Lab 1: Tool Familiarization

August 22, 2016

Introduction

In this exercise, you will be taken through the process of using the Synopsys Verilog Compiler and Simulator (VCS) for compiling and simulating verilog HDL code, as well as viewing the timing diagram of the simulation through DVE. You need not understand the actual operation of the circuit since the goal of this exercise is to familiarize yourselves with the tools that will be used for the remainder of the training.

Setting up the environment

1. In order to run VCS, we must first set-up the necessary UNIX environment variables. Open a text editor and create a file named set_synopsys.sh. This will be your set-up script and should be executed everytime you open a new terminal for use with VCS and the synthesis tool. The contents of the file should be the following:

```
#!/bin/bash
## License
export LM_LICENSE_FILE=27000@10.158.16.12
## VCS
export VCS_HOME=/eeei/tools/synopsys/vcs-mx/H-2013.06-SP1/
export VCS_BIN=/eeei/tools/synopsys/vcs-mx/H-2013.06-SP1/bin
## SYN
export SYN_BIN=/eeei/tools/synopsys/syn/I-2013.12/bin
## PATH
export PATH=$VCS_BIN:$SYN_BIN:$PATH
```

2. Run the script your created in the command line to set-up the necessary environment variables. Make sure that you run the following commands in the same directory where you created the set_synopsys.sh script.

```
[snap@artanis ~]$ source set_synopsys.sh
[snap@artanis ~]$
```

3. To make sure that all necessary environment variables have been set properly, run the following commands. The terminal should display the values set in the script file.

```
[snap@artanis ~]$ echo $LM_LICENSE_FILE
27000@10.158.16.12
[snap@artanis ~]$ echo $VCS_HOME
/eeei/tools/synopsys/vcs-mx/H-2013.06-SP1/
[snap@artanis ~]$ echo $VCS_BIN
/eeei/tools/synopsys/vcs-mx/H-2013.06-SP1/bin
[snap@artanis ~]$ echo $SYN_BIN
/eeei/tools/synopsys/syn/I-2013.12/bin
[snap@artanis ~]$
```

Using VCS for compilation and simulation

1. Now that we have already set-up the necessary environment variables, we can now proceed to using the tools for compilation. Using the terminal where you ran the set-up script, unpack the lab_01.tar file you downloaded from the class web page. Proceed to the lab_01 directory. This directory should contain the necessary source verilog files that we will be using for this exercise. Don't worry if you still cannot

understand the contents of the verilog files because everything will be clearer as we progress through the course. If you wish to use another terminal, just invoke the set-up script again.

```
[snap@artanis ~]$ tar -xvf lab_01.tar
[snap@artanis ~]$ cd lab_01
[snap@artanis lab_01]$ ls
tb_updown.v updown.v
[snap@artanis lab_01]$
```

2. To use VCS for both compilation and simulation, your source verilog files should contain at least two modules: the design to be tested, and the testbench. Both modules can be placed in a single file or in separate files. In this exercise, the design to be tested is placed in the updown.v file while the testbench is placed in the tb_updown.v file. Compile both files using VCS by running the following command. If the command line flags an error that says the vcs command is not found, make sure that you have properly set-up the environment variables before running the command again.

```
[snap@artanis lab_01]$ vcs updown.v tb_updown.v -full64 -debug_pp
                         Chronologic VCS (TM)
       Version H-2013.06-SP1_Full64 -- Wed Apr  2 14:21:53 2014
               Copyright (c) 1991-2013 by Synopsys Inc.
                         ALL RIGHTS RESERVED
This program is proprietary and confidential information of Synopsys Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.
Parsing design file 'updown.v'
Parsing design file 'tb_updown.v'
Top Level Modules:
       tb_updown

                                    .
                                    .
                                    .

../simv up to date
CPU time: .146 seconds to compile + .063 seconds to elab + .224 seconds to link
[snap@artanis lab_01]$
```

3. If the compilation was successful, VCS should have created an executable simulation file named simv (by default) along with other files needed and generated by VCS. Run the simv file in the command line to execute the simulation.

```
[snap@artanis lab_01]$ ls
csrc  simv  simv.daidir  tb_updown.v  updown.v
[snap@artanis lab_01]$ ./simv
Chronologic VCS simulator copyright 1991-2013
Contains Synopsys proprietary information.
Compiler version H-2013.06-SP1_Full64; Runtime version H-2013.06-SP1_Full64;
VCD+ Writer H-2013.06-SP1_Full64 Copyright (c) 1991-2013 by Synopsys Inc.
$finish called from file "tb_updown.v", line 26.
$finish at simulation time                  120
           V C S   S i m u l a t i o n   R e p o r t
Time: 120
CPU Time:      0.380 seconds;       Data structure size:   0.0Mb
Wed Apr  2 14:31:14 2014
[snap@artanis lab_01]$
```

