Computer-Aided VLSI System Design, Fall 2011

Verilog Lab 2: Waveform Debugging & Memory Generator

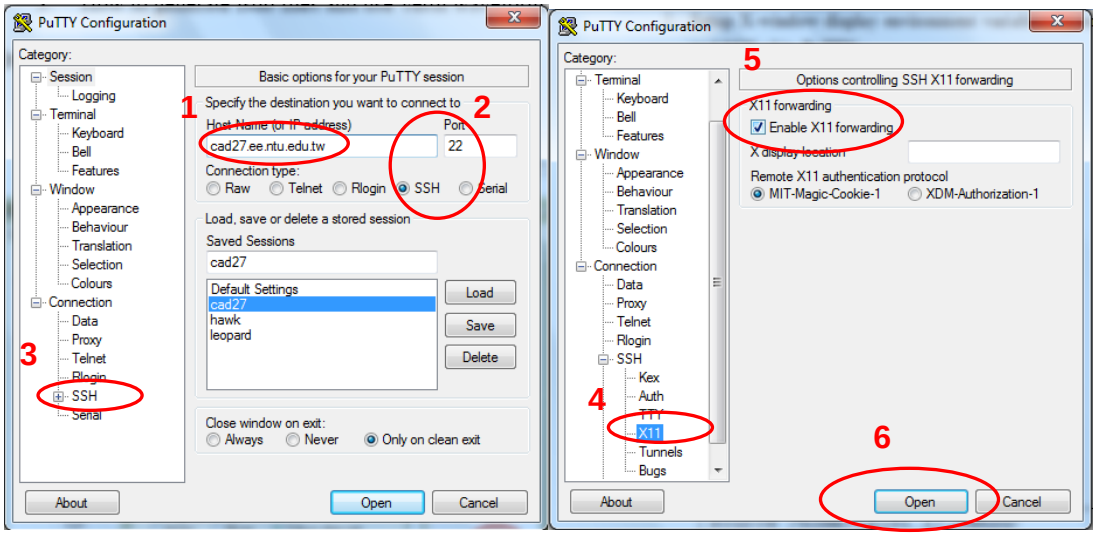
I. Objectives:

In this lab, you will learn:

1. How to dump FSDB waveform file through NC-Verilog simulator
2. Waveform debugging by using nWave
3. SDF annotation for gate-level simulation after synthesis
4. Generating memory model by using memory generator

II. Environment Setup:

1. Start X-window server (cygwin, X-win32, Xming, or Xmanager)
2. Setup X-window display environment variables by enabling X11 forwarding and SSH v2 in PuTTY



3. Connect to the workstation
4. Source the environment settings of CAD tools

```
source ~cvsd/cvsd.cshrc
source ~cvsd/verdi.cshrc
```

If you try entering the command "`ncverilog`" or "`nWave`" but it turns out "command not found," it means there's something wrong with the "`*.cshrc`" file or the software license is out of date.

III. Lab File Acquisition:

1. Copy the lab file from this directory

```
cp ~cvsd/11F/Verilog/Lab2.tar .
```

2. Un-tar the lab file

```
tar -xvf Lab2.tar
```

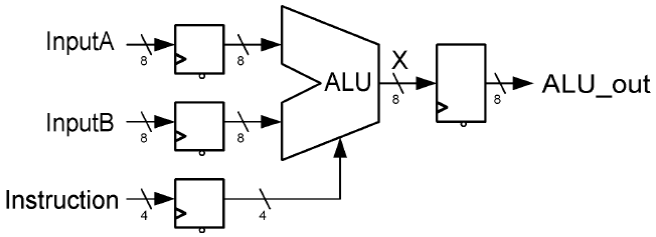
3. Enter the lab directory

```
cd Lab2/
```

