Design Gateway Co.,Ltd
54 BB Building 13th Fl., Room No.1302 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110
Phone: (+66) 02-261-2277
Fax: (+66) 02-261-2290
E-mail: 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
Simulation Tool Used	
ISim/Vivado Simulator/ModelSim	
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- TCP/IP stack implementation
- Support IPv4 protocol
- Support one port connection
- Support both Server and Client mode (Passive/Active open and close)
- Support Jumbo frame
- Packet size for transmit and total received data must be 8-byte (64-bit) aligned
- Transmit/Receive buffer size, adjustable for optimized resource and performance
- Simple data interface by standard FIFO interface
- Simple control interface by standard register interface
- One clock domain interface by fixed 156.25 MHz clock frequency
- Half-duplex and Full-Duplex Reference design available on KC705/VC707/ZC706 evaluation board

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices[1]	IOB[2]	RAMB36E1	RAMB18E1	Design Tools
Kintex-7	XC7K325TFFG900-2	156.25	3009	3287	1227	363	38	-	Vivado2014.1
Virtex-7	XC7VX485TFFG1761-2	156.25	3009	3287	1219	363	38	-	Vivado2014.1

Notes:

1) Actual logic resource dependent on percentage of unrelated logic

2) Assuming all core I/Os and clocks are routed off-chip

3) Block memory resources are based on 64kB Tx data buffer size, 16kB Tx packet buffer size, and 64kB Rx data buffer size. Minimum size of each buffer are 4kB Tx data buffer size, 4kB Tx packet buffer size, and 16kB Rx data buffer size for jumbo frmae.

4) For XC7Z045FFG900-2 (ZC706 board) device, maximum I/O pin on Zynq-7000 series is 362, so map report cannot be generated. Please refer the resource by using Kintex-7 series because of device compatibility.

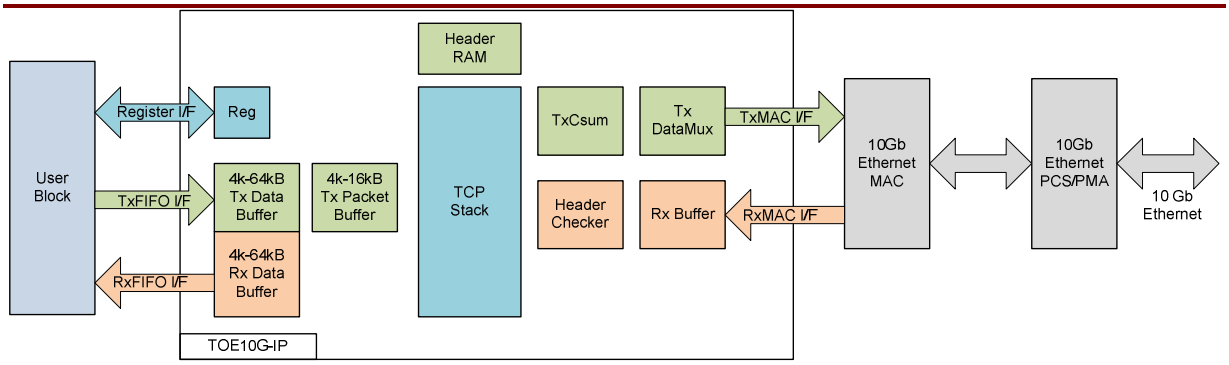

Figure 1: TOE10G-IP Block Diagram

Applications

TOE10G-IP is designed for network application using TCP/IP protocol for data reliability with high speed performance by using 10 Gb Ethernet. Ethernet data recorder or receiver can be designed by using the IP without CPU requirement.

General Description

TOE10G-IP core operating with 10 Gb EMAC IP and 10 Gb Ethernet PCS/PMA, provided by Xilinx, can operate 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 any network device through TCP/IP protocol by using this system.

