1.1. Session-related signals on the TCP interface

Table 1-1 shows the session-related signals of the TCP interface.

The mode in which a session is passively established by a connection request from the session destination is called server mode, while the mode in which a connection is established from SiTCP is called client mode. In both modes, OPEN_ACK (MAIN_OPEN_ACK in server mode, SUB_OPEN_ACK in client mode) is set to 1 when a session is established, 0 when a session is disconnected and CLOSE_REQ is set to 1 when a request is made by the connection peer to disconnect the session. The sequence for session disconnection is shown in 2.6.1 to 2.6.4.

To disconnect a session in server mode, set CLOSE_ACK to 1. CLOSE_ACK should be kept at 1 until CLOSE_REQ and MAIN_OPEN_ACK reach 0. When disconnecting on a disconnect request from the session destination, CLOSE_REQ connect to CLOSE_ACK either directly or via the required timing wait circuit.

When starting a session in client mode, set OPEN_REQ to 1 and set it to 0 when disconnecting (if the session cannot be established after setting OPEN_REQ to 1, TCP_OPEN_ERROR will be 1.)

When CLOSE_REQ becomes 1, a session disconnection request is sent by the connection partner, which requires OPEN_REQ to be set to 0.

In client mode, only the TCP main port can be available, not the sub-ports.

Even if receiving a session disconnection request from the connection destination, SiTCP tries to maintain the session until the transmission buffer is empty. At this time, if data is sent, the session may be forcibly disconnected by the other side, but this is not abnormal.

In addition, the server MAC address, IP address and TCP port number must be set when using the client mode (Ref. 2.2).

Table.1-1. Session-related signals on the TCP interface

Signal name	I/O	Clock synchronization	Description
OPEN_REQ	I	CLK	Signal set to 1 when a session start request is made in client mode and 0 when a session disconnection request is made (*1).
MAIN_OPEN_ACK	O		Signal set to 1 when a session is established on the main port.
SUB_OPEN_ACK	O		Signal sent to 1 when a session is established on the sub port (*2).
TCP_OPEN_ERROR	O		Signal that becomes 1 when OPEN_REQ is set to 1 after session disconnection from SiTCP in client mode and before the MSL timer (*3) expires.
CLOSE_REQ	O		Signal set to 1 on receiving a session disconnection request (FIN)
CLOSE_ACK	I		Signal sent to 1 when session disconnection is indicated in server mode.

*1: In server mode, input 0.

*2: In client mode, the subport cannot be used, so it does not changed (fixed at 0).

*3: Refer to the SiTCP internal register manual.

1.1.1. Sequence when the session is disconnected by the connecting partner (server mode)

In server mode, when a disconnection request is received from the connection partner, CLOSE_REQ is set to 1 and then MAIN/SUB_OPEN_ACK becomes 0.

CLOSE_REQ should be connected to CLOSE_ACK either directly or via the required timing wait circuit (Fig. 1-1).

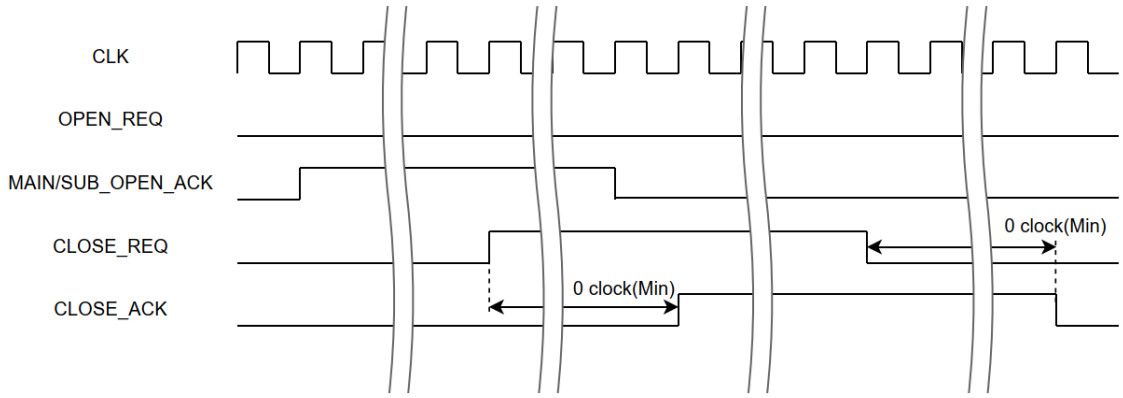

Figure.1-1. Sequence when the session is disconnected by the connecting partner (server mode)

