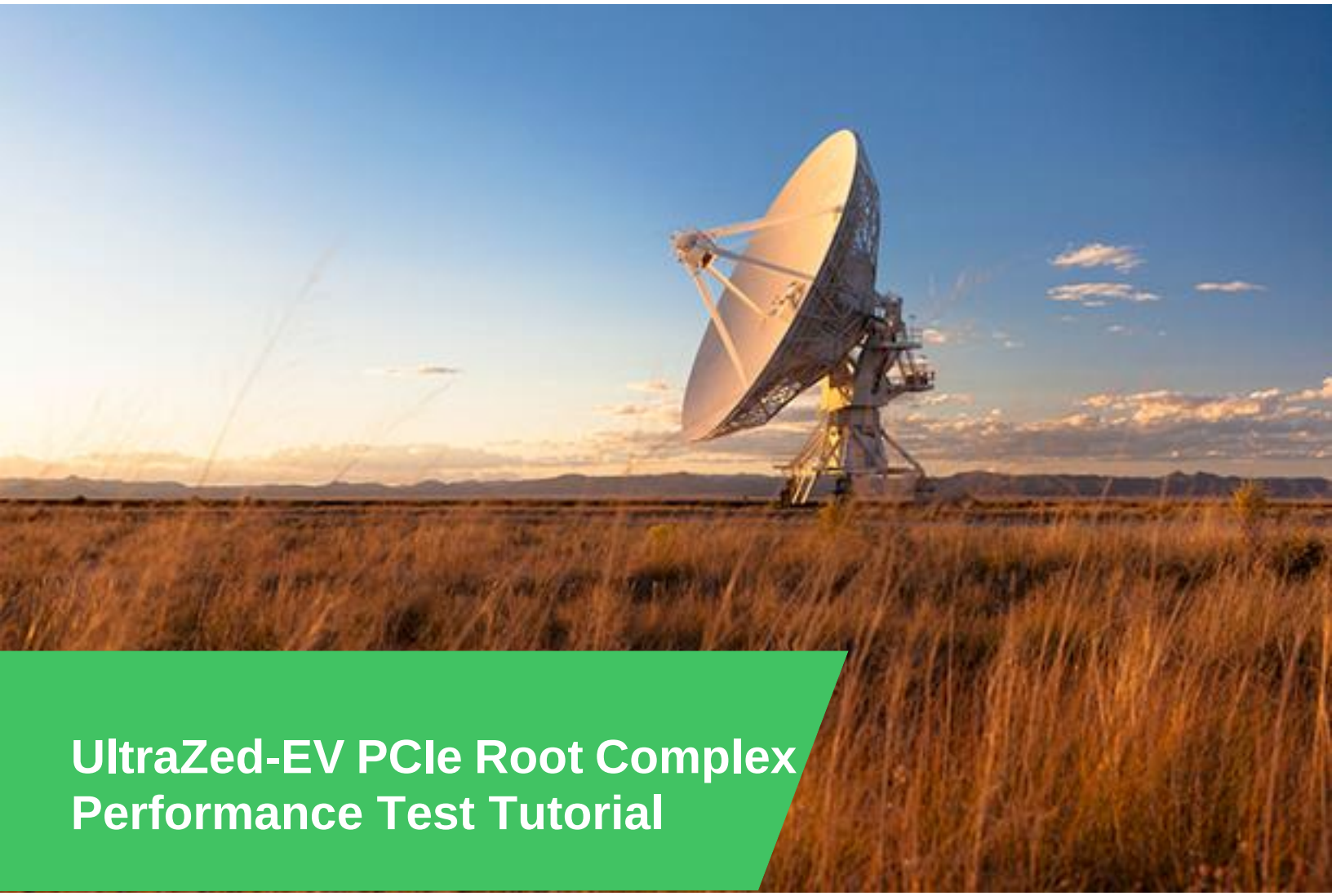


UltraZed-EV PCIe Root Complex Performance Test Tutorial

Introduction

This document describes a Zynq UltraScale+ PCIe Root Complex design implemented and tested on the Avnet UltraZed-EV SOM + EV Carrier development board.

The Zynq UltraScale+ (ZU+) All Programmable System on Chip (SoC) includes the serial transceivers and an Integrated Block for PCI Express that can be configured as an Endpoint or Root Port, compliant to the PCI Express Base Specification Revision 2.1. The Root Port can be used to build the basis for a compatible Root Complex, to allow custom communication between the ZU+ SoC and other devices via the PCI Express protocol, and to attach ASSP Endpoint devices such as Ethernet Controllers or Wireless Adapters to the ZU+ SoC. This example describes a PCIe Root Complex System on an Avnet UltraZed-EV platform with the existing Xilinx IPs and standard Linux software drivers. Connectivity with an off the shelf Ethernet NIC endpoint card is demonstrated with this design.

The Zynq UltraScale+ (ZU+) SoC integrates a quad-core ARM Cortex-A53 based processing system (PS) and Programmable Logic (PL) in a single device. This example design demonstrates the ZU+ device being used as PCI Express Root Complex System and it shows the following components working together on UltraZed-EV development board platform.
Zynq UltraScale+ Processing System with integrated PCIe Root Complex interfacing with

- External 4GB DDR4 memory
- Software drivers (installed on Linux running on the UltraZed) to enumerate and exercise a PCI Express Endpoint connected to the Root Complex System.

The intent of this design is to provide basic use cases for customers to build their own applications. The example will be a building block for designs which use the PCI Express link for passing control/data from Zynq UltraScale+ PS to a PCIe endpoint.

DISCLAIMER: This tutorial is provided for reference/educational purposes only and may not reflect results observed with other test equipment.

There are a number of factors which can impact Ethernet network performance and throughput in addition to transmission overheads, including latency, packet size, and system caching such that the calculated throughput typically does not reflect the maximum achievable throughput. As a result, the throughput over the Ethernet network can be substantially lower than the theoretical limits.

Design Objectives

This UltraZed tutorial offers system developers an example of how to:

- Target a prebuilt Xilinx release of PetaLinux to UltraZed
- Launch PCIe and Ethernet performance tests on Zynq UltraScale+ platform using a test script running a prebuilt open source Linux build created with Xilinx PetaLinux Tools

Experiment Setup

This tutorial builds upon the concepts and lab activities of the Avnet UltraZed Tutorials which cover the use of Xilinx Vivado Design Suite in creating/testing a basic Zynq UltraScale+ MPSoC hardware platform and running software applications. Please refer back to this reference material on the UltraZed community website for further information on how to configure the underlying UltraZed hardware platform.

The experiments in this tutorial use the following Linux applications:

iperf3 - This utility measures maximum achievable throughput on TCP/IP networks

For the example PCIe and Ethernet test configuration that was used in this tutorial, see the section Appendix I: Troubleshooting Ethernet Connection later in this document for further information.

The instructions in this tutorial assume that the cross build platform is an Ubuntu 16.04 LTS installation running in a virtual machine on an x86-based host. Though other systems may work using the same or similar instructions, those systems are not supported.

