

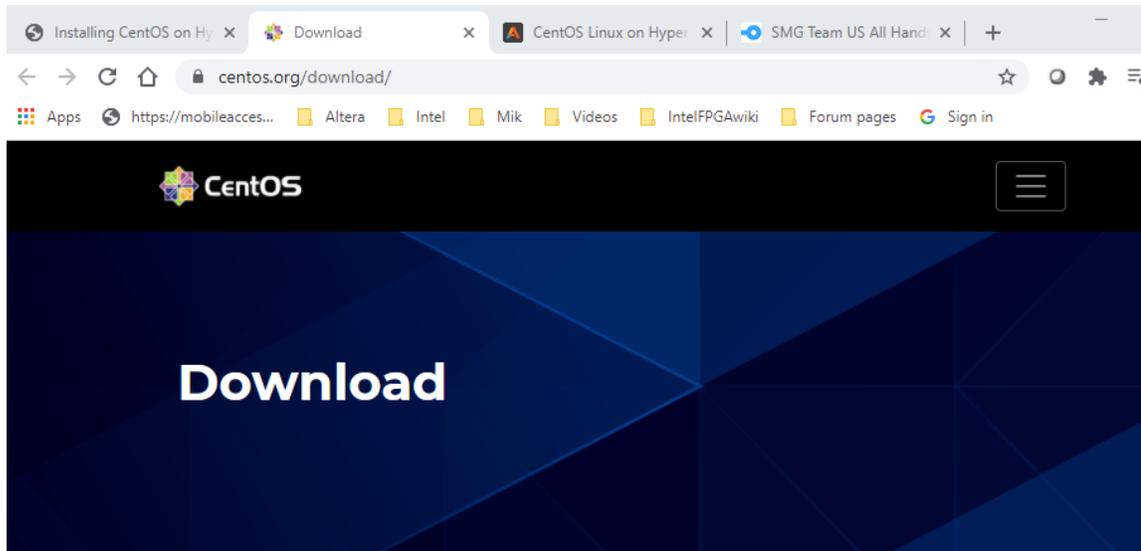
# Setting up a new CentOS Hyper-V VM for Linux NIOSII EDS development on a Windows PC.

## Introduction:

- The purpose of this article is to help Windows OS users work around the recent conflicts when running both Quartus 18.1 or earlier versions of NIOSII EDS and Cygwin when also having Quartus 19.1 and later versions of NIOSII EDS and WSL or WSL2. The article will describe how to install a virtual machine running Linux and using Linux installation of Quartus NIOSII EDS to work around conflicts and permissions issues that may be prohibiting NIOSII command line and Eclipse tools from functioning properly.

## Installation of CentOS:

- Go to the link referenced below and follow the instructions right up to where you reboot the CentOS VM. Do not reboot until the .iso file is removed once CentOS is installed (See more in the below steps).
  - o [https://linuxhint.com/install\\_centos\\_hyperv/](https://linuxhint.com/install_centos_hyperv/)
  - o Note\* - Turning on Hyper-V VM from windows will require a reboot of your PC.
  - o Look through the notes below in parallel to looking over the link reference above up until CentOS is installed and you can login to CentOS.
  - o During areas where network connections are being made between CentOS and Windows OS, you may need to be off the VPN. File transfer between Windows OS and CentOS via WinSCP may not work unless the VPN is disconnected.
  - o When you re-open your CentOS machine, you may be required to turn on the wired connection. If you are having problems reaching the PC IP addresses or seeing the remote JTAG server, make sure the wired connection is turned on.
- In the above article, there is a link to get the CentOS download.
- For CentOS install files go here: <https://centos.org/download/>
  - o Download ISO x86\_64



## CentOS Linux

8 (2004)	7 (2003)	6.10
<b>ISO</b>	<b>Packages</b>	<b>Others</b>
x86_64	RPMs	Cloud   Containers   Vagrant
ARM64 (aarch64)	RPMs	Cloud   Containers   Vagrant
IBM Power (ppc64le)	RPMs	Cloud   Containers   Vagrant

[Release Notes](#)
[Release Email](#)
[Documentation](#)

## CentOS Stream

Latest		
<b>ISO</b>	<b>Packages</b>	<b>Others</b>
x86_64	RPMs	Cloud   Containers   Vagrant
ARM64 (aarch64)	RPMs	Cloud   Containers   Vagrant
IBM Power (ppc64le)	RPMs	Cloud   Containers   Vagrant

[Release Notes](#)
[Release Email](#)
[Documentation](#)

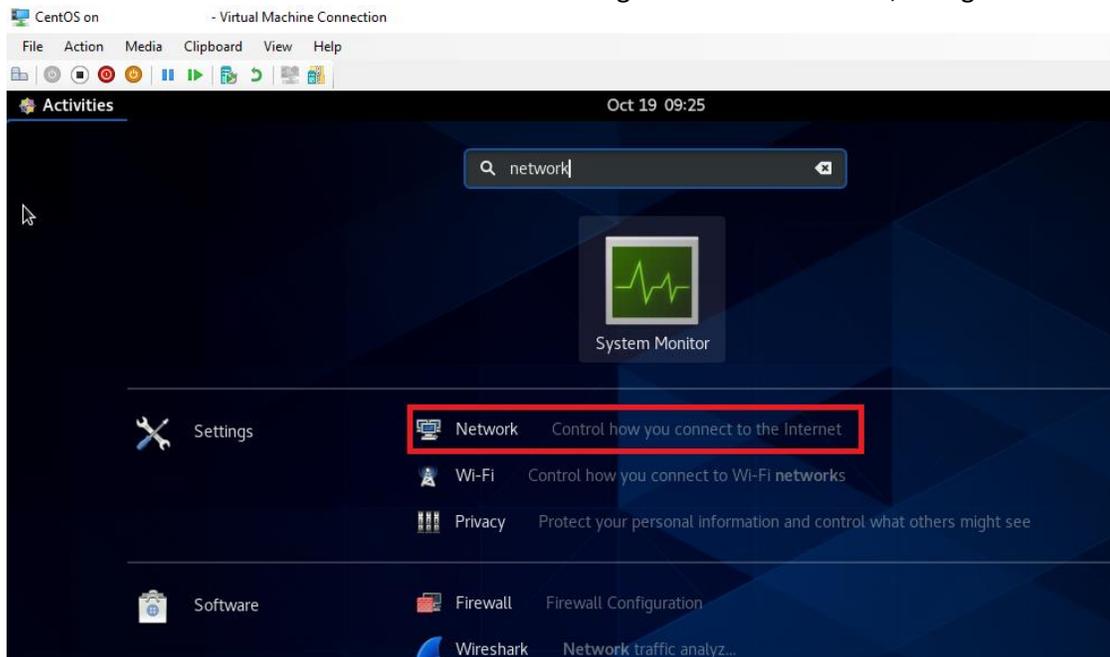
- This will bring you to a list of mirror sites.
  - o Use one of the more local mirrors and then click on the CentOS-8.2.2004-x86\_64-dvd1.iso file to download.
  - o None of the other files are needed.
- The article discusses creating root and another userID. I opted to create a root password and used the same password from my userID. I setup my userID to have admin privileges.
- Continue following the article until you see this:
- *"Upon reboot you will be greeted with the login prompt where you can login as root or any other user you may have created during installation."*
- At this point, DO NOT REBOOT YET.
- First, remove the .iso file from where you downloaded and used it for creating CentOSVM, otherwise, the VM will think the install has to happen again even though it has already been installed.
- Now, reboot.

