MicroZed Open Source Linux Ethernet Performance Test Tutorial

Version 14.5.01

Revision History

Version	Description	Date
14.5.01	Release to microzed.org	August 13, 2013

Table of Contents

Tutorial Overview

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

1. Build an OSL (open source Linux) application from the Linux command prompt building upon the lab activities of the Avnet 2013 SpeedWay training course entitled **Implementing Linux on the Zynq™-7000 SoC**.

2. Perform Ethernet performance tests on Zynq platforms running open source Linux.

Tutorial Setup Requirements

To complete this tutorial, the following software and hardware setups are recommended. This tutorial builds upon the concepts and lab activities of the Avnet 2012 SpeedWay training course entitled **Implementing Linux on the Zynq™-7000 SoC**. Please refer back to this training material on the MicroZed community website for further information on how to configure the underlying OSL platform.

The instructions in this tutorial assume that the cross build platform is a CentOS V6.3 installation running within a virtual machine on a Windows-based host. Though other build systems may work using the same or similar instructions, those build systems are not supported.

Software

Recommended software tools:

- Xilinx ISE WebPACK 14.5 (Free license and download from Xilinx website)
  - Check Xilinx AR51895 for required installation work-arounds
- Silicon Labs CP210x USB-to-UART Bridge Driver
  - Check "Silicon Labs CP210x USB-to-UART Setup Guide" under Documentation section on MicroZed.org for detailed instructions on installing and configuring this driver
- Tera Term (Exact version used for this tutorial is V4.75)
- VMware Player V5.0.0 (Exact version used for this tutorial is VMware-player-5.0.0-812388.exe)
- CentOS V6.3 64-bit installer image (Exact version used for this tutorial is CentOS-6.3-x86_64-bin-DVD1.iso)
- Sourcery CodeBench Zynq toolchain (Exact version used for this tutorial is xilinx-2011.09-50-arm-xilinx-linux-gnueabi.bin)
- Git SCM toolset (Exact version used for this tutorial is V1.7.1)
- Adobe Reader for viewing PDF content (Exact version used for this tutorial is Adobe Reader X 10.1.4)

Hardware

Targeted hardware consists of the following:

- Available SD card slot on PC or external USB-based SD card reader
- Avnet MicroZed (AES-Z7MB-7Z010-G)– *included in kit*
- USB cable (Type A to Micro-USB Type B) – *included in kit*
- 4GB MicroSD card – *included in kit*
- CAT-5e Ethernet patch cable

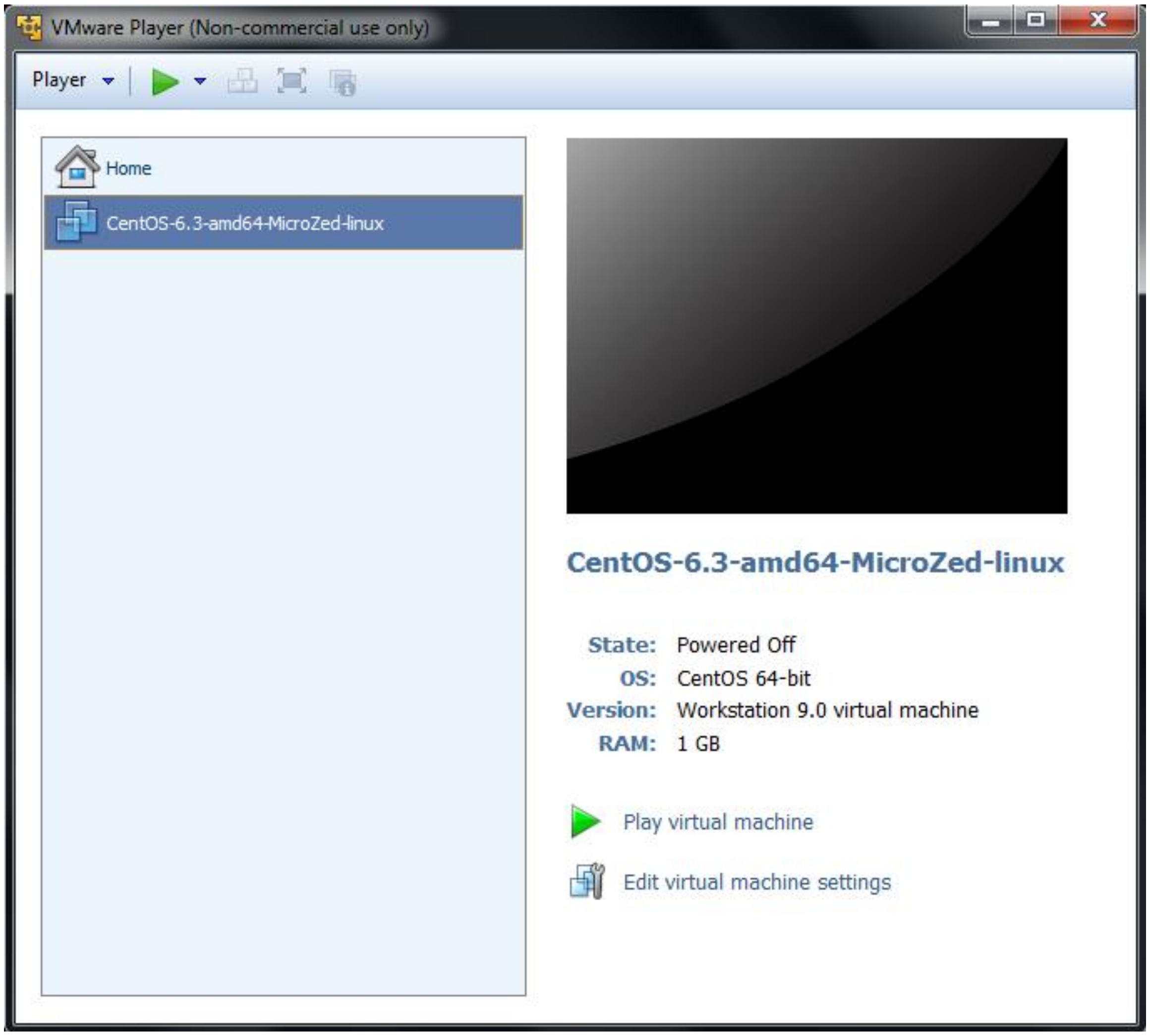
Building Iperf Test Application for Zynq Target

This section describes how to build the Iperf application from source code. Iperf is most commonly used for measuring maximum TCP and UDP bandwidth performance. Iperf allows the tuning of various parameters and UDP characteristics. Iperf also has capability to report bandwidth, delay jitter, and datagram loss.

1. If the CentOS virtual machine is not already open, launch the VMware Player application from by selecting **Start → All Programs → VMware → VMware Player**. If the CentOS virtual machine is already open, skip ahead to Step 4.

Figure 1 – The VMware Player Application Icon

2. Select the virtual machine named **CentOS-6.3-amd64-MicroZed-linux** from the selections on the left and then click on the **Play virtual machine** button to the right.


VMware Player (Non-commercial use only)
Player
Home
CentOS-6.3-amd64-MicroZed-linux
Play virtual machine
Edit virtual machine settings

CentOS-6.3-amd64-MicroZed-linux

State: Powered Off
OS: CentOS 64-bit
Version: Workstation 9.0 virtual machine
RAM: 1 GB

Figure 2 – The VMware Player Application

3. If prompted for a workstation login, click on the user entry **training** and enter the password **Avnet** in order to log into the system.

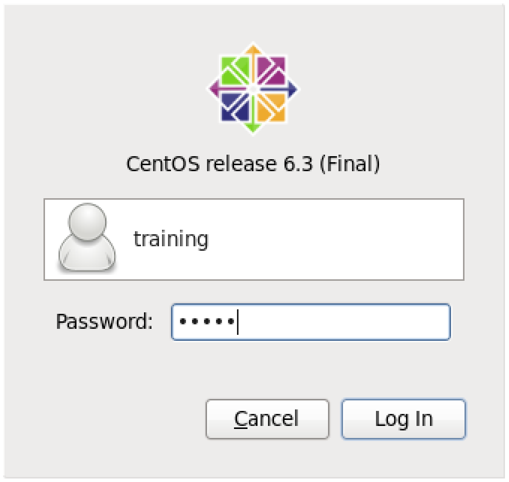


Figure 3 – The CentOS Workstation Login

4. If a terminal is not already open on the desktop of the CentOS guest operating system, open a terminal window through the **Applications→System Tools→Terminal** menu item. If a terminal is already open, bring that terminal session into focus on the desktop.