4. There are supposed to be some files as follows. Please check it!

Filename	*Description*
`Lab2_alu.v`	RTL code of 8-bit ALU
`Lab2_test_alu.v`	Testbench for 8-bit ALU
`Lab2_alu_run.f`	File list for RTL simulation
`Lab2_alu_s.v`	Gate-level code of 8-bit ALU
`Lab2_alu_run_s.f`	File list for gate-level simulation
`Lab2_alu_s.sdf`	`sdf` file for gate-level timing annotation
`Lab2_test_ram.v`	Testbench for generated RAM
`tsmc13.v`	Verilog model of TSMC .13 um CMOS technology. ***Please DO NOT download this file, for it is property of TSMC!***

5. The circuit diagram of the ALU used for this lab is illustrated as follows.



IV. Lab 2.1: Generating the FSDB Waveform

1. Run the RTL simulation by using `ncverilog`

```
ncverilog Lab2_test_alu.v Lab2_alu.v
```

or

```
ncverilog –f Lab2_alu_run.f
```

2. You would only see the text telling you “congratulations” but not knowing what’s going on inside the circuit. Please open the testbench.

```
joe Lab2_test_alu.v
```

Find line 17 & 18 of the file, you should note that something are commented out. Please un-comment these two lines.

```
$fsdbDumpfile("Lab2_alu.fsdb");
$fsdbDumpvars;
```

After editing, press “Ctrl+k” to turn to command mode, then type “x” to save and quit.

3. Now re-run the simulation, note we should add “`+access+r`” to enable FSDB file dumping.

```
ncverilog +access+r –f Lab2_alu_run.f
```

4. After simulation, it shows something like

```
*Novas* Create FSDB file 'Lab2_alu.fsdb'
*Novas* Begin traversing the top scope(test_alu), layer(0).
*Novas* End of traversing the top scope(test_alu)
```

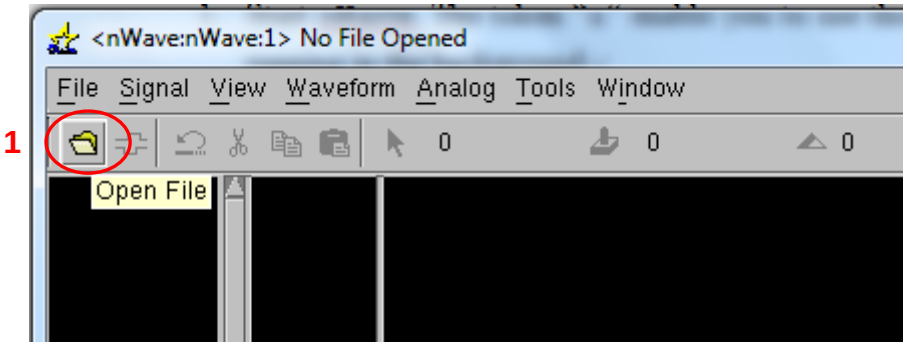
Please check if there exist a file called "`Lab2_alu.fsdb`".

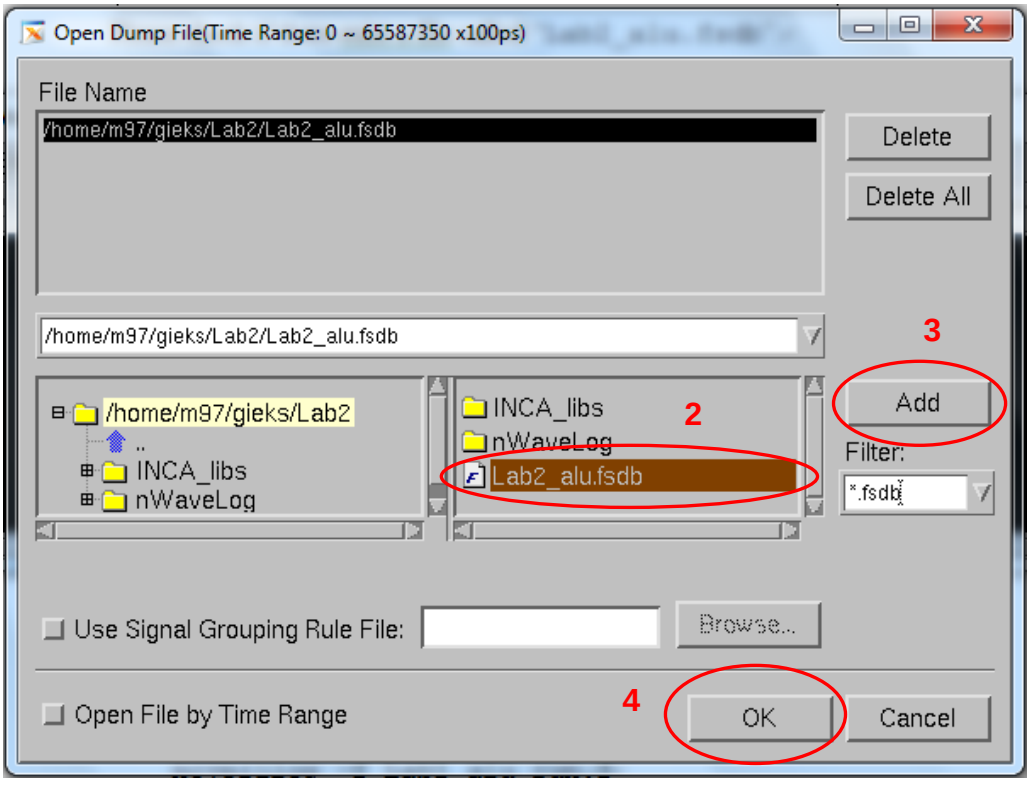
V. Lab 2.2: Using nWave for Waveform Debugging