Using DVE viewing simulation results

1. If the simulation was successful, a waveform dumpfile named tb_updown.vpd should have been generated. To view this waveform, run DVE from the command line. This will open up a GUI window for the tool, as shown in figure

```
[snap@artanis lab_01]$ ls
csrc  simv  simv.daidir  tb_updown.v  tb_updown.vpd  ucli.key  updown.v
[snap@artanis lab_01]$ dve -full64 &
[1] 27554
[snap@artanis lab_01]$
```

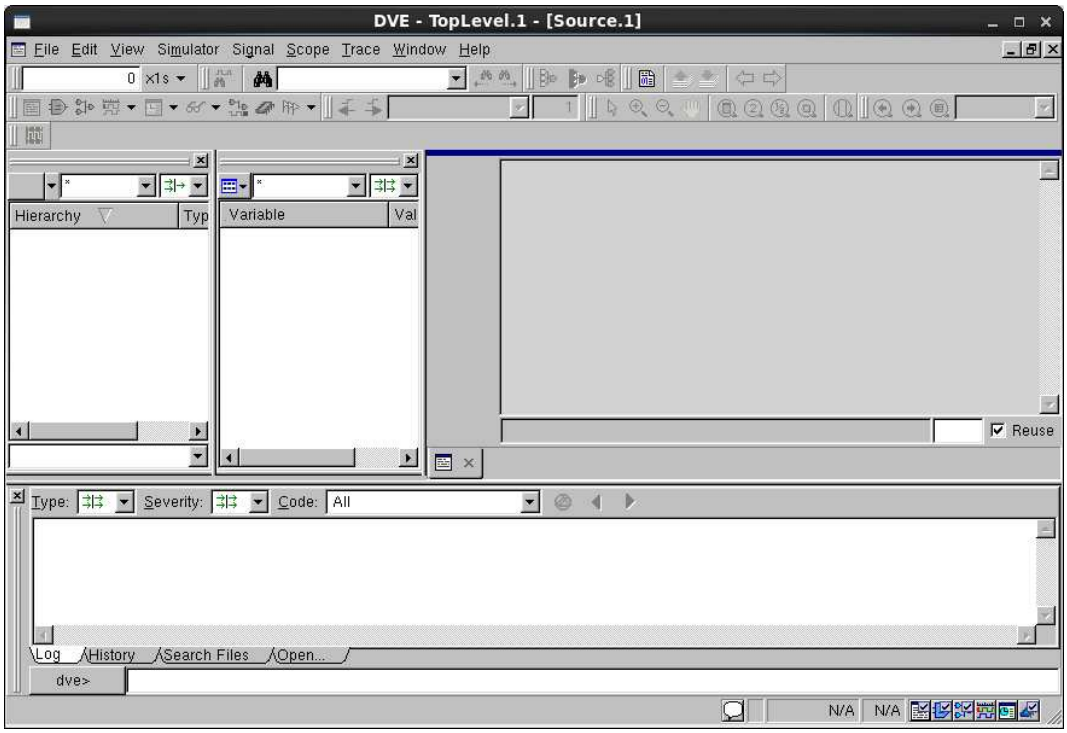

Figure 1: DVE Graphical User Interface

2. On the main toolbar, click File then select "Open Database" (or press Ctrl + O). This will open a window similar to the one shown in figure 2. Select the dumpfile (tb_updown.vpd) and click Open to load the dumpfile into DVE.