Example Design Requirements

Software

The software used to test this reference design is:

- Ubuntu 16.04 LTS 64-bit Desktop
    - VirtualBox v5.1.30 virtual machine
    - Windows-7 64-bit host OS
- Xilinx Vivado Design Suite 2017.4 (Design Edition)
- Xilinx PetaLinux 2017.4
- Git SCM toolset (version used for this tutorial is v2.7.4)

Hardware

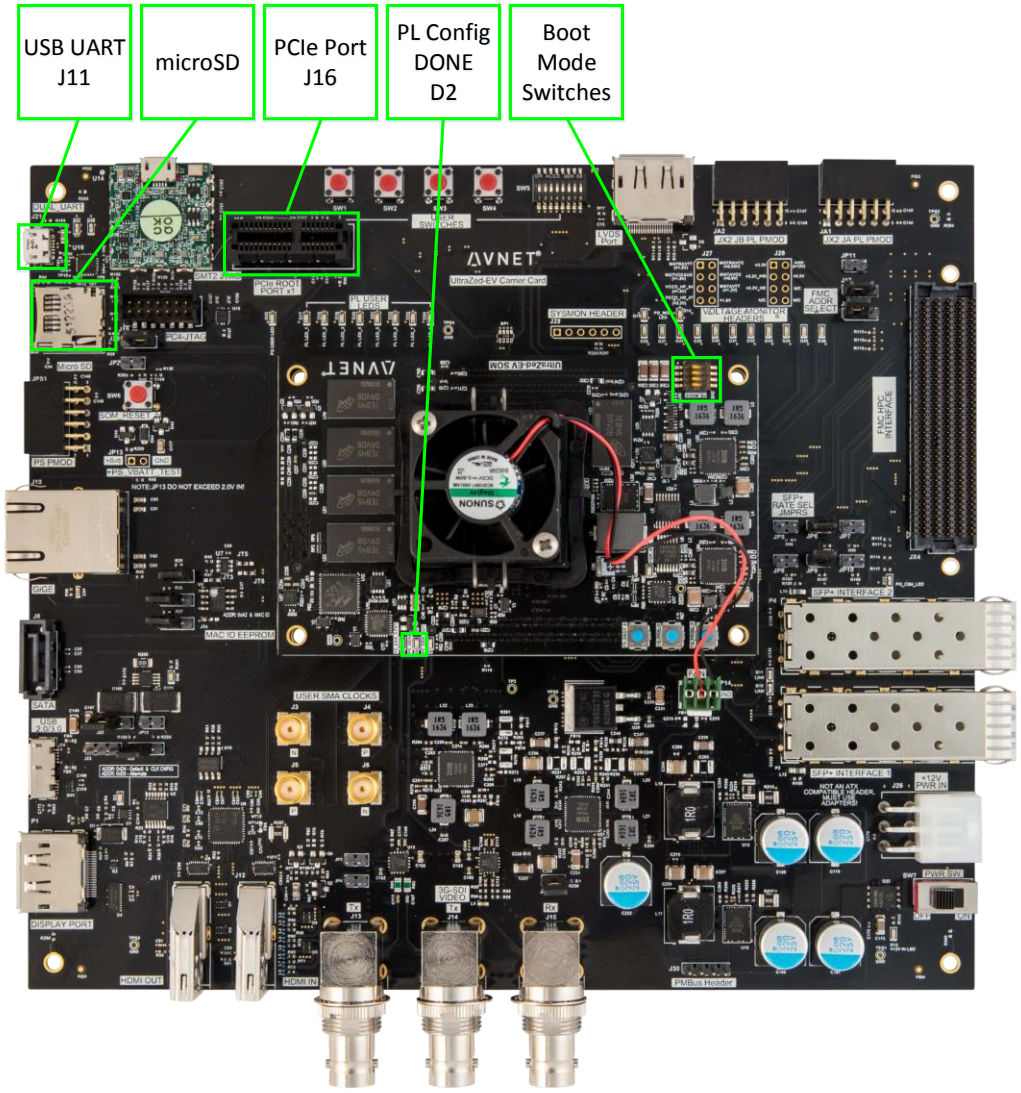
The hardware setup used to test this reference design includes:

- UltraZed-EV SOM (AES-ZU7EV-1-SOM-I-G) and EV Carrier Card (AES-ZUEV-CC-G)
- Lenovo ThinkPad T420 Laptop
    - Intel® Core i5-2540M CPU - 2.60 GHz
    - 4GB DDR3 Memory
    - SD card slot on PC or external USB SD card reader
- USB cable (Type A to Micro-USB Type B)
- 8GB MicroSD card
- CAT-5E Ethernet cable
- Broadcom “TIGON3” “BCM5751” (BCM95721A211) PCIe Ethernet NIC
    - https://www.amazon.com/Broadcom-BCM95721A211-PCI-E-Network-Adapter/dp/B001G2K18M
    - https://www.newegg.com/Product/Product.aspx?Item=14U-004V-00034

Experiment 0: Setting Up the UltraZed SOM with Carrier Card

UltraZed-EV with EV Carrier Card

Refer to the following figure and perform the following steps to set up the board when using the UltraZed-EV and EV Carrier.

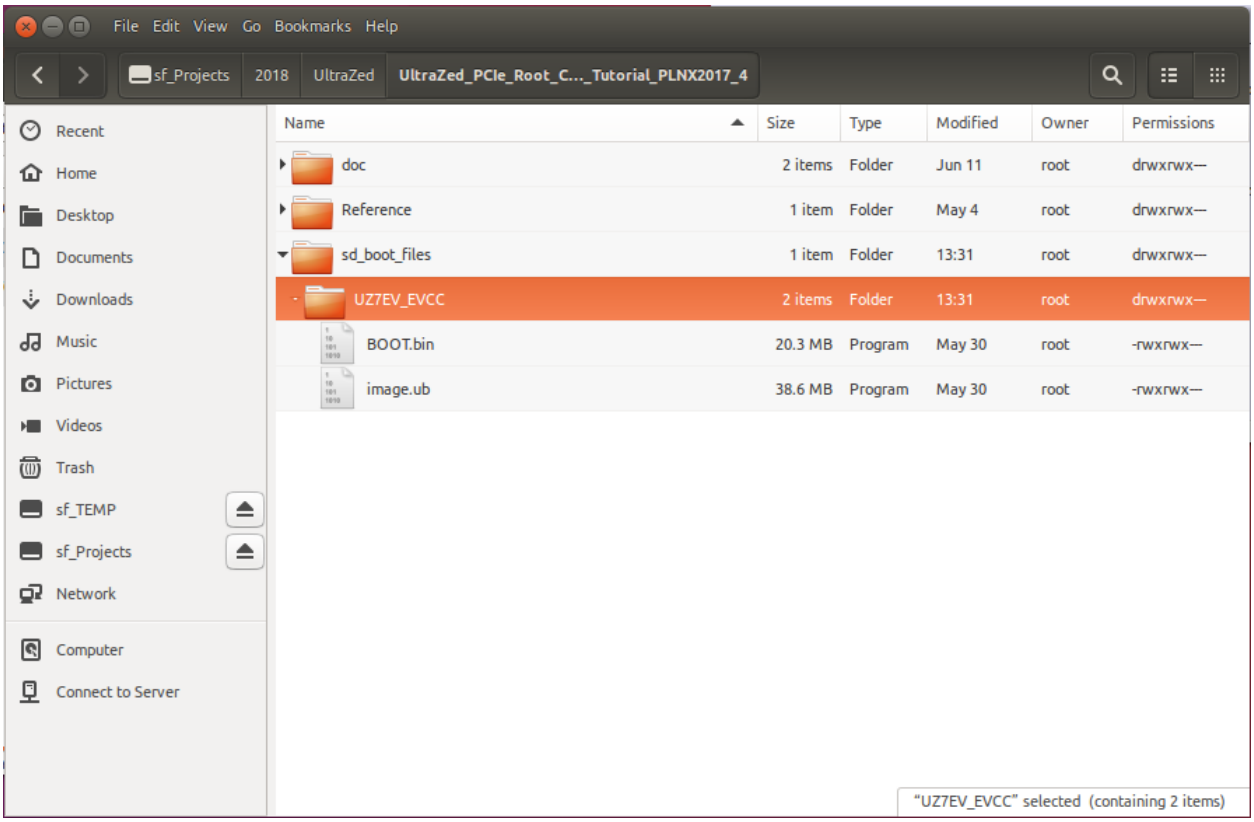


1. Plug the UltraZed-EV SOM onto the EV Carrier Card via JX1/JX2/JX3 connectors and connect the fan to the fan header (JP14) on the EV Carrier Card.
2. Set the UltraZed-EV SOM Boot Mode switch (SW2) (MODE[3:0] = SW2[1:4]) to OFF, ON, OFF, and ON positions (Boot Mode set to SD card, MODE[3:0] = 0x5).
3. Connect the USB-UART port on the EV Carrier Card (J21) to a free USB port on your PC.
4. Insert the PCIe Ethernet NIC into the PCIe slot connector (J16).
5. Connect the 12V power cable, but do not turn on the board yet.

