Homework 2
Logic Synthesis with Design Compiler

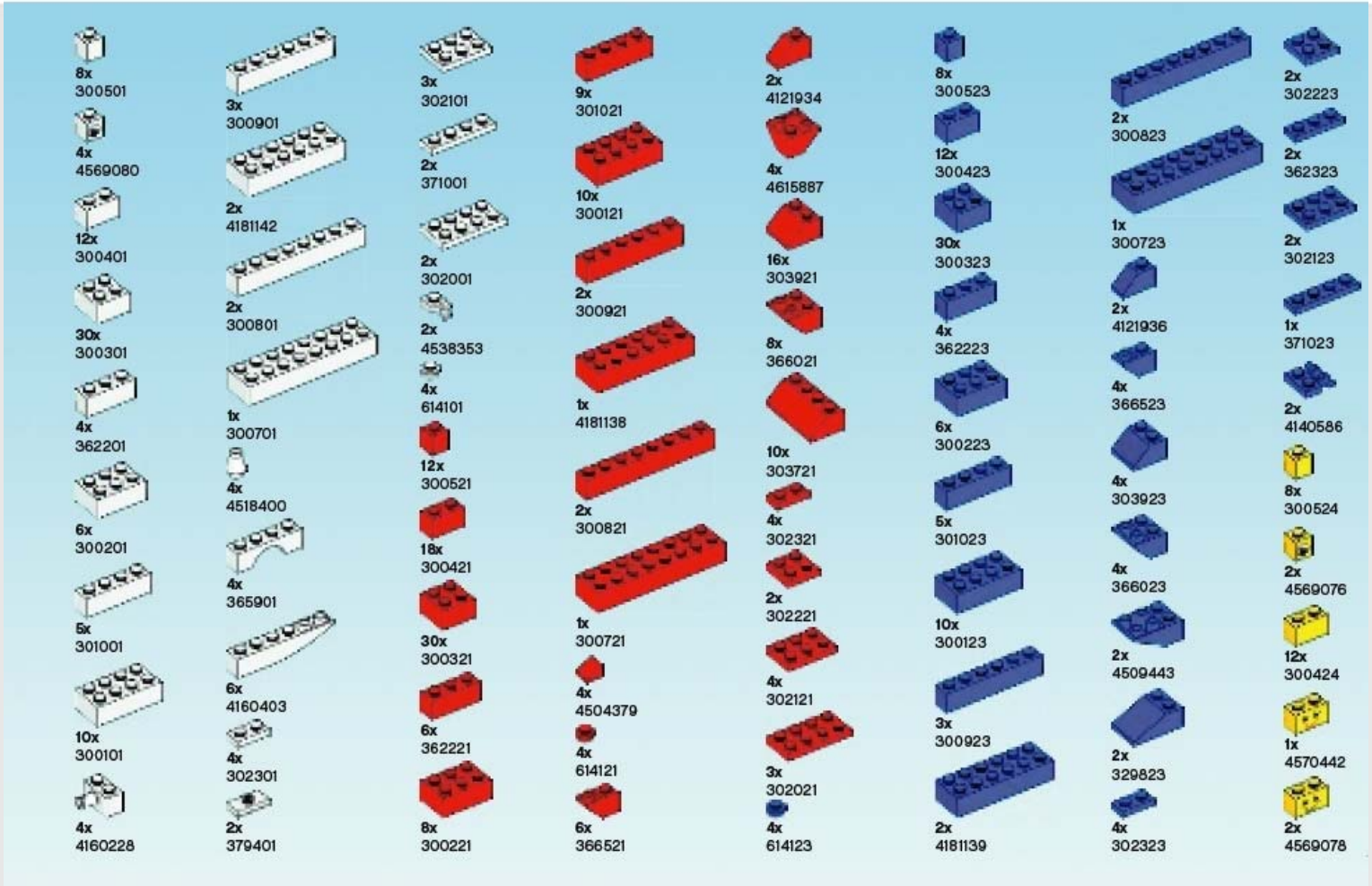
Teacher : Yu-Cheng Fan

ISLAB

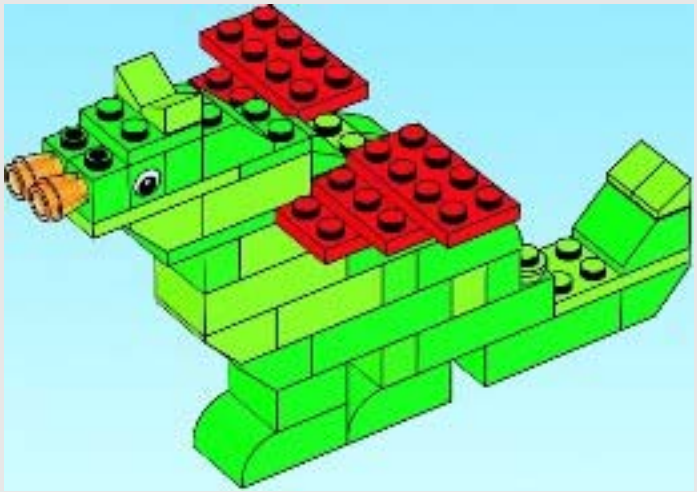
What is Logic Synthesis? We Say a Lego Story.

