TOE40G-IP Demo Instruction

Rev1.0 27-Dec-18

Contents

1 Overview

The demo is designed to run TOE40G-IP for transferring 40 Gb Ethernet data by using TCP/IP protocol. The destination device may be Test PC or TOE40G-IP on another FPGA as shown in Figure 1-1.

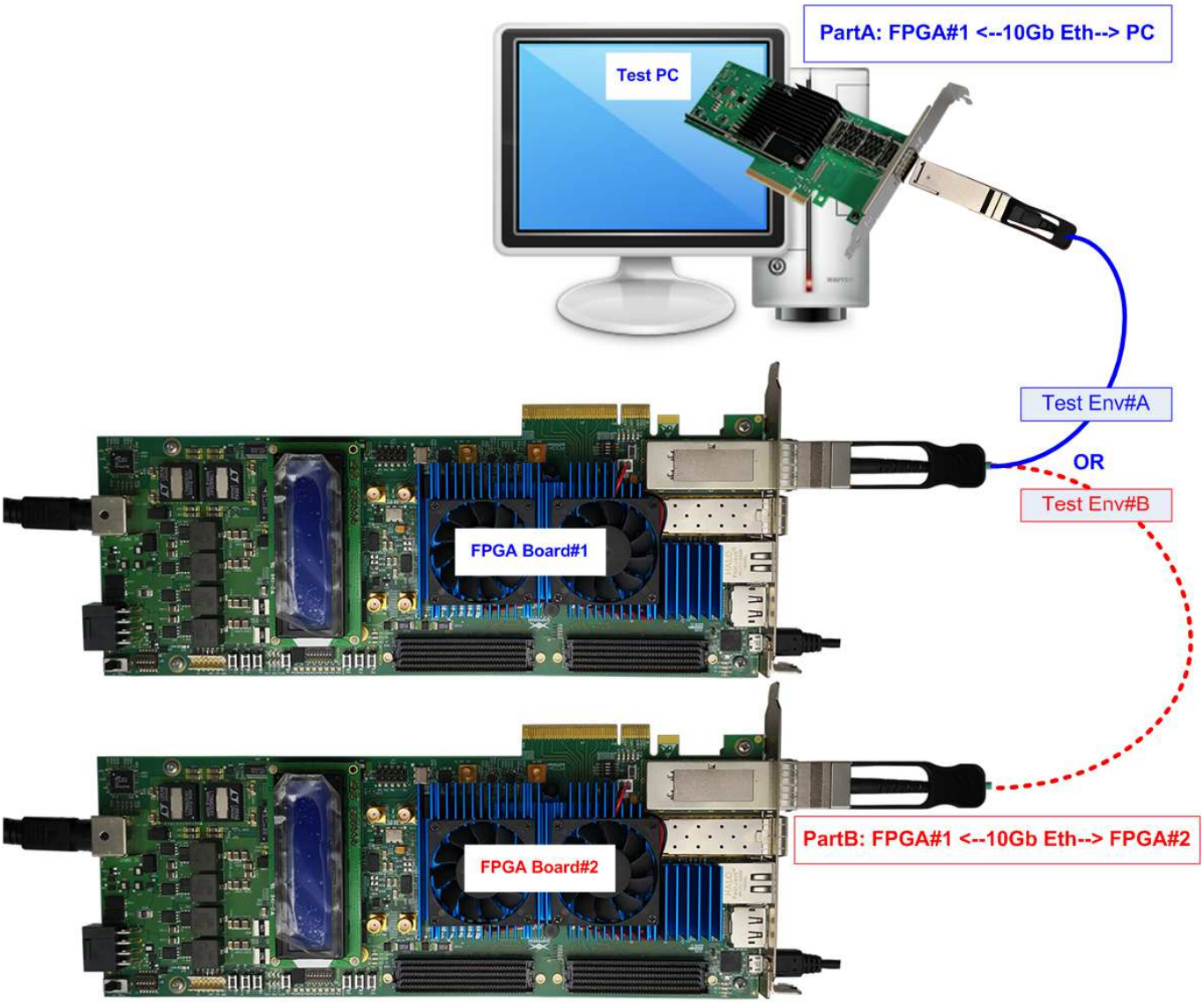

Figure 1-1 Two test environments for the demo

To describe the step for running the demo in more details, the document is split into two parts following the test environment, i.e. “Part A TOE40G-IP demo by using FPGA and PC” and “Part B TOE40G-IP demo by using two FPGAs”. User interface to set test parameters and monitor the operation on FPGA board is NiosII command shell by using JTAG UART.

Part A TOE40G-IP demo by using FPGA and PC

To transfer data between TOE40G-IP and Test PC, user selects to run half-duplex or full-duplex demo. “tcpdatatest” application is run on Test PC for half-duplex demo (sending data from TOE40G-IP to PC or receiving data from PC to TOE40G-IP). “tcp_client_txrx_40G.exe” application is run on Test PC for full-duplex demo (sending and receiving data with PC at the same time). More details of the demo are described as follows.

2 Environment Setup

To operate TOE40G-IP demo, please prepare following test environment.

1) FPGA development boards (Arria10 GX development kit)
2) PC with 40 Gigabit Ethernet support or 40 Gigabit Ethernet card
3) QSFP+ AOC cable for 40G network connection between FPGA board and 40G Ethernet card that plug-in on PC
4) micro USB cable for programming FPGA and JTAG UART, connecting between FPGA board and PC
5) “tcpdatatest.exe” and “tcp_client_txrx_40G.exe” which are test application provided by Design Gateway, installed on PC
6) QuartusII Programmer for programming FPGA and NiosII command shell, installed on PC

Note: Test result in this document is captured by using following test environment.

[1] 40G Network Adapter: Intel XL710-Q1
https://www.intel.com/content/www/us/en/ethernet-products/converged-network-adapters/ethernet-xl710-brief.html

[2] QSFP+ AOC: AOC-Q1Q1-001
http://www.10gtek.com/uploadfile/download/pdf/aoc_datasheet/10Gtek%20AOC-Q1Q1-xxx%20(JH)%20V1.0.pdf

[3] PC: Motherboard ASUS Z170-K, Intel i7-6700K CPU, 32 GB RAM

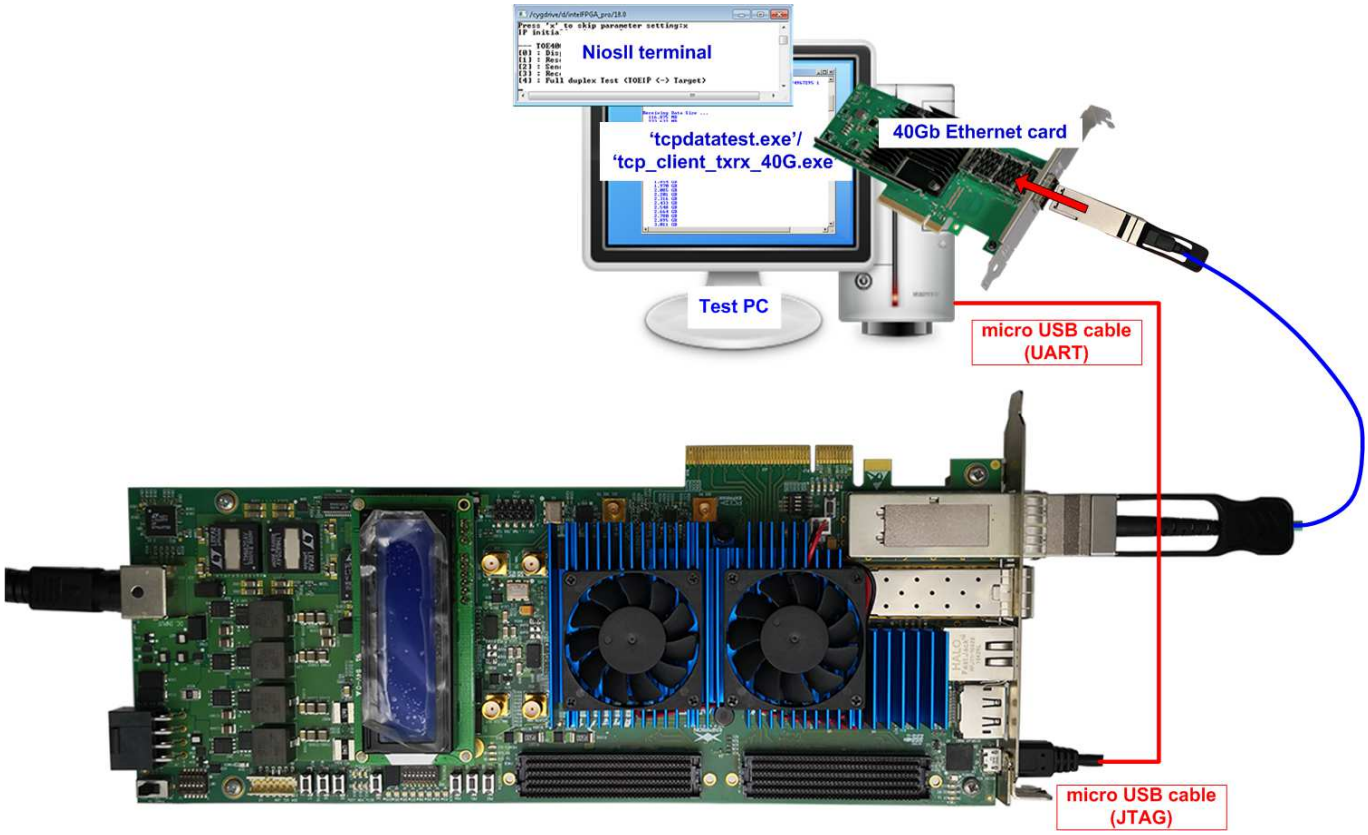

Figure 2-1 TOE40G-IP demo environment setup on Arria10 GX

3 PC Setup

Before running demo, network setting on PC is required. The example to set the network is described as follows.

3.1 IP Setting

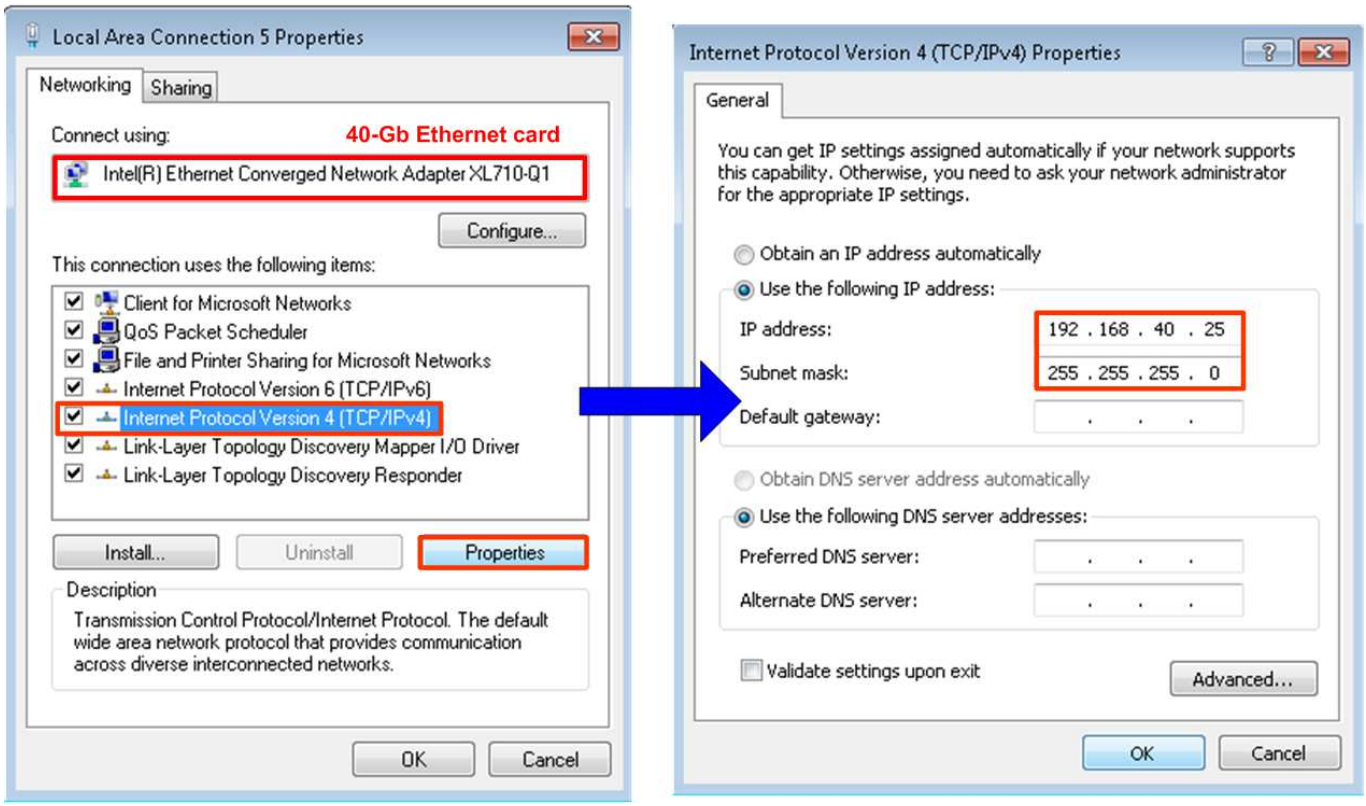

Figure 3-1 Setting IP address for PC

1) Open Local Area Connection Properties of 40-Gb connection, as shown in left window of Figure 3-1.
2) Select “TCP/IPv4” and then click Properties.
3) Set IP address = 192.168.40.25, and Subnet mask = 255.255.255.0, as shown in right window of Figure 3-1.

