication stops<br>1: Communication is enabled<br><br>Full duplex mode/transmission mode<br>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.<br><br>Receive mode:<br>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<br>0: SPI mode<br>1: SIO mode |
| 12 | MSTR | 1 | R/W | Master/slave selection<br>0: Slave operation<br>1: Master operation |
| 11:10 | TMMD[1:0] | 11 | R/W | Transfer mode selection<br>00: Reserved<br>01: Transmit only<br>10: Receive only<br>11: Full-duplex mode (Transmit/receive)<br><br>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<br>00: TSPIxCS0 is valid<br>01: TSPIxCS1 is valid<br>10: TSPIxCS2 is valid<br>11: TSPIxCS3 is valid<br><br>When slave operation and selected SIO mode, TSPIxCS0/1/2/3 cannot be used. Therefore, the function setting is not selected. (Note5) |
| 7:0 | FC[7:0] | 0x01 | R/W | Sets the number of transfer frames<br>0: Continuously transfer<br>(No limit of transfer times specification) (Note6)<br>1: Single transfer (one burst transfer)<br>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 write “0” to <TRXE> in continuously transfer. However, in the case of single transfer and burst transfer, <TRXE> is automatically cleared to “0” after the specified number of these transfers are complete. If single transfer and burst transfer are 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.<br>00: Hi-z<br>01: Last data in previous transfer<br>10: Fixed to low<br>11: Fixed to high |
| 21 | TXDEMP | 1 | R/W | Fixed output value function control when TSPIxTXD underruns (Slave operation).<br>0: Fixed to low<br>1: Fixed to high |
| 20:17 | - | 0 | R | Read as "0". |
| 16 | RXDLY | 1 | R/W | TSPIxSCK output frequency fcyc of transfer clock (Master)<br>0: When fcyc＝fsys/2(below condition)<br>(ΦT0=fsys, and ***[TSPIxBR]***<BRCK>=0 <BRS>=1)<br>1: When fcyc ≤ fsys/4<br><br>The setting value of <RXDLY> depending on the product. For the setting value, refer to "Product Information" of the reference manual. |
| 15:12 | TIL[3:0] | 0000 | R/W | Transmit fill level setting<br>Transmit FIFO interrupt occurrence condition (Note) |
| 11:8 | RIL[3:0] | 0001 | R/W | Receive fill level setting<br>Receive FIFO interrupt occurrence condition (Note) |
| 7 | INTTXFE | 0 | R/W | Transmit FIFO interrupt control<br>0: Disabled<br>1: Enabled<br><br>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<br>0: Disabled<br>1: Enabled<br><br>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<br>0: Disabled<br>1: Enabled<br><br>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<br>0: Disabled<br>1: Enabled<br><br>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<br>0: Disabled<br>1: Enabled<br><br>This is enable bit for the vertical parity error, trigger error at master device, or interrupt of overrun error and underrun error at slave device operation. |
| 1 | DMATE | 0 | R/W | Transmit DMA control<br>0: Disabled<br>1: Enabled<br><br>If <DMATE> is set to "0" while the transmit DMA request signal is asserted, the request signal is deasserted. It is reasserted if it satisfies the transmission DMA request signal generation requirement when set again to enable. |
| 0 | DMARE | 0 | R/W | Receive DMA control<br>0: Disabled<br>1: Enabled<br><br>If <DMARE> is set to "0" while the receive DMA request signal is asserted, the request signal is deasserted. It is reasserted if it satisfies the receive DMA request signal generation requirement when set again to enable. |

Note: Set the fill level within available values shown in "Table 3.3 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 | Clears transmit buffer<br>0: Invalid<br>1: Clear<br><br>By writing "1" to <TFEMPCLR>, 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. |
| 0 | RFFLLCLR | 0 | W | Clears receive buffer<br>0: Invalid<br>1: Clear<br><br>By writing "1" to <RFFLLCLR>, 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. |

