UDP10G IP loopback test instruction

UDP10G IP loopback test instruction

Rev1.0 30-Jun-23

This document describes the instruction to run loopback demo on FPGA development board by using AB15-SFPFMC board. The demo is designed to run TenGEMAC IP loopback test for measuring the latency time and total run time. Two EMAC IPs are integrated in the system to compare the latency time between two IPs. First EMAC IP is DG TenGEMAC IP and another IP is Intel TenGEMAC IP. Other hardwares (UDP10G IP and PCS/PMA IP) to run with TenGEMAC IP are similar. User can set test parameters on FPGA board and monitor the hardware status on NiosII command shell.

1 Environment Setup

To perform UDP10G IP loopback demo, please prepare the following test environment, as shown in Figure 1-1.

1) Cyclone10 GX FPGA development board
2) Two optical cables with 2x10 Gb SFP+ transceiver (10G BASE-R) for two loopback network connections.
3) micro USB cable for JTAG connection (FPGA programming and JTAG UART), connecting between FPGA board and PC.
4) AB15-SFPFMC, provided by Design Gateway.
5) QuartusII Programmer for programming FPGA and NiosII command shell, installed on PC

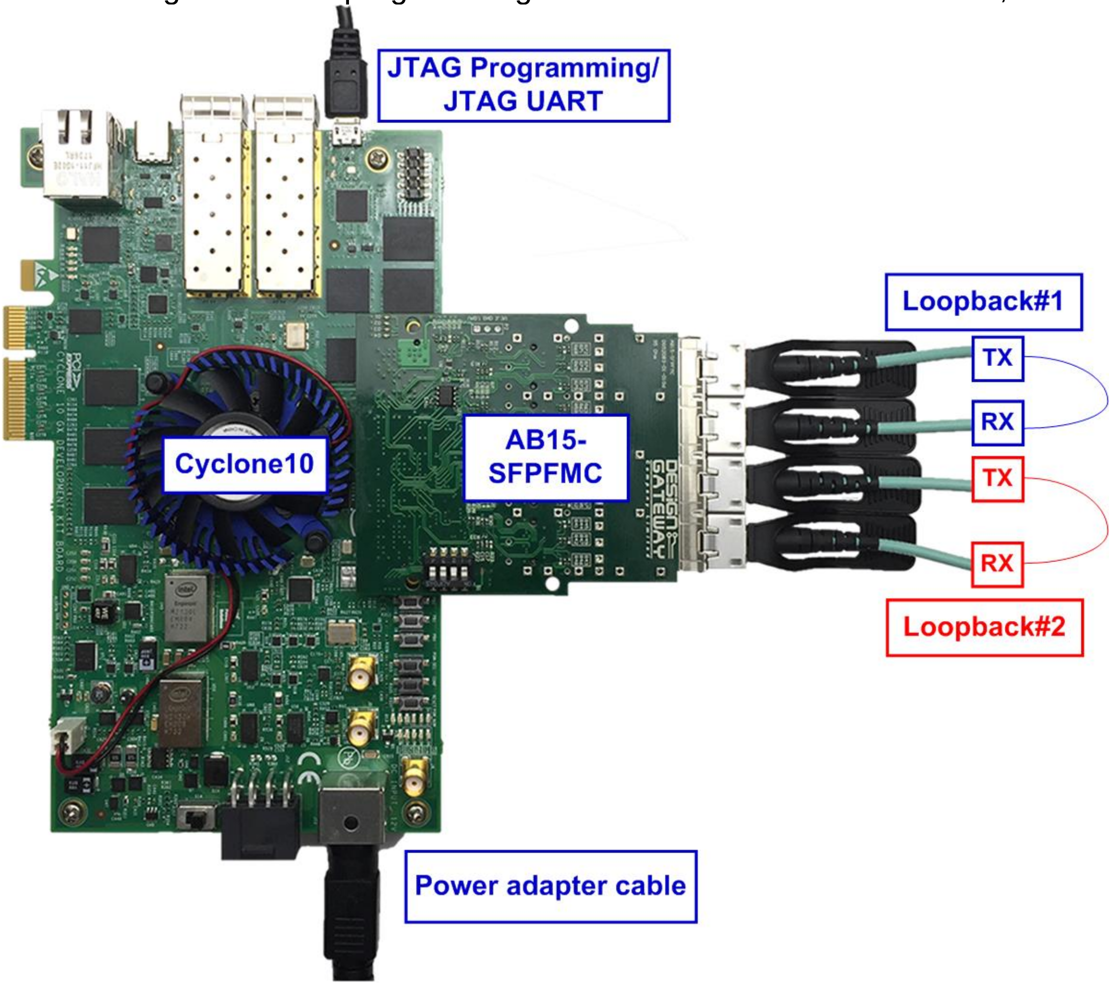


Figure 1-1Test environment for the demo

2 Demo setup

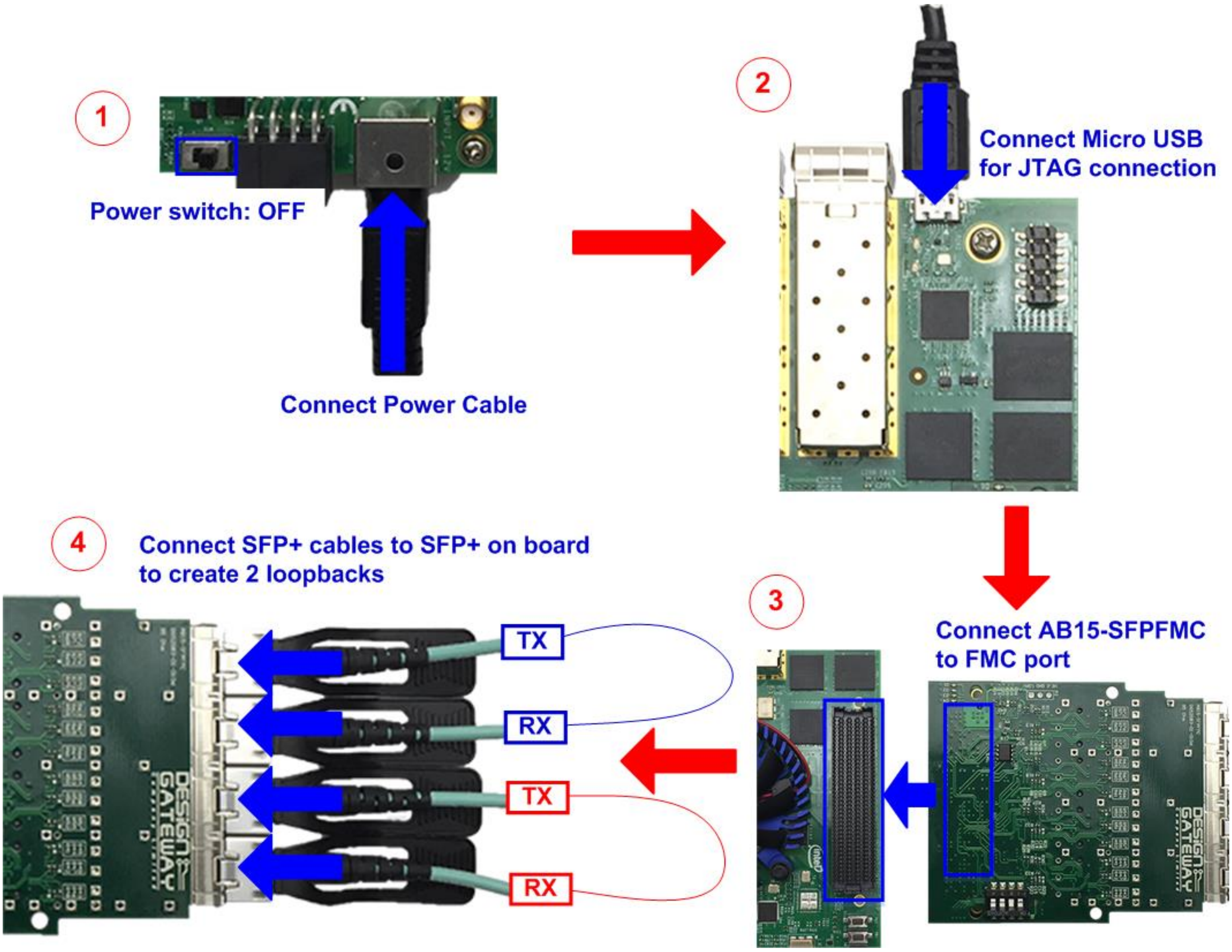
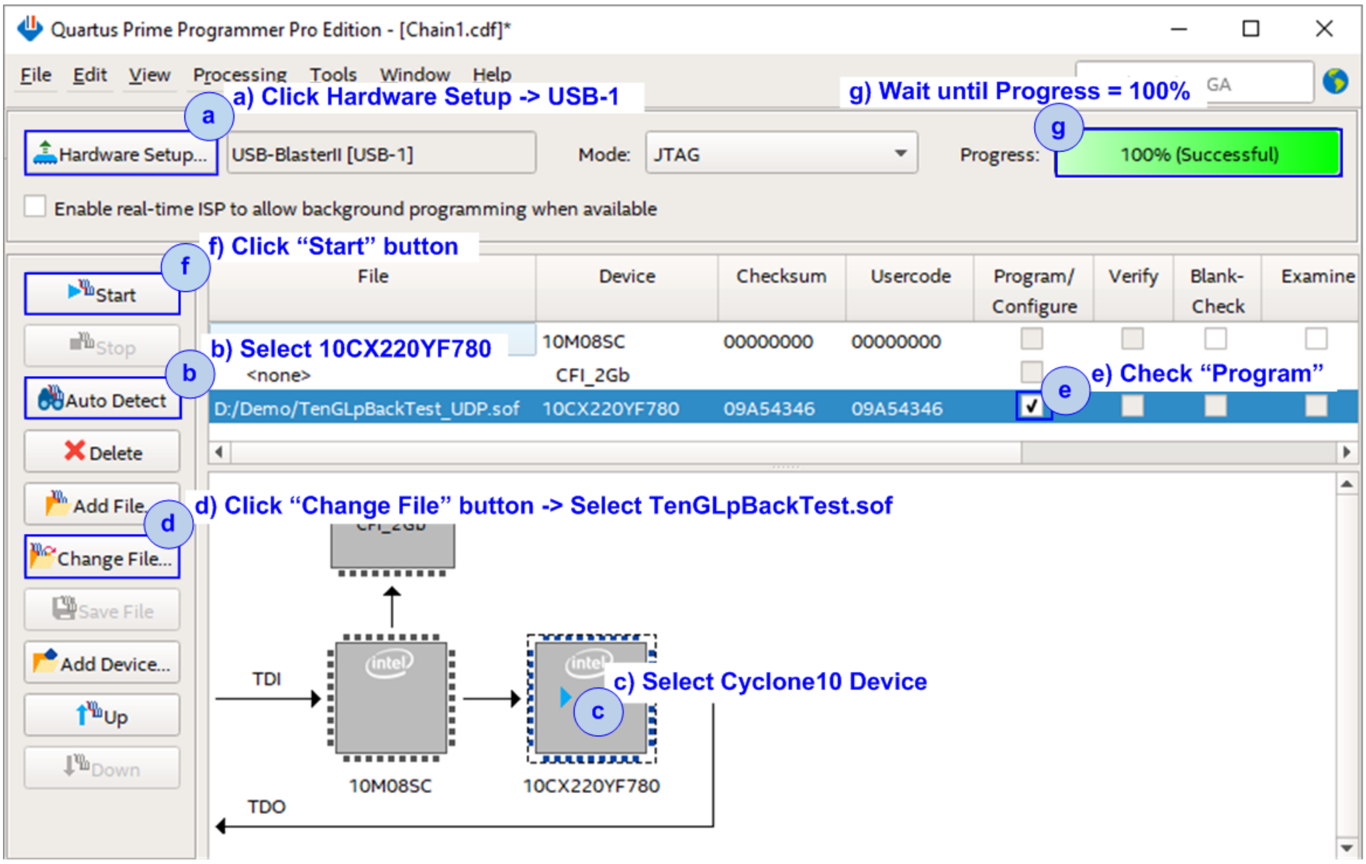


Figure 2-1 FPGA board setup

1) Turn off power switch and connect power supply to FPGA board.
2) Connect FPGA board with PC by micro USB cable.
3) Connect AB15-SFPFMC to FMC connector on FPGA board
4) Insert SFP+ transceivers with optical cables to create two loopbacks as shown in Figure 2-1.
5) Turn on power switch of FPGA board.

