Computer-Aided VLSI System Design

Synthesis Lab 1-1: Design Vision / Design Compiler
Synthesis Lab 1-2: Synopsys Memory Library Generation

Objectives:
In this lab, you will learn:
1. Basic concept about synthesis
2. How to use Synopsys Design Vision (GUI) / Design Compiler (text mode)
3. How to use Synopsys Design Compiler to generate the memory library file for Synopsys Design Compiler

Environment Setup:
1. Make sure you can run design compiler:

```
source /usr/cad/synopsys/CIC/synthesis.cshrc
```

```
dc_shell -h
```

There will be something wrong in Solaries and CentOS, please chose workstation with **RHEL 4.**

2. Make a working directory:

```
mkdir synthesis_lab
cd synthesis_lab
```

3. Copy synopsys_dc.setup to .synopsys_dc.setup into your working directory:

```
cp ~cvsd/CUR/Synthesis/Lab1/synopsys_dc.setup ./.synopsys_dc.setup
```

Note: If you already have a *.synopsys_dc.setup* file and you do not want to overwrite it, you can add only the Library and tools related lines that you miss into your own *.synopsys_dc.setup* file.

Copy Files from CVSD TA Directory
1. Copy all the files into your work directory,

```
cp –r ~cvsd/CUR/Synthesis/Lab1/* .
```

2. Check if you have these files

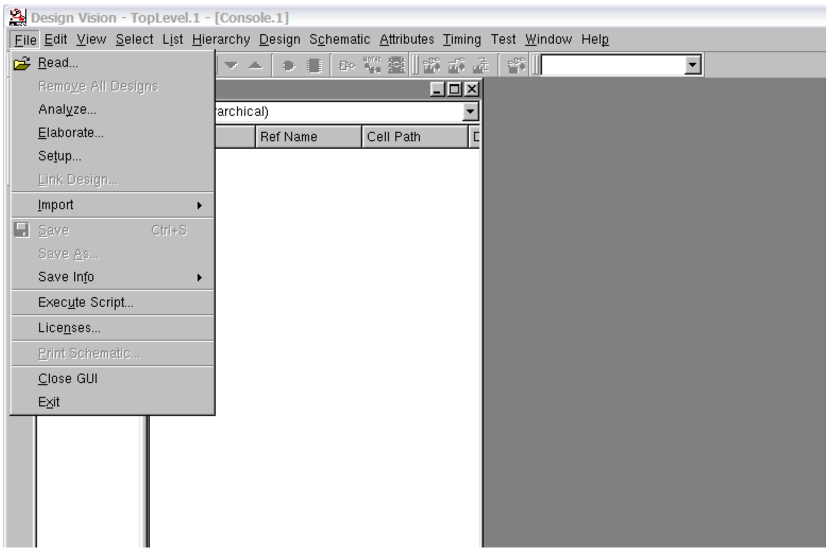
Filename	**Description**
.synopsys_dc.setup	Design compiler setting
Lab1_alu.v	The source ALU file
HSs13n_512x8_fast_syn.lib	The memory library under fast condition
HSs13n_512x8_slow_syn.lib	The memory library under slow condition
HSs13n_512x8_typical_syn.lib	The memory library under typical condition

