

**Department of Computer Science and Engineering  
State University of New York at Buffalo**

**CSE 497/597: Intro to VLSI (Fall 2004)**

**Cadence Verilog-XL Setup and Jumpstart**

**Setting up Cadence Verilog-XL on ENG machines:**

1. You need to add the following line to the `~/.cshrc` file:  
`set path = ($path /eng/tools/cds03/ldv/tools/bin)`
2. Now source the file by typing the following at the unix prompt:  
`source ~/.cshrc`
3. For CSE accounts, add the following to `~/.cshrc_cds03` file:  
`setenv CDS_LIC_FILE 1701@hadar.cse.buffalo.edu`  
`set path = ($path /apps/cds03/ldv/tools/bin)`
4. The `.cshrc_cds03` file needs to be sourced every time the tool is started. This can be typing the following at the unix prompt:  
`source ~/.cshrc_cds03`

**Cadence Verilog-XL Jumpstart**

In this jumpstart we look at how to use the Cadence Verilog-XL tool to simulate and generate waveforms from Verilog code. The example used here implements a simple boolean function  $(x,y) = (AB+C',C')$ . The code is provided here for you to work through this example. To learn coding in Verilog, you may refer to the links provided in the project description and the online help in Cadence.

First, create a working directory by using the `mkdir` command and `cd` into it. The next step is to create the Verilog source file that you intend to simulate. You can use any text editor such as `vi`, `vim`, `emacs`, `pico`, etc. Create the file `example.v` (Verilog source files typically have an extensions of `.v`):

```
//Stimulus for simple circuit
module stimcrct;
  reg A, B, C;
  wire x,y;
  circuit_with_delay cwd (A, B, C, x, y);
  initial
```

```

begin
    $shm_open();
    $shm_probe(A, B, C, x, y);
    A=1'b0; B=1'b0; C=1'b0;
    #100
    A=1'b1; B=1'b1; C=1'b1;
    #100
    $shm_close();
    $stop;
end
endmodule

//Description of circuit with delay
module circuit_with_delay (A, B, C, x, y);
    input A, B, C;
    output x, y;
    wire e;
    and #(30) g1 (e, A, B);
    not #(20) g2 (y, C);
    or #(10) g3 (x, e, y);
endmodule

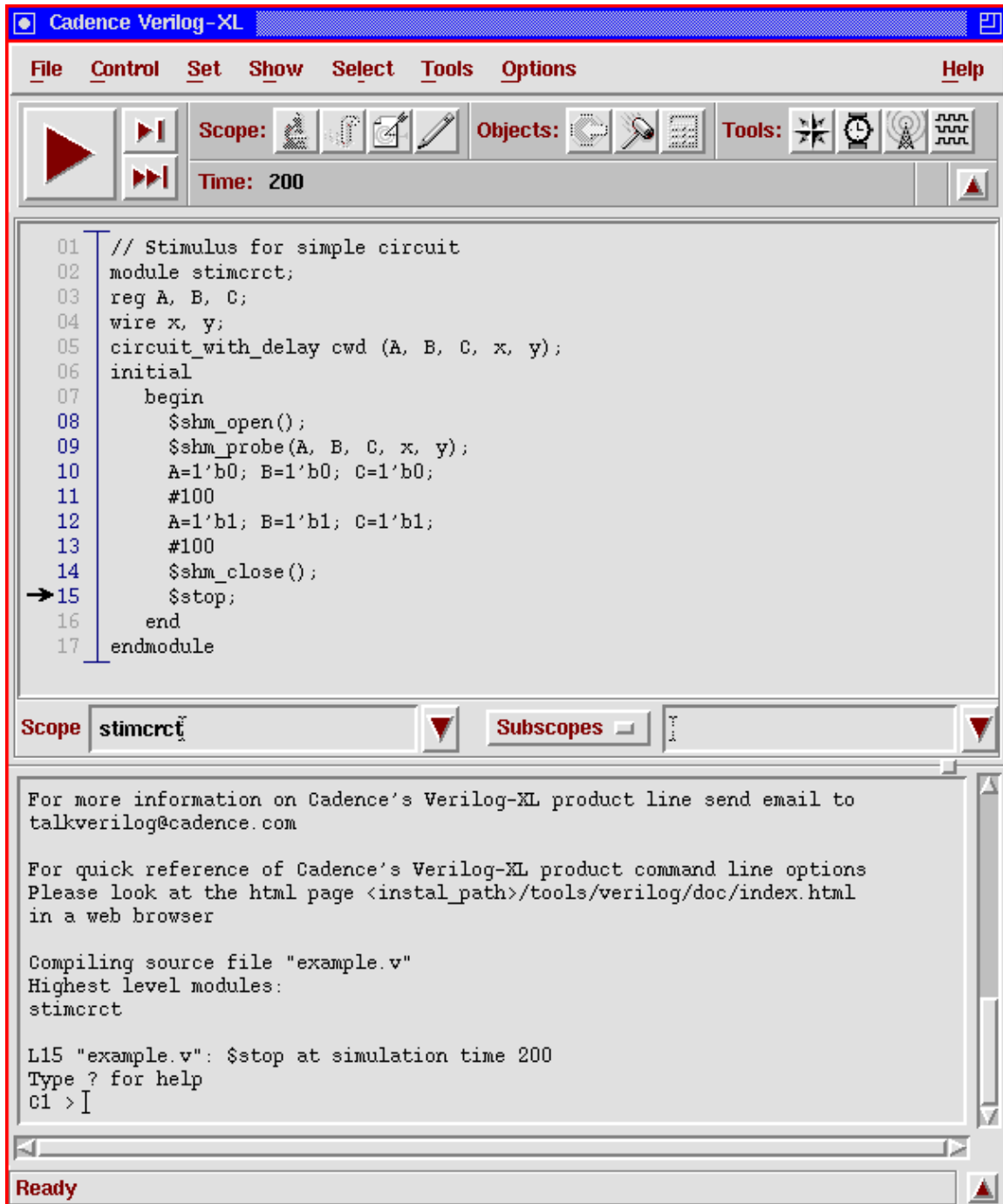
```

