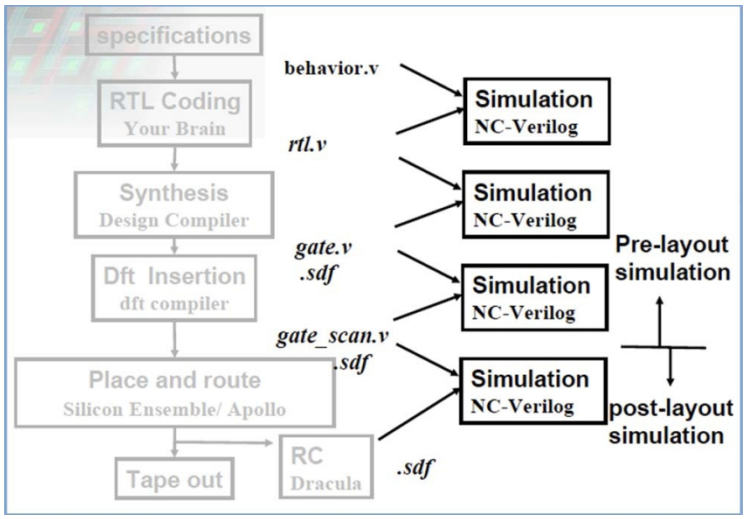
Verilog Lab 2 – **Waveform Debugging & Memory Generator**

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. Lab Files:

1. Copy the lab file form this directory

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

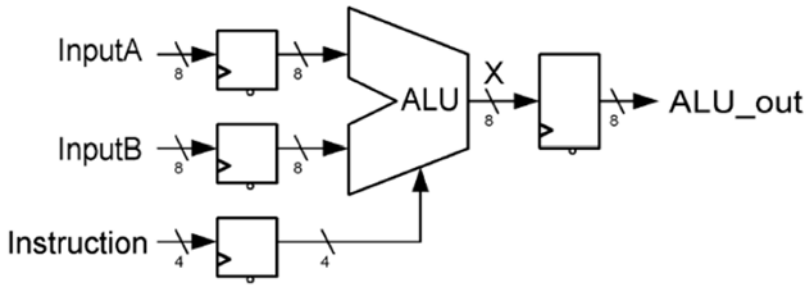
2. Type this command to extract your files.

   tar -xvf Lab2.tar

3. 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!***

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



5. Type this command to enter Lab2 directory:

   cd Lab2

III. Environment Setup:

1. Source the default ***cshrc*** file:

   source /usr/cadence/cshrc

   source /usr/spring_soft/CIC/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it turns out "command not found," it means there's something wrong with the "*.cshrc" file or the software license is out of date.

IV. Lab 2.1: Generating the VCD & FSDB Waveform

Value Change Dump (VCD) format**:**

- Indigenously supported by most simulators
- Using ASCII text for waveform recording — Extremely huge file size
- *$dumpfile("filename.vcd");*
  $dumpvars;

Fast Signal Database (FSDB) format:

- Defined by SpringSoft Verdi debugging system
- More compact format, small file size
- *$fsdbDumpfile("filename.fsdb");*
  $fsdbDumpvars;

for multi-dimention array
$fsdbDumpvars(0,test_module_name, "+mda");

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. Now we would like to generate VCD file. Please open the testbench.

gedit Lab2_test_alu.v

Please add following content in testbench to generate vcd file.

```
//$sdf_annotate(“Lab2_alu_s.sdf”,my_alu);
//$fsdbDumpfile(“Lab2_alu.fsdb”);
//$fsdbDumpvars;
$dumpfile("Lab2_alu.vcd");
$dumpvars;
```

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

   ncverilog +access+r –f Lab2_alu_run.f

4. Please check if there exists a file called “Lab2_alu.vcd”

5. Now we would like to generate FSDB file. Please open the testbench again.

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

```
//$sdf_annotate(“Lab2_alu_s.sdf”,my_alu);
$fsdbDumpfile(“Lab2_alu.fsdb”);
$fsdbDumpvars;
//$dumpfile("Lab2_alu.vcd");
//$dumpvars;
```