Lab1-1: Synopsys Design Vision

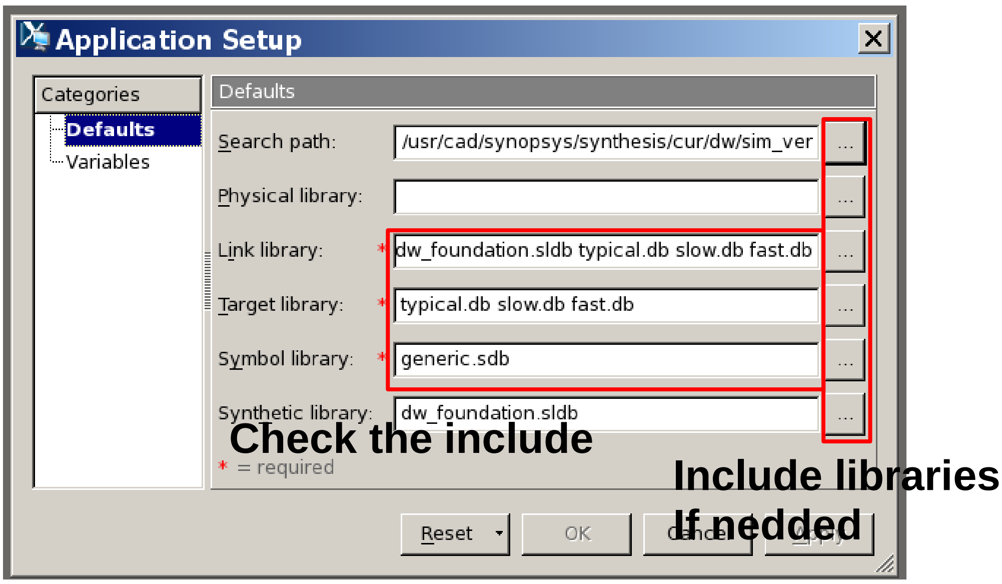
Read files

1. The function of *Lab1_alu.v* is from your previous Lab. We will try to use this sample and practice Synopsys synthesis tool step by step. We will show the GUI example with figures and text-mode with the strings in the text blocks.

2. Build your working directory by yourself, copy all files and start up design_vision at working directory by using the **dv &** command. Also , you could invoke text mode by using **dc_shell** command or **Command line** at the bottom of the window.

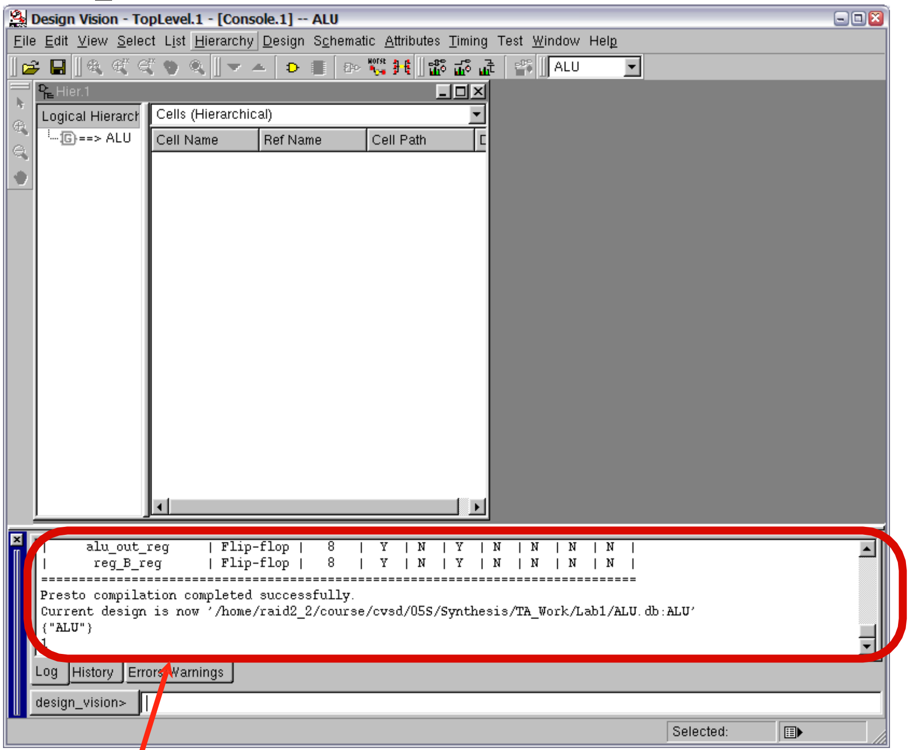


3. Check the GUI environment setting “File -> Setup”. If the “Line library” or “Target library” or “Symbol library” is empty, you have to check the .synopsys_dc.setup or use the GUI to choose the library files.



4.