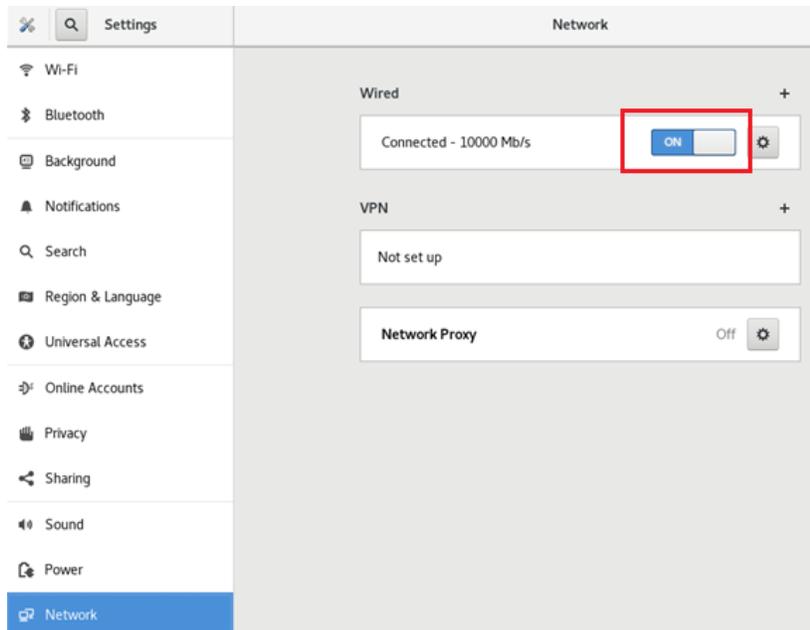
## Setup CentOS

- Login to CentOS with your userid and password.
- Agree on licensing and proceed.
- Open an xterm window via “Activities -> Terminal”



- Make sure network is on in CentOS network settings so that we can transfer files between Windows OS and CentOS. Go to “Activities->Settings -> Network-> Wired, change to ON.

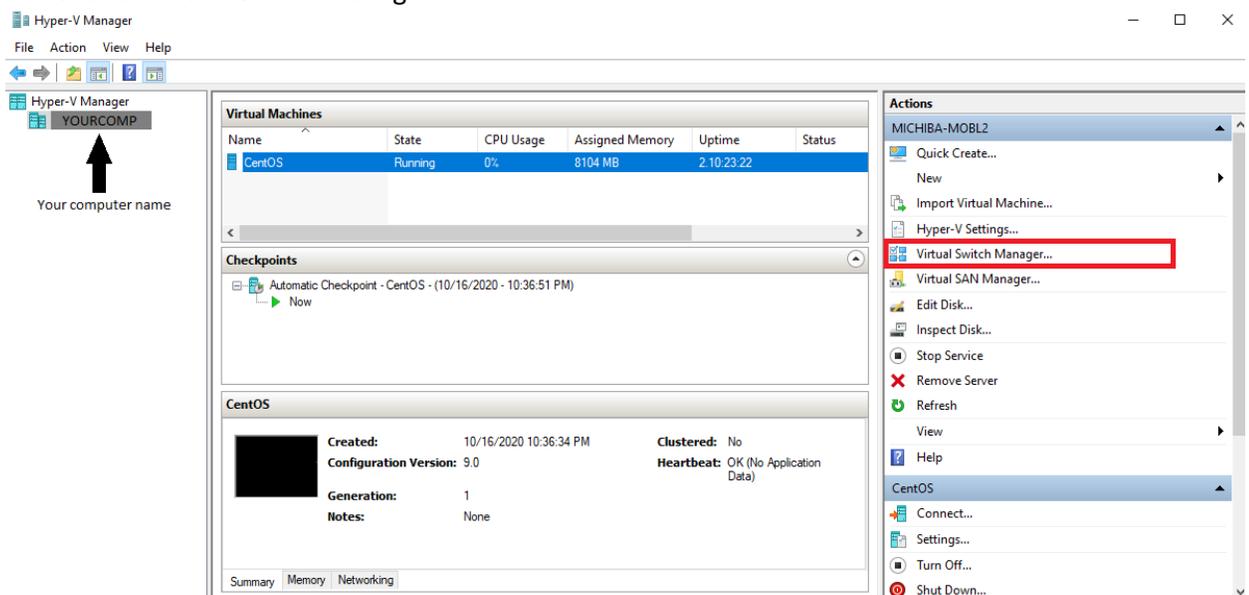




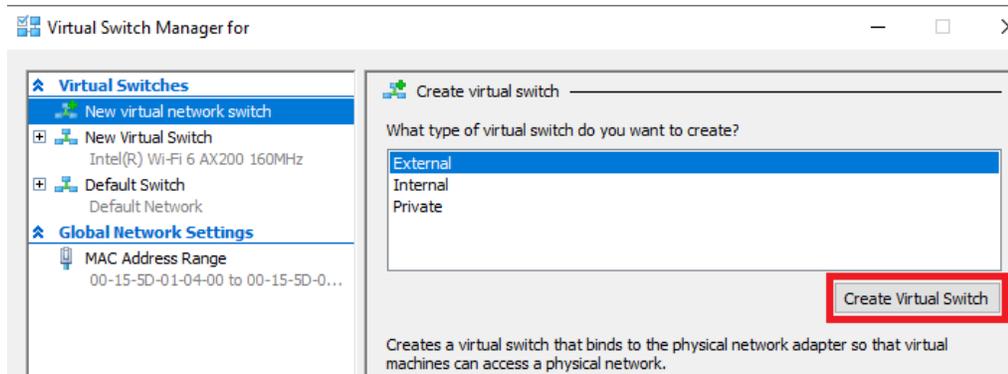
- Note\* After disconnecting from the VM, you may have to go back in and turn on the connection again if you cannot ping your computer IP address.