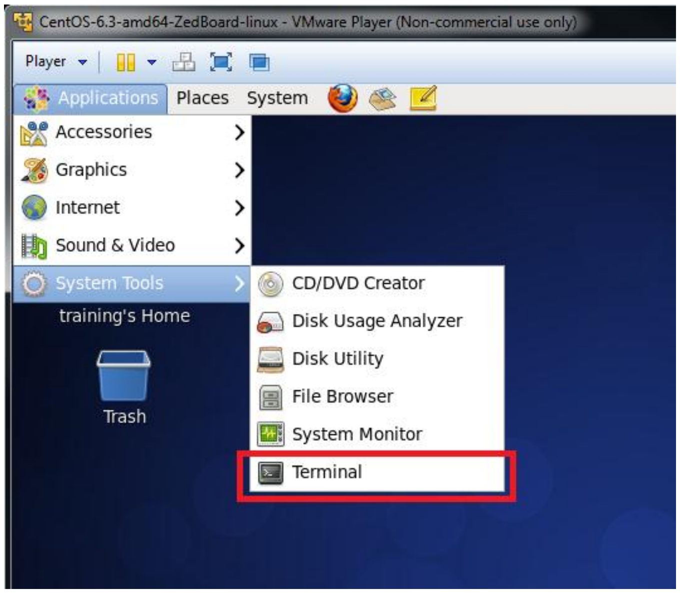


Figure 4 – Launching the CentOS Terminal from the Desktop

5. Change the current working directory to the training user home directory.

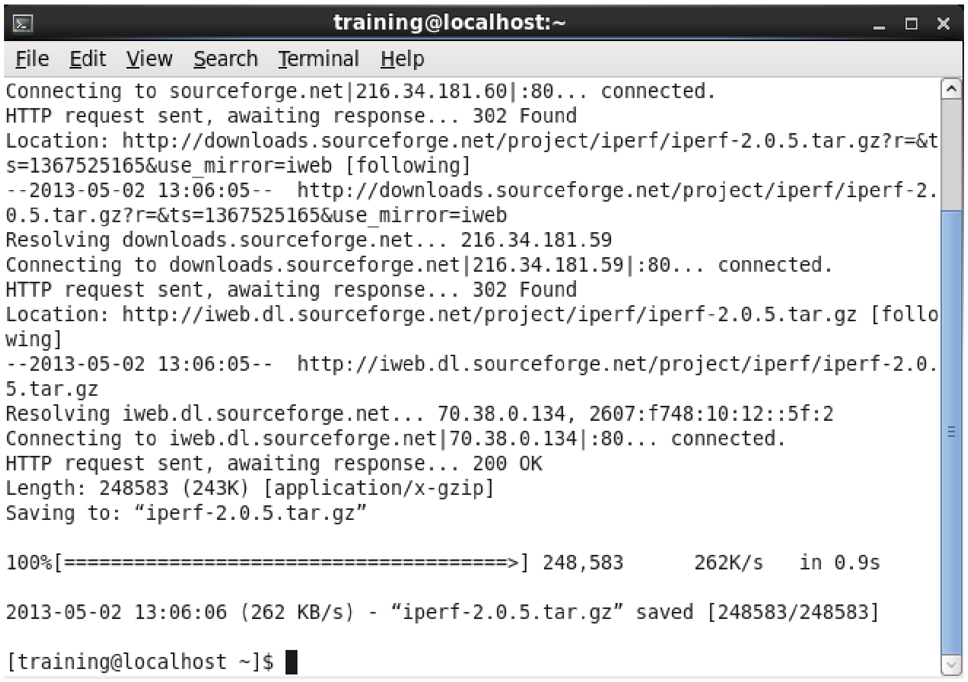
```
$ cd ~
```

6. The latest Iperf source can be obtained from the following SourceForge URL:

   http://sourceforge.net/projects/iperf/files/latest/download

   Use the following command to download the source code:

```
$ wget http://sourceforge.net/projects/iperf/files/latest/download
```

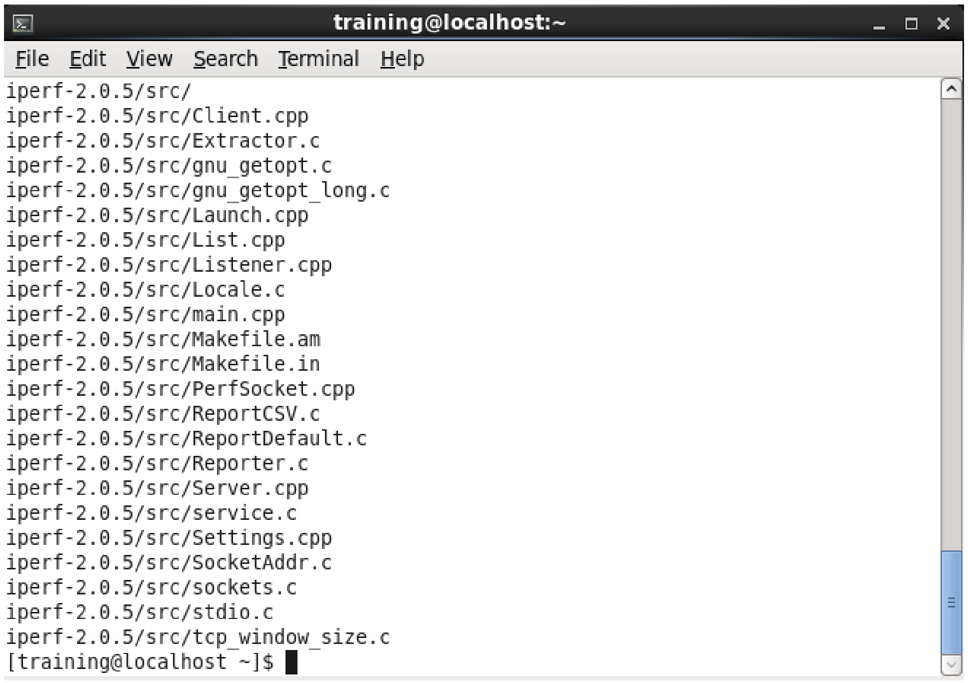


```
Connecting to sourceforge.net|216.34.181.60|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://downloads.sourceforge.net/project/iperf/iperf-2.0.5.tar.gz?r=&t
s=1367525165&use_mirror=iweb [following]
--2013-05-02 13:06:05--  http://downloads.sourceforge.net/project/iperf/iperf-2.
0.5.tar.gz?r=&ts=1367525165&use_mirror=iweb
Resolving downloads.sourceforge.net... 216.34.181.59
Connecting to downloads.sourceforge.net|216.34.181.59|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://iweb.dl.sourceforge.net/project/iperf/iperf-2.0.5.tar.gz [follo
wing]
--2013-05-02 13:06:05--  http://iweb.dl.sourceforge.net/project/iperf/iperf-2.0.
5.tar.gz
Resolving iweb.dl.sourceforge.net... 70.38.0.134, 2607:f748:10:12::5f:2
Connecting to iweb.dl.sourceforge.net|70.38.0.134|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 248583 (243K) [application/x-gzip]
Saving to: “iperf-2.0.5.tar.gz”

100%[======================================>] 248,583      262K/s   in 0.9s

2013-05-02 13:06:06 (262 KB/s) - “iperf-2.0.5.tar.gz” saved [248583/248583]

[training@localhost ~]$ 
```

Figure 5 – Obtaining the Iperf Source Code Archive

7. Unpack the Iperf source code archive.

```
$ tar -zxvf iperf-2.0.5.tar.gz
```



```
iperf-2.0.5/src/
iperf-2.0.5/src/Client.cpp
iperf-2.0.5/src/Extractor.c
iperf-2.0.5/src/gnu_getopt.c
iperf-2.0.5/src/gnu_getopt_long.c
iperf-2.0.5/src/Launch.cpp
iperf-2.0.5/src/List.cpp
iperf-2.0.5/src/Listener.cpp
iperf-2.0.5/src/Locale.c
iperf-2.0.5/src/main.cpp
iperf-2.0.5/src/Makefile.am
iperf-2.0.5/src/Makefile.in
iperf-2.0.5/src/PerfSocket.cpp
iperf-2.0.5/src/ReportCSV.c
iperf-2.0.5/src/ReportDefault.c
iperf-2.0.5/src/Reporter.c
iperf-2.0.5/src/Server.cpp
iperf-2.0.5/src/service.c
iperf-2.0.5/src/Settings.cpp
iperf-2.0.5/src/SocketAddr.c
iperf-2.0.5/src/sockets.c
iperf-2.0.5/src/stdio.c
iperf-2.0.5/src/tcp_window_size.c
[training@localhost ~]$
```

