UDP1G-IP Demo Instruction

Rev1.3 6-Mar-20

Contents

1 Overview 2

Part A UDP1G-IP demo by using FPGA and PC 3
2 Environment Setup 3
3 PC Setup 9
3.1 IP Setting 9
3.2 Speed and Frame Setting 11
3.3 Power Option Setting 13
3.4 Firewall Setting 14
4 FPGA board setup 15
5 Main menu 18
5.1 Display current parameter 18
5.2 Reset UDP1G-IP 19
5.3 Send Data Test 21
5.4 Receive Data Test 24
5.5 Full duplex Test 26

Part B UDP1G-IP demo by using two FPGAs 28
6 Environment Setup 28
7 FPGA board setup 30
8 Main menu 35
8.1 Display current parameter 35
8.2 Reset UDP1G-IP 36
8.3 Send and Receive data Test (Half-duplex test) 38
8.4 Full duplex Test 40
9 Revision History 42

1 Overview

The demo is designed to run UDP1G-IP for transferring 1 Gb Ethernet data by using UDP/IP protocol. The destination device may be Test PC or UDP1G-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 UDP1G-IP demo by using FPGA and PC” and “Part B UDP1G-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 UDP1G-IP demo by using FPGA and PC

"udpdatatest" is test application to run on Test PC for transferring data with FPGA by using UDP/IP protocol. The input parameters of "udpdatatest" by user are transfer direction (transmit or receive data), transfer size, network parameters, and test mode. Running one test application on PC can check UDP1G-IP sending or receiving data as half-duplex transferring.

Full-duplex mode could be tested by opening two command prompts to call two test applications. The first application runs to receive data from UDP1G-IP while the second application runs to send data to UDP1G-IP. Two test applications must use different port number to communicate with UDP1G-IP. More details of the demo are described as follows.

2 Environment Setup

To run UDP1G-IP demo by using FPGA and PC, please prepare following test environment.

1) FPGA development board
    - Arria 10 SoC Development Kit
    - Arria V GX FPGA Starter Kit
    - Cyclone 10 GX FPGA Development board
    - Cyclone V GT FPGA Development Kit
    - Cyclone V E FPGA Development Kit
2) PC with 1 Gb Ethernet connection
3) Ethernet cable (Cat5e or Cat6) for 1G network connection between FPGA development board and PC
4) USB cable for FPGA programming and JTAG UART, connecting between FPGA board and PC. USB cable type of each board is as follows.
    - micro USB cable:
      Arria 10 SoC Development Kit
      Cyclone 10 GX Development Kit
    - USB-AB cable:
      Arria V GX Starter Kit
      Cyclone V GT FPGA Development Kit
      Cyclone V E FPGA Development Kit
5) "udpdatatest.exe": Test application provided by Design Gateway, installed on PC
6) QuartusII programmer for programming FPGA and NiosII command shell, installed on PC

Figure 2-1 UDP1G-IP demo (FPGA<->PC) on Arria10 SoC

Figure 2-2 UDP1G-IP demo (FPGA<->PC) on Cyclone10 GX

Figure 2-3 UDP1G-IP demo (FPGA<->PC) on ArriaV GX Starter

Figure 2-4 UDP1G-IP demo (FPGA<->PC) on CycloneV GT

Figure 2-5 UDP1G-IP demo (FPGA<->PC) on CycloneV E

3 PC Setup

Before running the demo, please check the network setting on PC. The example for setting the network is shown as follows.

3.1 IP Setting

Figure 3-1 IPv4 setting

1) Open Ethernet setting option from Control Panel -> Network and Internet -> Network and Sharing Center.
2) Click Gigabit Ethernet icon which is used to connect with FPGA board.

Figure 3-2 Select IP address setting menu

3) Click Properties button in Ethernet Status window.
4) Select “TCP/IPv4”.
5) Click Properties button in Ethernet Properties window.

Figure 3-3 Setting IP address for PC

6) Set IP address = 192.168.11.25 and Subnet mask = 255.255.255.0. After that, click OK button to confirm IP address setting.

3.2 Speed and Frame Setting

Figure 3-4 Jumbo frame setting

1) Click Configure button in Ethernet Properties window.
2) On Advanced Tab, select “Jumbo Packet” and then set Value to “9014 Bytes” for Jumbo Frame support.