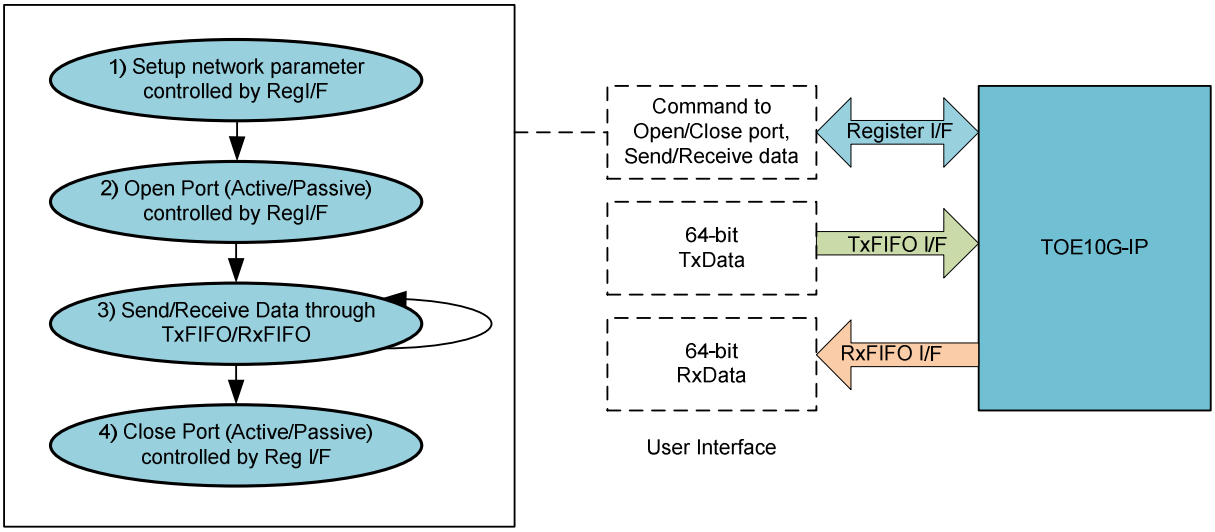

Figure 2 TOE10G-IP User Interface and operation sequence

There are three types of user interface, i.e. control signal by register access, transmit and received data signal by FIFO access. During initializing system, user needs to set up system parameter such as packet size, port number, IP address through register interface. After that, port can be opened by user logic (Active mode) or by external device (Passive mode) before start data transferring. To send/receive data, the interface at user side is typical FIFO interface, so user can write/read data by using simple design. If there is no more data transferring, the port can be closed by user logic (Active mode) or by external device (Passive mode), like opening the port.

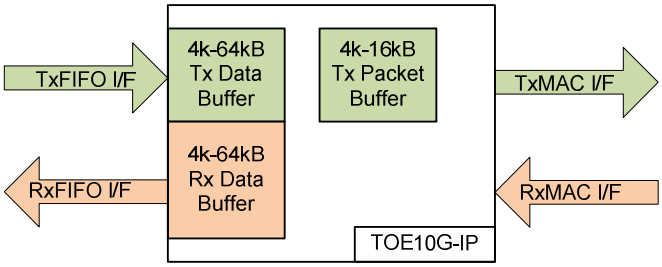

Figure 3 Adjustable Tx/Rx Buffer Size

Three buffers in TOE10G-IP, i.e. Tx Data buffer, Tx Packet Buffer, and Rx Data Buffer can set the size by setting parameter of the IP. The different size is provided to optimize resource utilization for user application. The bigger size takes much resource, but can achieve better transfer performance. Tx Data buffer size and Tx Packet buffer size are effect to transmit performance, while Rx Data buffer is effect to receive performance. Otherwise, Tx Data buffer size and Tx Packet buffer size are related to the packet size which user can be programmed through register interface. Tx Packet Buffer must be more than the Tx packet size while Tx Data Buffer size should be at least two times of the Tx packet size.

