Dec

dg\_udp10gip\_cpu\_instruction.doc

1	$\cap u$		1
י 2	PC	Satun	1
2	21	IP Setting	2
	2.2	Speed and Frame Setting	2
	2.3	Power Option Setting	6
	2.4	Firewall Setting	7
3	Tes	t result using FPGA and PC	8
	3.1	Display UDPIP parameters	8
	3.2	Reset UDPIP parameters	9
	3.3	Send Data Test	. 11
	3.4	Receive Data Test	. 13
	3.5	Full duplex Test	. 15
4	Tes	t result when using two FPGAs	. 18
	4.1	Display UDPIP parameter	. 18
	4.2	Reset UDPIP parameters	. 19
	4.3	Send and Receive data Test (Half-duplex test)	. 21
	4.4	Full-duplex Test	. 23
5	Rev	vision History	. 25



# **UDP10G-IP with CPU Demo Instruction**

Rev2.1 12-Jul-23

# 1 Overview

This document provides an example of running the UDP10G-IP demo using two different test environments. The first environment involves one FPGA board that transfers UDP data to a PC running a test application for transferring UDP data via 10G Ethernet. However, the test performance in this test environment is limited by the resource of the PC.

In contrast, the second environment involves two FPGA boards that work together to transfer 10G Ethernet data. This configuration achieves the best performance for transferring UDP data via 10G Ethernet using UDP10G-IP.

The document is divided into several topics. Topic 2 explains how to set up the 10G Ethernet card on the PC to obtain optimal performance for data transfer when running the test in the first environment with the FPGA and PC. Topic 3 describes the example console and test results when running the test in the first environment. Finally, topic 4 provides an example console when running the test in the second environment, using tow FPGAs.

For more details on each topic, please refer to the following sections.



# 2 PC Setup

Before running demo, please check the network setting on PC. Below is an example of how to set up the 10G Ethernet card.

## 2.1 IP Setting

Networking       Sharing         Connect using:       10-Gb LAN connection         Image: Intel(R) Ethernet Server Adapter X520-2       1         Configure       Configure         This connection uses the following items:       Image: Option of the appropriate IP settings assigned automatically         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         Image: Option of the appropriate IP settings       Image: Option of the appropriate IP settings         I	📮 Local Area Connection 2 Properties 📃 💌	Internet Protocol Version 4 (TCP/IPv4) Properties
Connect using:       10-Gb LAN connection         Intel(R) Ethernet Server Adapter X520-2         Configure         This cgnnection uses the following items:         Image: Client for Microsoft Networks         Image: Npcap Packet Driver (NPCAP)         Image: Npcap Packet Driver (NPCAP)         Image: Npcap Packet Scheduler         Image: Nicrosoft Networks         Image: Nicrosoft Net	Networking Sharing	General
Configure         This connection uses the following items:         ✓       Client for Microsoft Networks         ✓       Npcap Packet Driver (NPCAP) (Wi-Fi)         ✓       Npcap Packet Driver (NPCAP)         ✓       QoS Packet Scheduler         ✓       File and Printer Sharing for Microsoft Networks         ✓       Internet Protocol Version 6 (TCP/IPv6)         ✓       Internet Protocol Version 4 (TCP/IPv4)         ✓       III	Connect using: 10-Gb LAN connection	You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
This connection uses the following items:         Image: Client for Microsoft Networks         Image: Client for Microsoft Networks         Image: Npcap Packet Driver (NPCAP) (Wi-Fi)         Image: Npcap Packet Driver (NPCAP)         Image: Npcacket Driver (NPCAP) <td< td=""><td><u>C</u>onfigure</td><td>Obtain an IP address automatically</td></td<>	<u>C</u> onfigure	Obtain an IP address automatically
✓       Client for Microsoft Networks         ✓       Npcap Packet Driver (NPCAP) (Wi-Fi)         ✓       Npcap Packet Driver (NPCAP)         ✓       Npcap Packet Driver (NPCAP)         ✓       Npcap Packet Scheduler         ✓       Pile and Printer Sharing for Microsoft Networks         ✓       Internet Protocol Version 6 (TCP/IPv6)         ✓       III	This connection uses the following items:	Use the following IP address:
Subnet mask:       255.255.255.0         Default gateway:       .         Internet Protocol Version 6 (TCP/IPv6)       .         Internet Protocol Version 4 (TCP/IPv4)       .         Image: Subnet mask:       .         Umage: Subnet mask:       .         Default gateway:       .         .       .         <	Client for Microsoft Networks	IP address: 192 . 168 . 7 . 25
Internet Protocol Version 6 (TCP/IPv6)         Internet Protocol Version 4 (TCP/IPv4)         Image: State of the sta	✓ ■ Npcap Packet Driver (NPCAP)	Subnet mask: 255 . 255 . 0
Internet Protocol Version 6 (TCP/IPv6) 2 Internet Protocol Version 4 (TCP/IPv4) * Use the following DNS server addresses:	<ul> <li>QoS Packet Scheduler</li> <li>File and Printer Sharing for Microsoft Networks</li> </ul>	Default gateway:
Use the following DNS server addresses:	Internet Protocol Version 6 (TCP/IPv6)     Internet Protocol Version 4 (TCP/IPv4)	Obtain DNS server address automatically
	۲ III ا	O Use the following DNS server addresses:
Install Uninstall Properties Preferred DNS server:	Install Uninstall Properties	Preferred DNS server:
Description Alternate DNS server:	Description	Alternate DNS server:
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.	Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.	Vajidate settings upon exit Advanced
OK Cancel	OK Cancel	OK Cancel

Figure 2-1 Setting IP address for PC

- 1) Open Local Area Connection Properties of the 10G Ethernet connection, as shown on the left window of Figure 2-1.
- 2) Select "TCP/IPv4" and click on Properties.
- 3) Set IP address = 192.168.7.25 and Subnet mask = 255.255.255.0, as shown on the right window of Figure 2-1.



# 2.2 Speed and Frame Setting

Figure 2-2 Set frame size = jumbo frame

- On Local Area Connection Properties window, click "Configure" as shown in Figure 2-2.
   On Advanced Tab, select "Jumbo Packet". Set Value to "9014 Bytes" for Jumbo Frame
- 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 on the bottom window of Figure 2-2.



