Design Gateway Co.,Ltd
54 BB Building 14th Fl., Room No.1402 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110
Phone: 66(0)2-261-2277
Fax: 66(0)2-261-2290
E-mail: ip-sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	User Guide, Design Guide
Design File Formats	Encrypted HDL
Constraints Files	User constraint file
Verification	Test Bench, Simulation Library
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on KC705/VC707/ZC706/ KCU105/ZCU102
Simulation Tool Used	
ModelSim SE	
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- TCP/IP stack implementation
- Support IPv4 protocol
- Support one session per one TOE10G-IP (Multisession can be implemented by using multiple TOE10G-IP)
- Support both Server and Client mode (Passive/Active open and close)
- Support Jumbo frame
- Transmitted data bus size is 64-bit, so transmitted packet size and total transmitted size must be aligned to 64-bit
- Received data bus size is 64-bit, so total received size must be aligned to 64-bit
- Transmitted/Received buffer size, adjustable to optimize resource and performance
- Simple data interface by standard FIFO interface
- Simple control interface by standard register interface
- 64-bit AXI4 stream to interface for 10-Gbps Ethernet MAC
- One clock domain interface by fixed 156.25 MHz clock frequency
- Half-duplex, Full-Duplex, and Multisession Reference design available on KC705/VC707/ZC706/KCU105/ZCU102 evaluation board
- Not support data fragmentation feature

Table 1: Example Implementation Statistics for 7-Series device

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices[1]	IOB	BRAMTile[2]	Design Tools
Kintex-7	XC7K325TFFG900-2	156.25	3206	3850	1325	-	36	Vivado2017.4
Zynq-7000	XC7Z045FFG900-2	156.25	3206	3848	1369	-	36	Vivado2017.4
Virtex-7	XC7VX485TFFG1761-2	156.25	3206	3847	1359	-	36	Vivado2017.4

Table 2: Example Implementation Statistics for Ultrascale device

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB[1]	IOB	BRAMTile[2]	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	156.25	3211	3786	727	-	34.5	Vivado2017.4
Zynq-Ultrascale+	XCZU9EG-FFVB1156-2-I	156.25	3211	3785	714	-	34.5	Vivado2017.4

Notes:

1) Actual logic resource dependent on percentage of unrelated logic

2) Block memory resources are based on 64kB Tx data buffer size, 16kB Tx packet buffer size, and 64kB Rx data buffer size. Minimum

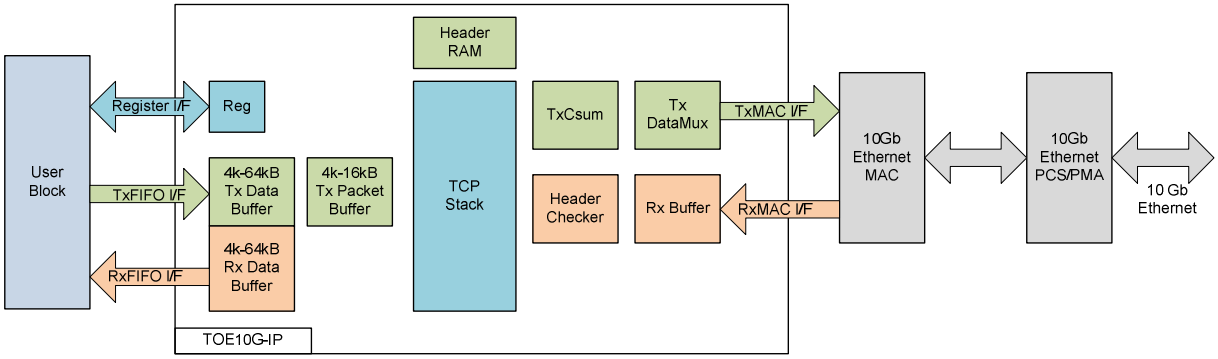

Figure 1: TOE10G-IP Block Diagram

Applications

TOE10G-IP is designed for network application which data reliability is required by using TCP/IP protocol. Also, ultra high speed performance is achieved by using 10 Gb Ethernet. Using TOE10G-IP, the system can easily transfer data with other devices through 10 Gb Ethernet without CPU and external memory.

General Description

TOE10G-IP core operating with Xilinx 10 Gb EMAC IP and 10 Gb Ethernet PCS/PMA implements as TCP/IP stack, Transport layer, Internet layer, Link layer, and Physical layer for network data transmission. User can send and receive 10 Gb Ethernet data with some network devices through TCP/IP protocol by using this system.

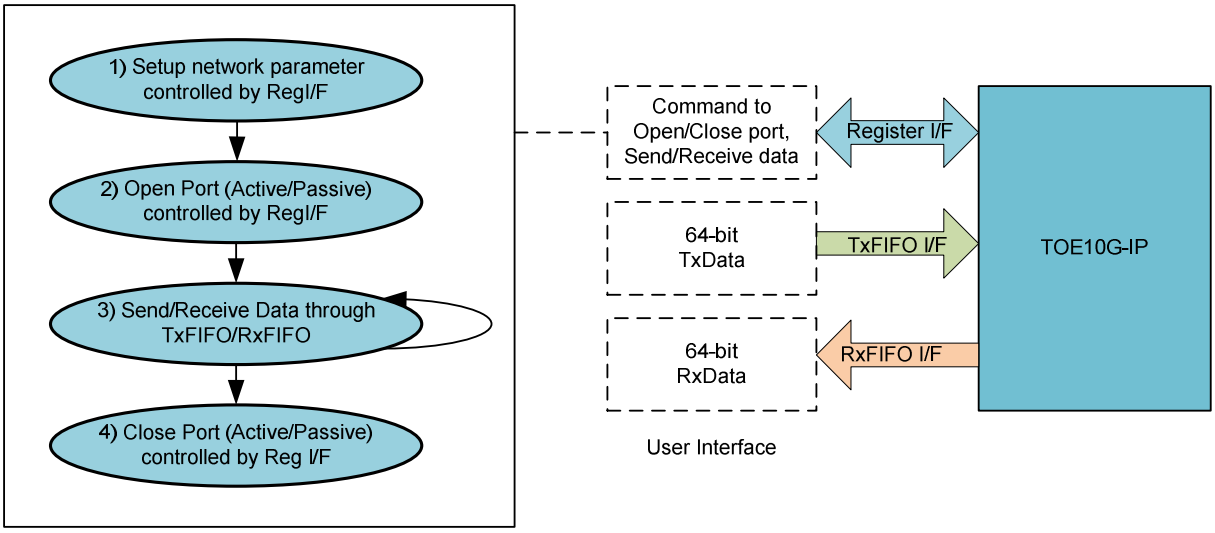

Figure 2: TOE10G-IP User Interface and operation sequence

There are three types of user interface, i.e. control signals by register access, transmitted and received data signals by FIFO access. During system initialization, user needs to set up system parameters such as MAC address, packet size, port number, IP address through register interface. After that, port is opened by user logic (Active mode) or by external device (Passive mode) before start data transferring. The user interface to send or receive data is typical FIFO interface, so user can design simple logic to send or receive data following FIFO timing diagram. After complete data transferring, the port is closed by user logic (Active mode) or by external device (Passive mode), like opening the port.