3) On Link Speed, select “1 Gbps Full Duplex” and click OK button, as shown in Figure 3-5.

Figure 3-5 Set link speed = 1 Gbps

3.3 Power Option Setting

1) Open Control Panel and select Power Options, as shown in the left window of Figure 3-6.
2) Set High Performance plan, as shown in the right window of Figure 3-6.

Figure 3-6 Power options

3.4 Firewall Setting

Figure 3-7 Firewall setting

1) Open Control Panel and select Windows Firewall.
2) Click “Turn Windows Firewall on or off”.
3) Select Turn off Firewall under Private and Public network settings.
4) Click OK button to confirm the setting.

4 FPGA board setup

1) Turn off power switch and connect power supply to FPGA board.
2) Connect USB cable, micro USB or USB AB cable depending on FPGA board, between FPGA board and PC for FPGA configuration and JTAG UART.
3) Connect CAT5e/CAT6 cable between PC and Ethernet connection of FPGA board. User must use the right port when FPGA board has two 1Gb Ethernet ports.

Figure 4-1 Power, USB, and Ethernet connection

4) Turn on power switch on FPGA board.

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 device.
   c) Select FPGA 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 terminal.
   a. Type “nios2-terminal”.
   b. Input ‘0’ to initialize UDP1G-IP in client mode (ask PC MAC address by sending ARP request).
   c. Default parameter in client mode is displayed on the console.

```
/cygdrive/e/Altera/18.0
-----------------------------------------------
Altera Nios2 Command Shell

Version 18.0, Build 219
-----------------------------------------------

GunPGunPC /cygdrive/e/Altera/18.0
$ nios2-terminal
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)


--- UDP1GIP with CPU Demo [Ver = 1.7] -

Input mode : [0] Client [1] Server => 0

+++ Current Network Parameter +++
Mode                  = CLIENT
FPGA MAC address      = 0x000102030405
FPGA IP               = 192.168.11.42
FPGA port number      = 4000
Target IP             = 192.168.11.25
Target port number (Target->FPGA) = 61000
Target port number (FPGA->Target) = 60000
Press 'x' to skip parameter setting:_
```

Green : User Input
Blue: Output to console
Command to run terminal
Input ‘0’ to initialize in client mode
Default client parameter display on boot-up screen

Figure 4-3 Message after system boot-up

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

```
/cygdrive/e/Altera/18.0
Press 'x' to skip parameter setting:x
IP initialization complete

--- UDP1GIP menu ---
[0] : Display UDPIP parameters
[1] : Reset UDPIP parameter
[2] : Send Data Test (UDPIP -> Target)
[3] : Receive Data Test (Target -> UDPIP)
[4] : Full duplex Test (UDPIP <-> Target)
```

Reset by using default parameter
Main Menu

Figure 4-4 Initialization complete

Note: Transfer performance in the demo depends on PC resource and network performance 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 NiosII command shell.

Figure 5-1 Display current parameter result

1) Mode : Mode of UDP1G-IP to run as server or client. To run with PC, please input '0' to initialize as client mode.
2) FPGA MAC address : 48-bit hex value to be MAC address of FPGA. Default value is 0x000102030405.
3) FPGA IP : IP address of FPGA. Default value is 192.168.11.42.
   Note: This value is used to be FPGA IP address, parameter for test application on PC.
4) Target IP : IP address of destination device (1 Gb Ethernet on PC). Default value is 192.168.11.25.
5) FPGA port number: Port number of FPGA. Default value is 4000.
   Note: This value is used to be FPGA port, parameter for test application on PC.
6) Target port number (Target->FPGA) : Port number of the destination device to send data to FPGA. Default value is 61000.
   Note: This value is used to be PC port, parameter for test application on PC in transmit mode.
7) Target port number (FPGA->Target) : Port number of the destination device to receive data from FPGA. Default value is 60000.
   Note: This value is used to be PC port, parameter for test application on PC in received mode.

To change some parameters, user can set by using menu [1] (Reset UDP1GIP parameter).

5.2 Reset UDP1G-IP

Select ‘1’ to reset the IP and change IP parameters. After user selects this menu, the current parameters are displayed. User enters ‘x’ to use same parameters or other keys to change some parameters. There are seven parameters which can be modified by user. When user enters invalid value such as ‘n’, the parameter does not change. After setting parameters, IP is reset. More details of each parameter are described as follows.