3) On Link Speed, select "10 Gbps Full Duplex" for running 10-Gigabit transfer test, as shown in Figure 2-3.

el(R) Ethernet	: Server Adapte	r X520-2 Properti	es	-
Teaming	VLANs	Boot Options	Driver	Details
General	Advanced	Link Speed	PROSet	Advanced
Link Status • Speed:	Link Speed a Intel(R) PROS 10.00 0	nd Duplex Settings Get Version: 25.0.0. Gbps Full Duplex	1000	
Speed and Duplex: 10 Gbps Full Duplex Diagnostics				
Identify <u>A</u> dapter				
Speed and Duplex Setting: By default, Intel® adapters are set to automatically detect and negotiate speed and duplex settings. A setting other than Auto Negotiation restricts what the adapter advertises during auto-negotiation.         Image: Information icon         This icon is displayed when the device is not linked at its maximum capable speed. In that case, if your device is set to				
		E	OK (	Cancel
Figure	e 2-3 Set	link speed	d = 10 C	Sbps



- 4) On PROSet Advanced Tab, select "Performance Options" and click "Properties" button.
- 5) Set "Interrupt Moderation Rate" = OFF.

Intel(R) Ethernet Server Adapter X520-2 Properties	Performance Options
Teaming         VLANs         Boot Options         Driver         Details           General         Advanced         Link Speed         PROSet Advanced           Image: Constraint of the system of the	Settings: Direct Cache Access Flow Control Interrupt Moderation Rate Low Latency Interrupts Receive Buffers Transmit Buffers Use Default
Cattinger	Interrupt Moderation Rate
Locally Administered Address Log Link State Event Offloading Options Performance Uptions Receive Side Scaling Receive Side Scaling Queues Performance Options Configures the adapter to use settings that can improve adapter	This sets the rate at which the controller moderates or delays the generation of interrupts making it possible to optimize network throughput and CPU utilization. The Adaptive setting adjusts the interrupt rates dynamically depending on traffic type and network usage. Choosing a different setting may improve network and system performance in certain configurations.
performance.	

Figure 2-4 Interrupt Moderation Rate

- 6) Select "Low Latency Interrupts" and click "Properties" button.
- 7) On "Low Latency Interrupts" window, select "Use Low Latency Interrupts" and click "OK" button.
- 8) Click "OK" button to save and exit all setting windows.

Performance Options	Low Latency Interrupts	×
Settings: Direct Cache Access Flow Control Interrupt Moderation Rate We Latency Interrupts Low Latency Interrupts Enables adapters to bypass interrupt moderation and immediately generate an interrupt when certain TCP packets arrive, allowing the system to handle the packet more quickly. Certain applications will have faster access to network data because of the reduced	Use for packets with TCP PSH flag      Use for these TCP ports:      Add New Port:      Add      Remove	
data latency.           NOTES:           • If this ontion is enabled system CPI Lutilization           OK	Configures which packets bypass interrupt moderation and trigger immediate interrupts. Use for packets with TCP Any incoming packet with the PSH flag: TCP PSH flag will trigger an immediate interrupt. The PSH flag is set by the sending	< III
	device. Use for these TCP ports: Every packet received on the 8 OK Cancel	Ŧ
Figure 2-5 Use L	Low Latency Interrupts	



### 2.3 Power Option Setting

- 1) Open Control Panel and select Power Options as shown on the left window of Figure 2-6.
- 2) Change setting to High Performance as shown on the right window of Figure 2-6.

G v 🔄 + Control Panel + All Control Panel Items + + 4 Search Control	😋 🔵 🗢 😻 « All Control Panel	Items > Power Options + 47 Search Control Panel P
Adjust your computer's settings View by:	Control Panel Home	Select a power plan
	Require a password on wakeup	Power plans can help you maximize your computer's performance or conserve
Power Options Reatures	Choose what the power buttons do	changing its power settings. <u>Tell me more about power plans</u>
	😗 Create a power plan	Preferred plans
🕺 🕺 Recovery 🛛 🕺 Region and Language	Coose when to turn off the	Balanced (recommended)     Change plan settings
Development Devileer	d	Automatically balances performance with energy consumption on capable hardware.
Connections Sound	when the computer	2
Connections		High performance     Change plan settings
🔒 Speech Recognition 🛛 🛞 Sync Center		ravors performance, but may use more energy.
T. T		Show additional plans
😡 System 🛛 Taskbar and Start Menu		-
		🚱 Change settings that are currently unavailable
Troubleshooting		
	See also	
Windows Anytime Windows CardSpace	Personalization	
Upgrade 📢	User Accounts	
🕮 Windows Defender 🧀 Windows Firewall		
Figure	2-6 Power optio	ns



### 2.4 Firewall Setting



Figure 2-7 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.



# 3 Test result using FPGA and PC

#### 3.1 Display UDPIP parameters

Select '0' to check the current parameter in the demo. This will display seven parameters on the console.

	Current parameter
+++ Current Network Param Mode = CLIE FPGA MAC address = 0x00 FPGA IP = 192. FPGA port number = 4000 Target IP = 192. Target port number (Targe Target port number (FPGA-	eter +++ NT 0102030405 168.7.42 168.7.25 t->FPGA> = 61000 >Target> = 60000
UDP10G-IP menu [0] : Display UDPIP param [1] : Reset UDPIP paramet [2] : Send Data Test (UDP [3] : Receive Data Test ( [4] : Full duplex Test (U	eters er IP -> Target> Target -> UDPIP> DPIP <-> Target>
Figure 3-1 Display of	current parameter result

- 1) Mode: Set the initialization mode to UDP10G-IP to act as a Server or Client. To run with a PC, input '0' to initialize the IP in Client mode.
- 2) FPGA MAC address: This is a 48-bit hex value that serves as the MAC address of the FPGA. The default value is 0x000102030405.
- 3) FPGA IP: This is the IP address of the FPGA. The default value is 192.168.7.42. <u>Note</u>: This value is used to be the FPGA IP address, a parameter for the test application on the PC.
- 4) FPGA port number: This is the port number of the FPGA. The default value is 4000. <u>Note</u>: This value is used to be the FPGA port, a parameter for the test application on the PC.
- 5) Target IP: This is the IP address of the Target device, which is the 10G Ethernet on the PC. The default value is 192.168.7.25.
- 6) Target port number (Target->FPGA): This is the port number of the Target device to which UDP payload data will be sent from the PC to the FPGA. The default value is 61000. <u>Note</u>: This value is used to be the PC port, a parameter for test application on the PC in transmit mode
- Target port number (FPGA->Target): This is the port number of the Target device from which UDP payload data will be sent from the FPGA to the PC. The default value is 60000.

