

# UltraZed PCIe Carrier Card + FMC-Network lwIP

## Overview

A more advanced topic that can be investigated is Ethernet connectivity. Lightweight IP (lwIP) is an open source TCP/IP networking stack for embedded systems. The Xilinx® Software Development Kit (SDK) provides lwIP software customized to run on various Xilinx embedded systems. This document describes how to use the lwIP library to add networking capability to an embedded system. In particular, lwIP is utilized to develop an echo server. Additional applications can be investigated by reading Xilinx Application Note [XAPP1026: LightWeight IP Application Examples](#).

## Objectives

When this tutorial is complete, you will be able to:

- Customize and export a hardware design implementing GEM Ethernet interfaces
- Create a bare metal BSP using a custom lwIP library
- Run and explore the lwIP

## Experiment Setup

### Software

The software used to test this reference design is:

- Windows-7 64-bit
- Xilinx Vivado 2018.2
- Xilinx SDK 2018.2

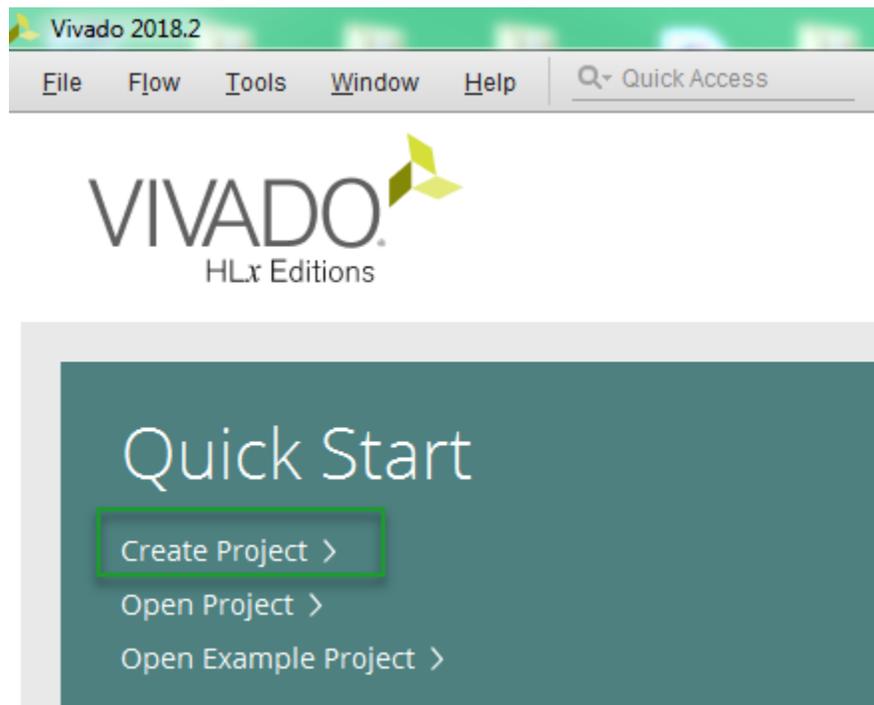
### Hardware

The hardware setup used to test this reference design includes:

- Win-7 PC
- UltraZed SOM
- UltraZed PCIe Carrier Card
- AES-FMC-NETW1-G - [Link](#)
- 2x USB cable (Type A to Micro-USB Type B)
- Ethernet Cable

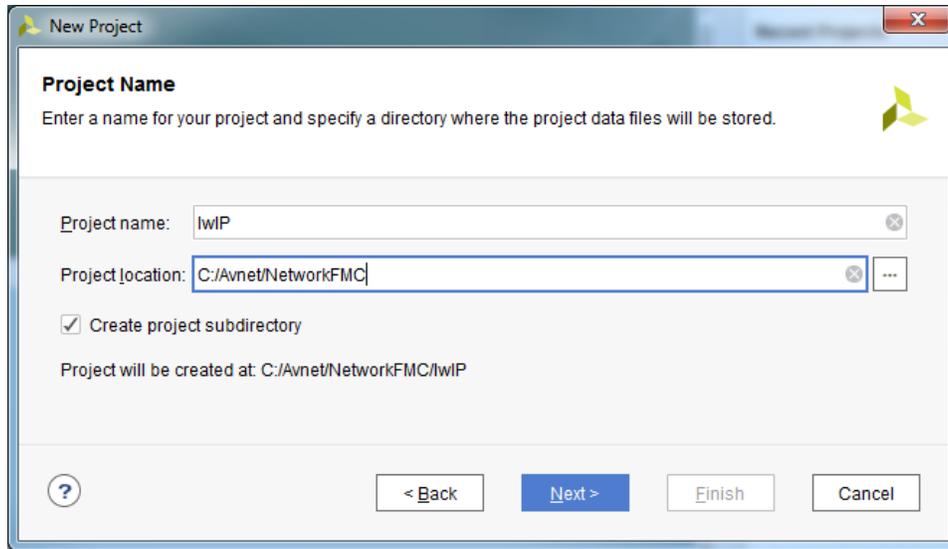
## Experiment 1: Create Hardware Design to implement GMII to RGMII

1. Install the folder Avnet at you C: drive
2. Launch Vivado by selecting **Start → All Programs → Xilinx Design Tools → Vivado 2018.2 → Vivado 2018.2.**
3. Select **File → Create Project** or use the Quick Start method