6) Open QuartusII Programmer to program FPGA through USB-1 by following steps.
   a. Click “Hardware Setup…” to select USB-BlasterII[USB-1].
   b. Click “Auto Detect” and select FPGA (10CX220YF780).
   c. Select Cyclone10 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 successful.



a) Click Hardware Setup -> USB-1

g) Wait until Progress = 100%

USB-BlasterII [USB-1]

Mode: JTAG

Progress: 100% (Successful)

□ Enable real-time ISP to allow background programming when available

f) Click “Start” button

b) Select 10CX220YF780

e) Check “Program”

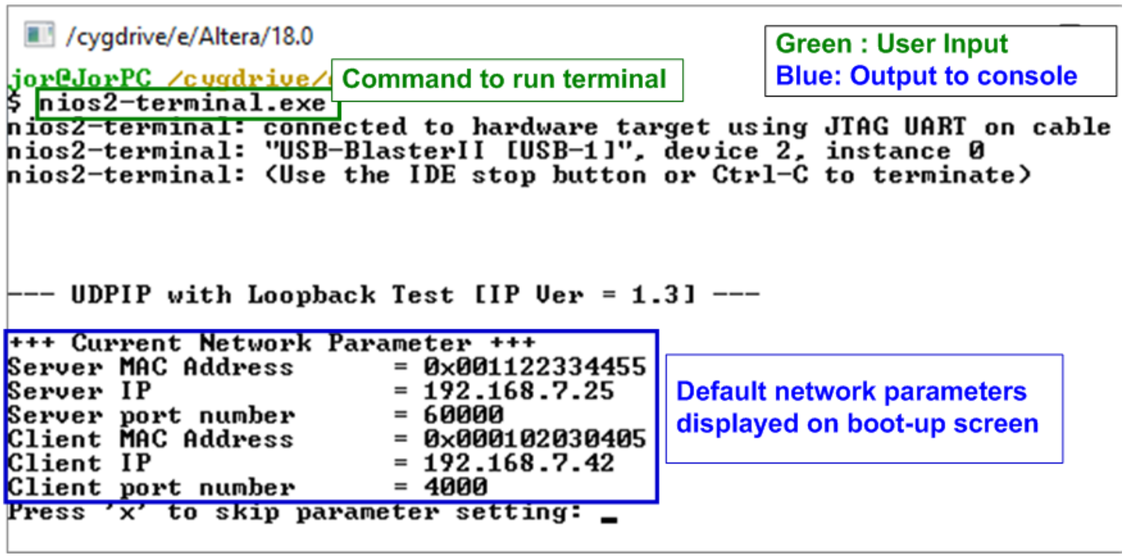
File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
	10M08SC	00000000	00000000	□	□	□	□
<none>	CFI_2Gb			□			
D:/Demo/TenGLpBackTest_UDP.sof	10CX220YF780	09A54346	09A54346	☑	□	□	□

d) Click “Change File” button -> Select TenGLpBackTest.sof

c) Select Cyclone10 Device

Figure 2-2 FPGA Programmer

7) Open NiosII command shell.
   a. Type “nios2-terminal”.
   b. Default parameters are displayed on the console.



Green : User Input
Blue: Output to console

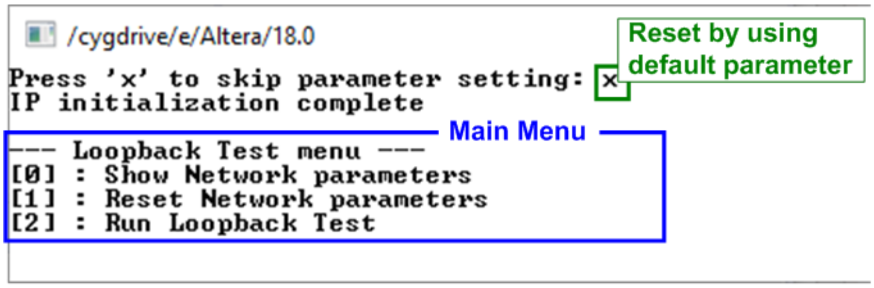
Command to run terminal

```
jor@JorPC /cygdrive/
$ nios2-terminal.exe
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "USB-BlasterII [USB-1]", device 2, instance 0
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)



--- UDPIP with Loopback Test [IP Ver = 1.3] ---

+++ Current Network Parameter +++
Server MAC Address          = 0x001122334455
Server IP                   = 192.168.7.25
Server port number          = 60000
Client MAC Address          = 0x000102030405
Client IP                   = 192.168.7.42
Client port number          = 4000
Press 'x' to skip parameter setting: _
```

Default network parameters displayed on boot-up screen

Figure 2-3 NiosII command shell after boot-up

8) Enter ‘x’ to skip parameter setting. The default parameters are applied for system initialization, as shown in Figure 2-4. If other keys are entered, the menu to change parameter will be displayed. The example of changing parameter is shown in topic 3.2 (Reset Network parameters).



Reset by using default parameter

Main Menu

```
Press 'x' to skip parameter setting: x
IP initialization complete
--- Loopback Test menu ---
[0] : Show Network parameters
[1] : Reset Network parameters
[2] : Run Loopback Test
```

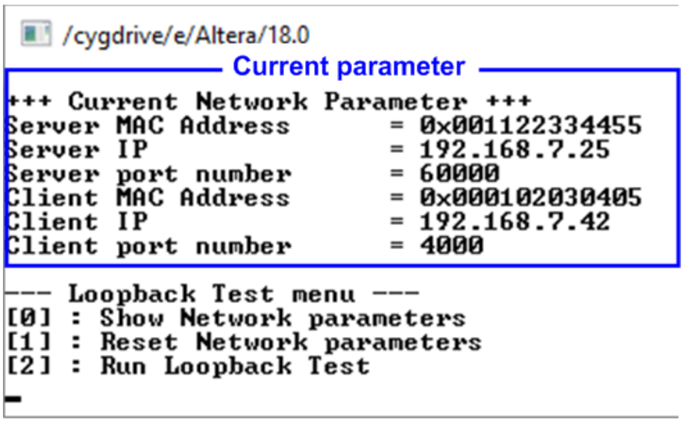
Figure 2-4 Initialization complete

3 Test menu

3.1 Show Network parameters

Select ‘0’ to check current parameter in the demo. There are six parameters displayed on the console. In this demo, both loopbacks share the same server and client network parameters.