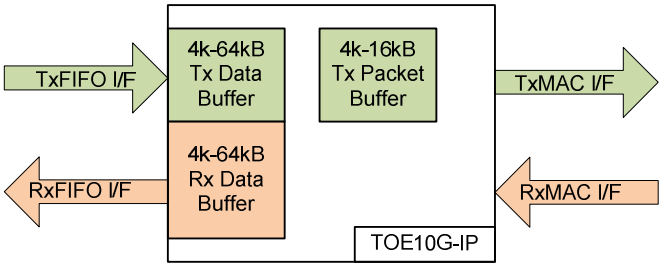

Figure 3: Adjustable Tx/Rx Buffer Size

The size of three buffers in TOE10G-IP (Tx Data buffer, Tx Packet Buffer, and Rx Data Buffer) can be specified by setting TOE10G-IP parameters. The different size is provided to minimize resource utilization while performance remains high for user application. Using bigger buffer size takes much resource, but the better performance is achieved.

Tx Data buffer size and Tx Packet buffer size are effect to transmitted performance, while Rx Data buffer is effect to received performance. The minimum size of Tx Data buffer and Tx Packet buffer depends on Tx packet size, set by user. Tx Packet Buffer size must be more than Tx packet size, while Tx Data Buffer size should be at least two times of Tx packet size.

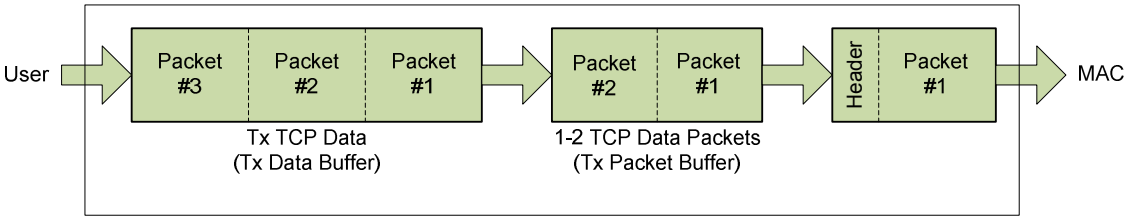

Figure 4: Transmitted Data Flow

To transmit data, data from Tx Data buffer is split into smaller size (equal to packet size which is set by user) and then fed to Tx Packet buffer. Data output from Tx Packet buffer is combined with header data in Header RAM before sending out to EMAC. TCP and IP checksum are auto calculated within TOE10G-IP. Acknowledge packet is monitored during sending operation. Acknowledge packets includes the information to show that the latest data size which is received successfully and remaining size of the received buffer for the receiver. So, TOE10G-IP decides to send next data packet when the receiver is ready to receive new data. If TOE10G-IP received duplicate ACK packet (data lost is found in the receiver), TOE10G-IP will retransmit the lost packet. Busy flag (monitored from register access) is cleared after total data are transferred (total data size is set from user). When complete sending command, busy is de-asserted to ‘0’.

User can set packet size and total transfer size for new transmission without closing the port. Busy flag of TOE10G-IP must be equal to ‘0’ before sending new command.

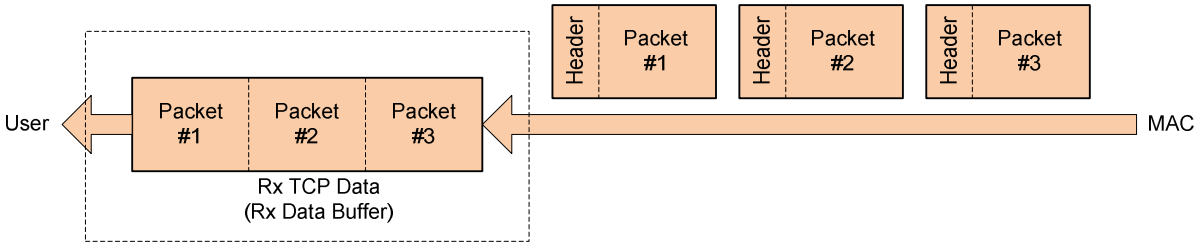

Figure 5: Received Data Flow

For receiving data, Rx packet is stored to temp buffer firstly. Header and checksum of Rx packet are verified. If header or checksum is error, the packet will be rejected and not store to Rx Data buffer. When correct data packet is received, data is stored to Rx Data buffer. Also, Acknowledge packet is created by TOE10G-IP to return the status. If data is lost, IP will generate duplicate ACK packet to request data retransmission. After creating ACK packet, TOE10G-IP goes back to Idle state (Busy flag='0').

Functional Description

TOE10G-IP core can be divided into three parts, i.e. control block, transmitted block, and received block.

Control Block

- **Reg**

  User can set parameters for TCP/IP operation by using register interface. Register address size is 4-bit (Maximum register = 16). The description of each register is defined as shown in Table 3. After system reset is released t0 '0', all internal parameters are loaded from value that is set from register.

- **TCP Stack**

  When user sends command to TOE10G-IP (active mode), TCP Stack decodes user command and starts transmitted block to create and send packet. Also, the received block decodes and monitors the Acknowledge packet. TCP Stack controls the sequence to send and receive the packet.

  For passive mode (new packet is received during idle status), TCP Stack decodes the received packet from received block. The next operation sequence depends on the command in the packet. For example, it creates ARP reply when receiving ARP request.