3.2 Speed and Frame Setting

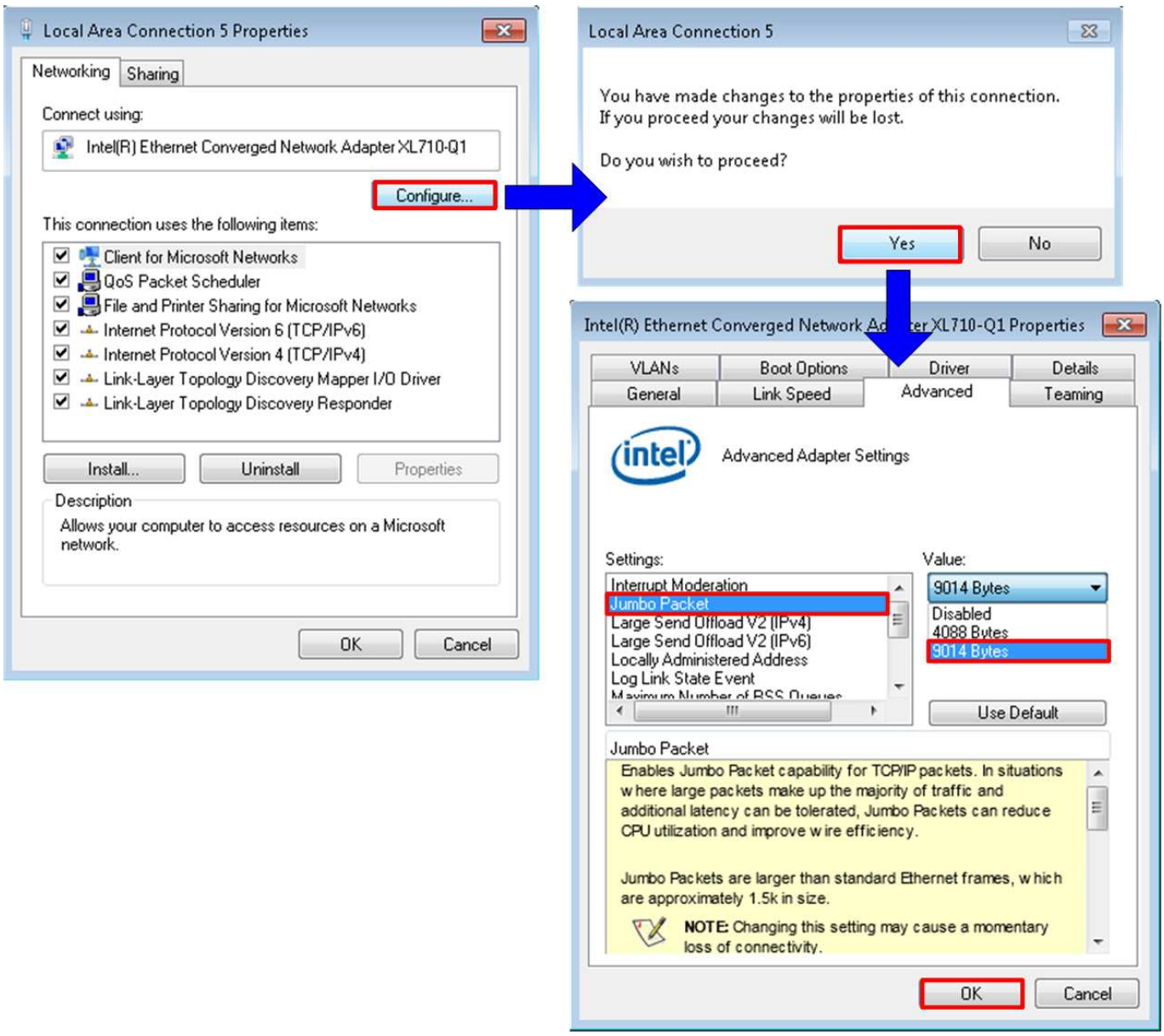

Figure 3-2 Set frame size = jumbo frame

1) On Local Area Connection Properties window, click “Configure” as shown in Figure 3-2.
2) On Advanced Tab, select “Jumbo Packet”. Set Value to “9014 Bytes” for Jumbo Frame support or set value to “Disabled” for non-Jumbo Frame support, as shown in bottom window of Figure 3-2.

3) On Advanced Tab, select “Performance Options” and click “Properties” button.

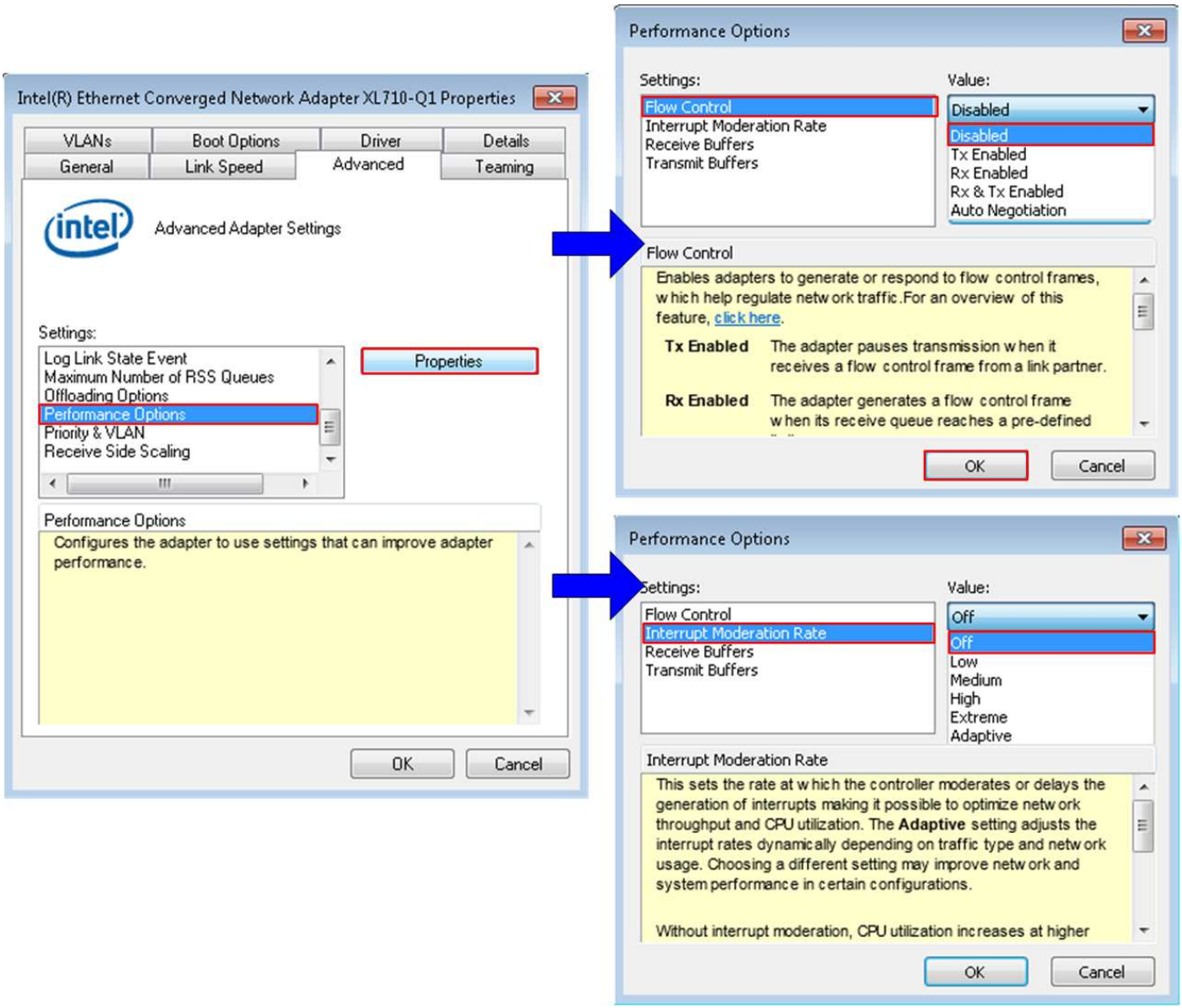

Figure 3-3 Performance option

4) Set “Flow Control” = Disabled.
5) Set “Interrupt Moderation Rate” = Off.
6) Click “OK” button to save and exit all setting windows.

3.3 Power Option Setting

1) Open Control Panel and select Power Options as shown in the left window of Figure 3-4.
2) Change setting to High Performance as shown in the right window of Figure 3-4.

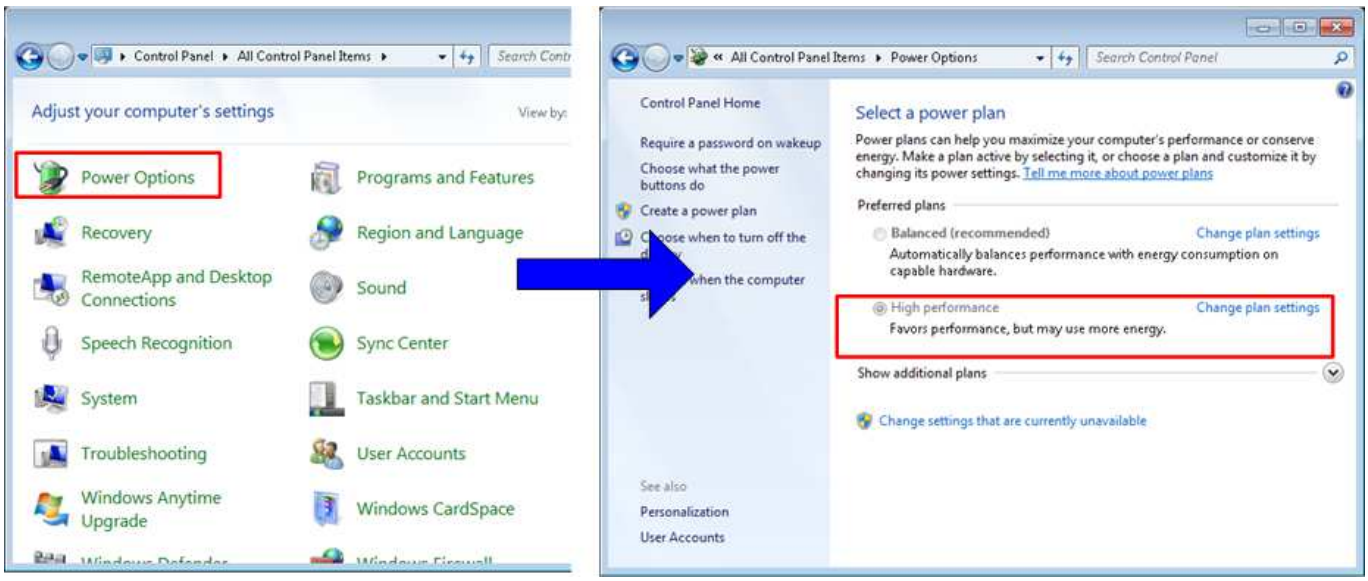

Figure 3-4 Power options

4 FPGA board setup

1) Power off system.
2) Connect micro USB cable from FPGA board to PC.
3) Connect QSFP+ AOC for network connection between FPGA board and PC
4) Power on system.

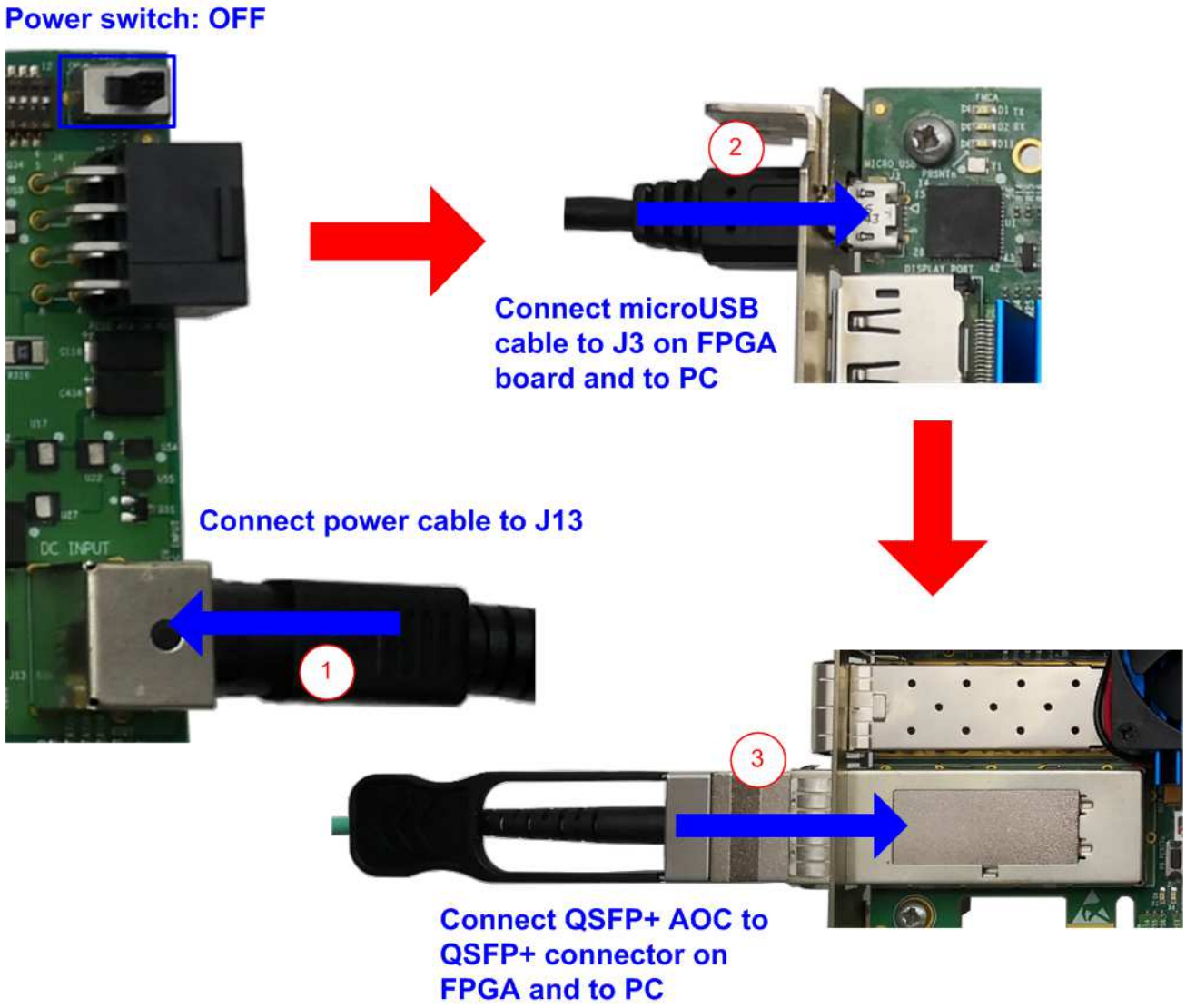

Figure 4-1 Power cable, microUSB cable, and QSFP+ cable

5) Open QuartusII Programmer to program FPGA through USB-1 by following step.
   a) Click “Hardware Setup…” to select USB-BlasterII[USB-1].
   b) Click “Auto Detect” and select FPGA (10AS066N3).
   c) Select A10 device icon
   d) Click “Change File” button, select SOF file in pop-up window, and click “open” button
   e) Check “program”
   f) Click “Start” button to program FPGA
   g) Wait until Progress status is equal to 100%

