# SiTCP XG Manual

Version 1.2

November 17, 2020


Bee Beans Technologies Co., Ltd.

# Revision history

| Version number | Date | Contents |
|---|---|---|
| 1.0 | Oct. 19, 2020 | First version. |
| 1.1 | Oct. 30, 2020 | Changed the definition of register mapping from 0xFFFF_FF08 to 0xFFFF_FF0F. |
| 1.2 | Nov. 17, 2020 | Added 8 FPGA's as the recommended FPGA. |
| | | |

Table of contents

# 1. Overview

This document describes SiTCPXG, which is a modification of SiTCP for 10GbE. It is assumed that the readers of this document are already familiar with SiTCP.

# 2. Connection for SiTCPXG

SiTCPXG is used by connecting to a 10GbE PCS/PMA with XGMII. The SiTCPXG clock is a single 156.25MHz XGMII clock that is common to both transmission and reception. Both TCP and RBCP operate in synchronization with this clock.

The PCS/PMA to be connected is assumed to be 10G Ethernet PCS/PMA provided by Xilinx (for 10GBASE-R).

Please connect AT93C46D as a memory for storing SiTCPXG setting information and license information.

This memory is not required when using a device with one SiTCPXG library for performance evaluation on an independent network (it will be used in the ForceDefault state).

The SiTCPXG library consists of the files shown in Table 2-1.

Table 2-1 SiTCPXG library file

| File name | Explanation |
|---|---|
| SiTCPXG_xxxx_xxxx_Vx.edf | This is the main body of SiTCPXG. The file name is composed of the FPGA family name, send buffer size, and version number. |
| SiTCPXG_xxxx_xxxx_Vx.v | This file defines the input/output of the SiTCPXG main unit. This is a Verilog file which has the same name as edf file. |
| TIMER_SiTCPXG.v | This is a module that generates the timing of SiTCPXG. It is assumed that the clock frequency is 156.25 MHz. |
| WRAP_SiTCPXG_xxxx_xxx.v | This is a wrapper to make SiTCPXG easier to be used. The file name contains the FPGA family name and the send buffer size. |

※ The number of characters indicated by x is just a guide and is subject to be changed.

# 3. Signal descriptions

This chapter describes the library itself and the wrapper port.

## 3.1. XGMII interface

This is a signal to connect to 10G Ethernet PCS/PMA. XGMII_CLOCK is the XGMII send/receive clock, but the entire SiCTPXG is also synchronized to this clock.

Table 3-1 XGMII interface signals

| Signal name | Input/Output | Explanation |
|---|---|---|
| XGMII_CLOCK | Input | It is the clock for both transmission and reception of XGMII and the clock for the entire SiCPXG.<br>All I/O signals are synchronized with this clock. |
| XGMII_RXC[7:0] | Input | Receive control signal |
| XGMII_RXD[63:0] | Input | Received data |
| XGMII_TXC[7:0] | Output | Transmission control signal |
| XGMII_TXD[63:0] | Output | Transmission data |

## 3.2. EEPROM interface

This port is used to connect to AT93C46D, which is a non-volatile memory for storing SiTCPXG setting information and license information. Please connect the ORG pin of the AT93C46D to GND to X8 mode.

If you want to use it only in ForceDefault state for the evaluation, enter 1 in EEPROM_DO and open other ports.

Table 3-2 EEPROM interface signals

| Signal name | Input/Output | Explanation |
|---|---|---|
| EEPROM_CS | Output | Connect to the CS terminal of AT93C46D |
| EEPROM_SK | Output | Connect to the SK terminal of AT93C46D |
| EEPROM_DI | Output | Connect to the DI terminal of AT93C46D |
| EEPROM_DO | Input | Connect to the DO terminal of AT93C46D |

## 3.3. RBCP interface

The RBCP interface also works in sync with XGMII_CLOCK. It is compatible with SiTCP. A single command can generate up to 255 bus cycles. When one command has started, RBCP_ACT becomes 1. The RBCP_ACT will continue to be 1 until one command finishes. The RBCP_ACT will become 0, when the command finishes or times out. The timeout period is 256ms per command, regardless of the number of bus cycles.

The beginning of the bus cycle is from when RBCP_WE or RBCP_RE is 1 for one clock to where the user circuit sets his RBCP_ACK to 1 for one clock