We Use Lego to Build a Dream in Our Childhood

How to Build a Dream? Based on Cells……

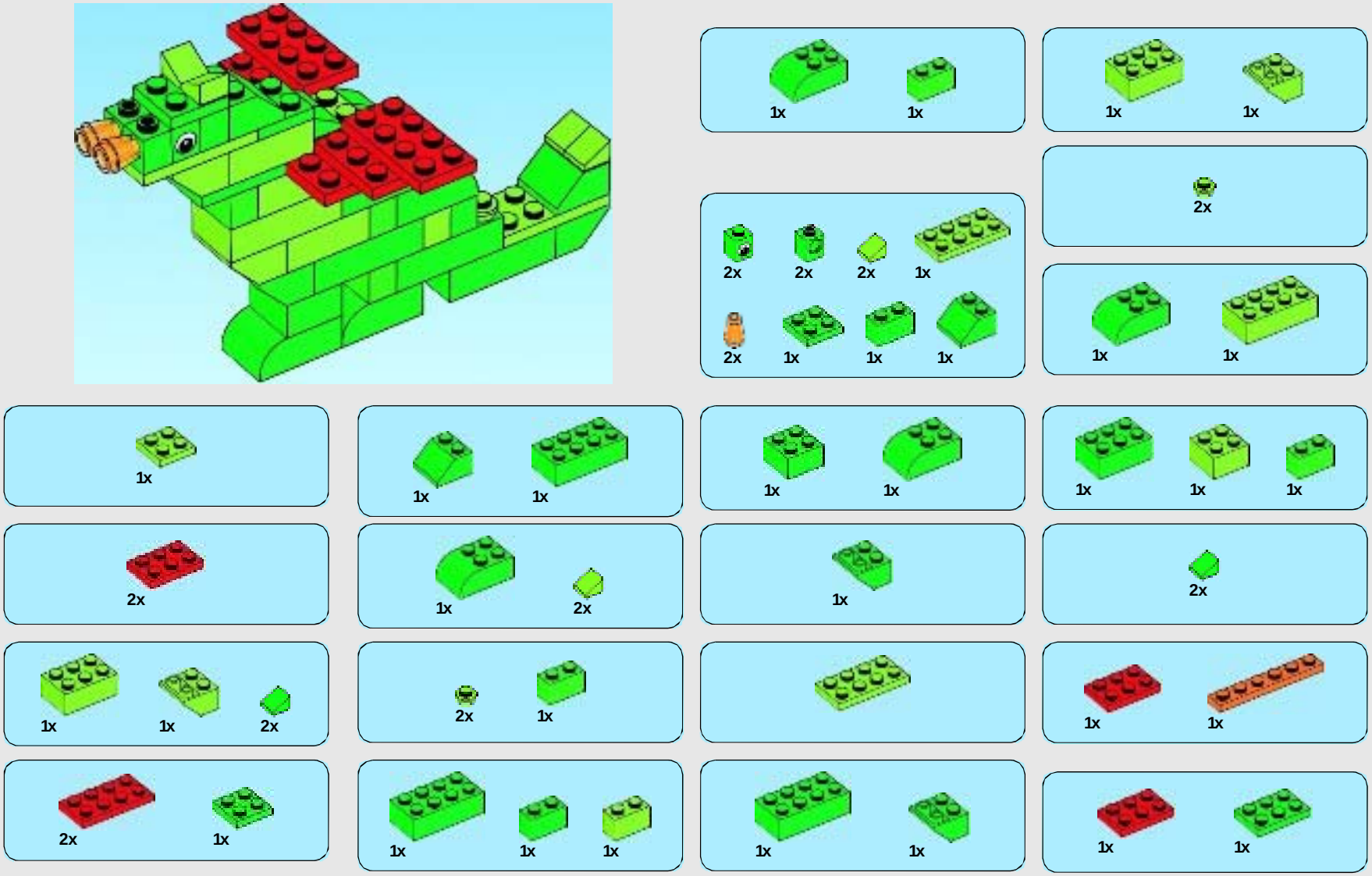