Experiment 1: Setup Linux for UltraZed

The experiments in this tutorial are based upon the Linux build that is provided by Avnet as part of the UltraZed-EV PetaLinux 2017.4 Enhanced BSP available here. For your convenience the SD card bootable binaries are included with this design tutorial.

1. Copy the **BOOT.BIN** and **image.ub** files from the accompanying archive into the root folder of the microSD card.



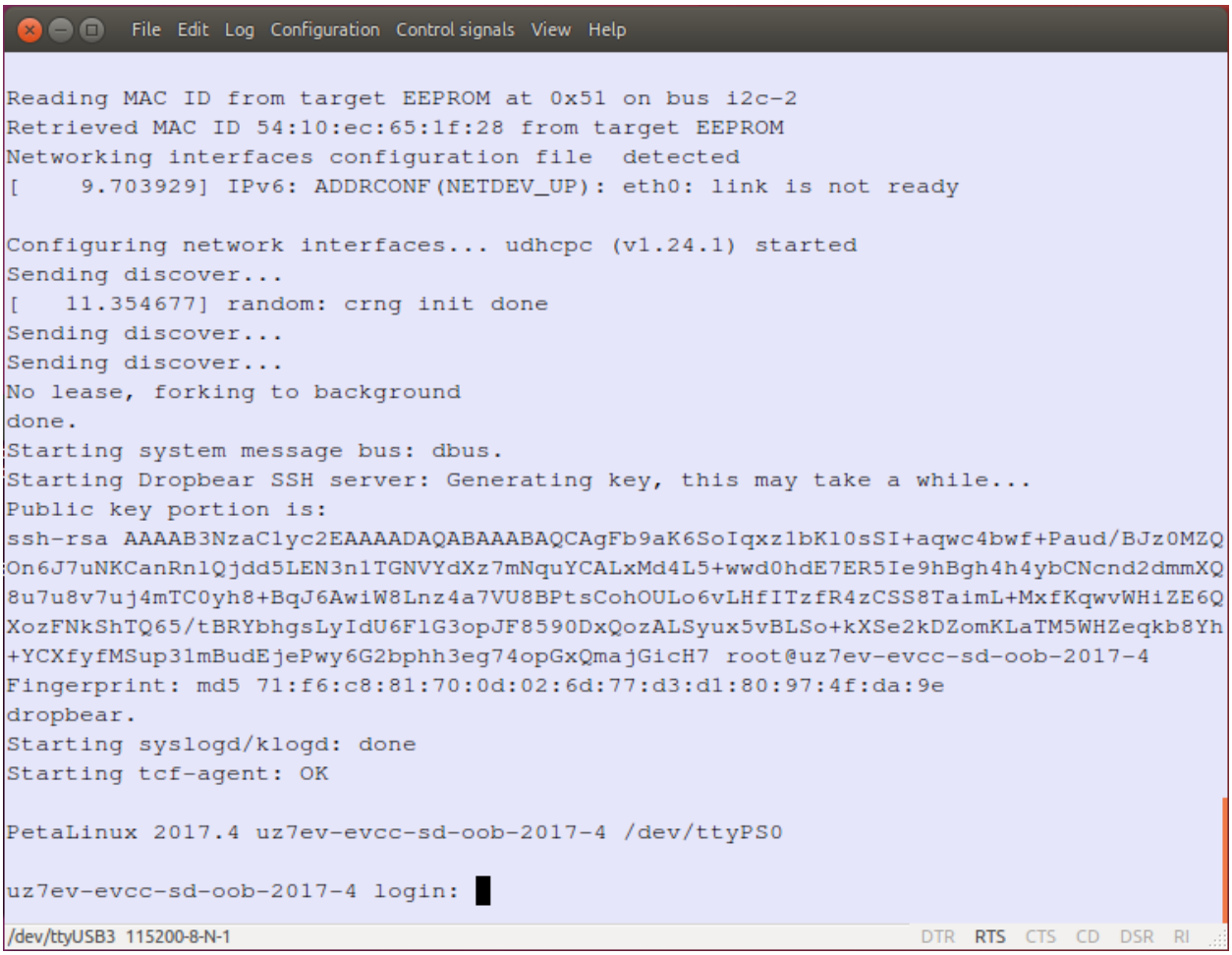
   Replace any existing versions of these files that may already be on the microSD card.

2. Insert the microSD card, prepared using the steps above, into the UltraZed Carrier microSD card cage.

3. Set the UltraZed Carrier power switch to the ON position. The UltraZed system will power on and the Power Good LED should illuminate.

4. Launch a terminal program with the 115200/8/n/1/n settings. For the example output shown here, **gtkterm** was used. For information on setting up **gtkterm** to use with the UltraZed USB-UART port, see the section **Appendix II: Troubleshooting Serial Connection** later in this document for further information.

5. You should observe terminal output from U-Boot and then Linux output appear in the **gtkterm** window.



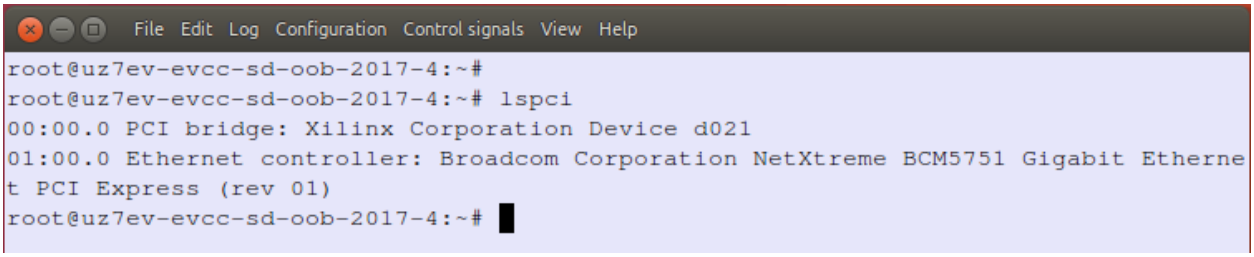
Experiment 2: Verify that the PCIe NIC is Recognized

Now that the embedded target software has been setup and UltraZed is booted with Linux to a login prompt, the file Read/Write Tests can be launched on the attached SATA drive.