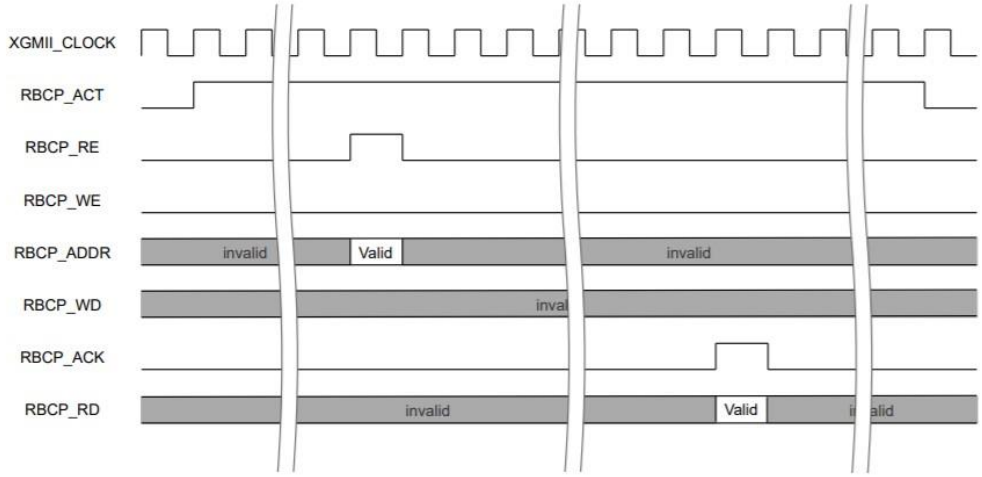


Figure 3-1 RBCP read access

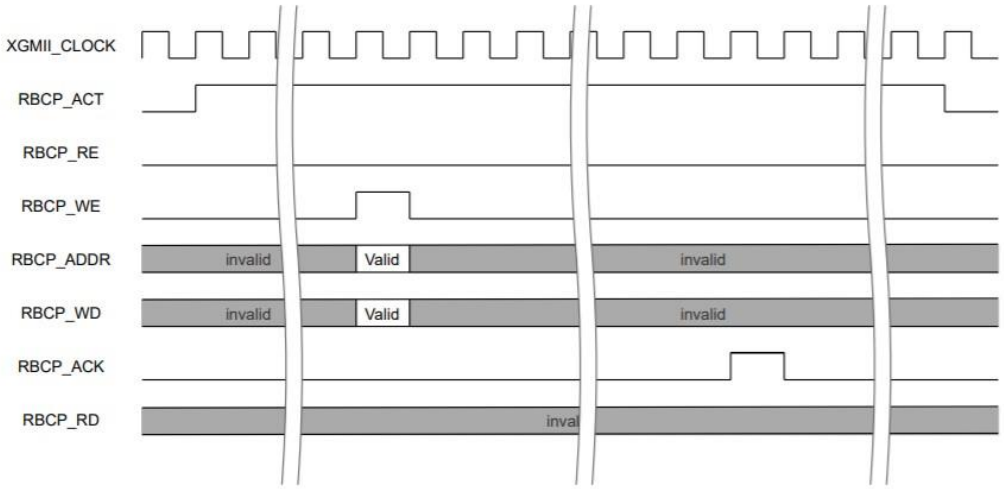


Figure 3-2 RBCP write access

Table 3-3 RBCP interface signals

| Signal name | Input/Output | Explanation |
|---|---|---|
| RBCP_ACT | Output | Bus cycle is running |
| RBCP_ADDR[31:0] | Output | Address |
| RBCP_WE | Output | Start write access |
| RBCP_WD[7:0] | Output | Write data |
| RBCP_RE | Output | Start read access |
| RBCP_ACK | Input | Access termination response |
| RBCP_RD[7:0] | Input | Read data |

## 3.4. TCP interface

The TCP interface is a streaming data-based interface as well as SiTCP. The data input/output interface has changed as the bit width has increased. The TCP interface also works in sync with “XGMII_CLOCK”.

### 3.4.1. TCP session establishment and disconnection

The mode in which the session is not established from SiTCPXG is called the server mode, and the mode in which the session is connected from SiTCPXG is called the client mode. In either mode, “USER_SESSION_ESTABLISHED” becomes 1 when the session is established, and it becomes 0 when the session is disconnected.

Also, “USER_SESSION_CLOSE_REQ” becomes 1 when there is a disconnection request from the other party.

Set “USER_SESSION_CLOSE_ACK” to 1 to disconnect the session in server mode. “USER_SESSION_CLOSE_ACK” should continue to 1 until “USER_SESSION_CLOSE_REQ” and “USER_SESSION_ESTABLISHED” become 0. If you do not want to disconnect the session from SiTCPXG, “USER_SESSION_CLOSE_REQ” connects to “USER_SESSION_CLOSE_ACK” either directly or via the required timing wait circuit.

Set “USER_SESSION_OPEN_REQ” to 1 when starting a session in client mode. If you want to disconnect, set “USER_SESSION_OPEN_REQ” to 0.

If “USER_SESSION_CLOSE_REQ” is 1 in client mode, the end of the session has been sent from the connection destination, so “USER_SESSION_OPEN_REQ” should be set to 0.

When receiving a session end from the other party, SiTCPXG will try to keep the session until the send buffer is empty. At this time, if you send data, the session may be forcibly disconnected from the other party, but this is not a malfunction.

In the current specifications, when using in client mode, you need to set not only the IP address but also the MAC address of the connection destination.

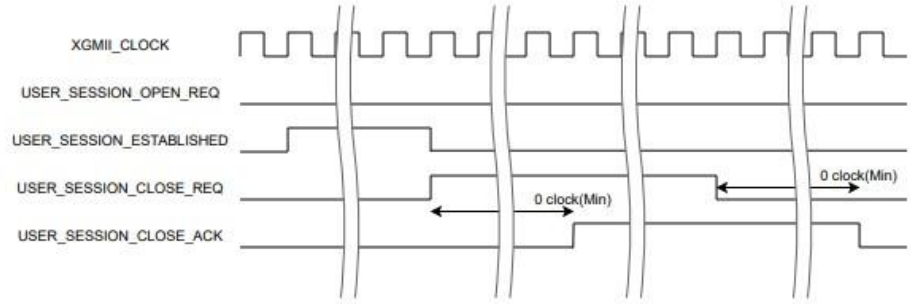


Figure 3-3 Sequence when a session is disconnected by the other party (in server mode)

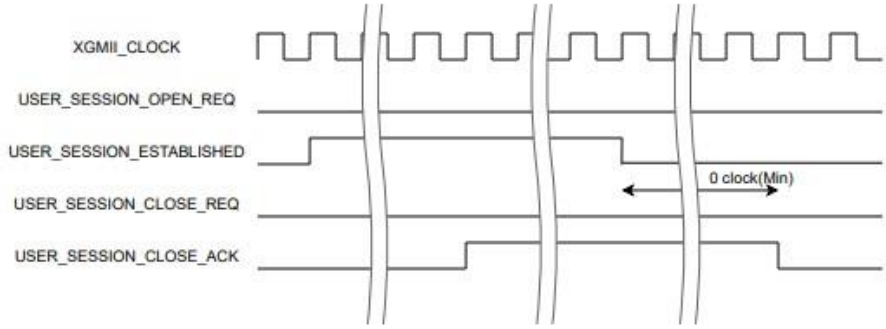


Figure 3-4 Sequence when a session is disconnected by SiTCPXG (in server mode)