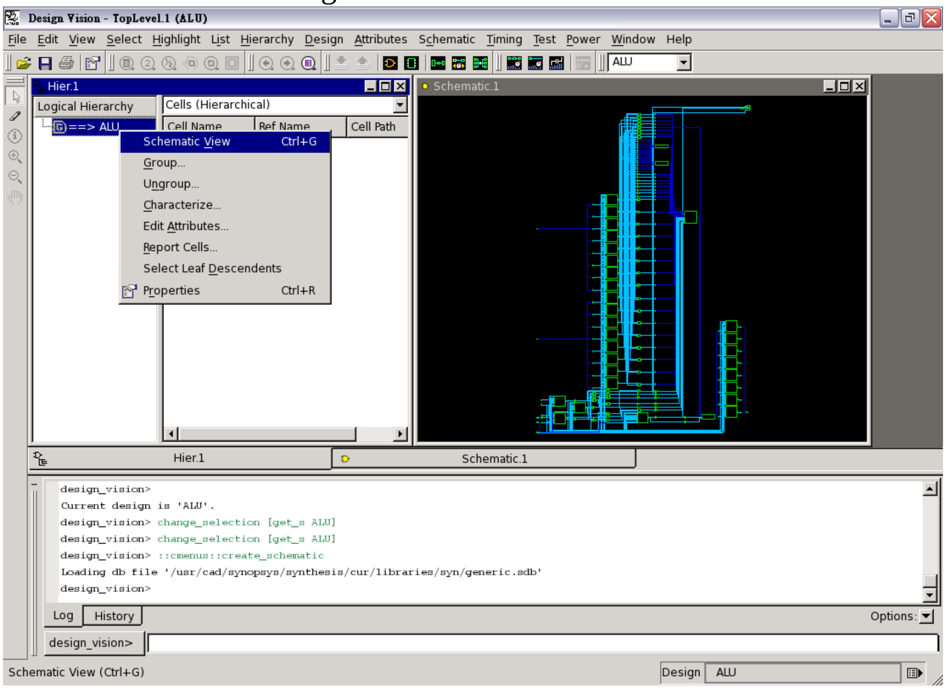
5. read *Lab1_alu.v*



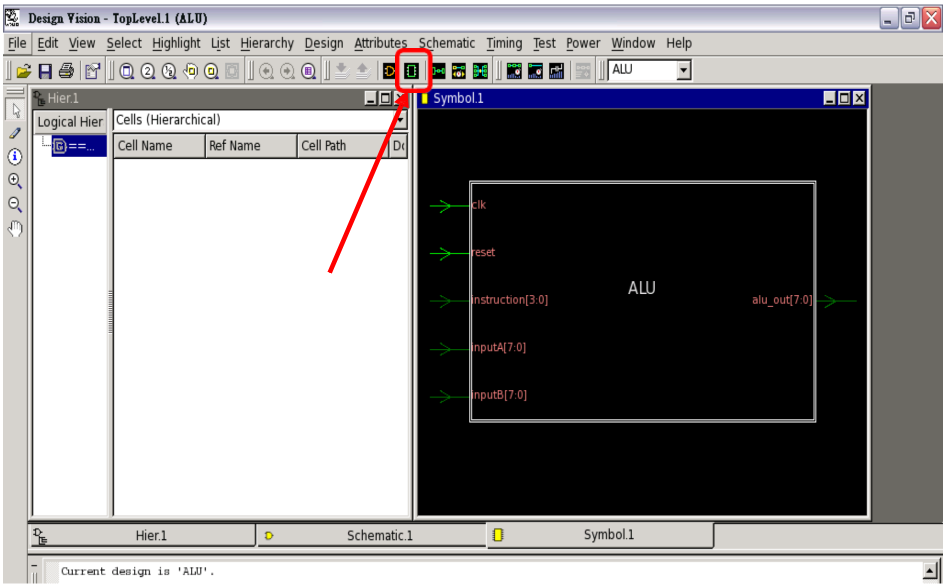
5. Check the log information. If any error or warning message, you have to fix it! After that, checking all the registers are filp-flop type. You have to modify your verilog code, if there is the **latch** in your circuit!

Register Name	Type	Width	Bus	MB	AR	AS	SR	SS	ST
reg_B_reg	Flip-flop	8	Y	N	Y	N	N	N	N
reg_A_reg	Flip-flop	8	Y	N	Y	N	N	N	N
reg_ins_reg	Flip-flop	4	Y	N	Y	N	N	N	N
alu_out_reg	Flip-flop	8	Y	N	Y	N	N	N	N

6. Synopsys Design analyzer will translate verilog code into G-tech model. Double click the icon “ALU”, and click the right button then choose Schematic view. We can get the G-tech MAP.