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

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

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

| Bit | Bit Symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31 | DIR | 1 | R/W | Transfer direction<br>0: LSB first<br>1: MSB first |
| 30 | - | 0 | R | Read as "0". |
| 29:24 | FL[5:0] | 001000 | R/W | Sets a frame length.(Note1)<br>Sets a data length of one frame including a parity bit.<br>001000: 8 bits<br>001001: 9 bits<br>:<br>011111: 31 bits<br>100000: 32 bits<br>Other than the above is prohibited. |
| 23:20 | FINT[3:0] | 0000 | R/W | Interval time between frames in the burst transfer.<br>0000: 0 (No interval)<br>0001: 1 x TSPIxSCK cycle<br>0010: 2 x TSPIxSCK cycles<br>:<br>1110: 14 x TSPIxSCK cycles<br>1111: 15 x TSPIxSCK cycles<br><br>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)<br>0: Negative logic<br>1: Positive logic |
| 18 | CS2POL | 0 | R/W | Polarity of TSPIxCS2(Master operation)<br>0: Negative logic<br>1: Positive logic |
| 17 | CS1POL | 0 | R/W | Polarity of TSPIxCS1(Master operation)<br>0: Negative logic<br>1: Positive logic |
| 16 | CS0POL | 0 | R/W | Polarity of TSPIxCS0(Master operation)<br>Polarity of TSPIxCSIN(Slave operation) (Note2)<br>0: Negative logic<br>1: Positive logic |
| 15 | CKPHA | 1 | R/W | Polarity of serial clock<br>0: Data is sampled on the first edge.(Master operation)<br>1: Data is sampled on the second edge. |
| 14 | CKPOL | 1 | R/W | Polarity of idle period of serial clock (Note 2)<br>0: TSPIxSCK is "Low" level at idle.<br>1: TSPIxSCK is "High" level at idle. |

| | | | | |
|---|---|---|---|---|
| 13:10 | CSINT[3:0] | 0001 | R/W | Idle time (Note 3)<br>TSPIxCS0/1/2/3 invalid→TSPIxCS0/1/2/3 valid time<br><br>0000: Prohibited<br>0001: 1 x TSPIxSCK cycle<br>0010: 2 x TSPIxSCK cycles<br>:<br>1110: 14 x TSPIxSCK cycles<br>1111: 15 x TSPIxSCK cycles<br><br>CS deassertion period until the start of next frame of continuously transfer.<br>Even if selecting SIO mode, transfer waits <CSINT> setting value time equivalent.<br>The setting is valid only in master mode. |
| 9:8 | - | 0 | R | Read as "0" |
| 7:4 | CSSCKDL[3:0] | 0000 | R/W | Serial clock delay<br>TSPIxCS0/1/2/3 valid→TSPIxSCK valid time<br><br>0000: 1 x TSPIxSCK 1000: 9 x TSPIxSCK<br>0001: 2 x TSPIxSCK 1001: 10 x TSPIxSCK<br>0010: 3 x TSPIxSCK 1010: 11 x TSPIxSCK<br>0011: 4 x TSPIxSCK 1011: 12 x TSPIxSCK<br>0100: 5 x TSPIxSCK 1100: 13 x TSPIxSCK<br>0101: 6 x TSPIxSCK 1101: 14 x TSPIxSCK<br>0110: 7 x TSPIxSCK 1110: 15 x TSPIxSCK<br>0111: 8 x TSPIxSCK 1111: 16 x TSPIxSCK<br><br>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. |
| 3:0 | SCKCSDL[3:0] | 0000 | R/W | TSPIxCS0/1/2/3 deassertion delay<br>Last data →TSPIxCS0/1/2/3 invalid time<br><br>0000: 1 x TSPIxSCK 1000: 9 x TSPIxSCK<br>0001: 2 x TSPIxSCK 1001: 10 x TSPIxSCK<br>0010: 3 x TSPIxSCK 1010: 11 x TSPIxSCK<br>0011: 4 x TSPIxSCK 1011: 12 x TSPIxSCK<br>0100: 5 x TSPIxSCK 1100: 13 x TSPIxSCK<br>0101: 6 x TSPIxSCK 1101: 14 x TSPIxSCK<br>0110: 7 x TSPIxSCK 1110: 15 x TSPIxSCK<br>0111: 8 x TSPIxSCK 1111: 16 x TSPIxSCK<br><br>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. |