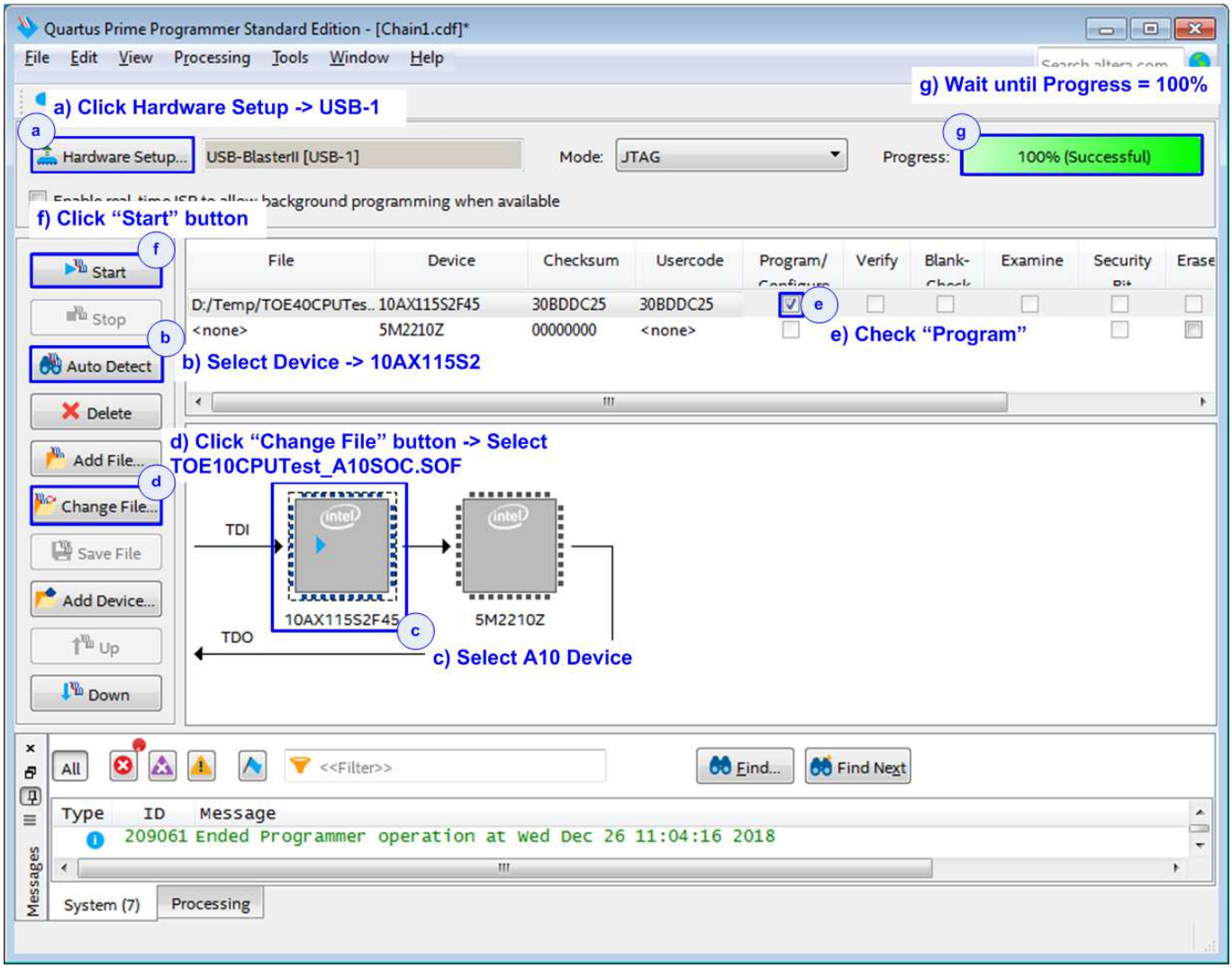

Figure 4-2 FPGA Programmer

6) Open NiosII command shell.
   a) Type “nios2-terminal”.
   b) Input ‘0’ to initialize TOE40G-IP in client mode (ask PC MAC address by sending ARP request).
   c) Default parameter in client mode is displayed on the console.

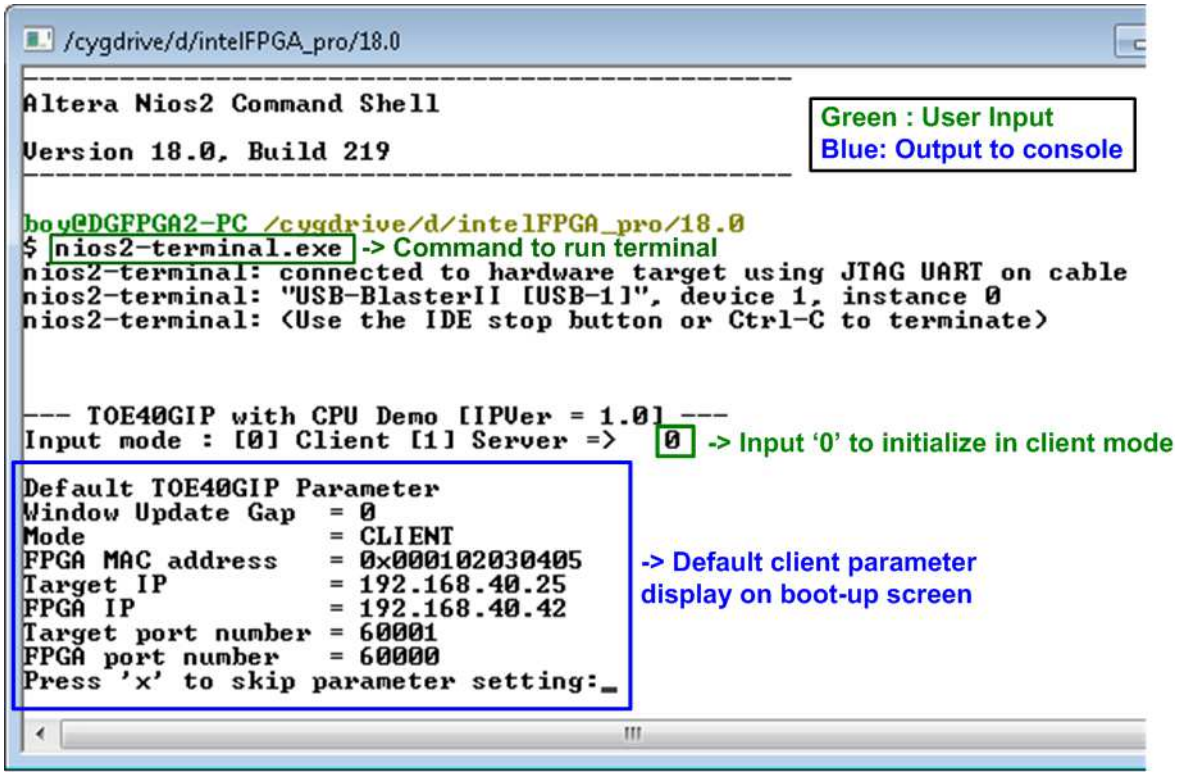

Figure 4-3 NiosII terminal after boot-up

7) User inputs ‘x’ to skip parameter setting and use default parameters for system initialization, as shown in Figure 4-4. If user inputs other keys, the menu to change parameter will be displayed. The example to change parameter is shown in topic 5.2.

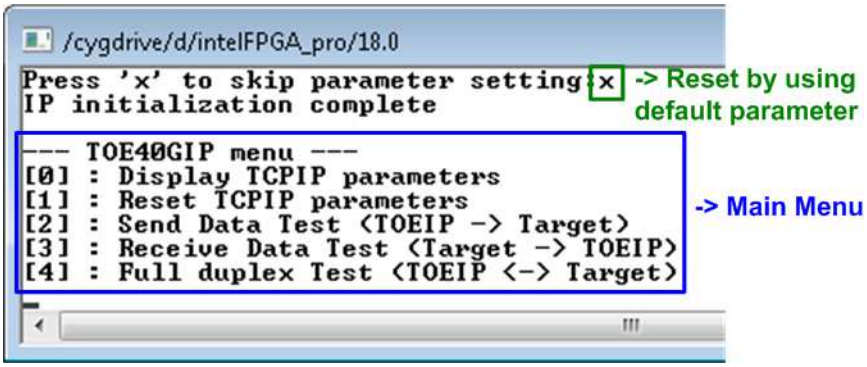

Figure 4-4 Initialization complete

Note: Transfer performance depends on Test PC resource in Test platform.

5 Main menu

5.1 Display current parameter

Select '0' to check current parameter in the demo. There are seven parameters displayed on the console.

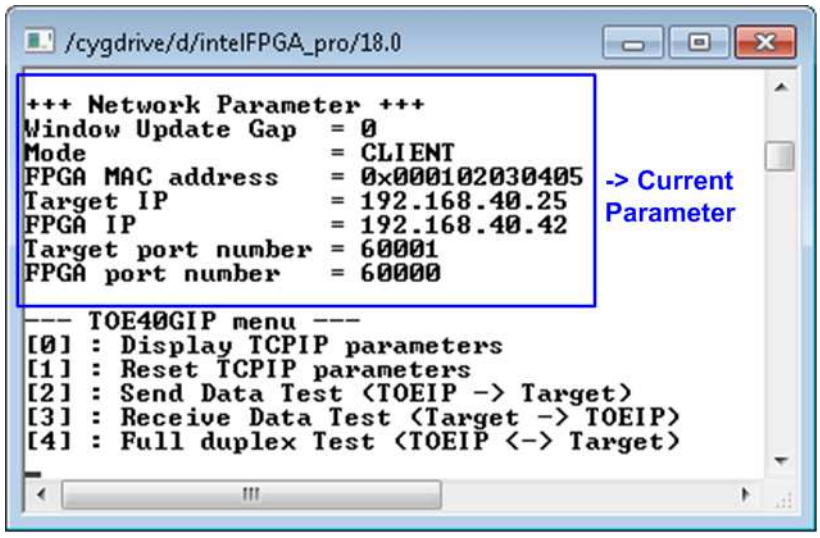

Figure 5-1 Display current parameter result

1) Window Update Gap: Set threshold value to transmit window update packet. Valid value is 0x00 – 0x3F (0-63). The unit size of threshold value is 1 Kbyte. Default value is 0 (disable window update feature).
2) Mode: Set mode to TOE40G-IP to act as server or client. To run with PC, please input '0' to initialization the IP in client mode.
3) FPGA MAC address: 48-bit hex value to be MAC address of FPGA. Default value is 0x000102030405.
4) Target IP: IP address of destination device (40 Gb Ethernet card on PC) to transfer 40 Gb Ethernet data. Default value is 192.168.40.25.
5) FPGA IP: IP address of FPGA. Default value is 192.168.40.42.
   Note: This value is used to be server IP address parameter for test application on PC.
6) Target port number: Port number of destination device to transfer 40 Gb Ethernet data. Default value is 60001.
7) FPGA port number: Port number of FPGA. Default value is 60000.
   Note: This value is used to be server port for test application on PC.

To change some parameters, user can set by using menu [1].

5.2 Reset TOE40G-IP

Select '1' to reset the IP and change IP parameters.
This menu is used to change IP parameters. After user selects this menu, the current parameters are displayed. User inputs 'x' to use same parameters or inputs other keys to change parameters. Seven parameters are designed to change from user. If user inputs invalid value such as 'n', the parameter will not change. After setting parameters completely, IP is reset. More details of each parameter are described as follows.

1) Window Update Gap: Set threshold value to transmit window update packet. Valid value is 0x00 – 0x3F (0-63). The unit size of threshold value is 1 Kbyte. Default value is 0 (disable window update feature).
2) Mode: Input '0' to initialization the IP as client mode.
3) FPGA MAC address: Input 12-digit of hex value. Add "0x" as a prefix to input as hex value.
4) FPGA IP address: A set of four decimal digits is separated by ".". The valid range of each decimal digit is 0-255.
5) FPGA port number: Valid range is 0-65535.
6) Target IP address: A set of four decimals like FPGA IP address. This value is IP address of Test PC.
7) Target port number: Valid range is 0-65535.

After complete to assign all parameters, new parameters are displayed on the console. Next, reset signal is sent to the IP to load new parameters. Finally, "IP initialization complete" is shown after IP completes initialization process, as shown in Figure 5-2.

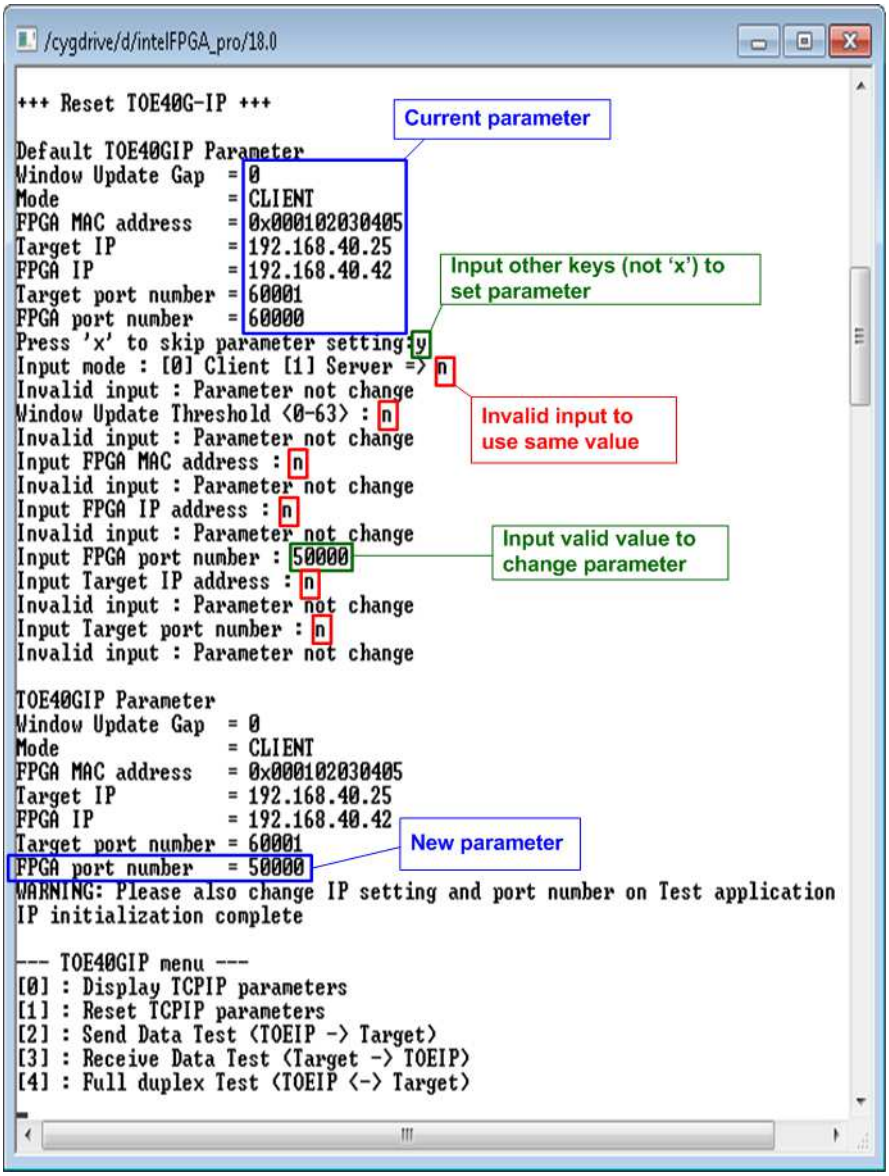

Figure 5-2 Change IP parameter result

5.3 Send Data Test

To transfer data from FPGA to PC, select '2' to run send data test on FPGA and prepare "tcpdatatest.exe" on PC to receive data. User inputs test parameters on FPGA through NiosII terminal and inputs test parameters on PC through command prompt. The sequence to run the test is shown as below.