### Setting up Hyper-V in Windows PCOS

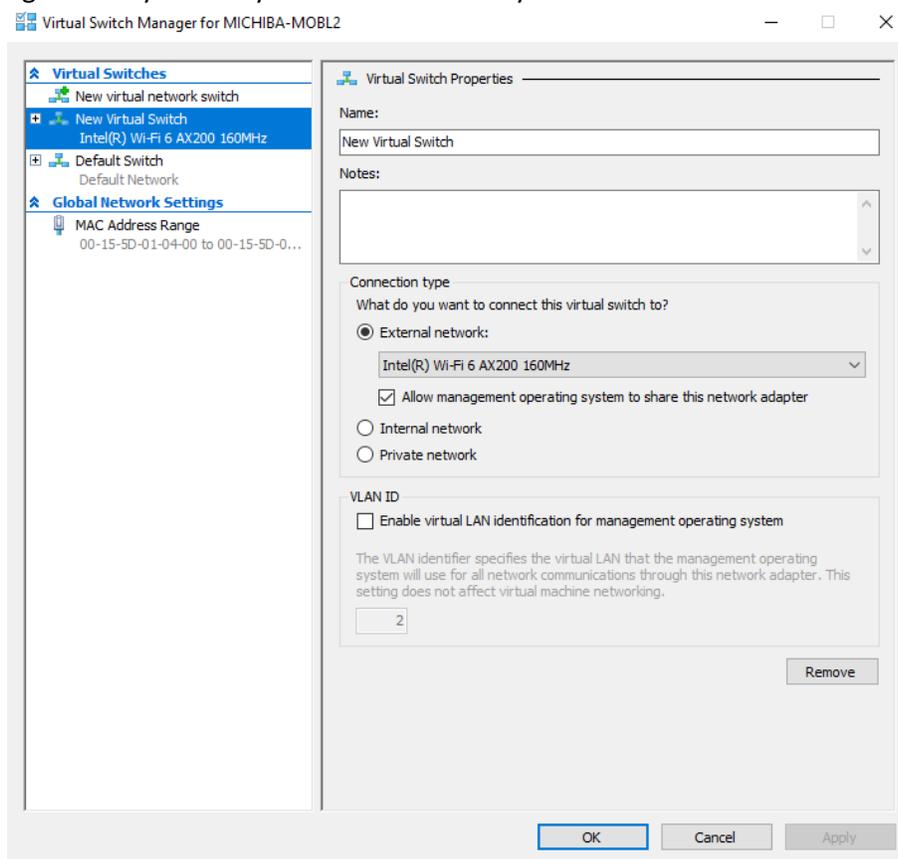
- Open Hyper-V Manager and setup a new virtual switch:
  - o Click on Virtual Switch Manager....



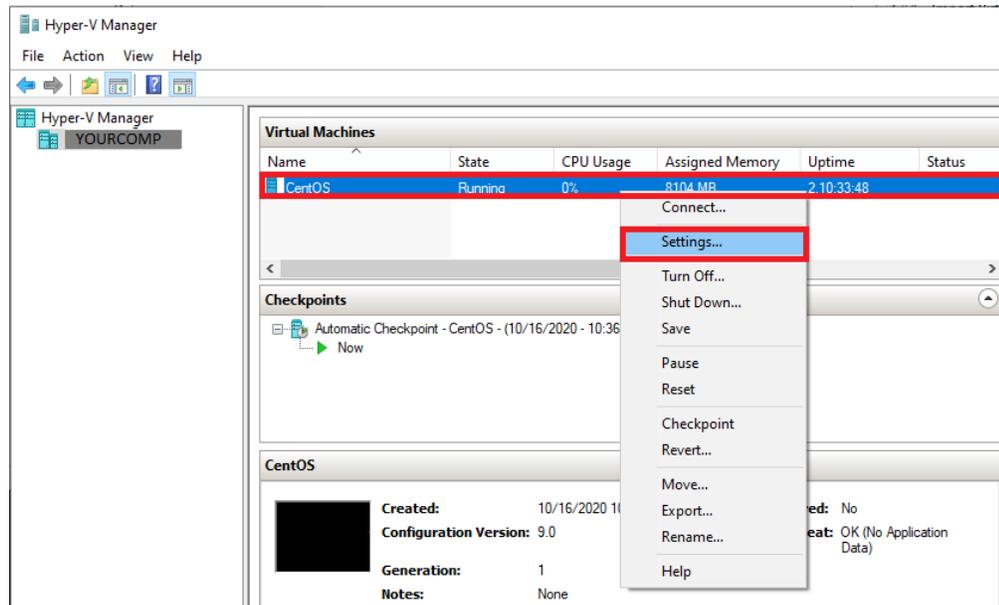
- Create virtual switch



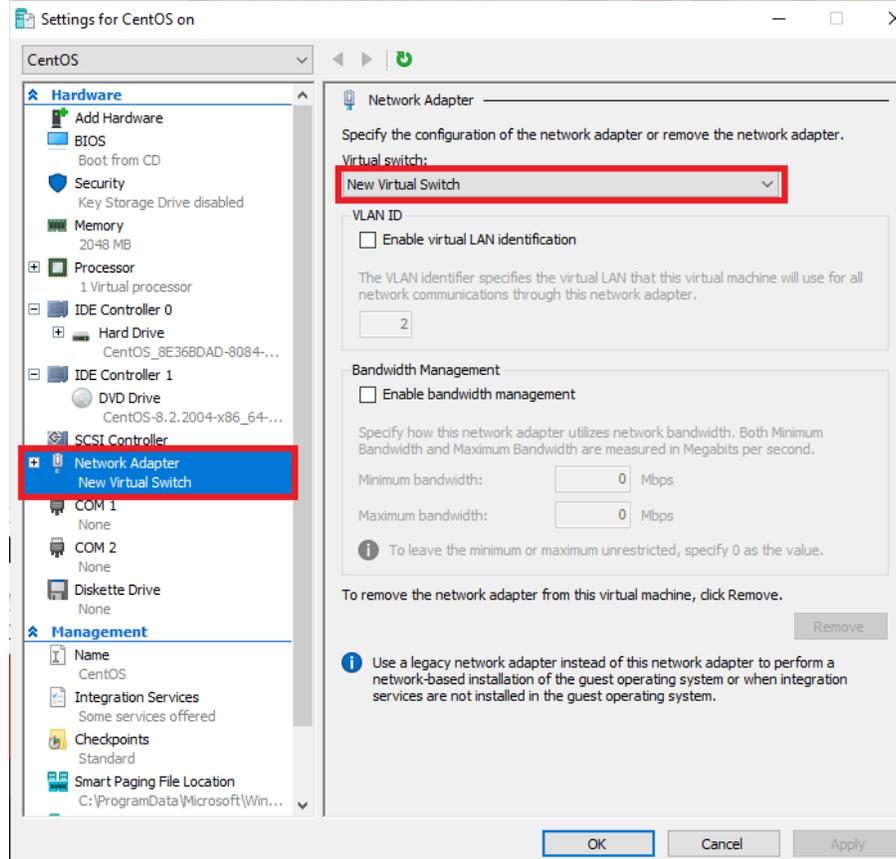
- Setup the new virtual switch to be External network. You can use your Wi-Fi or Ethernet depending on how you have your PC connected to your network.