1. Start `nWave`. The token "&" enable you to use the terminal while `nWave` is running in the background.

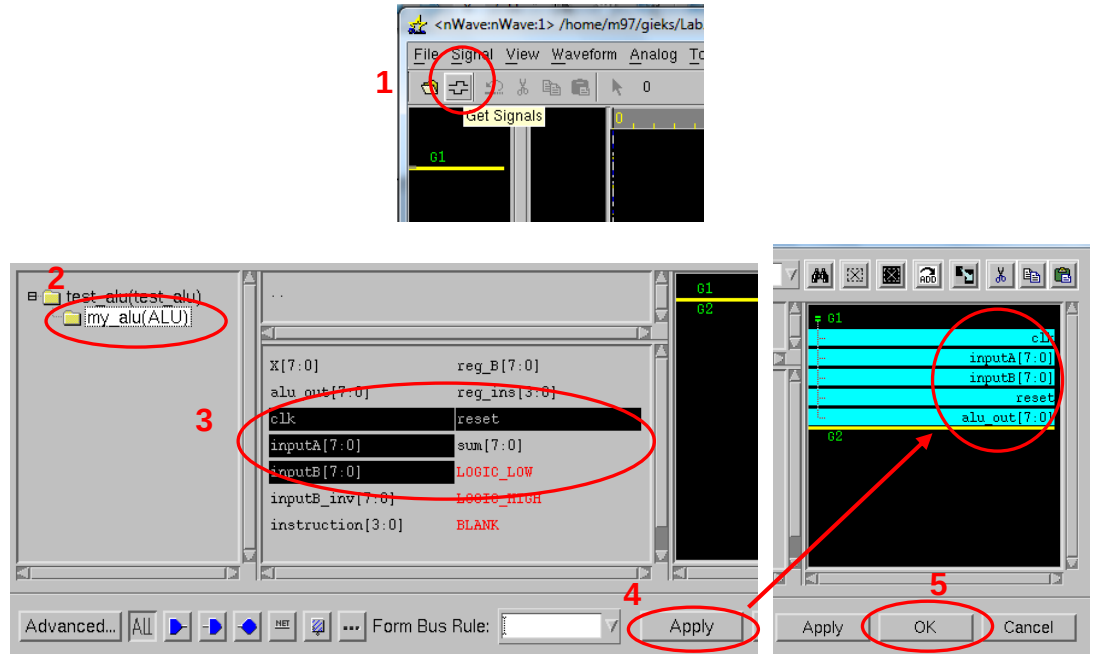
```
nWave &
```

2. Open the FSDB file we obtained.

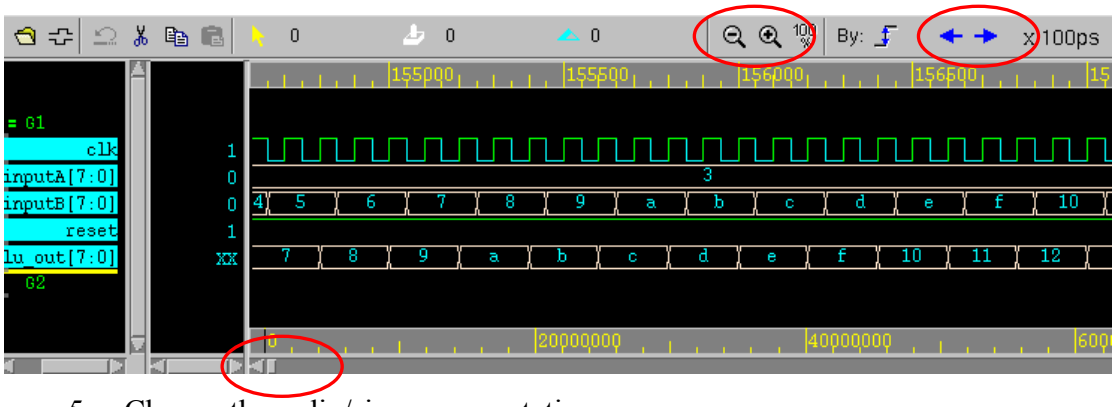




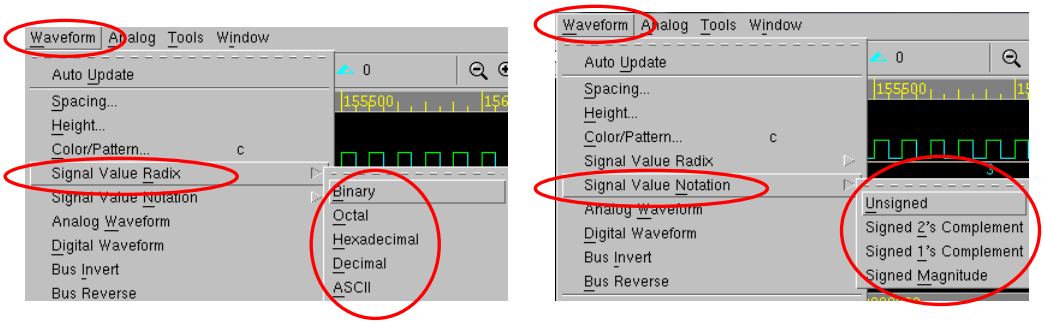
3. Choose signals we are interested in.



4. Browse the waveform.



5. Change the radix/sign representation



VI. Lab 2.3: Gate-Level Simulation

1. After synthesizing `Lab2_alu.v`, we can derive the gate-level netlist file `Lab2_alu_s.v`. Try this command for gate-level simulation:

```
ncverilog +access+r Lab2_test_alu.v Lab2_alu_s.v tsmc13.v
```