Table 3: Register map Definition

RegAddr [3:0]	Reg Name	Dir	Bit	Description
0000b	RST	Wr /Rd	[0]	Reset IP. '0': No reset, '1': Reset. Default value is '1'. **After all parameters are assigned, user sets '0' to this register to load parameter and start system intialization. Reset needs to be set/clear again to reload parameter if user changes the value of SML, SMH, DIP, SIP, DPN, or SPN register.**
0001b	CMD	Wr	[1:0]	User command in active mode. "00": Send data, "10": Open connection (active), "11": Close connection (active), "01": Undefined. **Before setting this register to send command (active mode), user needs to check system busy flag that is equal to '0' by reading bit[0] of this register.** The command operation starts after user sets this register.
		Rd	[0]	System busy flag. '0': Idle, '1': system is busy.
			[3:1]	Current operation. "000": Send data, "001": Idle, "010": Active open connection, "011": Active close connection, "100": Receive data, "101": Initialization, "110": Passive open connection, "111": Passive close connection.
0010b	SML	Wr /Rd	[31:0]	Define 32-bit lower MAC address (bit [31:0]) for this IP. **User needs to set this register before setting RST register='0'.**
0011b	SMH	Wr /Rd	[15:0]	Define 16-bit upper MAC address (bit [47:32]) for this IP. **User needs to set this register before setting RST register='0'.**
0100b	DIP	Wr /Rd	[31:0]	Define 32-bit target IP address. **User needs to set this register before setting RST register='0'.**
0101b	SIP	Wr /Rd	[31:0]	Define 32-bit IP address for this IP. **User needs to set this register before setting RST register='0'.**
0110b	DPN	Wr /Rd	[15:0]	Define 16-bit target port number. Set when the connection is opend by the IP (active open). **User needs to set this register before setting RST register='0'.** Target port number is auto defined in case of passive open.
0111b	SPN	Wr /Rd	[15:0]	Define 16-bit port number for this IP. **User needs to set this register before setting RST register='0'**
1000b	TDL	Wr	[31:0]	Total Tx data length in byte unit, but the size must be aligned to 8-byte. Valid from 8-0xFFFFFFF8 (Bit[2:0] is ignored by the IP). **User needs to set this register before setting CMD register = "00". This value are loaded when CMD register is set. User can set the new value for the next transfer after TOE10G-IP starts sending operation.** If the next transmission uses the same length, this register will not need to set again. TOE10G-IP uses the latest value for the next transmission.
		Rd	[31:0]	Remaining transfer length in byte unit which still not transmit.

RegAddr [3:0]	Reg Name	Dir	Bit	Description
1001b	TMO	Wr	[31:0]	Define timeout value for waiting Rx packet during running the command. The counter runs by using 156.25 MHz, so timer unit is equal to 6.4 ns. This value is recommended to be more than 0x6000.
		Rd		[0]-Timeout from not receiving ARP reply packet After timeout, IP resends ARP request until ARP reply is received. [1]-Timeout from not receiving SYN and ACK flag during active open operation After timeout, IP resends SYN packet for 16 times and then sends FIN packet to close connection. [2]-Timeout from not receiving ACK flag during passive open operation After timeout, IP resends SYN/ACK packet for 16 times and then sends FIN packet to close connection. [3]-Timeout from not receiving FIN and ACK flag during active close operation After 1st timeout, IP sends RST packet to close connection. [4]-Timeout from not receiving ACK flag during passive close operation After timeout, IP resends FIN/ACK packet for 16 times and then sends RST packet to close connection. [5]-Timeout from not receiving ACK flag during data sending operation After timeout, IP resends previous data packet. [6]-Timeout from Rx packet lost, Rx data FIFO full, or wrong sequence number IP generates duplicate ACK to request data retransmission. [22]-Receive FIN packet but data sending still not complete. [23]-Rx packet ignored because of Rx data buffer full (fatal error) [21],[27]-Rx packet lost detected [30]-RST flag is detected in Rx packet [31],[29:28],[26:24]-Internal test status
1010b	PKL	Wr /Rd	[15:0]	Data size of Tx packet in byte unit. The size must be aligned to 8-byte. Valid from 8-16000. Default value is 1456 byte (Maximum size with 8-byte alignment for non-jumbo frame). Bit[2:0] of this register is ignored by the IP. **This value must not be changed during data transmission not complete (Busy='1'). If the next transmission uses same packet size, user will not need to set this register. The IP loads the previous value from latched register.**
1011b	PSH	Wr /Rd	[1:0]	Sending mode setting when TOE10G-IP transmits the last packet. [0]-Disable to retransmit packet. Set '0' to generate duplicate data packet for the last transmitted packet (Default = '0'). [1]-Enable to set PSH flag in transmitted packet. Set '1' to insert PSH flag in TCP header of all transmitted packet (Default = '0').

RegAddr [3:0]	Reg Name	Dir	Bit	Description
1100b	WIN	Wr /Rd	[5:0]	Threshold value in 1Kbyte unit for sending windows update packet. Default value is 0 (Not enable window update feature). The IP transmits windows update packet when the different value between the received buffer size and the windows size in the latest transmitted packet is more than threshold value. For example, if WIN="000001b" (1 Kbyte) and window size of the latest transmitted packet is equal to 2 Kbyte. The IP sends Windows update packet after user read data out from the IP more than 1 Kbyte. The window size in windows update packet is equal to 3 Kbyte (2 Kbyte which is previous free space + 1 Kbyte which is additional free space after user reading) .
1101b	ETL	Wr	[31:0]	Extended total Tx length in byte unit. The size must be aligned to 8-byte. Bit[2:0] is ignored by the IP. User sets this register during running CMD="00" to increase total Tx length transfer, so total transmitted size can be increased without re-sending the new command to IP. For example, assumed that the 1st value of TDL=4 GB. When remaining size is 1 GB, user can set ETL=2 GB to complete 6 GB transmitted data (4 GB + 2 GB). The caution point is that 1) ETL register must be programmed when read value of TDL is not less than 128 Kbyte. 2) The set value of ETL must be less than (0xFFFFFFF8 – read value of TDL) to avoild data counter overflow (data counter size is 32-bit).
1110b	SRV	Wr/ Rd	[0]	'0': Client mode. The IP sends ARP request to get Target MAC address from IP address after IP reset is deasserted. After IP receives ARP reply, IP busy is deasserted to '0'. '1': Server mode. The IP waits for ARP request from the Target to get Target MAC address after IP reset is desserted. After IP receives ARP request and returns ARP reply, IP busy is deasserted to '0'. Default value is '0' (Client mode)

Table 4: TxBuf/TxPac/RxBufBitWidth Parameter description

Value of BitWidth	Buffer Size	TxBufBitWidth	TxPacBitWidth	RxBufBitWidth
9	4kByte	Valid	Valid	Valid
10	8kByte	Valid	Valid	Valid
11	16kByte	Valid	Valid	Valid
12	32kByte	Valid	No	Valid
13	64kByte	Valid	No	Valid

Transmitted Block

- **Tx Data Buffer**

This buffer size is set by "TxBufBitWidth" parameter of the IP. The valid value is 9-13 which is equal to the address size of 64-bit buffer, as shown in Table 4.