- Click OK
- Right click on your newly created VM and open Settings



- Change the Network Adapter to use the newly created “New Virtual Switch”



- Click OK and go back to your CentOS session
- Get the network information in CentOS by using the terminal window.
- Do an ifconfig.

```

[localhost spi bridge test]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.23 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::af60:8e71:204a:b395 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:01:04:0a txqueuelen 1000 (Ethernet)
    RX packets 3221330 bytes 3591783476 (3.3 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 146238 bytes 17233079 (16.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8446 bytes 643106 (628.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8446 bytes 643106 (628.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

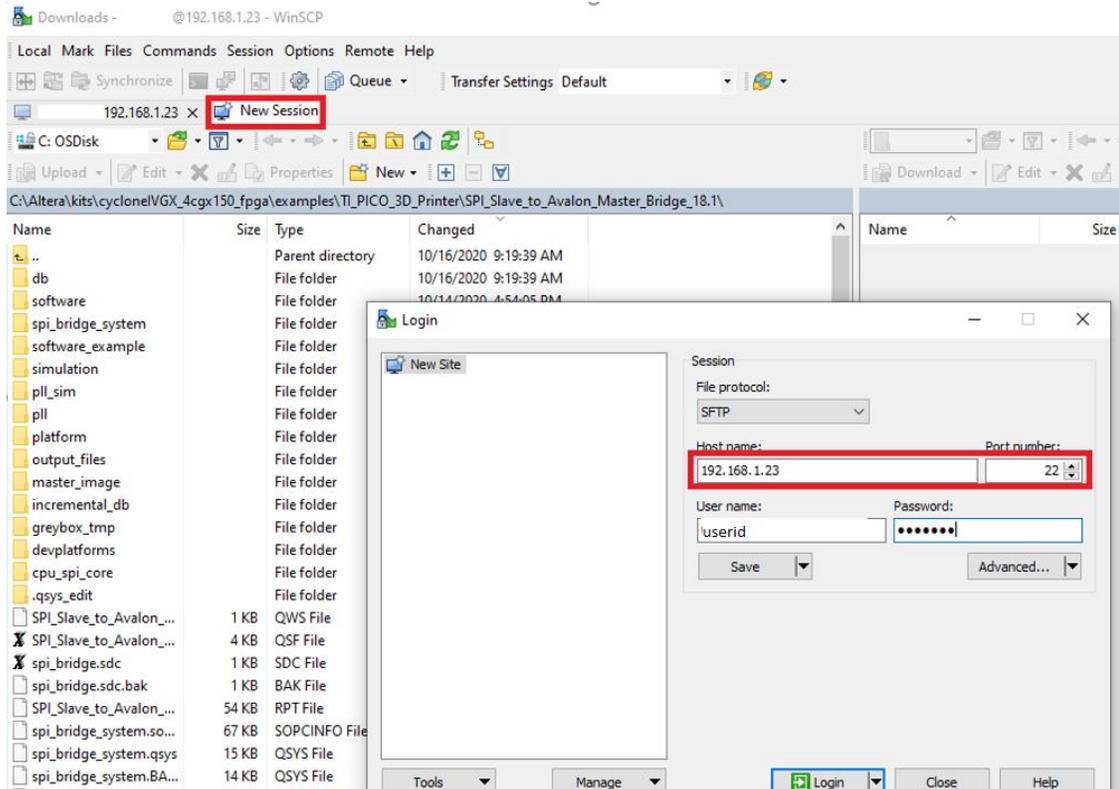
virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:53:8e:9e txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

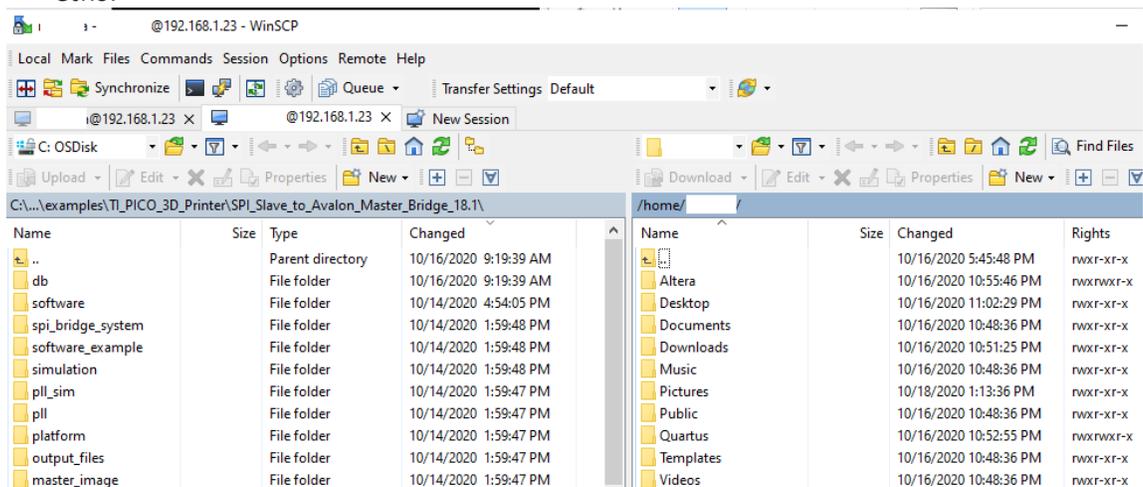
At this point, you should see an IP address for your Windows OS based PC listed in red for port eth0. Even though your PC connection may be WiFi, CentOS will see it as physical hardware. If no IP address shows up in red box area, try and go back to re-create a new virtual switch in Hyper-V manager. Remember the IP address in the red box.

### Install and configure WinSCP for file transfer between Windows OS and CentOS

- Download and install WinSCP:
  - o <https://winscp.net/eng/index.php>
- Launch WinSCP and create a new session that will allow you to transfer files from your Windows OS to your CentOS. Use the IP address found in the previous section CentOS ifconfig for your Host Name. Default port of 22 should be OK.



- Use the username and password you created when installing CentOS.
- Once logged in, the right side of WinSCP should look like the below.
  - o Note\* you may not be able to connect to the CentOS using the IP address if connected to a company network via VPN. Make sure VPN is disconnected.
  - o Note\* Each time you disconnect from VPN or from CentOS session or turn on the wired connection within CentOS, the IP address in the above may change. If you cannot connect via WinSCP, go back into CentOS and run ifconfig to check the IP address for eth0.



- At this point, you can now transfer from your Windows OS C: drive to your CentOS /home/userid directory structure.

## Download and install Quartus for Linux

- Download Quartus for Linux, in this example we will use Quartus Prime Standard 18.1
  - o <https://fpgasoftwre.intel.com/18.1/?edition=standard&platform=linux>

### Download Center for FPGAs

The screenshot shows the Intel Quartus Prime Standard Edition download page. On the left is a navigation menu with categories like Design Software, Embedded Software, Archives, Licensing, Programming Software, Drivers, Board System Design, Board Layout and Test, and Legacy Software. The main content area displays 'Quartus Prime Standard Edition' with release information (September 2018, v20.1) and a 'Select edition' dropdown set to 'Standard' and 'Select release' dropdown set to '18.1'. The 'Operating System' section has radio buttons for Windows and Linux, with Linux selected and highlighted by a red box. Below this is a yellow information box with three green checkmarks providing update and support information. At the bottom, there are tabs for 'Combined Files', 'Individual Files', 'Additional Software', and 'Updates'. A section titled 'Download and install instructions' includes links for 'Read Intel FPGA Software v18.1 Installation FAQ' and 'Quick Start Guide'. An 'Updates Available' notification is also present. At the very bottom, a download button for 'Quartus Prime Standard Edition' is shown, with a red box highlighting the file name 'Quartus Prime (includes Nios II EDS)' and its size '2.7 GB'.

- Now transfer the download to your CentOS VM via WinSCP

The screenshot shows the WinSCP interface. The top bar indicates the connection is to '192.168.1.23 - WinSCP'. The interface is split into two panes. The left pane shows the local file system 'C:\Users\...\Downloads\' with a list of folders and files. The right pane shows the remote file system '/home/.../Downloads/' with a list of files. A red arrow points from the file 'QuartusSetup-18.1.0.625-linux.run' in the local pane to the same file in the remote pane. The file details in the remote pane are: Name: QuartusSetup-18.1.0.625-linux.run, Size: 2,851,44..., Changed: 10/16/2020 8:41:29 AM, Rights: rwx-----, rwxrwxr-x.

- Go to your CentOS VM.
- Open an xterm window and navigate to where the Quartus 18.1 run file was placed.

- Change the .run file to be executable via chmod +x