<u>Note</u>: This value is used to be the PC port, a parameter for the test application on the PC in receive mode

To change some parameters, the user can set them by using menu [1] (Reset UDPIP parameters).



#### 3.2 Reset UDPIP parameters

Select '1' to modify IP parameters or reset UDP10G-IP. Once selected, the current parameters are displayed, and the user can enter 'x' to keep the same parameters or modify individual parameters by entering a different value.

There are seven parameters that can be modified, as described in topic 3.1(Display UDPIP parameter). The range of each parameter is as follows.

<u>Note</u>: If the user sets the invalid input, the input will be rejected and the same value will be used for that parameter.

- 1) Mode : Input '0' to initialize IP as Client mode.
- 2) FPGA MAC address : Input a 12-digit of hexadecimal value. Add "0x" as a prefix to indicate a hex value.
- 3) FPGA IP address : Input a set of four decimal digits separated by ".". Each digit must be the range of 0-255.
- 4) FPGA port number : Input a value in the range of 0-65535.
- 5) Target IP address : Input a set of four decimal digits, similar to the FPGA IP address. This is the IP address of the PC.
- 6) Target port number (Target->FPGA)
- 7) Target port number (FPGA->Target)
- : Input a value in the range of 0-65535.
- : Input a value in the range of 0-65535.

Once all the parameters are entered, the new parameter set is displayed on the console. The reset signal is then asserted, and the IP begins initialization with the new parameters. Upon completion of the initialization process, the console displays "IP initialization complete", as shown in Figure 3-2.







Figure 3-2 Change IP parameter result



#### 3.3 Send Data Test

To transfer data from FPGA to PC, select '2' to run the Send data test on the FPGA console and run the "udpdatatest.exe" application to receive data via Command Prompt on the PC. The user inputs the test parameters for sending data on the FPGA console. The steps to run this menu are shown below.

- 1) On the FPGA console, input two parameters under Send data test menu.
  - i) Input transfer size: The unit of transfer size is byte. Valid values are 0x8 0xFFFF\_FF8 and the input must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
  - ii) Input packet size: The unit of packet size is byte. Valid values are 8 8968, and the value must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
     <u>Note</u>: If packet size is over 1472, the packet output will be a jumbo frame, and the PC must support this.
- 2) If all inputs are valid, the recommended parameters for running the test application on the PC will be displayed. "Press any key to start data sending ..." is displayed to begin sending data when user enters any key(s).
- 3) On the Command prompt, input the test parameters following the recommended values. There are five mandatory parameters and two optional parameters for executing "udpdatatest" to receive data.

>> udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [ByteLen] <Pattern> <Timeout>

#### Mandatory parameters

- i) Dir : Set 'r' to receive test data from FPGA
- ii) FPGA IP : Set the same value as FPGA IP address
- iii) FPGA port : Set the same value as FPGA port number
- iv) PC port : Set the same value as Target port number (FPGA->Target)
- v) Bytelen : Set the same value as "Input transfer size" of step 1)

#### **Optional parameters**

- i) Pattern : '1'- enable data verification, '0'-disable data verification. The default value is '1' which is applied when there is no input.
- ii) Timeout : Timeout in msec unit. Valid value is 50-65536. It is recommended to set 100 for 10G Ethernet. The default value is 100 which is applied when there is no input.
- 4) After running the test application, a summary of the setting parameters will be displayed and the application will wait for received data from FPGA.
- 5) On the FPGA console, input any key(s) to start sending data. The current number of transfer data size will be displayed on both the FPGA console (transmitted data size) and the Command prompt (receive data size) every second.
- 6) Once all data has been sent, "Send data complete" will be displayed on the FPGA. The test application on the PC will finish when either the total number of received data equals the set value (indicating no lost data) or a timeout when no new data is received is reached, resulting in an error message being displayed on the Command prompt. If data verification is enabled, the first error position will also be displayed. Finally, the total number of receive data and performance will be displayed on both the FPGA console (transmit performance) and the Command prompt (receive performance).



Figure 3-3 shows an example of Send data test using jumbo frame size.

FPGA console	
Input '2' to run send data test	
+++ UDP10G-IP Send Mode +++	
Enter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8	
inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 [Call udpdatatest.exe with recommended parameter and the second s	eters
Run test application on PC by following command	
udpdatatest r 192.168.7.42 4000 60000 4294967288 1	
(2)	
Press any key to start data sending Start data sending Start data sending Display setting par	meters
Brend 1240 MBute Recv 0 Bute Display recommended parameter and Provide 17 192.168.7.424000 () on PC, inputs from PC, post	user
Send 2480 MByte Recv 0 Byte (5) wait user input keys to start sending [Fineout 100 mSec	
Send 3720 MByte Recv 0 Byte period of a contract of the second se	
2.483 GB (5) Display transfer size every second	
Total tx transfer size = 536870911 64-bit	
[otal = 4294[MB], Time = 3463[ms], Transfer speed = 1240[MB/s] Spend 3.46 Second(s) for Receiving 4.295 GByte(s)	1
Receiving Data Rate 1239.89 MByte(s)Sec	
$\begin{bmatrix} 101a & rx & traisfer & size & -0 & 0 & rult \\ \hline 101a & 1 & men & -3463 $	
C:\SW> Display performance after	
UDP10C-IP nenu complete to transfer all data	
[0] : Display UDPIP parameters	
[2]: Send Data Test (UDPIP -> Target)	
[3] : Receive Data Test (Target -> UDPIP)	
[4] : Full duplex Test (UDPIP <-> Target)	

Figure 3-3 Send data test by using jumbo frame

If the input is invalid, "Out-of-range input"/"Invalid input" is displayed and then the operation is cancelled, as shown in Figure 3-4 - Figure 3-5.