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

   ncverilog +access+r –f Lab2_alu_run.f

8. 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)
```

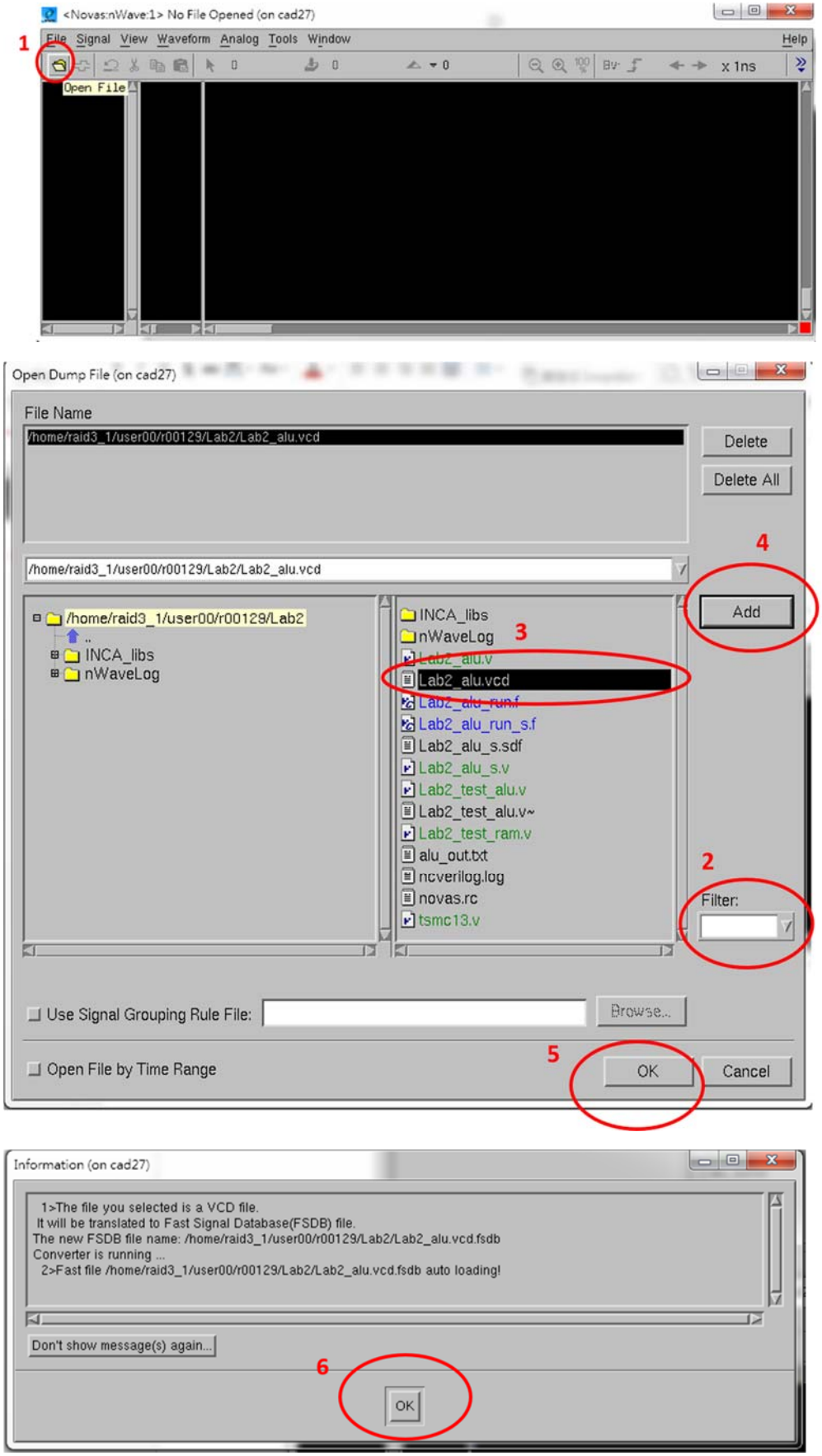
Please check if there exists a file called “Lab2_alu.fsdb”.

V. Lab 2.2: Using nWave for Waveform Debugging