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)<br>000: 2/fc<br>001: 4/fc<br>010: 8/fc<br>011: 16/fc<br>100: 32/fc<br>101: 64/fc<br>110: 128/fc<br>111: Reserved |
| 3:2 | - | 0 | R | Read as "0". |
| 1 | VPE | 0 | R/W | Vertical parity function(Note)<br>0: Disabled<br>1: Enabled |
| 0 | VPM | 0 | R/W | Vertical parity mode selection(Note)<br>0: Even parity<br>1: Odd parity |

Note: 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 | TSPIDR[31:0] | 0x00000000 | R | Read from the receive FIFO |
| | | | W | Write to the transmit FIFO |

Note1: Do not write data to this register when the transmit FIFO is full.
Note2: Do not read data from this register when the receive FIFO is empty.

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>After reset</th><th>Type</th><th>Function</th></tr>
<tr><td>31</td><td>TSPISUE</td><td>0</td><td>R</td><td>TSPI modify status flag<br>0: Modification is enabled.<br>1: Modification is disabled.<br><br>If <TSPISUE> is "0", the TSPI is not transmitting or receiving, thus the register setting can be modified.<br><TSPISUE> is "0" in the following conditions: (Please refer to Table 4.2.)<br><br>1. Reset is input.<br>2. Software reset occurs.<br>3. In the continuously transfer mode, the time when current transferring frame is finished when [TSPIxCR1] <TRXE>=0 is set.<br>4. In the burst mode, the timing when specified number of transfers are finished.<br>5. The time when current transferring frame is finished when [TSPIxCR1]<TRXE>=0 is set during burst transfer.<br><br>However, even if above conditions are satisfied, <TSPISUE> does not become "0" when the transmit FIFO or receive shift register is full. To set <TSPISUE>=0, read the receive FIFO and transfer a receive value in the receive shift register to the receive FIFO.</td></tr>
<tr><td>30:24</td><td>-</td><td>0</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>23</td><td>TXRUN</td><td>0</td><td>R</td><td>Transmit shift operation flag<br>0: Stop<br>1: Operation<br><br>A status flag indicates the transmit shift operation is ongoing.<br>Combination of <TXRUN> and <TFEMP> bits indicates the following status:<br><br>
<table>
<tr><th><TXRUN></th><th><TFEMP></th><th>Conditions</th></tr>
<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>
</table>
<br><TXRUN> is set when data exists in the transmit shift register even if data does not exist in the transmit FIFO.</td></tr>
</table>

