# TOE10G-IP Full Duplex Reference design manual

Rev1.0 19-May-16

## 1. Introduction

Please read “dg_toe10gip_refdesign_altera_en.pdf” document which describes about standard demo of TOE10G-IP firstly. For full duplex demo, User Module is modified from standard demo to show the example to transfer data in both directions at the same time while standard demo shows data transfer in one direction. Only modification point will be described in this document.

## 2. Environment

This reference design is based on the following environment as shown in Figure 1.

- Arria10 SoC development board
- Quartus 15.1
- 10-Gigabit SFP+ DAC cable or 2x10-Gigabit SFP+ Transceiver with optical cable
- PC with 10Gigabit Ethernet support or 10Gigabit Ethernet card
- USB Micro-B cable for FPGA configuration
- Test Application “tcp_client_txrx_10G”, provided by Design Gateway

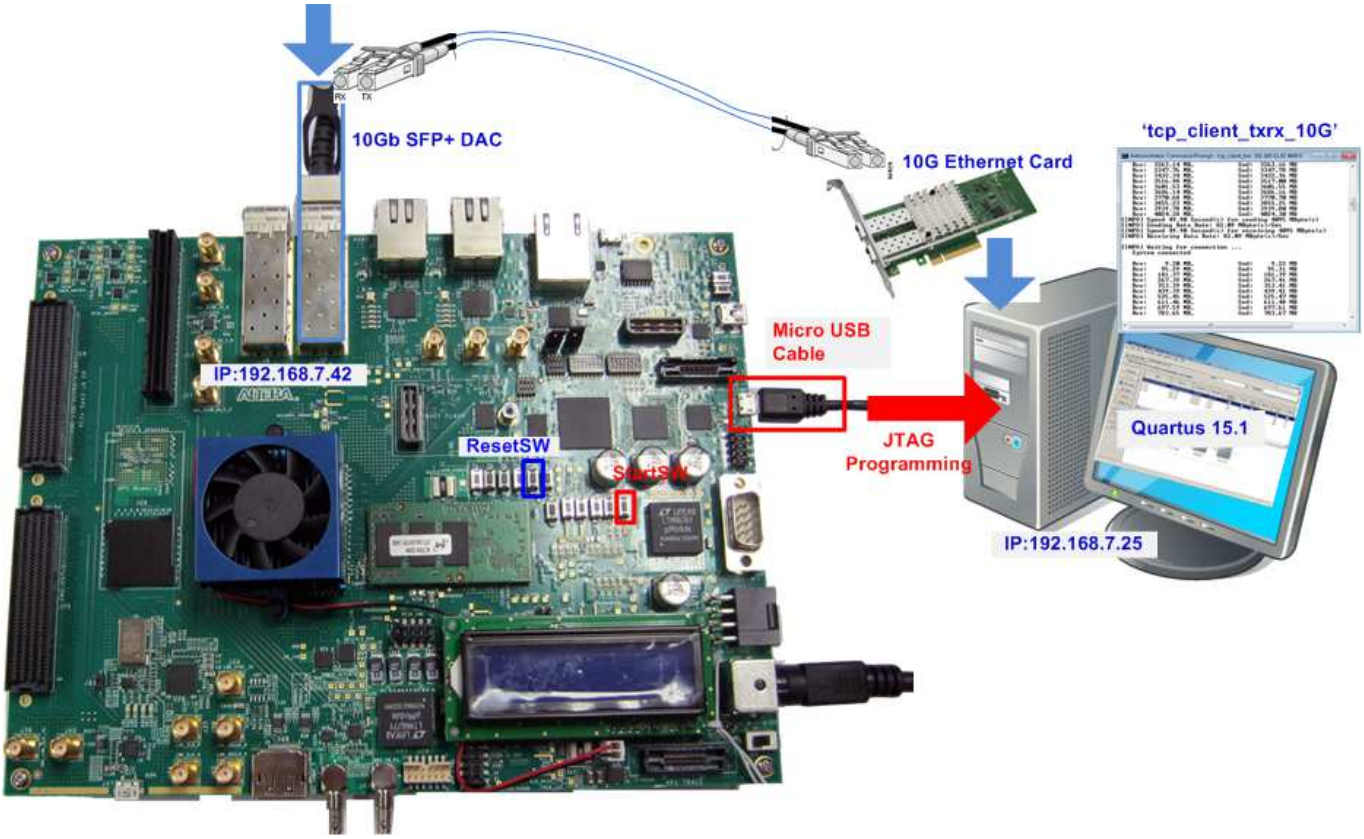


Figure 1 TOE10GIP Demo on Development board

# 3. Hardware description

Comparing to TOE10G-IP standard demo, only User Module is modified to show data sending/receiving with PC at the same time. In the demo, test application on PC will generate test pattern and send out to FPGA board through 10Gigabit Ethernet. TOE10G-IP receives TCP packet and returns test pattern data from PC to User Module through TCPRxFf port. Test logic in User Module loopback test pattern data from TCPRxFf port to TCPTxFf port of TOE10G-IP. So, all test data from PC will be returned to PC and data verification will be processed in test application.