1) On NiosII terminal, input three parameters in send data test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   b) Input packet size: Unit of packet size is byte. Valid value is 32 – 8960. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   c) Input Mode: Mode of FPGA to transfer data. Input '1' to set server mode.
2) If inputs are valid, recommended parameters to run test application on PC and "Wait Open connection …" will be displayed.
3) On Command prompt, input test parameters following the recommended value. There are six parameters for "tcpdatatest".
   >> tcpdatatest <mode> <dir> <server IP> <server port> <bytelen> <pattern>
   a) Mode: Input 'c' to run Test PC as a client.
   b) Dir: Input 'r' to run Test PC for receiving test data from FPGA
   c) Server IP: Input same value as IP address of FPGA
   d) Server port: Input same value as port number of FPGA
   e) Bytelen: Input same value as "Input transfer size" of step 1a)
   f) Pattern: Input '1' to verify data from FPGA
4) After running test application, the port is created. Current transfer size is displayed on NiosII terminal and Command prompt every second. "Send data complete" is displayed on NiosII terminal after all data are sent.
5) FPGA closes the connection. Finally, total transfer size and performance are displayed on NiosII terminal and Command prompt.

Figure 5-3 shows the example of send data test when using non-jumbo frame size. Left window is NiosII terminal on FPGA operating as server and right window is Command prompt on PC operating as client.

Figure 5-4 shows the example of send data test when using jumbo frame size.

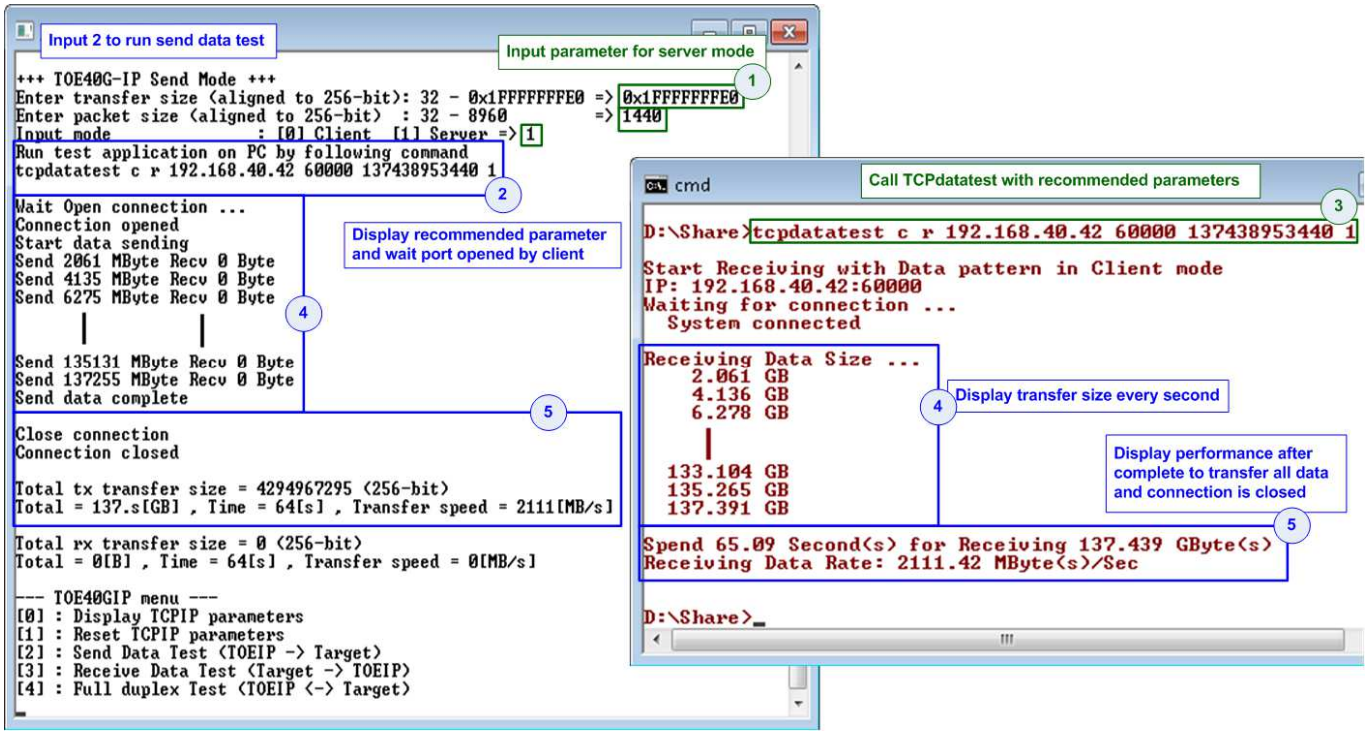

Figure 5-3 Send data test by using non-jumbo frame

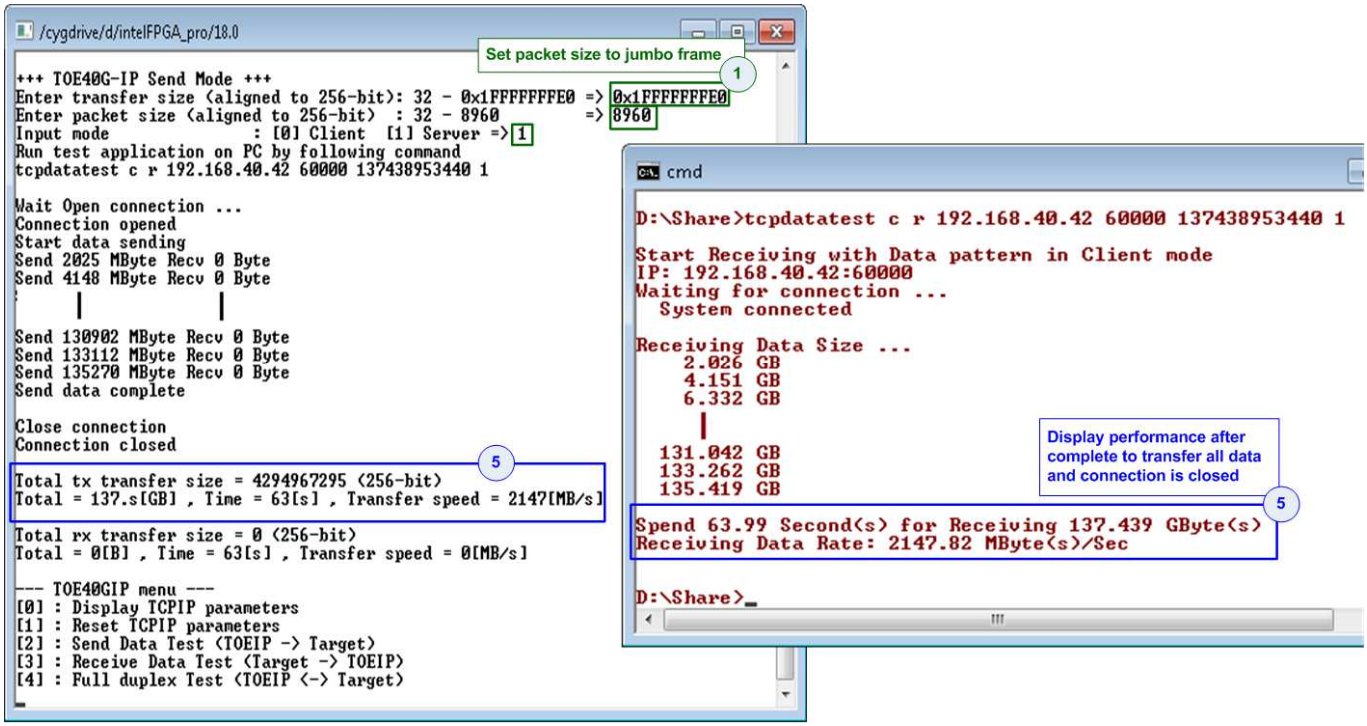

Figure 5-4 Send data test by using jumbo frame

If the input is invalid, “Out-of-range input”/”Invalid input” will be displayed and the operation will be cancelled, as shown in Figure 5-5 - Figure 5-7.

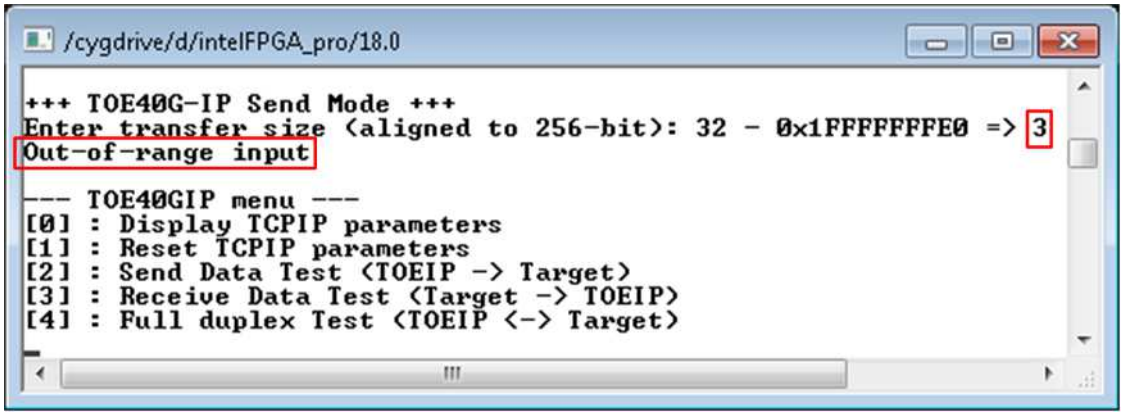

Figure 5-5 Error from invalid transfer size

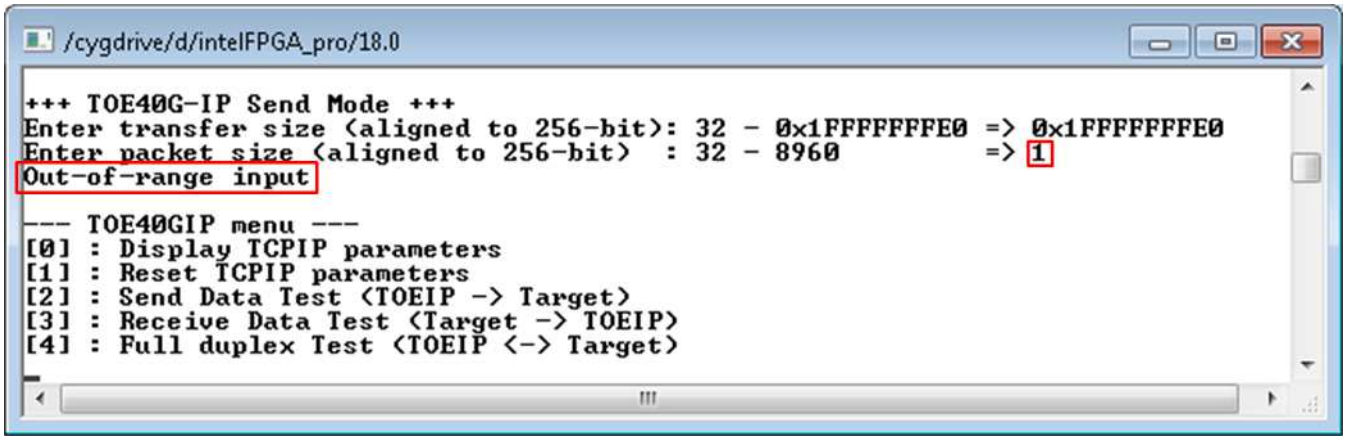

Figure 5-6 Error from invalid packet size

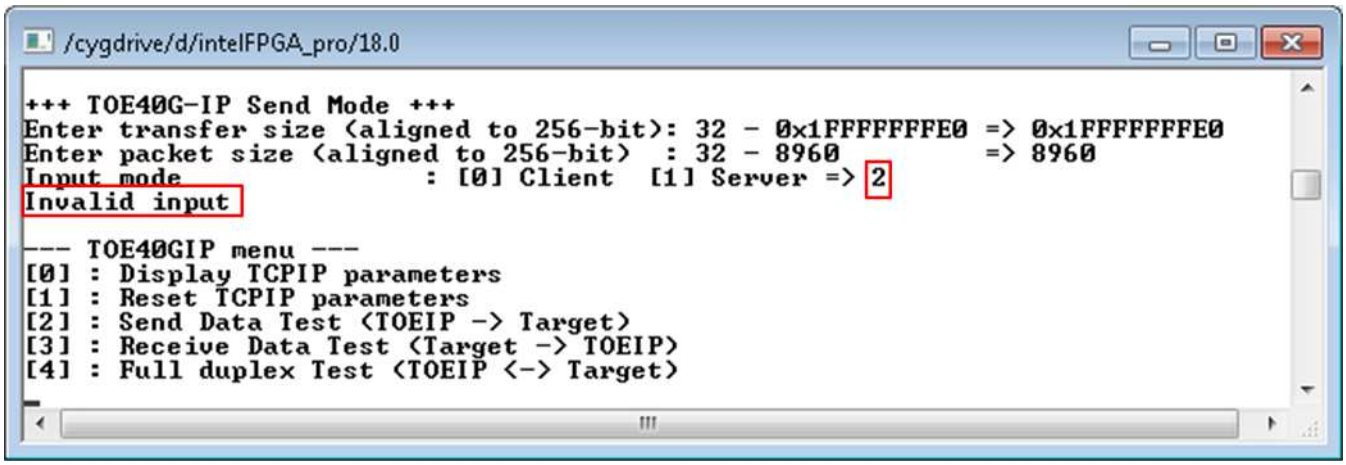

Figure 5-7 Error from invalid mode

5.4 Receive Data Test

To transfer data from PC to FPGA, select '3' to run receive data test on FPGA and prepare "tcpdatatest.exe" on PC to send data. User inputs test parameters on FPGA through NiosII terminal and inputs test parameters on PC through Command prompt. The sequence to run the test is shown as below.

1) On NiosII terminal, input three parameters in receive data test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   b) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from PC.
   c) Input Mode: Mode of FPGA to transfer data. Input '1' to set server mode.
2) If inputs are valid, recommended parameters to run test application on PC and "Wait Open connection …" will be displayed.
3) On Command prompt, input test parameters following the recommended value. There are six parameters for "tcpdatatest".
   >> tcpdatatest <mode> <dir> <server IP> <server port> <bytelen> <pattern>
   a) Mode: Input 'c' to run Test PC as a client.
   b) Dir: Input 't' to run Test PC for sending test data to FPGA
   c) Server IP: Input same value as IP address of FPGA
   d) Server port: Input same value as port number of FPGA
   e) Bytelen: Input same value as "Input transfer size" of step 1a)
   f) Pattern: Input same value as "Input data verification mode" of step 1b). Select '0' to send dummy data or '1' to send increment data.
4) After running test application, the port is created. Current transfer size is displayed on NiosII terminal and Command prompt every second.
5) "Connection closed" and "Receive data completed" are displayed on NiosII terminal after PC completes to send all data and closes the connection. Finally, total transfer size and performance are displayed on NiosII terminal and Command prompt.