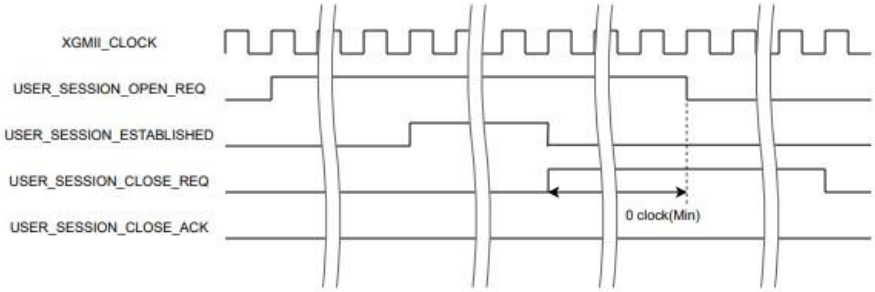


Figure 3-5 Sequence when a session is disconnected by the other party (in client mode)

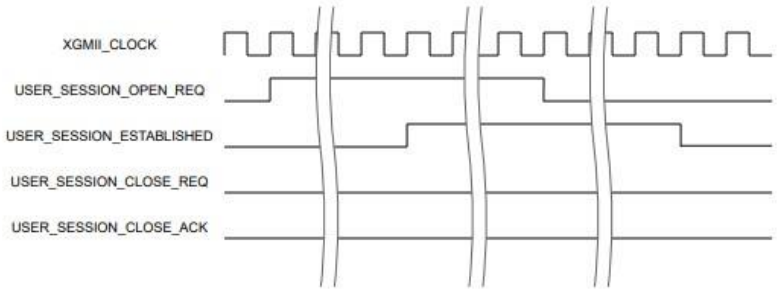


Figure 3-6 Sequence when a session is disconnected by SiTCPXG (in client mode)

Table 3-4 TCP interface session related signals

| Signal name | Input/Output | Explanation |
|---|---|---|
| USER_SESSION_OPEN_REQ | Input | Session start request signal.<br>Fixed to 0 in server mode. |
| USER_SESSION_ESTABLISHED | Output | Display session establishment. |
| USER_SESSION_CLOSE_REQ | Output | Display session end request (FIN reception) |
| USER_SESSION_CLOSE_ACK | Input | Instruct session end in server mode. |

### 3.4.2. TCP transmission data

The send buffer is in the area which the library manages as well as SiTCP. SiTCPXG can write to this buffer on up to 64bit buses. To maximize the performance, the write bus width should be 64bit. The bus width can be changed dynamically from 1Byte to 8Byte in 1Byte units in 8-bit units.

Please specify the number of bytes by setting “USER_TX_B” from 1 to 8. If you specify 0, the transmission data will not be written. Also, the numbers from 9 to F are prohibited from setting. The data is set by “USER_TX_D”, but since the bus width of “USER_TX_D” is 64bit, if the data is less than 64bit, it will be packed from MSB and used. As for the sending order of “USER_TX_D”, the Byte data on the MSB side of the data is sent first. If “USER_TX_AFULL”, which indicates “Almost Full” in the send buffer, becomes 1, please suspend writing within 16 clocks.

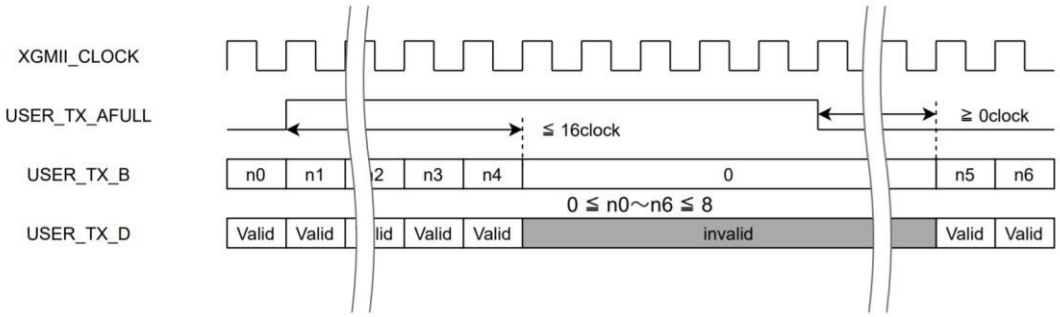


Figure 3-7 TCP transmission data flow control

Table 3-5 TCP data transmission related signals

| Signal name | Input/Output | Description |
|---|---|---|
| USER_TX_AFULL | Output | Transmit buffer “Almost Full” |
| USER_TX_B[3:0] | Input | Number of transmission data (set to 0 if there is no data to transfer) |
| USER_TX_D[63:0] | Input | Transmission data |

### 3.4.3. TCP received data

As with SiTCP, the receive buffer must be provided in the user circuit. SiTCPXG requires writing on 64bit bus. The SiTCPXG requires writing on a 64-bit bus, and since writing must be done independently of the byte lane, a normal FIFO cannot be used. Therefore, the interface uses a write address (USER_RX_WADR) and a read address (USER_RX_RADR). The read address should be provided in the user circuit. The read address should be provided in the user circuit. For each address, the number of bytes written will be added in units of bytes. The read address is used by SiTCPXG to calculate the free space in the buffer, assuming that reading has been completed up to that address. To initialize the read address, copy the value of the write address.

With Xilinx block memories, it is possible to set separate bus widths for write and read. By using this function, the bus width for reading can be fixed. When the bus widths of write and read are different, the LSB side of the data bus becomes the youngest address, but in SiTCPXG, the MSB side becomes the data received first. For this reason, the wrapper swaps the byte lane according to the bus width of the readout to make the connection easier when using a fixed data bus. If you want to use a fixed bus width, set the bus width in the “RxBufferSize” parameter of the wrapper. Please set "Byte" for a fixed 8-bit bus, "Word" for a fixed 16-bit bus, and "LongWord" for a fixed 32-bit bus. Unless a read circuit with a read bus width smaller than the read circuit is provided, the transmission data must be an integral multiple of the bus width.