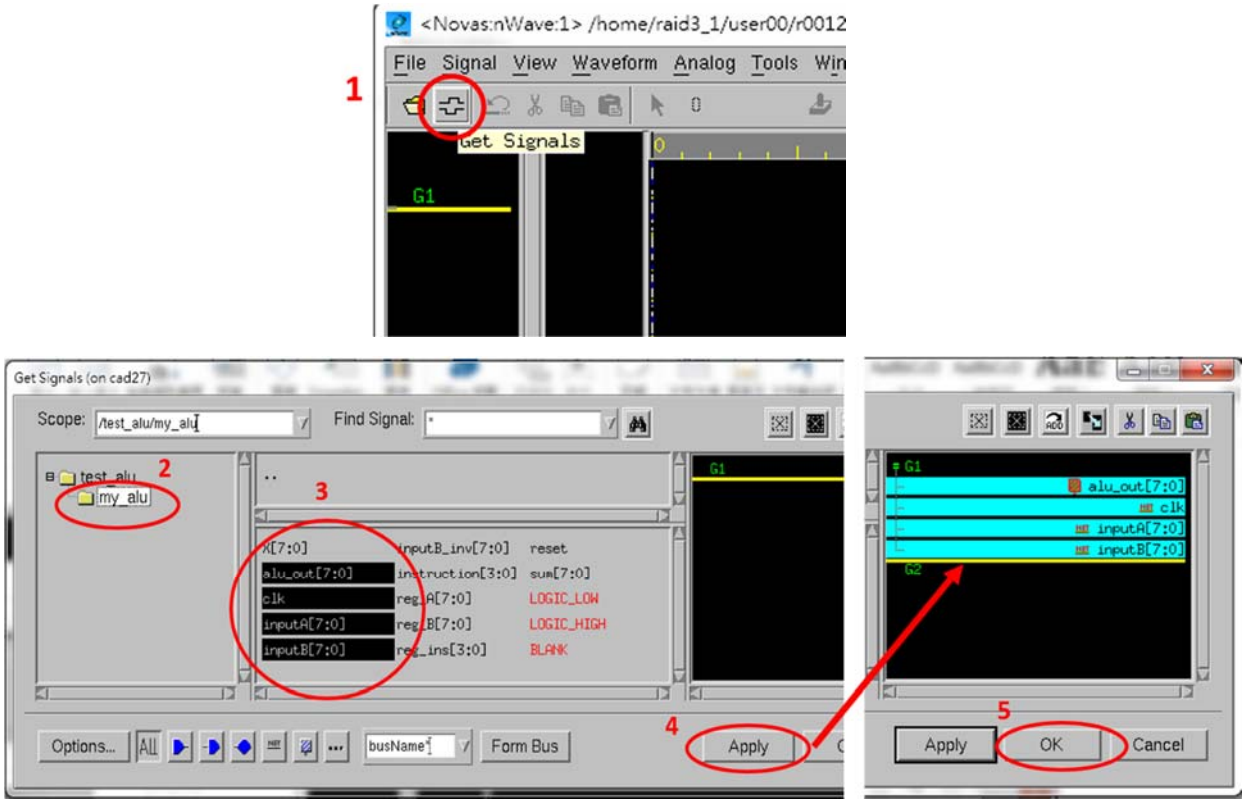
1. Start nWave. The token “&” enable you to use the terminal while nWave is running in the background.

   nWave &

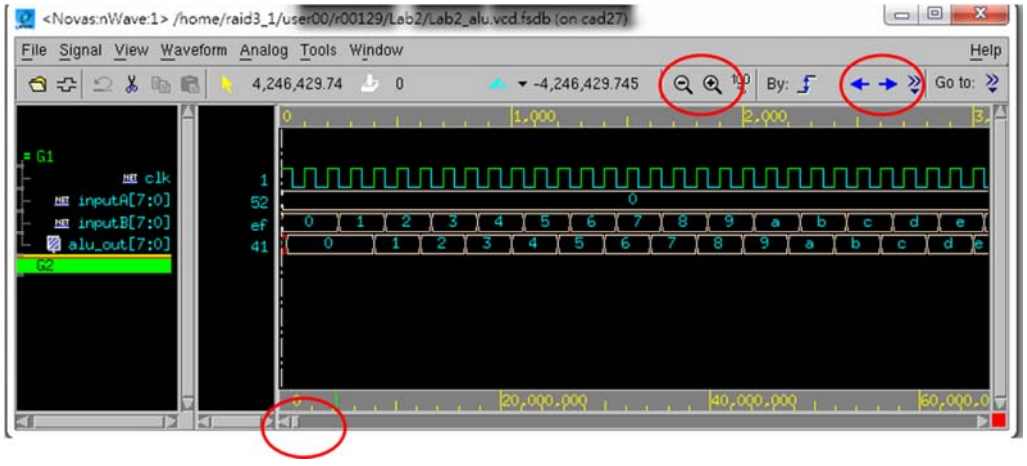
2. Open the VCD file we obtained. VCD file will be translated to FSDB file.



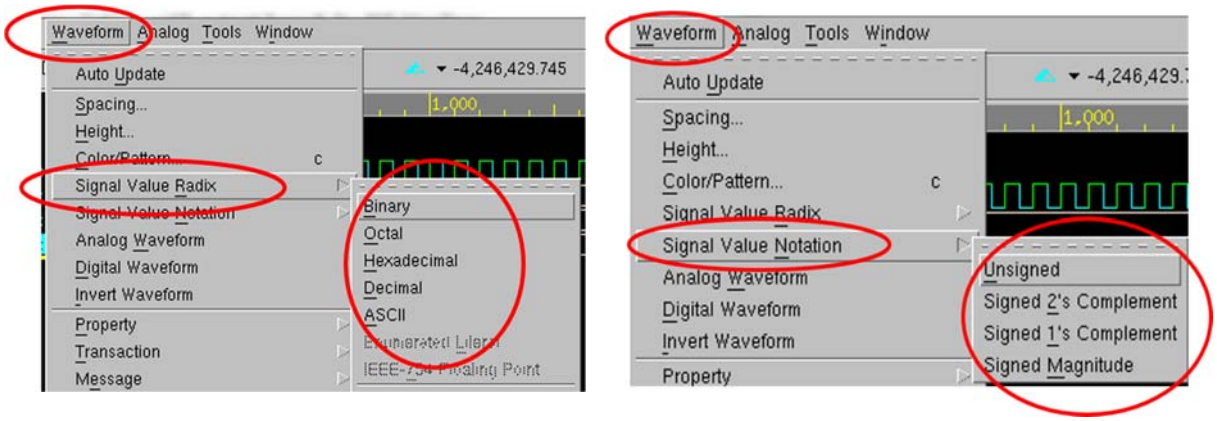
3. Choose signals we are interested in.



4. Browse the waveform.



5. Change the radix/sign representation.



注意!!:
1. sdf檔案名稱與位置是否與tb內所寫吻合
2. 相對應的module name是否與tb中的吻合

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.

   gedit Lab2_test_alu.v

   Please un-comment the line 16 and rename the vcd file:

```
$sdf_annotate("Lab2_alu_s.sdf",my_alu);
//$fsdbDumpfile("Lab2_alu.fsdb");
//$fsdbDumpvars;
$dumpfile("Lab2_alu_s.vcd");
$dumpvars;
```

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 is 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.

-Memory Block或著是IO pad都是在大型系統設計中，額外的IP block，最後lay時不論是CIC還是full custom based的電路到時候就會擺上填補缺口。
-但是像是tb timing模擬資訊，又或著是對於要設計的IP走IC design flow的過程中都需要這些外加IP的接口資訊才能夠夠真實
-最後是timing diagram可以幫助在設計IP時的protocol要怎麼訂Handshake Signal才可以讓之後module間資料交換有效，不會有timing issue such as violation or 差一兩個cycle

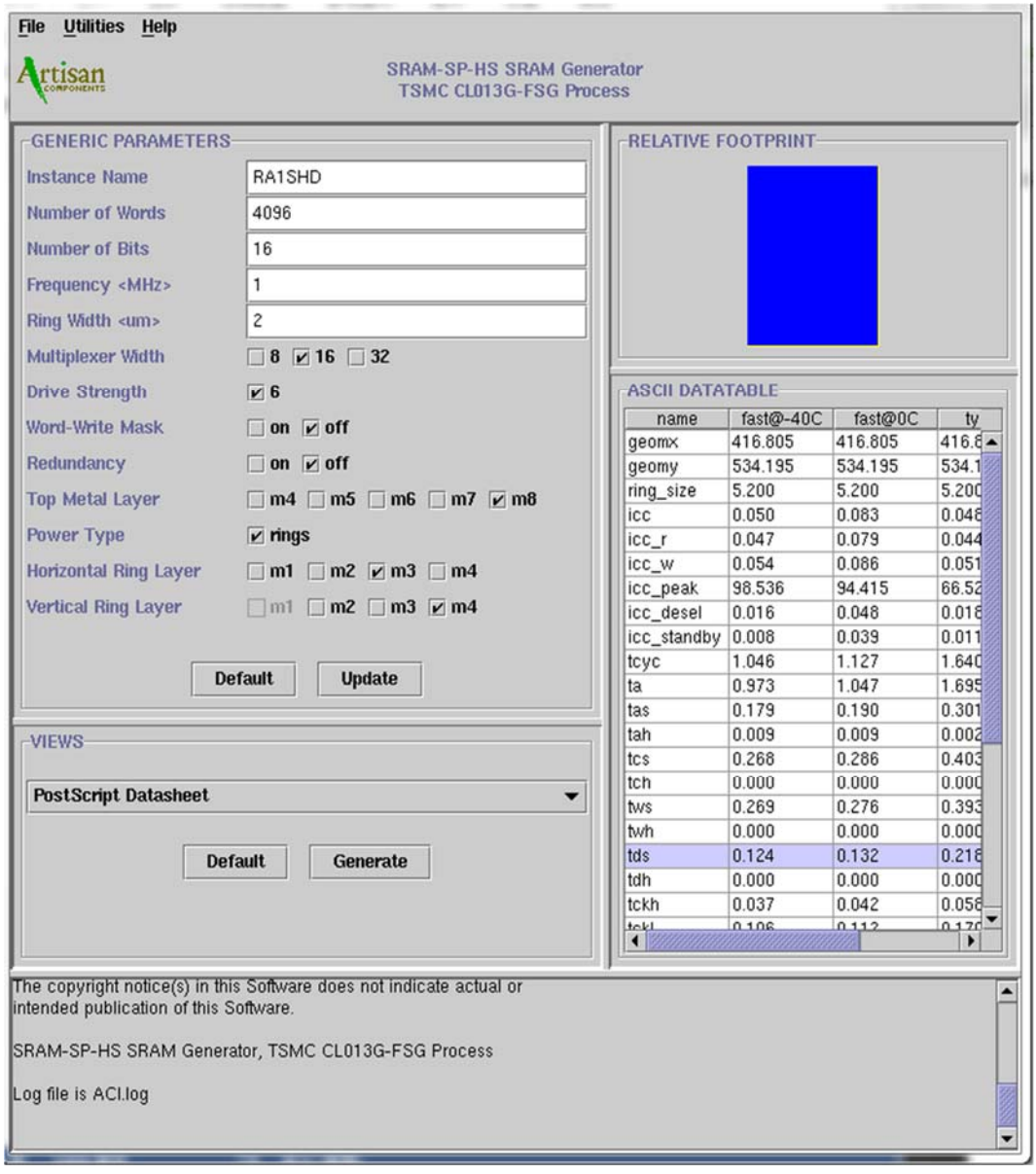
"ra1shd" is used for larger size SRAM.
"ra2sh" is used for high-speed dual port SRAM designs.