Step 1 ：Function Description

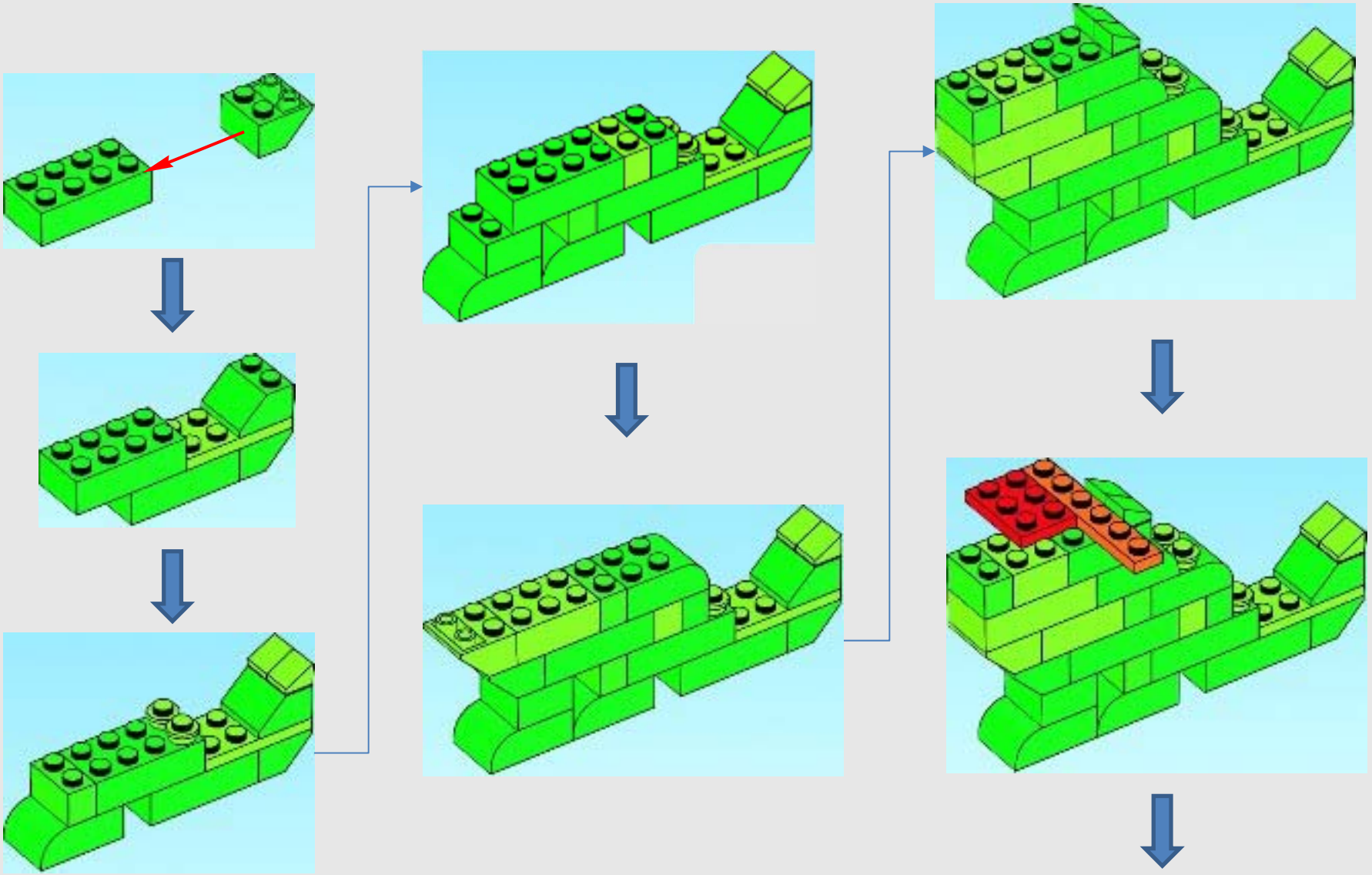


System Spec. → Algorithm → RTL Code