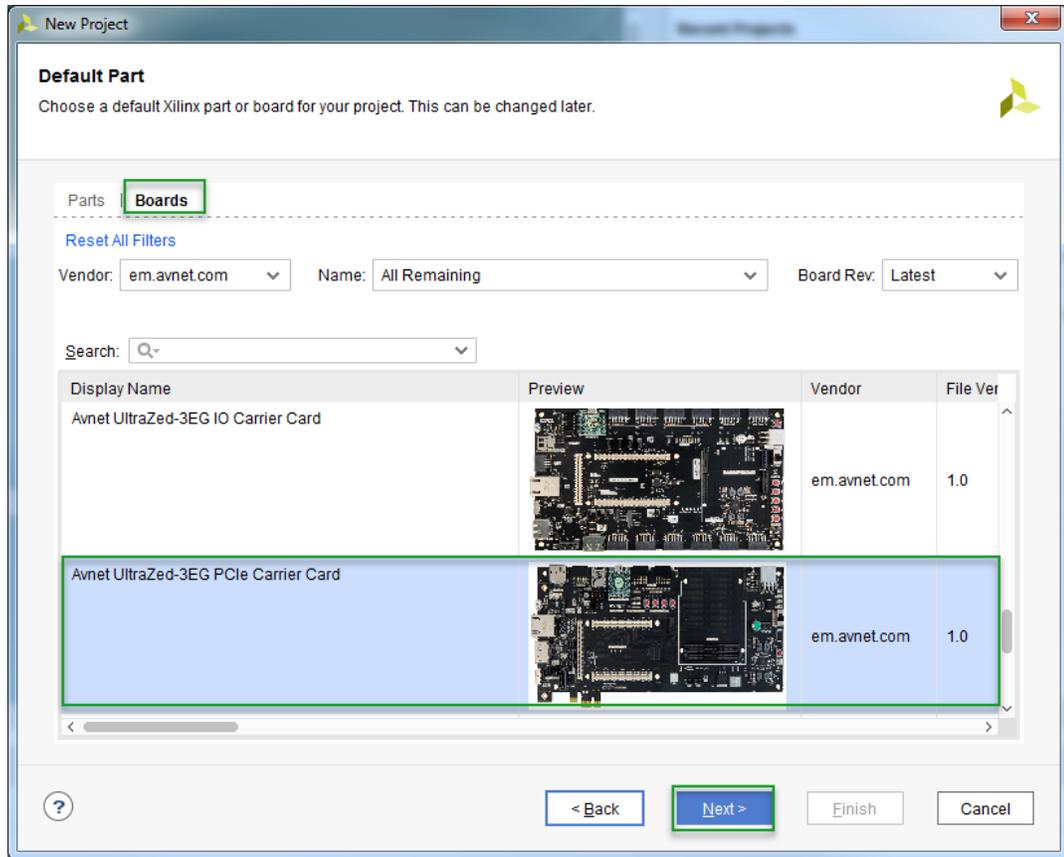
**Figure 1 – Vivado Launched**

4. Select Next and then enter the Project name and location below as seen in Figure 2.



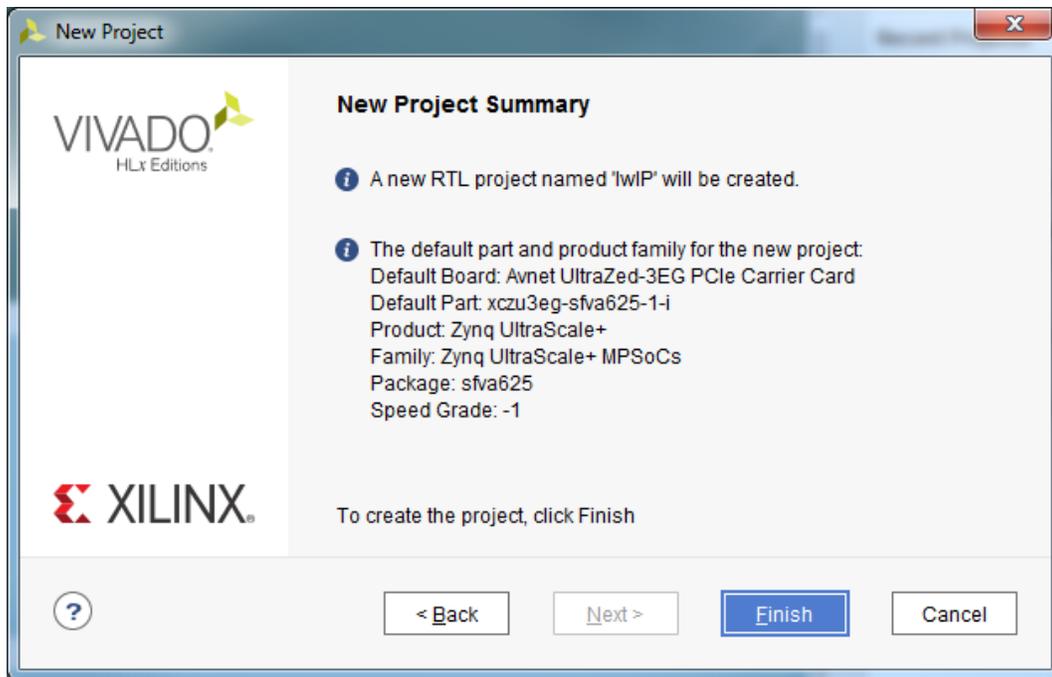
**Figure 2 -- Project Name**

5. After entering Project name and Project location select **Next** until you reach the Default part selection page. When here select the boards tab and scroll down and select **Avnet UltraZed-3EG PCIe Carrier Card**. Select Next. Please see Figure 3.
6. Note if you do not see the Board Definition file for the UltraZed PCIe Carrier Card you may not have installed them as of yet. Please refer to the following link and install the latest release of Board Definition Files.  
<http://zedboard.org/support/documentation/14201>



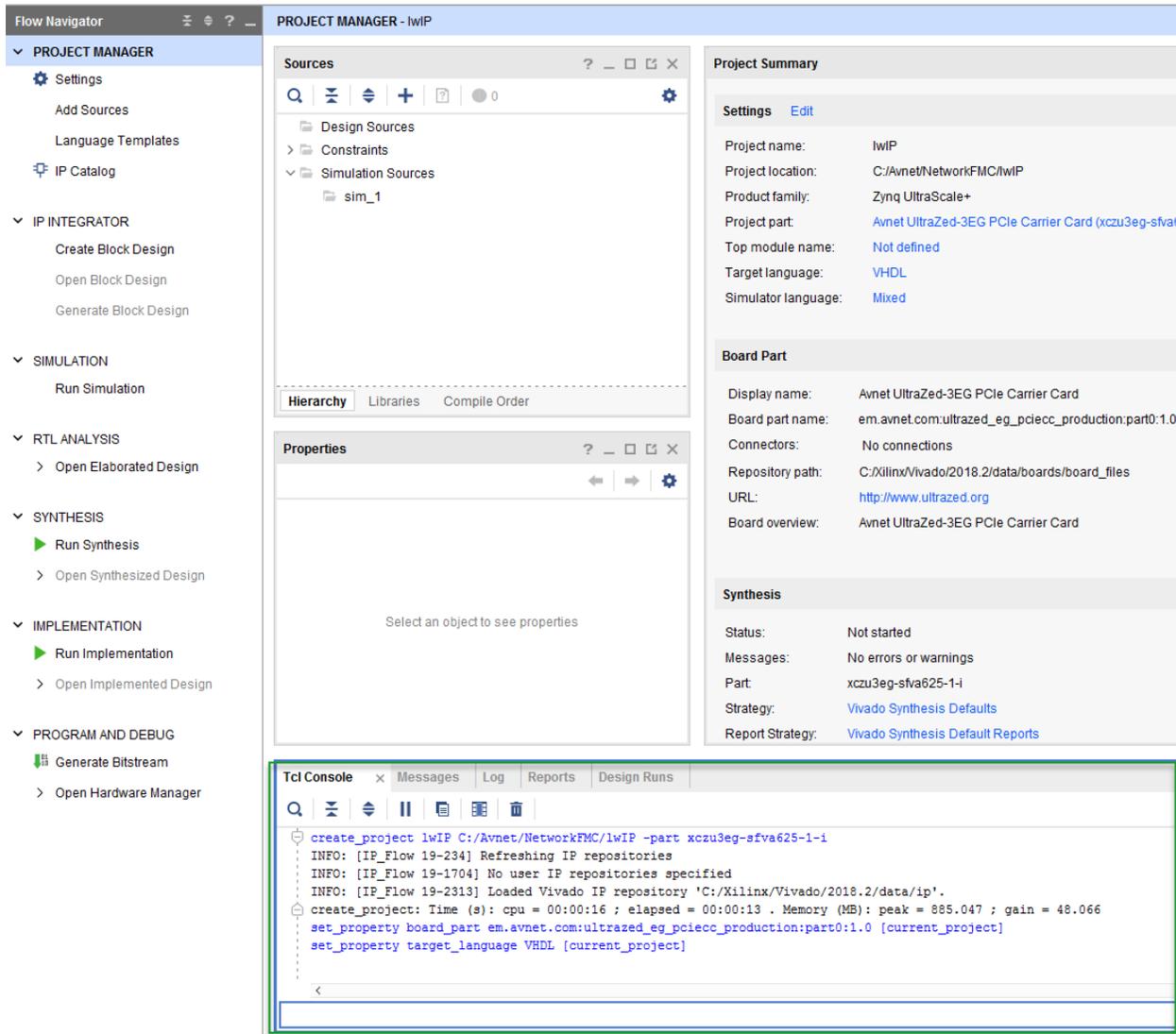
**Figure 3 -- Board Definition File Selection**

7. You will now be at the New Project Summary page. Verify all information matches Figure 4 and select Finish.



**Figure 4 -- Project Summary**

8. You have successfully created your project, now we will use a Tcl file to source our design.
9. Open the Tcl Console at the bottom of the page.



**Figure 5 -- Open Tcl Console**

10. Now that the Tcl Console is open type in the following two commands

**cd C:/Avnet/NetworkFMC/SupportingDocs**

**source ./design\_1.tcl**

11. Note: The above commands will only work if you followed step 1 of this Tutorial.

12. You will now see our block diagram is properly connected to use the gmii to rgmii interfaces.



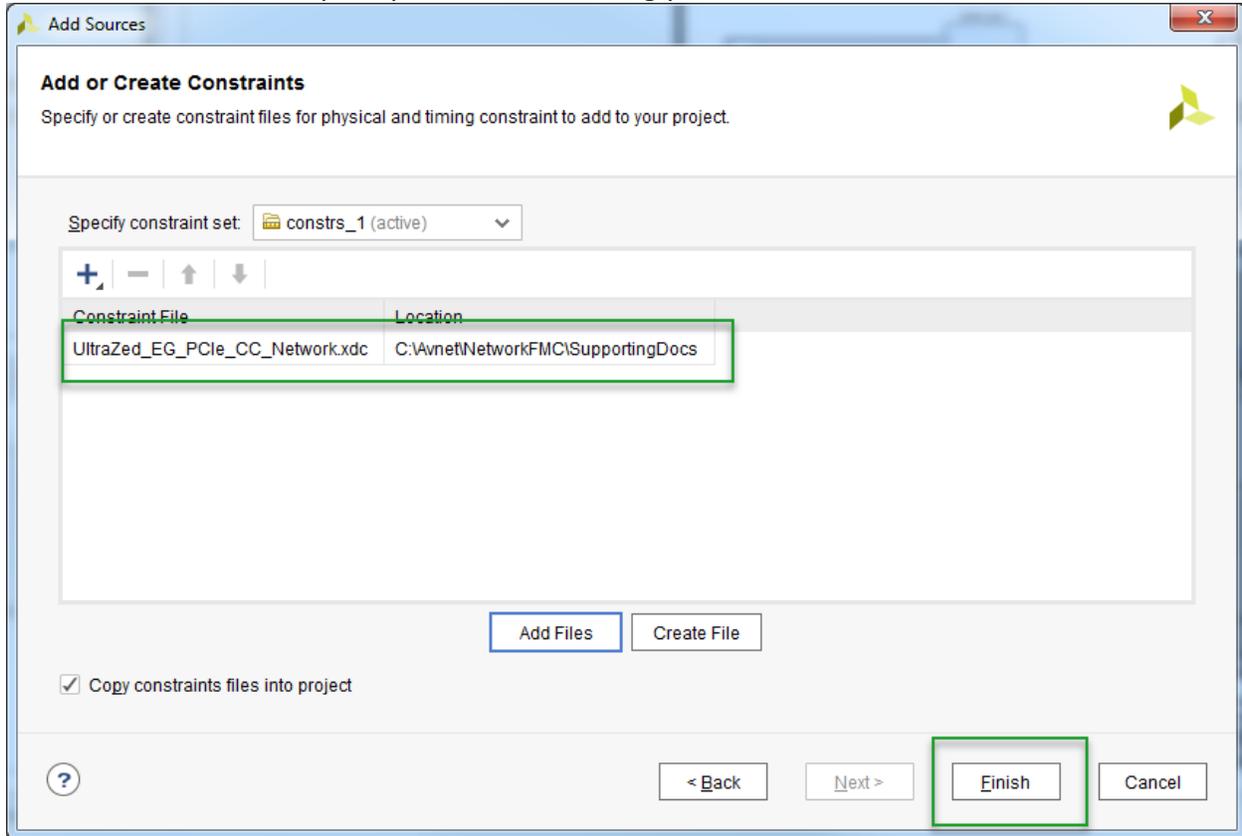


**Figure 7 -- Add Sources**

14. Make sure **Add or create constraints** is selected and hit **next**

15. Select Add files and now navigatge to the SupportingDocs foler located at **C:\Avnet\NetworkFMC\SupportingDocs** and then select the UltraZed\_EG\_PCl\_e\_CC\_Network.xdc constraint file then select OK