Figure 6 – Unpacking the Iperf Source Code Archive

8. Change from the current directory into the Iperf source directory.

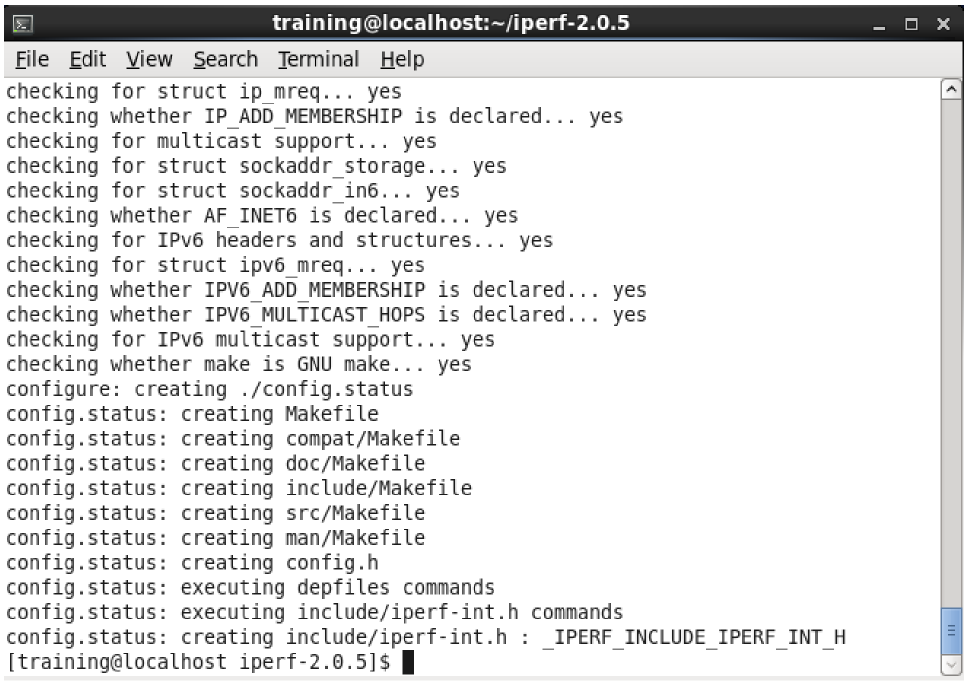
```
$ cd ~/iperf-2.0.5/
```

9. Configure Iperf for the Zynq target by using the configure command.

   Hint: Up until this point many of the command line entry instructions have been manageable to enter by hand. Some of the commands in the following steps take up multiple lines and can be difficult to manually enter accurately. One workaround to this challenge is to **Copy** the following single line from this PDF document into the VM clipboard by right-clicking in Adobe Reader and selecting the **Copy** option. Then **Paste** contents from the VM clipboard to the CentOS terminal window by right-clicking in the terminal window and selecting the **Paste** option. The backslash character **'\'** allows a command entry to be followed by an additional line entry when followed by the **<Enter>** key.

```
$ ac_cv_func_malloc_0_nonnull=yes ./configure \
--build=i686-linux --host=arm-xilinx-linux-gnueabi \
--target=arm-xilinx-linux-gnueabi \
--prefix=/home/training/iperf-2.0.5/
```

   Once the Iperf configuration has completed, the terminal output should look similar to that shown in Figure 7.


training@localhost:~/iperf-2.0.5
File Edit View Search Terminal Help

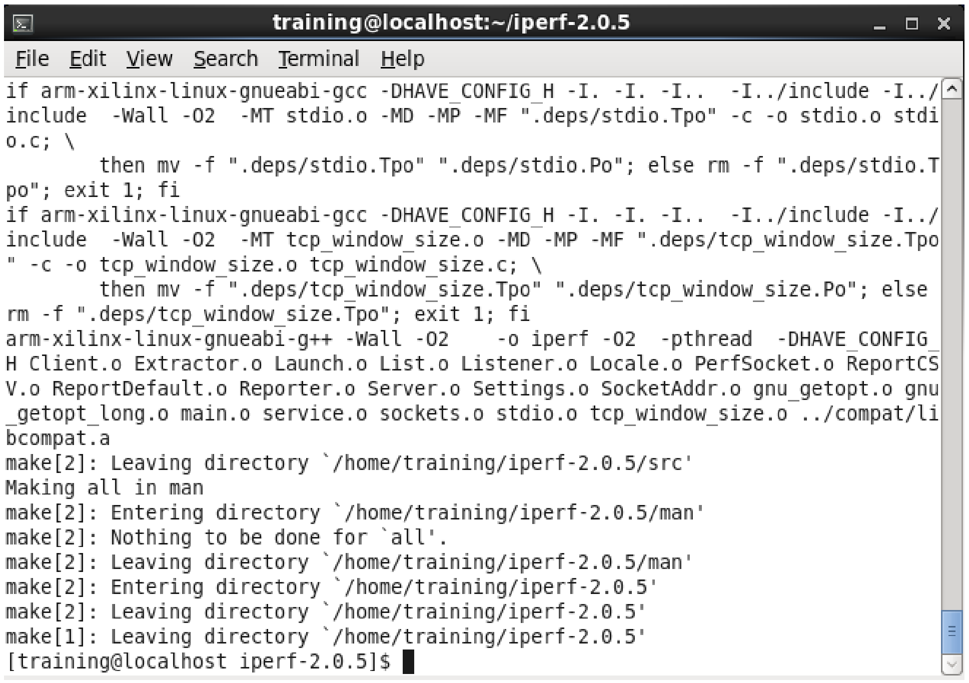
```
checking for struct ip_mreq... yes
checking whether IP_ADD_MEMBERSHIP is declared... yes
checking for multicast support... yes
checking for struct sockaddr_storage... yes
checking for struct sockaddr_in6... yes
checking whether AF_INET6 is declared... yes
checking for IPv6 headers and structures... yes
checking for struct ipv6_mreq... yes
checking whether IPV6_ADD_MEMBERSHIP is declared... yes
checking whether IPV6_MULTICAST_HOPS is declared... yes
checking for IPv6 multicast support... yes
checking whether make is GNU make... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating compat/Makefile
config.status: creating doc/Makefile
config.status: creating include/Makefile
config.status: creating src/Makefile
config.status: creating man/Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing include/iperf-int.h commands
config.status: creating include/iperf-int.h : _IPERF_INCLUDE_IPERF_INT_H
[training@localhost iperf-2.0.5]$ 
```

Figure 7 – Configuring Iperf

10. Build Iperf using the following command using the configuration that was just created:

```
$ make
```

Once the build has completed, the terminal output should look similar to that shown in Figure 8.


training@localhost:~/iperf-2.0.5
File Edit View Search Terminal Help

```
if arm-xilinx-linux-gnueabi-gcc -DHAVE_CONFIG_H -I. -I. -I..  -I../include -I../
include  -Wall -O2  -MT stdio.o -MD -MP -MF ".deps/stdio.Tpo" -c -o stdio.o stdi
o.c; \
        then mv -f ".deps/stdio.Tpo" ".deps/stdio.Po"; else rm -f ".deps/stdio.T
po"; exit 1; fi
if arm-xilinx-linux-gnueabi-gcc -DHAVE_CONFIG_H -I. -I. -I..  -I../include -I../
include  -Wall -O2  -MT tcp_window_size.o -MD -MP -MF ".deps/tcp_window_size.Tpo
" -c -o tcp_window_size.o tcp_window_size.c; \
        then mv -f ".deps/tcp_window_size.Tpo" ".deps/tcp_window_size.Po"; else
rm -f ".deps/tcp_window_size.Tpo"; exit 1; fi
arm-xilinx-linux-gnueabi-g++ -Wall -O2    -o iperf -O2  -pthread  -DHAVE_CONFIG_
H Client.o Extractor.o Launch.o List.o Listener.o Locale.o PerfSocket.o ReportCS
V.o ReportDefault.o Reporter.o Server.o Settings.o SocketAddr.o gnu_getopt.o gnu
_getopt_long.o main.o service.o sockets.o stdio.o tcp_window_size.o ../compat/li
bcompat.a
make[2]: Leaving directory `/home/training/iperf-2.0.5/src'
Making all in man
make[2]: Entering directory `/home/training/iperf-2.0.5/man'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/training/iperf-2.0.5/man'
make[2]: Entering directory `/home/training/iperf-2.0.5'
make[2]: Leaving directory `/home/training/iperf-2.0.5'
make[1]: Leaving directory `/home/training/iperf-2.0.5'
[training@localhost iperf-2.0.5]$ 
```

Figure 8 – Building Iperf

11. If a file browser window is not already open from a previous exercise, open a file browser window through the **Applications→System Tools→File Browser** menu item.