1) Mode : ‘0’ to initialize as client mode.
2) FPGA MAC address : 12-digit of hex value. Add “0x” to be prefix to input as hex value.
3) FPGA IP address : A set of four decimal digits is separated by “.”. The valid range of each decimal digit is 0-255.
4) FPGA port number : Valid range is 0-65535.
5) Target IP address : A set of four decimal digits like FPGA IP address. This value is IP address of Test PC.
6) Target port number (Target->FPGA) : Valid range is 0-65535.
7) Target port number (FPGA->Target) : Valid range is 0-65535.

After finishing all parameter assignment, new parameters are displayed on NiosII command shell. Next, reset signal is sent to the IP for loading new parameters. Finally, “IP initialization complete” is shown after IP completes initialization process, as shown in Figure 5-2.

```
/cygdrive/e/Altera/18.0

+++ Reset UDPIP +++

+++ Current Network Parameter +++
Mode                    = CLIENT
FPGA MAC address        = 0x000102030405
FPGA IP                 = 192.168.11.42
FPGA port number        = 4000
Target IP               = 192.168.11.25
Target port number (Target->FPGA) = 61000
Target port number (FPGA->Target) = 60000
Press 'x' to skip parameter setting:y
Input mode : [0] Client [1] Server => n
Invalid input : Parameter not change
Input FPGA MAC address    : n
Invalid input : Parameter not change
Input FPGA IP address     : n
Invalid input : Parameter not change
Input FPGA port number    : 50000
Input Target IP address   : n
Invalid input : Parameter not change
Input Target port number (Target->FPGA)   : n
Invalid input : Parameter not change
Input Target port number (FPGA->Target)   : n
Invalid input : Parameter not change

+++ Current Network Parameter +++
Mode                    = CLIENT
FPGA MAC address        = 0x000102030405
FPGA IP                 = 192.168.11.42
FPGA port number        = 50000
Target IP               = 192.168.11.25
Target port number (Target->FPGA) = 61000
Target port number (FPGA->Target) = 60000
WARNING: Please also change IP setting and port number on Test appli
IP initialization complete

--- UDP1GIP menu ---
[0] : Display UDPIP parameters
[1] : Reset UDPIP parameter
[2] : Send Data Test (UDPIP -> Target)
[3] : Receive Data Test (Target -> UDPIP)
[4] : Full duplex Test (UDPIP <-> Target)
```

Green : User Input
Blue: Output to console

Current parameter 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

Reset complete

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 "udpdatatest" on PC to receive data. User sets test parameters on FPGA for sending data through NiosII command shell. On PC, user sets test parameters of "udpdatatest" to receive data through command prompt. The step to run the test is described as follows.

1) On NiosII command shell, set two parameters under send data test menu.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   b) Input packet size: Unit of packet size is byte. Valid value is 1 – 8972. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   Note: When packet size is more than 1472, the packet output from UDP1G-IP is jumbo frame size. In this case, Test PC must support jumbo frame.
2) When all inputs are valid, the recommended parameters to run test application on PC are displayed. After that, "Press any key to start data sending ..." is displayed to begin data sending when user enters some keys.
3) On command prompt, user runs the application by using the recommended parameters. There are five parameters for "udpdatatest".
   >> udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [ByteLen] <Pattern> <Timeout>
   a) Dir : Set 'r' to receive test data from FPGA
   b) FPGA IP : Set the same value as FPGA IP address
   c) FPGA port : Set the same value as FPGA port number
   d) PC port : Set the same value as target port number (FPGA->Target)
   e) ByteLen : Set the same value as "Input transfer size" of step 1a)
   f) Pattern : '1'- enable data verification, '0'-disable data verification. Default value is '1'.
   g) Timeout : Timeout in msec unit. Valid value is 100-65536. It is recommended to set to 1000 (1 sec). Default value is 100.
4) After running test application, the summary of setting parameter is displayed. After that, PC waits received data from FPGA.
5) On NiosII command shell, user enters any key(s) to start sending data. During transferring data, current transfer size is displayed on NiosII command shell (transmit size) and command prompt (received size) every second.
6) "Send data complete" is displayed on NiosII command shell after all data are sent. On PC, test application is finished by one of two conditions. First, total data have been received (no lost data is found within networking). Second is timeout condition when no additional data is received more than 1 second. After timeout is found, timeout message with the 1st error position is displayed. Finally, total transfer size and performance are displayed on NiosII command shell (transmit performance) and command prompt (received performance).