Figure 5-8 shows the example of receive data test when disable data verification mode on FPGA and send dummy data on PC. The left window is test result on NiosII terminal while the right window is test result on Command prompt.

Figure 5-9 shows the example of receive data test when enable data verification mode on FPGA and send increment data on PC. The performance when disable data verification is better than the performance when enable data verification because generating dummy data uses less PC resource than increment data.

Figure 5-10 shows the example of error when data verification is failed. In the example, the error is caused from mismatch verification mode value. FPGA enables data verification while "tcpdatatest" sends dummy data. The error message is displayed on NiosII terminal.

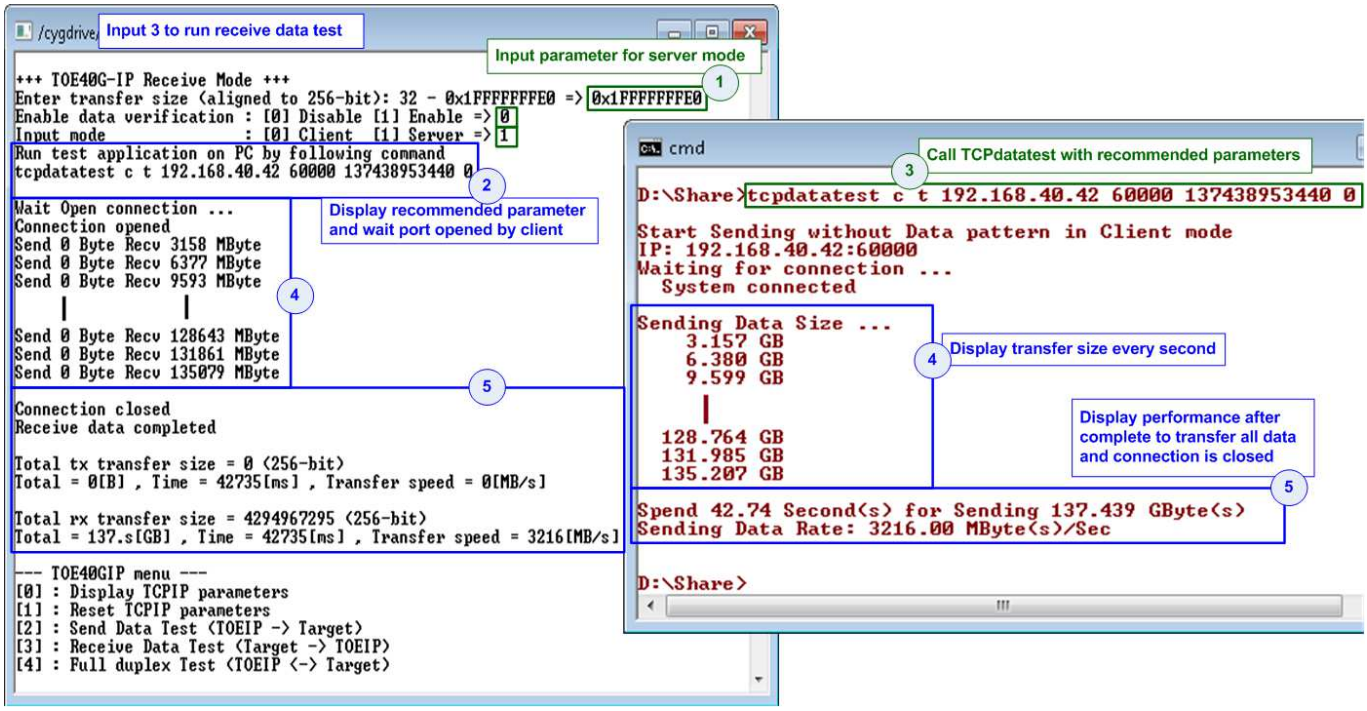

Figure 5-8 Receive data test without data verification

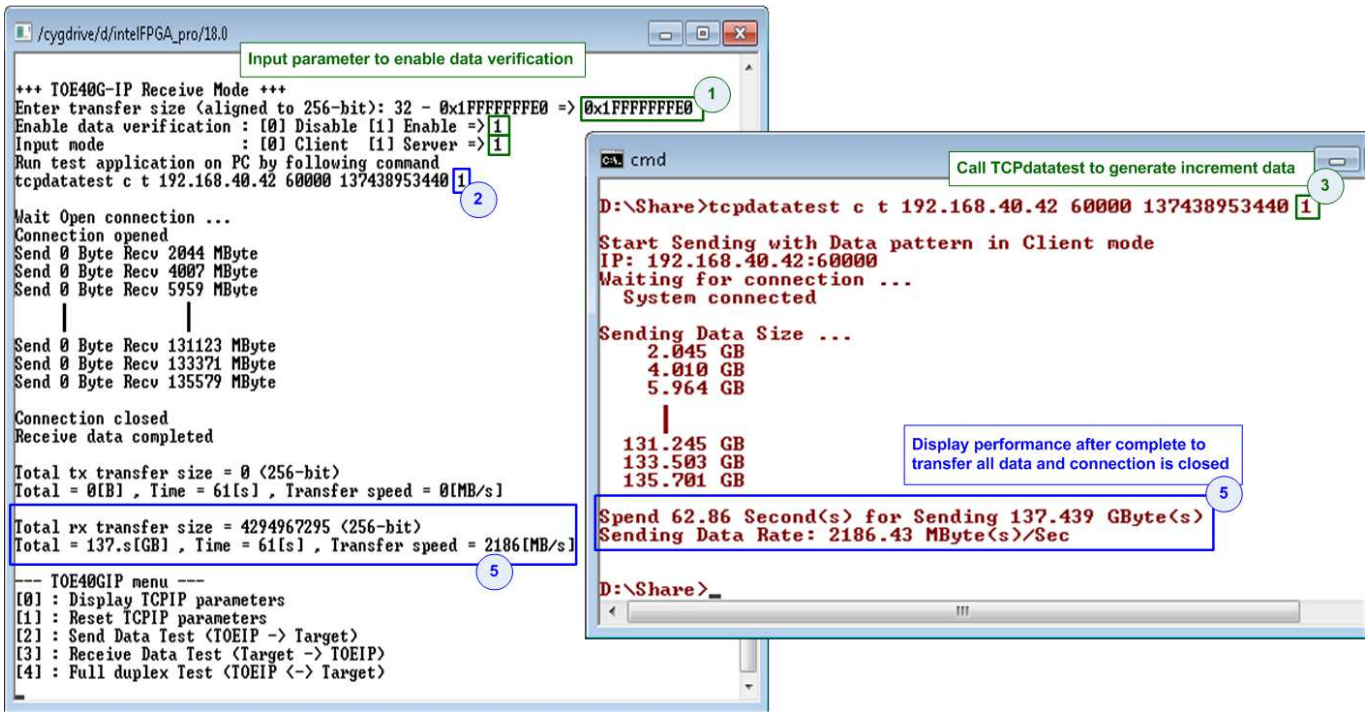

Figure 5-9 Receive data test when enable data verification

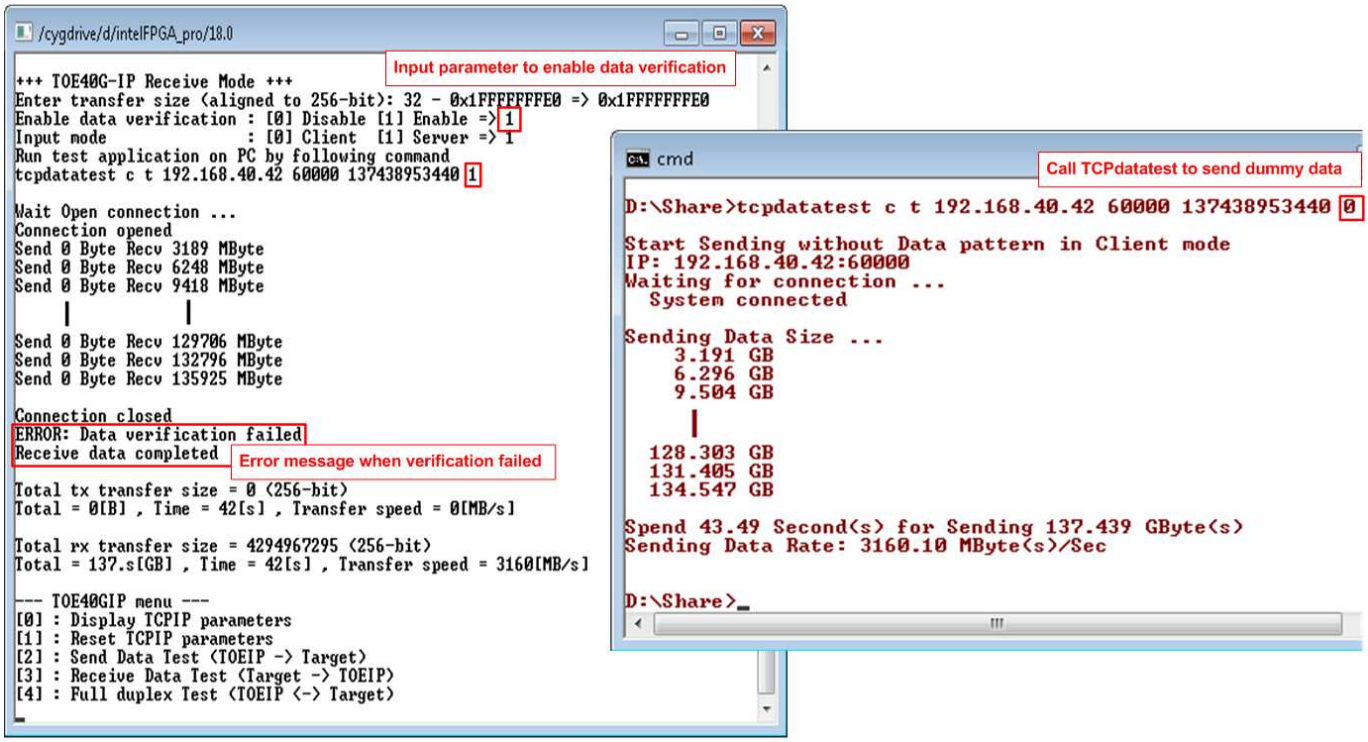

Figure 5-10 Receive data test when data verification is failed

5.5 Full duplex Test

Select '4' to run full duplex test to transfer data between FPGA and PC in both directions at the same time. User inputs test parameters on FPGA through NiosII terminal and inputs test parameters on PC through Command prompt. The sequence to run the test is shown as below.

1) On NiosII terminal, input four parameters in full duplex test.
   a) Input transfer size: Unit of transfer size is byte. Input "0x7FFFFFC0" which is equal to total transfer size setting on test application.
   b) Input packet size: Unit of packet size is byte. Valid value is 32 – 8960. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   c) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from PC.
   d) Input Mode: Mode of FPGA to transfer data. Input '1' to set server mode.
2) If inputs are valid, recommended parameters to run test application on PC and "Wait Open connection …" will be displayed.
3) On Command prompt, input test parameters following the recommended value. There are three parameters for "tcp_client_txrx_40G".
   >> tcp_client_txrx_40G <server IP> <server port> <pattern>
   a) Server IP: Input same value as IP address of FPGA
   b) Server port: Input same value as port number of FPGA
   c) Pattern: Input same value as "Input data verification mode" of step 1b). Select '0' to send dummy data or '1' to send increment data.
4) After running test application, the port is created. Current transfer size is displayed on NiosII terminal and Command prompt every second.
5) "Send data complete" is displayed on NiosII terminal after it completes to send and receive all data. "Connection closed" is displayed after connection is closed. Finally, total transfer size and performance are displayed on NiosII terminal and Command prompt.
   Step 4) – 5) run in forever loop until user cancels the operation. To cancel the operation, input "Ctrl+C" on Command prompt, and then input any keys on NiosII terminal.

As shown in Figure 5-11 - Figure 5-12, transfer performance when running full duplex without data verification shows the better performance than the test with data verification. The left window is the test result on NiosII terminal while the right window is the test result on Command prompt.