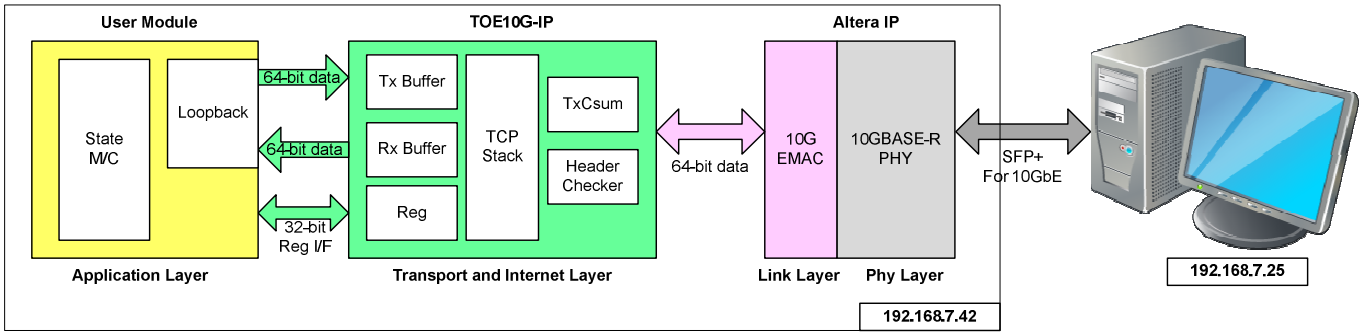


Figure 2 Hardware Architecture in reference design

- User Module

Similar to standard demo, the logic is split into three parts, i.e. TxFIFO interface, RxFIFO interface, and Control interface. Data port from RxFIFO is loopback connected to TxFIFO. Control interface is designed by state machine like standard demo as shown in Figure 4.

