

CSEE 4840

Embedded System Design

Tutorial: Installing Quartus and Related Tools

Stephen A. Edwards
Columbia University

Spring 2020

This shows how to install Quartus 18.1 under CentOS 6.10 running on a VirtualBox virtual machine. This will enable you to use your own machine to develop hardware and software for the DE1-SoC and configure the FPGA from your virtual machine.

1 Download and Install VirtualBox

Download a binary for your platform from <https://www.virtualbox.org/wiki/Downloads>. Download both the platform-specific “platform package” and the platform-independent “VirtualBox Extension Pack” with the matching version number.

Install the platform package and run it.

Install the VirtualBox Extension Pack: either double-click the file’s icon or start VirtualBox, go to File→Preferences→Extensions, and add the extension. This enables USB 2.0 devices, among other things.

If your host machine is Linux, VirtualBox needs permissions to access USB devices. Add yourself to the vboxusers group by running

```
sudo usermod -a -G vboxusers $(whoami)
```

You will need to log out and back in for this change to take effect. Without this step, VirtualBox will not be able to “see” any of the usb devices plugged into your host (e.g., the DE1-SoC).

2 Download and install CentOS 6.10 on your VM

Find a CentOS download mirror and download the first CentOS 6.10 DVD image:
CentOS-6.10-x86_64-bin-DVD1.iso

Start VirtualBox and click “New” to create a new virtual machine. Call it “4840-2020,” set the type to “Linux,” and the version to “Red Hat (64 bit)”.

Select at least “2048 MB” (2 GB) for memory size. Bigger is better, but do not ask for more than, about half of your host machine’s memory. If necessary, change this later with Settings→System→Motherboard→Base Memory.

On the next screen, select “Create a virtual hard drive now.” Then select “VDI (VirtualBox Disk Image).” Then select “Dynamically allocated.” Finally, set the size to “50 GB” and click “Create.”

The virtual machine should now be configured.

Click “Start” to boot the virtual machine. Select the CentOS 6.10 installation DVD image that you downloaded earlier as the startup disk.

At this point, clicking in the virtual machine window passes control of the mouse and keyboard to the virtual machine; the right control key returns control to your host operating system. We will simplify this later.

Press Return in the VM to install CentOS 5.9 in graphical mode:

Select “Skip” when it asks about testing the installation media.

Select your preferred language and keyboard layout.

Select “Basic Storage Device.” When it asks about “ATA VBOX HARDDISK,” allow it to discard any data (it’s empty anyway).

Set the hostname to 4840-2020 (this does not really matter). Click on “Configure Network,” edit “system eth0” and make sure “Connect automatically” is checked before clicking “Apply.” Close the Network Connections dialog.

Use America/New York as the time zone.

Assign the root password, such as CSee4840!

Select “Use All Space” then allow it to write (partitioning) changes to disk.

Select “Minimal Desktop” installation.

Start the installation process; it will take a while.

Click on “Reboot” after it congratulates you. Your virtual machine should now reboot into CentOS 6.10 and present you with a welcome screen.

Click “Forward.” Agree to the EULA.

Add a user account. We used `quartus` and `CSee4840!` for our image. Click “Forward.”

Tell it to synchronize date and time over the network and click “Forward.”

Don’t enable `kdump`; click “Finish.” It’s not necessary to reboot the system again just now.

Log in to the user account you created earlier and open a terminal window (Applications→System Tools→Terminal).

Turn into root by typing `su` and typing the root password. The prompt should change to a `#`.

Add your user to the *sudoers* file:

```
echo "quartus ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

and then turn back into yourself by typing `exit`. The prompt should change back to a `$`.

Type “`sudo yum update -y`” to install system updates (`-y` tells it to skip prompts). This will take a while.

Type “`sudo yum install -y gcc kernel-devel`” to install the packages you will need to install the Guest Additions.

Reboot your VM. Use System→Shut down...or `sudo poweroff`.

3 Install VirtualBox Guest Additions on the VM

From <http://download.virtualbox.org/virtualbox/6.0.2/> (replace 6.0.2 with the VirtualBox version you installed; see Help→About Virtualbox... to check the version), download [VBoxGuestAdditions_6.0.2.iso](#)

Reboot your VM and Instruct it to mount the Guest Additions ISO file by selecting Devices→Optical Drives→Choose a disk image... and then selecting the VBoxGuestAdditions ISO file. CentOS should mount this at /media/VBox_GAs_6.0.2 Then, install the additions by typing