One loopback consists of two UDP10G IPs. The server network parameters are applied to one UDP10G IP which sends data while the client network parameters are applied to another UDP10G IP which receives data.



```
+++ Current Network Parameter +++
Server MAC Address        = 0x001122334455
Server IP                 = 192.168.7.25
Server port number        = 60000
Client MAC Address        = 0x000102030405
Client IP                 = 192.168.7.42
Client port number        = 4000

--- Loopback Test menu ---
[0] : Show Network parameters
[1] : Reset Network parameters
[2] : Run Loopback Test
_
```

Figure 3-1 Display current parameter result

1) Server MAC address: 48-bit hex value to be MAC address of server. Default value is 0x001122334455.
2) Server IP: IP address of server. Default value is 192.168.7.25.
3) Server port number: Port number of server. Default value is 60000.
4) Client MAC address: 48-bit hex value to be MAC address of client. Default value is 0x000102030405.
5) Client IP: IP address of Client. Default value is 192.168.7.42.
6) Client port number: Port number of client. Default value is 4000.

To change some parameters can be done by using menu [1] Reset Network parameters.

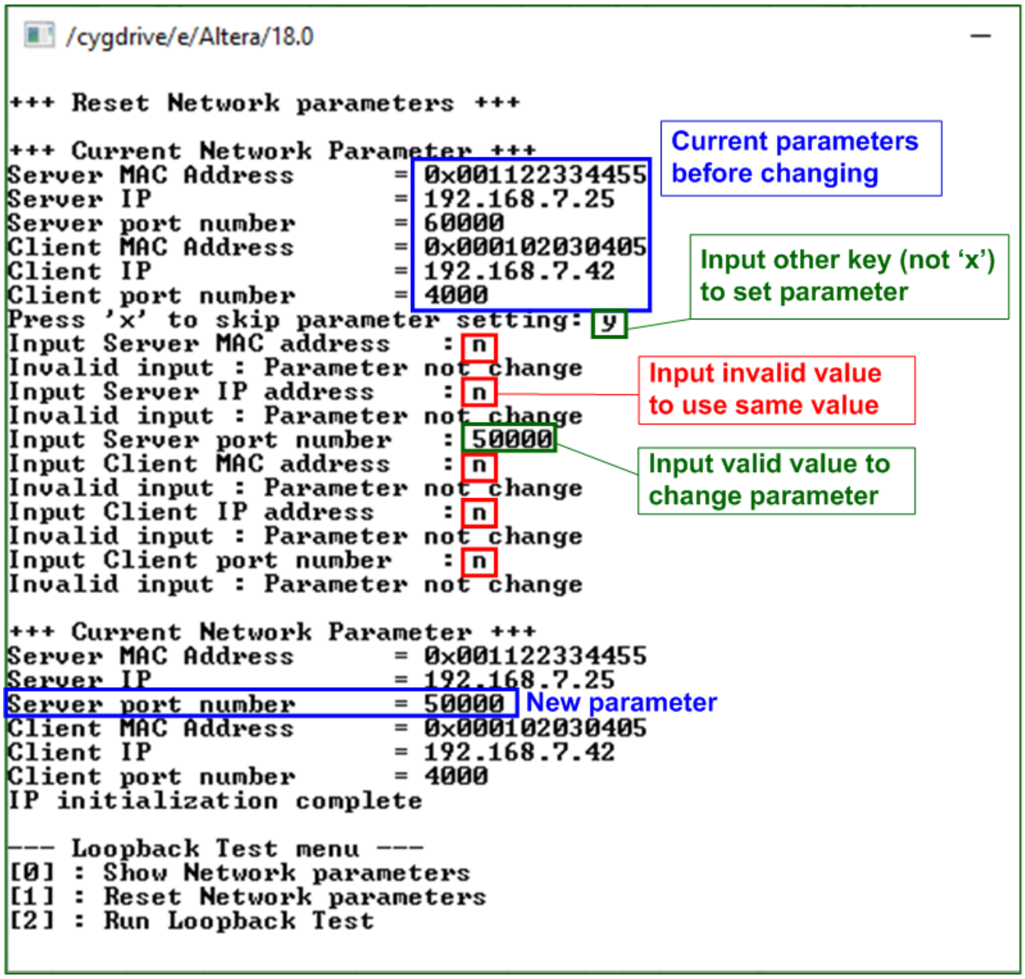
3.2 Reset Network parameters

Select '1' to reset the IP or change IP parameters. This menu is used to change IP parameters or send reset to all UDP10G IP. After user selects this menu, the current parameters are displayed on the console. Enter 'x' to use same parameters and send reset to all UDP10G IPs. Other keys are entered to change some parameters and then reset UDP10G IPs.

There are six parameters to set in this menu. Each of parameters is verified by CPU. The parameters are updated to UDP10G IP when the input is valid. If the input is not valid, the value will not change. After user assigns all parameters, UDP10G IP is reset. The description of each parameter is shown in topic 3.1 (Show Network parameter) and the range of each parameter is described as follows.

1) Server MAC address: Input 12 digit of hex value. Add "0x" as a prefix to input as hex value.
2) Server IP address: A set of four decimal digits is separated by ".". The valid range of each decimal digit is 0-255.
3) Server port number: Valid range is 0-65535.
4) Client MAC address: Input 12 digit of hex value. Add "0x" as a prefix to input as hex value.
5) Client IP address: A set of four decimal digits is separated by ".". The valid range of each decimal digit is 0-255.
6) Client port number: Valid range is 0-65535.