+++ UDP10G-IP Send Mode +++
Enter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0x10000000
Dut-of-range input
--- UDP10G-IP menu --[0] : Display UDPIP parameters
[1] : Reset UDPIP parameter
[2] : Send Data Test (UDPIP -> Target)
[3] : Receive Data Test (UDPIP <-> Target)
[4] : Full duplex Test (UDPIP <-> Target)
Eigure 3-4 Error from invalid transfer size

```
+++ UDP10G-IP Send Mode +++
Inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8
Inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8969
Uut-of-range input
--- UDP10G-IP menu ---
[0] : Display UDPIP parameters
[1] : Reset UDPIP parameter
[2] : Send Data Test (UDPIP -> Target)
[3] : Receive Data Test (Iarget -> UDPIP)
[4] : Full duplex Test (UDPIP <-> Target)
```





#### 3.4 Receive Data Test

To received data on FPGA from PC, select '3' to run the Receive data test on FPGA and input the required parameters for receiving data on the FPGA console. On the PC, run "udpdatatest.exe" to send data with setting the test parameters of "udpdatatest" on the Command prompt. The steps to run this menu are shown below.

- 1) On the FPGA console, input two parameters under Receive data test menu.
  - i) Input transfer size: The unit of transfer size is byte. Valid values are 0x8 0xFFFF\_FF8 and the input must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
  - ii) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from the PC.
- 2) If inputs are valid, the recommended parameters for running the test application on the PC will be displayed. The message "Wait data from Target ..." is displayed to indicate that the FPGA is waiting for data to be received from the PC.
- 3) On the Command prompt, input the test parameters following the recommended values. There are five mandatory parameters and one optional parameter for executing "udpdatatest" to send data.

>> udpdatatest [Dir] [FPGAIP] [FPGAPort] [PCPort] [ByteLen] <pattern>

#### Mandatory parameters

- i) Dir : Set 't' to send test data from PC
- ii) FPGA IP : Set the same value as FPGA IP address
- iii) FPGA port : Set the same value as FPGA port number
- iv) PC port : Set the same value as Target port number (Target->FPGA)
- v) Bytelen : Set the same value as "Input transfer size" of step 1)

Optional parameters

- i) Pattern : '1'- Incremental pattern, '0'- dummy pattern. The default value is '1' which is applied when there is no input.
- 4) Once the test application is initiated, the data is sent from the PC to the FPGA. The current transfer size is displayed on both the FPGA console (receive data size) and the Command prompt (transmit data size) every second.
- 5) Upon completion of the data transfer, the FPGA console displays "Receive data completed". If no new data is received until timeout, the FPGA will also complete the operation by timeout condition. Finally, the total number of transfer data size and performance are displayed on both the FPGA console (receive performance) and the Command prompt (transmit performance).



Figure 3-6 shows an example of Receive data test when the data verification mode is disabled on the FPGA console. The left window displays the test results on the FPGA console, while the right window displays the test results on the Command prompt.

Figure 3-7 shows and example of Receive data test when the data verification mode is enabled on the FPGA console. The PC sends incremental data to the FPGA, and the left window displays the test result on the FPGA console, while the right window displays the test results on the Command prompt. If the verification module fails, an error message is displayed.





#### 3.5 Full duplex Test

Select '4' to run full duplex test which transfers data between FPGA and PC in both directions simultaneously. User inputs test parameters on the FPGA console and PC Command prompt. Two "udpdatatest" applications must be executed by the user on two separate Command prompts using different port number, one for sending data and another for receiving data. The steps to run the test are as follows.

1) On the FPGA console, input three parameters under Full duplex test.

- i) Input transfer size: Unit of transfer size is byte. Valid value is 0x8 0xFFF\_FF8". The input must be aligned to 8. The input is decimal unit when input only digit number. User adds "0x" to be a prefix for hexadecimal unit. This value must be equal to total transfer size, set on test application.
- ii) Input packet size: Unit of packet size is byte. Valid value is 8 8968. The input must be aligned to 8. The input is decimal unit when input only digit number. User adds "0x" to be a prefix for hexadecimal unit.
- iii) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from the PC.
- 2) If all inputs are valid, two recommended parameter sets to run two applications on the PC are displayed, sending data and receiving data. "Press any key to start data sending ..." is displayed to begin the operation when user enters any key(s).
- 3) Open two Command prompts to execute the applications using the recommended parameter values. Five mandatory parameters and one optional parameter for executing "udpdatatest" are displayed.

<u>Receive data (The 1<sup>st</sup> command prompt)</u> >> udpdatatest r [FPGAIP] [FPGAPort] [PCPort] [ByteLen] <Pattern> <Timeout>

#### Send data (The 2<sup>nd</sup> command prompt)

>> udpdatatest t [FPGAIP] [FPGAPort] [PCPort] [ByteLen] <Pattern>

#### Mandatory parameters

- i) Dir : The 1<sup>st</sup> command prompt : Set 'r' to receive data The 2<sup>nd</sup> command prompt : Set '1' to send data
   ii) FRCA IR = 1 Set the same value as FRCA IR addression (Section 1)
- ii) FPGA IP : Set the same value as FPGA IP address
- iii) FPGA port : Set the same value as FPGA port number
- iv) PC port

The 1<sup>st</sup> command prompt: Set the same value as Target port number (FPGA->Target) The 2<sup>nd</sup> command prompt: Set the same value as Target port number (Target->FPGA)

v) ByteLen : Set the same value as "Input transfer size" of step 1)

#### Optional parameters

- i) Pattern : '1'- enable data verification, '0'-disable data verification. The default value is '1' which is applied when there is no input.
- ii) Timeout : Timeout in msec unit. Valid value is 50-65536. It is recommended to set 100 for 10G Ethernet. The default value is 100 which is applied when there is no input.



- 4) On the FPGA console, press any key(s) to initiate data transmission to the PC. The FPGA console and two Command prompts will display the current number of data transfers in both directions every second.
- 5) Once UDP10G-IP has sent and received all data, the FPGA console will display "Transfer data complete". Finally, the total transfer size and performance will then be displayed on the FPGA console and the Command prompts.

Figure 3-8 displays the transfer performance during full-duplex operation with data verification. The left window is the test result on FPGA console while the right window is the test result on the Command prompts: the upper window shows receive performance and the lower window shows transmit performance.