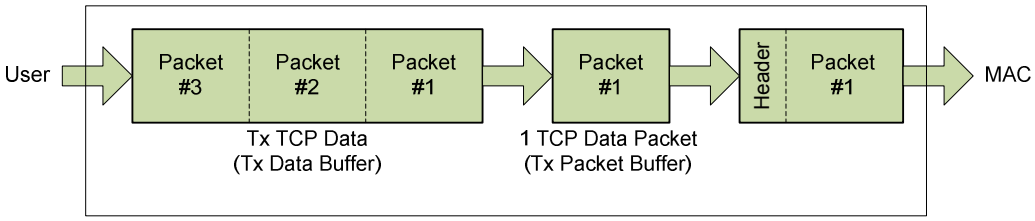

Figure 4 Transmit Data Flow

To transmit data, data from Tx Data buffer will be split into packet size and then fed to Tx Packet buffer. Data output from Tx Packet buffer will be combined with header data in Header RAM before sending out to EMAC. TCP and IP checksum will be auto calculated within TOE10G-IP. Acknowledge number of Rx packet will be monitored to make decision to send next data packet for normal case or resend data packet in Tx buffer for packet lost case. Busy flag within register will be cleared after completed data transfer size is equal to setting value from user. User can monitor this busy flag to check transfer status.

User can change packet size and total transfer size for new transmit without closing the port if the IP is Idle state.

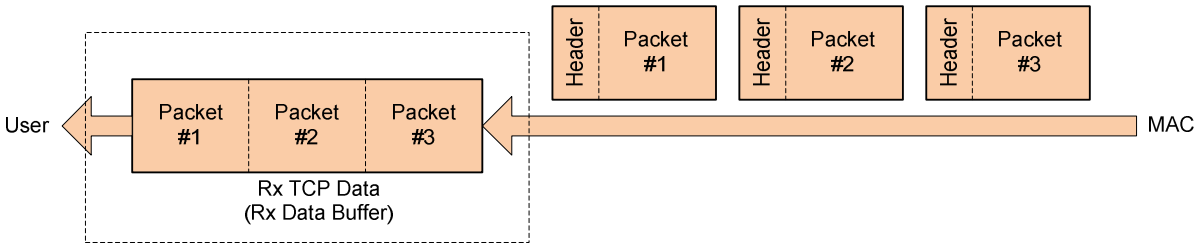

Figure 5 Received Data Flow

For receiving data, Rx packet will be stored to temp buffer firstly. Header and checksum within Rx packet will be verified. If header or checksum is error, the packet will be ignored and not store to Rx Data buffer.

When correct data packet is received, data will be stored to Rx Data buffer and Acknowledge packet will be sent out from TOE10G-IP to request more data packet from external network device. TOE10G-IP will go back to Idle state (Busy flag='0') when no more packet is received and received packet sequence is correct.

Functional Description

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

Control Block

- **Reg**

  User can set parameters for TCP/IP operation by using register interface. Register address of this interface is equal to 4-bit for 11 registers. The description of each register address is defined as shown in Table 2. After system reset is released, all internal parameter will be updated by setting value in each register.

- **TCP Stack**

  To operate active command from user, TCP Stack will decode user command to transmit packet for opening the port, transferring data, or closing the port. During transferring packet, received packet will be also monitored to check the status of external device.

  To operate passive command from external device, the received packet will be decoded by TCP Stack, and then process the command for opening the port, receiving the data, or closing the port. Packet will be transmit out to send acknowledge and status to external device.

Table 2 Register map Definition

RegAddr [3:0]	Reg Name	Dir	Bit	Description
0000b	RST	Wr	[0]	Reset IP. '0': Release reset, '1': Reset. Default value is '1'. **After user set all parameters, set '0' to this register to release reset and start system parameter intialization. Reset needs to be set/clear again to reload parameter if the value of SML, SMH, DIP, SIP, DPN, or SPN register is changed by user.**
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 start any active command, user needs to check system busy flag by reading bit[0] of this register to confirm that there is no operation running.** Active command will auto-start after this register is set by user.
		Rd	[0]	System busy flag. '0': Idle, '1': System operating.
			[3:1]	Current operation. "000": Send data with/without received data, "001": Idle, "010": Active open connection, "011": Active close connection, "100": Receive data, "101": Undefined, "110": Passive open connection, "111": Passive close connection.
0010b	SML	Wr	[31:0]	Define 32-bit lower MAC address (bit [31:0]) for this IP. **User needs to set this register before clearing RST register.**
0011b	SMH	Wr	[15:0]	Define 16-bit upper MAC address (bit [47:32]) for this IP. **User needs to set this register before clearing RST register.**
0100b	DIP	Wr	[31:0]	Define 32-bit target IP address. **User needs to set this register before clearing RST register.**