The buffer size should be at least two times of Tx Packet Size in PKL register. Transmitted data from user is stored to this buffer. Data in FIFO is flushed after the target returns acknowledge packet to confirm that data is received completely. Data from this buffer is forwarded to Tx Packet Buffer which is the buffer to store the next transmitted packet.

This buffer size is effect to the total performance. If the size is large, IP could send data out continuously without waiting acknowledge returned from the target. So, data latency from all processes and the carrier are not much effect to the performance.

If user sends data more than the total transmit size, remaining data will be available in the buffer for the next transfer. The data in the buffer is flushed when the connection is closed or reset is detected. If the data in the buffer is not enough for the current transaction, IP will not send out the packet and wait additional data from user.

- **Tx Packet Buffer**

The size is set by "TxPacBitWidth" parameter of the IP. The valid value is 9-11 and the description of the parameter is shown in Table 4. This buffer size must be at least Tx Packet size (setting in PKL register) to store at least one packet that data is transferred from Tx Data Buffer. Data in Tx Packet buffer is sent out when EMAC and the target are ready to receive data. During sending current packet, the next packet is forwarded from Tx Data buffer. So, the packet can be transmitted continuously.

- **Header RAM**

This RAM is applied to store header part of transmitted packet. The network parameters in the packet are set by register during reset the system through RST register. Some parameters such as Target MAC address and Target port number can be updated by ARP Reply (Client mode), ARP Request (Server mode), and Passive open packet.

- **TxCsum**

This module is desigend to calculate checksum of Tx packet before sending out. The checksum value is set to Header RAM.

- **TxDataMux**

This module is designed to merge header from Header RAM and data from Tx Packet Buffer. After that, full ethernet packet is forwarded to EMAC.

Received Block

- **Rx Buffer**

This is temporary buffer to store all Rx packets from EMAC. The objective of this buffer is used to wait Header Checker to validate the received packet. Only valid TCP data is forwarded to Rx data buffer.

- **Header Checker**

Header in Rx packet is compared with the network parameters, set by user. Received packet will be ignored if any parameter is not matched or checksum is error. Also, if duplicate data is detected (same packet is received), the new data will be ignored.

- **Rx Data Buffer**

This buffer size is set by “RxBufBitWidth” parameter of the IP. The valid value is 9-13. This buffer size is used as received window size of this TCP connection. Setting bigger size of this buffer can increase received performance because the data source can continue sending data without waiting the acknowledge returned from TOE10G-IP which may be delayed from network routing, the process within the data source, or received buffer full. Also, using big buffer increases the chance to rearrange received data when the received packet sequene is swapped from network routing.

User Block

This block is user module for setting and monitoring register interface, writing data to TxFIFO, and reading data from RxFIFO. This module can be designed by simple hardware logic.

10G or 10G/25G Ethernet MAC and PCS/PMA

The reference design uses 10 Gb Ethernet MAC and 10 Gb Ethernet PCS/PMA core from Xilinx. For Ultrascale+ device, it needs to use 10G/25G Ethernet Subsystem to run 10G Ethernet. More details of the IP are provided in following website.

10G Ethernet
https://www.xilinx.com/products/intellectual-property/do-di-10gemac.html
https://www.xilinx.com/products/intellectual-property/10gbase-r.html

10G/25G Ethernet
https://www.xilinx.com/products/intellectual-property/ef-di-25gemac.html

Core I/O Signals

Descriptions of all parameters and I/O signals are provided in Table 5 and Table 6. MAC Interface of the IP is 64-bit AXI4 stream bus.

Table 5: Core Parameters

Name	Value	Description
TxBufBitWidth	9-13	Setting Tx Data buffer size. The value is the address bus size of this buffer.
TxPacBitWidth	9-11	Setting Tx Packet buffer size. The value is the address bus size of this buffer.
RxBufBitWidth	9-13	Setting Rx Data buffer size. The value is the address bus size of this buffer.

Table 6: Core I/O Signals

Signal	Dir	Description
Common Interface Signal		
RstB	In	Reset IP core. Active Low.
Clk	In	156.25 MHz clock input from PHY layer (Xilinx block)
User Interface		
RegAddr[3:0]	In	Register address bus
RegWrData[31:0]	In	Register write data bus. Synchronous to RegAddr signal for write process.
RegWrEn	In	Register write enable pulse. Synchronous to RegAddr and RegWrData signals.
RegRdData[31:0]	Out	Register read data bus. Available the next clock after setting RegAddr.
ConnOn	Out	Connection Status (‘1’: connection is opened, ‘0’: connection is closed)
TimerInt	Out	Timer interrupt. Assert to high for 1 Clk period when timeout is detected. User can read TMO[6:0] register to check interrupt status.
RegDataA1[31:0]	Out	32-bit read value of CMD register (RegAddr=0001b). Bit[0] is busy flag of TOE10G-IP.
RegDataA8[31:0]	Out	32-bit read value of TDL register (RegAddr=1000b). Used to monitor remaining transfer size when ETL register is applied.
RegDataA9[31:0]	Out	32-bit read value of TMO register (RegAddr=1001b). Used to be interrupt status when timeout is found.
Tx Data Buffer Interface		
TCPTxFfFlush	Out	Transmitted buffer in TOE10G-IP is reset. Assert to high for 1 Clk period when connection is closed or IP is reset.
TCPTxFfFull	Out	Transmitted buffer full flag. User needs to stop writing data within 4 clock period after this flag is asserted to high.
TCPTxFfWrEn	In	Transmitted buffer write enable. Assert to write data to Transmitted buffer.
TCPTxFfWrData[63:0]	In	Transmitted buffer write data bus. Synchronous with TCPTxFfWrEn.
Rx Data Buffer Interface		
TCPRxFfFlush	Out	Received buffer in TOE10G-IP is reset. Assert to high for 1 Clk period when connection is opened.
TCPRxFfRdCnt[12:0]	Out	Data counter of received buffer to show total received data in 64-bit unit.
TCPRxFfLastRdCnt[2:0]	Out	Remaining byte of last data in received buffer when total received data is not aligned to 8-byte.
TCPRxFfRdEmpty	Out	Received buffer empty flag. User needs to stop reading data immediately when this signal is asserted.
TCPRxFfRdEn	In	Received buffer read enable. Assert to read data from received buffer.
TCPRxFfRdData[63:0]	Out	Read data from received buffer. Valid the next clock after TCPRxFfRdEn is asserted.