Figure 3-9 shows an example result when lost data is detected on the PC. In this scenario, the receive software is halted due to a timeout condition, where no new data is received until timeout value is reached. The application displays an error message indicating the first lost position and the total amount of lost data.



Figure 3-8 Full-duplex test when no lost data



	-
FPGA conso	le

PC Command Prompt

Figure 3-9 Full-duplex test when lost data is detected on PC



# 4 Test result when using two FPGAs

#### 4.1 Display UDPIP parameter

Select '0' to check current parameter in the demo. This will display seven parameters on the console.

	Current parameter on server			Current para	meter on client	
+++ Curre Mode FPGA MAC FPGA IP FPGA port Target II Target po Target po	ent Network Parameter +++ = SERUER address = 0x001122334455 = 192.168.7.25 t number = 60000 P = 192.168.7.42 ort number (Target->FPGA) = 46 ort number (FPGA->Target) = 46	+ FFF 00 00	H++ Cur lode PGA MA PGA IP PGA po arget arget arget	rent Networ C address rt number IP port number port number	k Parameter ++ = CLIENT = 0×000102030 = 192.168.7.4 = 4000 = 192.168.7.2 < (Target->FPGA < (FPGA->Target	++ 3405 42 25 3) = 61000 t) = 60000
UDP10 [0] : Dis [1] : Res [2] : Sen [3] : Rec [4] : Fu]	0G-IP menu splay UDPIP parameters set UDPIP parameter nd Data Test (UDPIP -> Target) ceive Data Test (Target -> UD) 11 duplex Test (UDPIP <-> Targ	IP) et)	UDP: [0] : D [1] : R [2] : S [3] : R [4] : F	10G-IP menu isplay UDPI eset UDPIP end Data Te eceive Data ull duplex	P parameters parameter st (UDPIP -> 1 Test (Target Test (UDPIP <-	[arget) -> UDPIP) -> Target)

Figure 4-1 Display current parameter result

- 1) Mode: Set the initialization mode to UDP10G-IP to act as a Server or Client. Input '0' for Client and '1' for Server.
- 2) FPGA MAC address: This is a 48-bit hex value that serves as the MAC address of the FPGA. The default value is 0x000102030405 for Client and 0x001122334455 for Server.
- 3) FPGA IP: This is the IP address of the FPGA. The default value is 192.168.7.42 for Client and 192.168.7.25 for Server.
- 4) FPGA port number: This is the port number of the FPGA. The default value is 4000 for Client and 60000 for Server. <u>Note</u>: This value is used to be the FPGA port, a parameter for the test application on the PC.
- 5) Target IP: This is the IP address of the Target device. The default value is 192.168.7.25 for Client and 192.168.7.42 for Server.
- 6) Target port number (Target->FPGA): This is the port number of the Target device to receive data from the Target. The default value is 61000 for Client and 4000 for Server.
- 7) Target port number (FPGA->Target): This is the port number of the Target device to send data to the Target. The default value is 60000 for Client and 4000 for Server.

To change some parameters, the user can set them by using [1] (Reset UDPIP parameters).

<u>Note</u>: When running a test using two FPGA boards, it is important to ensure that the parameters of both FPGA boards are matched. The Target parameters of the first board must be equal to the FPGA parameters of the second board, and vice versa



#### 4.2 Reset UDPIP parameters

Select '1' to modify IP parameters or reset UDP10G-IP. Once selected, the current parameters will be displayed on the console. The user can enter 'x' to keep the same parameters or modify individual parameters by entering a different value.

There are seven parameters that can be modified, as described in topic 4.1(Display UDPIP parameter). The range of each parameter is as follows.

#### Note:

- 1. If the user sets the invalid input, the input will be rejected and the same value will be used for that parameter.
- 2. When resetting parameters on the Server FPGA, the Client FPGA must also be reset. The Server should be reset first and wait for an ARP request sent from the Client.
- 3. The Target parameters of the first board must be equal to the FPGA parameters of the second board, and vice versa.
- 4. When using two FPGA board test, the Target port number for Target->FPGA and FPGA->Target must be the same value, which should be equal to the FPGA port number of the other board.
- 1) Mode : Input '0' for Client or '1' for Server to determine FPGA initialization mode. It needs to set the different mode for two FPGA boards. One board is client and another board is server.
- 2) FPGA MAC address : Input a 12-digit of hexadecimal value. Add "0x" as a prefix to indicate a hex value.
- 3) FPGA IP address : Input a set of four decimal digits separated by ".". Each digit must be the range of 0-255.
- 4) FPGA port number : Input a value in the range of 0-65535.
- 5) Target IP address : Input a set of four decimal digits. Use the same value as FPGA IP address on another board.
- 6) Target port number (Target->FPGA)
- : Input a value in the range of 0-65535.
- 7) Target port number (FPGA->Target)
- : Input a value in the range of 0-65535.

Once all the parameters are entered, the new parameter set is displayed on the console. The reset signal is then asserted, and the IP begins initialization with the new parameters. Upon completion of the initialization process, the console displays "IP initialization complete", as shown in Figure 4-2.



Client

# Server



Figure 4-2 Change IP parameter result



#### 4.3 Send and Receive data Test (Half-duplex test)

To execute a half-duplex test with two FPGA boards, one board runs the Receive data test (menu 3), while another board runs the Send data test (menu 2). The user inputs the test parameters on the FPGA console. The steps to run the test are as follows.

- 1) Under menu 3 (Receive data test) on the FPGA console, input two parameters.
  - i) Input transfer size: The unit of transfer size is byte. Valid values are 0x8 0xFFFF\_FF8 and the input must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
  - ii) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from another FPGA.
- 2) If inputs are valid, the message "Wait data from Target ..." is displayed, indicating that the FPGA is waiting for data to be received from another FPGA.
- 3) Under menu 2 (Send data test) on another FPGA console, input two parameters.
  - i) Input transfer size: The unit of transfer size is byte. Valid values are 0x8 0xFFFF\_FF8 and the input must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix. *Note: This value must be equal to the transfer size set in step 1*).
  - ii) Input packet size: The unit of packet size is byte. Valid values are 8 8968, and the value must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
     <u>Note</u>: If the packet size is more than 1472, the packet output from UDP10G-IP is a jumbo frame. The user needs to confirm that the network device supports jumbo frames when two FPGA boards are connected through the network device.
- 4) If all inputs are valid, the message "Press any key to start data sending ..." is displayed, indicating that the data transfer will begin when the user enters any key(s).
- 5) The user enters any key(s) to start data sending, and then the data starts transferring. During the transfer, the current number of transfer data size is displayed on both FPGA consoles every second.
- 6) When all data has been sent and received, "Send data complete" is displayed on the FPGA console that runs Send data test, and "Receive data completed" is displayed on the FPGA console that runs Receive data test. Finally, the total transfer size and performance are displayed on both FPGA consoles.