1. Use the terminal window to enter the login **root** along with password **root** in order to gain access

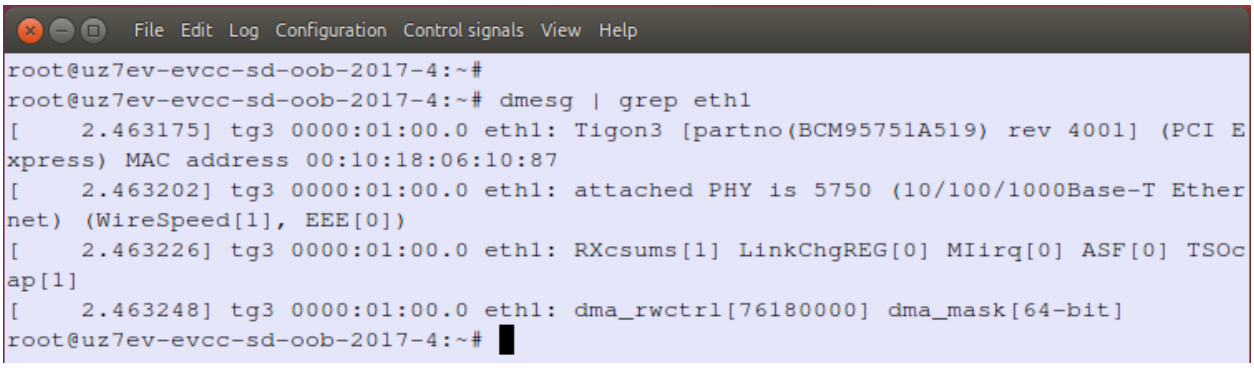
2. Check the PCIe device enumeration to verify that the Ethernet NIC is recognized and that the Ethernet interface is functioning correctly. If the **eth1** device listing does not appear as expected, verify your PCIe Ethernet NIC installation and Ethernet cable connection matches the one shown in Appendix I: Troubleshooting Ethernet Connection.

```
# lspci
```



3. It is also helpful to examine the Linux kernel boot messages to confirm the PCIe Ethernet NIC interface (eth1) was recognized successfully.

```
# dmesg | grep eth1
```



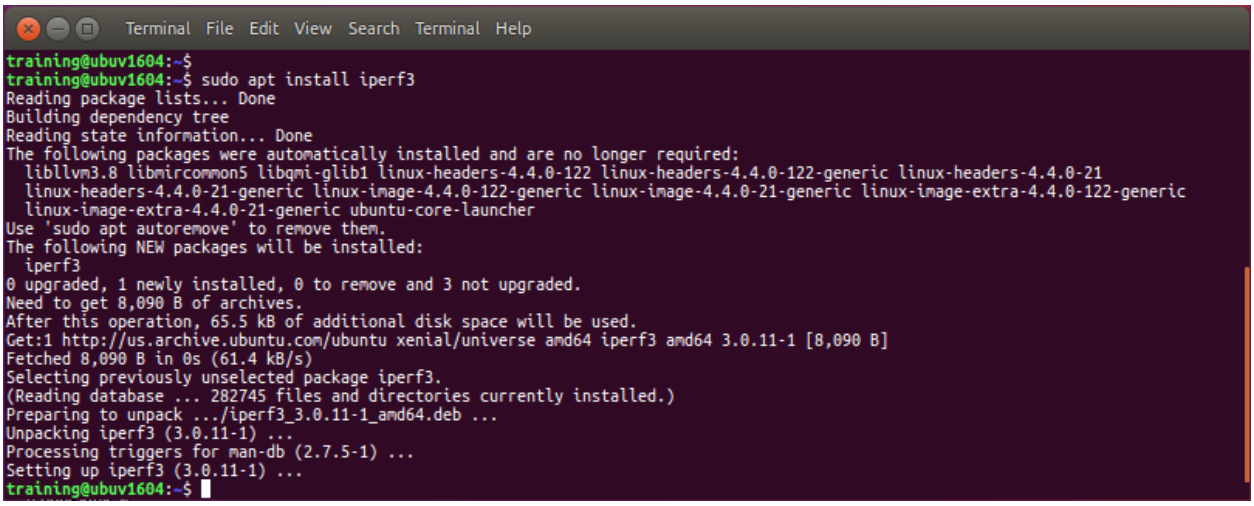
Experiment 3: Install iperf3 in the Ubuntu Virtual Machine

The Ethernet performance test runs the iperf3 application to test the achievable Ethernet throughput. The iperf3 application works by setting up a server on one end and a client on the other end of the connection. This requires that iperf3 also be installed in the Ubuntu Linux host.

1. Connect the Ubuntu Linux host to the internet.

2. Open a Ubuntu command window and use the Linux apt package manager utility to install iperf3:

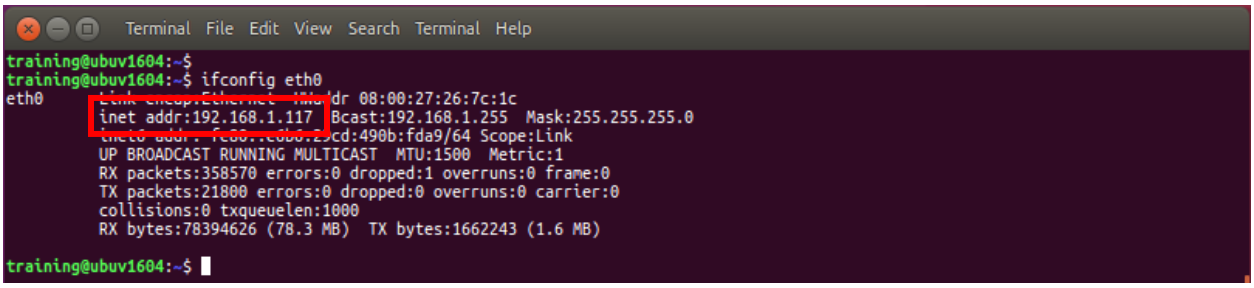
```
$ sudo apt install iperf3
```



Experiment 4: Running iperf3 Ethernet Tests

1. Use the same command window as in the previous step to determine the IP address assigned to the Ubuntu Linux host.

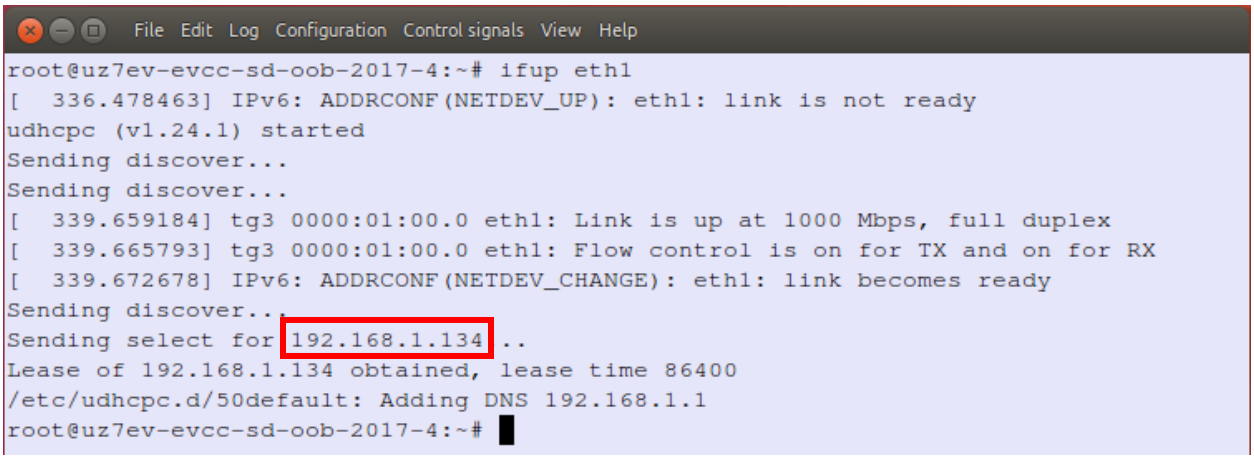
```
$ ifconfig eth0
```



In this case the IP address of the Ubuntu host is set to 192.168.1.117. This may be different for your network.

2. Use the GtkTerm serial terminal to enable the eth1 Ethernet interface on the PCIe NIC. You may first see a message that the “link is not ready”. You may safely ignore this message.