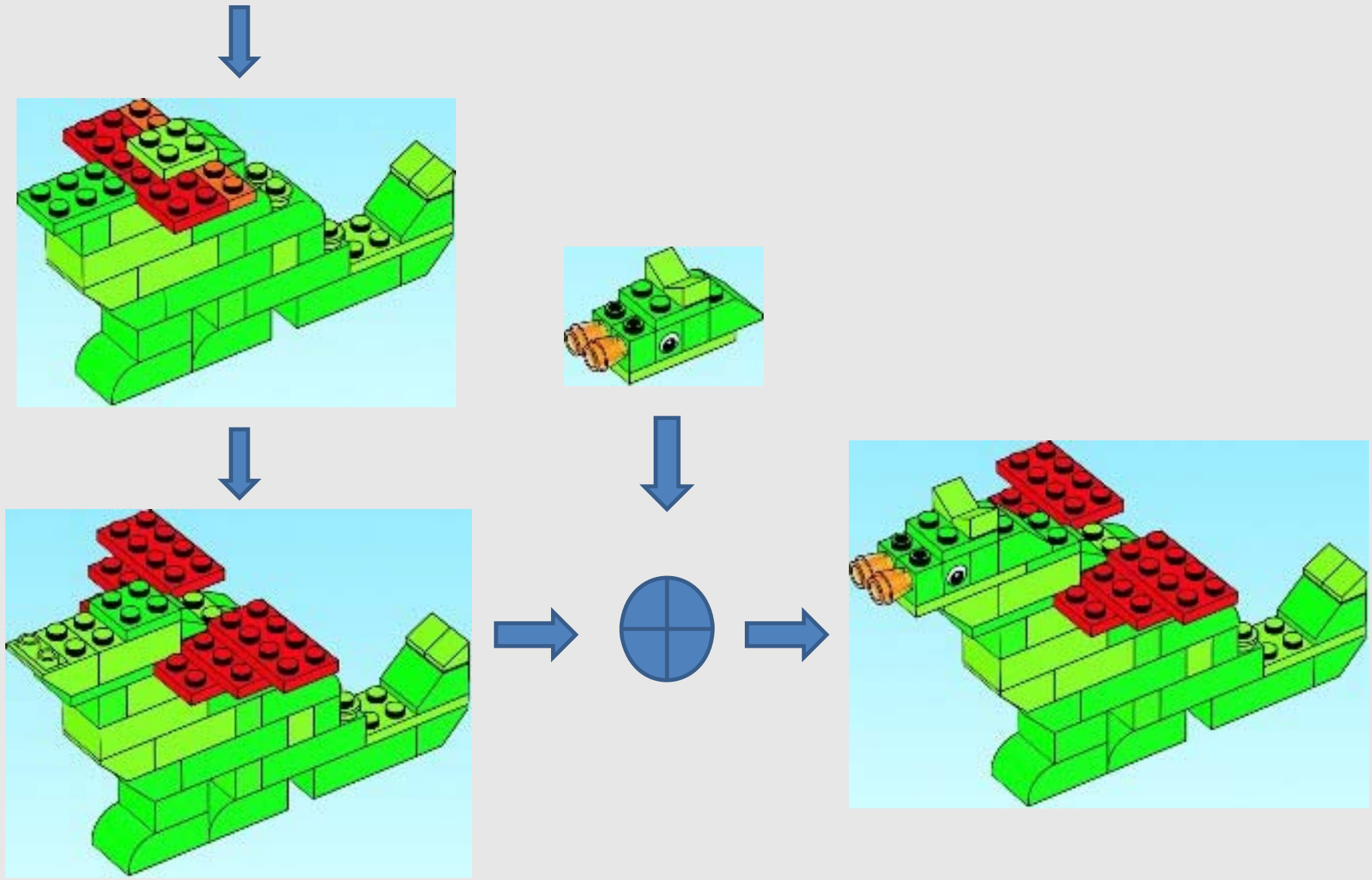
Step 2 ：Blocks Analysis
(Design Compiler and Cell Library Design)