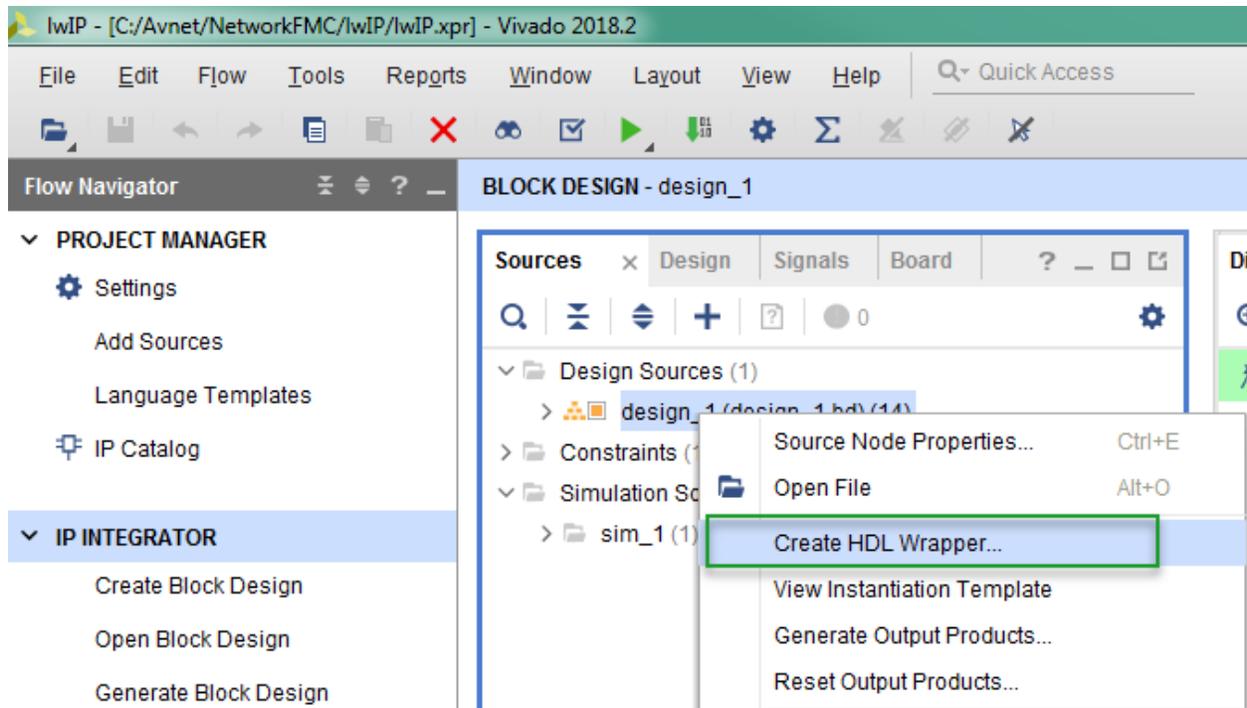
16. You will now be prompted to finish adding your constraints, if all is correct select **Finish**.



**Figure 8 -- Add Constraints**

17. Now that we have added our constraints file, one more thing we need to do before we generate our bitstream is to create a Top Level HDL wrapper.

18. Select the Sources Tab, Right click on design\_1(design\_1.bd) and select **Create HDL Wrapper**. Then select OK to default configuration.



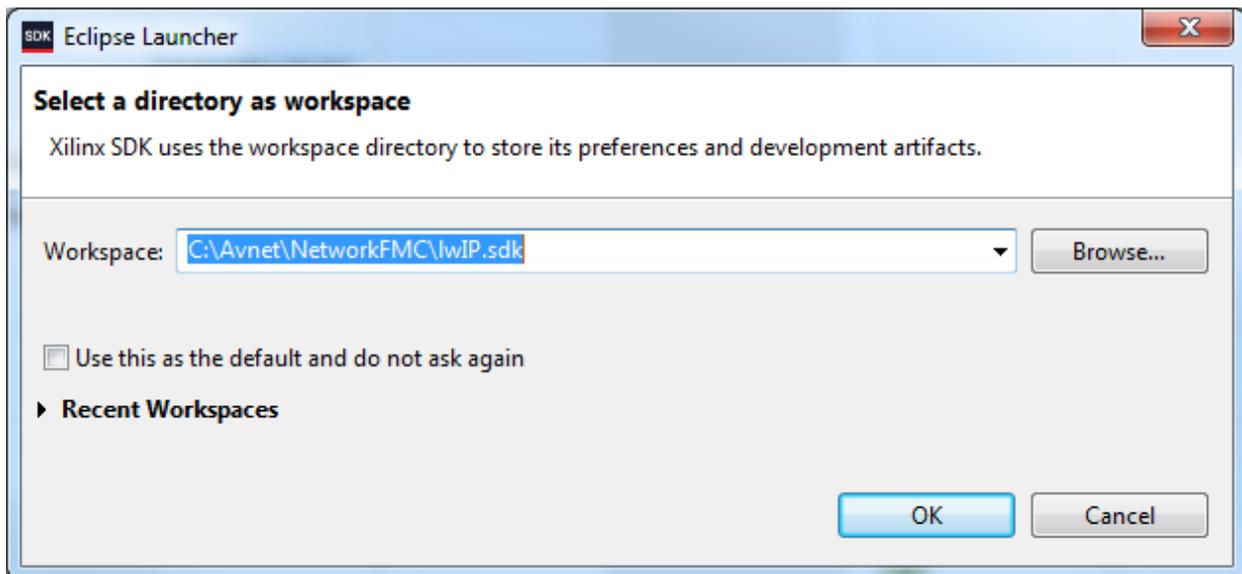
**Figure 9 -- Create HDL Wrapper**

19. Feel free to explore the block design and constraints file at this point to see what we did to generate this hardware platform
20. Now you may generate the bitstream if you wish, however this will take roughly 30-40 minutes pending your computer. In the SupportingDocs file you have the .hdf file that is generated during the Bitstream generation for your use if time is sensitive.
21. Select Generate Bitstream in the Flow Navigator, Open implemented design when prompted and then export the Bitstream and HDF file to be used in Experiment 2.
22. Note any timing warnings may be ignored.
23. Proceed to Experiment 2.

## Experiment 2: Launch SDK and create Application

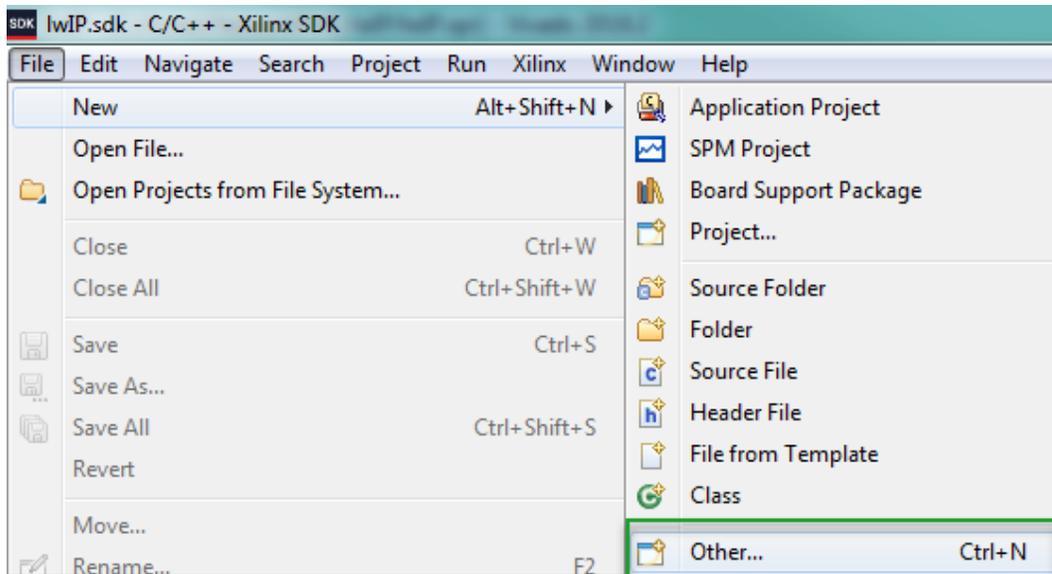
Now that we've modified our embedded system, we must make this platform available to the Software Development Kit (SDK). This is done by exporting the hardware platform.