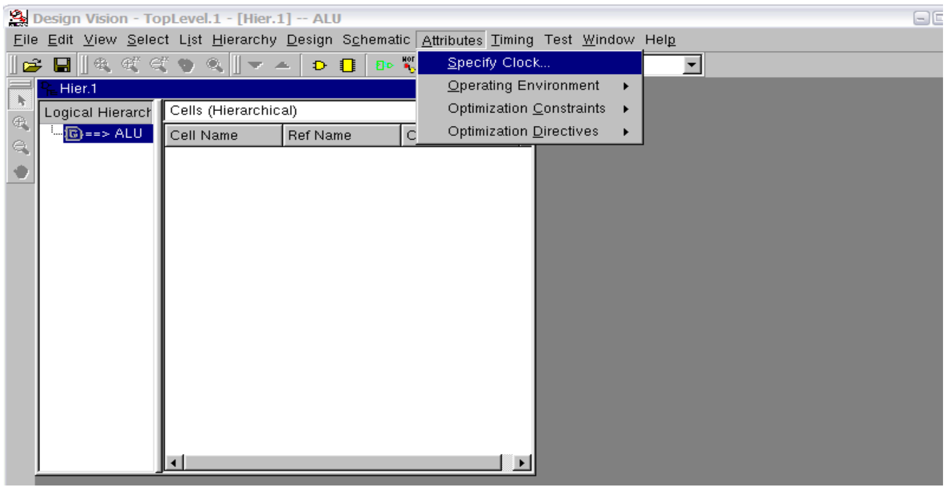


7. Or you can create a symbol view by click on the following symbol view button. The symbol view is as the right window.

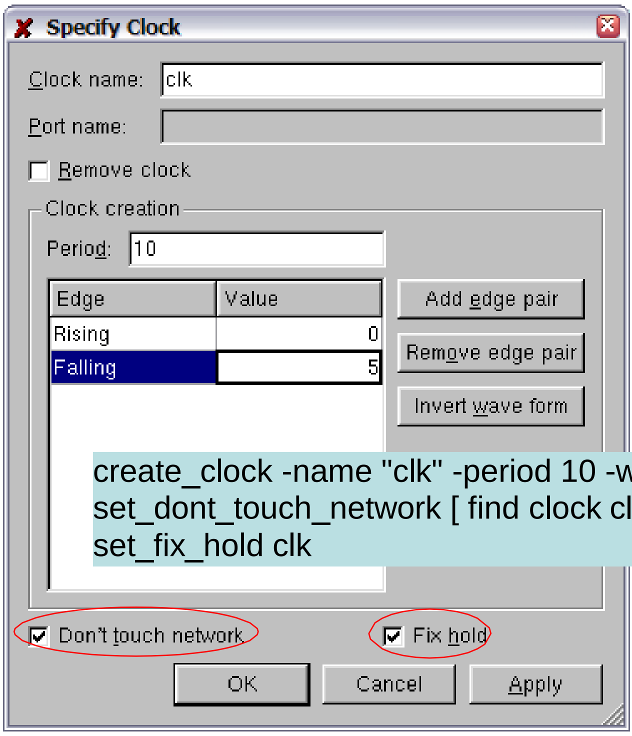


Set clock

8. Back to the symbol view window. **Select the CLK**(at Symbol View) port and click “**Attributes**”-“**Specify Clock**”.