After user assigns all parameters, new parameter set is displayed on the console. Next, reset signal is sent to the UDP10G IPs to load new parameter set to all UDP10G IPs in the design. Finally, "IP initialization complete" is shown after all UDP10G IPs finish initialization process, as shown in Figure 3-2.



```
+++ Reset Network parameters +++

+++ Current Network Parameter +++
Server MAC Address          = 0x001122334455
Server IP                   = 192.168.7.25
Server port number          = 60000
Client MAC Address          = 0x000102030405
Client IP                   = 192.168.7.42
Client port number          = 4000
Press 'x' to skip parameter setting: y
Input Server MAC address    : n
Invalid input : Parameter not change
Input Server IP address     : n
Invalid input : Parameter not change
Input Server port number    : 50000
Input Client MAC address    : n
Invalid input : Parameter not change
Input Client IP address     : n
Invalid input : Parameter not change
Input Client port number    : n
Invalid input : Parameter not change

+++ Current Network Parameter +++
Server MAC Address          = 0x001122334455
Server IP                   = 192.168.7.25
Server port number          = 50000
Client MAC Address          = 0x000102030405
Client IP                   = 192.168.7.42
Client port number          = 4000
IP initialization complete

--- Loopback Test menu ---
[0] : Show Network parameters
[1] : Reset Network parameters
[2] : Run Loopback Test
```

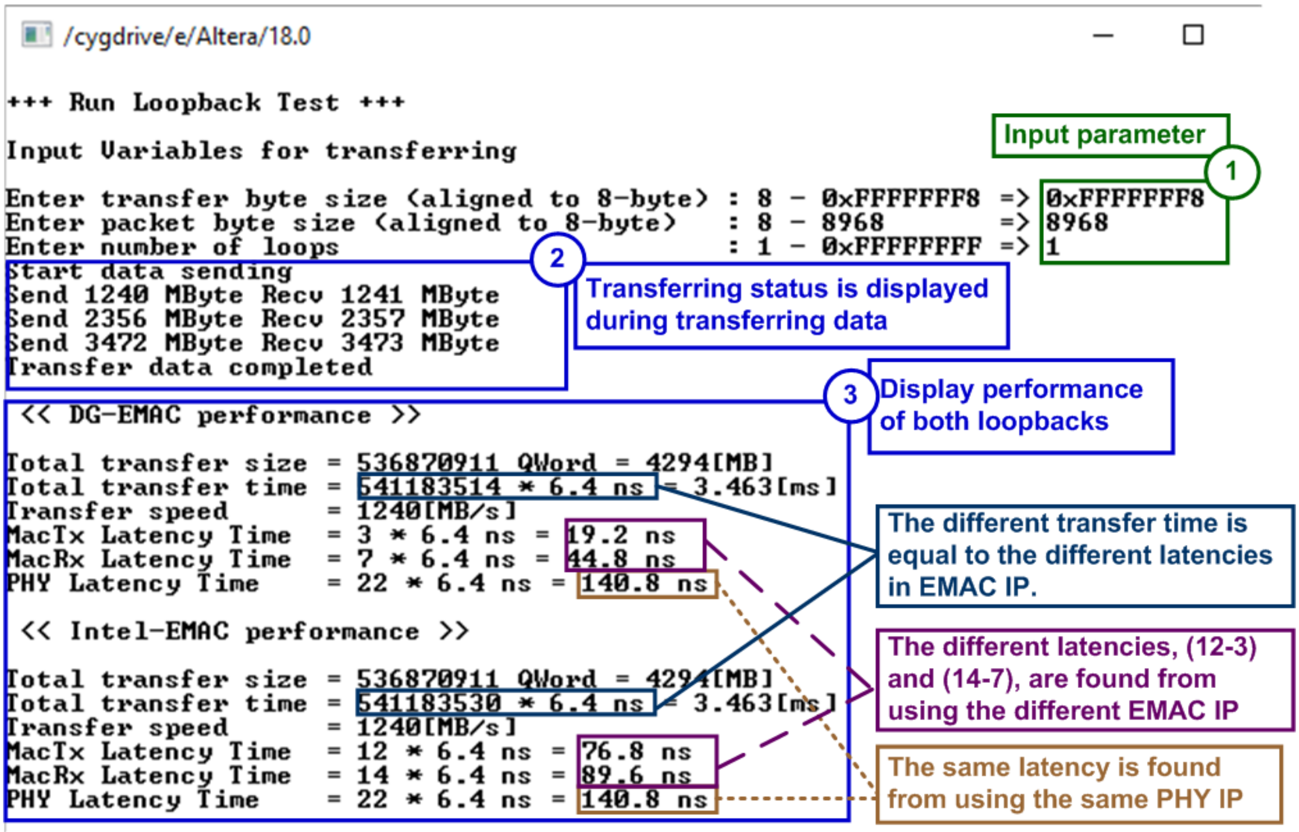
Current parameters before changing
Input other key (not 'x') to set parameter
Input invalid value to use same value
Input valid value to change parameter
New parameter

Figure 3-2 Change Network parameters result

3.3 Run Loopback Test

Select ‘2’ to run the loopback test of both loopbacks on FPGA. User enters test parameters on FPGA through NiosII command shell. The sequence to run the test is shown as follows.

1) On NiosII command shell, enter three parameters under run loopback test menu.
   a) Input transfer byte size: Unit of transfer size is byte. Valid value is 0x8 - 0xFFFFFFF8. The value must be aligned to 8. The input is decimal when only digit number is assigned. User can add “0x” as a prefix when the input is hexadecimal.
   b) Input packet byte size: Unit of packet size is byte. Valid value is 8 – 8968. The value must be aligned to 8. The input is decimal when only digit number is assigned. User can add “0x” as a prefix when the input is hexadecimal.
   c) Input number of loops: Valid value is 0x1 – 0xFFFFFFFF. Number of loops to repeat the test with the same inputs as a) and b). The input is decimal when only digit number is assigned. User can add “0x” as a prefix when the input is hexadecimal.
2) After all parameters are given and valid, both loopbacks start transferring data at the same time. Server UDP10G IP sends data to client UDP10G IP in the same loopback. All timers run to measure latency and usage time in the transfer process. During transferring data, the transmitted data size and the received data size are displayed on the console every second.
3) “Transfer data completed” is displayed on the console after all data are sent and received in the loopbacks. Finally, total transfer size, performance, and latency of both loopbacks are displayed as shown in Figure 3-3.