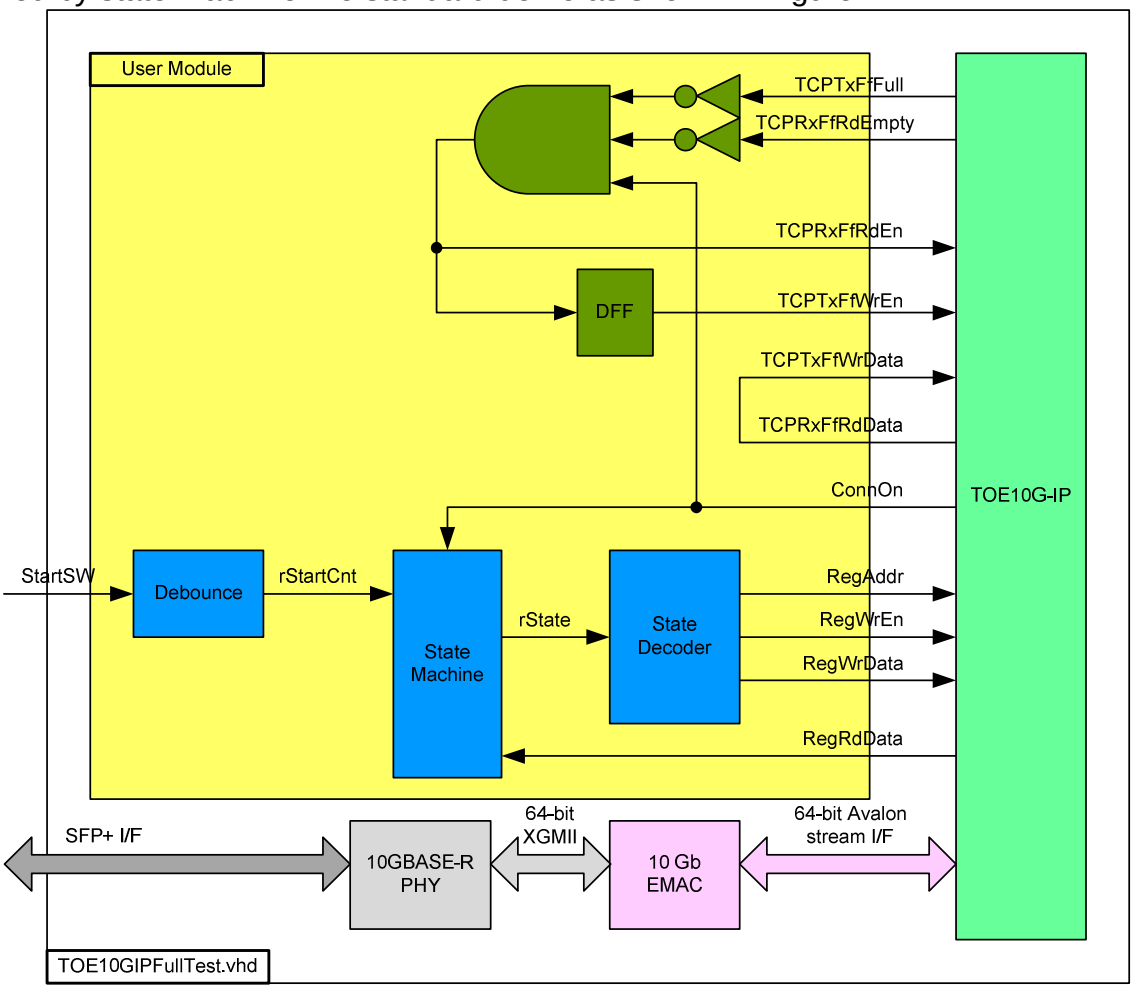


Figure 3 User Module and Test System block diagram

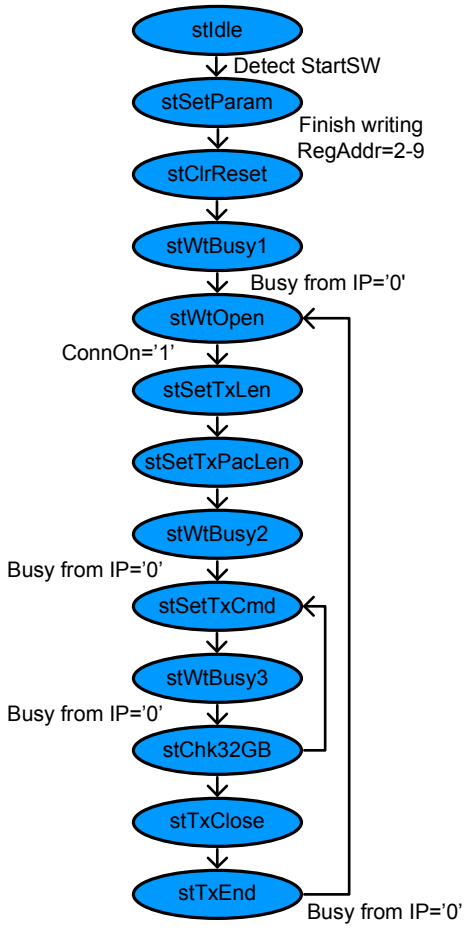


Figure 4 State Machine Diagram within User Module

After system initialization by receiving Start signal from user, state machine will be Idle (stWtOpen state) like TOE10G-IP standard demo. User needs to start test operation by running test application on PC to make new TCP connection between PC and FPGA. After connection is opened by PC, ConnOn will be asserted and transmit parameter will be programmed by state machine. Total transfer size in the demo is fixed to 4 GB (TDL Reg=0xFFFF_FFF8), programmed by stSetTxLen. Transmit Packet size is fixed to 8960 byte (PKL Reg=0x2300), programmed by stSetTxPacLen. Before sending command to transmit data, stWtBusy2 is designed to check Busy status because the IP may be busy during processing received data from PC. State is changed from stSetTxCmd to stWtBusy3 when Busy flag from IP is cleared by all 4 GB data transferred completely. stChk32GB is designed to send 4 GB data for 8 times, so total data transfer size will be equal to 32 GB. stTxClose is designed to send Active Close command to the IP. When port is closed, PC can monitor that current loop is completed.

## 4. Test Software description

"tcp_client_txrx_10G" application is used for running this demo. 32 GB test data will be generated and transferred through 10Gigabit Ethernet. At the same time, receive function will be called to read with/without received data verification. There are three input parameters for the test application, i.e.

- FPGA IP address: This demo sets IP address to "192.168.7.42". User can modify HDL code of User Module to change this value.
- FPGA Port number: This demo sets Port number to "4000". User can modify HDL code of User Module to change this value.
- Verification On/Off:
  '0': Generate dummy data (all 0) for sending function and bypass data verification for receive function. This feature is set to check the best performance on TestPC.
  '1': Generate 32-bit increment for sending function and enable verification process for receive function.

The operation sequence of the application is follows.

(1) Get three parameters from user.
(2) Create socket and then set properties of transmit and receive buffer.
(3) Set IP address and Port number from user parameter and then connect.
(4) Fill test pattern with dummy (all '0') or increment pattern to transmit buffer and send data out. 60 kByte data will be input to sending function.
(5) Receive function will be called after 60 kByte sending function complete to get data from the receive buffer. If verification is enabled, received data will be compared with 32-bit increment data. Error message will be printed out if received data is not correct.
(6) Sending and Receive function are called alternately until total 32 GB data are sent and received. Current transmit and receive size are printed out every second on the console.
(7) Socket is closed by FPGA side and then performance with total transferred data size will be printed out as test result. If total received size is not equal to transmit size, error message will be printed out.
(8) Go back to step (2) to run test in forever loop until operation is cancelled by user.

## 5. Revision History

| Revision | Date | Description |
|---|---|---|
| 1.0 | 19-May-16 | Initial Release |
| | | |