“USER_RX_CLR_REQ” aligns the write buffer to 8-byte alignment. This function cannot be performed during the period when there is a possibility of writing to the buffer; a clear can be performed during the period when “USER_RX_CLR_ENB” is 1. In many applications, a 1 in “USER_RX_CLR_ENB” will set “US-ER_RX_CLR_REQ” to 1 and copy “USER_RX_WADR” to “USER_RX_RADR” (clearing the receive buffer).

“USER_RX_SIZE” is the maximum window in TCP. Please enter a fixed value less than or equal to [buffer size-16].

“USER_RX_WADR” indicates the next address to start writing in Byte units. Therefore, when writing data to the buffer, bit15 to bit3 of “USER_RX_WADR” specifies the address in 64bit to be written, “USER_RX_WENB” specifies the byte lane to be written, and the data to be written is specified by “USER_RX_WDAT”. In the output from SiTCPXG library, MSB side is the youngest address. The byte position relation between write enable and data is the same for both SiTCPXG library and wrapper, where “USER_RX_WENB[7]” is the write enable of “USER_RX_WDAT[63:56]” and “USER_RX_WENB[0]” is the write enable of “USER_RX_WDAT[7:0]”.

Table 3-6 TCP data reception-related signals

| Signal name | Input / Output | Explanation |
|---|---|---|
| USER_RX_SIZE[15:0] | Input | Maximum receive window size.<br>Set a fixed value less than or equal to buffer size -16.<br>Set 16'hFFF0 when reception id not in use. |
| USER_RX_CLR_ENB | Output | Indicates initialization enable period for “USER_RX_WADR”. |
| USER_RX_CLR_REQ | Input | USER_RX_WADR initialization request/<br>Connect USER_RX_CLR_ENB when the reception not in use. |
| USER_RX_WADR[15:0] | Output | Next write start address (in byte unit). |
| USER_RX_WENB[7:0] | Output | Write enable. |
| USER_RX_WDAT[63:0] | Output | Write data. |
| USER_RX_RADR[15:0] | Input | Read address.<br>Connect “USER_RX_WADR[15:0]” when the read address is not used. |

Table 3-7 Data receive buffer setting parameter (wrapper file only)

| Parameter name | Set value | Explanation |
|---|---|---|
| RxBufferSize | “LongLong” | Default<br>64bit: Same as the bus order in the configuration library for fixed or variable bus widths.<br>64bit data order is network order. |
| | “LongWord” | Setting for fixed 32bit bus.<br>32bit data order is network order. |
| | “Word” | 1Setting for fixed 16-bit bus.<br>16bit data order is network order. |
| | “Byte” | Setting for fixed 8bit bus. |

※Network order: The order in which the byte received first becomes the upper byte

## 3.5. Other signals

This chapter describes the signals common to the SiTCPXG library and wrapper files that have not been covered before.

Table 3-8 Other signals

| Signal name | Input / Output | Explanation |
|---|---|---|
| RSTs | Input | Set to 1 to initialize SiTCPXG.<br>Please use XGMII_CLOCK as a synchronization signal. |
| SiTCP_RESET_OUT | Output | Initialization signal for SiTCPXG's connection.<br>Set to 1 while SiTCPXG is initializing.<br>XGMII_CLOCK will be a synchronous output. |
| FORCE_DEFAULTn | Input | A value of 0 places the device in the ForceDefault state. |
| REG_FPGA_VER[31:0] | Input | This value is displayed in 4 bytes from 0xFFFFFF00 of RBCP register.<br>Set the synthesized date of the user circuit and use it as a version register. |
| REG_FPGA_ID[31:0] | Input | This value is displayed in 4 bytes from 0xFFFFFF04 of RBCP register.<br>Use this register to identify the circuit type of the user circuit. It is recommended that the lower 4 bytes of the MAC address used for development be used so that the identifier is unique. |

The ForceDefault state is a state that allows operation without a license; it can only be used in a closed network of one SiTCPXG and a PC. It can be used in case you lose the initial settings, configured IP address, etc. In the ForceDefault state, a fixed MAC address, IP address, and port number are set.

Table 3-9 ForceDefault status

| Item | Set value |
|---|---|
| MAC address | 02:00:C0:A8:0A:0A |
| IP address ※ | 192.168.10.10 |
| TCP port number ※ | 24 |
| RBCP port number ※ | 4660 |

※These values, except for the MAC address, are register values. It is possible to set any value from the port.

## 3.6. Library-specific signals

This chapter describes the signals in the SiTCPXG library that are not used in the wrapper file.

Table 3-10 Library-specific signals

| Signal name | Input / Output | Explanation |
|---|---|---|
| TIM_1US | Input | Timing signal with one pulse per 1μs. |
| TIM_1MS | Input | Timing signal with one pulse per 1ms. |
| TIM_1S | Input | Timing signal with one pulse per second. |
| MY_MAC_ADDR[47:0] | Output | MAC Address. |
| MY_IP_ADDR[31:0] | Input | IP address setting (Note 1).<br>The wrapper connects “IP_ADDR_DEFAULT”. |
| IP_ADDR_DEFAULT[31:0] | Output | Register value for IP address (Note 2) |
| MY_TCP_PORT[15:0] | Input | TCP port setting (Note 1).<br>The wrapper connects “TCP_PORT_DEFAULT”. |
| TCP_PORT_DEFAULT[15:0] | Output | TCP port register value (Note 2). |
| MY_RBCP_PORT[15:0] | Input | RBCP port setting (Note 1).<br>The wrapper connects “RBCP_PORT_DEFAULT”. |
| RBCP_PORT_DEFAULT[15:0] | Output | RBCP port register value (Note 2). |
| TCP_SERVER_MAC_IN[47:0] | Input | MAC address of the server to be connected (Note 1).<br>The wrapper connects “TCP_SERVER_MAC_DEFAULT”. |
| TCP_SERVER_MAC_DEFAULT[47:0] | Output | MAC address register of the destination server (Note 2). |
| TCP_SERVER_ADDR_IN | Input | IP address of the destination server (Note 1) .<br>The wrapper connects “TCP_SERVER_ADDR_DEFAULT”. |
| TCP_SERVER_ADDR_DEFAULT | Output | Register for IP address of the destination server (Note 2). |
| TCP_SERVER_PORT_IN | Input | TCP port number of the destination server (Note 1).<br>The wrapper connects “TCP_SERVER_PORT_DEFAULT”. |
| TCP_SERVER_PORT_DEFAULT | Output | Register for TCP port number of the destination server (Note 2) |

(Note 1) You can check and set the initial value by connecting to the corresponding dedicated register.
(Note 2) The default value is the value in EEPROM or ForceDefault.