Signal	Dir	Description
MAC Interface		
rx_axis_tdata[63:0]	In	Received data.
rx_axis_tvalid	In	Received data valid signal. Synchronous with rx_axis_tdata.
rx_axis_tlast	In	Control signal to indicate the final word in the frame.
rx_axis_tuser	In	Control signal asserted at the end of received frame to indicate that the frame has an error. '1': normal packet, '0': error packet.
rx_axis_tready	Out	Handshaking signal. Asserted when rx_axis_tdata has been accepted.
tx_axis_tdata[63:0]	Out	Transmitted data.
tx_axis_tkeep[7:0]	Out	Tranmistted data byte enable. Synchronous with tx_axis_tdata.
tx_axis_tvalid	Out	Transmitted data valid signal. Synchronous with tx_axis_tdata.
tx_axis_tlast	Out	Control signal to indicate the final word in the frame.
tx_axis_tuser	Out	Control signal to indicate an error condition. This signal is always '0'.
tx_axis_tready	In	Handshaking signal. Asserted when tx_axis_tdata has been accepted.

Timing Diagram

IP Initialization

For initialization process after RST register=’0’, TOE10G-IP can operate in two modes depending on SRV register setting, i.e. Client mode and Server mode.

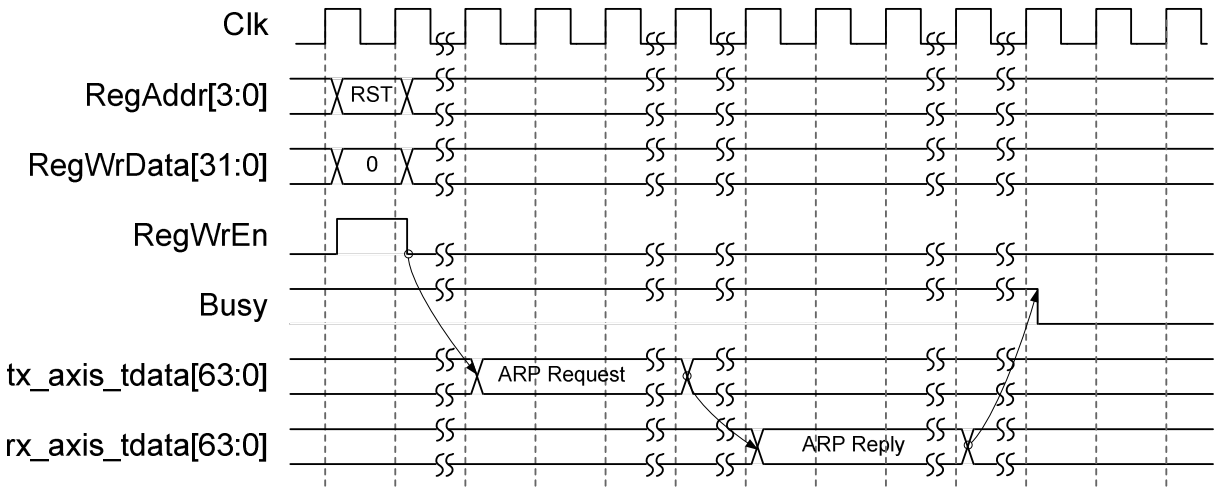

Figure 6: IP Initialization in Client mode

In Client mode, TOE10G-IP sends ARP request and waits ARP reply from the target. Target MAC address is extracted from ARP reply packet. After that, Busy signal is de-asserted to ‘0’.

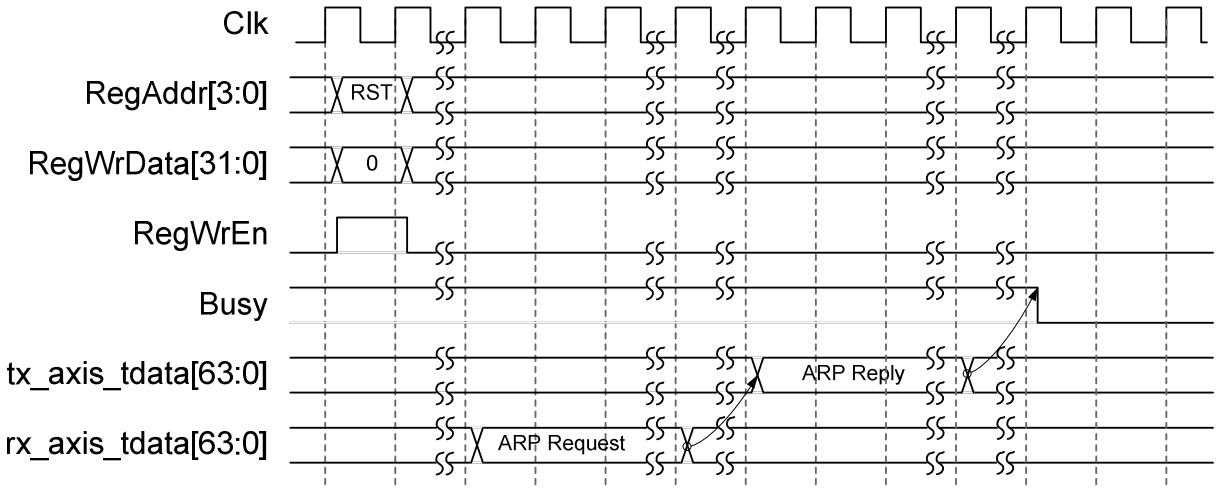

Figure 7: IP Initialization in Server mode

In Server mode, after TOE10G-IP reset is released to ‘0’, TOE10G-IP waits ARP request from the target. After receiving ARP request which the header is matched to setting value of network parameters, TOE10G-IP returns ARP reply to the target. Target MAC address is extracted from ARP request packet. Finally, busy signal is de-asserted to ‘0’.

Register Interface

User can access control signals of TOE10G-IP by using Register interface. The timing diagram of register interface is shown in Figure 9. Register map address is designed as shown in Table 3. To write register, user sets RegWrEn=’1’ with valid value of RegAddr and RegWrData. To read control signal, user sets valid RegAddr and reads RegRdData in the next clock.