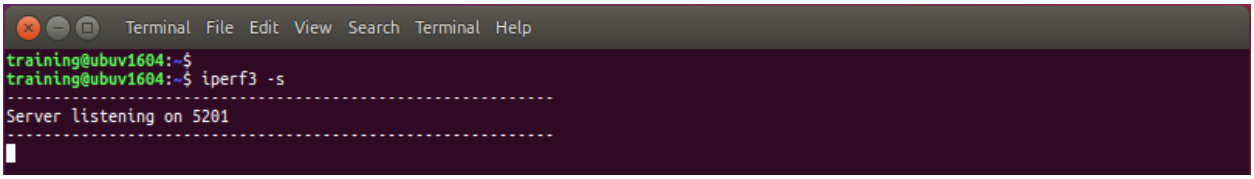
```
# ifup eth1
```



In this case the IP address of the UltraZed-EV is set to 192.168.1.134. This may be different for your network.

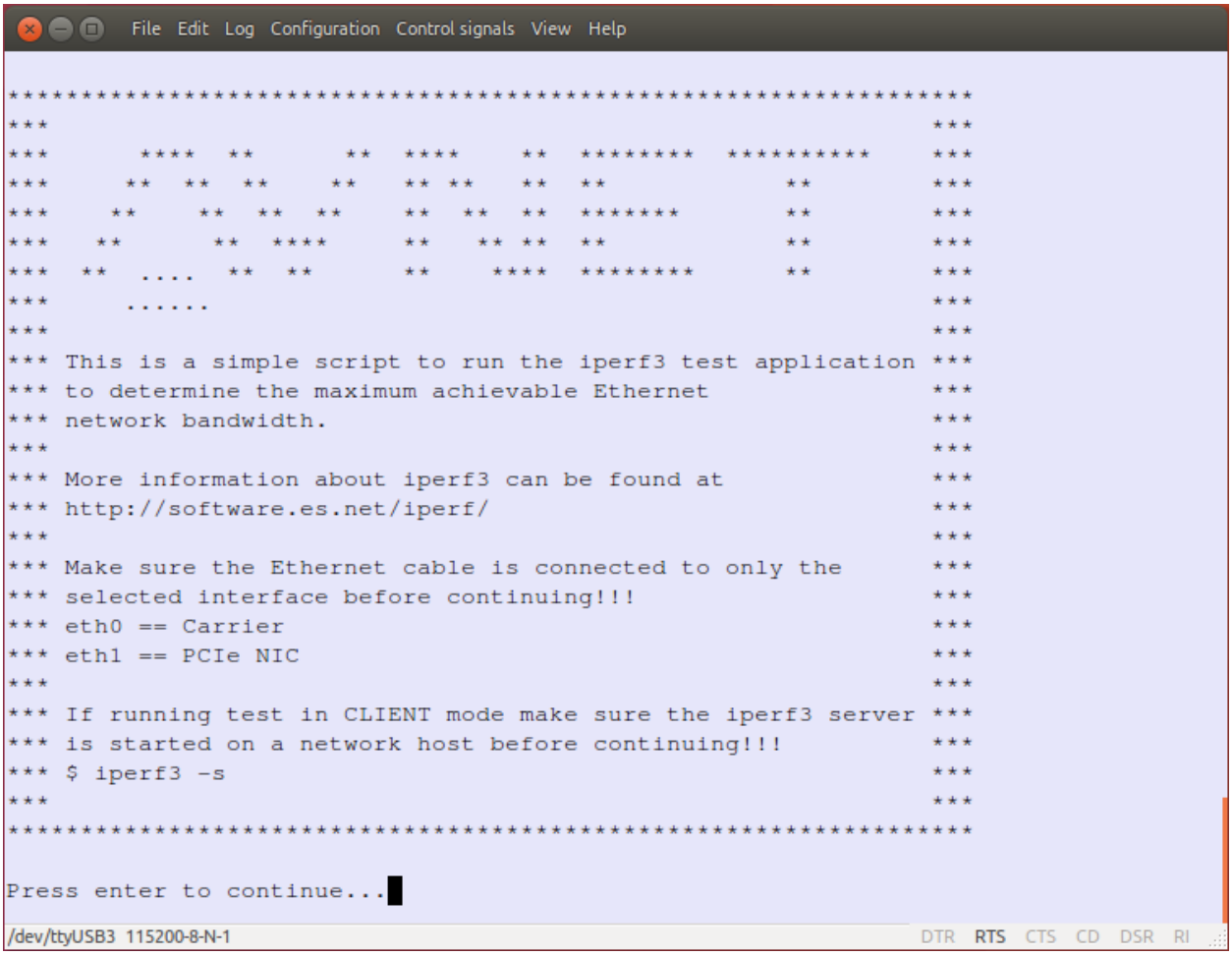
3. Use the same command window as in the previous step and start iperf3 in server mode on the Ubuntu Linux host.

```
$ iperf3 -s
```

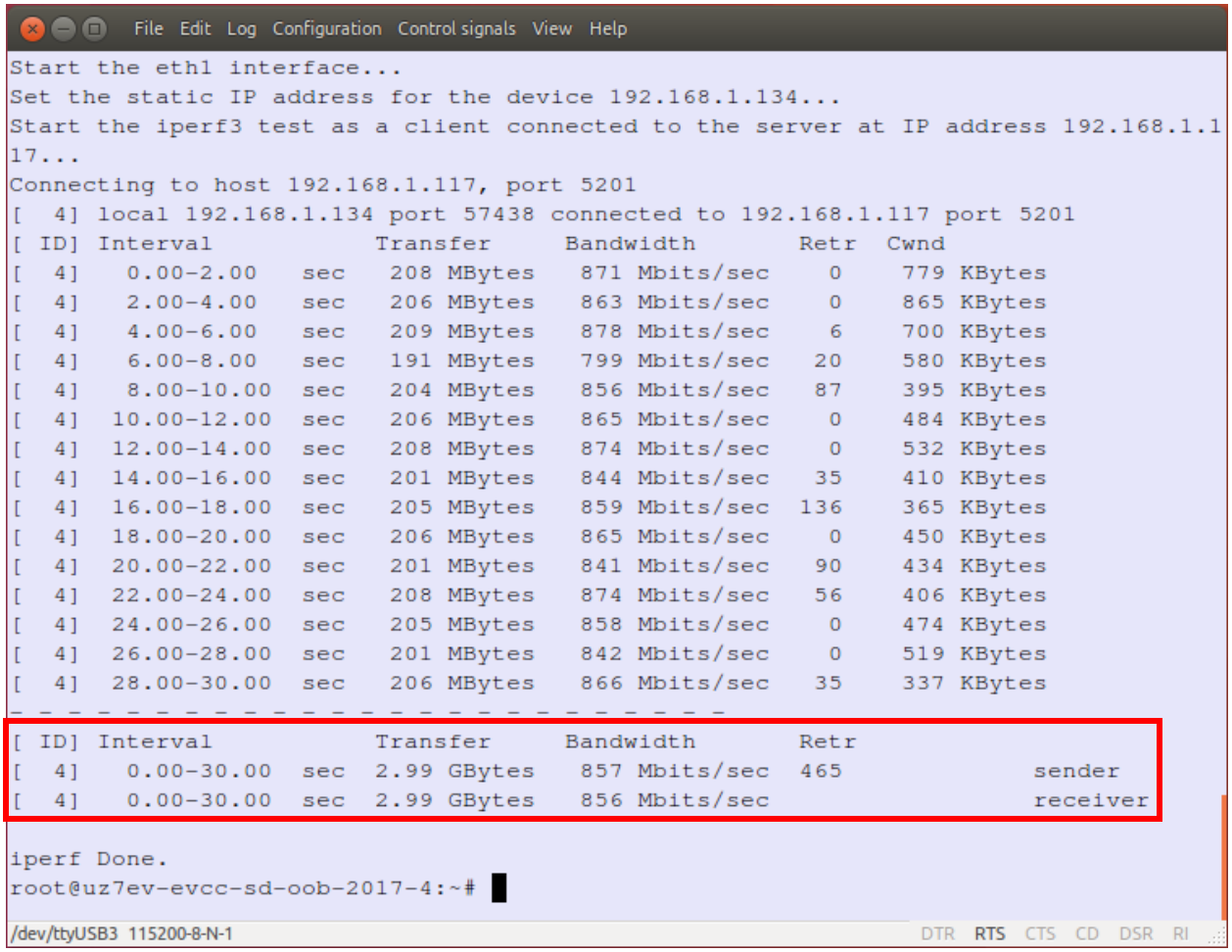


4. At the UltraZed serial terminal launch the included **network-test.sh** script to perform Ethernet performance tests using the Linux iperf3 command as a client on the eth1 interface using the following command. Press enter to continue when prompted at the splash screen.