```
[ userid @localhost ~]$ cd Downloads/
[ userid @localhost Downloads]$ ls
QuartusSetup-18.1.0.625-linux.run
[ userid @localhost Downloads]$ ll
total 4179840
-rw-rw-r--. 1 userid userid 2919876208 Oct 16 09:41 QuartusSetup-18.1.0.625-linux.run
[ userid @localhost Downloads]$ chmod +x QuartusSetup-18.1.0.625-linux.run
userid a@localhost Downloads]$ ll
total 2851444
-rwxrwxr-x. 1 userid userid 2919876208 Oct 16 09:41 QuartusSetup-18.1.0.625-linux.run
```

- Then, execute the .run file to install Quartus Standard 18.1.

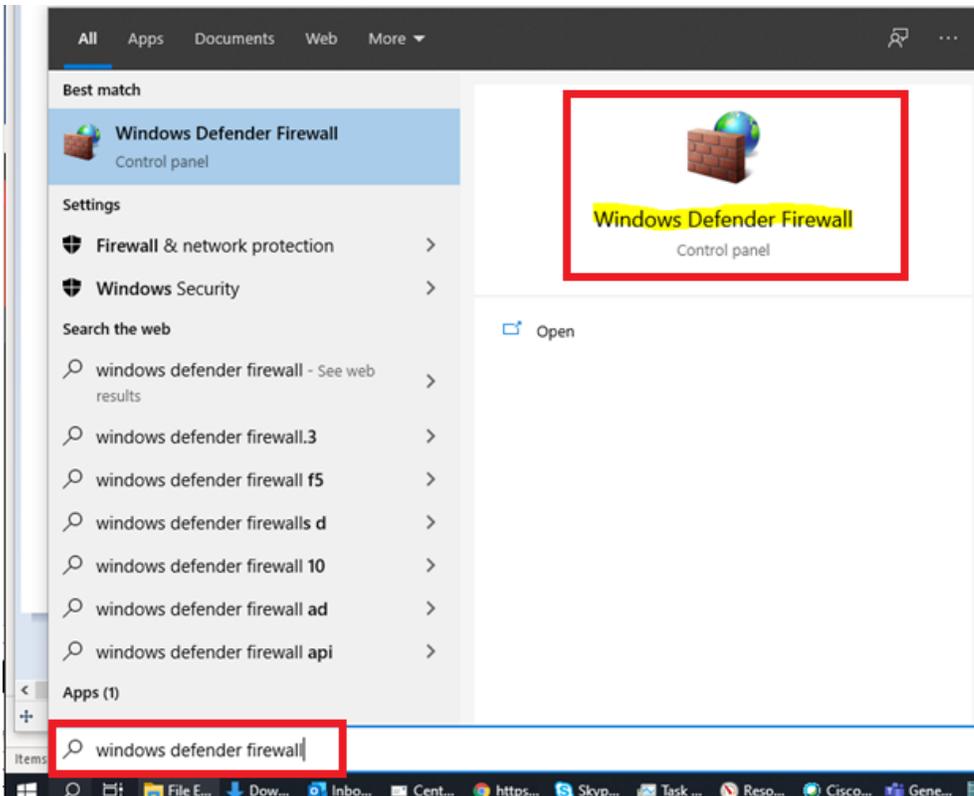
```
[ userid @localhost Downloads]$ ./QuartusSetup-18.1.0.625-linux.run
```

### Installing CentOS development tools

- After installing Quartus on your CentOS Linux VM, we need to Install CentOS tools for developers. The developer tools allow NIOSII EDS to operate as needed to create bsp and applications.
- From an xterm window:
  - o sudo yum groupinstall "Development Tools"
  - o sudo yum install libpng12
  - o sudo yum install libns1
- At this point, if you have a NIOSII project example with app and bsp directories, you can run create-this-app and generate an elf file.
  - o Note\* when you transfer executable files from Windows OS to CentOS, you may need to do a chmod +x on the executable files for them to run in CentOS