```
cd /media/Vbox*  
sudo ./VBoxLinuxAdditions.run
```

Reboot the VM (System→Shut down..., then “Reboot”). It should start up in a larger window and no longer require you to click to focus the mouse in the VM window.

To get the screen to resize with my VM window, I also had to switch to the “VBoxVGA” graphics controller (shut the VM down, then go to Settings→Display).

Unmount the Guest Additions ISO by typing “eject” in your VM.

4 Download Quartus 81.1 and the SoC FPGA Embedded Development Suite

Download Quartus Prime Lite Edition (18.1) from <http://fpgasoftware.intel.com>. You may have to sign up for a free account to do this. Select the “Lite” edition, release 18.1.

You can either download the “Combined Files” (6.2 GB), or under “Individual Files,” Quartus Prime (includes Nios II EDS) (2 GB), and the Cyclone V device support (1.1 GB).

Now, select the “Standard” edition (remain on the 18.1 release), select “Additional Software” and download the “Intel SoC FPGA Embedded Development Suite Standard Edition.”

5 Install Quartus 18.1 and the SoC EDS

Your VM needs some way to access the Quartus installation files. One way is to mount a directory from your host machine.

Move the downloaded Quartus files into a folder on your host machine named, e.g., “Quartus.”

On the VM, select Devices→Shared Folders→Shared Folder Settings and add a share (click on the “add” button). Select the path to the folder in which the downloaded Quartus .tar files resides (don’t use a symbolic link). Name the Folder “Quartus.”

Log into the VM and open a terminal window.

```
sudo mount -t vboxsf Quartus /mnt
cd /mnt
sudo tar xf Quartus*.tar
sudo ./setup.sh
```

Once the Quartus installation wizard starts, direct it to install in `/opt/intelFPGA/18.1`.

Under “Devices,” make sure “Cyclone V” is selected (the others are unnecessary).

Let the installation complete; this will take a while.

Terminate the installer (don’t start Quartus yet).

Install the Embedded Design Suite: Type “`chmod +x /mnt/SoC*`” and “`/mnt/Soc*.run`.”

Specify `/opt/intelFPGA/18.1` as the installation directory (i.e., as you specified earlier).

6 Enable USB JTAG

To program the FPGA on the DE1-SoC board using your virtual machine, on your VM, add the following *udev* rules file:

```
sudo cat > /etc/udev/rules.d/51-altera.rules <<EOF
ATTR{idVendor}=="09fb", ATTR{idProduct}=="6010", MODE="0666"
ATTR{idVendor}=="09fb", ATTR{idProduct}=="6810", MODE="0666"
EOF
```

When the VM is running and you have the DE1-SoC board powered on and connected to your host machine, select Devices→USB Devices→Altera to make the VM see the connected JTAG device.

Now, when you have a DE1-SoC board connected to your computer, “/opt/intelFPGA/18.1/quartus/bin/jtagconfig” should report something like

```
1) DE-SoC [1-1.5.2.2]
   4BA00477   SOCVHPS
   02D120DD   5CSE(BA5|MA5)/5CSTFD5D5/..
```

7 Debugging USB JTAG

Turn on and connect the DE1-SoC board. Check that the Altera “USB Blaster” appears:

```
$ lsusb | grep 09fb:
Bus 001 Device 036: ID 09fb:6810 Altera
```

If it is not listed, the board may not be powered on or the JTAG USB port (near the power jack) may not be connected to the workstation. Note that your bus and device number may differ.

Check the permissions on the port, based on the bus and device you saw above:

```
$ ls -l /dev/bus/usb/001/036
crw-rw-rw- 1 root root 189, 35 Jan 17 21:16 /dev/bus/usb/001/036
```

Finally, verify that the JTAG daemon is able to locate the hardware:

```
$ jtagconfig
1) DE-SoC [1-1.5.2.2]
   4BA00477   SOCVHPS
   02D120DD   5CSE(BA5|MA5)/5CSTFD5D5/..
```

The most common problem is incorrect permissions (no world write) on the device, e.g.,

```
$ jtagconfig
No JTAG hardware available
$ ls -l /dev/bus/usb/001/036
crw-rw-r-- 1 root root 189, 35 Jan 17 21:16 /dev/bus/usb/001/036
$ sudo killall jtagd
$ lsusb | grep 09fb:
Bus 001 Device 041: ID 09fb:6810 Altera
$ sudo chmod 666 /dev/bus/usb/001/041
$ jtagconfig
1) DE-SoC [1-1.5.2.2]
   4BA00477   SOCVHPS
   02D120DD   5CSE(BA5|MA5)/5CSTFD5D5/..
```

The *udev* rule is meant to automate this; check that the `51-altera.rules` file is correct.