1. Open SDK 2018.2 by going **All Programs > Xilinx Design Tools > SDK 2018.2 > Xilinx SDK 2018.2** .
2. Now you are prompted, set the Xilinx SDK Workspace to **C:\Avnet\NetworkFMC\lwIP.sdk** and select OK. You will now be at the Welcome page. Exit out of the welcome page by clicking the tab on the top left of SDK



**Figure 10 -- Workspace**

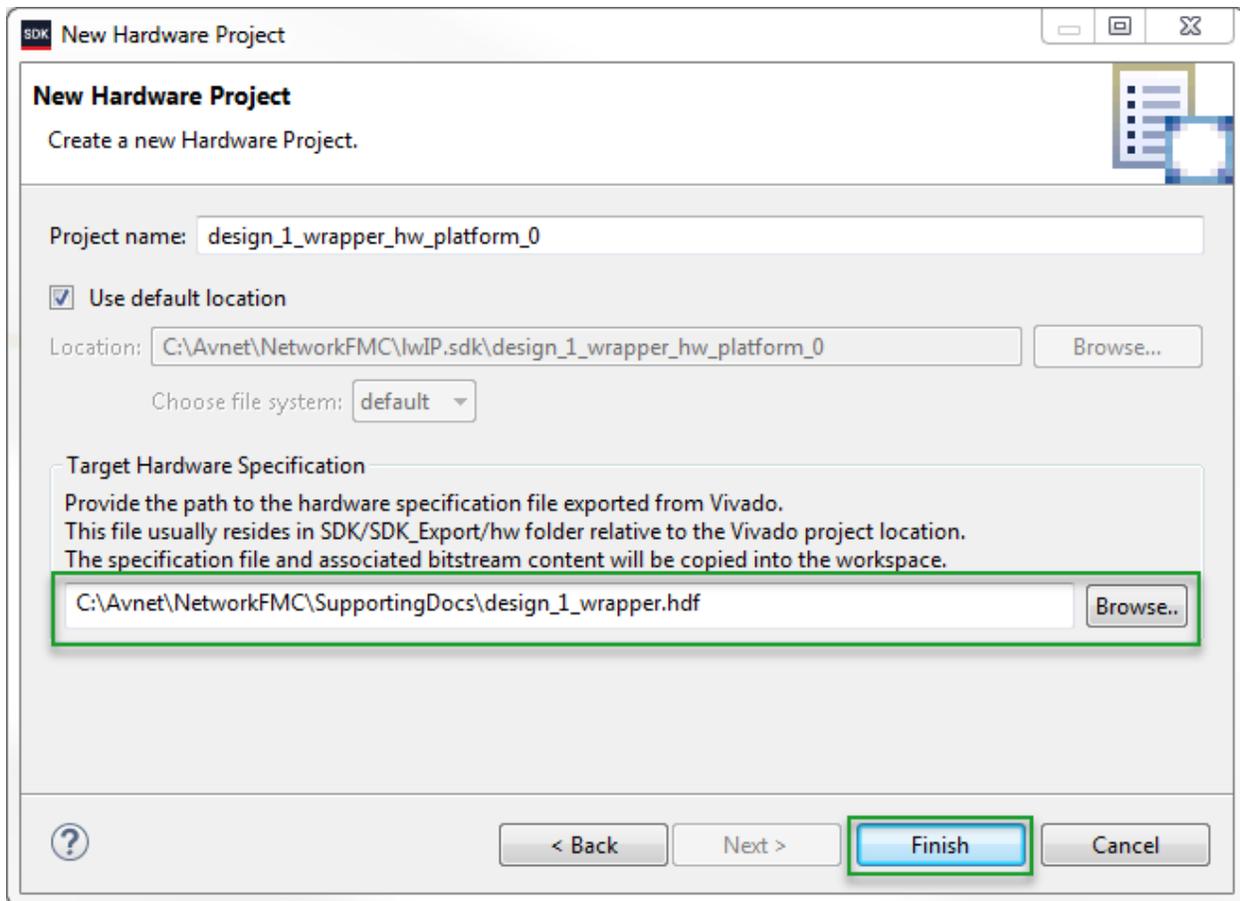
3. We now need to import the HDF file we generated in lab 1, select **File > New > Other**.



**Figure 11 -- Importing HDF File**

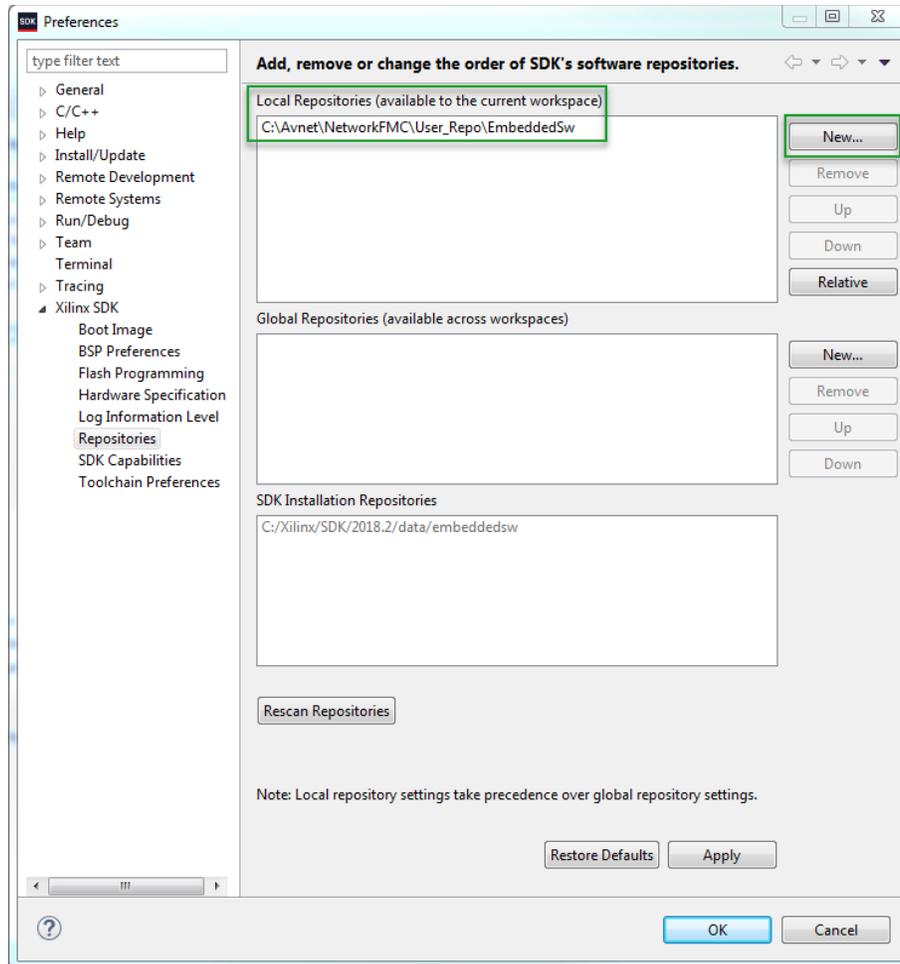
4. Now Select **Hardware Platform Specification** by selecting **Xilinx > Hardware Platform Specification**. Select **Next**
5. Now you are prompted to browse for the HDF file. Either browse to the HDF file you generated in Lab 1 or select the pre-built HDF file located at the follow directory. Afterwards select **Finish**.

**C:\Avnet\NetworkFMC\SupportingDocs\design\_1\_wrapper.hdf**



**Figure 12 -- Import New Hardware Project**

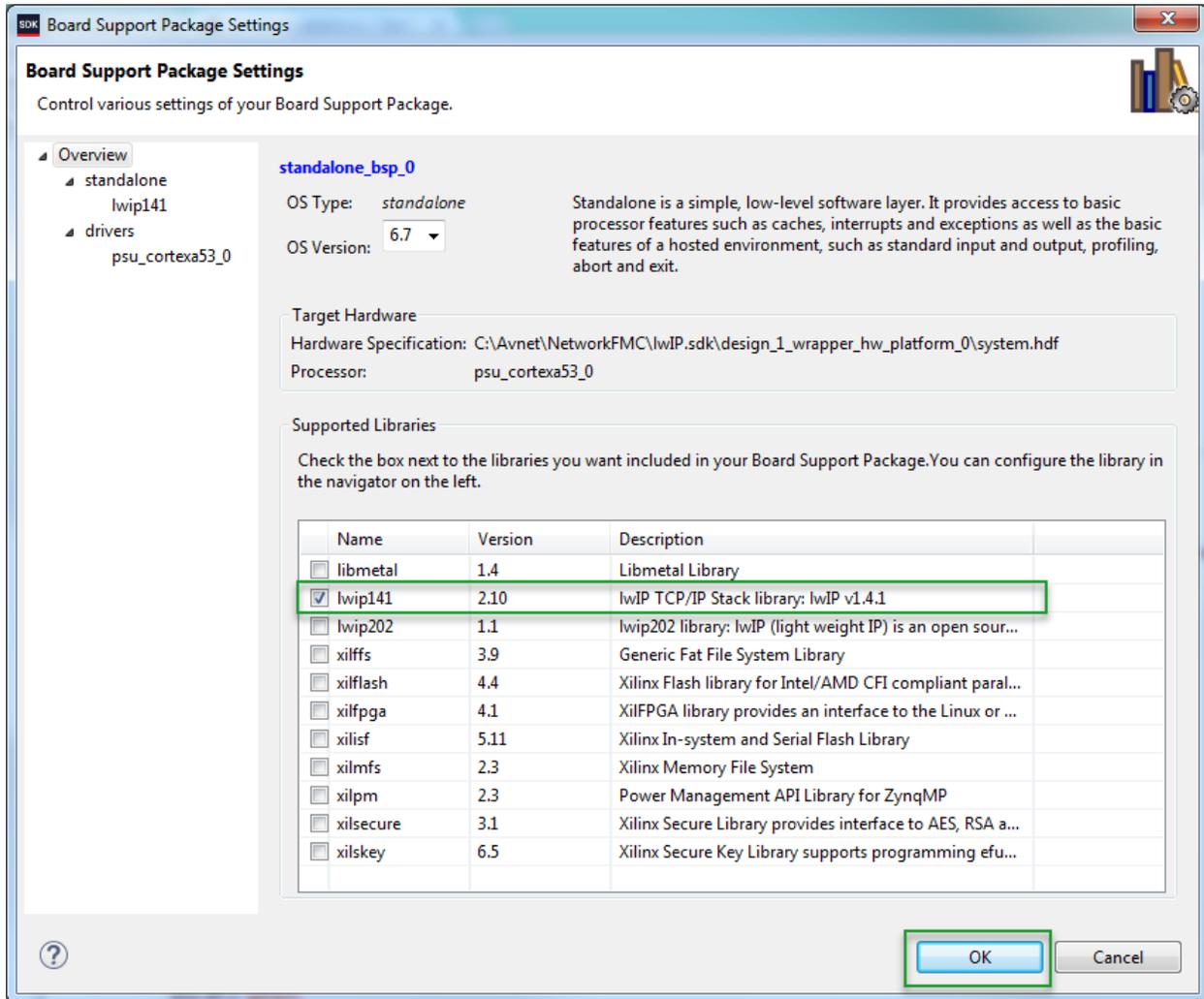
6. Note: Project Name will automatically populate after your browse and select the file.
7. You will now see the HDF file on the left hand side of SDK under the Project Explorer Window
8. Next, you must import the lwIP repository that was provided with this lab. The reason this must be done is that as of SDK 2018.2 the Microchip PHY KSZ9031RNX on the Network FMC is not supported natively within the built in lwIP driver.
9. In the Xilinx SDK menu tabs click on **Xilinx > Repositories**. This will open up a page that will allow you to remove, add, or change SDK software repositories.
10. Select **New** at the top right of the page and go to the following directory and select **OK**