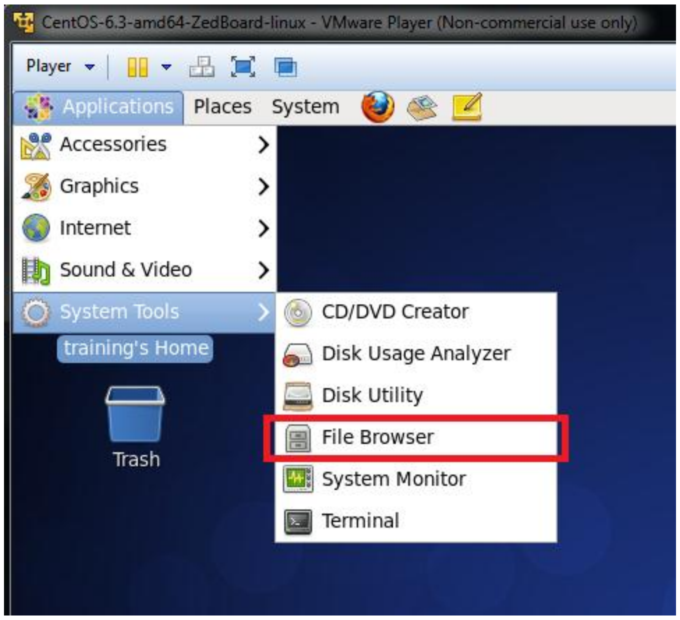


Figure 9 – CentOS File Browser

12. Locate the file **/home/training/iperf-2.0.5/src/iperf** which is the Iperf ARM executable which will be run on top of Zynq open source Linux. Right click on this file and select the **Copy** option which will place the selected file in the Virtual Machine clipboard.

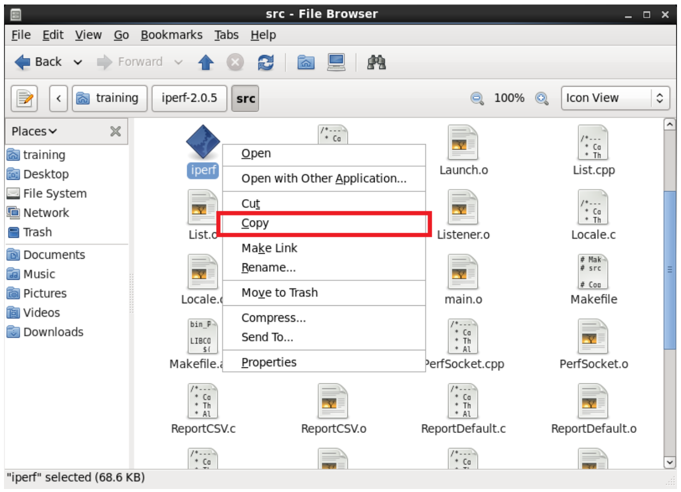


Figure 10 – Copying Iperf Executable to Virtual Machine Clipboard

13. Paste the Iperf executable into the MicroSD Card staging folder on the host operating system by using Windows Explorer to navigate to the following folder:

C:\Avnet\MicroZed\sd_image\

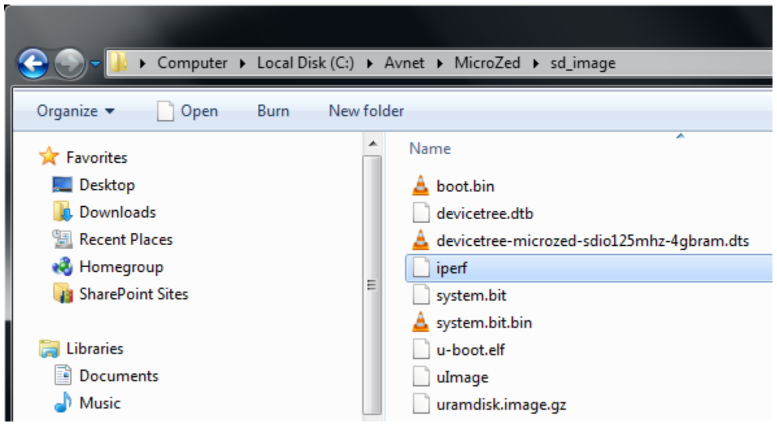


Figure 11 – Iperf Executable Copied to the Host Machine

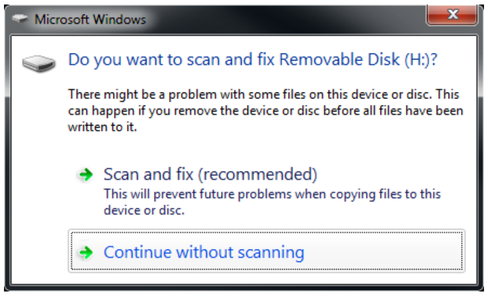
A basic MicroZed Linux platform is now staged in the above directory along with the Iperf executable which will be used for the remainder of the tutorial.

Setup MicroZed MicroSD Card

This section of the tutorial demonstrates how to setup a MicroZed MicroSD card to boot into an open source Linux platform.

1. If the Iperf executable was not built successfully in the previous section, a prebuilt executable can be copied from the folder **C:\Anvet\MicroZed\sw\iperf_microzed_osl_prebuilt\** to the MicroSD staging folder **C:\Avnet\MicroZed\sd_image\** before completing the remainder of this section of the tutorial.

2. Insert the MicroSD card into the PC or SD card reader and wait for it to enumerate as a Windows drive. If prompted by Windows when inserting the SD card, select the **Continue without scanning** option.



Microsoft Windows

Do you want to scan and fix Removable Disk (H:)?

There might be a problem with some files on this device or disc. This can happen if you remove the device or disc before all files have been written to it.

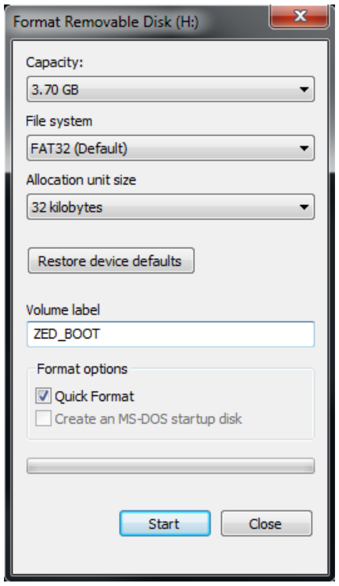
Scan and fix (recommended)
This will prevent future problems when copying files to this device or disc.

Continue without scanning

Figure 12 – Windows Prompt for Scanning and Fixing an SD Card

The Zynq BootROM is capable of interpreting the FAT32 file system for SD card boot mode. If the MicroSD card used is not already formatted for FAT32 file system, right click on the drive in Windows Explorer and select the **Format** option. Select the options shown in Figure 13 and click the **Start** button.

When the format process is complete, click the **Close** button.