RegAddr [3:0]	Reg Name	Dir	Bit	Description
0101b	SIP	Wr	[31:0]	Define 32-bit IP address for this IP. **User needs to set this register before clearing RST register.**
0110b	DPN	Wr	[15:0]	Define 16-bit target port number. **User needs to set this register before clearing RST register if user wants to use active open connection.** Target port number will be auto defined from passive open connection packet which parameters in header are matched with setting value.
0111b	SPN	Wr	[15:0]	Define 16-bit port number for this IP. **User needs to set this register before clearing RST register.**
1000b	TDL	Wr	[31:0]	Total Tx data length transfer in byte unit, but the size must be aligned to 8-byte. Valid from 8-0xFFFFFFF8 (Bit[2:0] will be ignored by the IP). **User needs to set this register before setting CMD register = "00". This value will be latched to internal logic when CMD register is set. So, user can prepare the new value for next transmit after setting CMD register.** If user transmit data with same length, this register will not need to set again. Previous value will be used from internal latch.
		Rd	[31:0]	Remain data transfer length in byte unit which still not transmit.
1001b	TMO	Wr	[31:0]	Define timeout value for waiting Rx packet during running any command. This register is used by 156.25 MHz counter, so timer unit is about 6.4 ns. This value must be more than 0x6000.
		Rd		[0]-Timeout from not receiving ARP reply packet [1]-Timeout from not receiving SYN and ACK flag during active open operation [2]-Timeout from not receiving ACK flag during passive open operation [3]-Timeout from not receiving FIN and ACK flag during active close operation [4]-Timeout from not receiving ACK flag during passive close operation [5]-Timeout from not receiving ACK flag during data transmit operation [6]-Timeout from Rx packet lost, Rx data FIFO full, or wrong sequence number [23]-Rx packet ignored because of Rx data buffer full [27]-Rx packet lost detected [30]-RST flag is detected in Rx packet
1010b	PKL	Wr	[15:0]	Data length of Tx packet in byte unit, but packet length 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 will be ignored by the IP. **This value must not be changed during data transmission not complete (Busy='1'). If next transmit use same packet size, user will not need to set this register because the previous value is latched in the logic.**

Note:

1. Target Mac address is defined from returned value in ARP Reply packet, so user does not need to set this parameter.
2. Target Port number is defined from received packet when the port is opened in passive mode. So, in this mode, user does not need to set Target Port number.

Table 3 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

Transmit 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 3.

The buffer size should be at least two times of Tx Packet Size in PKL register for sending data continuously. Transmit data from user will be stored within this buffer until the target returns acknowledge packet to confirm that data can be received completely. One packet data will be read out from this buffer to store to Tx Packet Buffer and mixed with the header to generate TCP packet.

This buffer size relates to the total performance. Bigger size can make the IP sending out the data continuously though acknowledge returned from the target has much latency from the processing and the carrier.

If user sends data more than the total transmit size, remain data will be available in the buffer for next transfer. The data will be flushed when the port 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 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 3. This buffer size must be more than or equal to Tx Packet size (setting in PKL register) to store at least one packet data splitting from Tx Data Buffer. Data in Tx Packet Buffer is stored to wait until EMAC and target ready to receive data before sending out.

- **Header RAM**

This RAM is applied to store header part of Transmit packet. Parameter in Header RAM will be updated by register value when user release RST register. Some parameters such as Target MAC adderss and Target port number will be updated by ARP Reply or Passive open packet.

- **TxCsum**

This module is desigend to calculate checksum of Tx packet before sending out.

- **TxDataMux**

This module is desigend to combine header from Header RAM and data from Tx Packet Buffer, and then send out to EMAC.

Received Block

- **Rx Buffer**

  This is temporary buffer to store each Rx packet from EMAC for waiting Header Checker processing.