# 4. Register map

Of the RBCP memory space, 0xFFFF0000 to 0xFFFFFFFF is reserved for SiTCPXG internal use. Currently, only the EEPROM space of 0xFFFFFC00 to 0xFFFFFCFF and the SiTCPXG register space of 0xFFFFFF00 to 0xFFFFFFFF are defined.

Because this area controls the basic operation of SiTCPXG, please understand it well before making any changes. In addition, it is not recommended to read as well as write to the area without explanation. Please do not change the undefined bit of the register because it may be extended in the future.

## 4.1. EEPROM space

The EEPROM space can be read out at any time, but the write protection must be released to write to it. Although the value to be written when releasing the write protect is currently arbitrary, it should be 0x00 to ensure compatibility for future expansion.

Table 4-1 EEPROM memory map

| Address | Explanation |
|---|---|
| 0xFFFFFC10～0xFFFFFC4F | Initial values of 0xFFFFFF10 to 0xFFFFFF4F. |
| 0xFFFFFCFF | Write 0x00 to release write protection. |

## 4.2. SiTCPXG register space

The list of SiTCPXG parameter setting registers is shown in Table 4-2. These registers are usually set to standard values by default values or values stored in EEPROM. Please fully understand the meaning of the register's parameter before you are going to make a change in the register. And please note that you should set the information of the server to connect to SiTCPXG if you use it in client mode.

Table 4-2 SiTCPXG register map

| Address | Explanation |
|---|---|
| 0xFFFFFF00～0xFFFFFF03 | User Version register |
| 0xFFFFFF04～0xFFFFFF07 | User Identifier register |
| 0xFFFFFF08～0xFFFFFF0B | SiTCPXG Identifier register |
| 0xFFFFFF0C～0xFFFFFF0F | SiTCPXG Version register |
| 0xFFFFFF10 | Control register |
| 0xFFFFFF12～0xFFFFFF17 | MAC Address register |
| 0xFFFFFF18～0xFFFFFF1B | IP Address register |
| 0xFFFFFF1C～0xFFFFFF1D | TCP Port Number register |
| 0xFFFFFF20～0xFFFFFF21 | TCP maximum segment size register |
| 0xFFFFFF22～0xFFFFFF23 | UDP Port Number register |
| 0xFFFFFF24～0xFFFFFF25 | TCP Keepalive Time(buffer not empty) register |
| 0xFFFFFF26～0xFFFFFF27 | TCP Keepalive Time(buffer empty) register |
| 0xFFFFFF28～0xFFFFFF29 | TCP Timeout (Connecting) register |
| 0xFFFFFF2A～0xFFFFFF2B | TCP Timeout(Disconnect) register |
| 0xFFFFFF2C～0xFFFFFF2D | TCP Maximum Segment Lifetime register |
| 0xFFFFFF2E～0xFFFFFF2F | TCP Retransmission time register |
| 0xFFFFFF32～0xFFFFFF37 | TCP Server MAC Address register |
| 0xFFFFFF38～0xFFFFFF3B | TCP Server IP Address register |
| 0xFFFFFF3C～0xFFFFFF3D | TCP Server Port Number register |
| 0xFFFFFF40～0xFFFFFF41 | Transmission rate register |

#### 4.2.1. User Version register

It cannot be written. The value entered in the REG_FPGA_VER[31:0] port is displayed.

#### 4.2.2. User Identifier register

It cannot be written. The value entered on the REG_FPGA_ID[31:0] port is displayed.

#### 4.2.3. SiTCPXG Identifier register

It cannot be written. This is an identifier for SiTCPXG, which is 0x58544350.

#### 4.2.4. SiTCPXG Version register

It cannot be written. This is the BCD data in 4-bit units. 8 digits indicate the type and version of SiTCPXG. The first two digits indicate the FPGA family, the next two digits indicate the option type, the next two digits indicate the major version of SiTCPXG, and the last two digits indicate the minor version of SiTCPXG.

#### 4.2.5. Control register

This sets or resets the SiTCPXG operation mode.

Table 4-3 Control register bit map

| Bit position | Symbol | Explanation |
|---|---|---|
| bit7 | RESET | Reset SiTCPXG on a write of 1 |
| bit6 | NOT_USE | Not used. Set to 0. |
| bit5 | NOT_USE | Not used. Set to 0. |
| bit4 | WINDOW_SCALING | Window scaling. 1: Enable 0: Disable |
| bit3 | NOT_USE | Not used. Set to 0. |
| bit2 | KEEPALIVE | Keepalive timer 1: enable 0: disable |
| bit1 | FAST_RETRANS | Fast re-transmission. 1: Enable 0: Disable |
| bit0 | NAGLE | Nagle's algorithm. 1: Enable 0: Disable |

#### 4.2.6. MAC Address register

It cannot be written. This is the MAC address of your station.

#### 4.2.7. IP Address register

This is the IP address register of your own station. The value read is the value entered on the “MY_IP_ADDR” port. And the set value is output to the “IP_ADDR_DEFAULT” port.

#### 4.2.8. TCP Port Number register

This is the TCP port number register of your own station. The value to be read is input to the “MY_TCP_PORT” port. The set value is output to the “TCP_PORT_DEFAULT” port.

#### 4.2.9. TCP maximum segment size register

It sets the TCP maximum segment size (MSS). Set a value between 1 and 1460.

#### 4.2.10. UDP Port Number register

This is the UDP port number register for the RBCP of your own station. The value to be read is the value input to the "MY_RBCP_PORT port", and the set value is output to the "RBCP_PORT_DEFAULT" port.