Figure 5-3 shows the example of send data test when using non-jumbo frame size. The left window is NiosII command shell on FPGA while the right window is command prompt on PC.

Figure 5-4 shows the example of send data test when using jumbo frame size. Performance when running jumbo frame is better than non-jumbo frame.

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

Figure 5-4 Send data test by using jumbo frame

When some inputs are invalid, “Out-of-range input” is displayed. After that, the operation is cancelled, as shown in Figure 5-5 - Figure 5-6.

```
/cygdrive/e/Altera/18.0

+++ UDPIP Send Mode +++
Enter transfer size : 1 - 0xFFFFFFFF => 0x100000000
Out-of-range input
                                      Error transfer size
--- UDP1GIP menu ---
[0] : Display UDPIP parameters
[1] : Reset UDPIP parameter
[2] : Send Data Test (UDPIP -> Target)
[3] : Receive Data Test (Target -> UDPIP)
[4] : Full duplex Test (UDPIP <-> Target)
```

Figure 5-5 Error from invalid transfer size

```
/cygdrive/e/Altera/18.0

+++ UDPIP Send Mode +++
Enter transfer size : 1 - 0xFFFFFFFF => 0xFFFFFFFF
Enter packet size   : 1 - 8972       => 8973
Out-of-range input
                                      Error packet size
--- UDP1GIP menu ---
[0] : Display UDPIP parameters
[1] : Reset UDPIP parameter
[2] : Send Data Test (UDPIP -> Target)
[3] : Receive Data Test (Target -> UDPIP)
[4] : Full duplex Test (UDPIP <-> Target)
```

Figure 5-6 Error from invalid packet size

5.4 Receive Data Test

To transfer data from PC to FPGA, select ‘3’ to run receive data test on FPGA and prepare “udpdatatest” on PC to send data. User sets test parameters on FPGA for receiving data through NiosII command shell. On PC, user sets parameters of “udpdatatest” to send data through Command prompt. The step to run the test is shown as follows.

1) On NiosII command shell, set two parameters under send data test menu.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add “0x” as a prefix for hexadecimal input.
   b) Input data verification mode: ‘0’-disable data verification, ‘1’-enable data verification sent from PC.
2) When all inputs are valid, the recommended parameters to run test application on PC are displayed. After that, “Wait data from Target ...” is displayed on the console when IP is ready to receive data from PC.
3) On command prompt, user runs the application by using the recommended parameters. There are five parameters for “udpdatatest”.
   >> udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [Bytelen]
   a) Dir : Set ‘t’ to send test data from FPGA
   b) FPGA IP : Set the same value as FPGA IP address
   c) FPGA port : Set the same value as FPGA port number
   d) PC port : Set the same value as target port number (Target->FPGA)
   e) Bytelen : Set the same value as “Input transfer size” of step 1a)
4) After running test application, test application starts sending data out to FPGA. During transferring data between FPGA and PC, current transfer size is displayed on both NiosII command shell (received size) and command prompt (transmit size) every second.
5) “Receive data completed” is displayed on NiosII command shell after FPGA receives all data or 1-second timeout is found. Finally, total transfer size and performance are displayed on NiosII command shell (received performance) and Command prompt (transmit performance).

Figure 5-7 shows the example of receive data test when disable data verification mode on FPGA. The left window is test result on NiosII command shell while the right window is test result on command prompt.

Figure 5-8 shows the example of receive data test when enable data verification mode on FPGA and PC sends incremental data to FPGA. When verification module is failed, error message is displayed.

Figure 5-7 Receive data test without data verification

Figure 5-8 Receive data test with enable data verification

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 sets test parameters on FPGA through NiosII command shell. On PC, user opens two applications on two command prompts for sending and receiving data with FPGA by using different port number. The step to run the test is described as follows.

1) On NiosII command shell, set three parameters under full duplex test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal unit.
   b) Input packet size: Unit of packet size is byte. Valid value is 1 – 8972. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal unit.
   c) Input data verification mode: '0'-disable data verification, '1'-enable data verification.
2) When all inputs are valid, the recommended parameters for two applications are displayed, sending data test and receiving data test.
   Next, "Press any key to start data sending ..." is displayed to wait user start IP sending data.
3) On PC, open two command prompts to run the test applications by using the recommended parameters.