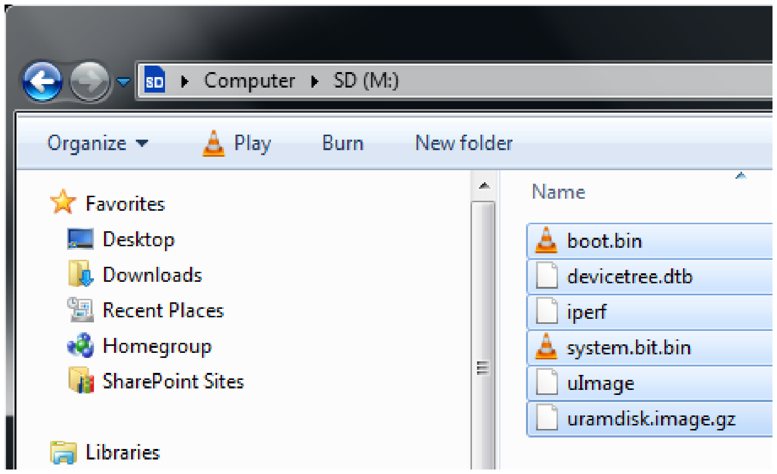


Format Removable Disk (H:)

Capacity: 3.70 GB
File system: FAT32 (Default)
Allocation unit size: 32 kilobytes
Restore device defaults
Volume label: ZED_BOOT
Format options
[✓] Quick Format
[ ] Create an MS-DOS startup disk
Start
Close

Figure 13 – Formatting the MicroSD Card with FAT32 File System

3. Copy the **boot.bin**, **devicetree.dtb**, **iperf**, **system.bit.bin**, **uImage**, and **uramdisk.image.gz** files from the **sd_image** staging folder to the top level of the MicroSD card. Replace any existing versions of these files that may already be on the MicroSD card.



boot.bin
devicetree.dtb
iperf
system.bit.bin
uImage
uramdisk.image.gz

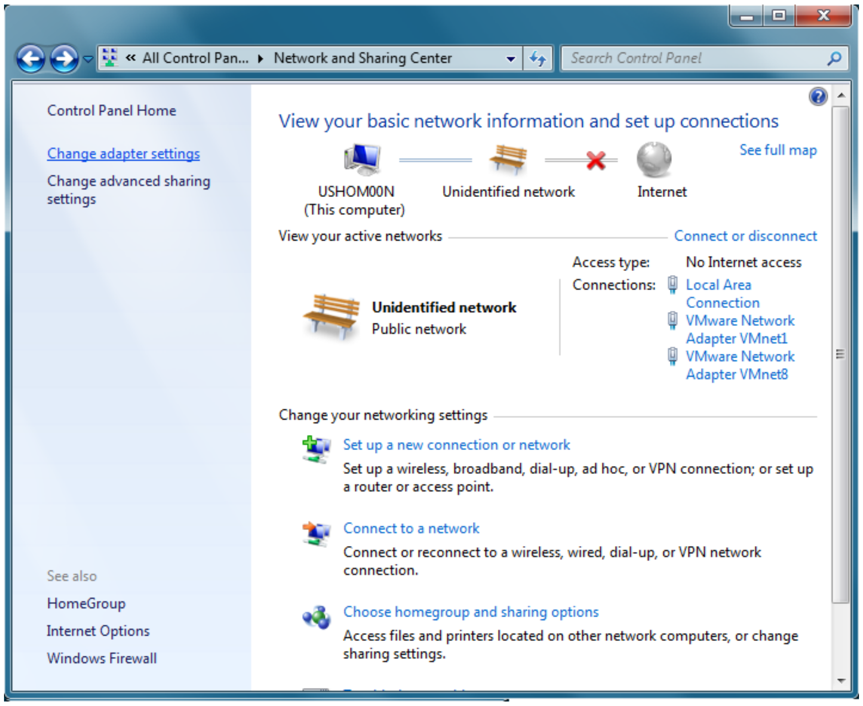
Figure 14 – The MicroZed Linux Platform Files Copied to the MicroSD Card

Once these files are copied to the MicroSD card, eject the MicroSD card from the PC or SD card reader.

Host PC Networking Configuration

This tutorial utilizes the Gigabit Ethernet hardware and networking capability of MicroZed. To complete this tutorial, you may have to configure the network properties on your PC. The following steps will guide you through this process for a Windows 7 host PC.

1. Attach a standard Ethernet Cable between MicroZed Gigabit Ethernet Port (J1) and the host PC network interface adapter.
2. Open the **Change adapter settings** from the **Start→Control Panel→Network and Sharing Center**.



View your basic network information and set up connections

View your active networks

Unidentified network
Public network

Access type: No Internet access
Connections: Local Area Connection
VMware Network Adapter VMnet1
VMware Network Adapter VMnet8

Change your networking settings

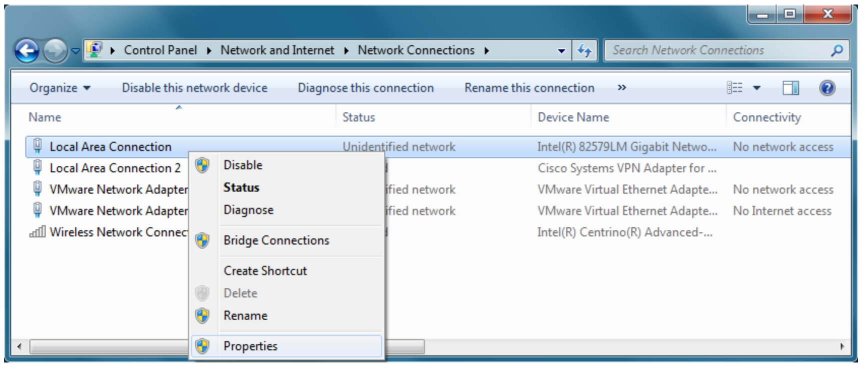
Set up a new connection or network
Set up a wireless, broadband, dial-up, ad hoc, or VPN connection; or set up a router or access point.

Connect to a network
Connect or reconnect to a wireless, wired, dial-up, or VPN network connection.

Choose homegroup and sharing options
Access files and printers located on other network computers, or change sharing settings.

Figure 15 – Network and Sharing Center

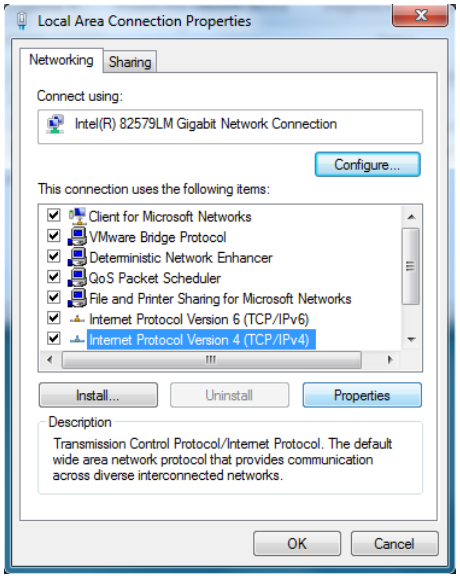
3. In the **Network Connections** window, right-click on the Local Area Connection adapter entry corresponding to the network interface that is connected to MicroZed and select **Properties**.



Name	Status	Device Name	Connectivity
Local Area Connection	Unidentified network	Intel(R) 82579LM Gigabit Netwo...	No network access
Local Area Connection 2		Cisco Systems VPN Adapter for ...	
VMware Network Adapter	ified network	VMware Virtual Ethernet Adapte...	No network access
VMware Network Adapter	ified network	VMware Virtual Ethernet Adapte...	No Internet access
Wireless Network Connec		Intel(R) Centrino(R) Advanced-...	

Figure 16 – Network Connections

4. In **Local Area Connection Properties**, select **Internet Protocol Version 4 (TCP/IPv4)**, then click the **Properties** button.



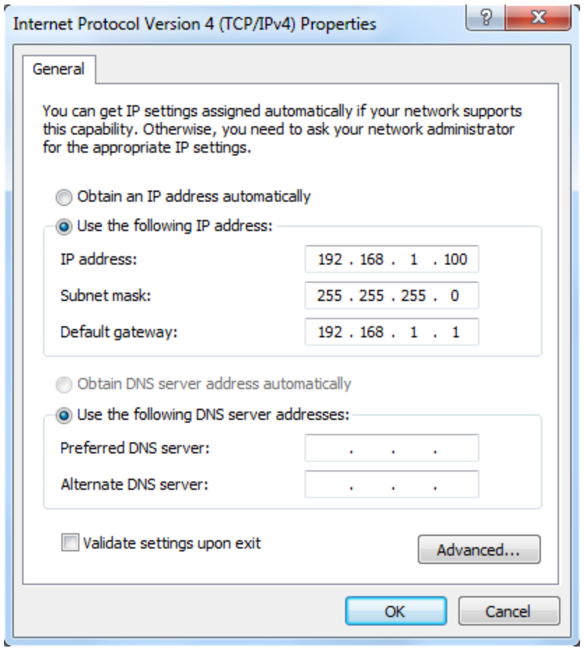
Connect using:
Intel(R) 82579LM Gigabit Network Connection

This connection uses the following items:
- [✓] Client for Microsoft Networks
- [✓] VMware Bridge Protocol
- [✓] Deterministic Network Enhancer
- [✓] QoS Packet Scheduler
- [✓] File and Printer Sharing for Microsoft Networks
- [✓] Internet Protocol Version 6 (TCP/IPv6)
- [✓] Internet Protocol Version 4 (TCP/IPv4)

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Figure 17 – Local Area Connection Properties

5. Set the IP address to 192.168.1.100 and the Subnet mask to 255.255.255.0 in the **Internet Protocol Version 4 (TCP/IPv4) Properties** window and then click the OK button.



