

1. Running Cadence Verilog

The purpose of this part of the tutorial is to take you through the steps that are required in order to use Cadence Verilog on the ECE unix machines in order to write and simulate verilog code. There are two sections of this part of the tutorial. The first section is the setup section which will prepare your personal unix environment to use Cadence and will only have to be completed once. The second section includes general instructions for using Cadence Verilog on unix and will be relevant every time you use Verilog in unix.

1.1 Section #1 - Setup (i.e. do this once):

In order to use unix tools for 18-360 such as Cadence Verilog, you must set up your unix environment to use these tools. This involves setting environment variables and including certain directories in your path so that you can use the applications required in 18-360. We have written scripts that configure your environment for you so that all you have to do is execute these scripts in order to set all the necessary variables.

There are two versions of the environment variable script - you must run the one that is appropriate for the shell that you use. As of F02, all new ECE accounts default to use the bash shell. However, some users may be running CSH or TCSH as well.

1. Telnet to your favorite ECE unix machine (see Appendix A for available machines)
2. Type "echo \$SHELL" at the prompt.
 - If "/usr/local/bin/bash" is displayed, you are running BASH as a shell
 - If "/usr/local/bin/tcsh" is displayed, you are running TCSH as a shell

If you're running BASH, do the following:

1. Type "cd ~" to change to your home directory
2. Type "pico .bashrc" to open the file ".bashrc" using the text editor "pico"
3. Type ". /afs/ece/class/ee360/bin/setvar360" on a single line in the file (note there is a space between the "." and the "/afs, etc.")
4. Hit "ctrl o" to save your changes
5. Hit "ctrl x" to exit pico
6. Type "pico .profile" to open the file ".profile" using the text editor "pico"
7. Type ". ~/.bashrc" on a single line in the file (note there is a space between the "." and the "~")
8. Hit "ctrl o" to save your changes
9. Hit "ctrl x" to exit pico
10. Log out and log back in to save your changes!

If you're running TCSH, do the following:

1. Type "cd ~" to change to your home directory
2. Type "pico .cshrc" to open the file ".cshrc" using the text editor "pico"

3. Type “source /afs/ece/class/ee360/bin/setvar360_tclsh” on a single line in the file
4. Type “ctrl o” to save your changes
5. Type “ctrl x” to exit pico
6. Type “source ~/.cshrc”, or log out and log back in to save your changes!

1.2 Section #2 - Usage (i.e. do this every time):

In this section we will go through the process of logging in to a unix machine, re-directing the output of our session to the machine we are working from, editing some verilog source code, and simulating that code! You can reproduce these steps in order to write and simulate your own code at a later date.

Section 2a: Re-Directing Your Display

NOTE: This section does not apply if you are actually logged in to (i.e. sitting at) one of the greek letter machines in the ECE undergraduate computing cluster.

Xwin-32 is an application that allows you to redirect the display of your unix session to a remote location such as a lab computer or your laptop. This enables you to use “point and click” unix applications from a remote location. This is useful if you want to edit source code using Xemacs or if you want to open multiple terminal windows without logging in several times. You must do the following to use Xwin-32:

1. Start the Xwin-32 application on the machine that you want to use. This will cause an “X” icon to appear in your system tray. If a “connection method” window appears, hit cancel.
2. Double click on the “X” icon to open up the configuration panel for Xwin-32.
3. Click on the “security” tab. This allows you to control what machines are allowed to start remote sessions on your computer.
4. Click “Add...” and specify the host that you are going to be running your remote applications on. This is probably one of the [Greek Letter].ece.cmu.edu machines but *.ece.cmu.edu and *.cmu.edu are also acceptable if you are not terribly worried about security.
5. Telnet to your favorite ECE unix machine (see Appendix A for available machines)
6. You must now determine the host-name of the Windows machine you are using. Type “last -n 1” at the unix prompt which will produce a line like the following:

```
“joeuser pts/10    SWISS      Wed Oct 16 15:14  still logged in”
```

This line tells me that I am currently logged in through a machine called “SWISS”. Sometimes the “last” command may display something like “SWISS.ECE.CMU” which is obviously not a valid host name. In this case, you can either use just “SWISS” or the whole host name “SWISS.ECE.CMU.EDU”

7. You must now set your DISPLAY environment variable to point to the machine you using:
 - For BASH: type “export DISPLAY=[your HOST]:0.0” where [your HOST] is replaced by the host you found in step #6 (i.e. SWISS in this case)
 - For TCSH: type “setenv DISPLAY [your HOST]:0.0”

You have now re-routed your display to a remote machine. Typing “xterm&” should open up a terminal window on your remote machine. If you get an error about access or the window does not open, make sure you have specified the hosts correctly in both the security tab of Xwin-32 and your DISPLAY variable.

Section 2b: Editing Your Source Code

Xemacs is a general purpose word processor/text editor that is used in “unix land.” If you want to become an emacs hacker you can configure the program to do everything from read your e-mail to make your morning cup of coffee. This tutorial is only going to tell you how to use it to edit source code.