"ra1shd" is used in this lab, you can type following command to start:

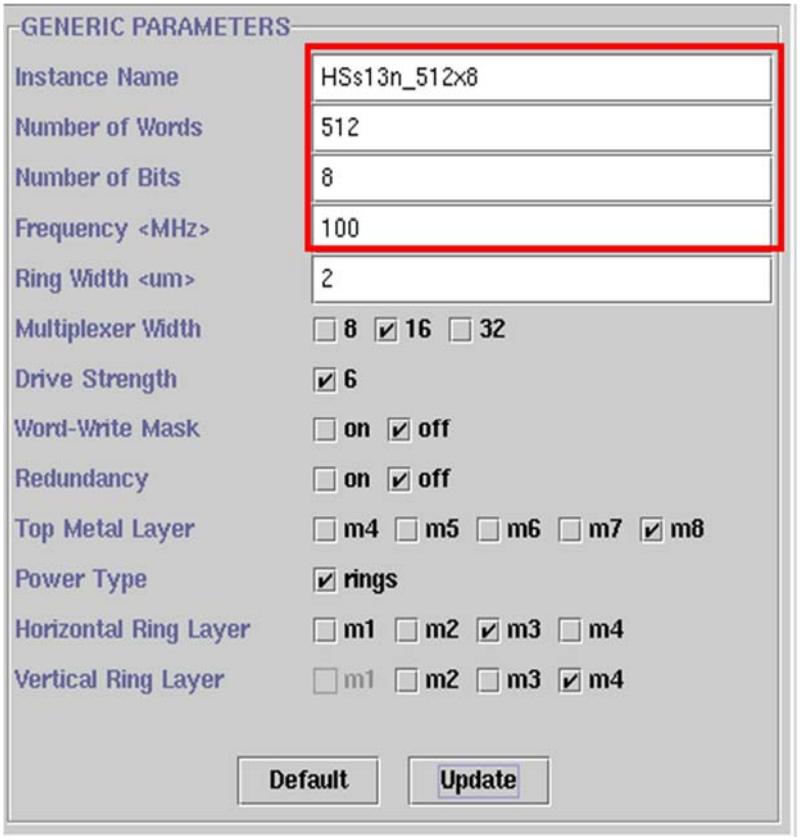
~cvsd/CBDK_IC_Contest/CIC/Memory/ra1shd/bin/ra1shd &

(Note 1: it is a single-line command!)
(Note 2: when typing a long path, try TAB key!)
(Note 3: The memory generator of TSMC .13 technology is only executable on workstations with Solaris series OS. The information of workstation is: http://cad.ee.ntu.edu.tw/ws_list.htm.)

3. And then you should see this!



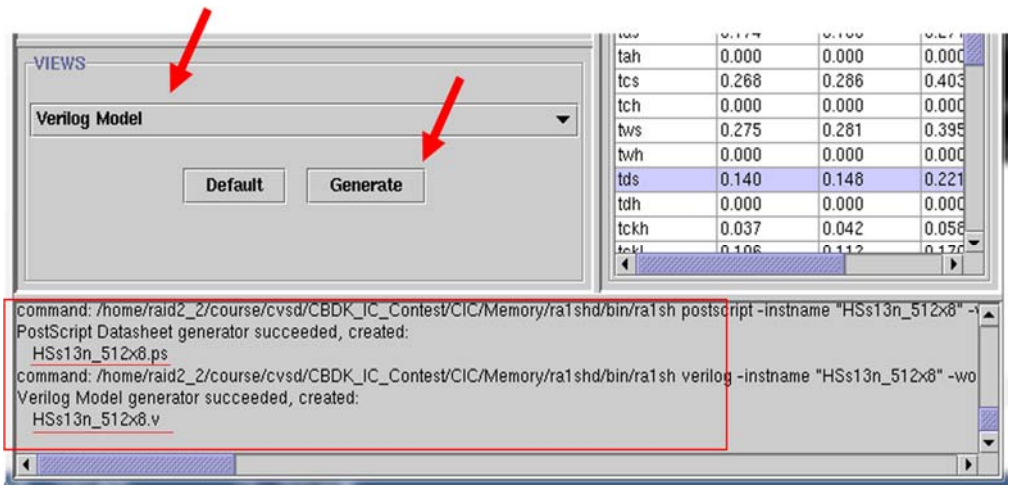
4. Click in the specification



這只是行為描述檔，描述模擬器在模擬的時候timing delay，到時候跟其他所設計的模組組合起來可以做整個系統的行為模擬

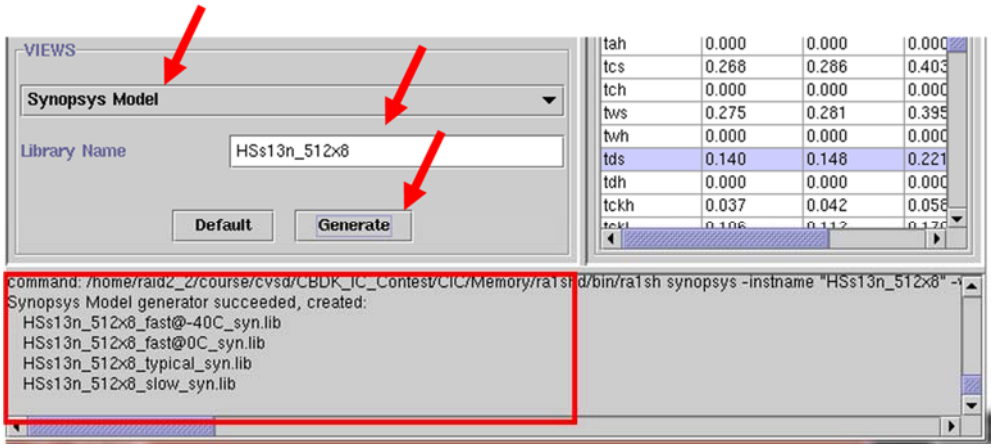