9. Specify the clock as period 10ns. (100 MHz). Don’t forget to select “**don’t touch network**” and “**fix hold**”.

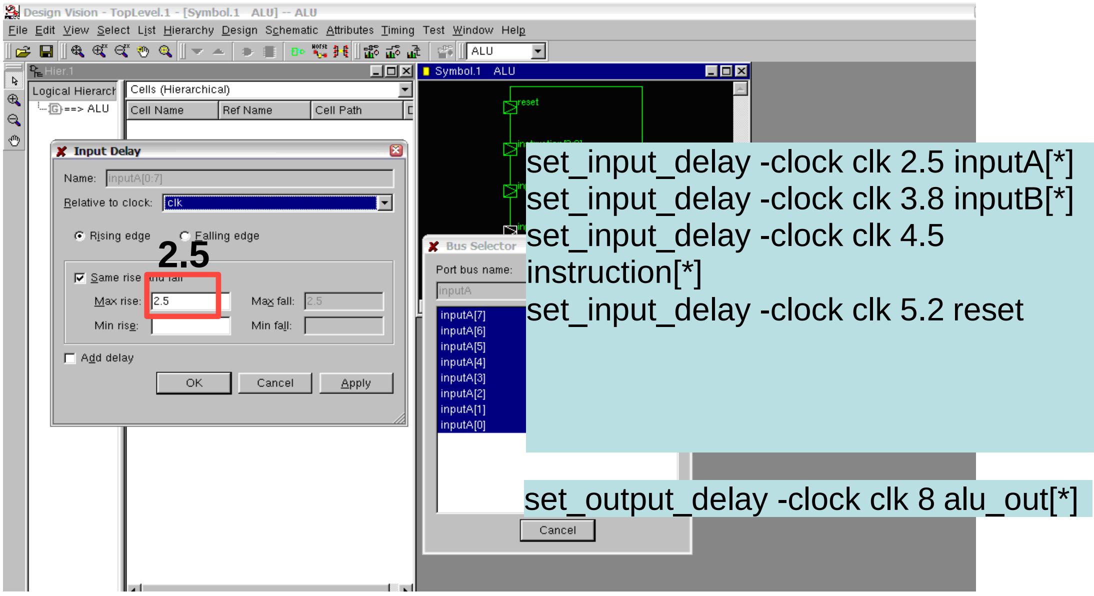


create_clock -name "clk" -period 10 -waveform {"0" "5"} {"clk"}
set_dont_touch_network [find clock clk]
set_fix_hold clk

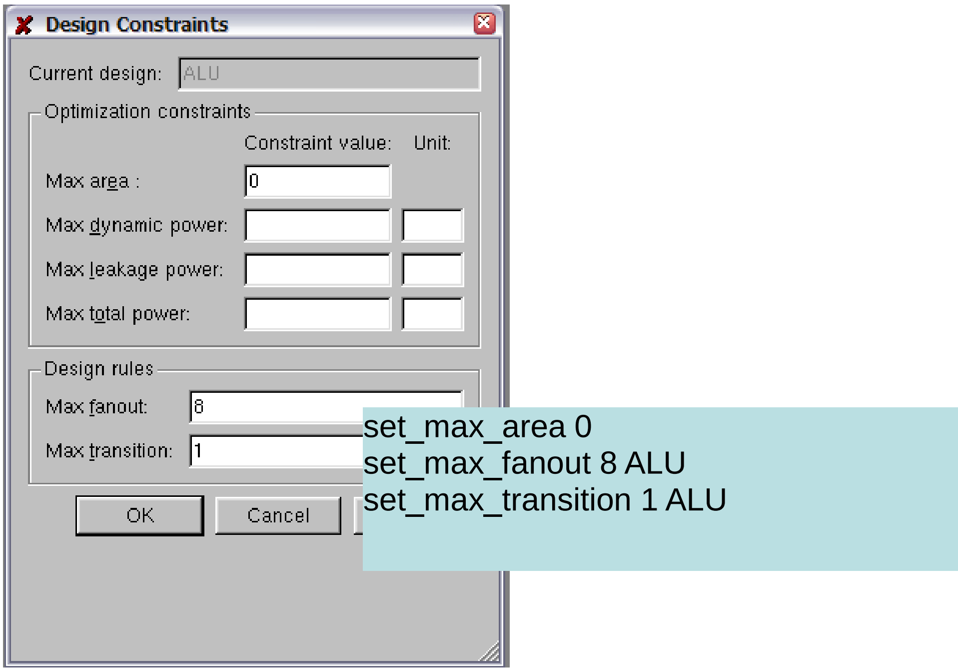
And then type in following command to change the wire load model:
set_operating_conditions "typical" -library "typical"
set_wire_load_model -name "ForQA" -library "typical"
set_wire_load_mode "segmented"

Set operating environment

10. Select “inputA” in the Symbol View and click “**Attribute**”-“**operating environment**”-“**input delay**”. Set 2.5ns input delay.



11. Set “input B” as 3.8ns input delay; “instruction” as 4.5ns input delay; “reset” as 5.2ns input delay.
12. By the same way, setup output delay 8ns on “alu_out”.
13. Click “**Attribute**”-”**optimization Constraints**”-“**Design constraints**”. Set max area is 0. Max fan-out is 8. max transition is 1. The error message in GUI mode could be ignored. And input following command:
set_boundary_optimization "*"
set_fix_multiple_port_nets -all -buffer_constant