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.

Obtain an IP address automatically [✓ Use the following IP address:]

IP address:	192 . 168 . 1 . 100
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	192 . 168 . 1 . 1

Obtain DNS server address automatically [✓ Use the following DNS server addresses:]

Preferred DNS server:	. . .
Alternate DNS server:	. . .

Validate settings upon exit

Figure 18 – Internet Protocol Version 4 (TCP/IPv4) Properties

The host PC networking is now configured and ready to proceed with the remainder of the tutorial.

Boot MicroZed Using New MicroSD Card Image

This section of the tutorial demonstrates how to boot MicroZed using the MicroSD card that was setup from the previous section.

1. Insert the 4GB MicroSD card included with MicroZed into the MicroSD card slot (J6) located on the underside of MicroZed module.

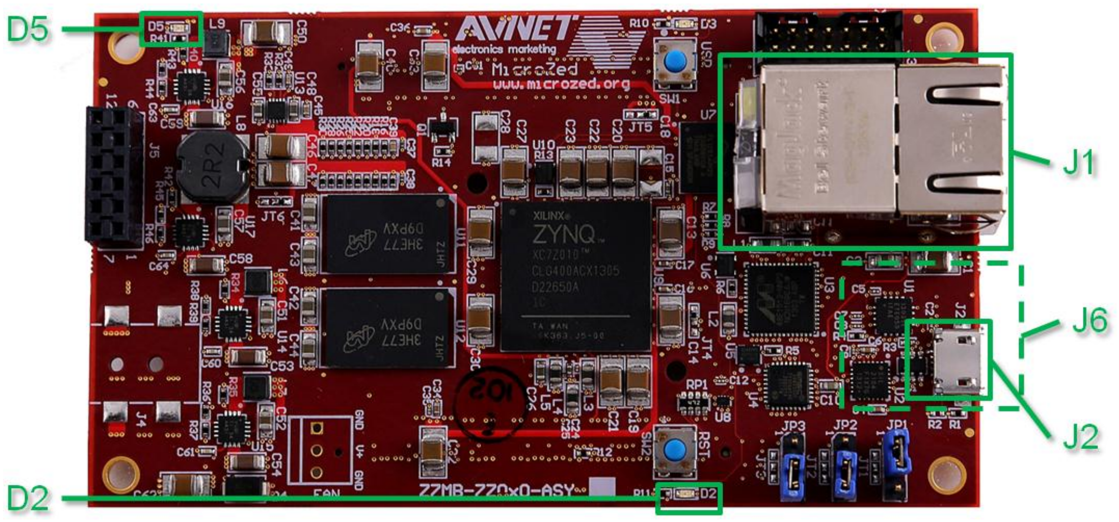


Figure 19 – MicroZed Hardware Reference

2. Verify the MicroZed boot mode (JP3-JP1) jumpers are set to SD card mode as shown in Figure 5. Further boot mode options are described in the MicroZed Hardware Users Guide.

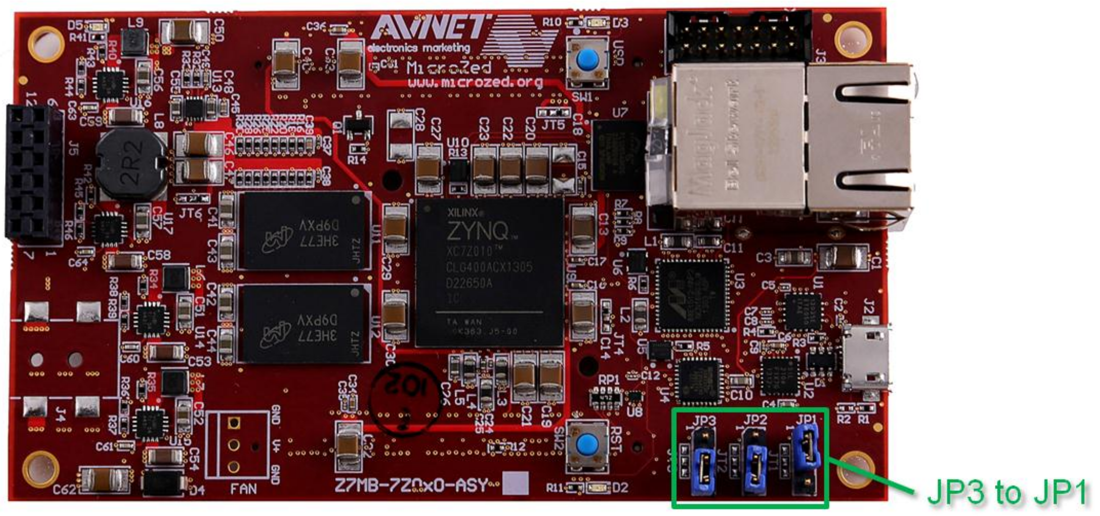


Figure 20 – MicroZed Jumper Settings

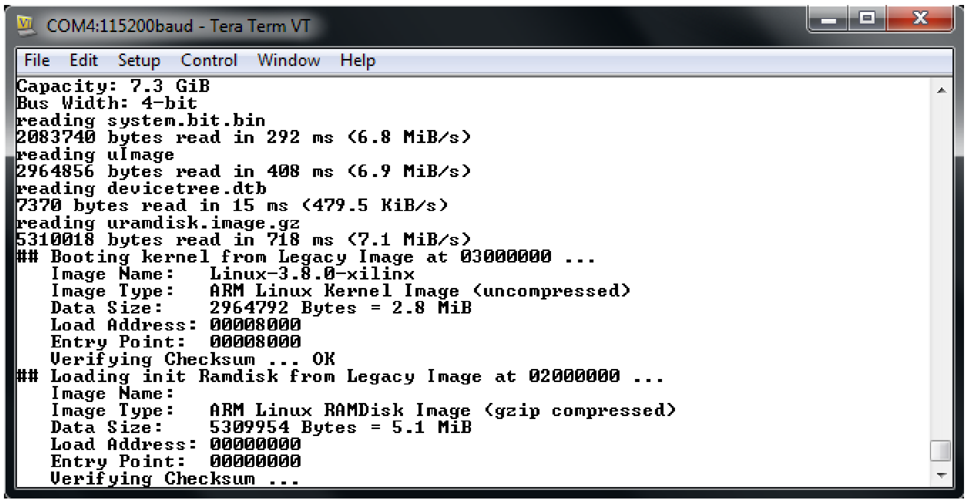
3. Connect the USB-UART port of MicroZed (J2) to a PC using the MicroUSB cable. MicroZed will power on and the Green Power Good LED (D5) should illuminate.

4. Wait approximately 7 seconds. The blue Done LED (D2) should illuminate.

5. On the PC, if a serial terminal session is not already open, open a serial terminal program. Tera Term was used to show the example output for this lab document.

Figure 21 – Tera Term Icon

6. When the terminal output from U-Boot and a countdown is observed, allow the countdown to expire.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Capacity: 7.3 GiB
Bus Width: 4-bit
reading system.bit.bin
2083740 bytes read in 292 ms (6.8 MiB/s)
reading uImage
2964856 bytes read in 408 ms (6.9 MiB/s)
reading devicetree.dtb
7370 bytes read in 15 ms (479.5 KiB/s)
reading uramdisk.image.gz
5310018 bytes read in 718 ms (7.1 MiB/s)
## Booting kernel from Legacy Image at 03000000 ...
   Image Name:   Linux-3.8.0-xilinx
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2964792 Bytes = 2.8 MiB
   Load Address: 00008000
   Entry Point:  00008000
   Verifying Checksum ... OK
## Loading init Ramdisk from Legacy Image at 02000000 ...
   Image Name:
   Image Type:   ARM Linux RAMDisk Image (gzip compressed)
   Data Size:    5309954 Bytes = 5.1 MiB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ...
```

Figure 22 – MicroZed U-Boot Booting Linux

7. When the Linux command prompt is reached, the FAT32 file system on the MicroSD card should already be mounted by the startup script.

   The first partition on the SD card shows up as mmcblk0p1 in the device tree. This device represents MMC block device 0, partition 1. This device should already be mounted to the mount point **/mnt** and the working directory needs to be changed to that folder so that the FAT32 file system can be accessed.