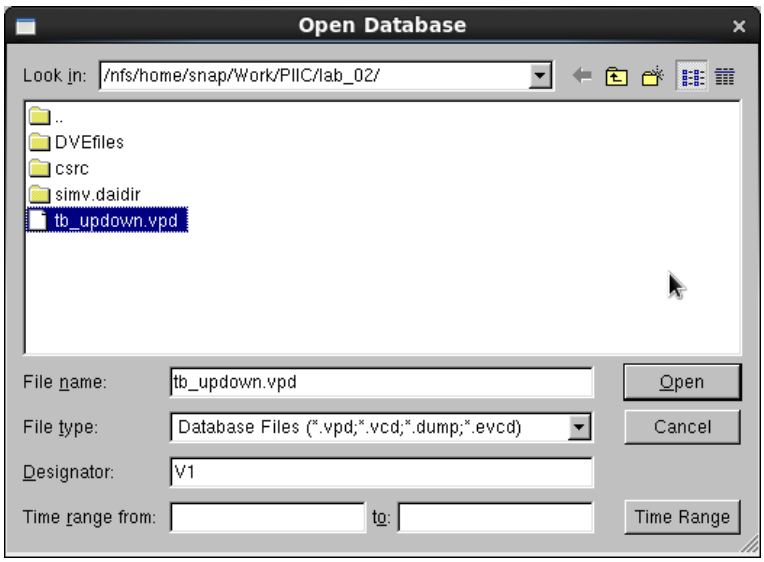

Figure 2: DVE Open Database Window

3. Figure 3 shows the DVE GUI after loading the dumpfile. There are three panes of interest shown. The hierarchy pane shows the hierarchy of modules of the compiled verilog files. The data pane shows the variables, signals, and registers of the currently selected module in the hierarchy pane. The console pane shows a console that can be used to execute DVE commands. Do not worry if you cannot find all three

panes. You can toggle them into view by selecting Window -> Panes in the main toolbar along with other panes.

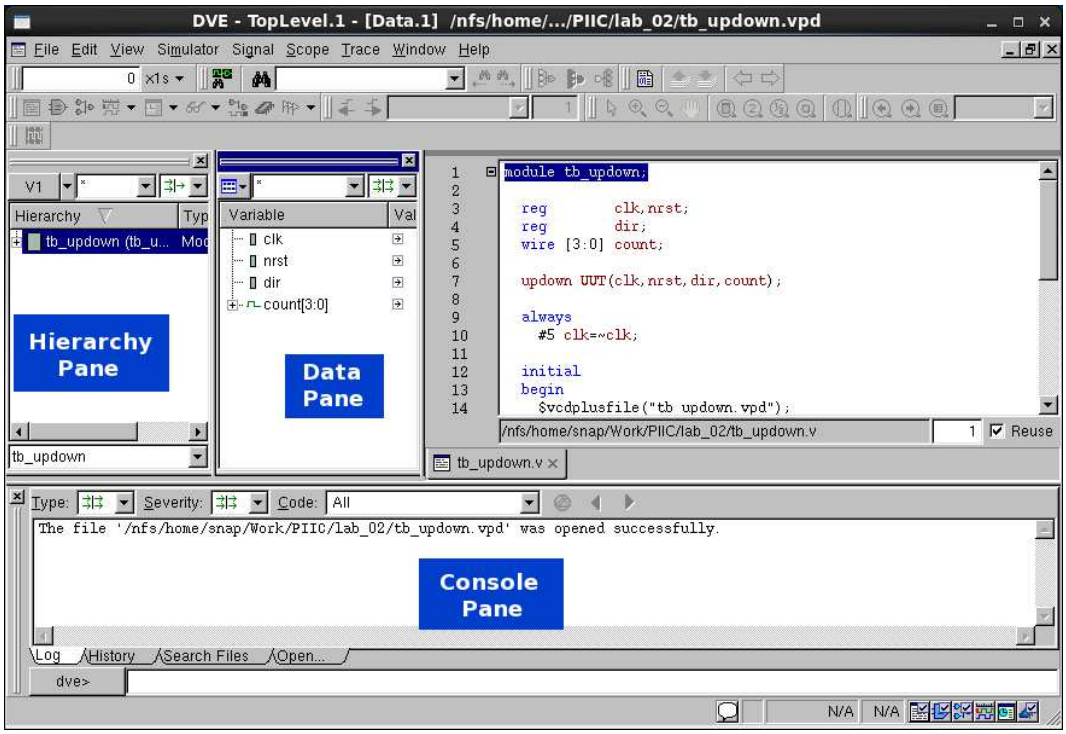

Figure 3: DVE GUI with loaded dumpfile

4. To view the signals of the testbench, select tb_updown in the hierarchy pane, then select all four signals (clk, nrst, dir, and count) in the data pane by holding Ctrl and clicking on each signal. While selected, right-click anywhere on the GUI to pop-up a menu and select "Add to waves" -> "New wave view". A new window will pop-up similar to the one shown in figure 4. This waveform viewer window will show a timing diagram of the simulation dumped into the dumpfile. A signal pane will display all signals that were selected. You can add additional signals to the waveform viewer by going back to the main DVE GUI, selecting a signal, right-clicking to bring up the menu, and selecting "Add to waves" -> "Add to Wave.X" where Wave.X is the name of the waveform viewer window (alternatively, you can just select the signals and press Ctrl + 4).