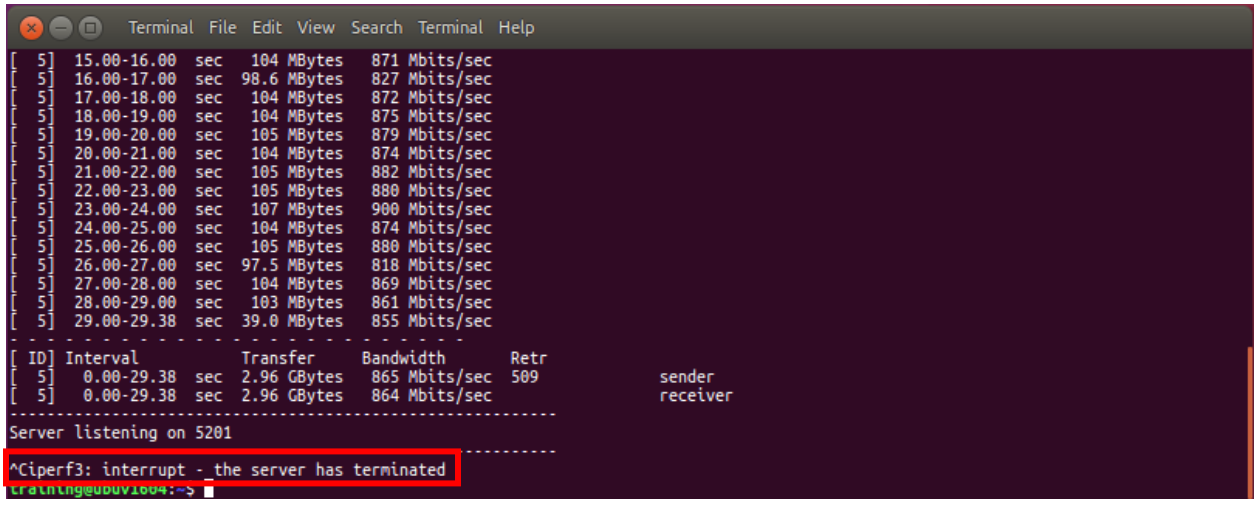
```
# ./network-test.sh -i eth1 -l 192.168.1.134 -m c -r 192.168.1.117
```



5. The network test script will run for 30 seconds and report the observed data throughput every 2 seconds. Upon completion of the iperf3 test, a throughput summary will be shown which shows the total amount of data transferred along with the measured throughput rate over the course of the test.