/cygdrive/e/Altera/18.0
Input parameter
1
2
Transferring status is displayed during transferring data
3
Display performance of both loopbacks
The different transfer time is equal to the different latencies in EMAC IP.
The different latencies, (12-3) and (14-7), are found from using the different EMAC IP
The same latency is found from using the same PHY IP

```
+++ Run Loopback Test +++

Input Variables for transferring

Enter transfer byte size (aligned to 8-byte) : 8 - 0xFFFFFFF8 => 0xFFFFFFF8
Enter packet byte size (aligned to 8-byte)   : 8 - 8968       => 8968
Enter number of loops                        : 1 - 0xFFFFFFFF => 1
Start data sending
Send 1240 MByte Recv 1241 MByte
Send 2356 MByte Recv 2357 MByte
Send 3472 MByte Recv 3473 MByte
Transfer data completed
 << DG-EMAC performance >>

Total transfer size  = 536870911 QWord = 4294[MB]
Total transfer time  = 541183514 * 6.4 ns = 3.463[ms]
Transfer speed       = 1240[MB/s]
MacTx Latency Time   = 3 * 6.4 ns = 19.2 ns
MacRx Latency Time   = 7 * 6.4 ns = 44.8 ns
PHY Latency Time     = 22 * 6.4 ns = 140.8 ns

 << Intel-EMAC performance >>

Total transfer size  = 536870911 QWord = 4294[MB]
Total transfer time  = 541183530 * 6.4 ns = 3.463[ms]
Transfer speed       = 1240[MB/s]
MacTx Latency Time   = 12 * 6.4 ns = 76.8 ns
MacRx Latency Time   = 14 * 6.4 ns = 89.6 ns
PHY Latency Time     = 22 * 6.4 ns = 140.8 ns
```

Figure 3-3 Loopback test with maximum transfer size and packet size for 1 loop

There are six values displayed as the performance result in Figure 3-3.

a) Total transfer size: Total received size which is sum of the size in each loop. The value is displayed in 64-bit (Qword) unit and B(Byte)/KB(KiloByte)/MB(MegaByte).
b) Total transfer time: Total run time displayed in clock cycle unit. One clock cycle is equal to 6.4 ns. Total time in s(second}/ms(millisecond)/us(microsecond) unit is displayed. The time duration is only measured when data is transferring in the loopback. The time using for other operations in CPU such as sending command process and calculation process is not included.
c) Transfer speed: Performance which is calculated from total transfer size and total transfer time. The unit of transfer speed is B/s (byte per second), KB/s (KiloByte per second), or MB/s (MegaByte per second).
d) MacTx Latency Time: The latency time of Tx data path within TenGEMAC IP in clock cycle unit (6.4 ns per 1 clock cycle) and nanosecond unit.
e) MacRx Latency Time: The latency time of Rx data path within TenGEMAC IP in clock cycle unit (6.4 ns per 1 clock cycle) and nanosecond unit.
f) PHY Latency Time: The latency time of physical layer and external hardware for loopback connection in clock cycle unit (6.4 ns per 1 clock cycle) and nanosecond unit.

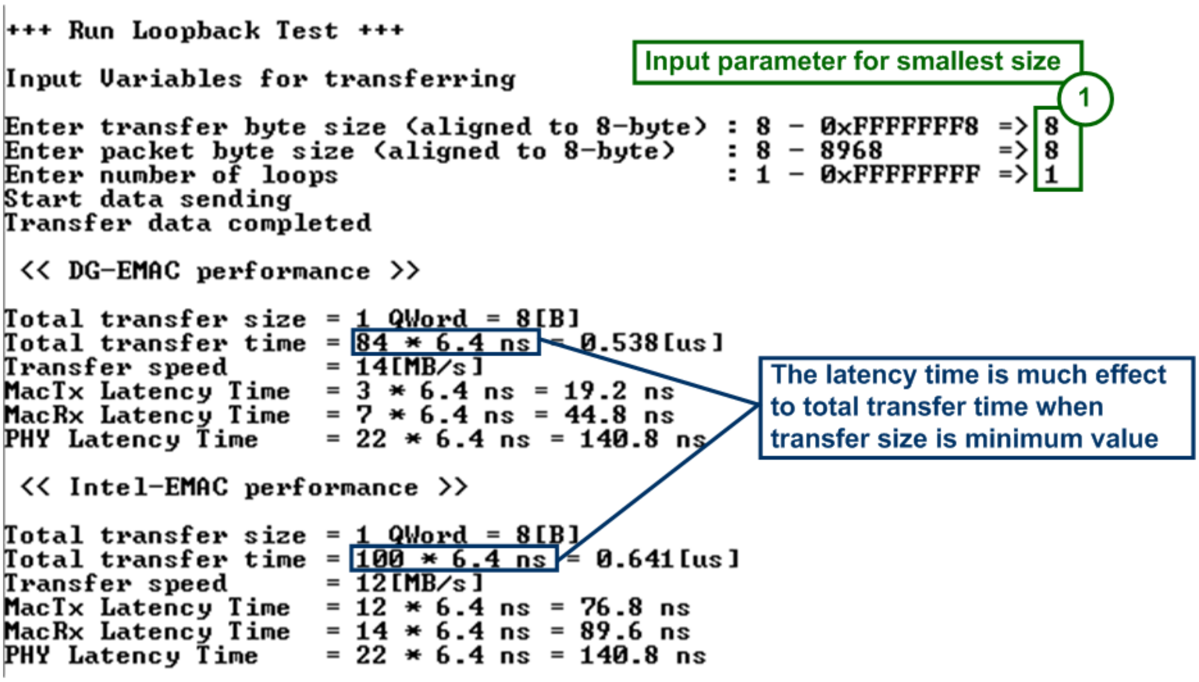
The results from two loopback designs are displayed for performance comparison. The different module is EMAC IP, so Tx latency time and Rx latency time within EMAC is different. But PHY latency time is same.

As shown in Figure 3-3,

- MacTx latency time of DG EMAC is less than Intel EMAC about 9 clock cycles.
- MacRx latency time of DG EMAC is less than Intel EMAC about 7 clock cycles.
- So, total transfer time of DG EMAC is less than Intel EMAC about 16 clock cycles (9 + 7).

