Using the EECS 470 tools

Purpose:
This document is intended as a quick reference for the tools you'll be using in EECS 470. In addition to describing basic usage of VCS it enumerates the main points of course-specific tools including the Makefile & synthesis scripts. Be sure to check the newsgroup if you have a question that is not answered within this document.

Preparing the Makefile for a new project:

A) Copy the Makefile from the course homepage.

B) Change the 'FILES =' line so that it lists your test-bench and module verilog files.

C) Change the 'SYNFILES =' line so that it lists your test-bench verilog file and your modules' .vg files. The .vg files are produced by Design Compiler during synthesis.

Basic 470 commands (After preparing the Makefile):

A) Running the module's test-bench
   a. Type: 'make' or 'make run'

B) Running the GUI post-processor (logic analyzer, logic viewer, etc).
   a. Make sure your test-bench contains these seven lines in an initial block:

```
`ifdef DUMP
   `ifdef XL
      $dumpvars;
   `else
      $vcdpluson;
   `endif
`endif
```

   b. Type: 'make wave'

C) Synthesizing your design
   a. Copy default.scr from the course webpage
   b. Edit default.scr such that:
      1) The 'design_name = ' line names your top-level module. Not your test-bench!
      2) The read line lists the files containing all of your design's modules.
      3) The 'clock_name = ' line matches the name of the clock signal in your design.
      4) The 'reset_name = ' line matches the name of the reset signal in your design.
   c. Type: 'make syn'

D) Viewing test-bench waveforms for the structural verilog produced during synthesis
   a. Following synthesis, type 'make synwave'

E) Removing the temporary files created during compilation
   a. Type: 'make clean'

Using the GUI front-end to VCS:

A) Functions applicable to all windows:
   a. To add a module or signal to any of the views, drag and drop that signal from the hierarchy window to your target using the middle mouse button.
   b. To link the waveform, source and logic windows together, select the same "channel" (all A, or all B) by using the drop-box with the yellow chain-link icon in the upper right corner of each window. This will allow time changes in any window to update all other windows.

B) Waveform window:
   a. To change the display of a signal in the waveform window from binary to another format, right click on the signal values to the right of a signal name. (2'b00, 1'bx, etc)
   b. Time measurements between signal events can be accomplished using cursors. Place the 'C1' cursor on the first event with the left mouse button. Place the 'C2' cursor on the second event with the middle mouse button. Cursor placement will snap to the nearest signal change for whichever signal the mouse is over.

C) Source window:
   a. To display signal values in the source window, select 'Display -> Show Values'

Using Design Analyzer:

Default Design Analyzer scripts are not available on the course homepage since design_analyzer can accept the same script format as dc_shell. The example scripts used in discussion are available at: /afs/engin.umich.edu/caen/generic/mentor_lib-D.1/public/eecs470 The following points assume you're using the example scripts & have correctly named modules, files and signals as described in the preceding section on synthesis.

A) To view Design Compiler's interpretation of your verilog type: design_analyzer –f dav.scr

B) To view you're synthesis result type: design_analyzer –f dadb.scr

   a. Double click the module you would like to expand. This will display the I/O ports for the module.

   b. To view the internal logic, double click the module once again.

C) Remember, DC's interpretation of your verilog code will be different than the synthesized result. (ie. The verilog interpretation may display sequence generators when it in-face synthesizes delay flip-flops)