Before user sets CMD register, busy flag (RegAddrA1[0] signal) must be monitored until it is equal to ‘0’ (IP is in Idle status). After CMD register is set, busy flag is asserted to ‘1’, as shown in Figure 9. Busy signal is de-asserted to ‘0’ when the command is completed.

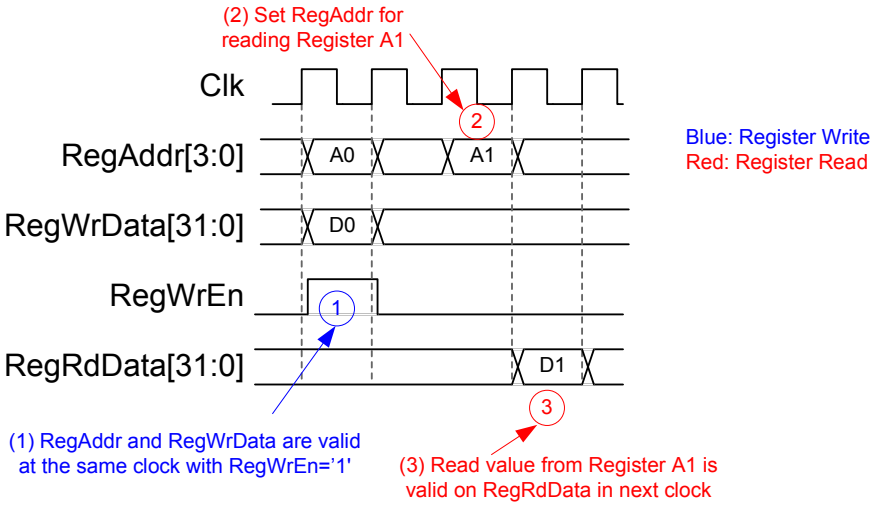

Figure 8: Register Interface Timing Diagram

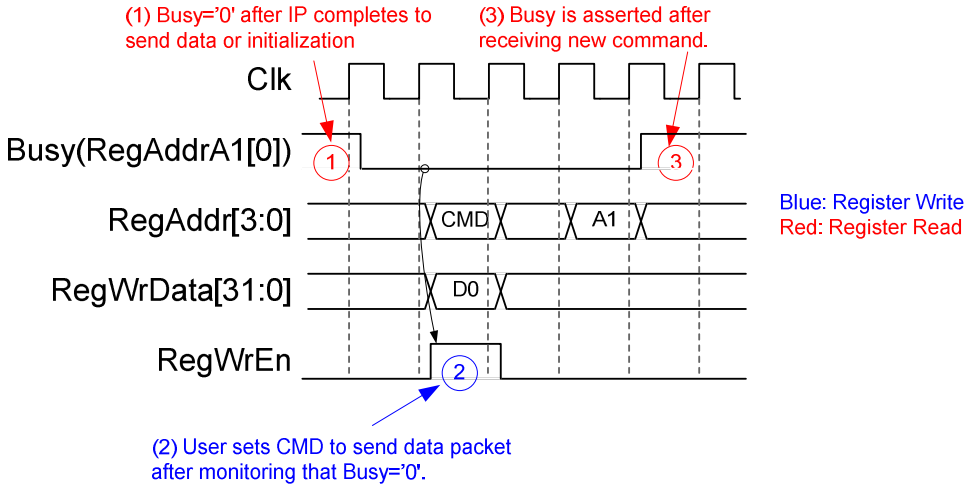

Figure 9: Set CMD register when busy is de-asserted

Tx FIFO Interface

User sends data to TOE10G-IP by using FIFO interface, as shown in Figure 10. Before sending data, user needs to check that full flag (TCPTxFfFull) is not asserted to '1' and ConnOn is equal to '1'. Next, set TCPTxFfWrEn='1' with valid TCPTxFfWrData. TCPTxFfWrEn must be cleared within 4 clocks to stop data sending after TCPTxFfFull is asserted to '1'. TCPTxFfFlush is asserted to '1' from TOE10G-IP to notify that all data in TxFIFO are flushed (connection is closed or IP is reset).

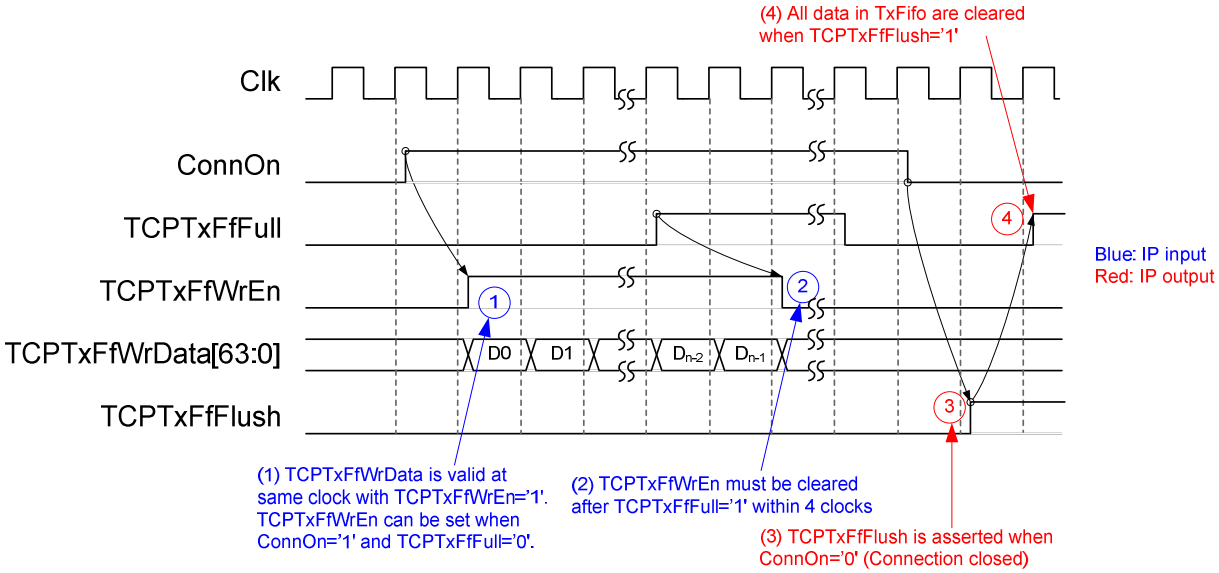

Figure 10: Tx Data Buffer Interface Timing Diagram

Rx FIFO Interface

The received data extracted from valid received packet is stored in Rx Data buffer. User can read data from the buffer through RxFIFO interface, as shown in Figure 11. RxFIFO status is monitored through TCPRxFfEmpty signal (data can be read when TCPRxFfEmpty is cleared to '0'). TCPRxFfRdEn is asserted to '1' to read data from Rx data buffer. TCPRxFfRdData is valid in the next clock. Data reading must stop immediately (TCPRRxFfRdEn='0') when TCPRxFfEmpty = '1' (same clock). All data in Rx data buffer are flushed when the new connection is opened (TCPRxFfFlush is also asserted to '1').