1.1.2. Sequence when the session is disconnected by the SiTCP (server mode)

If the session is disconnected from SiTCP in server mode, set CLOSE_ACK to 1. CLOSE_ACK should remain 1 until MAIN/SUB_OPEN_ACK and CLOSE_REQ reach 0 (Fig.1-2.).

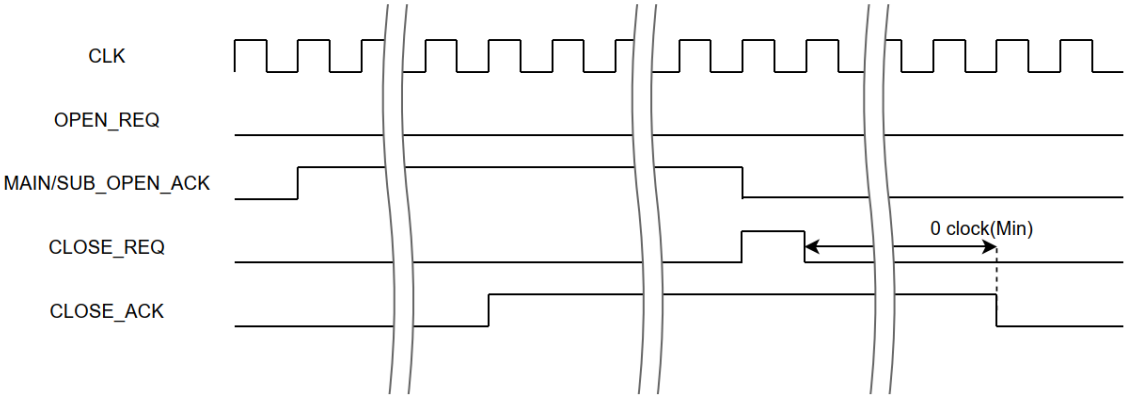

Figure.1-2. Sequence when the session is disconnected by the SiTCP (server mode)

1.1.3. Sequence at session start and disconnection by the connection partner (client mode)

In client mode, CLOSE_REQ becomes 1 when a disconnection request is received from the connection partner. In this case, the request is answered by setting OPEN_REQ to 0 (Fig.1-3).

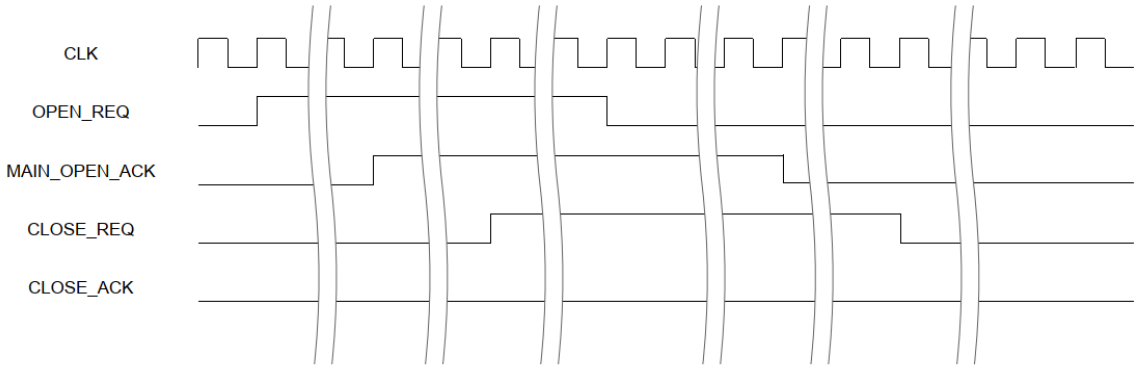

Figure.1-3. Sequence when a session is disconnected by the connection partner (client mode)

1.1.4. Sequence at session start and disconnection by the SiTCP (client mode)

In client mode, set OPEN_REQ to 0 when disconnecting the session from SiTCP (Fig.1-4).