1. Change to your home directory by typing: `cd ~`
2. Create a new directory for this verilog test by typing: `mkdir 360_test`
3. Change to the newly created directory by typing: `cd 360_test`
4. Copy the verilog test file to your directory by typing:
`cp /afs/ece/class/ee360/public/vlog_test.v ~/360_test`
5. Make sure you have followed the instructions in Section 2.a to redirect your session output
6. Type “`xemacs vlog_test.v &`” to open the verilog file in xemacs! (The “`&`” makes sure that xemacs doesn’t take over your terminal process!)

Section 2c: Simulating Your Source Code

1. Make sure you are in the “360_test” directory you created
2. Type “`vlog vlog_test.v`” in order to simulate your code. (if you had more than one verilog file you could type “`vlog *.v`” to simulate all of them.)
3. For the first time you run it, the simulation should produce the following output:

```
nverilog: v03.11.(s020): (c) Copyright 1995 - 2001 Cadence Design Systems, Inc.
nverilog: v03.11.(s020): Started on Oct 11, 2002 at 14:50:50
nverilog
```

```
  vlog_test.v
```

```
file: vlog_test.v
```

```
  module worklib.testBench:v
```

```
    errors: 0, warnings: 0
```

```
  module worklib.rec2_nand:v
```

```
    errors: 0, warnings: 0
```

```
  module worklib.test_module:v
```

```
    errors: 0, warnings: 0
```

```
    Caching library 'worklib' ..... Done
```

```
  Elaborating the design hierarchy:
```

```
  Building instance overlay tables: ..... Done
```

```
  Generating native compiled code:
```

```
    worklib.test_module:v <0x0fa91895>
```

```
      streams: 2, words: 653
```

```
  Loading native compiled code: ..... Done
```

```
  Building instance specific data structures.
```

```
  Design hierarchy summary:
```

```
    Instances Unique
```

```
    Modules:      3    3
```

```
    Primitives:   6    2
```

```
    Registers:    1    1
```

```
    Scalar wires: 6    -
```

```
    Expanded wires: 7    1
```

```
    Initial blocks: 1    1
```

```
  Writing initial simulation snapshot: worklib.testBench:v
```

```
  Loading snapshot worklib.testBench:v ..... Done
```

```
ncsim> source /usr/cds/ldv-3.1/tools/inca/files/ncsimrc
```

```
ncsim> run
```

```
  0 ABCDEF= 0, G=1
```

```
 10 ABCDEF= 1, G=0
```

```
 20 ABCDEF= 2, G=1
```

```
 30 ABCDEF= 3, G=0
```

```
 40 ABCDEF= 4, G=0
```

```
 50 ABCDEF= 5, G=0
```

```
 60 ABCDEF= 6, G=1
```

```
 70 ABCDEF= 7, G=0
```

```
 80 ABCDEF= 8, G=1
```

```

    90 ABCDEF= 9, G=0
    100 ABCDEF= 10, G=1
Simulation complete via $finish(1) at time 110 NS + 0
./vlog_test.v:37    #10 $finish;
ncsim> exit
neverilog: v03.11.(s020): Exiting on Oct 11, 2002 at 14:50:57 (total: 00:00:07)

```

If the “vlog vlog_test.v” command doesn’t produce the above output, something is wrong so go back and make sure you followed all of the instructions! After you correctly simulate this once, a directory called “INCA_libs” will be created in the “360_test” directory. If you run the same simulation (i.e., vlog vlog_test.v) again, it will produce the following output only:

```

neverilog: v03.11.(s020): (c) Copyright 1995 - 2001 Cadence Design Systems, Inc.
Loading snapshot worklib.testBench.v ..... Done
ncsim> source /usr/cds/ldv-3.1/tools/inca/files/ncsimrc
ncsim> run
    0 ABCDEF= 0, G=1
    10 ABCDEF= 1, G=0
    20 ABCDEF= 2, G=1
    30 ABCDEF= 3, G=0
    40 ABCDEF= 4, G=0
    50 ABCDEF= 5, G=0
    60 ABCDEF= 6, G=1
    70 ABCDEF= 7, G=0
    80 ABCDEF= 8, G=1
    90 ABCDEF= 9, G=0
    100 ABCDEF= 10, G=1
Simulation complete via $finish(1) at time 110 NS + 0
./vlog_test.v:37    #10 $finish;
ncsim> exit

```

Appendix A: Available ECE Unix Machines as of Fall 02

Sun Ultra 1:

```

aqua    black    blue    brick    brown    carnation
cyan    eggshell fuchsia grey    indigo   ivory
lemon   magenta  mauve   olive   orange   periwinkle
purple  red      salmon  taupe   teal     violet
white

```

SunBlade 100:

```

alpha   aphrodite apollo  ares    artemis  athena
beta    chi        delta   demeter dionysys epsilon
eta     gamma     hephaestus hera    hermes   iota
kappa   lambda    mu      nu      phi      pi
poseidon psi       rho     sigma   tau      theta
xi      zeta

```