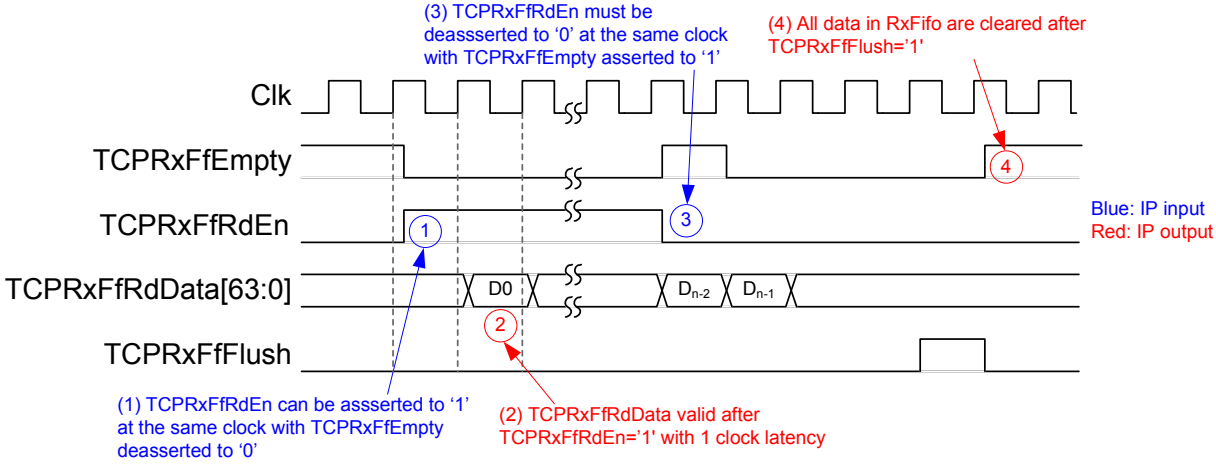

Figure 11: Rx Data Buffer Interface by Empty flag Timing Diagram

Rx data buffer status can be also monitored by using TCPRxFfRdCnt. RdCnt is used when the logic to read data is designed as burst transfer. RdCnt shows total data in Rx data buffer. So, user can assert TCPRxFfRdEn='1' for many clocks to read more than one data from the buffer, as shown in Figure 12.

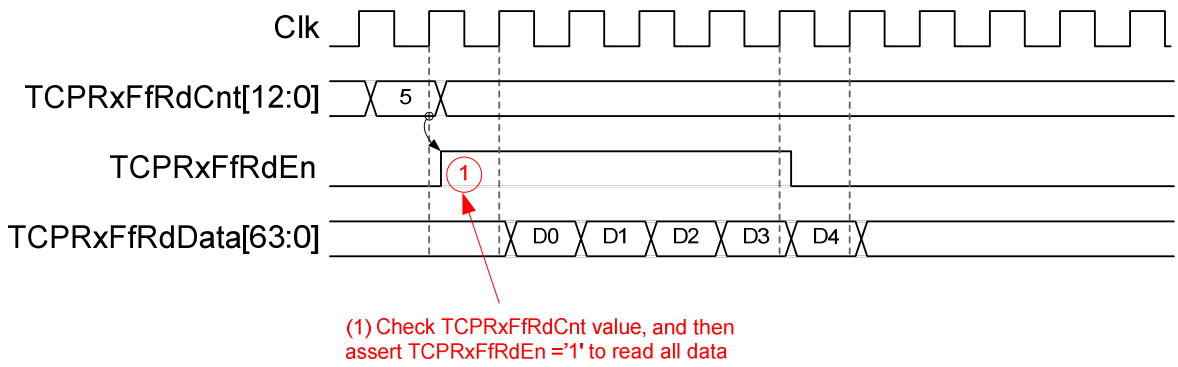

Figure 12: Rx Data Buffer Interface by Read counter Timing Diagram

EMAC Interface

To transmit packet, TOE10G-IP asserts tx_axis_tvalid with the first data of the packet. The signals is latched until tx_axis_tready is asserted to '1' to acknowledge data transmission request. After that, tx_axis_tready must be asserted to '1' until the packet is end (total packet is transferred continuously). tx_axis_tlast and tx_axis_tvalid are asserted to '1' with the last transmitted data to show end-of-packet status.

The new version of Xilinx EMAC de-asserts tx_axis_tready signal between 1st tvalid and tlast signal. TOE10G-IP cannot support this feature. So, the adapter logic with small buffer to connect between TOE10G-IP and Xilinx EMAC must be designed. This adapter logic has been provided in the reference design as HDL code.

Also, 10G/25G Xilinx Ethernet MAC does not include zero padding function for small packet (packet size is less than 60 byte). So, the adapter logic in the reference design for Ultrascale+ device adds zero padding to ethernet packet when total size is less than 60 byte.

