ORG DIGITL COMP LAB (EECS 112L)
Cadence Incisive Tutorial

EECS Department
Henry Samueli School of Engineering
University of California, Irvine
Winter 2016

This very brief tutorial explains how to set up the project folder and how to simulate VHDL designs in Cadence.

Simulations in Cadence

Getting familiar with UNIX/LINUX

After you logged in your account, you can make a folder using:

mkdir *folderName*

You can navigate into the folder by typing:

cd *folderName*

There are different text editors available to make a text file. A simple text editor available on the server is emacs or gedit. To make a new text file using emacs/gedit you can use the following command:

emacs *newFileName.fileExtention* &

or

gedit *newFileName.fileExtension* &

If you use this command with an already existing file name it will open the old file. To delete a file or folder use the following command:

rm *FileName.fileExtention* for file

rm -rf *folderName* for folder

For more comprehensive information about Unix/Linux commands please see "Linux_commands" in the class website

Alternatively, there are many useful tutorials and references available online.

Cadence Startup Files

Cadence Incisive is installed on the EECS servers (Sun SPARC servers: malibu or vivian). In order to be able to use this program you have to specify the installation location by typing the following commands in the terminal:

```
source /usr/skel/syswide.cshrc
source /ecelib/eceware/profile/cadence.linux
source /ecelib/eceware/profile/cadence61
source /ecelib/eceware/profile/mentor09
setenv T_DIR/ecelib/eceware/cadence61/tk_2011
```

Alternatively, from your home folder open the *.cshrc* file in a text editor and copy the above commands at the end of it (These commands are also available in source.csh file in the class website).

With this method you don't have to type them every time you log into your account. In order for these commands to take effect for the first time, you have to log out and log in again. Make sure you do not remove the content of the *.cshrc* file. Just add these lines at the end and save the file.

Setting up Project Folder

For each project you create, you have to make a folder. In order to keep you Linux/Unix home page clean, I suggest you first make a folder for this course and make all your project folders inside the course folder. To set up a project folder follow these steps:

1. Make a directory for your project. (e.g. /HW1)
2. Make a text file named cds.lib inside the project folder and copy the content shown in the following box into it. (You can also download this file from the class website).

```
DEFINE std/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/STD
DEFINE synopsys/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/SYNOPSYS
DEFINE ieee/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/IEEE
DEFINE ambit/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/AMBIT
DEFINE vital_memory/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/VITAL_MEMORY
DEFINE ncutils/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/NCUTILS
DEFINE ncinternal/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/NCINTERNAL
DEFINE ncmodels/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/NCMODELS
DEFINE cds_assertions/ecelib/eceware/cadence61/ius62/tools.sun4v/inca/files/CDS_ASSERTIONS
DEFINE work work
```

This file specifies the library paths for simulator.

3. Make a text file hdl.var. Save it with no contents.
4. Make a folder named *work*. This is where your work files will be stored.

You should follow these steps for each new project you want to set up.

Simulations in Cadence

Everything explained in this part should be done in the Project Folder. (e.g. HW1)

Creating and Compiling Design Files.

In this course, the digital design will be written in VHDL language. You have to put your VHDL code inside a text file with *.vhd* extension. You can have multiple *.vhd* files each implementing a part of the design. First let's make a file named *orgate.vhd* and copy the following code into the file and save it
(You can also download this file from the class website).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity orgate is
  Port (IN1 : in STD_LOGIC; -- OR gate input
        IN2 : in STD_LOGIC; -- OR gate input
        OUT1 : out STD_LOGIC); -- OR gate output
  end orgate;

architecture Behavioral of orgate is
begin
  OUT1 <= IN1 or IN2; -- 2 input OR gate
end Behavioral;
```

This is the VHDL code for a simple two input OR gate. After making this file, we should compile it for possible errors. To compile a VHDL file you can use the following command:

ncvhdl -messages inputFile.vhd

For example:

ncvhdl -messages orgate.vhd

Using the option messages for NCVHDL, compiler will show any errors or warnings on the terminal. If there are any errors you have to fix them until you can successfully compile the file.

Now we will make a System Verilog file to test the VHDL design. In this file named orgatetb.sv, we will instantiate the OR gate made earlier and define the inputs to the gate. This is a very simple test bench file that provides the required inputs to test the OR gate. Make a new text file named orgatetb.sv and copy and paste the following code into it (You can also download this file from the class website).