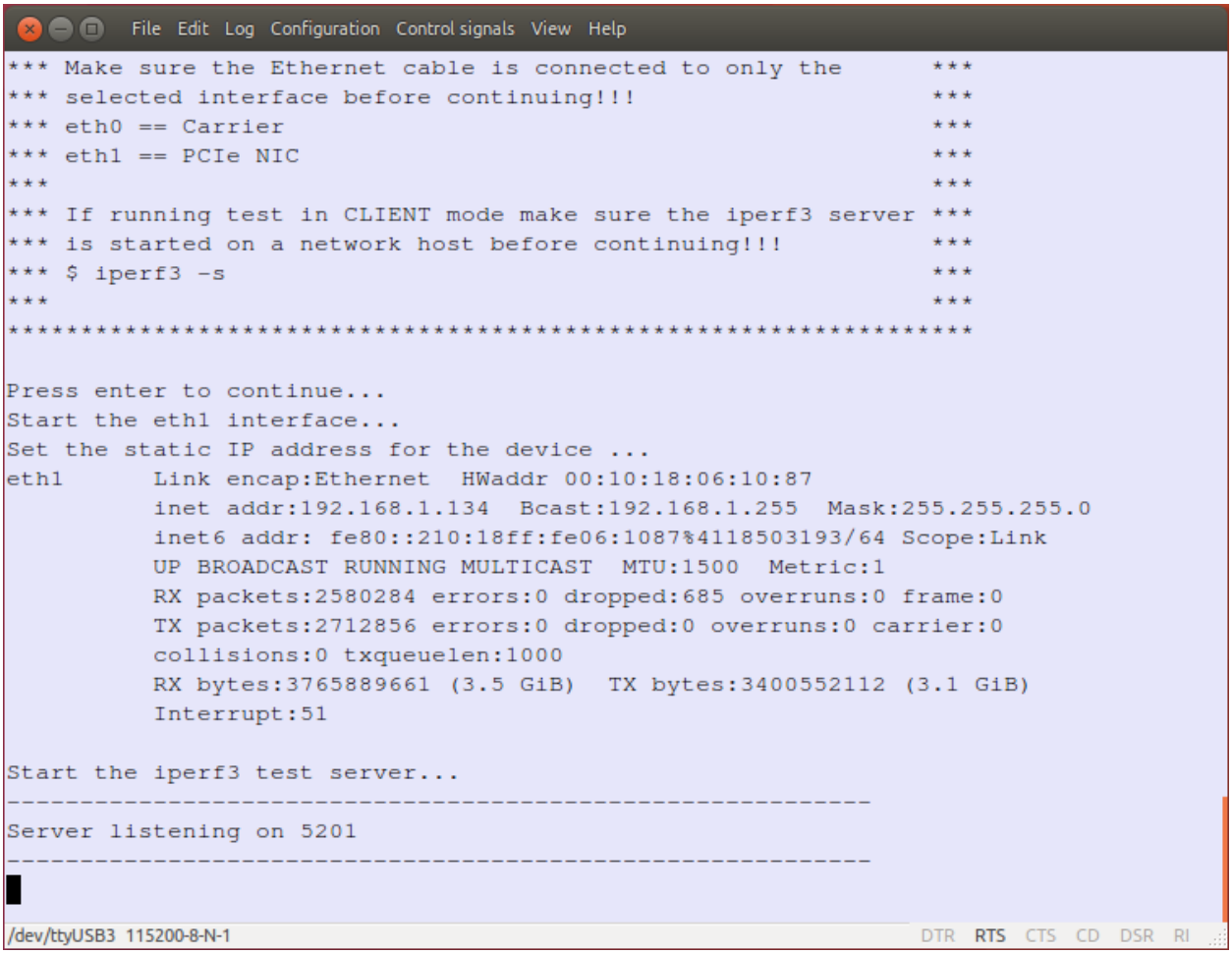


6. Go to the Ubuntu host command window and stop the iperf3 server with a <ctrl>-c.



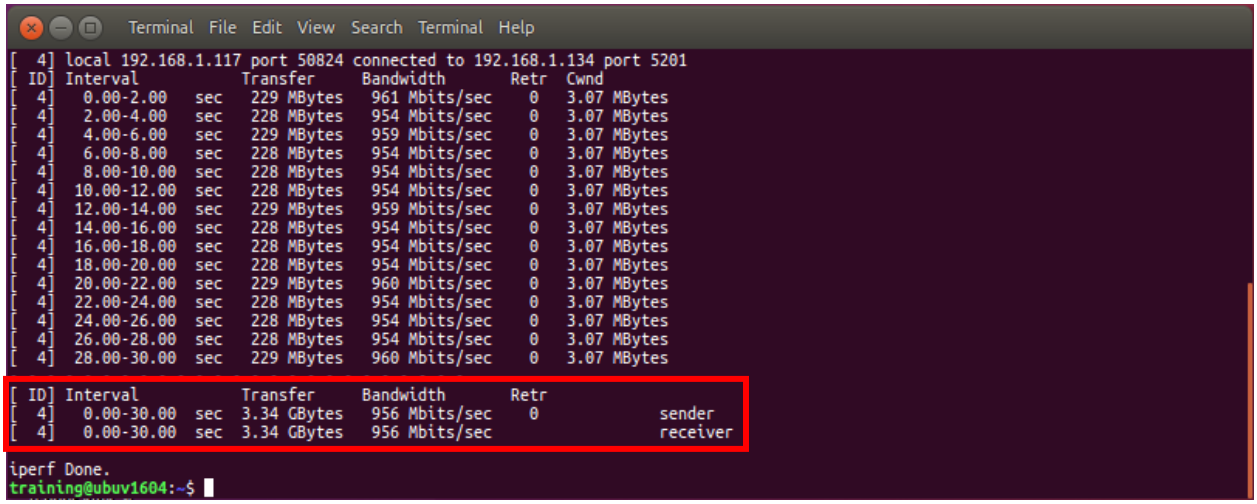
7. Now experiment with running the iperf3 tests as a server on the UltraZed-EV using the following command. Press enter when prompted at the splash screen to start iperf3 in server mode.

```
# ./network-test.sh -i eth1 -m s
```



7. Go back to the Ubuntu host command window and start iperf3 as a client to connect to the UltraZed-EV iperf3 server. Recall that the IP address of the UltraZed-EV is set to 192.168.1.134. This may be different on your network. The command below will run the iperf3 application in client mode and will run for 30 seconds and report the throughput every 2 seconds.

```
$ iperf3 -c 192.168.1.134 -t 30 -i 2
```

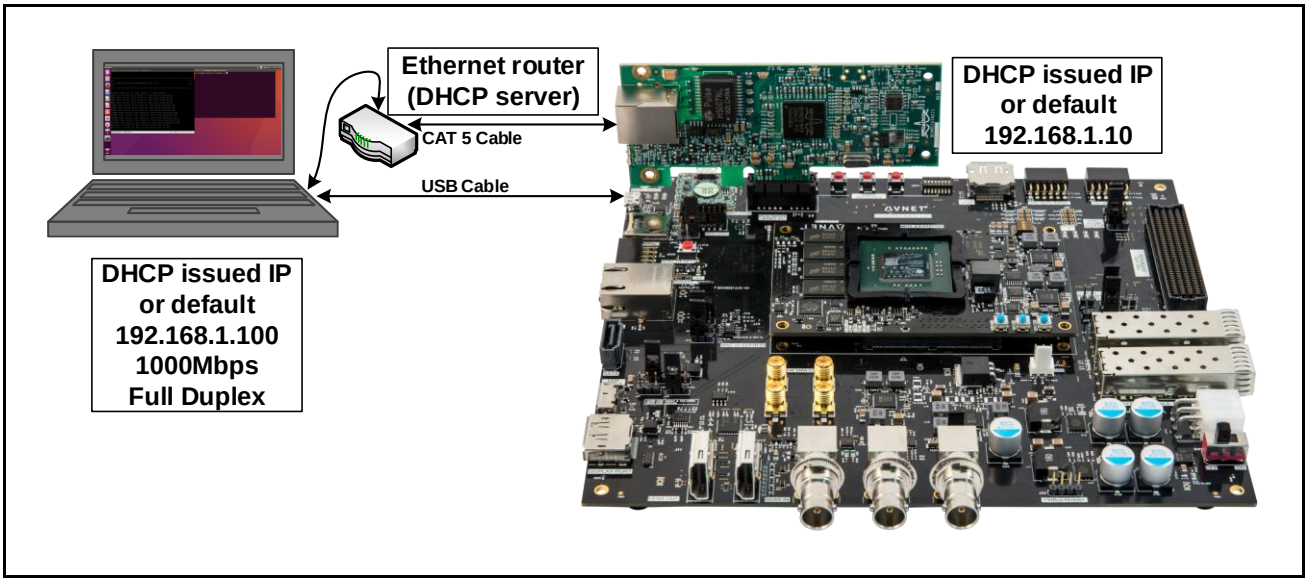


8. You may wish to experiment with the iperf3 command to specify different TCP/IP packet sizes, etc. to examine their effect on the Ethernet throughput to aid in tuning the performance of the Ethernet interface.

Appendix I: Troubleshooting Ethernet Connection

This section provides troubleshooting information for the Ethernet connection used in this UltraZed Open Source Linux PCIe Performance Test Tutorial.

1. The basic configuration for the UltraZed Open Source Linux PCIe Performance Test Tutorial is shown below:



2. Verify that the Broadcom BCM5751 PCIe Ethernet NIC add-in card is installed in the UltraZed-EV Carrier Card correctly.

3. Verify that the Ethernet cable is connected to an Ethernet router/switch or directly to a PC. If connected directly to a PC the IP address will need to be statically assigned and the link speed will need to be set to 1000Mbps full duplex.

4. Verify that the USB UART cable is connected between the PC and the UltraZed-EV Carrier Card. See Appendix II: Troubleshooting Serial Connection for more details.

Appendix II: Troubleshooting Serial Connection

This section provides troubleshooting information for the USB-UART serial connection used in this UltraZed Open Source Linux SATA Performance Test Tutorial. The experiments in this tutorial use Ubuntu **Serial port terminal** (gtkterm) as the serial terminal application is recommended for this tutorial but other serial terminal applications might work as well.

1. To make it easier to launch the terminal app (GtkTerm) without needing to provide the root password each time, open a command window and add the **current username** to the group for the **/dev/ttyUSBx** devices used for USB-UART:

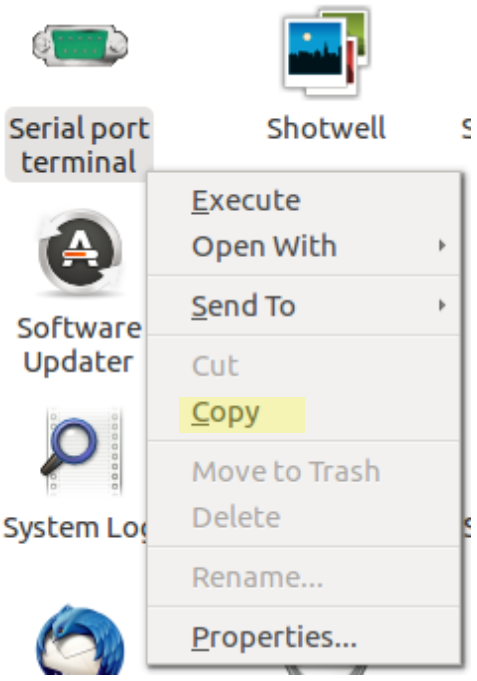
```
$ sudo usermod -a -G dialout <current username>
```