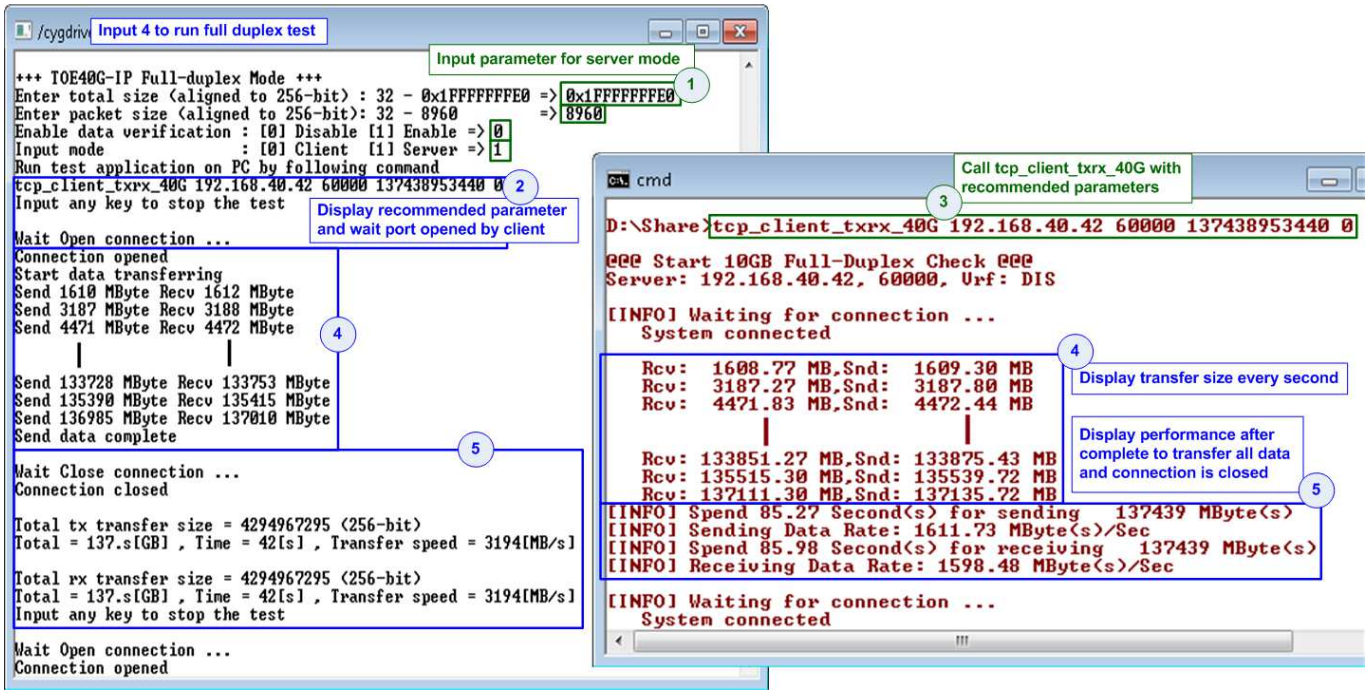

Figure 5-11 Full duplex test without data verification

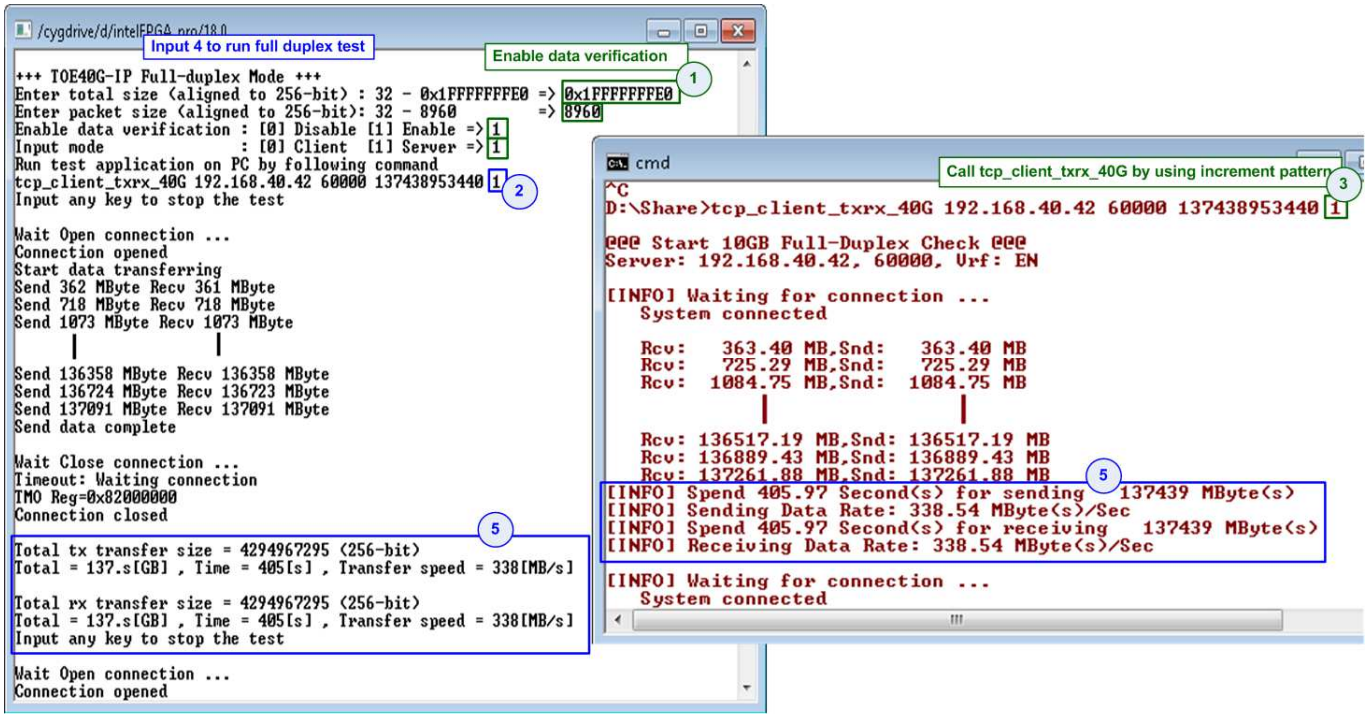

Figure 5-12 Full duplex test with data verification

Part B TOE40G-IP demo by using two FPGAs

6 Environment Setup

As shown in Figure 6-1, to run TOE40G-IP demo by using two FPGAs, please prepare following test environment.

1) Two FPGA development boards (Arria10 GX development kit)
2) QSFP+ AOC for 40G network connection between two between FPGA boards
3) Connect micro USB cable for programming FPGA and NiosII terminal between each FPGA board and PC (two cables are used for two FPGA boards)
4) QuartusII programmer for programming FPGA and NiosII command shell, installed on PC

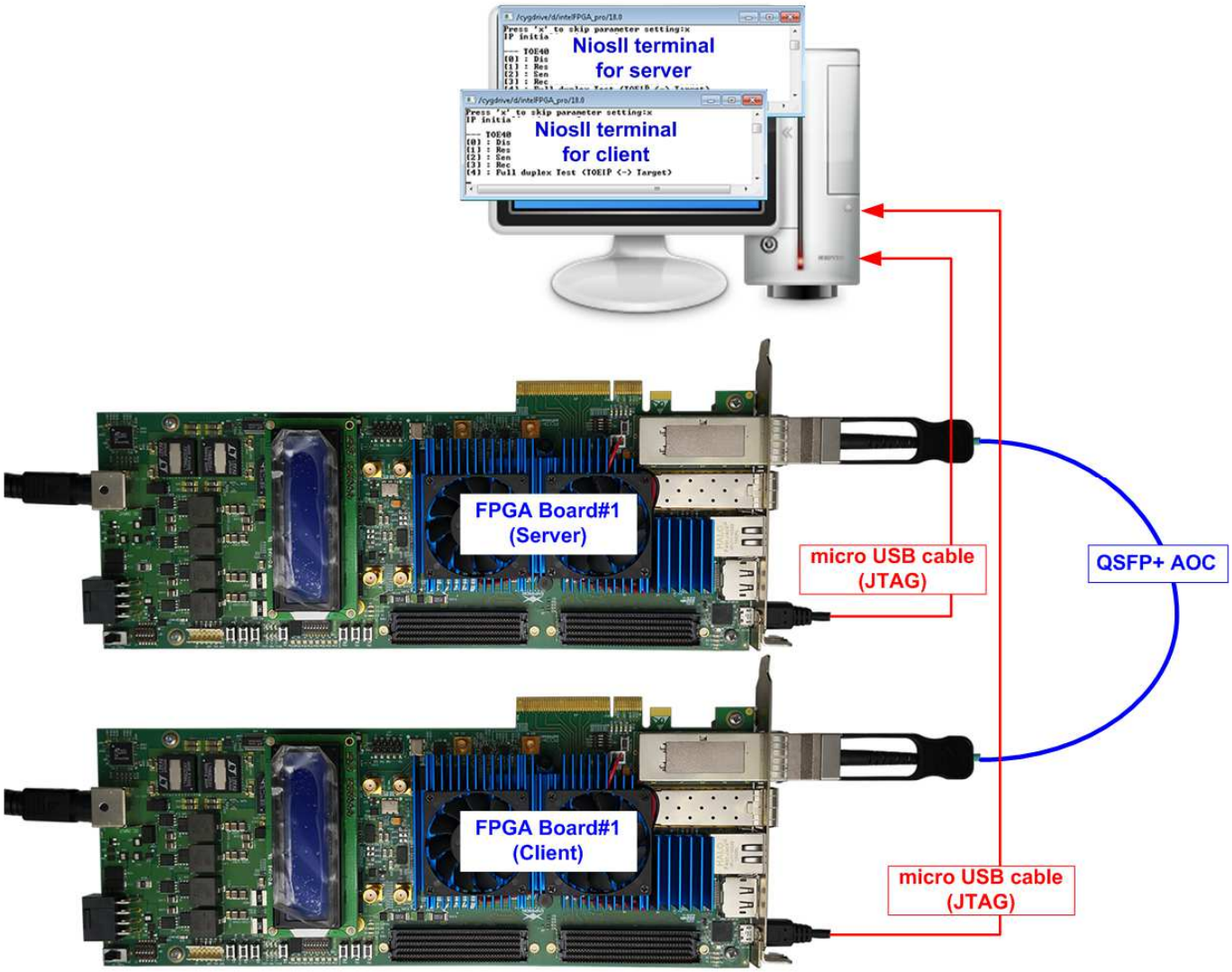

Figure 6-1 TOE40G-IP with CPU demo (FPGA<->FPGA) by Arria10GX board

7 FPGA board setup

Please follow step1) – 4) of topic 4 FPGA board setup to prepare two FPGA boards and QSFP+ AOC connections for running the demo. The next steps are described as follows.

1) When connect two FPGA boards to same PCs, two USB-Blasters are found on USB-1 and USB-2.
2) Select hardware setup to USB-1 and configure FPGA#1 following Figure 4-2.

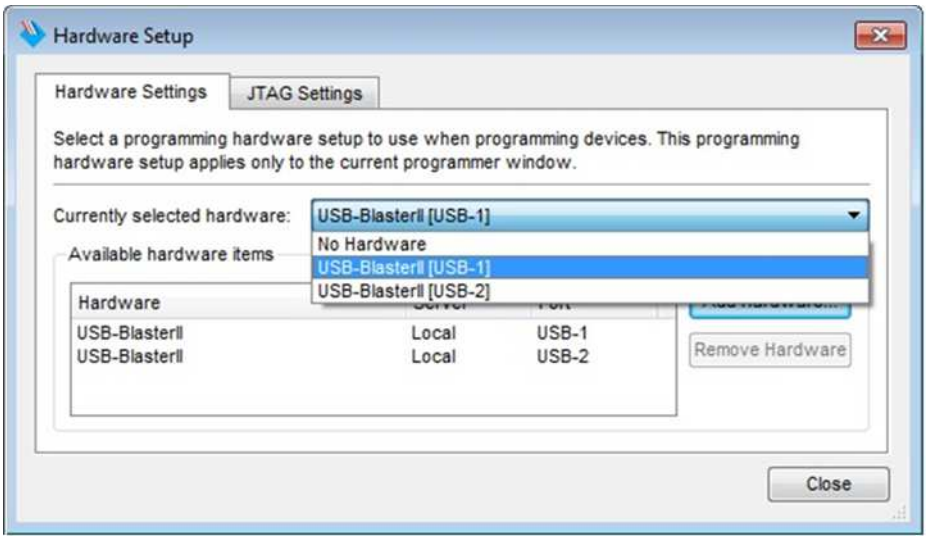

Figure 7-1 Configure FPGA#1 through USB-1

3) Change hardware setup to USB-2 and configure FPGA#2 following Figure 4-2.

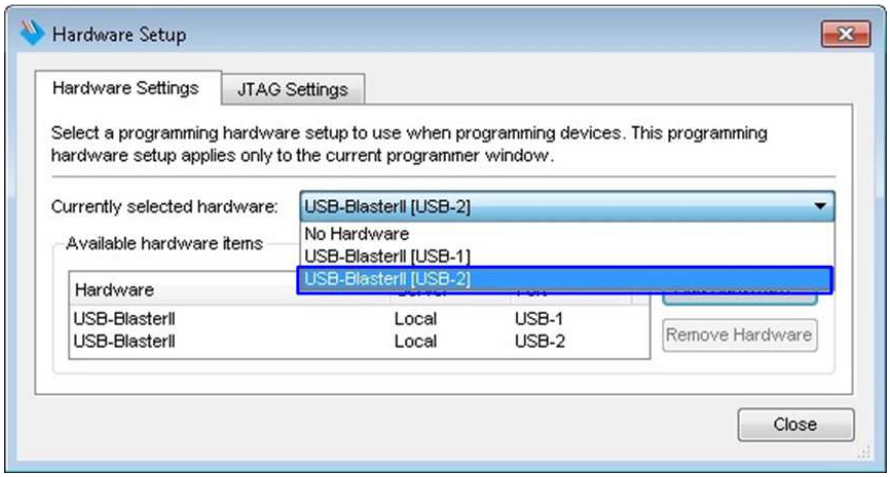

Figure 7-2 Configure FPGA#2 through USB-2

4) Open NiosII terminal for board#1 and board#2.
   a. Type “nios2-terminal -- cable 1” to open NiosII terminal of FPGA board#1 (USB-1 port)
   b. Type “nios2-terminal -- cable 2” to open NiosII terminal of FPGA board#2 (USB-2 port)
   c. Set ‘1’ on NiosII terminal of FPGA board#1 for running server mode. Set ‘0’ on NiosII terminal of FPGA board#2 for running client mode.

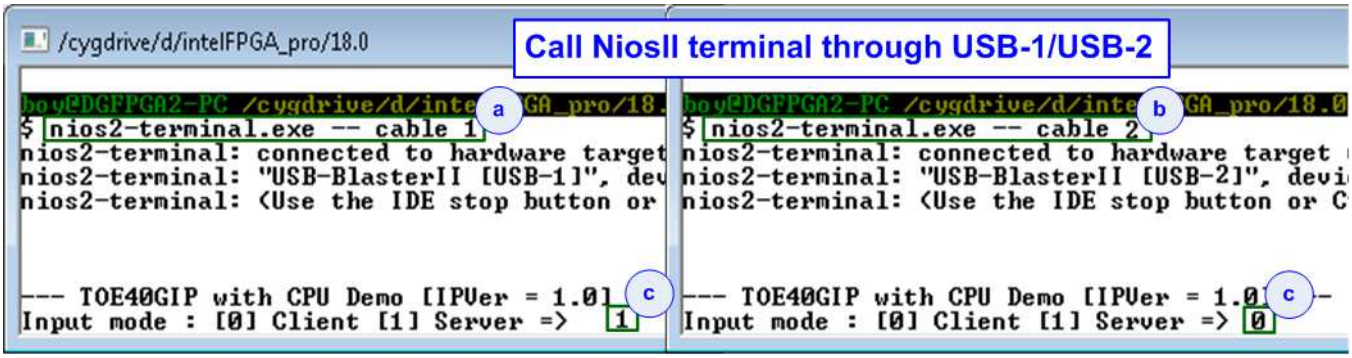

Figure 7-3 NiosII terminal setup

5) Default parameters for server or client are displayed on the console, as shown in Figure 7-4.

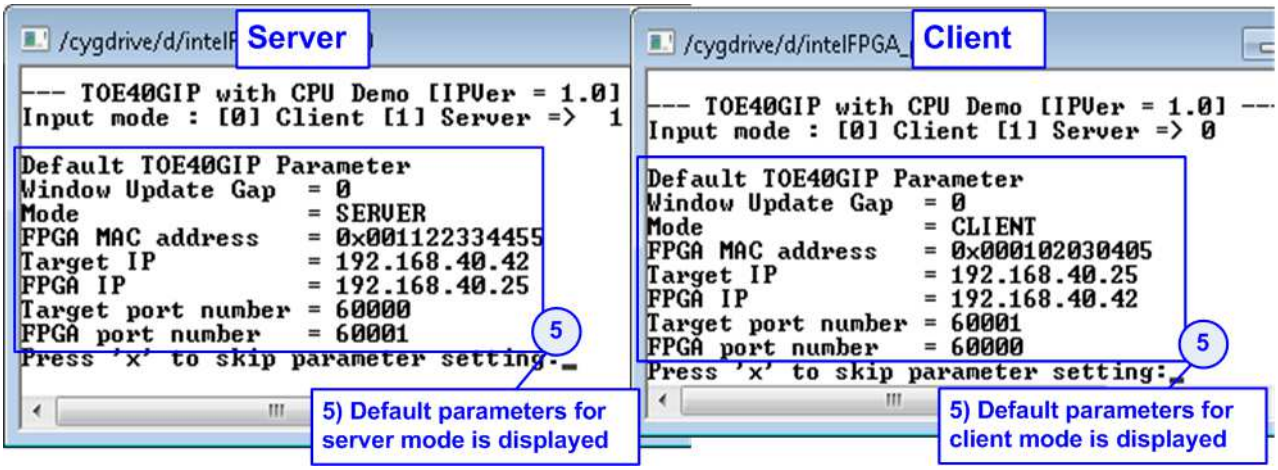

Figure 7-4 Default parameters of Server/Client mode

6) Input 'x' to use default parameters or other keys to change parameters. Please complete to set parameters on server NiosII terminal before client NiosII terminal. Server must be reset to start IP initialization before client.