```
module orgatetb;
 logic input1;
 logic input2;
 wire output1;
 orgate L1     (.IN1(input1),
                .IN2(input2),
                .OUT1(output1));
 initial begin
 input1 = 1'b0;
 input2 = 1'b0;
 #10;
 input2 = 1'b1;
 #10;
 input1 = 1'b1;
 input2 = 1'b0;
 #10;
 input2 = 1'b1;
 end
endmodule
```

Save the file and compile it using the following command.

ncvlog –sv –work work_ directory testbench.sv

For Example

ncvlog -sv -work work orgatetb.sv

If your design includes more .vhd files you have to compile all of them before moving to the next part.

Elaborating the Design.

After compiling all the blocks used in a design, the design should be elaborated. During elaboration process all the blocks used in the design will be linked together and the hierarchy of the design is built. Before starting the elaborator, it is necessary to compile all the blocks that will be used. Then the elaborator can be invoked with the following command:

ncelab work.topmodule -access +R+W -UPDATE -timescale 1ns/10ps -work work_directory

Elaboration is done only on the top module (in this example orgatetb). In this example we should type:

ncelab work.orgatetb -access +R+W -UPDATE -timescale 1ns/10ps -work work

Running Simulations

After the design is elaborated successfully and the hierarchy of the design is made, you can run the simulations on the design. In order to run the simulation, we should make another text file that specifies the options of the simulator. Make file run.cmd and copy the following lines into it (You can also download this file from the class website).

```
database -open waves -into waves.shm -default
probe -create -shm -all -variables -depth all
run 50ns
```

These lines tell the simulator to make a database named wave.shm and save all the waveforms inside this database. The last line states that the simulation will run for 50ns. You can change these options if desired. After setting up the options file you can run the simulation by typing the following command in the terminal:

```
ncsim work.topModuleName -input run.cmd
```

In this example you should type:

```
ncsim work.orgatetb -input run.cmd
```

After running the simulation, the terminal will enter the NCSim environment you can type more NCSim commands here. In order to exit NCSim and return to terminal type:

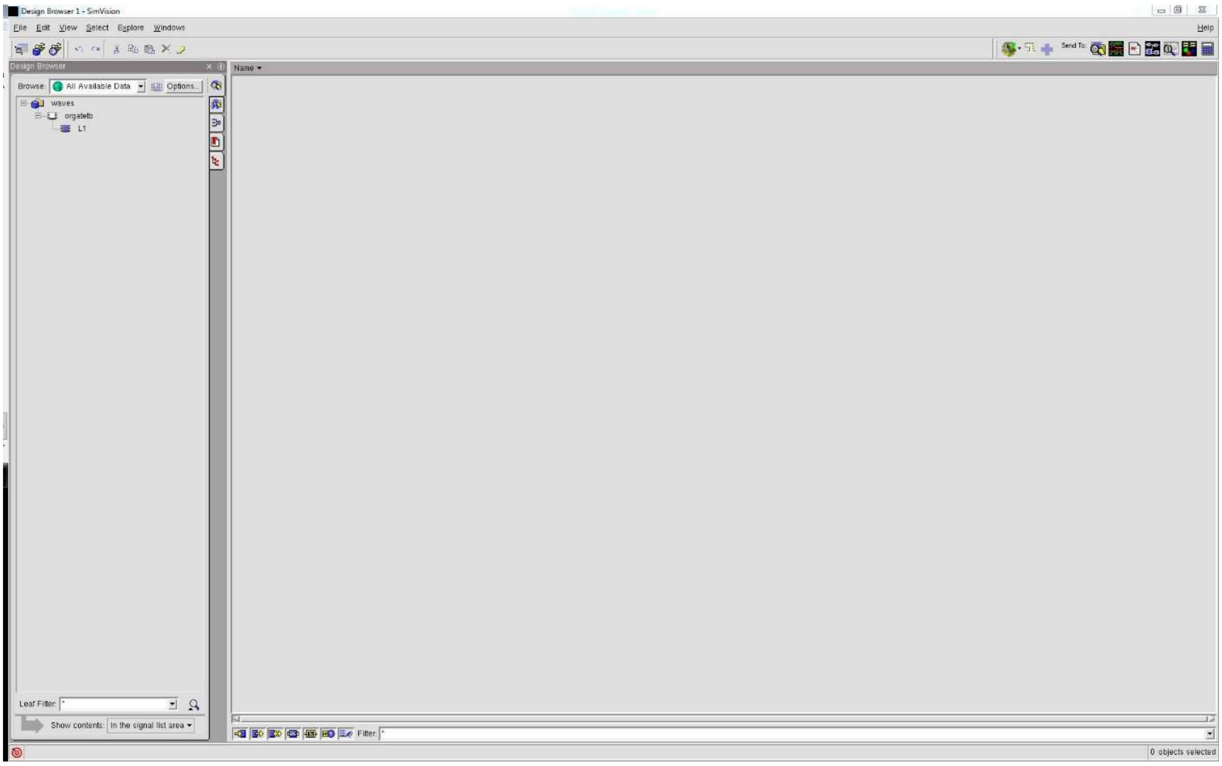
```
exit
```

Viewing Waveform

In order to view the waveform, open SimVision by typing:

```
simvision &
```

If the simulation waveform does not load automatically, in the Design Browser Window shown in Figure 1 click on *File*, *select Open Database*. In the browse window open *waves.sch* folder and select *waves.trn*. Click on (orgatetb -> L1) in the left half of the Design Browser window. The list of the signals will be shown on the right half of the window. Select the waveform you want to see (use Ctrl or Shift buttons to select multiple waveform), right-click and select *Send to Waveform Window* from the menu. A window will pop up showing the signal waveform (Figure 1). If you cannot see the whole waveform, from the *View* menu select *Zoom* and click on *Full X* option.



Browse: All Available Data

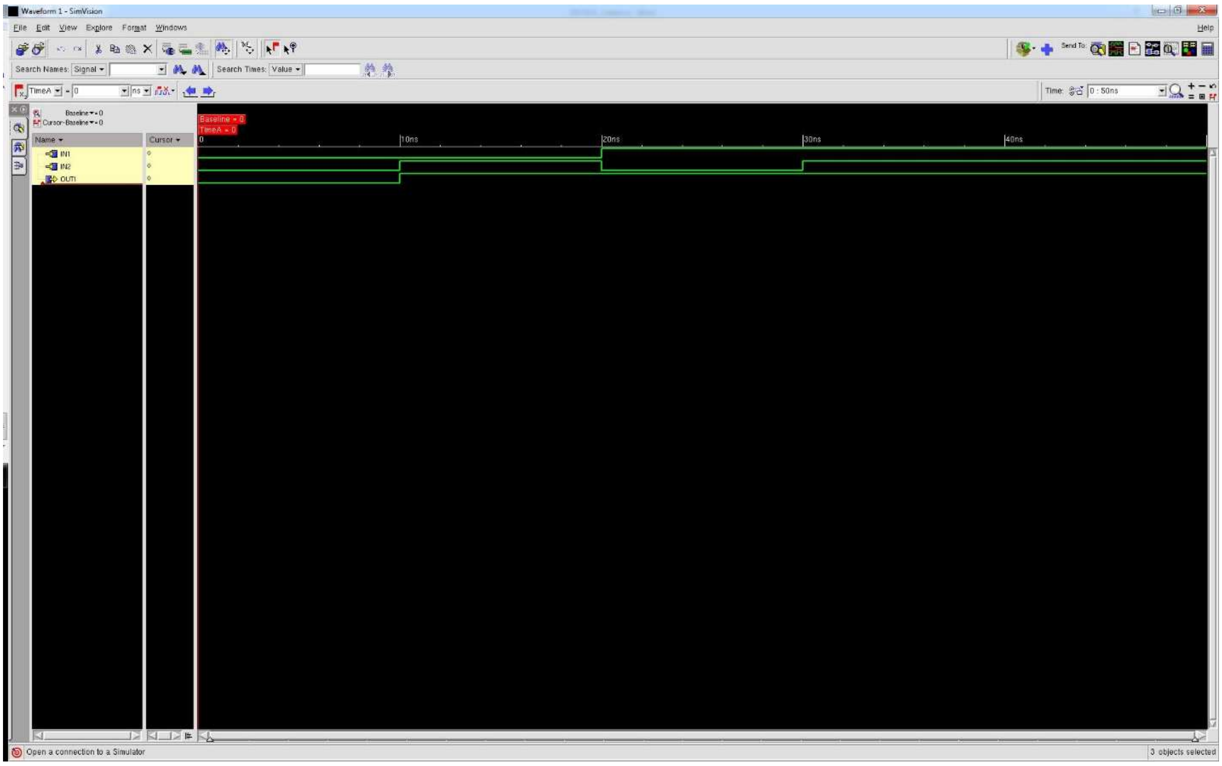
Leaf Filter: *

Show contents: In the signal list area

Filter: *

0 objects selected

Figure 1: Simvision-Design Browser Window



Name	Cursor
IN1	0
IN2	0
OUT1	0

Open a connection to a Simulator

3 objects selected

Figure 2: Simvision-Waveform Window