```
# cd /mnt/
```

8. Now that the FAT32 partition on the SD card has been mounted and the working directory changed to the **/mnt** mount point, we are ready to launch the Iperf client. Before the client is launched, the server on the PC side must already be running so that the IP traffic can be measured correctly.

   On the host PC, open a Windows Command Prompt session by clicking on the **Start→Accessories→Command Prompt** menu item.

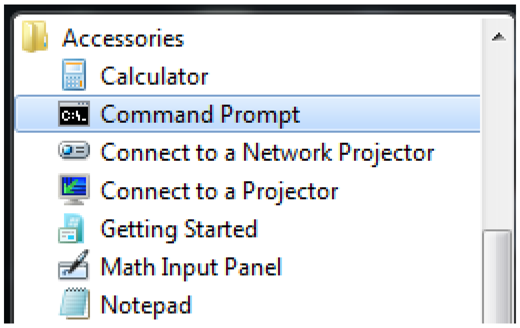
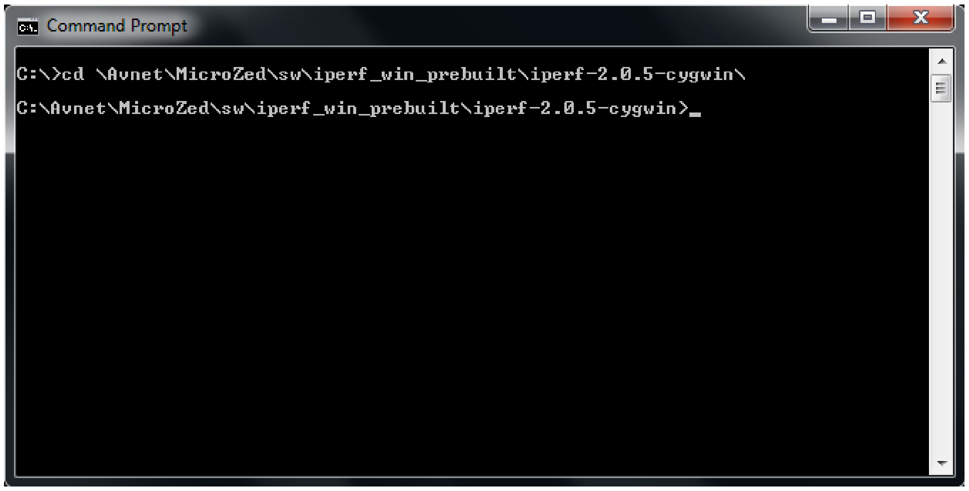


Figure 23 – Opening the Windows Command Prompt

9. Using the Windows Command Prompt window, change into the following working folder containing a prebuilt version of Iperf for Windows:

   C:\Avnet\MicroZed\sw\iperf_win_prebuilt\iperf-2.0.5-cygwin\



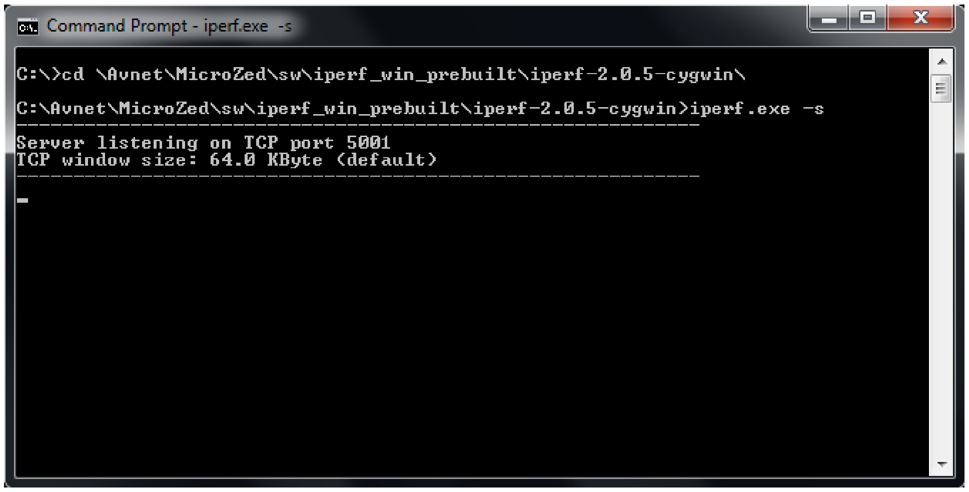
```
C:\>cd \Avnet\MicroZed\sw\iperf_win_prebuilt\iperf-2.0.5-cygwin\

C:\Avnet\MicroZed\sw\iperf_win_prebuilt\iperf-2.0.5-cygwin>_
```

Figure 24 – Changing the Working Folder

10. From the Windows command prompt, launch the Iperf application using server mode on the PC side. This will put the Iperf application on the host PC in a listening state, waiting for a client to connect and initiate performance test requests.

    If you observe a prompt from Windows Firewall, be sure to allow access to the Iperf application.

```
> iperf.exe -s
```



```
C:\>cd \Avnet\MicroZed\sw\iperf_win_prebuilt\iperf-2.0.5-cygwin\

C:\Avnet\MicroZed\sw\iperf_win_prebuilt\iperf-2.0.5-cygwin>iperf.exe -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
------------------------------------------------------------
_
```

Figure 25 – Launching the Iperf Application in Server Mode

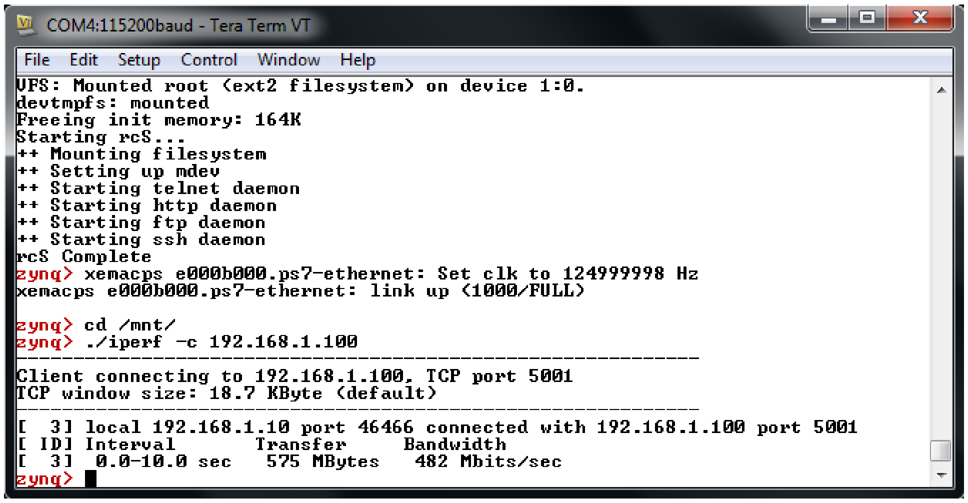
As a side note, for those wondering why a Cygwin binary is used, you can find more information on the Google Code page for iperf-cygwin:

https://code.google.com/p/iperf-cygwin/

According to this page, even though Iperf is a Linux application, "*Sometimes people would like to use Iperf 2.x on Windows but they don't have time to setup Cygwin. The objective of this project is to produce a Iperf 2.x Windows executable which can be used directly.*"

11. From the MicroZed Linux command prompt, launch the Iperf application and specify the IP address of the server PC. In this case, our PC IP address is 192.168.1.100 so this address is used to specify the remote Iperf server. This will put the Iperf application on the MicroZed in an active request state, connecting to the host PC to initiate the performance test requests.