- Once an elf file is created, a user needs to be able to run nios2-download and nios2-terminal from the CentOS VM.
- The best way to do this it to connect your development kit or custom FPGA board to your PC and then create a JTAG server so that the CentOS can connect to the JTAG server setup on the Windows based PC.

### Setting up the JTAG server on the Windows PC.

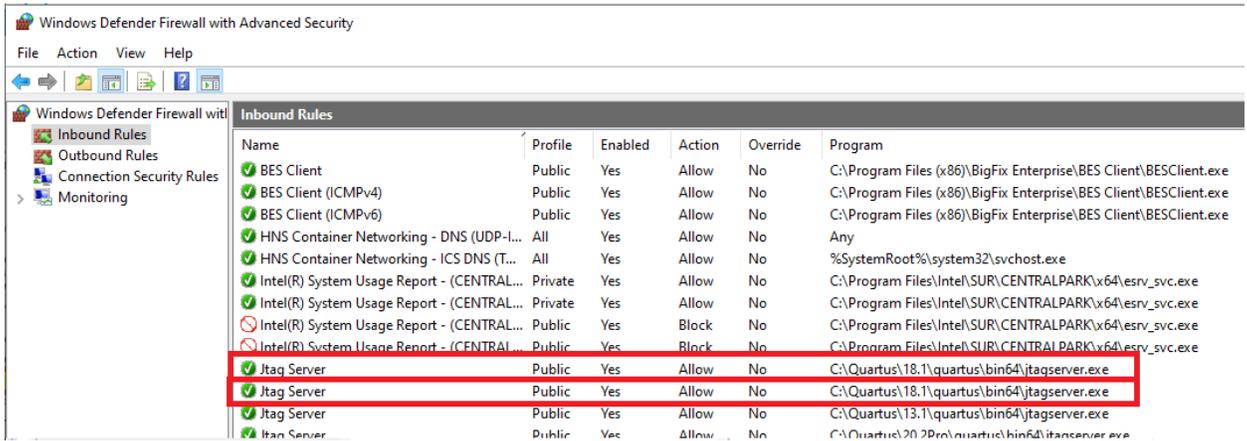
- The following assumes that you have already installed Quartus tools on your Windows based PC.
- In this example case, Quartus Prime Standard 18.1 Windows was installed on the PC.
- Open Windows Defender Firewall.



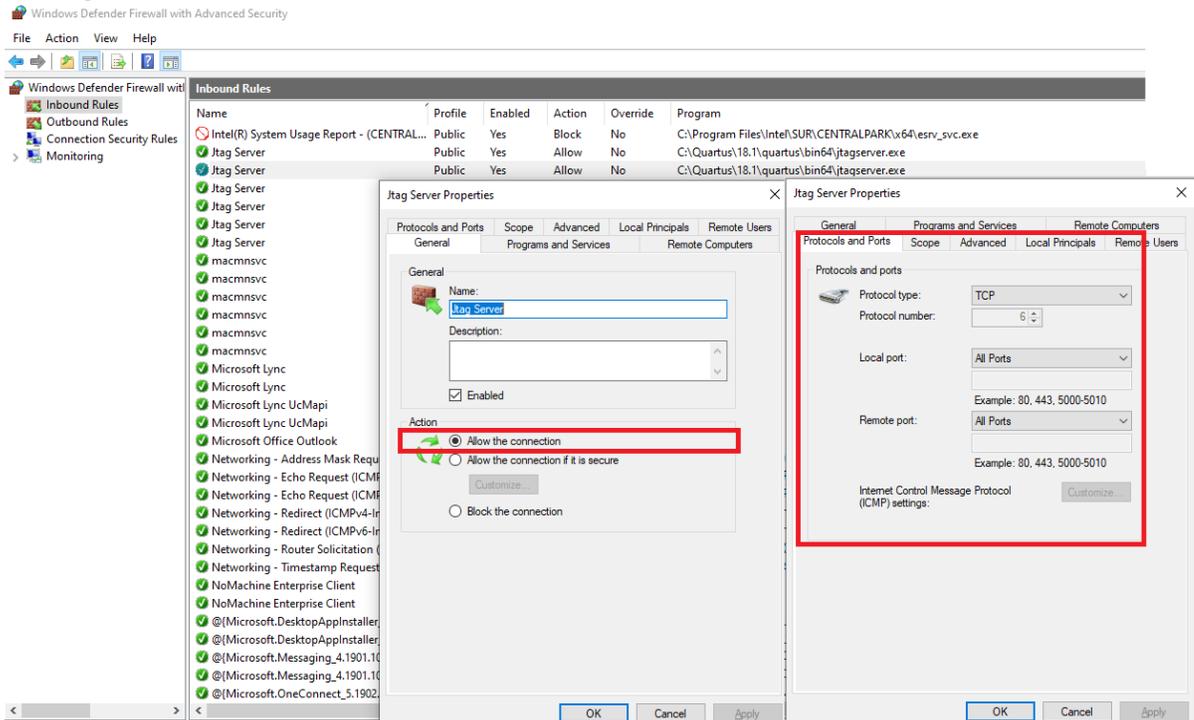
- Go to Advanced settings:



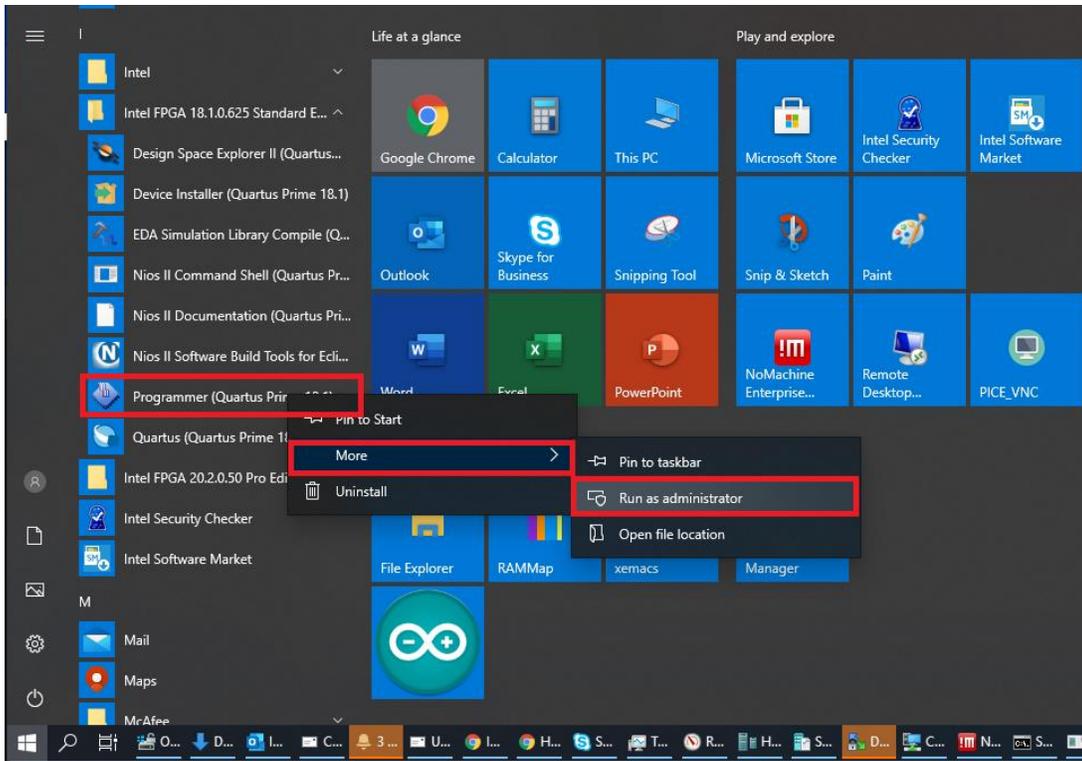
- Click Inbound rules and find the rules for JTAG Server associated with the Quartus 18.1 installation on the Windows PC.



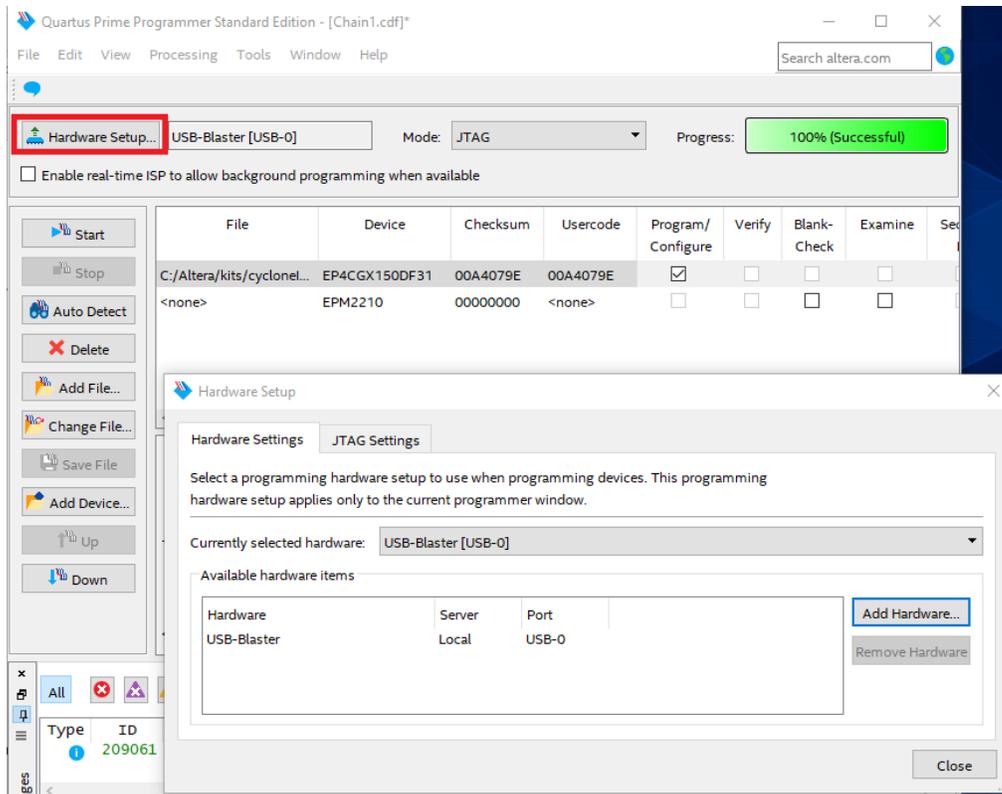
- Right click on each one and figure out which one is setup for TCP. One will be setup for UDP and one for TCP.
- For the one that uses TCP for Protocols and Ports, make sure it is setup to allow “All Ports”
- In the general tab, make sure the action is set to “Allow the connection”



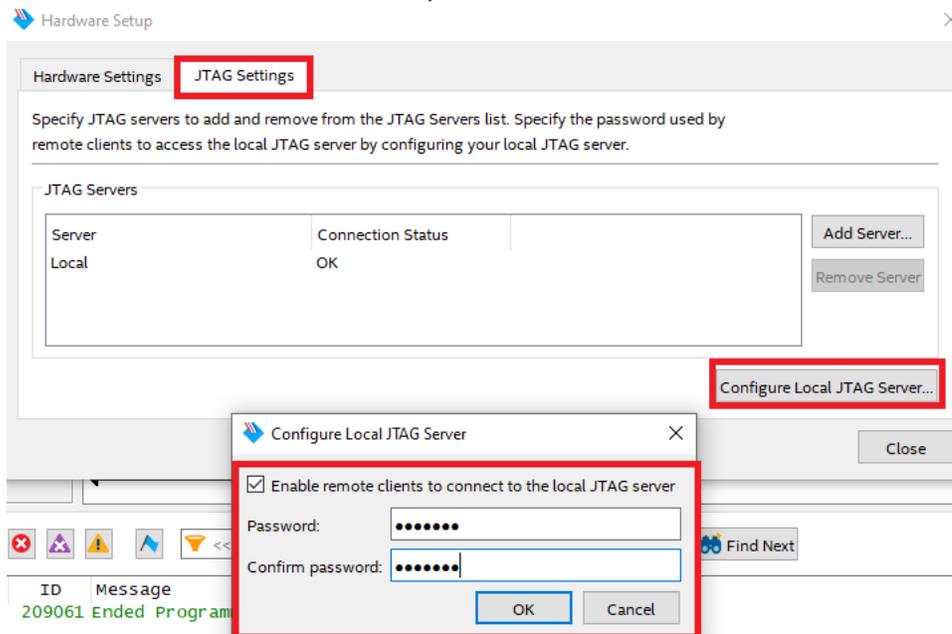
- Start the Quartus programmer either via Quartus or open the Quartus programmer directly.
  - o Make sure you run as ADMINISTRATOR



- If you do not run as administrator, the JTAG server will make registry changes required for remote machines like the CentOS VM to connect.
- Make sure the board with the FPGA is physically connected via USB port and launch "Hardware Setup"



- You can see that a Cyclone IV development kit is connected to port USB-0.
- Click on the JTAG Settings tab, click Configure JTAG Server, and enable remote clients to connect to the JTAG server via TCP. Enter a password for the remote client to use.



- Click OK.
- Now, let's setup the remote JTAG server in CentOS.

## Setting up the remote JTAG server in CentOS

- Login to the CentOS VM, launch Quartus, and then open programmer or open the programmer directly within CentOS.
  - o In an xterm window, launch a NIOSII command shell
    - Go to the Quartus install directory, nios2eds directory and run `nios2_command_shell.sh`
  - o This should setup the Quartus environment variables.
  - o Then, launch Quartus and then the programmer from the GUI or launch the programmer directly via `quartus_pgmw`
- In the Windows OS, launch a command shell and do an `ipconfig /all`
  - o Some non-pertinent information from the below was removed to focus on the Hyper-V Virtual Switch information