<table>
<tr><td rowspan="2">22</td><td rowspan="2">TXEND</td><td rowspan="2">0</td><td>R</td><td>Transmit completion flag<br>A flag that is set at the time when transmission is complete.<br>0: -<br>1: Transmit is complete.<br><br>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.</td></tr>
<tr><td>W</td><td>This bit is cleared by writing "1".<br>0: Don’t care<br>1: Flag is cleared.<br><br>When the setting by transmission completion and clearing by writing "1" occur simultaneously, the setting by transmission receives a higher priority.</td></tr>
<tr><td rowspan="2">21</td><td rowspan="2">INTTXWF</td><td rowspan="2">0</td><td>R</td><td>Transmit FIFO interrupt flag<br>This bit is set when remaining data in the transmit FIFO reaches a TIL value from a fill level setting value (TIL)+1.<br>0: No interrupt<br>1: Interrupt occurs</td></tr>
<tr><td>W</td><td>This bit is cleared by writing "1".<br>0: Don’t care<br>1: Flag is cleared.</td></tr>
<tr><td>20</td><td>TFEMP</td><td>1</td><td>R</td><td>Transmit FIFO empty flag<br>0: Data exists in the FIFO<br>1: Empty<br><br>When the transmit FIFO is empty, "1" is set.<br>If transmit data is written to the transmit FIFO, this bit is automatically cleared to "0".</td></tr>
<tr><td>19:16</td><td>TLVL[3:0]</td><td>0000</td><td>R</td><td>Transmit FIFO fill level status<br>Indicates the current value of the transmit FIFO fill level (number of data). Stages of the FIFO vary depending on the length of frame. Table 4.3 shows the display range.</td></tr>
<tr><td>15:8</td><td>-</td><td>0</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>7</td><td>RXRUN</td><td>0</td><td>R</td><td>Receive operation flag<br>0: Stop<br>1: Operation<br><br>A status flag indicates the receive shift operation is ongoing.<br>Combination of <RXRUN> and <RFFLL> indicate the following status.
<table>
<tr><th><RXRUN></th><th><RFFLL></th><th>Conditions</th></tr>
<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>
</table></td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| 6 | RXEND | 0 | R | Receive completion flag<br>A flag that is set at the time when reception is complete..<br>0: -<br>1: Receiving is complete.<br><br>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".<br>0: Don't care<br>1: Flag is cleared.<br><br>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<br>0: No interrupt<br>1: Interrupt occurs<br><br>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<br>1: Flag is cleared.<br><br>This bit is cleared by writing "1". |
| 4 | RFFLL | 0 | R | Receive FIFO full flag<br>0: A space exists in the FIFO<br>1: Full<br>Indicates that the receive FIFO is full.<br>This bit is automatically cleared if data is read from the receive FIFO. |
| 3:0 | RLVL[3:0] | 0000 | R | Receive FIFO fill level status<br><br>Indicates that the current value of the receive FIFO fill level (number of data).<br>Stages of the FIFO varies depending on the length of frame. Table 4.3 shows the display range on <RLVL>. |

**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"<br>(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<br>When a trigger is input, this bit is set if communication has not been started.<br>0: No error<br>1: Error exists |
| | | | W | This bit is cleared by writing "1". Do not clear the bit during transmission/reception.<br>0: Don't care<br>1: Flag is cleared |
| 2 | UDRERR | 0 | R | Underrun error flag<br>This bit is set when an underrun error occurs.<br>0: No error<br>1: Error exists |
| | | | W | This bit is cleared by writing "1". Do not clear the bit during transmission/reception.<br>0: Don't care<br>1: Flag is cleared |
| 1 | OVRERR | 0 | R | Overrun error flag<br>This bit is set when an overrun error occurs.<br>0: No error<br>1: Error exists |
| | | | W | This bit is cleared by writing "1". Do not clear the bit during transmission/reception.<br>0: Don't care<br>1: Flag is cleared |
| 0 | PERR | 0 | R | Vertical parity error flag<br>This bit is set when a vertical parity error occurs.<br>0: No error<br>1: Error exists |
| | | | W | This bit is cleared by writing "1". Do not clear the bit during transmission/reception.<br>0: Don't care<br>1: Flag is cleared |

# 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.<br>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.<br>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]***<TSPISUE> bit was returned to "0".<br>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.<br>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.<br>If a burst transfer is attempted again, set "1" to <TRXE> after the confirmation that ***[TSPIxSR]***<TSPISUE> bit was returned to "0".<br>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.<br>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. |
| | Burst transfer (Include single transfer) | A reception starts if receive buffer (receive FIFO or receive shift register) is not full.<br>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 <TRXE> after ***[TSPIxSR]***<TSPISUE> bit was returned to "0". | In reception, if <TRXE> is set to stop, a transfer is stopped after a frame in progress is complete. |