2. Install the **gtkterm** package:

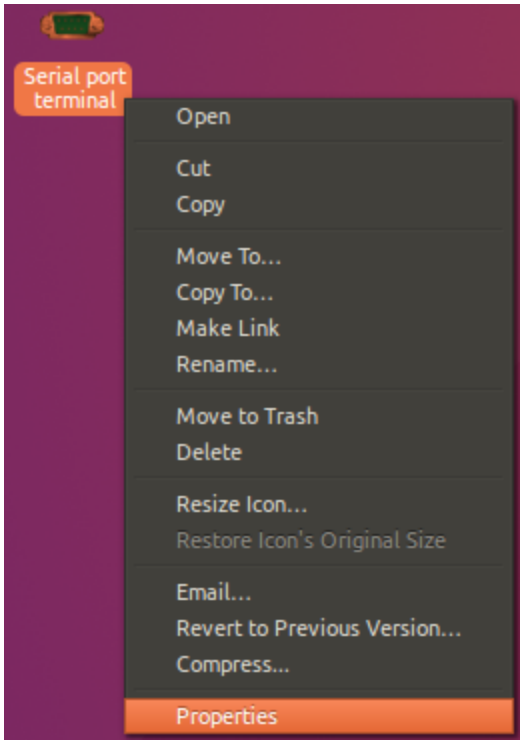
```
$ sudo apt-get install gtkterm
```

3. **Reboot the Virtual Machine to force the changes to take effect.**

4. Create a Desktop icon by copying and pasting **Serial port terminal** (gtkterm**)** application from the **/usr/share/applications** folder directly to the **~/Desktop** folder:

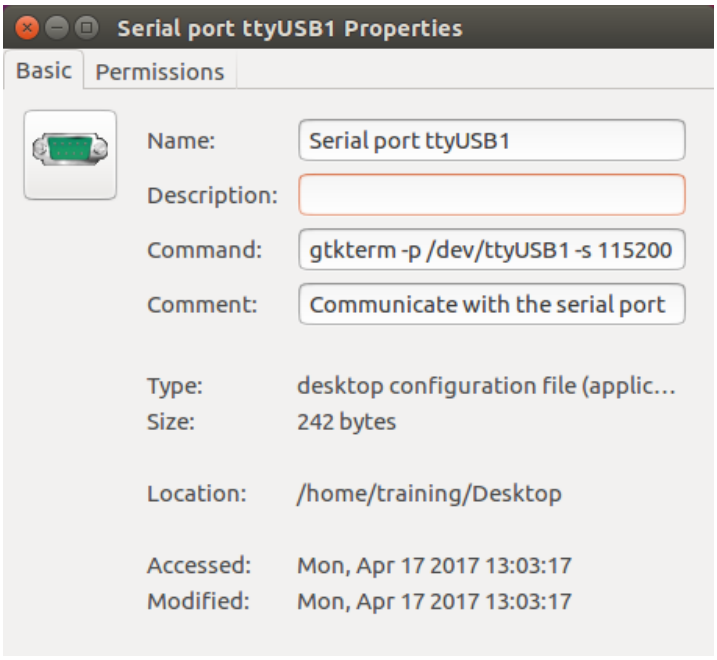


5. Right-click on the new **Serial port terminal** (gtkterm**)** application Desktop icon and select the **Properties** option.



6. Within the Properties window, set the attributes to match the USB-UART device attached to the system. In this example the USB-UART is attached to the **/dev/ttyUSB1** device entry. Close the properties window when finished.

   Name: **Serial port ttyUSB1**
   Command: **gtkterm -p /dev/ttyUSB1 -s 115200**



7. Insert a bootable microSD card prepared with the prebuilt binaries available with this tutorial into the UltraZed Carrier Card J4 slot.

8. Setup the UltraZed hardware as described in Experiment 0: Setting Up the UltraZed SOM with Carrier Card.

9. Set the UltraZed SOM Boot Mode switch (SW2) (MODE[3:0] = SW2[4:1]) to ON, OFF, ON, and OFF positions (Boot Mode set to SD Card, MODE[3:0] = 0xA).

10. Make sure the UltraZed Carrier Card power switch SW7 is in the OFF position.

11. Insert the UltraZed SOM module onto the UltraZed Carrier Card using the JX1, JX2, and JX3 connectors.

12. Close or disconnect the terminal that may have previously been open on your PC.

13. Plug in the UltraZed USB-UART cable between the host PC and the UltraZed Carrier Card USB-UART port (J21).

14. Insert the appropriate country plug into the 12V AC/DC adapter. Plug it into the J7 2x3 power connector. (NOTE – this 2x3 connector is NOT compatible with ATX power supplies.)

15. Set the UltraZed Carrier power switch SW7 to the ON position. The UltraZed system will power on and the Power Good LED (D2) should illuminate.

16. Check the kernel output log for signs that the USB-UART device has enumerated and note the ttyUSB device that is enumerated. USB-UART device should enumerate as **/dev/ttyUSB0** or similar.

```
$ dmesg |grep ttyUSB
```

17. Create system default udev rules to give USB-UART devices sufficient permissions for all users similar.

```
$ sudo cp /lib/udev/rules.d/50-udev-default.rules /etc/udev/rules.d/.
```

18. Edit the system default udev rules with **vi** text editor.

```
$ sudo vi /etc/udev/rules.d/50-udev-default.rules
```

19. Add the following 2 lines to **50-udev-default.rules** somewhere just after the 'tty' section.

```
# relax the permissions just for ttyUSB0

KERNEL=="ttyUSB0",              MODE="0666"
```

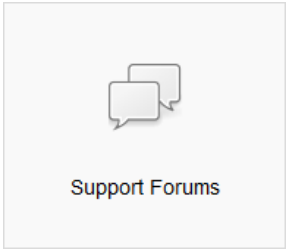
20. Save the changes to the **50-udev-default.rules** file and exit **vi** text editor.

21. Run gtkterm again in normal user mode and check for terminal output.

Appendix III: Getting Support

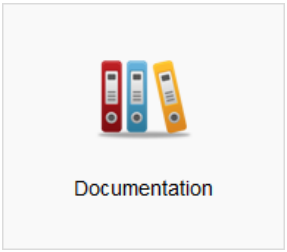
Avnet Support

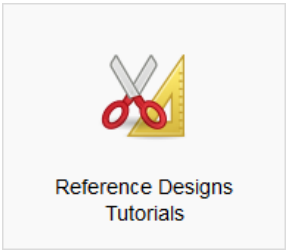
- Technical support is offered online through the ultrazed.org website support forums. UltraZed users are encouraged to participate in the forums and offer help to others when possible.
  http://ultrazed.org/forums/zed-english-forum
  http://ultrazed.org/forums/software-application-development



- For questions regarding the UltraZed community website, please direct questions to the ultrazed.org Web Master (webmaster@ultrazed.org).

- To access the most current collateral for the UltraZed, visit the community support page (www.ultrazed.org/content/support) and click one of the icons shown below:





  - UltraZed-EV SOM Documentation
    http://zedboard.org/support/documentation/21811

  - UltraZed-EV Carrier Card
    - Documentation
      TBD
    - Reference Designs
      TBD

- Instructions for how to setup the Ubuntu virtual machine if using a Windows host PC
  http://zedboard.org/sites/default/files/design/VirtualBox_Installation_Guide_2017_2.zip

Internet Support

Here are some helpful links regarding some of the Linux software applications mentioned in this tutorial:

iperf3

- Benchmark Ethernet throughput with iperf3
  https://software.es.net/iperf/

- iperf3 FAQ
  https://software.es.net/iperf/faq.html

Xilinx Support

For questions regarding products within the Product Entitlement Account, send an email message to the Customer Service Representative in your region:

- Canada, USA and South America - isscs_cases@xilinx.com
- Europe, Middle East, and Africa - eucases@xilinx.com
- Asia Pacific including Japan - apaccase@xilinx.com

For technical support, including the installation and use of the product license file, contact Xilinx Online Technical Support at www.xilinx.com/support. The following assistance resources are also available on the website:

- Software, IP and documentation updates
- Access to technical support Web tools
- Searchable answer database with over 4,000 solutions
- User forums

Revision History

Date	Version	Revision
13 Jun 2018	01	Initial Release