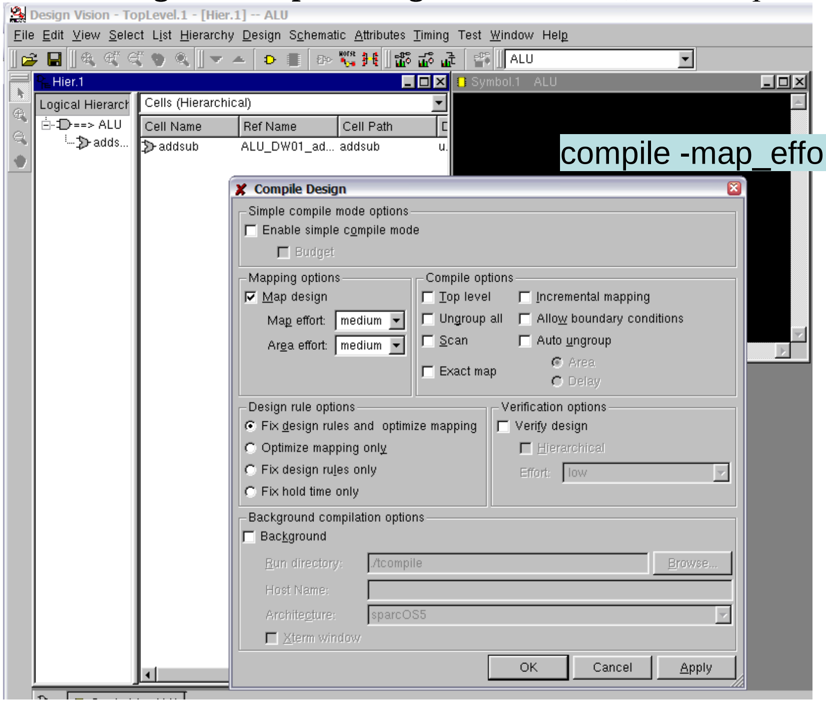


Check Design

14. Click "**Design**"-"**Check Design**". Click "OK". If any error message, there may be error at above steps. Redo these steps!

Design Optimization

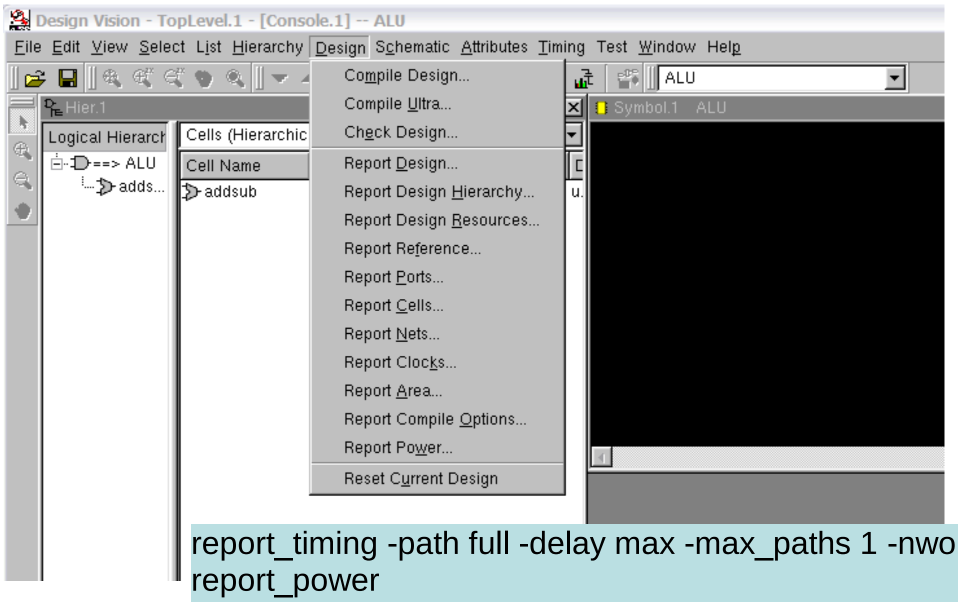
15. Click "**Design**"-"**Compile Design**". Click "OK", start to optimize ALU.



compile -map_effort medium

Report

16. Few seconds later, We will get our gate level circuit. We must to check our circuit met our conditions or not at first.



report_timing -path full -delay max -max_paths 1 -nworst 1
report_power
report_area -nosplit

17. Try to select different report item and check the report. For example: Check timing, use “Timing”=>“Report Timing Paths”; check power, use “Design” => “Report Power”; check area, use “Design”=>“Report Area”

==========================
== The following is timing report.
==========================

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by group
Design : ALU
Version: G-2012.06
Date   : Tue Oct 27 13:20:37 2015
****************************************

Operating Conditions: typical  Library: typical
Wire Load Model Mode: segmented

  Startpoint: alu_out_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: alu_out[0] (output port clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  ALU                ForQA                 typical