The different latency time in EMAC IP has much effect when transfer size per loop is less.

As shown in Figure 3-4, the test result by using minimum size is displayed. In this case, total time of DG EMAC is less than total time of Intel EMAC about 16% ((100-84)/100).



```
+++ Run Loopback Test +++

Input Variables for transferring

Enter transfer byte size (aligned to 8-byte) : 8 - 0xFFFFFFF8 => 8
Enter packet byte size (aligned to 8-byte)   : 8 - 8968       => 8
Enter number of loops                        : 1 - 0xFFFFFFFF => 1
Start data sending
Transfer data completed

 << DG-EMAC performance >>

Total transfer size  = 1 QWord = 8[B]
Total transfer time  = 84 * 6.4 ns = 0.538[us]
Transfer speed       = 14[MB/s]
MacTx Latency Time   = 3 * 6.4 ns = 19.2 ns
MacRx Latency Time   = 7 * 6.4 ns = 44.8 ns
PHY Latency Time     = 22 * 6.4 ns = 140.8 ns

 << Intel-EMAC performance >>

Total transfer size  = 1 QWord = 8[B]
Total transfer time  = 100 * 6.4 ns = 0.641[us]
Transfer speed       = 12[MB/s]
MacTx Latency Time   = 12 * 6.4 ns = 76.8 ns
MacRx Latency Time   = 14 * 6.4 ns = 89.6 ns
PHY Latency Time     = 22 * 6.4 ns = 140.8 ns
```

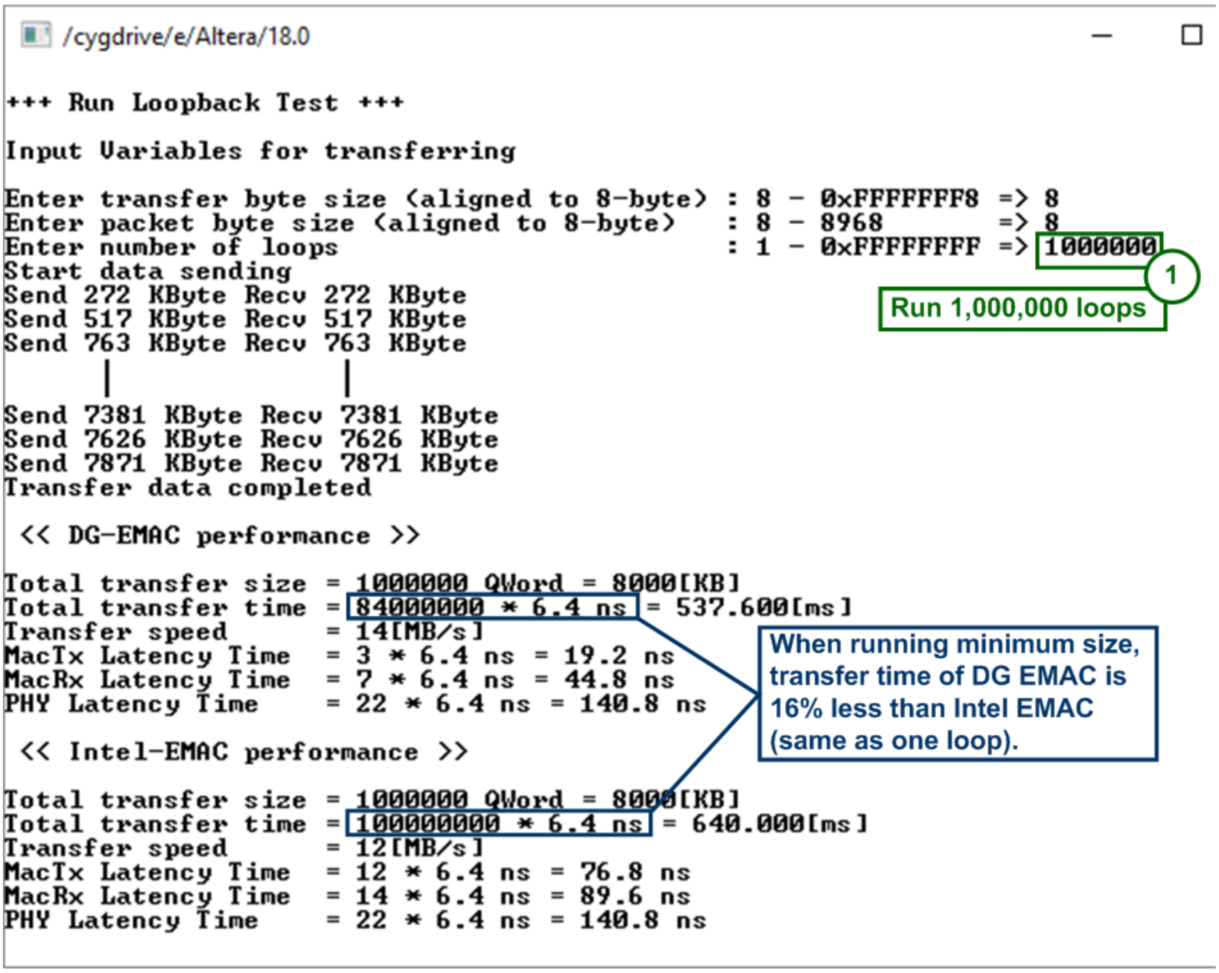
Input parameter for smallest size

1

The latency time is much effect to total transfer time when transfer size is minimum value