or

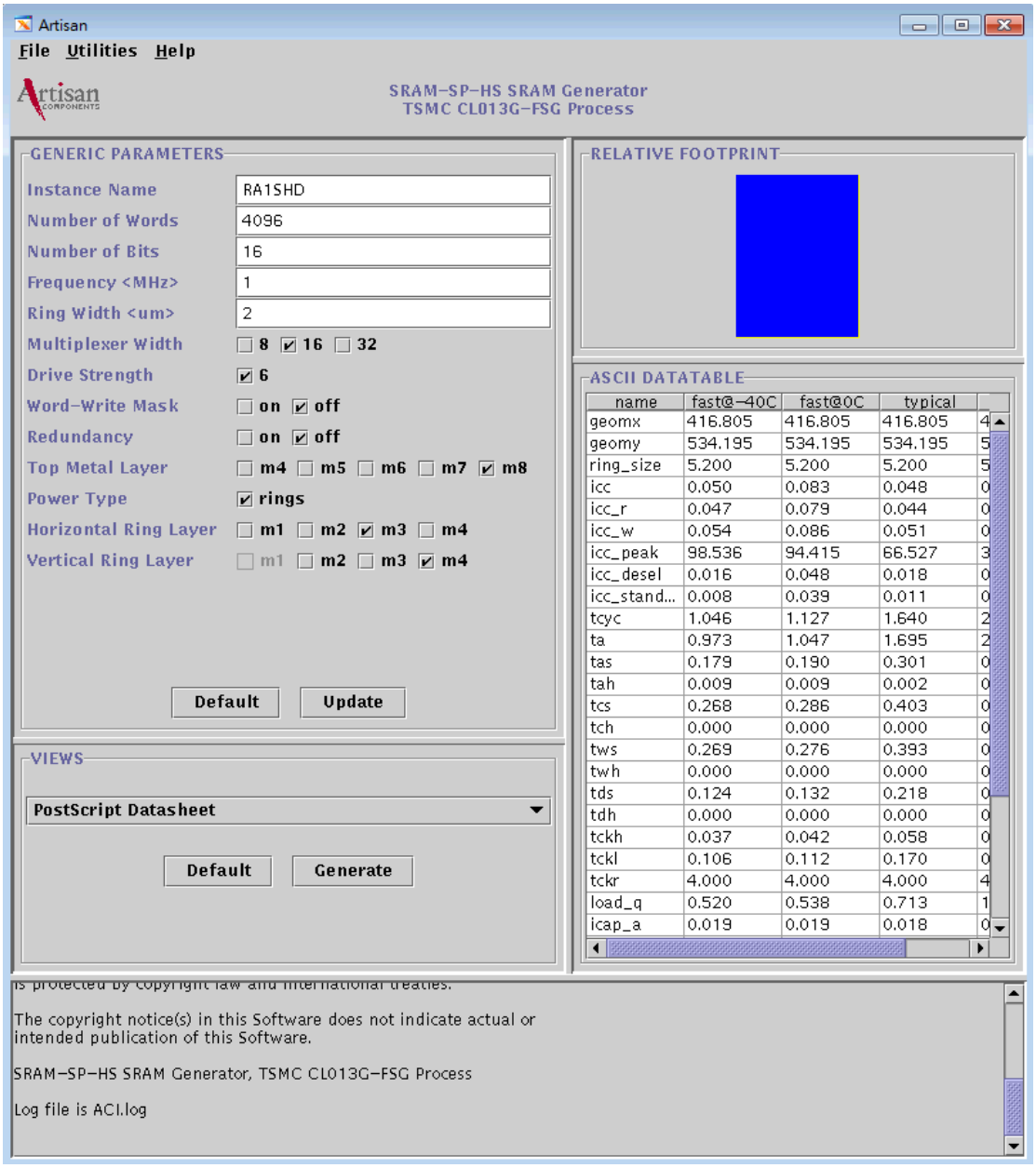
```
ncverilog +access+r –f Lab2_alu_run_s.f
```

2. You will see a lot of warnings about "timing violation." That's because the simulator does not know about the propagation delay of each gate. So the simulator cannot combine the delay information with the "`tsmc13.v`" to check if your design can run at the clock frequency defined in the testbench. Please open the testbench again and un-comment the line 16:
```
$sdf_annotate("Lab2_alu_s.sdf",my_alu);
```
3. Save the modified testbench and re-run step 1.
4. During the simulation, there should be lots of "traversing the top scope" information, and should never prompt out any "timing violation."
5. Open the waveform file. See how different it is between RTL waveform and gate-level waveform.