- **Header Checker**

  Header in Rx packet will be checked and compared with parameter in register. Packet will be ignored if any parameter is not matched or checksum is error. Only TCP data will be split out and store into Rx Data Buffer.

- **Rx Data Buffer**

  This buffer size is set by “RxBufBitWidth” parameter of the IP. The valid value is 9-13 and the description of the parameter is shown in Table 3. This buffer size will be mapped to be window size of this TCP connection. Setting bigger size to this buffer can increase received data 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 can help TOE10G-IP to store the received packet and rearrange data when the received packet sequence 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.

10 Gb Ethernet MAC and 10 Gb Ethernet PCS/PMA

Both blocks are softIPcore provided by Xilinx.

Core I/O Signals

Descriptions of all parameters and I/O signals are provided in Table 4 and Table 5. Signals in MAC Interface group are designed to connect to Xilinx EMAC port directly.

Table 4: Core Parameters

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

Table 5: Core I/O Signals

Signal	Dir	Description
Common Interface Signal		
RstB	In	Reset IP core. Active Low.
Clk	In	156.25 MHz fixed clock frequency input from PHY layer of 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 after asserting RegAddr with 1 Clk period latency
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.
Tx Data Buffer Interface		
TCPTxFfFlush	Out	Transmit buffer within IP is reset. Assert to high only 1 Clk period when connection is closed or reset.
TCPTxFfFull	Out	Transmit buffer full flag. User needs to stop writing data within 4 clock period after this flag is asserted to high.
TCPTxFfWrEn	In	Transmit buffer write enable. Assert to write data to Transmit buffer.
TCPTxFfWrData[63:0]	In	Transmit buffer write data bus. Synchronous with TCPTxFfWrEn.
Rx Data Buffer Interface		
TCPRxFfFlush	Out	Received buffer within IP is reset. Assert to high only 1 Clk period when connection is opened.
TCPRxFfRdCnt[12:0]	Out	Received buffer data counter to show total number of 64-bit received data in buffer.
TCPRxFfRdEmpty	Out	Received buffer empty flag. User needs to stop reading data immediately.
TCPRxFfRdEn	In	Received buffer read enable. Assert to read data from Received buffer.
TCPRxFfRdData[63:0]	Out	Received buffer read data bus. Valid after TCPRxFfRdEn assert with 1 Clk period latency.

Signal	Dir	Description
MAC Interface		
rx_axis_tdata[63:0]	In	Received data bus.
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.
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 to EMAC. 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

User can write/read control signal with TOE10G-IP by using Register interface which has timing diagram as shown in Figure 6. Register map address is designed as shown in Table 2. To write control signal, User needs to set RegWrEn='1' with valid value of RegAddr and RegWrData. To read control signal, User set only RegAddr value and then RegRdData will be valid in next clock period.

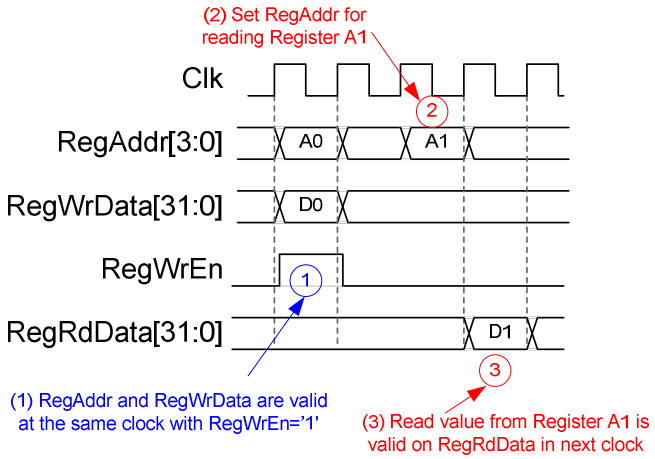

Figure 6: Register Interface Timing Diagram

User can send data to IP core by using FIFO interface, as shown in Figure 7. Before sending data, user needs to check full flag (TCPTxFfFull) that is not asserted to '1' and ConnOn is equal to '1'. Then, set TCPTxFfWrEn='1' with valid value of TCPTxFfWrData. TCPTxFfWrEn must be cleared within 4 clock period to stop data sending after TCPTxFfFull is asserted to '1'. TCPTxFfFlush will be asserted to '1' from IP core to inform user that all data in TxFIFO are cleared which is caused from close connection detect.