Figure 3-4 Loopback test with minimum transfer size and packet size for 1 loop

The different time usage is much when running small packet size for many loops. Figure 3-5 shows the test result when the size is 8 byte and runs for 1,000,000 loops.



```
+++ Run Loopback Test +++

Input Variables for transferring

Enter transfer byte size (aligned to 8-byte) : 8 - 0xFFFFFFF8 => 8
Enter packet byte size (aligned to 8-byte)   : 8 - 8968       => 8
Enter number of loops                        : 1 - 0xFFFFFFFF => 1000000
Start data sending
Send 272 KByte Recv 272 KByte
Send 517 KByte Recv 517 KByte
Send 763 KByte Recv 763 KByte
      |              |
Send 7381 KByte Recv 7381 KByte
Send 7626 KByte Recv 7626 KByte
Send 7871 KByte Recv 7871 KByte
Transfer data completed

 << DG-EMAC performance >>

Total transfer size  = 1000000 QWord = 8000[KB]
Total transfer time  = 84000000 * 6.4 ns = 537.600[ms]
Transfer speed       = 14[MB/s]
MacTx Latency Time   = 3 * 6.4 ns = 19.2 ns
MacRx Latency Time   = 7 * 6.4 ns = 44.8 ns
PHY Latency Time     = 22 * 6.4 ns = 140.8 ns

 << Intel-EMAC performance >>

Total transfer size  = 1000000 QWord = 8000[KB]
Total transfer time  = 100000000 * 6.4 ns = 640.000[ms]
Transfer speed       = 12[MB/s]
MacTx Latency Time   = 12 * 6.4 ns = 76.8 ns
MacRx Latency Time   = 14 * 6.4 ns = 89.6 ns
PHY Latency Time     = 22 * 6.4 ns = 140.8 ns
```

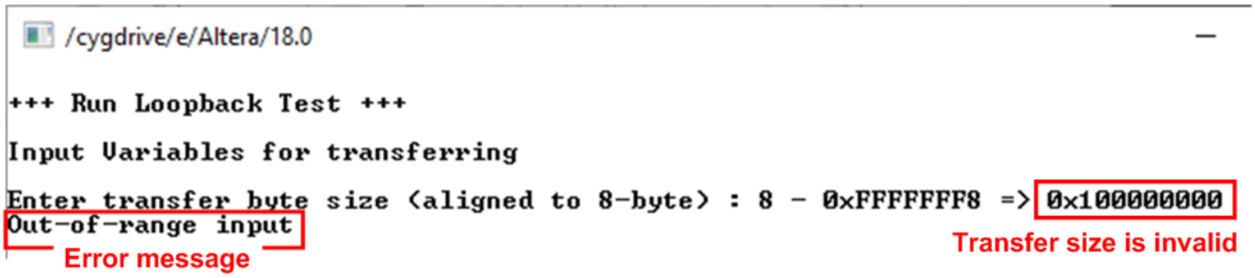
1

Run 1,000,000 loops

When running minimum size, transfer time of DG EMAC is 16% less than Intel EMAC (same as one loop).

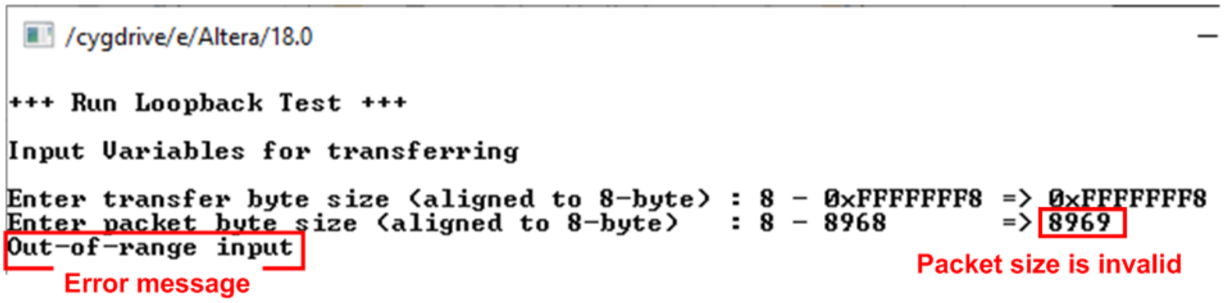
Figure 3-5 Loopback test with minimum transfer size and packet size for 1,000,000 loops

If the input is invalid, “Out-of-range input” will be displayed and the operation will be cancelled, as shown in Figure 3-6-Figure 3-8.



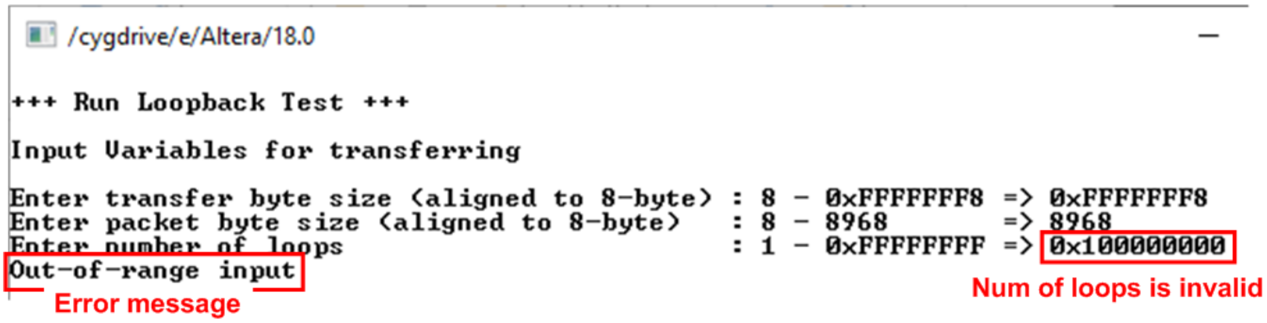
```
+++ Run Loopback Test +++

Input Variables for transferring

Enter transfer byte size (aligned to 8-byte) : 8 - 0xFFFFFFF8 => 0x100000000
Out-of-range input
```

Figure 3-6 Error from invalid transfer size



```
+++ Run Loopback Test +++

Input Variables for transferring

Enter transfer byte size (aligned to 8-byte) : 8 - 0xFFFFFFF8 => 0xFFFFFFF8
Enter packet byte size (aligned to 8-byte)   : 8 - 8968       => 8969
Out-of-range input
```

Figure 3-7 Error from invalid packet size



```
+++ Run Loopback Test +++

Input Variables for transferring

Enter transfer byte size (aligned to 8-byte) : 8 - 0xFFFFFFF8 => 0xFFFFFFF8
Enter packet byte size (aligned to 8-byte)   : 8 - 8968       => 8968
Enter number of loops                        : 1 - 0xFFFFFFFF => 0x100000000
Out-of-range input
```

Figure 3-8 Error from invalid number of loops

4 Revision History

Revision	Date	Description
1.0	15-Aug-19	Initial version release