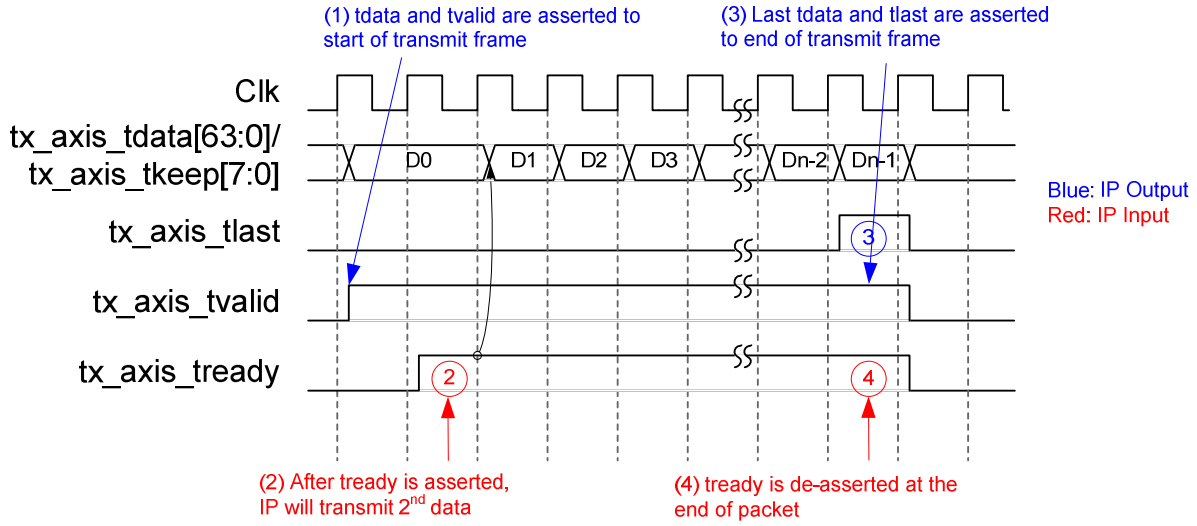

Figure 13: Transmitted EMAC Interface

Figure 14 shows timing diagram of received side. TOE10G-IP monitors start of received frame from rx_axis_tvalid which changes from '0' to '1'. rx_axis_tdata is transferred continuously until rx_axis_tlast is asserted at the end of packet. So, rx_axis_tvalid must be always asserted to '1' during sending one packet (between 1st tvalid and tlast). rx_axis_tuser is also valid at the same time that rx_axis_tlast='1'.After receiving one packet, rx_axis_tready is de-asserted to '0' to pause data transmission for 2 clocks to complete to verify received packet. So, new packet must not be transferrred during tready='0'.

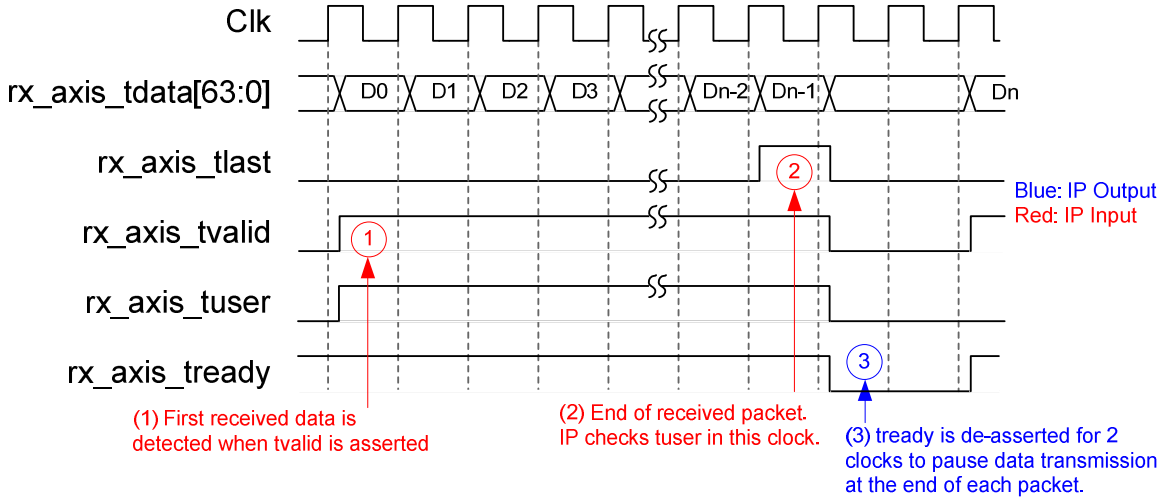

Figure 14: Received EMAC Interface

Example usage

Client mode (SRV[0]='0')

The example of the sequence to set register for data transmission and reception in Client mode is shown as follows.

1) Set RST register='1' to reset the IP.
2) Set SML/SMH for MAC address, DIP/SIP for IP address, and DPN/SPN for port number. (DPN is optional setting in case of the port is opened by IP or active open).
3) Set RST register='0' to clear the reset and then IP starts initialization by sending ARP request to get Target MAC address from ARP reply. After end of intialization, busy signal (RegAddrA1[0]) is cleared to '0'.
4) The new connection is created by two modes.
   a. Active open: set CMD register to create the connection (SYN packet is sent from TOE10G-IP).
   b. Passive open: wait until "ConnOn" signal = '1' (SYN packet is sent from the target).
5) a. For data transmission, set TDL for total transmitted length and PKL for packet size. Next, set CMD register to start data transmission. Transmitted data is also fed to TxFIFO. Busy flag is monitored until it is equal to'0'. For next transmission, user can set new value of TDL/PKL value and set CMD register without IP reset.
   b. For data reception, user monitors RxFIFO status and read data until RxFIFO is empty.
6) Similar to create the connection, the connection is destroyed by two modes.
   a. Active close: set CMD register to close the connection (FIN packet is sent from TOE10G-IP).
   b. Passive locse: wait until "ConnOn" signal = '0' (FIN packet is sent from the target).

Server mode (SRV[0]='1')

The different point between Server mode and Client mode is the initialization process to get MAC address of the target. In Client mode, MAC address is received from ARP reply after TOE10G-IP sends ARP request. In Server mode, MAC address is decoded from ARP request which has matched Target IP address. The process to send and receive data is same as Client mode. The example sequence of Server mode is shown as follows.

1) Set RST register='1' to reset the IP.
2) Set SML/SMH for MAC address, DIP/SIP for IP address, and DPN/SPN for port number.
3) Set RST register='0' to clear the reset. IP starts initialization by waiting ARP request to get Target MAC address. Next, IP creates ARP reply to the Target. After end of intialization, busy signal is cleared to '0'.
4) Remaining steps are similar to step 4 – 6 of Client mode

Verification Methods

The TOE10G-IP Core functionality was verified by simulation and also proved on real board design by using KC705/VC707/ZC706/KCU105/ZCU102 evaluation board.

Recommended Design Experience

User must be familiar with HDL design methodology to integrate this IP into their design.

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gatway Co., Ltd. For pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision History

Revision	Date	Description
1.0	May-29-2014	New release
1.1	Sep-9-2014	Update IP to support full-duplex
1.2	Sep-24-2014	Update valid value of Rx Data Buffer in page7
1.3	Nov-14-2014	Add ZC706 board support
1.4	Oct-20-2015	Add PSH, WIN, and ETL register, and RegDataAx port
1.5	Nov-3-2015	Update read value of CMD[3:1] register
1.6	Dec-23-2015	Update register to support readback
1.7	Feb-23-2017	Add KCU105 support and TCPRxFfLastRdCnt signal
1.8	Jan-18-2018	Add SRV register and rx_axis_tready signal