   Seven parameters are applied for the 1st command prompt (receiving data).
   >> udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [ByteLen] <Pattern> <Timeout>

   Five parameters are applied for the 2nd command prompt (sending data).
   >> udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [ByteLen]

   a) Dir : Set 'r' to the 1st command prompt
      Set 't' to the 2nd command prompt
   b) FPGA IP : Set the same value as FPGA IP address
   c) FPGA port : Set the same value as FPGA port number
   d) PC port : Set the same value as target port number (FPGA->Target) for the 1st command prompt. Set the same value as target port number (Target->FPGA) for the 2nd command prompt.
   e) Bytelen : Set the same value as "Input transfer size" of step 1a)
   f) Pattern : '1'- enable data verification, '0'-disable data verification.
      Default value is '1'.
   g) Timeout : Timeout in msec unit. Valid value is 100-65536.
      It is recommended to set to 1000 (1 sec). Default value is 100.
4) On NiosII command shell, press any key(s) to start sending data to PC. After that, current transfer size in both directions is displayed on NiosII command shell and two command prompts every second.
5) "Transfer data complete" is displayed on NiosII command shell after UDP1G-IP finishes sending and receiving all data. Finally, total transfer size and performance are displayed on NiosII command shell and command prompt.

As shown in Figure 5-9, transfer performance when running full duplex with data verification is displayed. The left window is the test result on NiosII command shell while the right window is the test result on command prompt (the upper window is received performance and the lower window is transmit performance).

Figure 5-9 Full duplex test

Part B UDP1G-IP demo by using two FPGAs

6 Environment Setup

As shown in Figure 6-1, to run UDP1G-IP demo by using two FPGAs, please prepare following test environment.

1) Two FPGA development boards from following list
    - Arria 10 SoC Development Kit
    - Arria V GX FPGA Starter Kit
    - Cyclone 10 GX FPGA Development board
    - Cyclone V GT FPGA Development Kit
    - Cyclone V E FPGA Development Kit

    Note: Two FPGA development board can either be the same board or not.
2) Ethernet cable (Cat5e or Cat6) cable for network connection between two FPGA boards
3) Two USB cables for connecting two FPGA boards to PC. USB cable type depends on FPGA board as follows.
    - micro USB cable:
      Arria 10 SoC Development Kit
      Cyclone 10 GX Development Kit
    - USB-AB cable:
      Arria V GX Starter Kit
      Cyclone V GT FPGA Development Kit
      Cyclone V E FPGA Development Kit
4) QuartusII Programmer for FPGA programming and NiosII command shell, installed on PC.

Figure 6-1 UDP1G-IP demo (FPGA<->FPGA) by Arria10 SoC

7 FPGA board setup

Please follow step 1) – 4) of topic 4 (FPGA board setup) to prepare FPGA board for running the demo. Next, configure FPGA by following steps.

1) Insert Ethernet cable (Cat5e or Cat6) cable between two FPGA boards.

Figure 7-1 Ethernet cable connection between two FPGA boards

2) Turn on power switch on FPGA board#1 and board#2. Now PC detects USB cable from two FPGA boards as USB-1 and USB-2.
3) Open QuartusII Programmer to program FPGA board#1 through USB-1.
   a. Click “Hardware Setup…” to select USB-BlasterII[USB-1].
   b. Click “Auto Detect” and select FPGA.

Figure 7-2 Select USB-BlasterII

4) Program SOF file to FPGA through USB-1.
   a. Click “Hardware Setup…” to select USB-BlasterII[USB-1].
   b. Click “Auto Detect” and select FPGA device.
   c. Select FPGA 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 7-3 Program SOF file

5) Change to USB-2 to program FPGA board#2 and repeat step 3) – 4).

Figure 7-4 Change to USB-2 for programming another FPGA

6) Open NiosII command shell.
   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 command shell of FPGA board#1 for running server mode.
      Set ‘0’ on NiosII command shell of FPGA board#2 for running client mode.

Figure 7-5 Input mode

7) Default parameters are displayed on NiosII command shell, depending on the mode (Server/Client). Please setup parameters of server mode before client mode.
   a. For server mode, when user does not change default parameters, enter ‘x’ to skip parameter setting, as shown in Figure 7-6.