# 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 | 2017-12-26 | First release |
| 2.0 | 2018-03-20 | - Conventions<br>Modified “This flash” to “the flash” in trademark.<br>- 1. Outline<br>Modified “Minimum idle time at the time of single transfer” to “Idle time at the time of continuously transfer.” in Special Control of Table1.1/Table1.3.<br>Modified explanation of DMA request in Table1.1 to Table1.4<br>Modified explanation of “TSPIxCS0/1/2/3 deassertion delay", final data"-->"last data" of Table1.1/Table1.3.<br>- Configuration<br>Added (Note) to TSPIxCS0/TSPIxCSIN pin in Figure 2.1<br>-3.1.3 Start and stop transfer<br>Added “continuously transfer” at stop transfer explanation<br>-3.3.1 Transmission clock<br>Modified “Production” to “Product” in Table3.2 condition,<br>- 3.3.4.1 Full duplex communication mode<br>Modified “single transfer” to “continuously transfer”.<br>- 3.3.4.2 Transmit mode<br>Modified “single transfer” to “continuously transfer”.<br>- 3.3.4.3 Receive mode<br>Modified “single transfer” to “continuously transfer”.<br>- 3.3.5 Transfer mode<br>Added continuously transfer.<br>-3.3.5.1 Single transfer<br>The order of explanation was changed.<br>- 3.3.7.1 Polarity of TSPIxCS0/1/2/3 signal and generation timing<br>Title review of Figure 3.20<br>- 3.3.7.3 TSPIxTXD Output during Idle<br>Modified waveform of Idle period in Figure 3.22<br>Added Note: under Table 3.4<br>- 3.3.9.1 Transmit Completion Interrupt/Receive Completion Interrupt<br>Modify explanation.<br>Modified “INTxTX” to “INTTxTX”, “INTxRX” to “INTTxRX”, “INTxERR” to “INTTxERR” in Figure 3.23.<br>- 3.3.10.1 Transmit DMA request<br>Modified “transmit DMA request” to “burst transmit DMA request”<br>- 3.3.10.2 Receive DMA request<br>Modified “receive DMA request” to “burst recieve DMA request”<br>-4.2.1 ***[TSPIxCR0]***<br>Modified “<VPERR> to <PERR> in Table4.1<br>-4.2.2 ***[TSPIxCR1]***<br>Added (Note4)(Note5) in <TRXE>, Added (Note6) in <CSSEL>, Change (Note4) to (Note7) in <FC[7:0]>.<br>- 4.2.3 ***[TSPIxCR2]***<br>Modified explanation of <INTTXWE><INTRXWE>.<br>- 4.2.6 ***[TSPIxFMTR0]***<br>Added explanation to <FINT[3:0]><br>Changed <CKPOL>, (Note2) to (Note3)<br>Added (Note4) to <CSINT[3:0]><br>Modified title and explanation of <CSINT><br>Changed title order of <CSSCKDL>/<SCKSCDL><br>Deleted “Transfer complete” from title.<br>-4.2.7 ***[TSPIxFMTR1]***<br>Added (Note1) in <EHOLD[2:0]><br>Changed <VPE><VPM>, (Note) to (Note2)<br>-4.2.9 ***[TSPIxSR]***<br>Modified <TSPISUE> explanation of 3.<br>Modified <TXEND<RXEND> explanation.<br>Moved part explanation of <INTRXFF>, “W” to “R”.<br>-4.2.10 ***[TSPIxERR]***<br>Modified <VPERR> to <PERR> |