5. *Generate the Data Sheet and Verilog Behavior Model.*

Click on the List box and choose "PostScript Datasheet", then click the "Generate" button. You'll get 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.



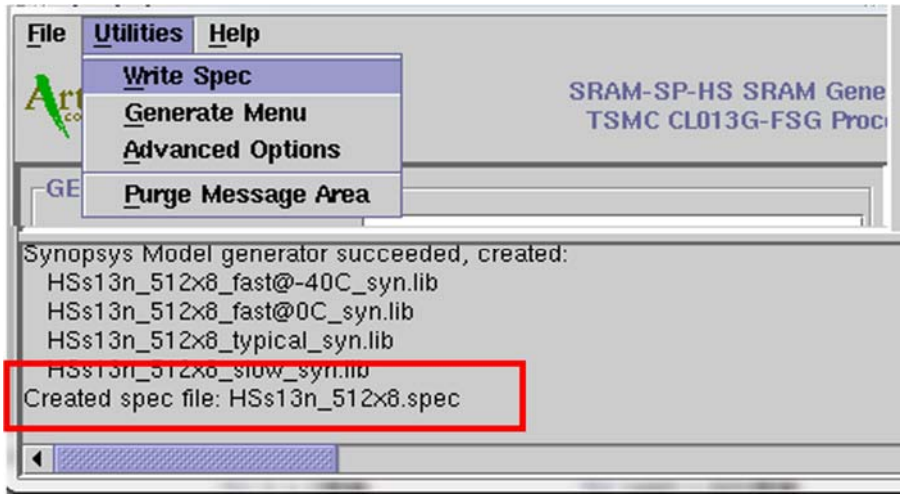
lib檔對於tool在做合成環境設定提供資訊
則tool根據constraint sdc檔案設定所做static等分析才能跟實際tb模擬以及實際下線相吻合
EX: node timing, loading
dv: 在setup檔中提供相關路徑並將檔案放置其中

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 four ".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.
(Note: Please following above step to generate VCD first then translated to FSDB.)

C. Please examine how Lab2_test_ram.v to control the synchronous RAM.

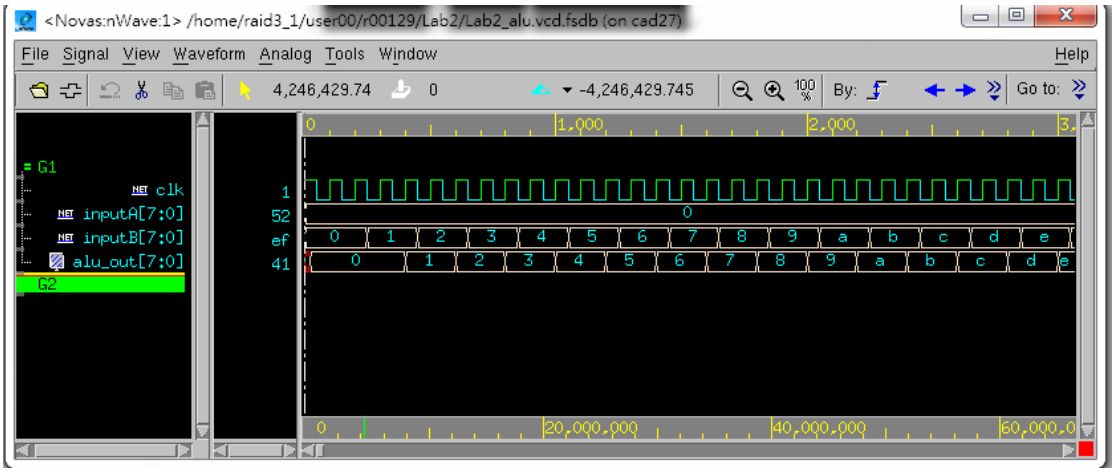
