ROCCC, ModelSim, and Xilinx setup in Linux

These instructions were put together on shell.cec.wustl.edu.

ROCCC compiler setup
1. A new directory exists in your home directory called “~/roccc”.
2. Run “rocc-eclipse” to start the Eclipse environment containing the ROCCC compiler plugin. It will first prompt you for the location of your workspace. You can either keep the default, or change it to another folder location (ex. “~/eclipse/cse565”).
3. Configure the ROCCC plugin to use your local “roccc” distribution.
   1. On the menu bar, click ROCCC → Settings → Preferences.
   2. In the left pane, click ROCCC. Then, for the ROCCC Distribution Directory, click Browse to browse to “~/roccc”.
   3. Click Verify ROCCC Distribution Folder to verify that it has found a valid ROCCC distribution.
   4. Click Ok to commit the settings.
4. It should next say that the installation was a success and ask if you would like ROCCC to load up the distributed examples. Click Yes. The examples will then be loaded in the Project Explorer.
5. Verify that you can run the ROCCC compiler.
   1. Go to the examples and load the BitWidth example program. (Expand ROCCCExamples / src / modules / BitWidth. Then double-click on “BitWidth.c”.)
   2. Start a build by either clicking on ROCCC → Build, or clicking on the hammer icon in the ROCCC toolbar.
   3. Use all the default options and click Finish.
   4. The result of the build will display in the Console window pane under where it says ROCCC Toolchain. It should say “Compilation of BitWidth.c finished.”.
   5. In the Project Explorer, verify that a vhdl folder was created containing the VHDL code of the BitWidth module.

Setup Linux environment to run ModelSim and Xilinx
1. On a CEC machine (such as shell.cec.wustl.edu), add the following to the “.cshrc.mine” or “.cshrc” startup script file:

   setenv LM_LICENSE_FILE 27007@licensing01.seas.wustl.edu
    setenv XILINXD_LICENSE_FILE 27002@licensing01.seas.wustl.edu
    set path=( $path /opt/modelsim/bin )
    set path=( $path /opt/xilinx/bin/lin64 )
   
2. Type “vsim &” to run ModelSim.
3. Type “ise &” to run Xilinx ISE.