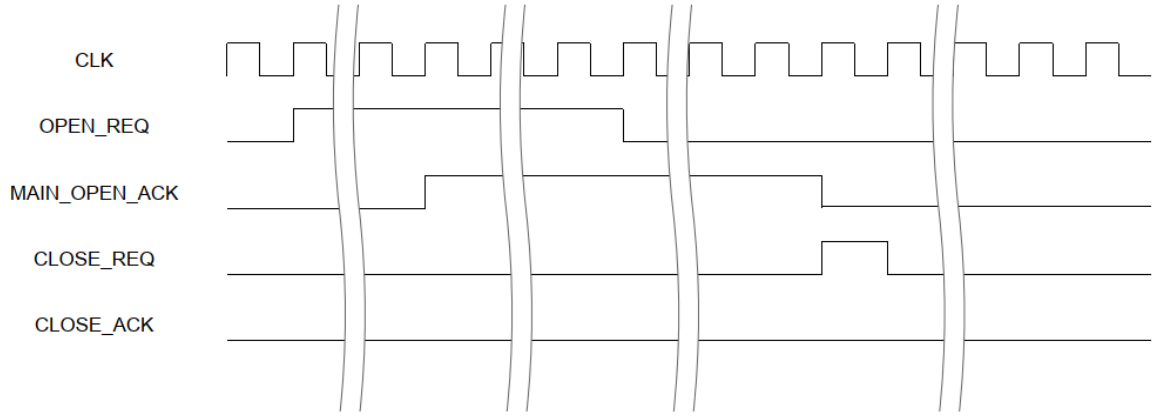

Figure.1-4. Sequence at disconnection by the SITCP (client mode)

1.2. Data transmission and reception related signals of the TCP interface

Table 1-2 shows the data transmission and reception related signals of the TCP interface.

Table.1-2. Data transmission and reception related signals of the TCP interface

Signal name	I/O	Clock synchronization	Description
TX_WR	I	CLK	1 to transmit data write enable
TX_DATA [7:0]	I		Transmission data
TX_FULL	O		Signal that becomes 1 when Almost Full flag in the transmit buffer is established (*1)
TCP_TX_OW_ERROR	O		Signal that becomes 1 when an overwriting error occurs in the transmit buffer (*2)
TX_FILL [15:0]	O		Number of data bytes stored in the transmit buffer
RX_WR	O		1 for receive data valid
RX_DATA [7:0]	O		Received data
RX_FILL [15:0]	I		Number of data bytes stored in the receive buffer

*1: This signal is set to 1 when the number of transmitted data storage exceeds the "Transmission buffer capacity (32 KB) - 32 Byte".

*2: This signal is set to 1 when the number of transmitted data storage exceeds the "Transmission buffer capacity (32 KB) - 16 Byte".

1.2.1. TCP data transmission

Flow control of TCP data transmission is shown in Fig. 1-5.

Data can only be written when a TCP connection is established and MAIN/SUB_OPEN_ACK=1.

Transmission data is written using TX_WR and TX_DATA [7:0] on the same interface as the synchronous FIFO memory.

Transmission data is written to the transmit buffer by synchronizing to CLK, setting TX_DATA [7:0] and TX_WR = 1.

As this is a synchronous circuit, every rising edge of CLK is regarded as one byte of data.

Data can be written when a TCP connection is established and TX_FULL=0.

If TX_FULL=1, the write operation should be stopped within 8 clocks by setting TX_WR=0.

After TX_FULL=0, writing can be resumed.

Keep repeating the above while the TCP connection is established.