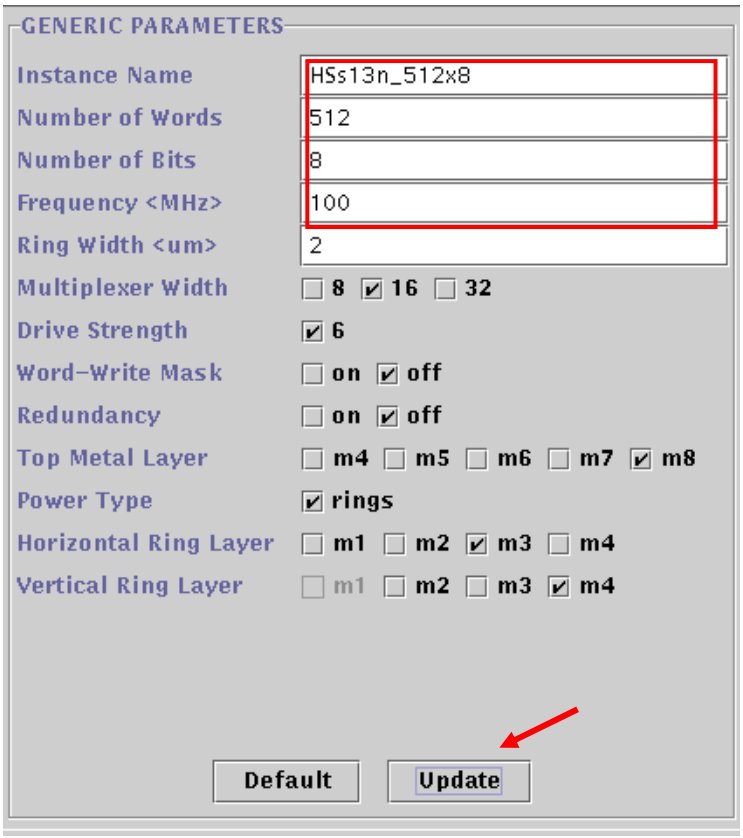
VII. Lab 2.4: Memory Generator

1. *Design your memory name and specification*.
   The name of your memory should be conformed to some kind of rule if you want to maintain them easily in the future. Here we need a high speed, single port memory with 512 words, each word is 8bit. Therefore we name our memory as `HSs13n_512x8`, which means High Speed single port 0.13μm normal RAM with 64words/8bit. Remember the name "`HSs13n_512x8`", it's the name of your memory.
2. *Run the memory compiler*.
   There are three kinds of memory compiler in TSMC 0.13μm cell library: `ra1sh`, `ra1shd`, and `ra2sh`. "`ra1sh`" is used for high-speed single-port SRAM, "`ra1shd`" is used for larger size SRAM, and "`ra2sh`" is for high-speed dual port SRAM designs. "`ra1shd`" is used in this lab, you can type following command to start:
   ```
   ~cvsd/CBDK_IC_Contest/CIC/Memory/ra1shd/bin/ra1shd &
   ```
   (Note 1: it is a **single-line** command!)
   (Note 2: when typing a long path, try TAB key!)

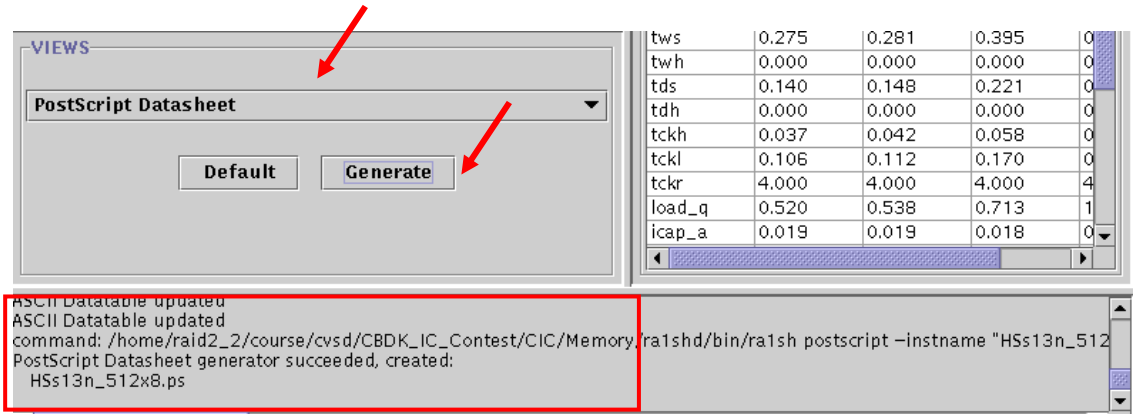
3. And then you should see this!