```
# ./iperf -c 192.168.1.100
```


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
VFS: Mounted root (ext2 filesystem) on device 1:0.
devtmpfs: mounted
Freeing init memory: 164K
Starting rcS...
++ Mounting filesystem
++ Setting up mdev
++ Starting telnet daemon
++ Starting http daemon
++ Starting ftp daemon
++ Starting ssh daemon
rcS Complete
zynq> xemacps e000b000.ps7-ethernet: Set clk to 124999998 Hz
xemacps e000b000.ps7-ethernet: link up (1000/FULL)

zynq> cd /mnt/
zynq> ./iperf -c 192.168.1.100
------------------------------------------------------------
Client connecting to 192.168.1.100, TCP port 5001
TCP window size: 18.7 KByte (default)
------------------------------------------------------------
[  3] local 192.168.1.10 port 46466 connected with 192.168.1.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec    575 MBytes    482 Mbits/sec
zynq>
```

Figure 26 – Launching the Iperf Client Application on MicroZed

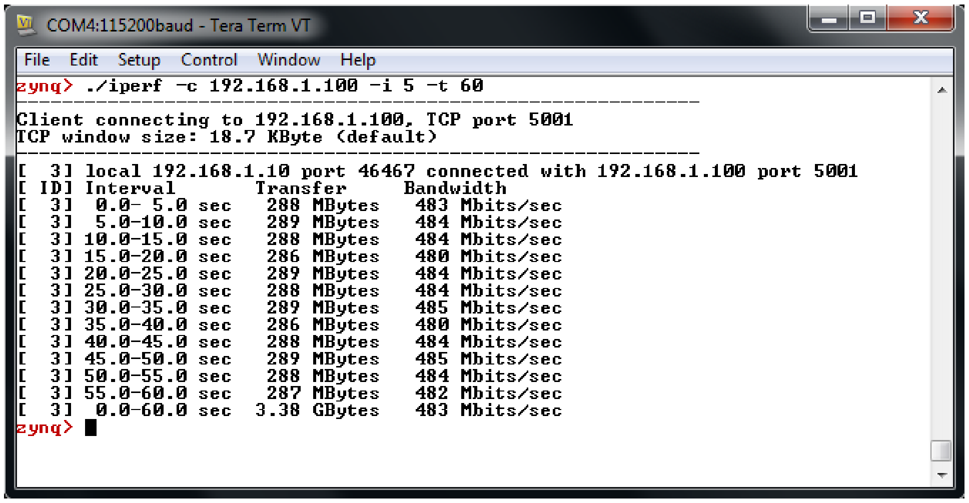
12. Observe the results shown by Iperf following the 10 second test interval. On the example system, during the 10 second test interval, 575 Megabytes of data were transferred between MicroZed and the test PC over a TCP connection for an average throughput of 482 Megabits per second.

    Keep in mind that your own results will vary based upon your network setup and the type of host machine you are running tests against. Adding a network switch to the Ethernet configuration or having several applications running in the background on the host PC can change the test results dramatically.

    Also keep in mind the amount of time spend on connection establishment overhead for the test over the default 10 second interval can vary and cause a reported decrease in overall throughput. By increasing the test duration, slightly better results may be observed during the later test intervals. To change the test interval, use the **-i** command line option and to change the total duration of the test, use the **-t** command line option.

    For example, on another PC configuration where only 94 Mbps were observed over the 10 second default interval, simply by reducing the interval to 5 seconds and increasing the test time to 60 seconds this resulted in a better overall reported throughput of 479 Mbps. This was done through the following command using the MicroZed Iperf client with results seen below in Figure 27:

```
# ./iperf -c 192.168.1.100 -i 5 -t 60
```


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
zynq> ./iperf -c 192.168.1.100 -i 5 -t 60
------------------------------------------------------------
Client connecting to 192.168.1.100, TCP port 5001
TCP window size: 18.7 KByte (default)
------------------------------------------------------------
[  3] local 192.168.1.10 port 46467 connected with 192.168.1.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.0 sec   288 MBytes   483 Mbits/sec
[  3]  5.0-10.0 sec   289 MBytes   484 Mbits/sec
[  3] 10.0-15.0 sec   288 MBytes   484 Mbits/sec
[  3] 15.0-20.0 sec   286 MBytes   480 Mbits/sec
[  3] 20.0-25.0 sec   289 MBytes   484 Mbits/sec
[  3] 25.0-30.0 sec   288 MBytes   484 Mbits/sec
[  3] 30.0-35.0 sec   289 MBytes   485 Mbits/sec
[  3] 35.0-40.0 sec   286 MBytes   480 Mbits/sec
[  3] 40.0-45.0 sec   288 MBytes   484 Mbits/sec
[  3] 45.0-50.0 sec   289 MBytes   485 Mbits/sec
[  3] 50.0-55.0 sec   288 MBytes   484 Mbits/sec
[  3] 55.0-60.0 sec   287 MBytes   482 Mbits/sec
[  3]  0.0-60.0 sec  3.38 GBytes   483 Mbits/sec
zynq> 
```

Figure 27 – Iperf Client Application Using Different Test Durations

13. Continue to experiment with Iperf to observe how the connection types can affect the transfer rate.

For further example test results see the document entitled “MicroZed - 1000Mbps Ethernet Performance Test Report.pdf”, included with this reference design tutorial.

14. Once the performance tests are complete, make sure the MicroSD card is un-mounted prior to powering off MicroZed. Un-mount the MicroSD card from the **/mnt** mount point.

```
# cd /
# umount /mnt/
```

Getting Help and Support

Avnet Support

MicroZed is a versatile development kit, with all technical support being offered through the MicroZed.org website support forums. MicroZed users are encouraged to participate in the forums and offer help to others when possible.

For questions regarding the MicroZed community website, please direct any questions to:

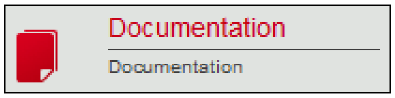
MicroZed.org Web Master – webmaster@microzed.org

To access the most current collateral for MicroZed please visit the community support page at:

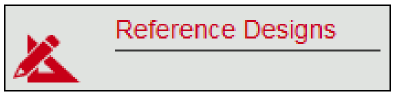
www.microzed.org/content/support

Once on the MicroZed.org support page:

To access the latest MicroZed documentation, click on the Documentation link:



To access the latest reference designs for MicroZed, click on the following link:



To access the MicroZed technical forums, click on the following link:

Xilinx Support

For questions regarding products within the Product Entitlement Account, send an e-mail 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

Appendix A: Troubleshooting

This section provides troubleshooting information for the MicroZed Open Source Linux Ethernet Performance Test Tutorial.

Troubleshooting the MicroZed Network Connection

The Basic network configuration for the MicroZed Open Source Linux Ethernet Performance Test Tutorial is shown below:

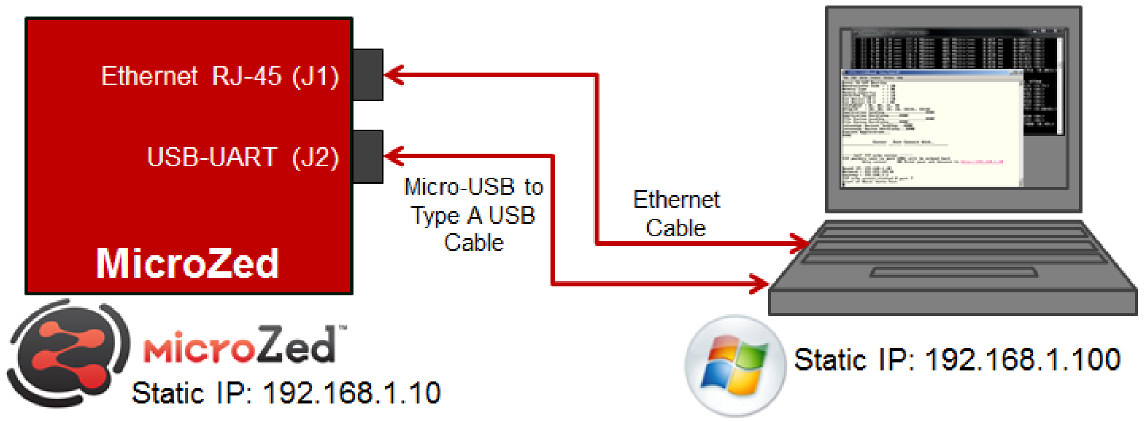


Figure 28 – Connecting the MicroZed USB-UART and Ethernet Cable

Make sure that Ethernet adapter on host PC Windows is configured as follows:

- IPv4 Address: 192.168.1.100
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1

Make sure the wireless internet adapter of the PC is disabled otherwise there may be a routing conflict that prevents the Zynq Linux host from being reached.