Figure 7-6 NiosII command shell to display default parameter in server mode

b. For client mode, user must change target port number (Target->FPGA) to use the same value as target port number (FPGA->Target), as shown in Figure 7-7.

```
/cygdrive/e/Altera/18.0

--- UDP1GIP with CPU Demo [Ver = 1.7] ---

Input mode : [0] Client [1] Server => 0

+++ Current Network Parameter +++
Mode                    = CLIENT
FPGA MAC address        = 0x000102030405
FPGA IP                 = 192.168.11.42
FPGA port number        = 4000
Target IP               = 192.168.11.25
Target port number (Target->FPGA) = 61000
Target port number (FPGA->Target) = 60000
Press 'x' to skip parameter setting: y
Input mode : [0] Client [1] Server => n
Invalid input : Parameter not change
Input FPGA MAC address    : n
Invalid input : Parameter not change
Input FPGA IP address     : n
Invalid input : Parameter not change
Input FPGA port number    : n
Invalid input : Parameter not change
Input Target IP address   : n
Invalid input : Parameter not change
Input Target port number (Target->FPGA)   : 60000
Input Target port number (FPGA->Target)   : n
Invalid input : Parameter not change

+++ Current Network Parameter +++
Mode                    = CLIENT
FPGA MAC address        = 0x000102030405
FPGA IP                 = 192.168.11.42
FPGA port number        = 4000
Target IP               = 192.168.11.25
Target port number (Target->FPGA) = 60000
Target port number (FPGA->Target) = 60000
WARNING: Please also change IP setting and port number on Test application
```

Blue: Output to console
Green: User input

Default parameters for client mode

Input to set parameter

Input invalid value to use old value

Change Target port to use same value for both Tx and Rx directions

New parameters for client mode

Figure 7-7 NiosII command shell to modify default parameter in client mode

Note: Please complete to setup parameter and reset process on server before client. Server must wait ARP request from client after reset process.

8) After that, IP starts initialization process. When finishing initialization process, “IP initialization complete” is displayed. Finally, main menu is displayed on NiosII command shell. More details of each menu are described in the next topic.

Figure 7-8 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 command shell.

Figure 8-1 Display current parameter result

1) Mode : Mode of UDP1G-IP to run as server or client.
‘0’- Client, ‘1’- Server
2) FPGA MAC address : 48-bit hex value to be MAC address of FPGA. Default value is 0x000102030405 (client mode) or 0x001122334455 (server mode).
3) FPGA IP : IP address of FPGA. Default value is 192.168.11.42 (client mode) or 192.168.11.25 (server mode).
4) FPGA port number : Port number of FPGA. Default value is 4000 (client mode) or 60000 (server mode).
5) Target IP : IP address of destination device. Default value is 192.168.11.25 (client mode) or 192.168.11.42 (server mode).
6) Target port number (Target->FPGA) : Port number of the destination device to send data to UDP1G-IP. Default value is 60001 (client mode) or 4000 (server mode).

   In client mode, this value must be changed from 60001 to 60000 (same as default value of Target port number for FPGA -> Target).
7) Target port number (FPGA->Target) : Port number of the destination device to receive data from UDP1G-IP. Default value is 60000 (client mode) or 4000 (server mode).

To change some parameters, user can use menu [1].

8.2 Reset UDP1G-IP

Select ‘1’ to reset the IP and change IP parameters. This menu is applied for changing IP parameters. After user selects this menu, the current parameters are displayed. User enters ‘x’ to use same parameters or other keys to change some parameters. There are seven parameters which can be modified by user. When user enters invalid value such as ‘n’, the parameter does 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 sent by client.*
2. *Target port number for sending and receiving data must be same value.*
3. *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 (both Tx and Rx direction) = FPGA port number of client*
   d. *FPGA port number of server = Target port number of client (both Tx and Rx direction)*

1) Mode : ‘0’- Client, ‘1’- Server to det FPGA initialization mode. Two FPGA boards must run in different mode (one is client and another is server).
2) FPGA MAC address : 12-digit of hex value. Add “0x” as a prefix for hex input.
3) FPGA IP address : A set of four decimal digits is separated by “.”. The valid range of decimal digit is 0-255.
4) FPGA port number : Valid range is 0-65535.
5) Target IP address : A set of four decimals like FPGA IP address.
6) Target port number (Target->FPGA) : Valid range is 0-65535.
7) Target port number (FPGA->Target) : Valid range is 0-65535.