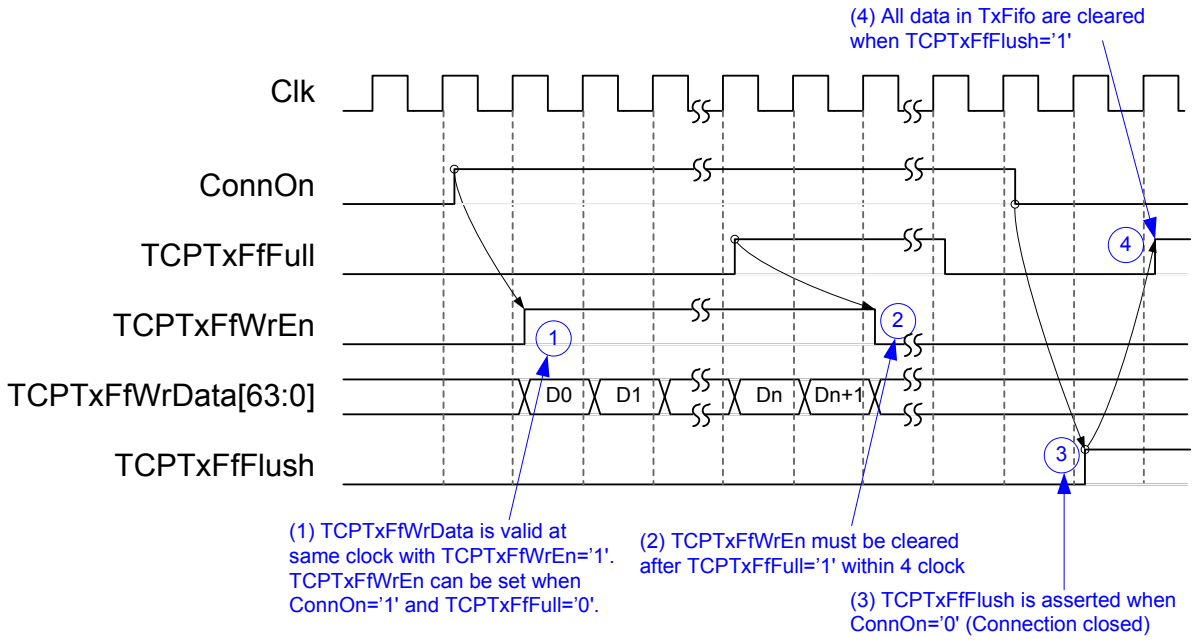

Figure 7: Tx Data Buffer Interface Timing Diagram

When IP core receives data from external, data will be stored in Rx Data buffer. User can read data from this buffer by using FIFO interface, as shown in Figure 8. User can monitor data available status from TCPRxFfEmpty. Data can be read when TCPRxFfEmpty is cleared to '0'. TCPRxFfRdEn can be set to '1' to read data from Rx data buffer and TCPRxFfRdData will be valid in next clock period. Data reading must be stopped by clearing TCPRRxFfRdEn='0' at the same clock with TCPRxFfEmpty setting = '1'. All data in Rx data buffer will be flushed from open connection detect, so user can monitor flush status from TCPRxFfFlush signal.