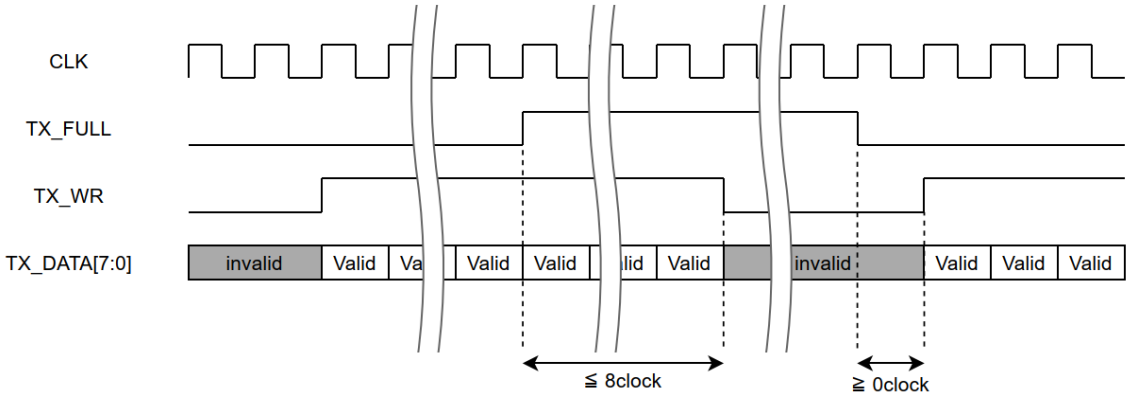

Figure.1-5. Flow control of TCP data transmission

1.2.2. TCP data reception

When using the TCP receive function of SiTCP, the FIFO memory for the receive buffer created by the Xilinx tool [Core Generator]- [FIFO Generator] in the ISE Design Suite or [IP Catalog]- [FIFO Generator] in the Vivado Design Suite should place between the user circuit and the SiTCP and each signal is connected to it. The FIFO memory settings are shown in Table 1-3 and the flow control of TCP data reception is shown in Fig. 1-6.

The SiTCP informs the user that it has received valid data by placing the received data in RX_DATA [7:0] as received order and setting RX_WR=1.

Connect RX_DATA [7:0] to the writing data pins of the synchronous FIFO and RX_WR to the writing enable pins.

In addition, inform the number of data stored in the receive buffer by RX_FILL [15:0] within 8 clocks after data reception. (This is reflected as the TCP window size). Connect RX_FILL [15:0] to the Data count pin of the synchronous FIFO from the lower bits and set all unused bits to 1. For example, when the receive FIFO has a capacity of 16 KB, the Data count pin is 14 bits.

Therefore, connect RX_FILL [13:0] = Data count [13:0] and set 1 to bits RX_FILL

[15:14] (Verilog HDL is RX_FILL [15:14] = 2'b11;).

If the TCP receive function is not used, set 0 to RX_FILL [15:0]. Although it will work with other values, setting 0 is recommended.

Table.1-3. FIFO memory setting

Items	Setting
Writing data bit range	8bit
Data-count option	Effective, bit width set to maximum
FIFO depth	4 KB or more than 64 KB (power of 2)

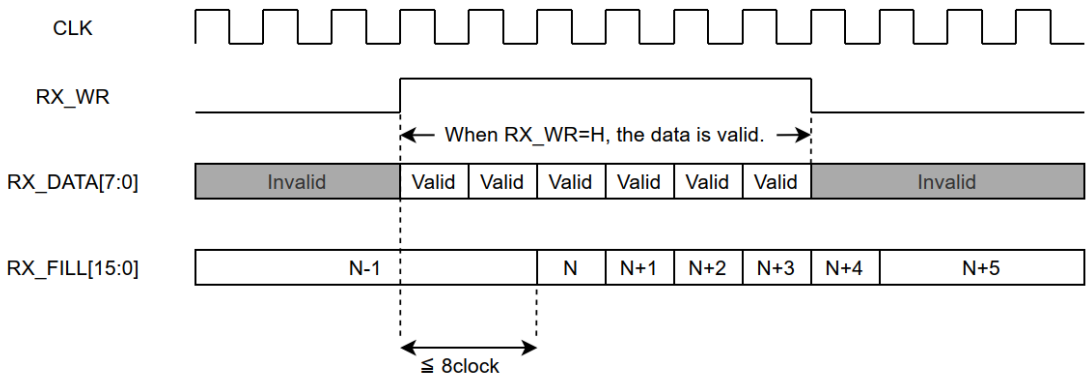

Figure-6 Flow control of TCP data reception