   After finish parameters setting, IP starts initialization process. "IP initialization complete" is displayed when all initialization sequence are completed. Finally, main menu is displayed on NiosII terminal.

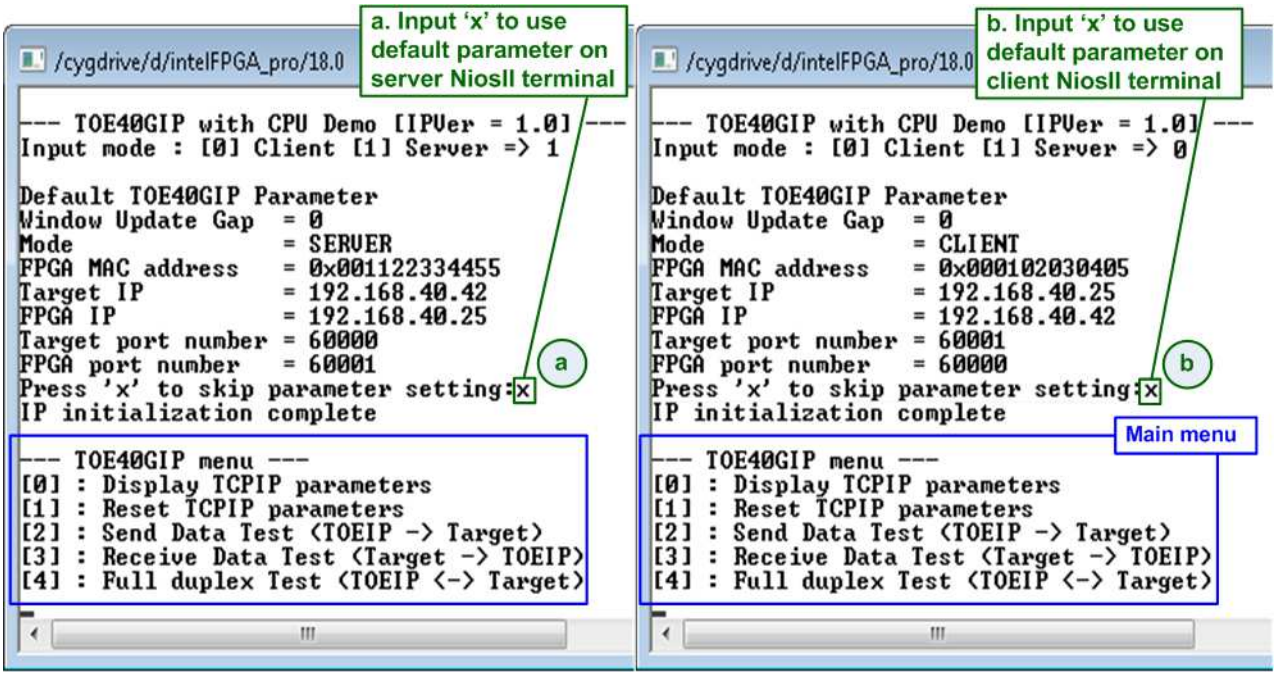

Figure 7-5 Main menu

8 Main menu

8.1 Display current parameter

Select '0' to check current parameter in the demo. There are seven parameters displayed on NiosII terminal.

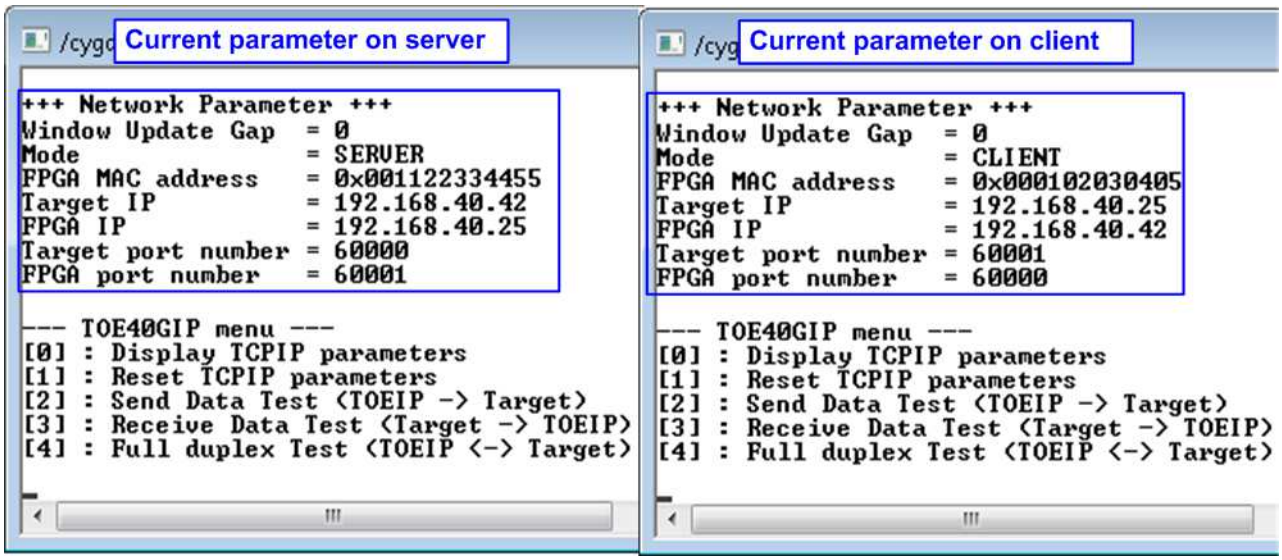

Figure 8-1 Display current parameter result

1) Window Update Gap: Set threshold value to transmit window update packet. Valid value is 0x00 – 0x3F (0-63). The unit size of threshold value is 1 Kbyte. Default value is 0 (disable window update feature).
2) Mode: Set mode to TOE40G-IP to act as server or client. Input '0' for client and '1' for server.
3) FPGA MAC address: 48-bit hex value to be MAC address of FPGA. Default value is 0x000102030405 (client mode) or 0x001122334455 (server mode).
4) Target IP: IP address of destination device to transfer 40 Gb Ethernet data. Default value is 192.168.40.25 (client mode) or 192.168.40.42 (server mode).
5) FPGA IP: IP address of FPGA. Default value is 192.168.40.42 (client mode) or 192.168.40.25 (server mode).
6) Target port number: Port number of the destination device to transfer 40 Gb Ethernet data. Default value is 60001 (client mode) or 60000 (server mode).
7) FPGA port number: Port number of FPGA. Default value is 60000 (client mode) or 60001 (server mode).

To change some parameters, user can set by using menu [1].

8.2 Reset TOE40G-IP

Select '1' to reset the IP and change IP parameters.
This menu is used to change IP parameters. After user selects this menu, the current parameters are displayed. User inputs 'x' to use same parameters or inputs other keys to change parameters. Seven parameters are designed to set by user. If user inputs invalid value such as 'n', the parameter will not change. After setting parameters completely, IP is reset. More details of each parameter are described as follows.

Note:
1. *When user desires to reset parameters on server, the client FPGA must be also reset. Server must be reset firstly to wait ARP request sending from client.*
2. *Parameter of client and server must be matched, i.e.*
   a. *Target IP of server = FPGA IP of client*
   b. *FPGA IP of server = Target IP of client*
   c. *Target port number of server = FPGA port number of client*
   d. *FPGA port number of server = Target port number of client*

1) Window Update Gap: Set threshold value to transmit window update packet. Valid value is 0x00 – 0x3F (0-63). The unit size of threshold value is 1 Kbyte. Default value is 0 (disable window update feature).
2) Mode: Input '0' (client) or '1' (server) to determine FPGA initialization mode. The valid range is 0-1. It needs to set different mode for two FPGA boards. One board is client and another board is server.
3) FPGA MAC address: Input 12-digit of hex value. Add "0x" as a prefix to input as hex value.
4) FPGA IP address: A set of four decimal digits is separated by ".". The valid range of decimal digit is 0-255.
5) FPGA port number: Valid range is 0-65535.
6) Target IP address: A set of four decimals like FPGA IP address.
7) Target port number: Valid range is 0-65535.

After complete to assign all parameters, new parameters are displayed on NiosII terminal. Next, reset signal is sent to the IP to load new parameters. Finally, "IP initialization complete" is shown after IP completes initialization process, as shown in Figure 8-2.

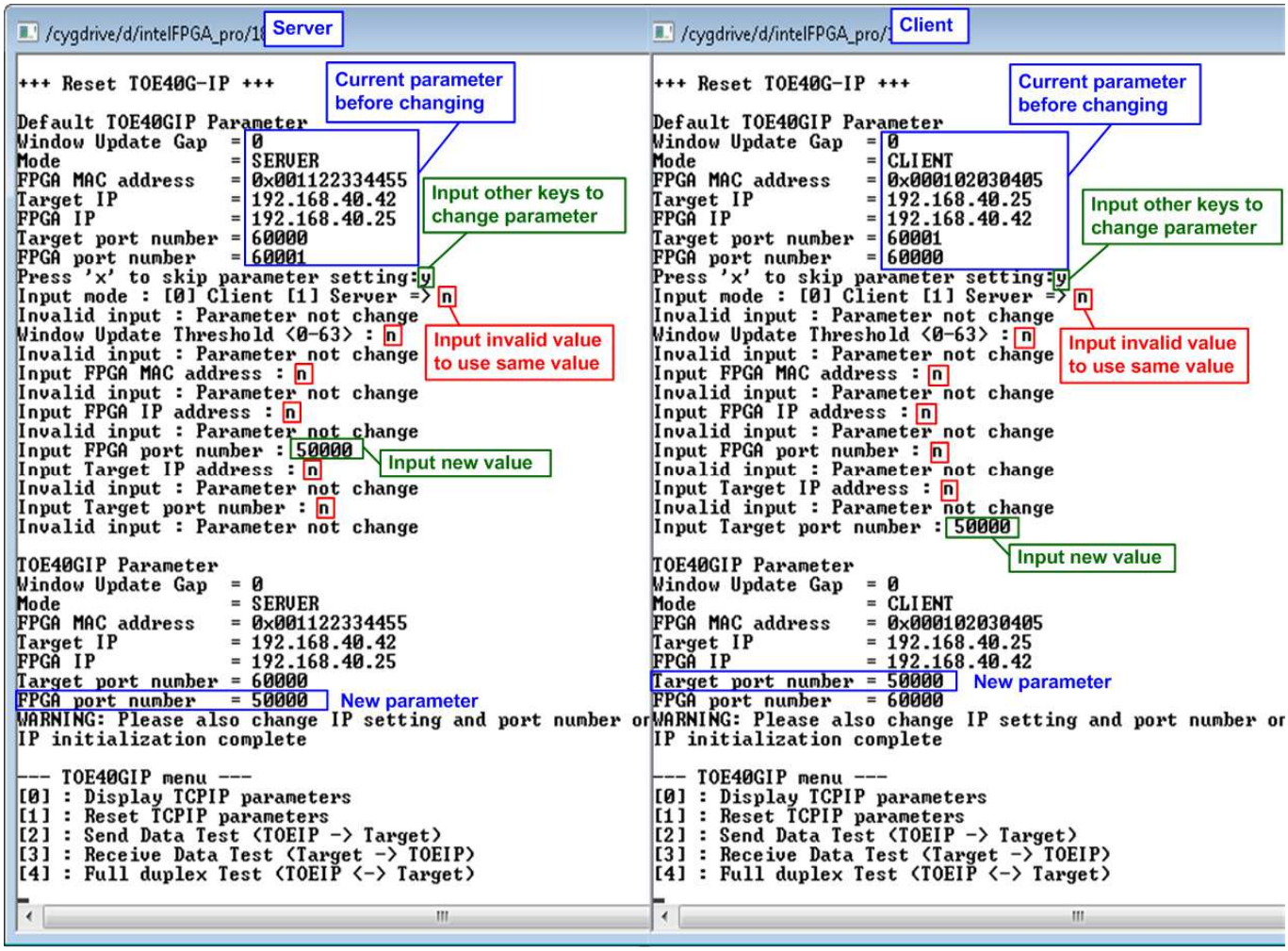

Figure 8-2 Change IP parameter result

8.3 Send Data Test (server to client)

To transfer data from server to client, select '2' to run send data test on server FPGA and select '3' to run receive data test on client FPGA. User inputs test parameters through NiosII terminal. The sequence to run the test is shown as below.

1) On NiosII terminal of server, input three parameters in send data test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   b) Input packet size: Unit of packet size is byte. Valid value is 32 – 8960. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   c) Input Mode: Mode of FPGA to transfer data. Input '1' to set server mode.
2) If inputs are valid, "Wait Open connection …" will be displayed.
3) On NiosII terminal of client, input three test parameters in receive data test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix. This value must be equal to transfer size on server FPGA.
   b) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification to verify data from FPGA running as server.
   c) Input Mode: Mode of FPGA to transfer data. Input '0' to set client mode.

   If inputs are valid, the operation will start.
4) After running client, current transfer size is displayed on both NiosII terminals every second. "Send data complete" is displayed on NiosII terminal of server after it sends all data.
5) Server closes the connection. Total transfer size and performance are displayed on both NiosII terminals.

Figure 8-3 shows the example of send data test when using non-jumbo frame size. Left window is NiosII terminal from FPGA operating as server and right window is NiosII terminal from FPGA operating as client.

Figure 8-4 shows the example of send data test when using jumbo frame size. When using jumbo frame size, performance is better than non-jumbo frame.

If the input is invalid, "Out-of-range input"/"Invalid input" will be displayed and the operation will be cancelled, as shown in Figure 5-5 - Figure 5-7 (same as FPGA<->PC test).