**Figure 13 - Add Local Repository**

11. Note: If repository directory is not exactly the same as seen in image above please go back and correct this else it will not work correctly.
12. Select Apply at the bottom of the page and then OK to close the page.
13. At this point the updated lwIP driver should be added to SDK project.
14. Next we will create the lwIP Board Support Package. In the SDK tabs window select **File > New > Board Support Package**
15. You will be Prompted to add a Project name, in this case we can leave it as default and then select Finish
16. You will then be prompted to Modify the Board Support Package Settings. Enable the lwip141 Version 2.10 driver by checking the box under the name section. Then select OK to finish creating your Board Support Package.

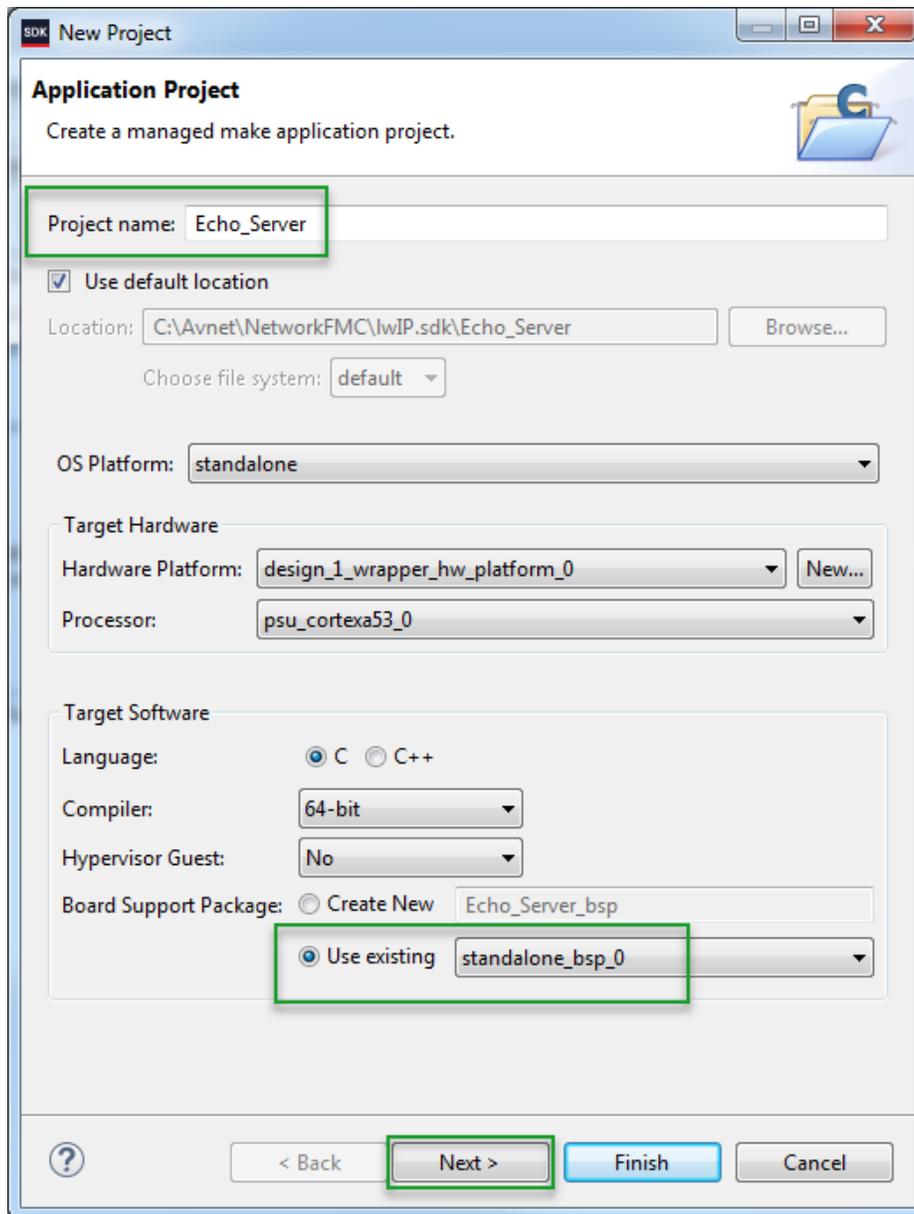
17. Note: If you do not see lwip141 version 2.10, please go back and double check you installed the local repository correctly.



**Figure 14 -- Enable lwIP driver**

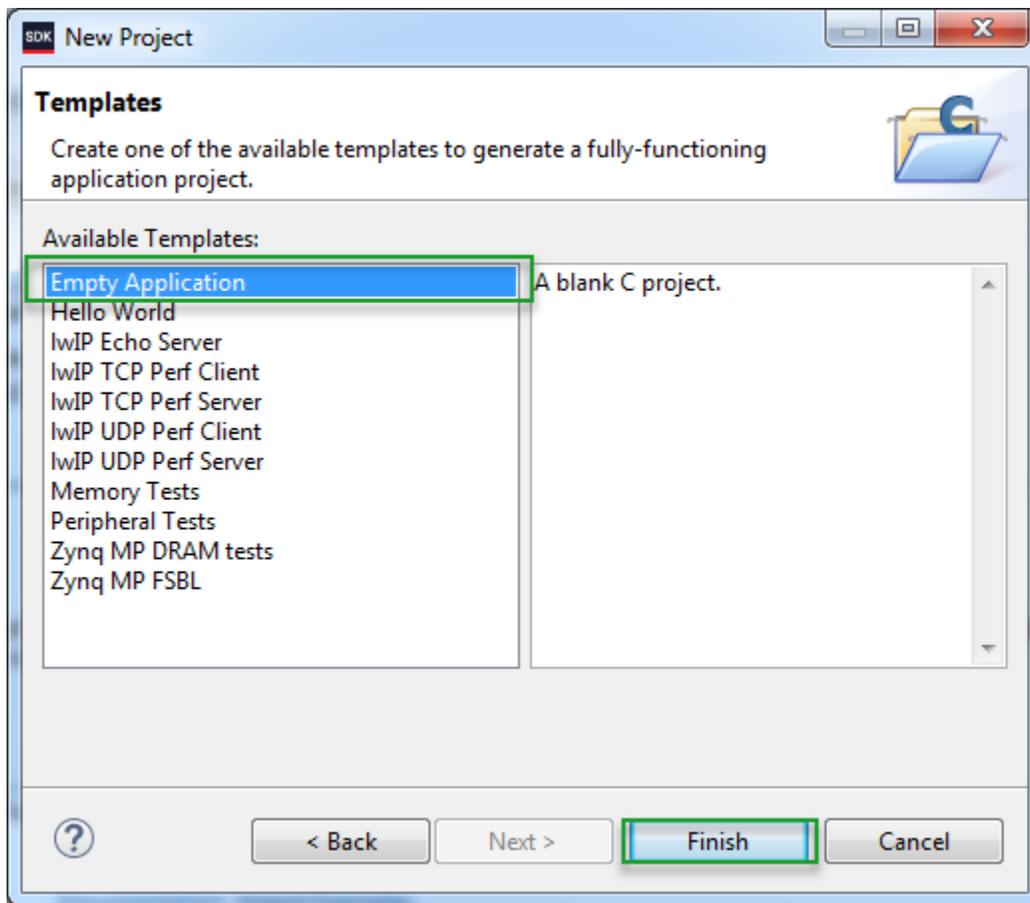
18. Next we will create the lwIP application. In the SDK tabs window select **File > New > Application Project**

19. Label the **Project name : Echo Server**, make sure **Use Existing** is selected for Board Support Package and select **Next**



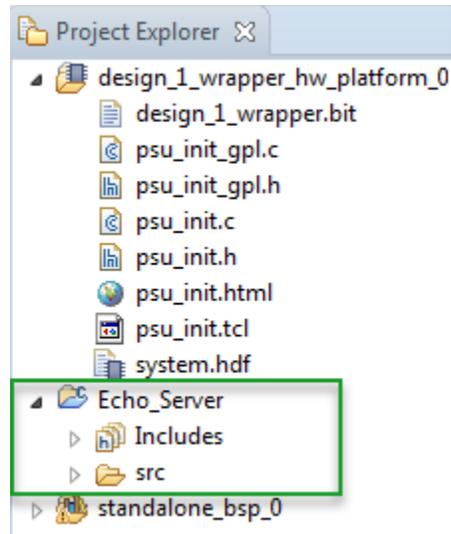
**Figure 15 -- New Application Project**

20. You are now prompted to select from one of the Available Templates. Select **lwIP Echo Server** and then select **Finish**.