#### 4.2.11. TCP Keepalive Time(buffer not empty) register

This sets the timeout value of the keepalive timer when there is still data in the transmit buffer in 1ms units. Please set the value in the range of 1 to 65,535.

#### 4.2.12. TCP Keepalive Time(buffer empty) register

This sets the timeout value of the keepalive timer when there is no data in the transmit buffer in 1ms units. Please set the value in the range of 1 to 65,535.

#### 4.2.13. TCP Timeout (Connecting) register

This register is to set the timeout value for session establishment in 1ms units. Please set it in the range of 1 to 65,535.

#### 4.2.14. TCP Timeout(Disconnect) register

When this timeout period elapses without receiving any valid packets during session establishment, the session is disconnected. The timeout time can be set in 256ms units. When the value N is set, the timeout period is (N+1)×256ms.Please set the value in the range of 0 to 65,535.

#### 4.2.15. TCP Maximum Segment Lifetime register

This sets the TCP Maximum Segment Lifetime (MSL) in units of 0.5ms. After a session is disconnected, a new connection cannot be established until twice this time has elapsed. It can be said that the time to stay in the Time Wait state in the TCP state transition diagram is set to 1ms. The value should be set in the range of 0 to 65,535.

#### 4.2.16. TCP Retransmission time register

This sets the retransmission time in 1ms units. When this time elapses without the ACK number being updated after data transmission, the data is retransmitted. Please set the value in the range of 1 to 65535.

#### 4.2.17. TCP Server MAC Address register

This is used only in client mode. It set the MAC address of the server to connect to. The value read is the value entered in the "TCP_SERVER_MAC_IN" port, and the set value is output to "TCP_SERVER_MAC_DEFAULT" port.

#### 4.2.18. TCP Server IP Address register

This is used only in client mode. It sets the IP address of the server to connect to. The value read is the value entered in the “TCP_SERVER_ADDR_IN” port, and the set value is output to “TCP_SERVER_ADDR_DEFAULT” port.

#### 4.2.19. TCP Server Port Number register

This is used only in client mode. It sets the TCP port number of the server to connect to. The value read is the value entered in the “TCP_SERVER_PORT_IN” port, and the set value is output to the “TCP_SERVER_PORT_DEFAULT” port.

#### 4.2.20. Transmission rate register

This is for the rate setting of the transmission shaper in 1 Mbps units. The shaper is a line rate using the leaky bucket algorithm.

SiTCPXG sends packets as fast as the received window size allows. If the PC cannot receive this burst transfer without loss, the packet will be retransmitted, and the data transmission performance will be significantly reduced. You can obtain the maximum performance by limiting the transmission rate with this register according to the performance of the PC. Please set in the range of 1 to 10,000.

# 5. SiTCPXG implementation

The XGMII in SiTCPXG is designed to connect to a 10G Ethernet PCS/PMA provided by Xilinx (10GBASE-R).

This chapter provides an example of how to create a 10G Ethernet PCS/PMA when using Virtex-7 or Kintex-7 devices. There are some differences depending on the version of VIVADO, so please refer to the 10G Ethernet PCS/PMA Product Guide for details. Please change the DRP clock, MDIO, and Shared Logic as necessary.

## 5.1. IP core generation for 10G Ethernet PCS/PMA

Click [IP Catalog] in VIVADO (left side of Figure 5-1). In the [IP Catalog] tab, open [Communication & Networking], and then open [Ethernet], and you will see [10G Ethernet PCS/PMA (10GBASE-R/KR)]. Under Ethernet, click on [10G Ethernet PCS/PMA (10GBASE-R/KR)] (right side of Figure 5-1).

In the [Configuration - BASE-R] tab, set the [XGMII Datapath Width] to 64bit, check the [MDIO Management] checkbox, and set the [DRP Clocking-Frequency (MHz)] to 156.25MHz (Figure 5-2). In the [Shared Logic] tab, select [Include Shared Logic in Core] (Figure 5-3). In this state, click [OK] to create the IP.