There are three cadence specific system commands in this code. These are **\$shm\_open**, **\$shm\_probe** and **\$shm\_close**. These cause a waveform subdirectory to be created during the simulation that saves the information for the signals and registers that are in the argument list of the **\$shm\_probe** command. **\$shm\_close** then closes the waveform files. The default name of the subdirectory is **waves.shm**.

With the file `example.v` in the working directory we are ready to simulate the code. Before simulating, source the file `~/cshrc_cds03`. From the working directory enter the following command at the unix prompt

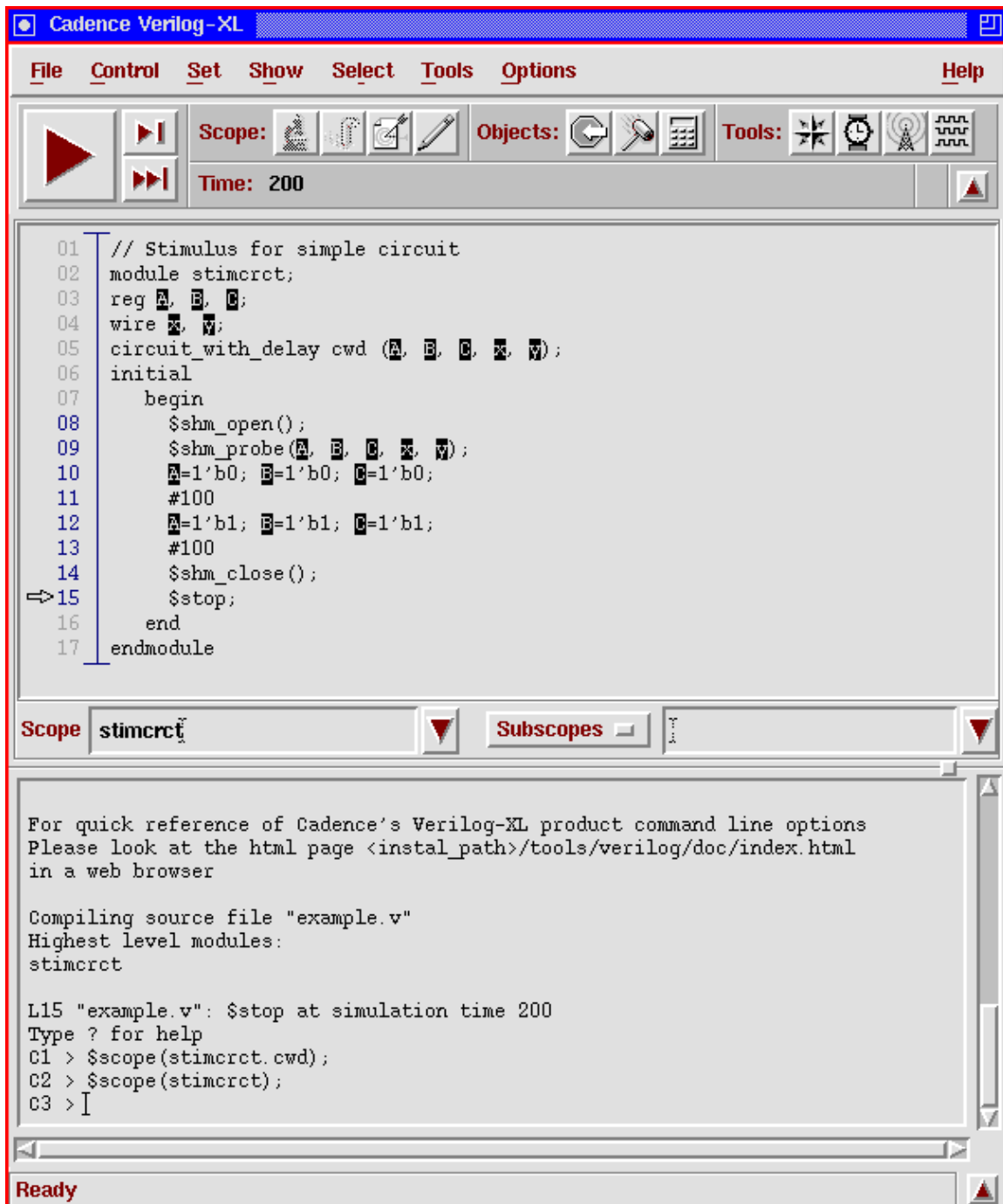
```
verilog +gui example.v &
```

The following window will open and the file will be simulated (if no errors are found). If errors are found, you should note the message and exit the simulator and fix your source code. If you are familiar with `vi`, you can invoke this editor directly from the File menu of the Verilog window, modify this file, save it, then choose “Reinvoke” from the File menu.