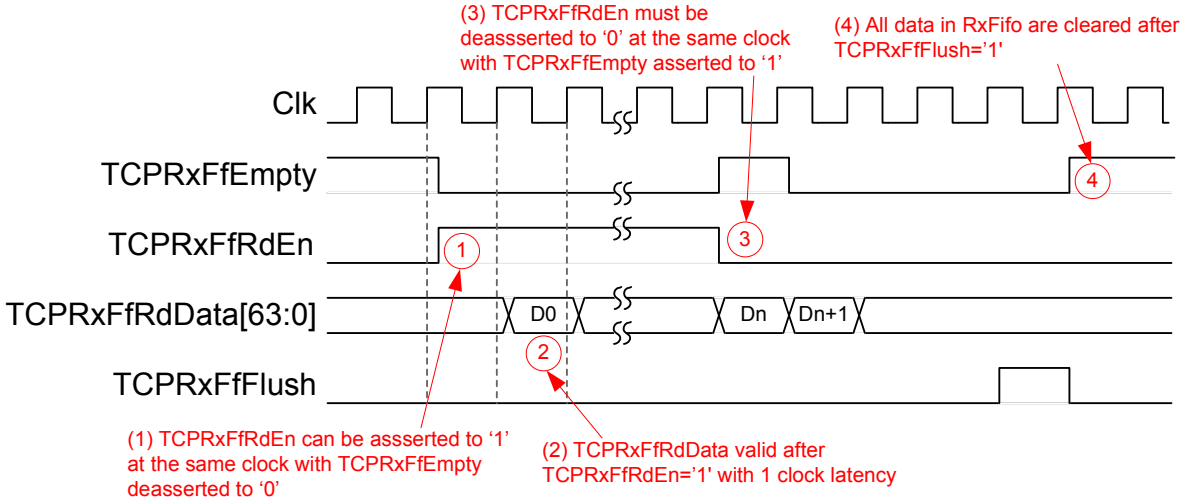

Figure 8: Rx Data Buffer Interface by Empty flag Timing Diagram

Rx data buffer status can be also monitored by using TCPRxFfRdCnt. This signal shows total number of available 64-bit data in Rx data buffer. So, user can assert TCPRxFfRdEn='1' for time period equal to total number of data, as shown in Figure 9.

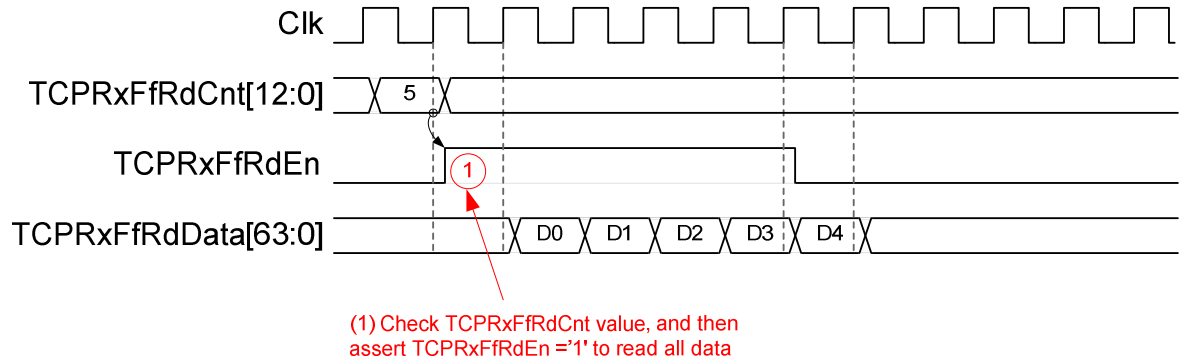

Figure 9: Rx Data Buffer Interface by Read counter Timing Diagram

Example usage

The example of register setting sequence for data transmission and reception is follows.

1) Set SML/SMH for MAC address, DIP/SIP for IP address, and SPN/DPN for port number (DPN is optional setting when port will be opend by IP or active open).
2) Set RST register to release reset
3) a. Active Open: set CMD to open the port

   b. Passive Open: wait “Connon” = ’1’
4) a. For data transmission, set TDL/PKL for total transfer length and packet size, and then set CMD to start data transmission. User can send total transmit data to TXFIFO, and wait busy flag=’0’ by monitoring CMD register.

   b. For data reception, user monitors RxFIFO status and dump data out until RxFIFO empty.

   When TOE10G-IP is in Idle state (no more data transmit or received), user can change packet size and total transfer size (TDL/PKL) for new transmit without closing the port. So, step 4) can run many times until the port closed.
5) a. Active Close: set CMD to close the port

   b. Passive Close: wait “Connon” = ‘0’

Verification Methods

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