**Figure 16 -- Project Templates**

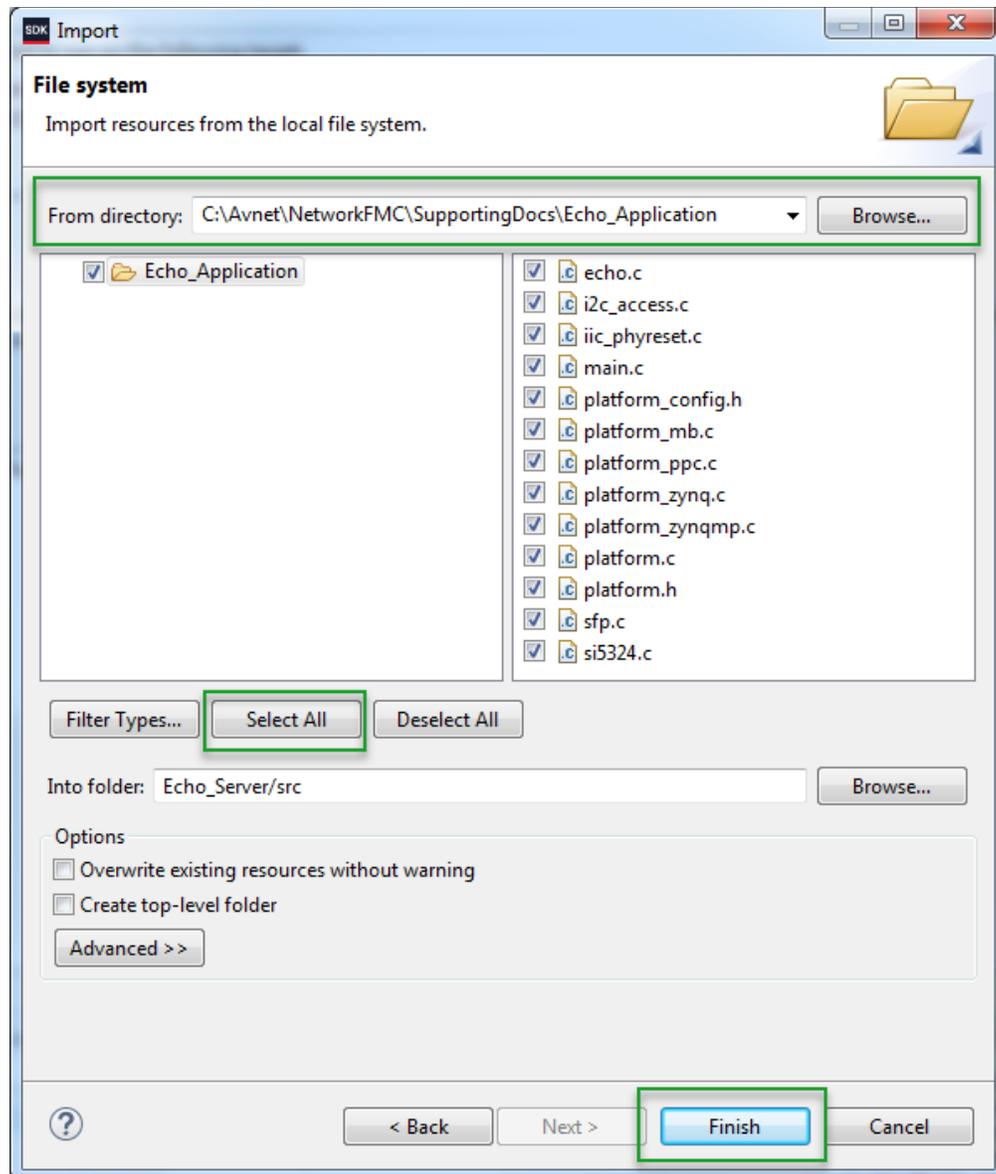
21. When your workspace is finished building look at the Project Explorer tab, you will now see the Application Project and Board Support Package you had just created.
22. In the Project Explorer tab. Expand the Echo\_Server application.



**Figure 17 -- Application**

23. Right Click on the SRC tab and select **Import**. At this point we are going to import the application source files provided with this lab into our new empty application.
24. The Import wizard will open Select **General > File System** and then select **Next>**.
25. Browse to the following directory, hit Select All, and then Finish. See figure below.

**C:\Avnet\NetworkFMC\SupportingDocs\Echo\_Application**



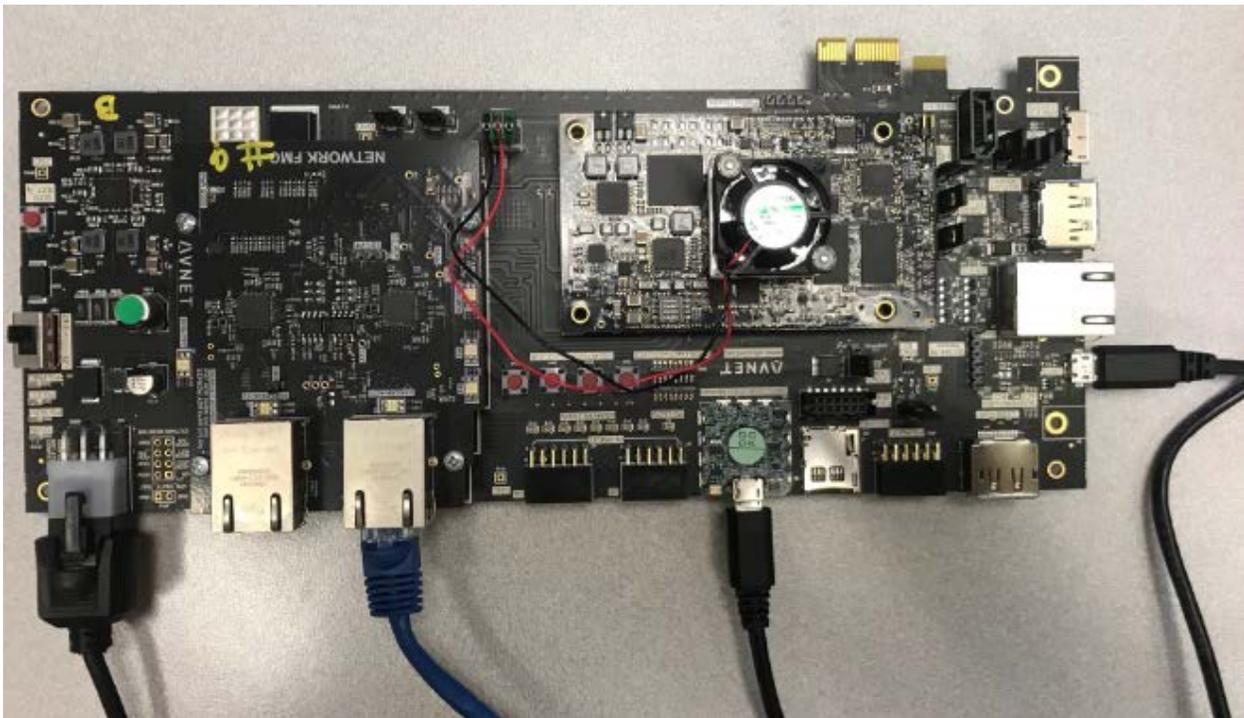
**Figure 18 -- Import Application**

26. Your application is complete

27. Next Step is to connect up your hardware and run the application. Please proceed to Experiment 3.

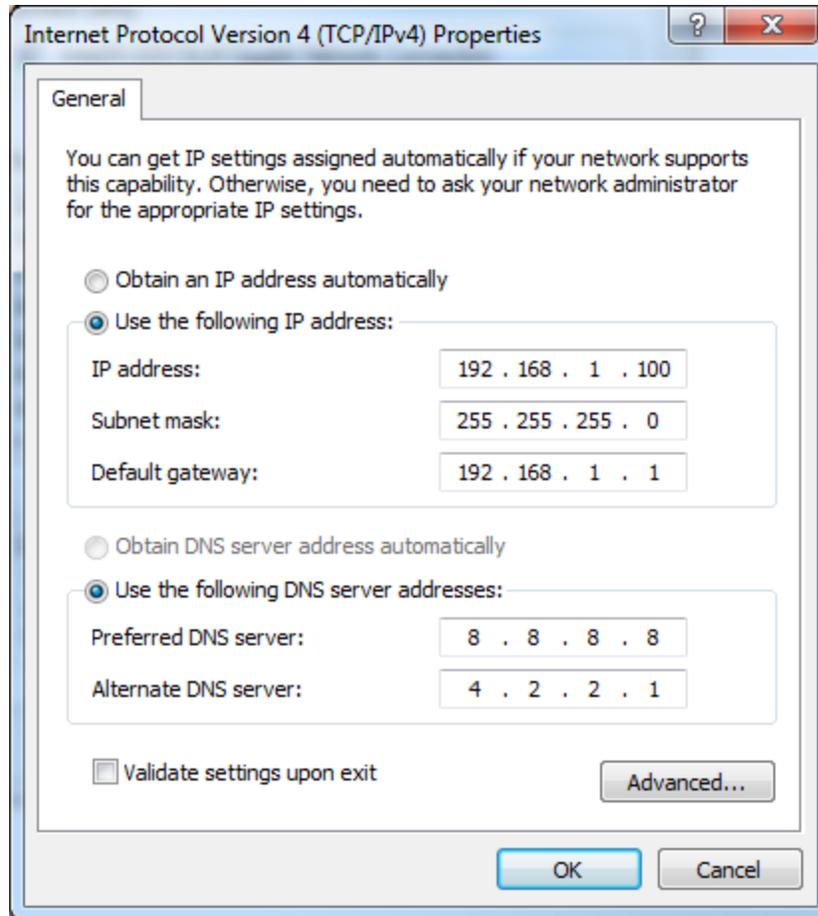
### Experiment 3: Connect Hardware and Run Application