```
Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

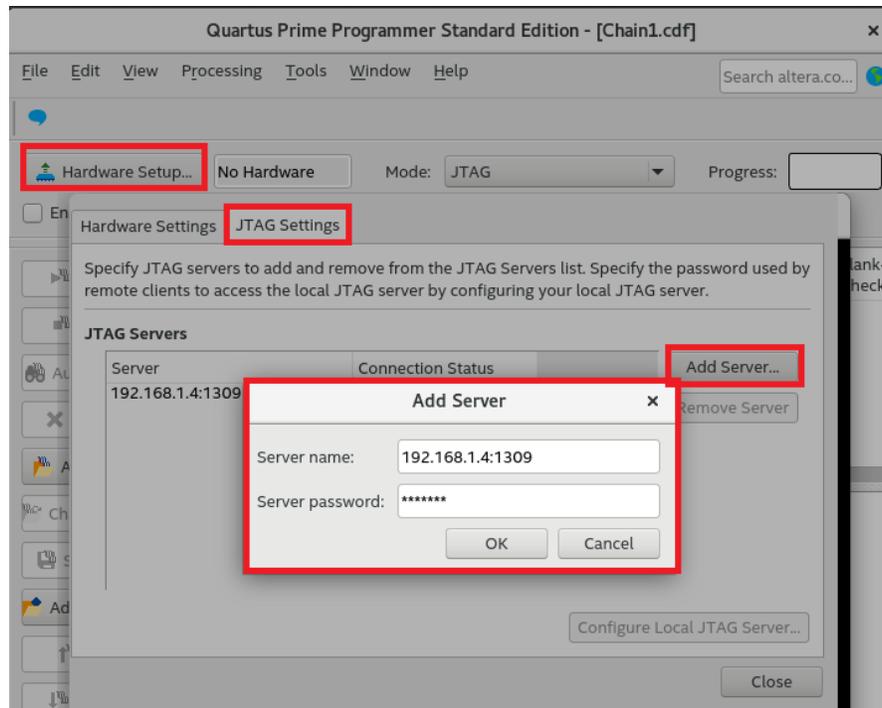
C:\windows\system32>ipconfig /all

Windows IP Configuration

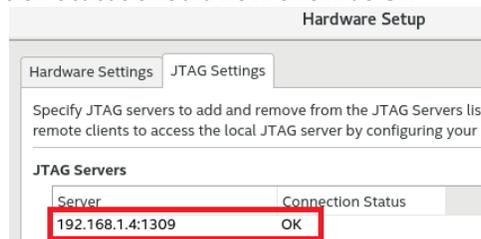
Ethernet adapter vEthernet (New Virtual Switch):

    Connection-specific DNS Suffix . . . . . : 
    Description . . . . . : Hyper-V Virtual Ethernet Adapter #2
    Physical Address. . . . . : 14-16-D8-63-84-C2
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::497f:61ba:d639:c061%16(Preferred)
    IPv4 Address. . . . . : 192.168.1.4(Preferred)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained. . . . . : Friday, October 16, 2020 10:33:27 PM
    Lease Expires . . . . . : Monday, October 19, 2020 10:33:27 AM
    Default Gateway . . . . . : 192.168.1.1
    DHCP Server . . . . . : 192.168.1.1
    DHCPv6 IAID . . . . . : 404027096
    DHCPv6 Client DUID. . . . . : 00-01-00-01-26-BB-E4-30-BC-E9-2F-7B-DE-F7
    DNS Servers . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
                          192.168.1.1
```

- Note that the Hyper-V Virtual Ethernet Adapter has an IP address of 192.168.1.4.
  - o Note that the IPv4 address can change after reboots or closing and re-opening a CentOS VM. ie. 192.168.1.24
- Go back into CentOS and in the Quartus programmer, add a server



- Use the IP address noted in the previous step and use port 1309 which is the port setup for the JTAG server on the Windows PC.
  - o Note that you may need to disconnect from your VPN connection for the server to be setup, otherwise you may see "Connecting ..." and it never show as connected "OK".
- Connection status should now show as OK



- In the NIOSII command shell, do a jtagconfig and this will also confirm that the connection the Windows OS JTAG server is working.

```
[userid @localhost spi_bridge_test]$ jtagconfig
1) USB-Blaster on 192.168.1.4:1309 [USB-0]
   028040DD   EP4CGX150
   020A40DD   5M(1270ZF324|2210Z)/EPM2210
```

- If you unable to connect to the JTAG server setup on the Windows OS, check your firewall, make sure you launched Quartus and/or the Quartus programmer on the Windows OS as admin, make sure the IP address is correct and using port 1309, make sure the hardware is connected and can be seen as expected on the Windows OS Quartus programmer, make sure you are using the correct JTAG server password in CentOS that was created in Windows OS.

## Test your NIOSII application

- Now, a user can use CentOS to perform the nios2-download and nios2-terminal directly to the Windows OS JTAG server that is connected to the FPGA board!
  - o Recall in the "Installing CentOS development tools" section that we ran create-this-app to generate the bsp and create an elf file for the application.

```
[userid@localhost spi_bridge_test]$ nios2-download -g spi_bridge_test.elf && nios2-terminal
Using cable "USB-Blaster on 192.168.1.4:1309 [USB-0]", device 1, instance 0x00
Pausing target processor: OK
Initializing CPU cache (if present)
OK
Downloaded 73KB in 0.9s (81.1KB/s)
Verified OK
Starting processor at address 0x00040244
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "USB-Blaster on 192.168.1.4:1309 [USB-0]", device 1, instance 0
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)

Creating random test data ...

Writing data to onchip memory ...
Write transaction successful

Reading data from onchip memory ...

Comparing data ...
Compare data completes error free
```