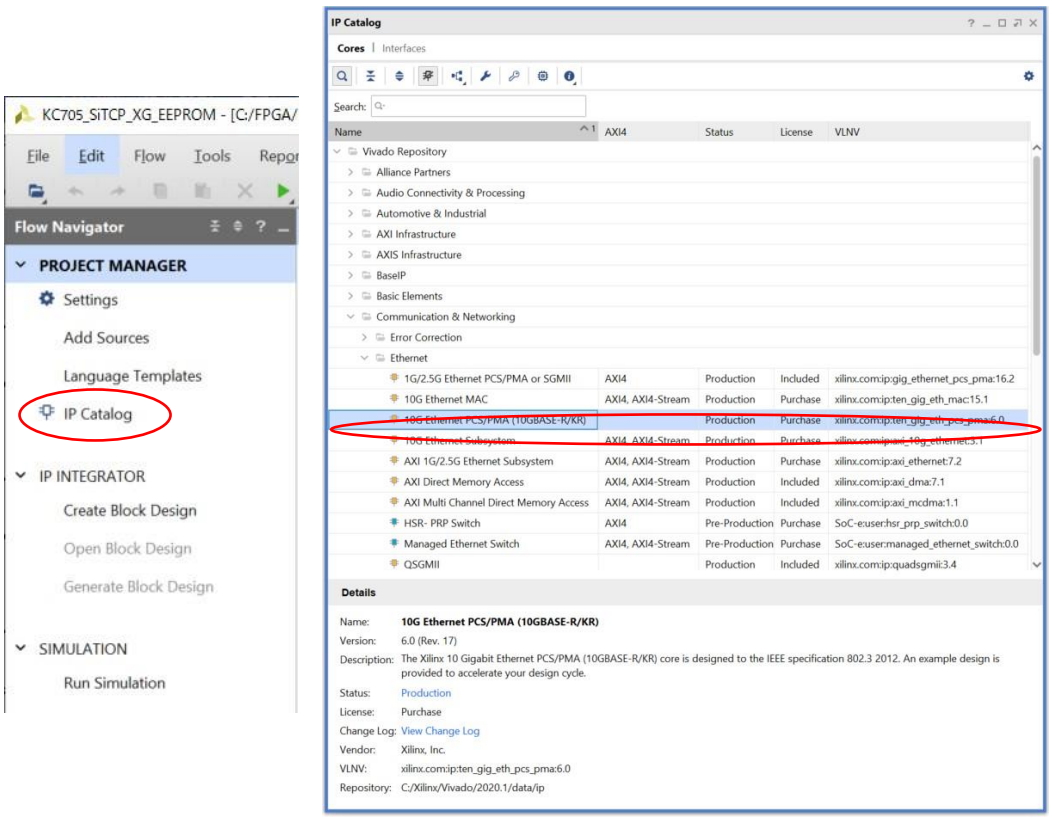


Figure 5-1 10G Ethernet PCS/PMA IP core selection

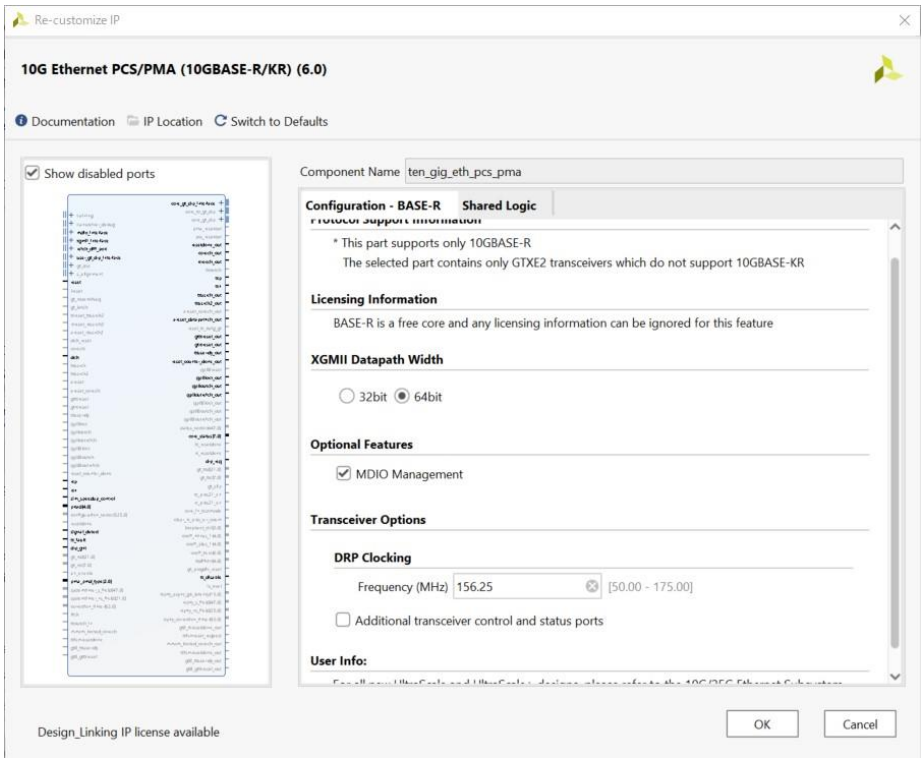


Figure 5-2 10G Ethernet PCS/PMA Configuration – BASE-R

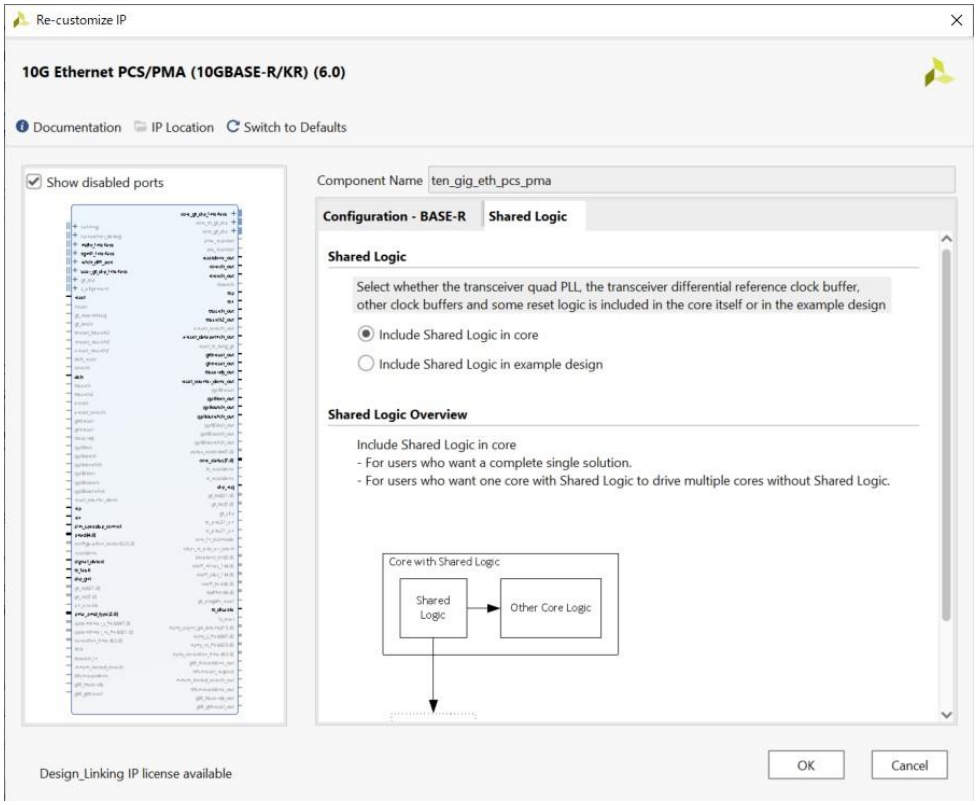


Figure 5-3 10G Ethernet PCS/PMA Shared Logic

## 5.2. IP core connection for 10G Ethernet PCS/PMA (10GBASE-R/KR)

This section describes the generated 10G Ethernet PCS/PMA core ports. The following is an explanation when "10G Ethernet PCS/PMA v6.0" is used with 7 Series. The port may differ depending on the version of VIGADO and the settings at the time of generation, so please use it as a reference only.