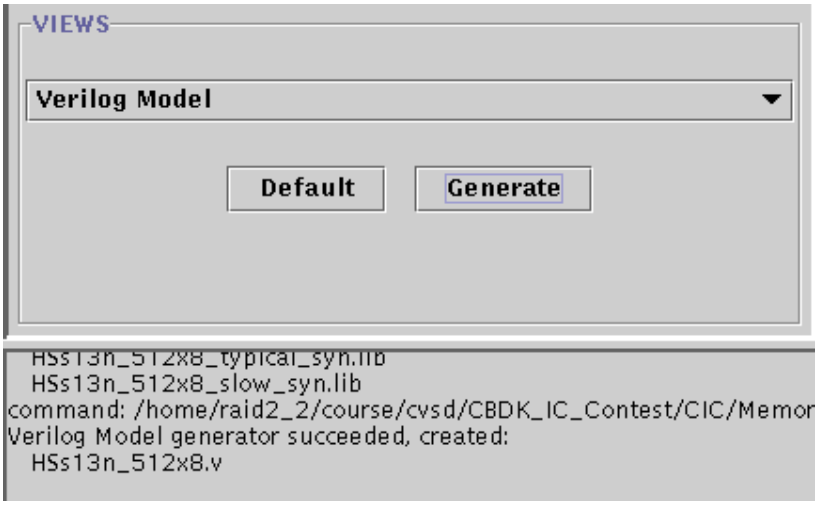


4. Click in the specification

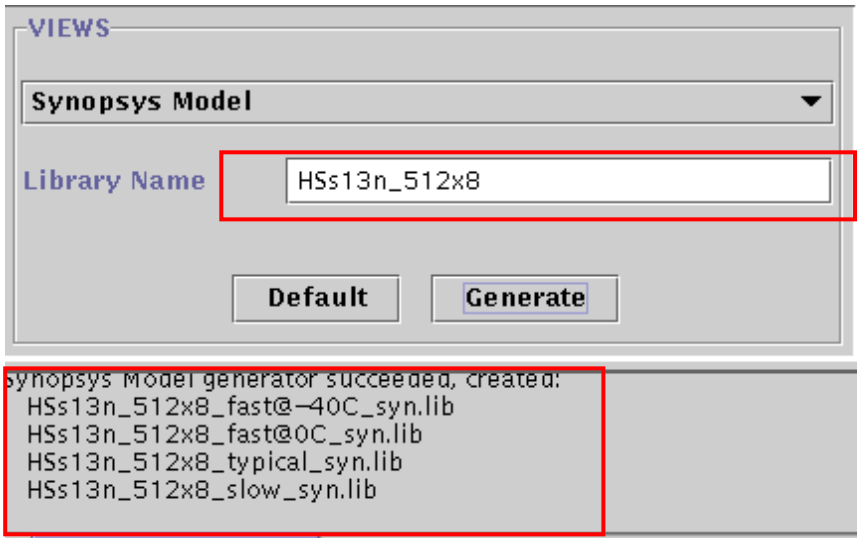


5. *Generate the Data Sheet and Verilog Behavior Model*.
Click on the List box and choose "**`PostScript Datasheet`**", then click the "**`Generate`**" button. You'll get a HSs13n_512x8.ps which is the data sheet of the SRAM. And then click on the list box and choose "**`Verilog Model`**", and click the "**`Generate`**" button. You'll get a file named HSs13n_512x8.v.





6. *Generate the library for using in Synthesis.*
   Choose "**`Synopsys Model`**" in the list box and type in your library name. Here we type "**`HSs13n_512x8`**" into it.



   Click on "Generate" then you can generate three ".lib" for use in Synopsys Synthesis tool. The three ".lib" files are HSs13n_512x8_fast@-40C_syn.lib, HSs13n_512x8_fast@0C_syn.lib, HSs13n_512x8_typical_syn.lib, HSs13n_512x8_slow_syn.lib.

7. After all, Click on the "Utilities -> Write Spec" to write down the spec of your SRAM. CIC will model your RAM according to the specification file you wrote.