1. On the UltraZed 3EG SOM, set SW2 to boot from JTAG. To do so set all switched to (1,1,1,1)
2. Plug the UltraZed-EG SOM onto the PCIe Carrier Card via JX1/JX2/JX3 connectors and connect the fan to the fan header(J10) on the PCIe Carrier Card
3. Plug the Network FMC onto the PCIe Carrier cards FMC connector (con1)
4. Connect the Uart port on the PCIe Carrier Card (J2) to a free USB port on your PC
5. Connect the USB-UART port on the PCIe Carrier Card (U8) to a free USB port on your PC
6. Connect the 12V power cable, but do not turn on the board yet
7. Connect an Ethernet cable between the Network FMC (J1) and your PC.
8. Your test setup should be similar to the figure below



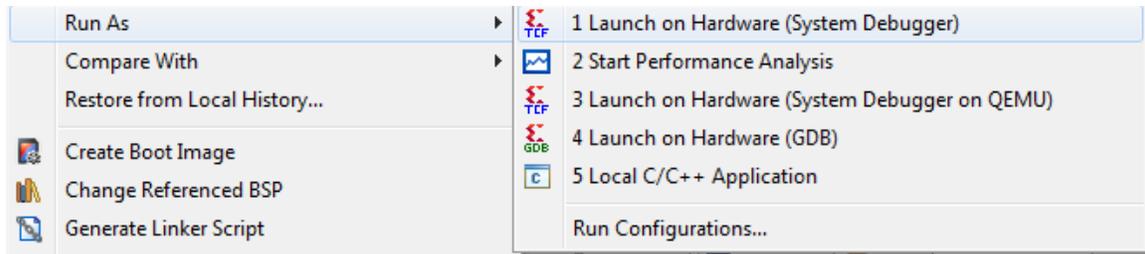
**Figure 19 -- Network FMC Setup**

9. Connect your PC to have a Static IP address of the following.



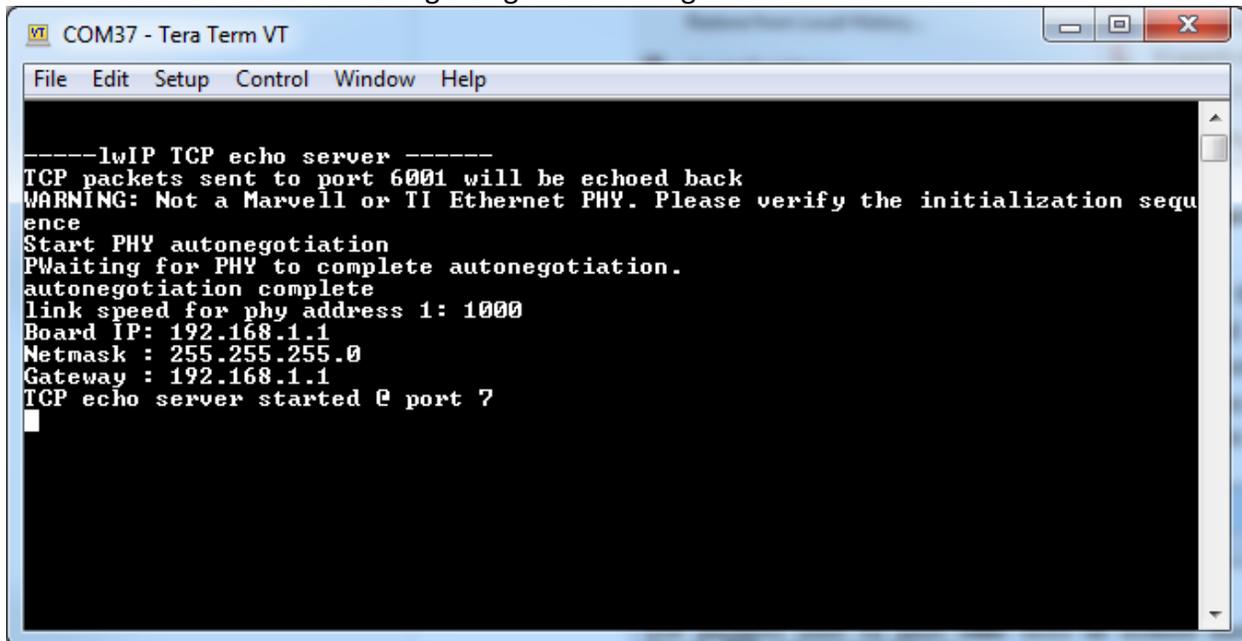
**Figure 20 -- Static IP Address**

10. Disable your wireless internet connection on your PC
11. Open a Terminal such a Tera Term, and set the COM port to the active COM setting for your board. You will notice for your UltraZed there are two active COM settings. Select the higher numbered COM. If in the later steps you do not see any information over the UART, come back to this step and select the lower COM port.
12. Power the Carrier by switching SW7 on the PCIe Carrier Card to the on position.
13. In SDK Program the PL first by clicking the  icon or selecting **Xilinx Tools** → **Program FPGA**. The default options are acceptable. Click **Program**. When complete, the Blue DONE LED on the SOM should light.
14. Right-Click on the Echo\_Server application under the Project Explorter and select **Run As -> Launch on Hardware (System Debugger)**



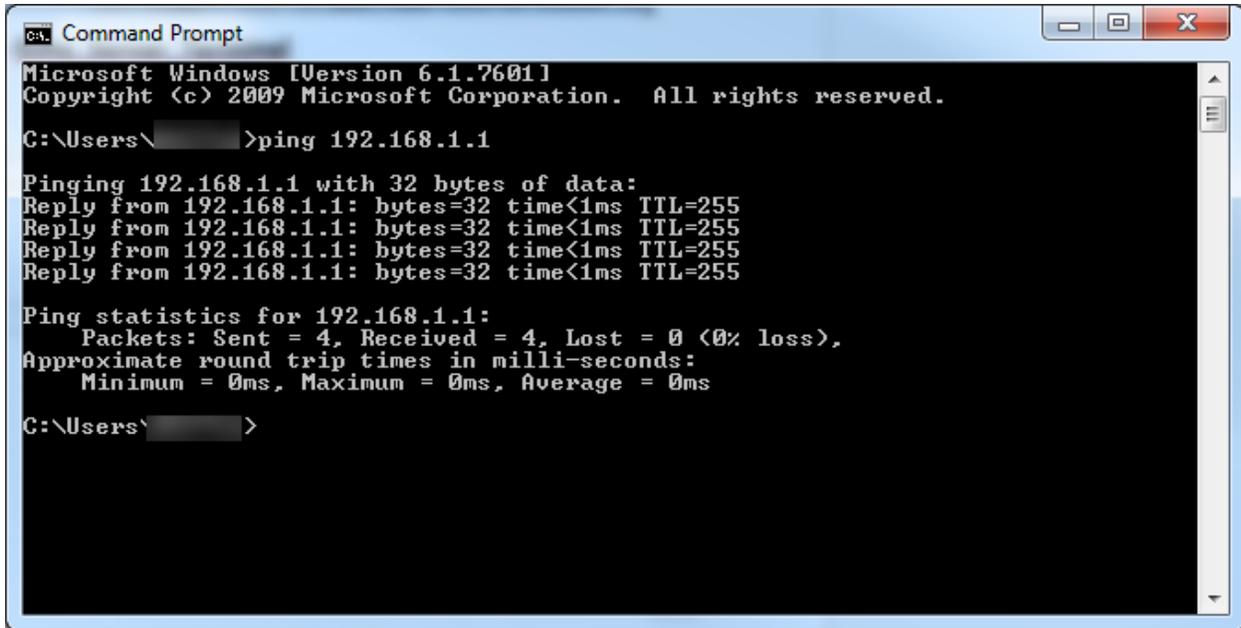
**Figure 21 -- Launch on Hardware**

15. The tools will now initialize the processor, download the Echo\_Server.elf to DDR, and then run Echo\_Server application. This takes a few seconds to complete, depending on the USB traffic in your system. You can follow the progress in the lower right corner of SDK. You should see something along the following in the Tera Term terminal.