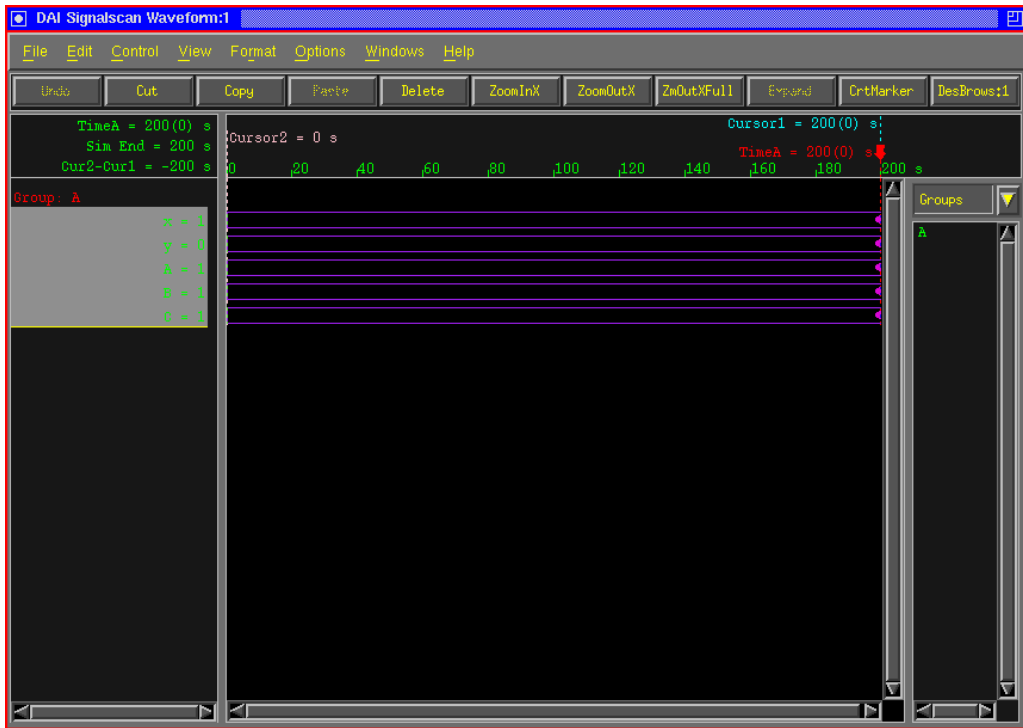


There are several things to note on this window. An important button is the one in the upper right-hand corner labeled "Help". Clicking on this button will open the Cadence help utility (called OpenBook) and display a list of topics on the use of the Verilog tools. Notice that the bottom portion of the window indicates that the simulation has stopped at time 200. This is because we used a \$stop that was scheduled to occur at time 200. In the top part of the window only the testbench is displayed. If you want to view circuit module you can click on the down arrow to the right of the "Subscopes" button and you will see that cwd may be chosen.

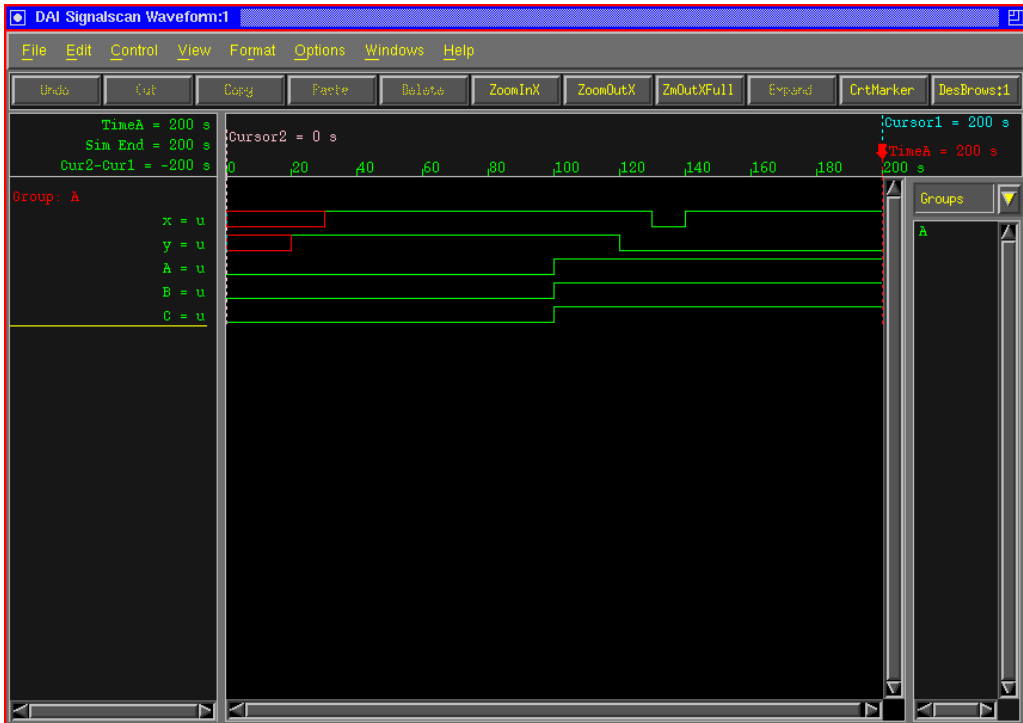
Next, we want to view the waveforms. To do this, choose the “Select” button at the top of the window and click on the “Signals” option from the menu. This will cause all signals to be reverse highlighted as shown below:



This is done to tell the waveform viewer which signals (that were recorded in the waves.shm subdirectory) we wish to examine. Next, we will invoke the waveform viewer by clicking on the “tools” button and selecting “waveform”. A new window will open that looks like this:



The purple lines indicate that the state of the signals is unknown, we need to resimulate or design to update the window. To do this, go to the “File” button on the Verilog GUI window and select option “Reinvoke”. This will cause the simulation to run again and the waveform window will be updated and should look like this:



Notice that **x** and **y** are red lines initially during the simulation. This is because they are set to the value **z** since the simulator cannot determine a logic level until the gate delays have been accounted for.

Acknowledgement:

This document has been adapted from

[http://www.engr.smu.edu/~mitch/class/3381/verilog\\_intro.pdf](http://www.engr.smu.edu/~mitch/class/3381/verilog_intro.pdf)