8. Read timing diagram of synchronous RAM.
   A. Download the HSs13n_512x8.ps and use Acrobat to read it.
   B. Find out the function and timing diagram for every port of *HSs13n_512x8* module in *HSs13n_512x8.v* according to the data sheet. How many words can be stored? How many bits for a word?

9. Run Simulation of RAM
   A. Run simulator:

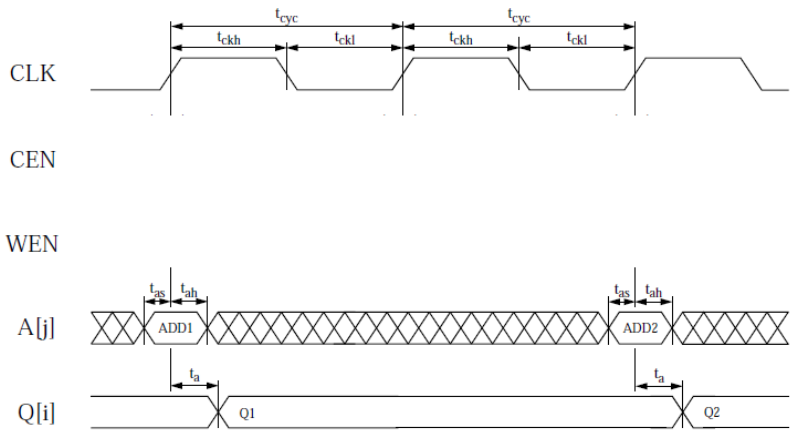
```
ncverilog +access+r Lab2_test_ram.v HSs13n_512x8.v
```

   B. Open the waveform *Lab2_ram.fsdb* via nWave. Compare the waveform of *HSs13n512x8* module with the timing diagram of data sheet. Furthermore, you can find that the output port *O[7:0]* has some timing delay because the timing information of *HSs13n_512x8* module is described in *HSs13n_512x8.v*.
   C. Please examine how *Lab2_test_ram.v* to control this synchronous RAM.

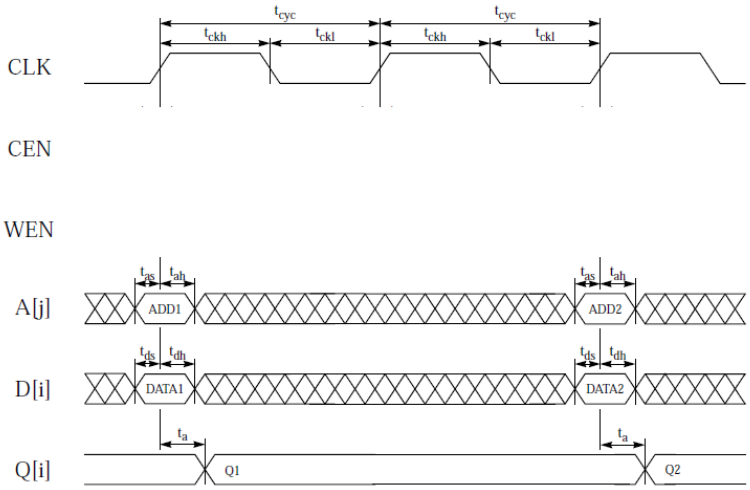
VIII. Checkpoint:

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Please draw the **read-cycle** timing diagram of synchronous single-port SRAM.



2. Please draw the **write-cycle** timing diagram of synchronous single-port SRAM.



Pin Description

Pin	Description
A[8:0]	Addresses (A[0] = LSB)
D[7:0]	Data Inputs (D[0] = LSB)
CLK	Clock Input
CEN	Chip Enable
WEN	Write Enable
Q[7:0]	Data Outputs (Q[0] = LSB)

IX. END of LAB

Creator:
1st Edition: Chao-Tsung Huang, 2002
2nd Edition: Yu-Lin Chang, 2004
3rd Edition: Yu-Lin Chang, 2006
4th Edition: Jui-Hsin Lai (Larry), 2008
5th Edition: Min-An Chao, 2009
6th Edition: En-Jui Chang, 2010
7th Edition: En-Jui Chang, 2011