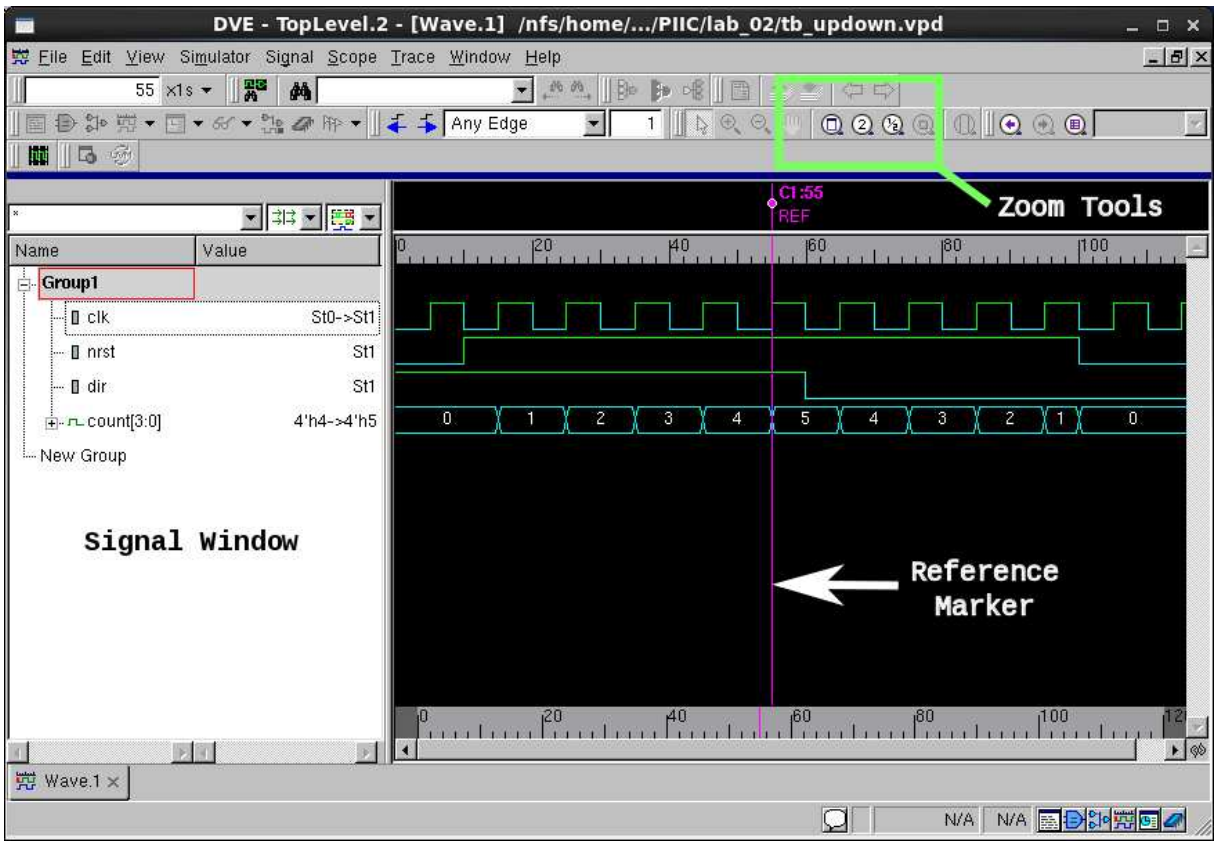

Figure 4: Waveform Viewer

5. Also present in the waveform viewer is the reference marker, which can be moved around the timing diagram by simply clicking on the timing diagram/waveform. The signal window shows the respective values of the signals in the timing diagram at the time marked by the reference marker. You can zoom in and out of the waveform by using the zoom tools (Zoom Full, Zoom In 2x, Zoom Out 2x). You can also navigate through the waveform using the arrow buttons and sliding navigation bar located at the bottom of the timing diagram. The ruler on the top of the waveform shows the time axis of the waveform displayed, while the one at the bottom shows all time values for the entire simulation, not just the one currently displayed. You can zoom in to a specific part of the waveform displayed by clicking on the top ruler and dragging it to the left or right to select the region you wish to zoom in. You can also zoom to any part of the simulation (not just the one currently displayed) by performing the same action on the bottom ruler. Thus, selecting the whole bottom ruler would just display the waveform of the whole simulation (similar to a Zoom Full).