**Figure 22 -- Echo\_Server\_Terminal**

16. Note: If you do not see the following, confirm your static IP address is configured properly, your connection is set to 1G, your lwIP setting in the BSP are setup correctly, and your wireless connection is disabled.
17. Lets verify our connection is setup properly. Open a command window on your PC and use the following command to ping your board: **ping 192.168.1.1**



```
cmd. Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255

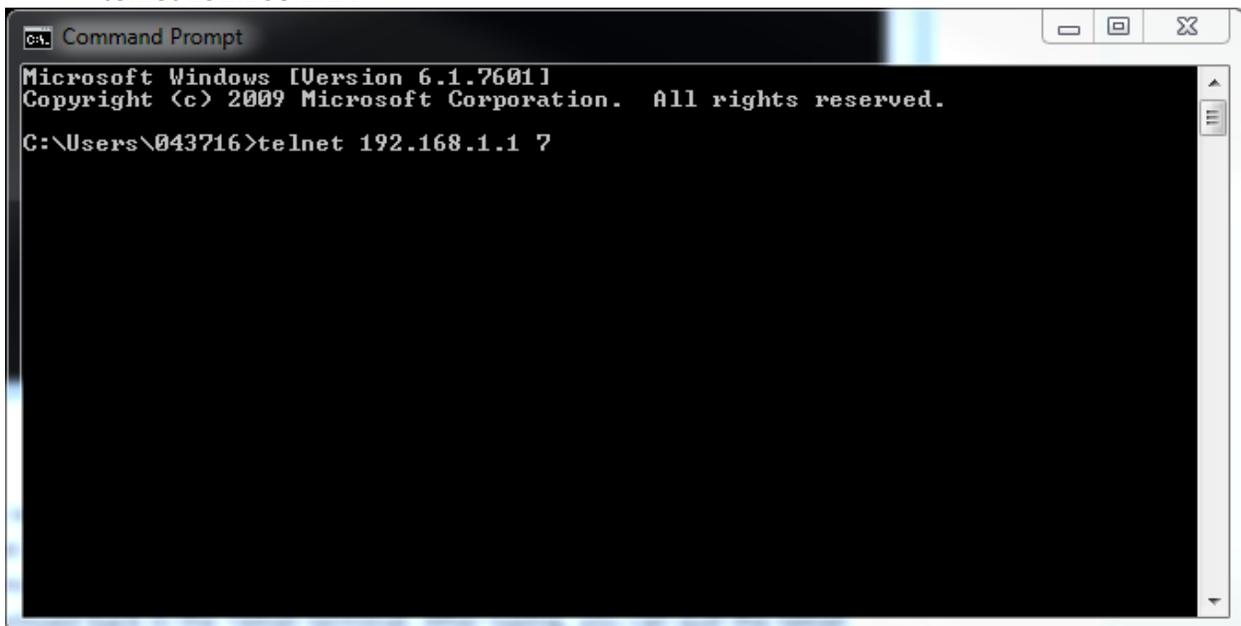
Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\>
```

Figure 23 - Pinging Network FMC PHY

18. Please proceed so long as you receive the same success as the figure above.

19. To connect to the echo server, use the telnet utility program. Type the following telnet command as shown below in figure 12 and hit the return key in the command window.  
**telnet 192.168.1.1 7**



```
cmd. Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\043716>telnet 192.168.1.1 7
```

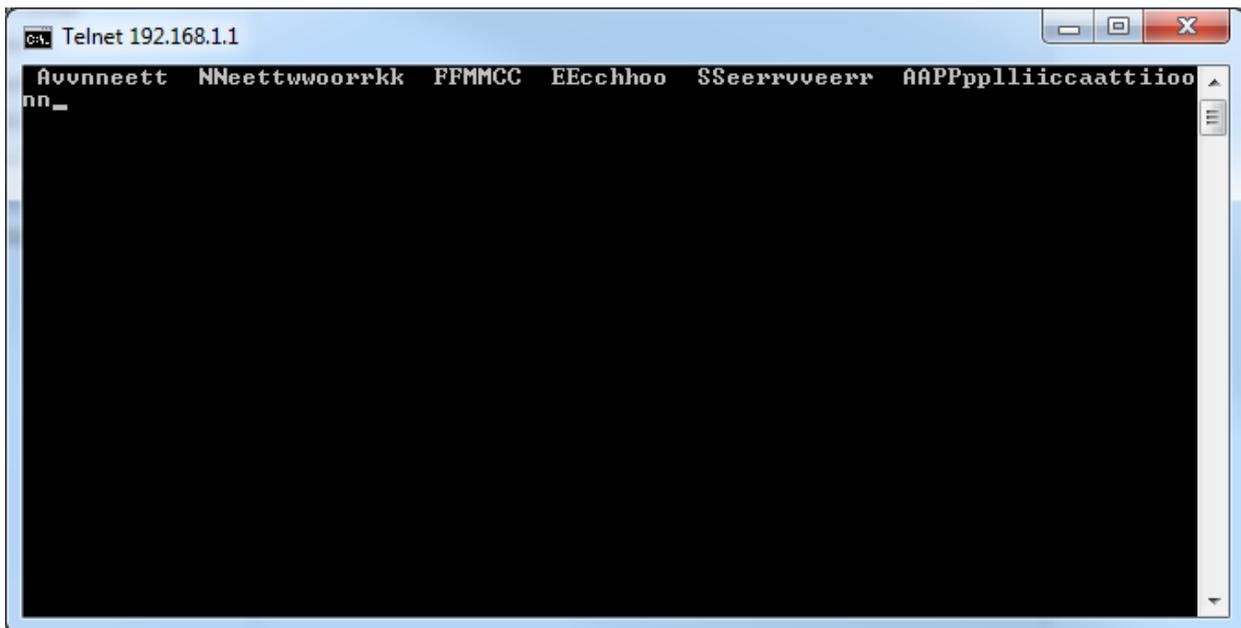
Figure 24 -- Start Echo Server

If your Telnet Utility Program is not enabled please follow the steps 1-5 below.

1. Click **Start > Control Panel**.
2. Click **Programs and Features**.
3. Click **Turn Windows features on or off**.
4. In the **Windows Features** dialog box, check the **Telnet Client** check box.
5. Click **OK**. The system installs the appropriate files. This will take a few seconds to a minute.

You can also install the Telnet Client by using the command line and following these instructions :

1. Open a command prompt window. Click **Start**, type **cmd** in the **Start Search** box, and then press **ENTER**.
2. Type the following command: `pkgmgr /iu:"TelnetClient"`
3. If the **User Account Control** dialog box appears, confirm that the action it displays is what you want, and then click **Continue**.
4. When the command prompt appears again, the installation is complete.



**Figure 25 -- Echo Server**

20. If the echo server works properly, any data sent to the board is echoed in response. Some telnet clients immediately send the character to the server and echo the received data back instead of waiting for the carriage return. Simply type a few characters and see them echoed back in the Telnet terminal. After typing, you can quit the telnet session gracefully by typing **<Ctrl key> ]** then at the prompt type **quit**.
21. You may now power down your board. Now if we wish to target PHY 2 on the Network FMC we would do the following.

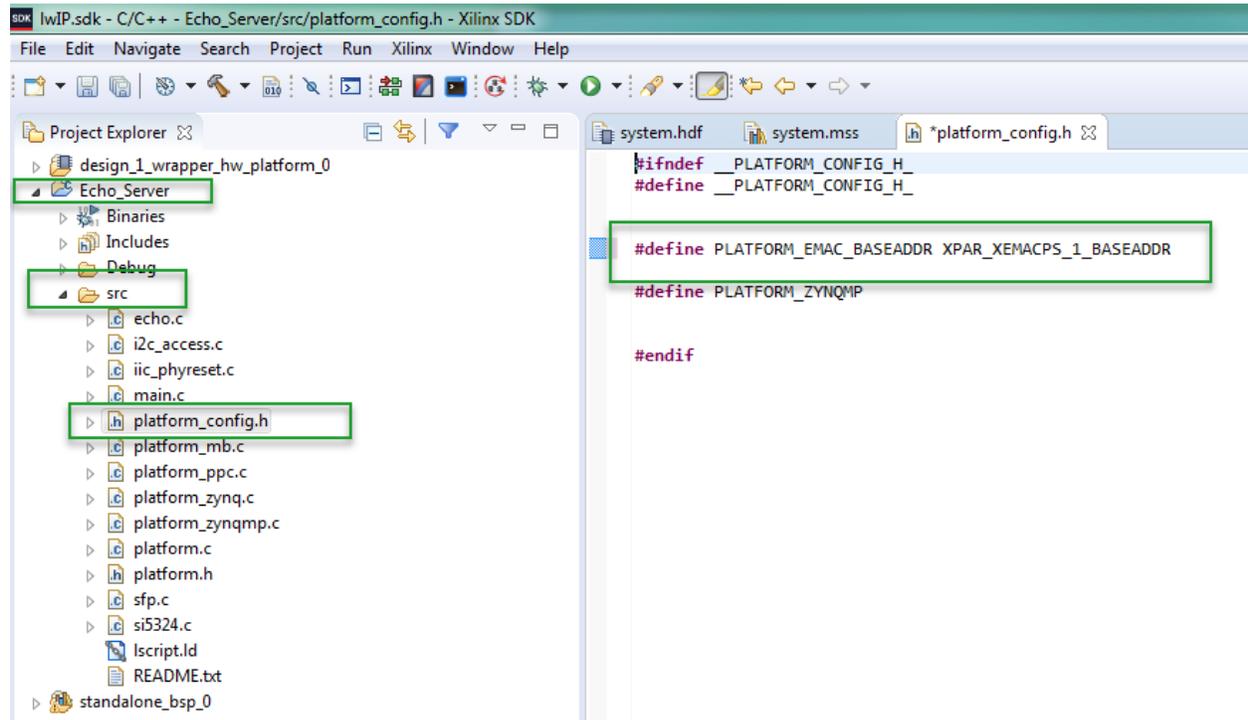
22. In the Project Explorer tab expand **Echo\_Server** -> **src**. Double click on **platform\_config.h**

In the platform\_config.h file change

**#define** PLATFORM\_EMAC\_BASEADDR XPAR\_XEMACPS\_0\_BASEADDR

To

**#define** PLATFORM\_EMAC\_BASEADDR XPAR\_XEMACPS\_1\_BASEADDR



**Figure 26 -- Targeting Network FMC PHY 2**

23. Save the platform\_config.h file and then change the Ethernet cable from port 1 to port 2 on the Network FMC.

24. Doing this we have successfully retargeted the application to PHY 2, please rerun the application above.

## Revision History

Date	Version	Revision
13 Aug 2018	1	Initial Avnet release Vivado 2018.2