After finishing all parameter assignment, new parameters are displayed on NiosII command shell. 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 and Receive data Test (Half-duplex test)

The operation to send and receive data test of server and client mode is similar. So, this topic describes the menu to transfer data between two FPGA boards. The first board runs receive data test (select menu '3') and the second board runs send data test (select menu '2').

User sets test parameters through NiosII command shell. The step to run the test is described as follows.

1) On NiosII command shell under menu 3 (receive data test), set two parameters.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   b) Input data verification mode: '0'-disable data verification, '1'-enable data verification.
2) When all inputs are valid, "Wait data from Target ..." is displayed.
3) On NiosII command shell under menu 2 (send data test), set two parameters.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input. This value must be equal to transfer size under receive data test.
   b) Input packet size: Unit of packet size is byte. Valid value is 1 – 8972. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   Note: When packet size is more than 1472, the packet output from UDP1G-IP is jumbo frame. User needs to confirm that network device supports jumbo frame when two FPGA boards connect through the network device.
4) When all inputs are valid, "Press any key to start data sending ..." is displayed. User enters some key(s) to start sending data.
5) During transferring data, current transfer size is displayed on both NiosII command shells every second.
6) "Send data complete" or "Receive data completed" are displayed on NiosII command shell after finishing transferring all data. Finally, total transfer size and performance are displayed on both NiosII command shells.

Figure 8-3 shows the example to transfer data between two FPGAs by using non-jumbo frame size. Left window is NiosII command shell from FPGA running receive data test and right window is NiosII command shell from FPGA running send data test.

Figure 8-4 shows the example of transfer data between two FPGAs by using jumbo frame size. When using jumbo frame size, performance is better than non-jumbo frame.

When some inputs are invalid, "Out-of-range input"/"Invalid input" is displayed and the operation is cancelled, as shown in Figure 5-5 - Figure 5-6 (same as FPGA<->PC test).

```
Input '3' to run receive data test
                                                  Input receiver parameter
+++ UDPIP Receive Mode +++                                                  1
Enter transfer size       : 1 - 0xFFFFFFFF => 0xFFFFFFFF
Enable data verification : [0] Disable [1] Enable => 1

Run test application on PC by following command
udpdatatest t 192.168.11.25 60000 4000 4294967295

Wait data from Target ...                                                   2
Send 0 Byte Recv 119 MByte                          Wait data from Target
Send 0 Byte Recv 227 MByte
Send 0 Byte Recv 335 MByte
     |            |                                                         5
Send 0 Byte Recv 4103 MByte
Send 0 Byte Recv 4211 MByte
Send 0 Byte Recv 4294 MByte
Receive data completed                                                      6

Total tx transfer size = 0 Byte
Total = 0[B] , Time = 36101[ms] , Transfer speed = 0[MB/s]

Total rx transfer size = 4294967295 Byte
Total = 4294[MB] , Time = 36101[ms] , Transfer speed = 118[MB/s]

--- UDP1GIP menu ---
```

```
Input '2' to run send data test
                                                  Input sender parameter
+++ UDPIP Send Mode +++                                                     3
Enter transfer size : 1 - 0xFFFFFFFF => 0xFFFFFFFF
Enter packet size   : 1 - 8972       => 1472
                                                                  Same value
                                                                  as receiver
Run test application on PC by following command
udpdatatest r 192.168.11.42 4000 60000 4294967295 1 1000

Press any key to start data sending ...                                     4
Start data sending
Send 119 MByte Recv 0 Byte
Send 239 MByte Recv 0 Byte
Send 358 MByte Recv 0 Byte                          Display transfer size every second
     |            |                                                         5
Send 3947 MByte Recv 0 Byte
Send 4067 MByte Recv 0 Byte                         Display performance after
Send 4187 MByte Recv 0 Byte                         finishing transferring all data
Send data complete                                                          6

Total tx transfer size = 4294967295 Byte
Total = 4294[MB] , Time = 35902[ms] , Transfer speed = 119[MB/s]

Total rx transfer size = 0 Byte
Total = 0[B] , Time = 35902[ms] , Transfer speed = 0[MB/s]

--- UDP1GIP menu ---
```