Figure 4-3 shows an example to data transfer between two FPGAs using non-jumbo frame size. The left window shows the FPGA console from FPGA running the Receive data test, and the right window shows the FPGA console from FPGA running the Send data test.

Figure 4-4 shows an example of data transfer between two FPGAs using jumbo frame size, which provides better performance, compared to non-jumbo frame size.

If the user input is invalid, the console displays an error message "Out-of-range input" or "Invalid input", and the operation is cancelled, as shown in Figure 3-4 - Figure 3-5, similar to FPGA<->PC test.



Input '3' to run receive data test	<ul> <li>♦ : User Input</li> <li>Input '2' to run send data test</li> <li>♦ : User Output</li> </ul>					
+++ UDP18G-IP Receive Mode +++	1 t++ UDP10G-IP Send Mode +++					
Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFF8 =>  <u>0xFFFFF8</u> Enable data verification : [0] Disable [1] Enable =>  <u>1</u>	nter transfer size in byte unit (aligned to 64-bit) : 8 - 6xFFFFFF8 =) [8xFFFFFF8 = inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => [472]					
Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1	Run test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1					
Ait data from Target Wait data from Target	Press any key to start data sending 4					
Send Ø Byte Recv 2265 MByte 5 Send Ø Byte Recv 3337 MByte 5 Receive data completed	Image: Send 1191 MByte Recv 0 Byte       5       Display transfer size every second         Image: Send 3575 MByte Recv 0 Byte       5					
fotal tx transfer size = 0 64-bit	Send data complete					
[otal = 0[B], Time = 3604[ms], Transfer speed = 0[MB/s]	otal tx transfer size = 536870911 64-bit otal = 4294[MB] , Time = 3603[ms] , Transfer speed = 1179[MB/s]					
otal rx transfer size = 536870911 64-Dit Total = 4294[MB] , Time = 3604[ms] , Transfer speed = 1191[MB/s]	[otal rx transfer size = 0 64-bit [otal = 0[B] , Time = 3603[ms] , Transfer speed = 0[MB/s]					
UDP10G-IP menu [0] : Display UDPIP parameters [1] : Pacet UDPIP parameters						
2] : Send Data Test (UDPIP -> Target)	11 : Reset UDPIP parameter					
[3] : Receive Data Test (Target -> UDPIP) [4] : Full duplex Test (UDPIP <-> Target>	[2] : Send Data lest (UDFIF -> larget) [3] : Receive Data Test (Target -> UDPIP)					
Figure 4-3 Send/receive dat	<u>141 : Full dunlex Test (IDPIP &lt;-&gt; Target&gt;</u>					
Figure 4-3 Send/receive data test by using non-jumbo frame size						
Input '3' to run receive data test	Input '2' to run send data test					
Input '3' to run receive data test	Input '2' to run send data test					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : (0) Disable (1) Enable => 1	Input '2' to run send data test +++ UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xPFFFFFF8 => 0xPFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 3					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1	Input '2' to run send data test +++ UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Set jumbo frame size					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 40800 4294967288 1 Hait data from Target	Input '2' to run send data test +++ UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8958 Run test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending Set jumbo frame size					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data from Target Send 0 Byte Recu 21240 MByte Send 0 Byte Recu 21240 MByte	Input '2' to run send data test +++ UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending Start data sending lend 1240 MByte Recv 0 Byte 1 2400 MByte Recv 0 Byte					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data from Target Send 0 Byte Recv 1240 MByte Send 0 Byte Recv 2357 MByte Send 0 Byte Recv 3473 MByte	Input '2' to run send data test ++ UDP10C-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending tart data sending end 1240 MByte Recv 0 Byte end 2240 MByte Recv 0 Byte end 3220 MByte Recv 0 Byte computed to transfer all data					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1 lait data from Target Send 0 Byte Recv 1240 MByte Send 0 Byte Recv 3473 MByte Receive data completed (6)	Input '2' to run send data test ++ UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending Start data sending end 1240 MByte Recv 0 Byte end 2480 MByte Recv 0 Byte end 3720 MByte Recv 0 Byte end data complete Display performance after complete to transfer all data					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data from Target Send 0 Byte Recu 1240 MByte Send 0 Byte Recu 2357 MByte Receive data completed Total tx transfer size = 0 64-bit Total = 0[B], Time = 3463[ns], Transfer speed = 0[MB/s]	Input '2' to run send data test +++ UDP10G-IP Send Mode ++++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 un test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending tart data sending lend 1240 MByte Recv 0 Byte lend 3720 MByte Recv 0 Byte lend data complete Inter size = 536870911 64-bit otal t x transfer size = 536870911 64-bit otal = 4294[MB], Time = 3463[ms], Transfer speed = 1227[MB/s]					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data from Target Send 0 Byte Recv 1240 MByte Send 0 Byte Recv 2357 MByte Receive data completed Total tx transfer size = 0 64-bit Total tx transfer size = 0 64-bit Total rx transfer size = 536870911 64-bit Total = 4294[MB], Time = 3463[ms], Transfer speed = 1240[MB/s]	Input '2' to run send data test ++ UDP10C-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending Start data sending tend 2490 MByte Recv 0 Byte tend 3720 MByte Recv 0 Byte tend data complete Iotal tx transfer size = 536870911 64-bit otal = 42941MB1, Time = 34631ms1, Transfer speed = 12271MB/s1 Iotal rx transfer size = 0 64-bit otal = 0[B], Time = 34631ms1, Transfer speed = 0(MB/s]					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Aun test application on PC by following connand udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data fron Iarget Send 0 Byte Recv 1240 MByte Send 0 Byte Recv 3257 MByte Send 0 Byte Recv 3257 MByte Receive data completed [6] Iotal tx transfer size = 0 64-bit Iotal = 0[B], Time = 3463[ms], Transfer speed = 0[MB/s] Iotal rx transfer size = 536870911 64-bit Iotal = 4294[MB], Time = 3463[ms], Transfer speed = 1240[MB/s]	Input '2' to run send data test +++ UDP10G-IP Send Mode ++++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 un test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending tart data sending lend 1240 MByte Recv 0 Byte lend 2480 MByte Recv 0 Byte lend 3720 MByte Recv 0 Byte lend data complete lotal tx transfer size = 536870911 64-bit otal = 4294[MB], Time = 3463[ms], Transfer speed = 1227[MB/s] lotal rx transfer size = 0 64-bit otal = 0[B], Time = 3463[ms], Transfer speed = 0(MB/s] UDP10G-IP menu					
Input '3' to run receive data test +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following connand udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data fron Target Send 0 Byte Recv 1240 MByte Send 0 Byte Recv 2357 MByte Send 0 Byte Recv 3473 MByte Receive data completed Total tx transfer size = 0 64-bit Total tx transfer size = 0 64-bit Total = 0[B], Time = 3463[ms], Transfer speed = 0[MB/s] Total rx transfer size = 536870911 64-bit Total = 4294[MB], Time = 3463[ms], Transfer speed = 1240[MB/s] UDP10G-IP menu [0] : Display UDPIP parameters [1] : Reset UDPIP parameters	Input '2' to run send data test ++ UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending Kart data sending end 1240 MByte Recv 0 Byte end 2480 MByte Recv 0 Byte Lend 2480 MByte Recv 0 Byte Lotal tx transfer size = 536870911 64-bit lotal = 4294[MB], line = 3463[ms], Transfer speed = 1227[MB/s] Lotal rx transfer size = 0 64-bit lotal = 0[B], Time = 3463[ms], Transfer speed = 0[MB/s] 					
Input '3' to run receive data test  +++ UDP10G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1  Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1  Vait data fron Target Send 0 Byte Recu 1240 MByte Send 0 Byte Recu 2357 MByte Receive data completed  for Iotal tx transfer size = 0 64-bit Iotal = 0(BI), Time = 3463(ms], Transfer speed = 0(MB/s] Iotal rx transfer size = 536878911 64-bit Iotal = 4294(MB), Time = 3463(ms], Transfer speed = 1240(MB/s]  UDP10G-IP menu [0] : Display UDPIP parameters [1] : Reset UDPIP parameter [2] : Send Data lest (UDPIP -> Target) [3] : Beacing math Target (Target) [3] : Beacing math	Input '2' to run send data test *+* UDP10C-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8958 Aun test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending Kart data sending tend 2400 MByte Recv 0 Byte tend 2480 MByte Recv 0 Byte tend 3220 MByte Recv 0 Byte tend data complete Otal tx transfer size = 536870911 64-bit otal = 42941MB], Time = 34631ms], Transfer speed = 12271MB/s] otal rx transfer size = 0 64-bit otal = 01B], Time = 34631ms], Transfer speed = 01MB/s] UDP10G-IP menu B1 : Display UDPIP parameters 11 : Reset UDPIP parameters 21 : Send Data Test (UDPIP -> Target)					
Input '3' to run receive data test  +++ UDP18G-IP Receive Mode +++ Enter transfer size in byte unit (aligned to 64-bit): 8 - 0xFFFFFFF8 => 0xFFFFFF8 Enable data verification : [0] Disable [1] Enable => 1 Run test application on PC by following command udpdatatest t 192.168.7.25 60000 4000 4294967288 1 Nait data from Target Eend 0 Byte Recu 1240 MByte Send 0 Byte Recu 2357 MByte Receive data completed  for a stransfer size = 0 64-bit Total = 0EBI , Time = 3463Ins] , Transfer speed = 0[MB/s] Total rx transfer size = 536870911 64-bit Total = 4294IMBI , Time = 3463Ins] , Transfer speed = 1240[MB/s]  UDP18G-IP menu [0] : Display UDPIP parameters [1] : Reset UDPIP parameters [2] : Send Data Test (Impressed = 2000) [3] : Receive Data Test (Impressed) [4] : Full duplex Test (UDPIP (-> Target)] [4] : Full duplex Test (UDPIP (-> Target)] [5] : Receive Data Test (UDPIP (-> Target)] [5] :	Input '2' to run send data test ++* UDP10G-IP Send Mode +++ inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8 inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 8968 un test application on PC by following command udpdatatest r 192.168.7.42 4000 60000 4294967288 1 Press any key to start data sending tart data sending end 1240 MByte Recv 0 Byte end 3220 MByte Recv 0 Byte end 3220 MByte Recv 0 Byte end data complete of fotal tx transfer size = 536870911 64-bit total = 4294(MB], Time = 3463[ms], Transfer speed = 1227(MB/s] otal rx transfer size = 0 64-bit total = 01B], Time = 3463[ms], Transfer speed = 0(MB/s] 					