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  alu out reg[0]/CK (DFFRX1)               0.00       0.00 r
  alu out reg[0]/Q (DFFRX1)                0.28       0.28 f
  alu_out[0] (out)                         0.00       0.28 f
  data arrival time                                   0.28

  clock clk (rise edge)                   10.00      10.00
  clock network delay (ideal)              0.00      10.00
  output external delay                   -8.00       2.00
  data required time                                  2.00
  -----------------------------------------------------------
  data required time                                  2.00
  data arrival time                                  -0.28
  -----------------------------------------------------------
  slack (MET)                                         1.72
```

Slack is Positive!!

==========================
== The following is power report.
==========================

```
****************************************
Report : power
        -analysis_effort low
Design : ALU
Version: G-2012.06
Date   : Tue Oct 27 13:27:36 2015
****************************************
```

```
Library(s) Used:

   typical (File:
/home/raid2_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/ty
pical.db)


Operating  Conditions:  typical  Library: typical
Wire Load Model Mode: segmented

Design       Wire  Load  Model         Library
------------------------------------------------
ALU                ForQA          typical


Global Operating Voltage = 1.2
Power-specific unit information :
   Voltage Units = 1V
   Capacitance Units = 1.000000pf
   Time Units = 1ns
   Dynamic Power Units  = 1mW   (derived from V,C,T units)
   Leakage Power Units = 1pW


  Cell  Internal  Power =  81.6075 uW   (94%)
  Net  Switching  Power =  5.3657 uW    (6%)
                        ---------
Total  Dynamic  Power   =   86.9732 uW  (100%)

Cell  Leakage  Power    = 441.1685 nW
```

==========================

== The following is area report.

==========================

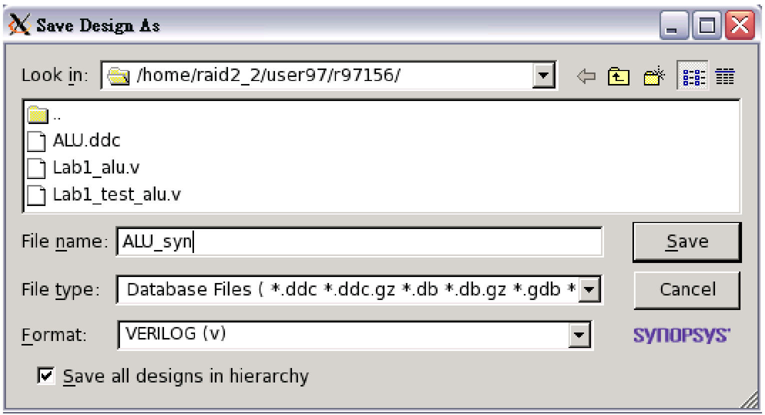
```
****************************************
Report : area
Design : ALU
Version: G-2012.06
Date   : Tue Oct 27 13:31:35 2015
****************************************
Library(s) Used:
typical (File:
/home/raid2_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/typ
ical.db)
Number of ports:                          30
Number of nets:                          140
Number of cells:                          83
Number of combinational cells:            53
Number of sequential cells:               28
Number of macros:                          0
Number of buf/inv:                         6
Number of references:                     14

Combinational area:       1003.163413
Noncombinational area:     903.016769
Net Interconnect area:      18.000000