Pin Description

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)

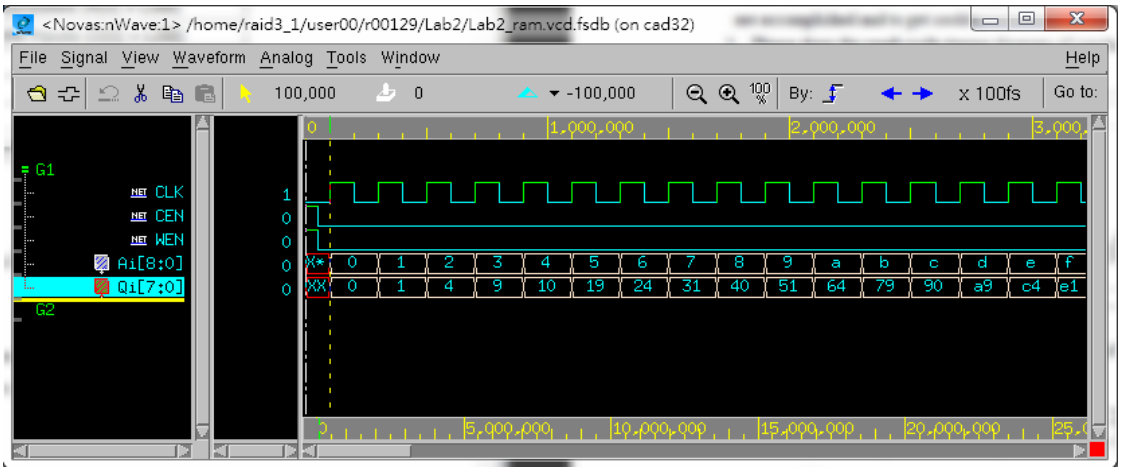
VIII. Checkpoints:

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

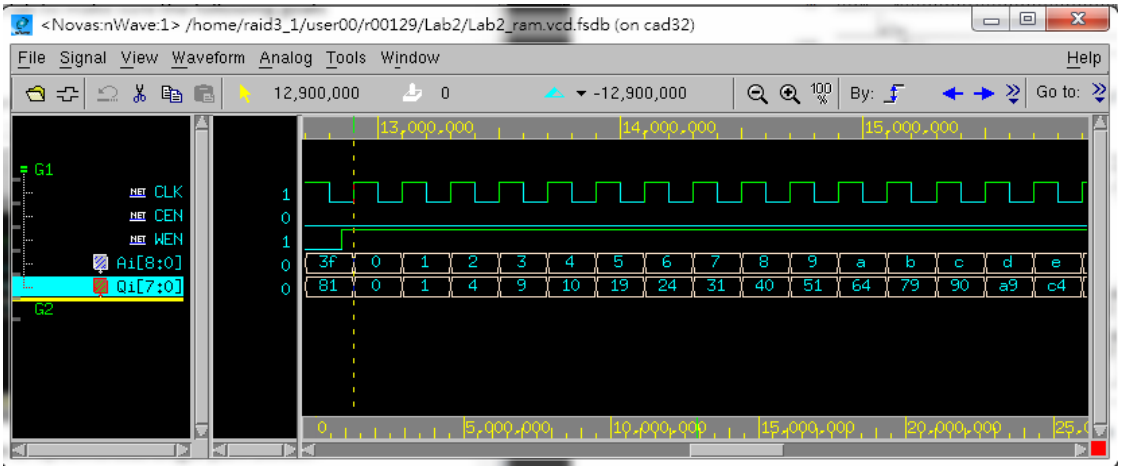
1. Please picture the waveform of ALU.



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



3. Picture the **read-cycle** timing diagram of synchronous single-port SRAM.



END of LAB

1st Edition: Chao-Tsung Huang, 2002

2nd Edition: Yu-Lin Chang, 2004

3rd Edition: Yu-Lin Chang, 2005

4th Edition: En-Jui Chang, 2010

5th Edition: En-Jui Chang, 2011

6th Edition: Yu-Min Lin, 2013

7th Edition: Ching-Yao Chou,2015