Figure 4-4 Send/receive data test by using jumbo frame size



#### 4.4 Full-duplex Test

Select '4' to run full-duplex test on two FPGA boards for transferring data in both directions simultaneously. The user inputs the test parameters on the FPGA console. The steps to run the test are as follows.

- 1) On the Server console, input three parameters.
  - i) Input transfer size: The unit of transfer size is byte. Valid values are 0x8 0xFFFF\_FF8 and the input must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
  - ii) Input packet size: The unit of packet size is byte. Valid values are 8 8968, and the value must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
  - iii) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from the Client FPGA.
- 2) If inputs are valid, the message "Wait data from Target ..." is displayed, indicating that the FPGA is waiting for data to be received from another FPGA.
- 3) On the Client console, input three parameters.
  - i) Input transfer size: The unit of transfer size is byte. Valid values are 0x8 0xFFFF\_FF8 and the input must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix. *Note: This input must be set by the same value set in step 1*).
  - ii) Input packet size: The unit of packet size is byte. Valid values are 8 8968, and the value must be aligned to 8. The input is in decimal unit when only digit number is entered. If entering a hexadecimal unit, add "0x" as a prefix.
  - iii) Input data verification mode: Set '0' to disable data verification or '1' to enable data verification sent from the Server FPGA.
- 4) If inputs are valid, the message "Press any key to start data sending ..." is displayed, indicating that the data transfer will begin when the user enters any key(s). The user enters any keys to start full-duplex test.
- 5) The data starts transferring, and the current number of transferred data is displayed on both FPGA consoles every second.
- 6) Once all data has been transferred, the console displays "Transfer data complete" on both FPGAs. Finally, the total number of transferred data and performance are displayed on both FPGA consoles.