Total cell area:          1906.180182
Total area:               1924.180182
```

Notice at the last line "Slack". We know this circuit is safe at timing.

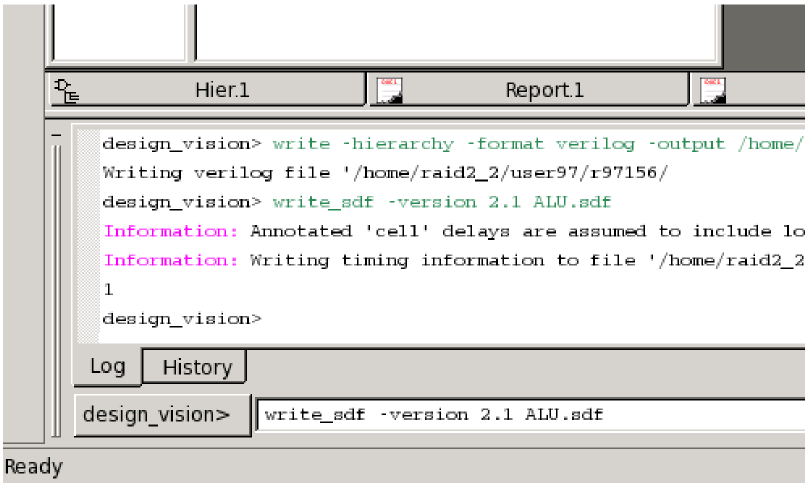
18. Repeat Step1. ~ Step 15. Try to change different settings and modify the constraints.

19. If the result is met your requirement, Synthesis is ending. Then, we must export the design to a file. Select "**File**"-"**Save**"(it will save the all the settings and results in **ALU.ddc**). You can also select **"File"-"Save As"** and choose **"DDC"** formate to save this design.

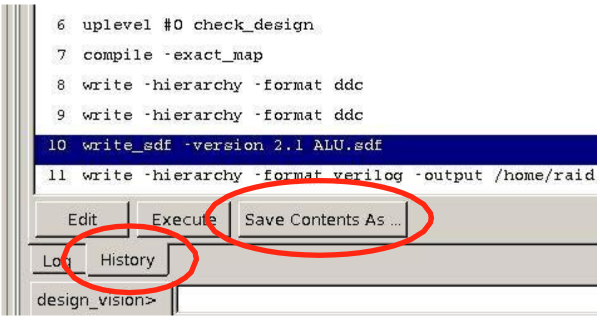
20. Save the gate level netlist in verilog formate. Select **"File"-"Save As"** , choose **"VERILOG"** formate with the File name **"ALU_syn"**, and click **"Save"**.



21. Finally, to save the timing information, you have to type the following commend in the command line: **"write_sdf -version 2.1 ALU.sdf"**. That will generate the timing information of this design.



22. You might use "**write_script > script_file**" command or save the comment history **"History"-"Save Contents As..."** to generate a script with the constraints you have made. After that, you could use "**include script_file**" command or **"File"-"Execute Script"** button to re-run all steps automatically.



23. For verilog gate-level simulation, you may add "$sdf_annotate(sdf_file, testbench_module.ALU_instance);" in initial block in your test bench to use timing information for simulation.

Lab1-2: Synopsys Memory Library Generation

Environment Setup:

1. Synopsys[TM] Design Compiler is not able to read the library file directly generated from the Memory Compiler. Please follow the below instructions to convert the `.lib` files to `.db` files that the Design Compiler can use.
2. We have gotten the TSMC 0.13µm 512 words * 8bit memory library files "HSs13n_512x8_fast_syn.lib", "HSs13n_512x8_slow_syn.lib", "HSs13n_512x8_typical_syn.lib" in the Verilog Lab2. Please check them whether they are in your working directory or not.

Compile the Memory Library:

3. Now we are going to compile the .lib files to .db files. Firstly, check your Synopsys environments are correctly setup.
4. Run **dc_shell -xg** under UNIX. And execute the following command in dc_shell:

```
dc_shell> read_lib NAME.lib
dc_shell> write_lib USER_LIB –output OUTPUT_FILE_NAME
```

You should fill the name of one of the three memory files onto "NAME.lib" here. "USER_LAB" is the Library Name you defined while generating the "Synopsys library model(.lib)" in Memory Compiler.

As for HSs13n_512x8_fast_syn.lib, just type in following instructions:

```
dc_shell> read_lib HSs13n_512x8_fast_syn.lib
dc_shell> write_lib HSs13n_512x8 –output
HSs13n_512x8_fast_syn.db
```

The produced .db files are the synthesis models that Synospys Design Compiler can use. Remember to add where you put the .db files in the "**search_path**" and add the .db file names in the "**link_library**" and "**target_library**" in your .synopsys_dc.setup before further synthesis of the RAM.

Checkpoints:

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

1. Show your synthesis result. Including synthesized gate level codes, sdf file, timing and area reports.
2. You have generated the memory db file. Please add them in to your .synopsys_dc.setup.

END of LAB

Creator:
1st Edition: Jazz Yang, 2002
2nd Edition: Lin Houng-mao, 2003
3rd Edition: Yu-Lin Chang, 2004, 2005
4th Edition: Yu-Lin Chang, 2006
5th Edition: Jui-Hsin Lai(Larry), 2008
6th Edition: Chih-Fan Lai, 2009
7th Edition: Chia-Wei Chang, 2010
8th Edition: Tung-Chien Chen, 2011
9th Edition: Yung-Lin Huang (Cary), 2012
10th Edition: Pin-Hung Kuo, 2015