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

```
/cygdrive/e/Altera/18.0

+++ UDPIP Receive Mode +++
Enter transfer size       : 1 - 0xFFFFFFFF => 0xFFFFFFFF
Enable data verification : [0] Disable [1] Enable => 1

Run test application on PC by following command
udpdatatest t 192.168.11.25 60000 4000 4294967295

Wait data from Target ...
Send 0 Byte Recv 124 MByte
Send 0 Byte Recv 235 MByte
Send 0 Byte Recv 347 MByte
     |            |
Send 0 Byte Recv 4032 MByte
Send 0 Byte Recv 4144 MByte
Send 0 Byte Recv 4256 MByte
Receive data completed                                                      6

Total tx transfer size = 0 Byte
Total = 0[B] , Time = 34801[ms] , Transfer speed = 0[MB/s]

Total rx transfer size = 4294967295 Byte
Total = 4294[MB] , Time = 34801[ms] , Transfer speed = 123[MB/s]

--- UDP1GIP menu ---
```

```
/cygdrive/e/Altera/18.0

+++ UDPIP Send Mode +++
Enter transfer size : 1 - 0xFFFFFFFF => 0xFFFFFFFF
Enter packet size   : 1 - 8972       => 8972                                3
                                                                  Set jumbo
                                                                  frame size
Run test application on PC by following command
udpdatatest r 192.168.11.42 4000 60000 4294967295 1 1000

Press any key to start data sending ...
Start data sending
Send 124 MByte Recv 0 Byte
Send 248 MByte Recv 0 Byte
Send 372 MByte Recv 0 Byte
     |            |
Send 3970 MByte Recv 0 Byte                         Display performance after
Send 4094 MByte Recv 0 Byte                         finishing transferring all data
Send 4218 MByte Recv 0 Byte                                                 6
Send data complete

Total tx transfer size = 4294967295 Byte
Total = 4294[MB] , Time = 34614[ms] , Transfer speed = 124[MB/s]

Total rx transfer size = 0 Byte
Total = 0[B] , Time = 34614[ms] , Transfer speed = 0[MB/s]

--- UDP1GIP menu ---
```

Figure 8-4 Send data test by using jumbo frame

8.4 Full duplex Test

Select '4' to run full duplex test by two FPGAs transferring data in both directions at the same time and the same port number. User sets test parameters through NiosII command shell. Firmware in the demo is designed to start full duplex by server FPGA before client FPGA. The step to run the test is shown as follows.

1) On server NiosII command shell, set three parameters under full duplex test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   b) Input packet size: Unit of packet size is byte. Valid value is 1 – 8972. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   c) Input data verification mode: '0'-disable data verification, '1'-enable data verification sent from PC.
2) When all inputs are valid, "Wait data from Target …" is displayed.
3) On client NiosII command shell, set three parameters under full duplex test.
   a) Input transfer size: Unit of transfer size is byte. Valid value is 0x1 - 0xFFFF_FFFF. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input. This input must be same as transfer size input of server FPGA.
   b) Input packet size: Unit of packet size is byte. Valid value is 1 – 8972. The input is decimal unit when the input is only digit number. User can add "0x" as a prefix for hexadecimal input.
   c) Input data verification mode: '0'-disable data verification, '1'-enable data verification sent from PC.
4) When all inputs are valid, "Press any key to start data transfer .." is displayed. After that, user enters some key(s) to start full duplex test.
5) During transferring data, current transfer size is displayed on both NiosII command shells every second.
6) "Transfer data complete" is displayed on both NiosII command shells. Finally, total transfer size and performance are displayed on both NiosII command shells.

Figure 8-5 and Figure 8-6 show full duplex test when running by using non-jumbo frame and jumbo frame size. The left window is NiosII command shell of server FPGA while the right window is NiosII command shell of client FPGA.

Figure 8-5 Full duplex test when using non-jumbo frame

Figure 8-6 Full duplex test when using jumbo frame

9 Revision History

Revision	Date	Description
1.0	3-Mar-17	Initial version release
1.1	17-Aug-18	Update user interface by using JTAG UART
1.2	4-Dec-19	Add Cyclone10 GX board
1.3	6-Mar-20	Add CycloneV GT board