| | | |
|---|---|---|
| | | 5. Example for use<br>Modified Transfer mode |
| 3.0 | 2018-07-20 | - Trademark<br>Modified explanation of SST trademark<br>- 1, Outline<br>Deleted “Data is sampled with 1st edge” from Data sampling timing in Table1.2 and Table1.4<br>Deleted “Trigger error interrupt” from Interruption in Table 1.2 and Table 1.4.<br>-2. Configuration<br>Modified “Signal name” of INTTxTX/INTTxRX/INTTxERR<br>- 3.3.1 Transfer clock<br>Tile was modified from Transmission clock to Transfer clock<br>Modified parameter of (1/x),<br>Modified the formula in case of ***[TSPIxCR2]***<RXDLY>=0<br>Added the formula of slave operation<br>Modified explanation of Note2 of Table3.1<br>- 3.3.2.2 SIO mode<br>Added Note<br>- 3.3.5.3 Continuously transfer<br>Deleted Note1 and Note2<br>- 3.3.6 Data sampling timing<br>Added Table 3.4,Table 3.5<br>Modified TSPIxCS_N to TSPIxCSn , <CSxPOL> to <CSnPOL> in Figure 3.16<br>Deleted Data sampling timing of SPI mode(slave) 1st edge from Figure 3.17<br>Modified TSPIxCS_N to TSPIxCSIN, <CS0POL> to <CS0POL> in Figure 3.17<br>Modified Data sampling timing of SIO mode(master) 2nd edge from Figure 3.18<br>Deleted Data sampling timing of SIO mode(slave) 1st edge from Figure 3.19<br>Modified Data sampling timing of SIO mode(slave) 2nd edge from Figure 3.19<br>-3.3.9.1 Transmit Completion Interrupt/Receive Completion Interrupt<br>Added explanation of slave device operation<br>-4.2.2 ***[TSPIxCR1]*** (TSPI Control Register 1)<br>Deleted (Note5) from <TRXE>, Changed Note No, previous. Note6 to Note5, previous Note7 to Note6.<br>-4.2.3 ***[TSPIxCR2]*** (TSPI Control Register 2)<br>Added explanation of <RXDLY><br>-4.2.5 ***[TSPIxBR]*** (TSPI Baud Rate Register)<br>Deleted Note<br>-4.2.6 ***[TSPIxFMTR0]*** (TSPI Format Control Register 0)<br>Deleted Note2. Changed Note No, previous Note3 to Note2, previous Note4 to Note3.<br>-4.2.7 ***[TSPIxFMTR1]*** (TSPI Format Control Register 1)<br>Deleted Note1. Changed Note No, previous Note2 to Note<br>-4.2.9 ***[TSPIxSR]*** (TSPI Status Register)<br>Modified headline of Table 4.2 |
| 3.1 | 2019-06-25 | -1. Outline<br>Modified TSPIxSCK delay of deassert in Table1.1 : “-a” to “to”<br>Modified DMA request in Table1.2/1.3 : “DMA demand” to “DMA request”<br>Modified Software reset in Table1.2 : “Reset by software” to “Reset by software is possible”<br>Added Underrun error condition of Output level of TSPIxTXD in Table1.4<br>-2 Configuration<br>Correced “TSPI Error Flag Register1” to “TSPI Error Flag Register” in Figure 2.1<br>-3.1.1 Clock supply<br>Modified 2nd paragraph : “***[TSPIxCR0]***<TSPIE>=0(TSPI mode)” to “***[TSPIxCR0]***<TSPIE>=0”<br>-3.3.1.1 Master Operation<br>Modified “(1) Master Operation” to “3.3.1.1 Master Operation”, “(2) Slave Operation” to “3.3.1.2 Slave Operation”<br>Corrected formula of transfer clock “x= 1,2, 4,8,16 to 256” to “x= 1,2, 4,8,16 to 256,512”<br>-3.3.2.1 SPI mode<br>- Modified “Master device operation” to “Master operation”, “Slave device operation” to “Slave operation”<br>-3.3.2.2 SIO mode<br>- Modified “Master device operation” to “Master operation”, “Slave device operation” to “Slave operation”<br>-3.3.9.1 Transmit Completion Interrupt/Receive Completion Interrupt |

| | | |
|---|---|---|
| | | Added Note in (1) Master operation<br>-3.3.9.3 Error Interruption<br>Corrected Figure 3.24 : “Transmit shift operation flag<TXRUN>” to “Underrun error flag<UDRERR>”<br>-4.2.3 ***[TSPIxCR2]***<br>Modified description of <INTERR> |
| 3.2 | 2021-10-15 | -3.3.8. Communication control by trigger (3) Receiving trigger Added description.<br>-4.2.6. In function description of ***[TSPIxFMTR0]*** 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.**