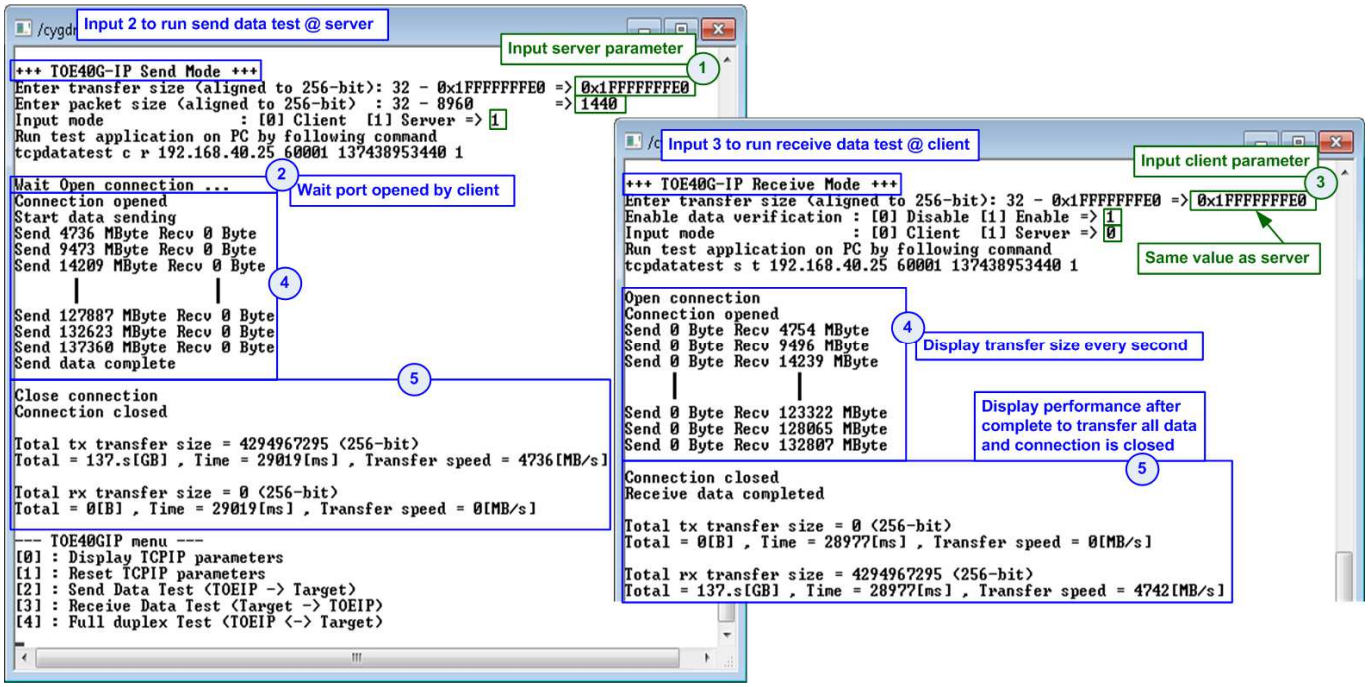

Figure 8-3 Send data test by using non-jumbo frame

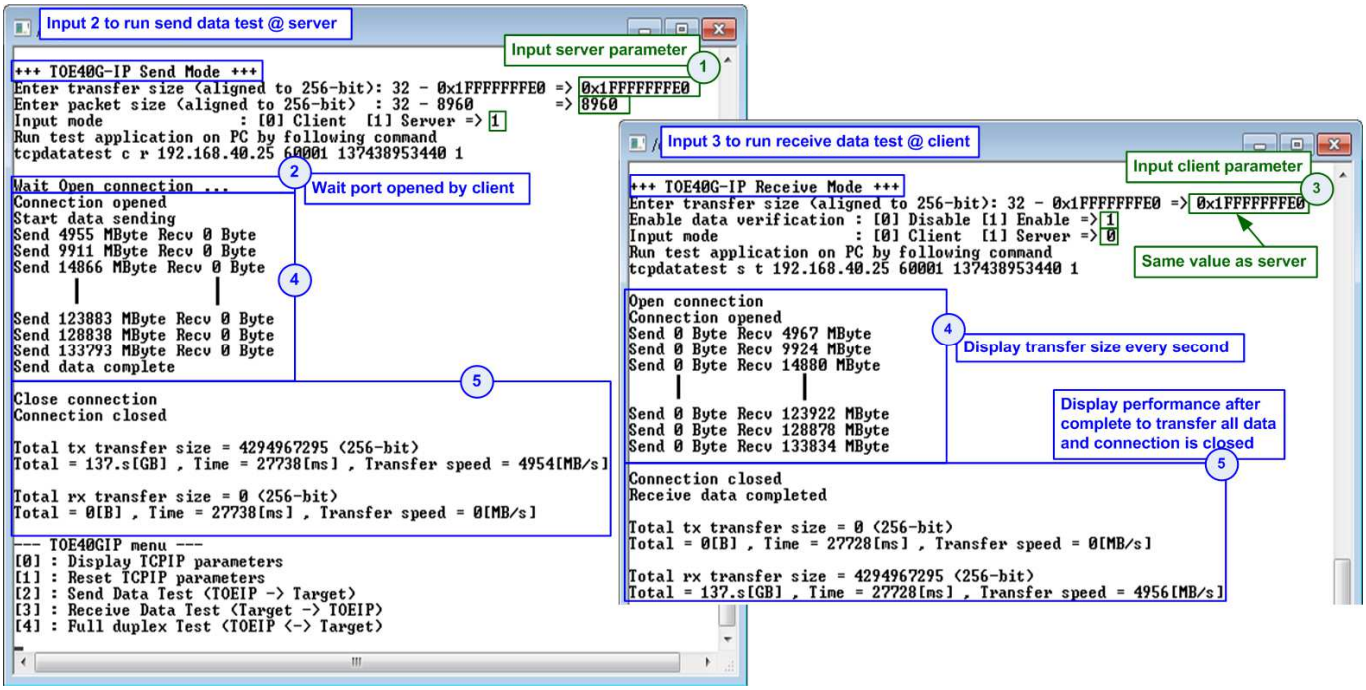

Figure 8-4 Send data test by using jumbo frame

8.4 Receive Data Test (client to server)

To transfer data from client to server, select '3' to run receive data test on server FPGA and select '2' to run send data test on client FPGA. User inputs test parameters through NiosII terminal. The sequence to run the test is shown as below.

1) On NiosII terminal of server, input three parameters in receive data test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32. User selects to input as or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   b) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification to verify data from FPGA running as client.
   c) Input Mode: Mode of FPGA to transfer data. Input '1' to set server mode.
2) If inputs are valid, "Wait Open connection …" will be displayed.
3) On NiosII terminal of client, input three test parameters in send data test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix. This value must be equal to transfer size on server FPGA.
   b) Input packet size: Unit of packet size is byte. Valid value is 32 – 8960. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   c) Input Mode: Mode of FPGA to transfer data. Input '0' to set client mode.

   If inputs are valid, the operation will start.
4) After running client, current transfer size is displayed on both NiosII terminals every second.
5) "Connection closed" and "Received data completed" are displayed on NiosII terminal of server after client completes to send all data and closes the connection. Total transfer size and performance are displayed on both NiosII terminals.

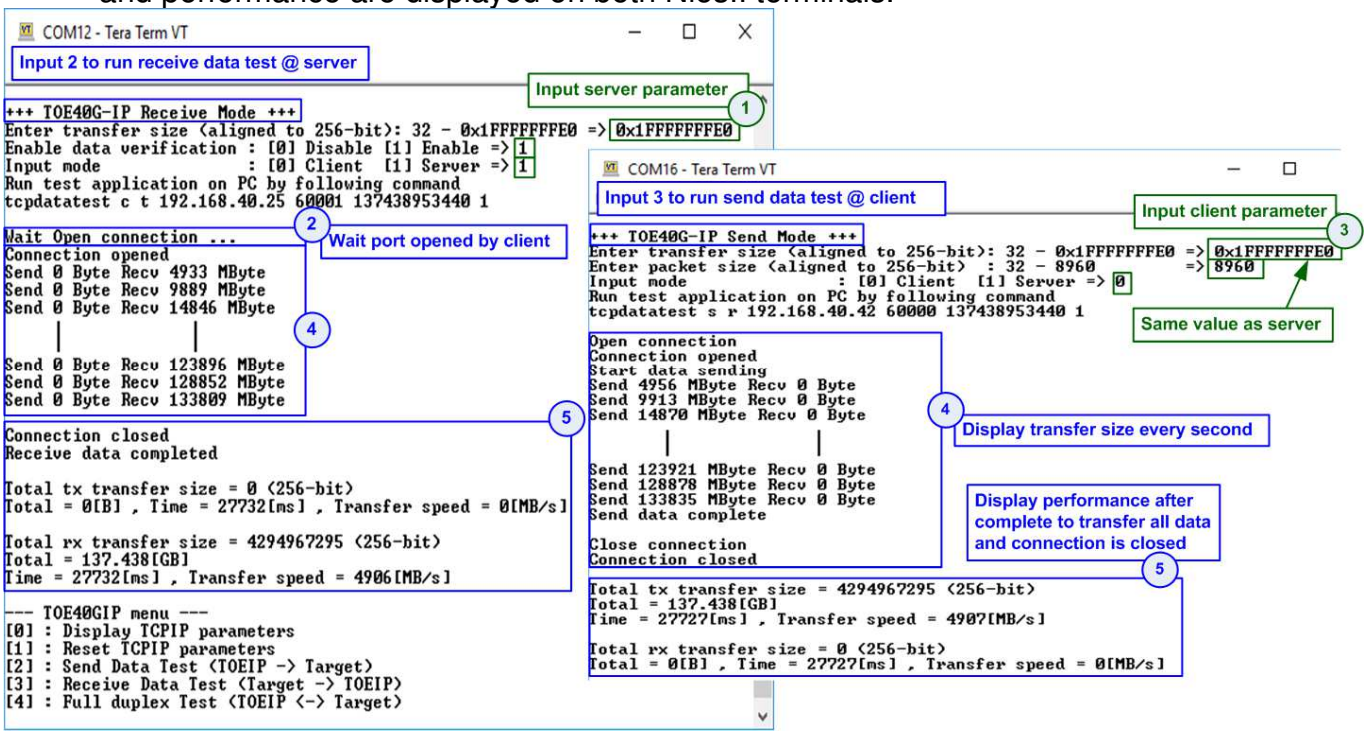

Figure 8-5 Receive data test with data verification

8.5 Full duplex Test

Select '4' to run full duplex test on server FPGA and client FPGA to transfer data in both directions at the same time. User inputs test parameters through NiosII terminal. The sequence to run the test is shown as below.

1) On NiosII terminal of server, input four parameters in full duplex test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 0x1F_FFFF_FFE0. The input must be aligned to 32 and must match with transfer size in FPGA running as client. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   b) Input packet size: Unit of packet size is byte. Valid value is 32 – 8960. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   c) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification to verify data from FPGA running as client.
   d) Input Mode: Mode of FPGA to transfer data. Input '1' to set server mode.
2) If inputs are valid, "Wait open connection…" will be displayed.
3) On NiosII terminal of client, input four test parameters in full duplex test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 32 - 1F_FFFF_FFE0. The input must be aligned to 32 and must match with transfer size in FPGA running as server. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   b) Input packet size: Unit of packet size is byte. Valid value is 32 – 8960. The input must be aligned to 32. User selects to input as decimal or hexadecimal unit. To input as hex value, user adds "0x" as a prefix.
   c) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification to verify data from FPGA running as server.
   d) Input Mode: Mode of FPGA to transfer data. Input '0' to set client mode.

   If inputs are valid, the operation will start.
4) After running client, current transfer size is displayed on both NiosII terminals every second.
5) "Send data complete" is displayed on NiosII terminal of client after it completes to send and receive all data and closes the connection. Total transfer size and performance are displayed on both NiosII terminals.

   Step 4) – 5) run in forever loop until user cancels the operation. To cancel operation, press any keys on server NiosII terminal and client NiosII terminal.

Figure 8-6 shows full duplex test. Left window is NiosII terminal from FPGA running as server and right window is NiosII terminal from FPGA running as client.

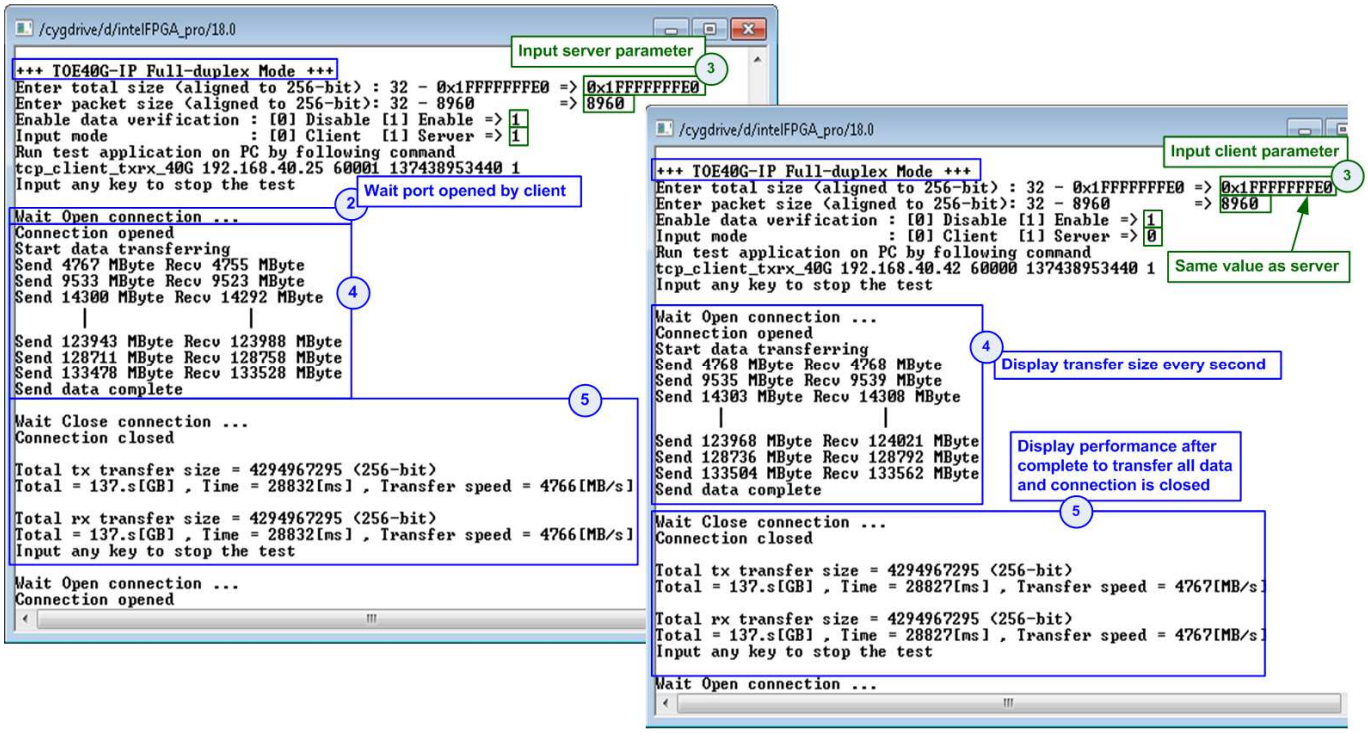

Figure 8-6 Full duplex test with data verification

9 Revision History

Revision	Date	Description
1.0	27-Dec-18	Initial version release