Table 5-1 10G Ethernet PCS/PMA v6.0 ports

| Port name | Description |
|---|---|
| refclk_p<br>refclk_n | It is an input of 156.25MHz. Please connect to the same bank as txp, txn, rxp, and rxn. |
| reset | It is a reset. Normally, please enter the same signal as the SiTCPXG RSTs port. |
| coreclk_out | This is clock output including 156.25MHz XGMII. Please connect with XGMII_CLOCK of SiTCPXG. |
| txp<br>txn | This is the transmission output. Please output to TD+ and TD- of SFP+. Please connect to your GTX, GTH TXP, TXN. |
| rxp<br>rxn | This is the input from RD+ and RD- of SFP+. Please connect to RXP and RXN of GTX and GTH to be used. |
| xgmii_txd[63:0]<br>xgmii_txc[7:0] | This is XGMII transmit path. Please connect to XGMII_TXD and XGMII_TXC of SiTCPXG. |
| xgmii_rxd[63:0]<br>xgmii_rxc[7:0] | This is XGMII receive path. Please connect to XGMII_RXD and XGMII_RXC of SiTCPXG. |
| mdc<br>mdio_in<br>mdio_out<br>mdio_tri | This is MDIO interface. If no changes are made to the configuration, the remaining mdio_out and mdio_tri can be used open with mdc=1 and mdio_in=1. |
| signal_detect | Please set SFP+ LOS inversion to input or fixed to 1. |
| tx_fault | Please connect to Tx Fault of SFP+ or fix to 0. |
| tx_disable | Please connect to Tx Disable of SFP+ or leave it open. |
| dclk | This is Dynamic Reconfiguration Port (DRP) Clock<br>(Please connect the clock with the frequency set during IP generation.) |
| drp_xxxx | This is the input/output signal for Dynamic Reconfiguration Port (DRP). This connects the signals with the same name to gnt with req (connects the subscripts _i and _o). |

Table 5-1 10G Ethernet PCS/PMA v6.0 ports (continued)

| Port name | Description |
|---|---|
| sim_speedup_control | For simulation. Please set to 0. |
| pma_pmd_type[2:0] | Sets the type of SFP.<br>111 : 10GBASE-SR<br>110 : 10GBASE-LR<br>101 : 10GBASE-ER |
| Other outputs | Use as needed. Please keep it open when it is not in use. |

# 6. How to change EEPROM

The initial values of the SiTCPXG registers from 0xFFFFFF10 to 0xFFFFFF4F can be saved in the EEPROM's range from 0xFFFFFC10 to 0xFFFFFC4F. In the ForceDefault state, the initial values of the register are fixed and cannot be changed.

Please note that, in the range of 0xFFFFFC00 to 0xFFFFFCFF, you should not access any area other than the area defined in the register map of EEPROM. Also note that you should not rewrite the MAC addresses of 0xFFFFFC12 to 0xFFFFFC17 because rewriting them will SiTCPXG will not start.

Since the initial state of the EEPROM is write protection, it is necessary to release the protection before writing. You can remove the protection by writing 0x00 to 0xFFFFFCFF.

Since writing to the EEPROM takes time, a timeout error will occur if you make a large number of writes at one time. Considering the deterioration of EEPROM, it is recommended to write in one command to 32 bytes or less.

To prevent accidental data rewriting after writing is complete, turn off the power once (it will return to the write protection state when the power is turned off).

# 7. Troubleshooting

This chapter describes matters that are likely to cause problems. It also includes the content described in the text.

## 7.1. Transmission transfer rate is extremely low

If the transmission speed is low, there are several possible causes.

### 7.1.1. Misconfiguration of SiTCPXG

The transmission rate is limited by the value of the Transmission rate register. If the value of Transmission rate register is too small, the performance will be decreased. Also, if the TCP maximum segment size register value (MSS) is small, the performance will be decreased, too.

Since the value (Speed) set in the Transmission rate register is the Line Rate, the maximum speed that can be transferred is Speed*MSS/(MSS+78).

### 7.1.2. Data transfer path failure

If a packet is destroyed in the data transfer path, it will be retransmitted, and the data transfer rate will be slower. The main causes are deterioration and failure of SFP, deterioration of optical cable, etc., as well as different types of SFP and optical fiber.

### 7.1.3. Data transfer path delay

If the delay of the data transfer path is large, the Round-Trip Time (RTT) will increase, and the performance will decrease.

The transmission speed is degraded by the ratio of the window size and the PC ACK response time plus the RTT to the transmission time of the smaller data in the transmission buffer.

### 7.1.4. When the reception performance of PC is a bottleneck

SiTCPXG sends packets at full speed as the received window size allows. If the PC cannot receive this burst transfer without loss, the packet will be retransmitted, and the data transmission performance will be significantly reduced. You can get the maximum performance by limiting the transmission rate according to the performance of your PC by setting the Transmission rate register for optimal transmission rate

## 7.2. Polar reversal connected to SFP

When TD+, TD- of SFP+ and TXP, TXN of FPGA are connected in reverse, or when RD+, RD- of SFP+ and RXP, RXN of FPGA are connected in reverse, you can change the polarity of gt0_txpolarity and gt0_rxpolarity of ten_gig_eth_pcs_pma_block.v according to “10G Ethernet PCS/PMA Ver6.0”.

In Rev1.0 of the KC705 evaluation board, as both transmission and reception are switched, please set both variables to 1.

# 8. Recommended FPGA's

The following table specifies the recommended FPGA sizes for using the released SiTCPXG cores.

Table 8-1 Recommended FPGA's

| Library name | Family | Recommended FPGA |
|---|---|---|
| SiTCPXG_XC7K_128K_V1 | Kintex-7 | XC7K70T and above |
| SiTCPXG_XC7V_128K_V1 | Virtex-7 | XC7V585T and above |