Figure 4-5 and Figure 4-6 shows the results when running full-duplex using non-jumbo frame size and jumbo frame size, respectively. The left window shows the Server console while the right window shows the Client console. Similar to the half-duplex test, using jumbo frame size shows better performance than using non-jumbo frame size



	<ul> <li>◆ : User Input</li> <li>◆ : User Output</li> </ul>
Server	Client
+++ UDP10G-IP Full-duplex Mode +++       Input server parameter         Inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFF8 => 0xFFFFFF8         inter packet size in byte unit (aligned to 64-bit) : 8 - 8968 => 1472         inable data verification       : [0] Disable [1] Enable => 1	Input client parameter         Inter transfer size in hyte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8 => 0xFFFFFF8         Inter packet size in hyte unit (aligned to 64-bit) : 8 - 8968 => 1472         Inable data verification       : [0] Disable [1] Enable => 1
Run two test application on PC by following command	Run two test application on PC by following command
l) PC receive data test (UDPIP -> PC) udpdatatest r 192.168.7.25 60000 4000 4294967288 1	) PC receive data test (UDPIP -> PC) udpdatatest r 192.168.7.42 4000 60000 4294967288 1
2) PC send data test (PC -> UDPIP) udpdatatest t 192.168.7.25 60000 4000 4294967288 1	2) PC send data test (PC -> UDPIP) udpdatatest t 192.168.7.42 4000 60000 4294967288 1
lait data fron Iarget     2     Wait data from Target       start data transferring     2     Wait data from Target       end 1191 MByte Recv 1191 MByte     5       end 3337 MByte Recv 3337 MByte     5       ransfer data complete     6	Press any key to start data transfer tart data transferring lend 1191 MByte Recv 1192 MByte lend 2264 MByte Recv 2265 MByte lend 3336 MByte Recv 3338 MByte ransfer data complete
lotal tx transfer size = 536870911 64-bit lotal = 4294[MB] , Time = 3603[ms] , Transfer speed = 1179[MB/s]	otal tx transfer size = 536870911 64-bit otal = 4294[MB] , Tine = 3604[ms] , Transfer speed = 1191[MB/s]
lotal rx transfer size = 536870911 64-bit lotal = 4294[MB] , Time = 3603[ms] , Transfer speed = 1179[MB/s]	otal rx transfer size = 536870911 64-bit otal = 4294[MB] , Time = 3604[ms] , Transfer speed = 1191[MB/s]
UDP10G-IP menu 10] : Display UDPIP parameters 11 : Reset UDPIP parameter 22] : Send Data Test (UDPIP -> Target) 13] : Receive Data Test (Target -> UDPIP) 14] : Full duplex Test (UDPIP <-> Target)	
Figure 4-5 Full-duplex test v	when using non-jumbo frame size
Server	Client
+++ UDP10G-IP Full-duplex Mode +++ Input jumbo frame inter transfer size in byte unit (aligned to 64-bit) : $8 - 0xFFFFFF8 => 0xFFFFFF8$ inter packet size in byte unit (aligned to 64-bit) : $8 - 8968 => 18968$ inable data verification : [0] Disable [1] Enable => 1	Input jumbo frame       inter transfer size in byte unit (aligned to 64-bit) : 8 - 0xFFFFFFF8       inter packet size in byte unit (aligned to 64-bit) : 8 - 8968       inter packet size in byte unit (aligned to 64-bit) : 8 - 8968       inable data verification       inable data verification
Run two test application on PC by following command	Run two test application on PC by following command
l) PC receive data test (UDPIP -> PC) Idpdatatest r 192.168.7.25 60000 4000 4294967288 1	.> PC receive data test (UDPIP -> PC) Idpdatatest r 192.168.7.42 4000 60000 4294967288 1
2) PC send data test (PC -> UDPIP) Idpdatatest t 192.168.7.25 60000 4000 4294967288 1	2) PC send data test (PC -) UDPIP) Idpdatatest t 192.168.7.42 4000 60000 4294967288 1
lait data fron Target tart data transferring Send 1240 MByte Recv 1240 MByte Send 2356 MByte Recv 2356 MByte Send 3472 MByte Recv 3472 MByte Transfer data complete	Press any key to start data transfer tart data transferring lend 1249 MByte Recv 1241 MByte lend 2356 MByte Recv 2357 MByte lend 3472 MByte Recv 3473 MByte ransfer data complete (6)
lotal tx transfer size = 536870911 64-bit lotal = 4294[MB] , Time = 3463[ms] , Transfer speed = 1227[MB/s]	lotal tx transfer size = 536870911 64-bit lotal = 4294[MB] , Time = 3463[ms] , Transfer speed = 1240[MB/s]
lotal rx transfer size = 536870911 64-bit lotal = 4294[MB] , Time = 3463[ms] , Transfer speed = 1227[MB/s]	lotal rx transfer size = 536870911 64-bit lotal = 4294[MB] , Time = 3463[ms] , Transfer speed = 1240[MB/s]
UDP10G-IP nenu 10] : Display UDP1P parameters 11] : Reset UDP1P parameter 12] : Send Data Test (UDP1P -> Target) 13] : Receive Data Test (Target -> UDP1P) 14] : Full dumler Test (UDP1P (-> Target)	UDP10G-IP nenu [0] : Display UDP1P parameters [1] : Reset UDP1P parameter [2] : Send Data Test (UDP1P -> Target) [3] : Receive Data Test (Target -> UDP1P) [4] : Review Data Test (Target -> UDP1P)

Figure 4-6 Full-duplex test when using jumbo frame size



# 5 Revision History

Revision	Date	Description
2.1	9-Mar-23	Update performance
2.0	21-Aug-20	Remove hardware setup from the document
1.1	8-Mar-19	Support FPGA <-> FPGA test and ZCU102
1.0	15-Sep-17	Initial version release