Step 3 ：Logic Synthesis and Optimization

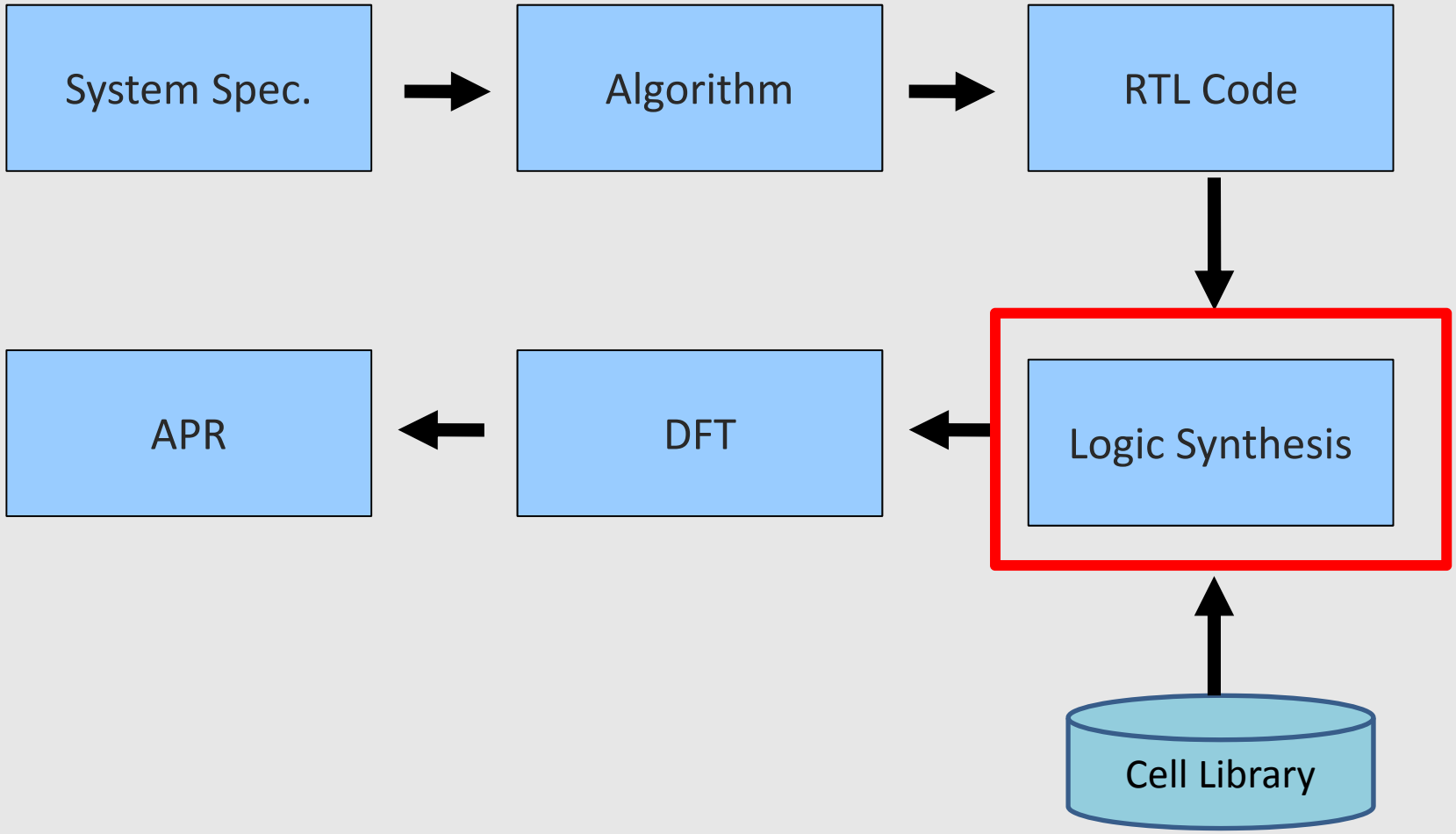


Step 3 ：Logic Synthesis and Optimization



How to Build a Circuit Dream?

We Adopt the Cell-Based Design Flow……



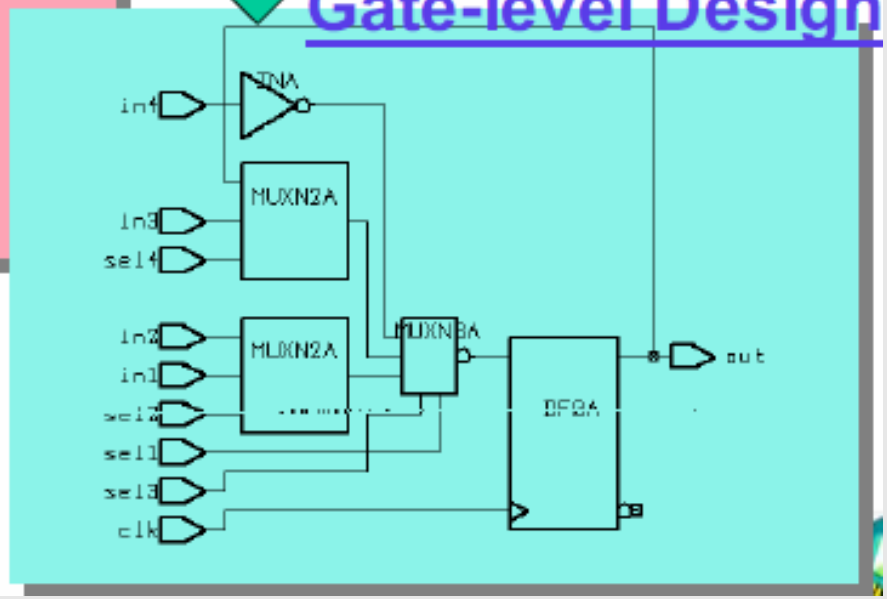
Logic Synthesis

```
always @(posedge clk) begin
  if (sel1) begin
    if (sel2)
      out=in1;
    else
      out=in2;
  end
  else if (sel3) begin
    if (sel4)
      out=in3;
  end
  else out=in4;
end
```

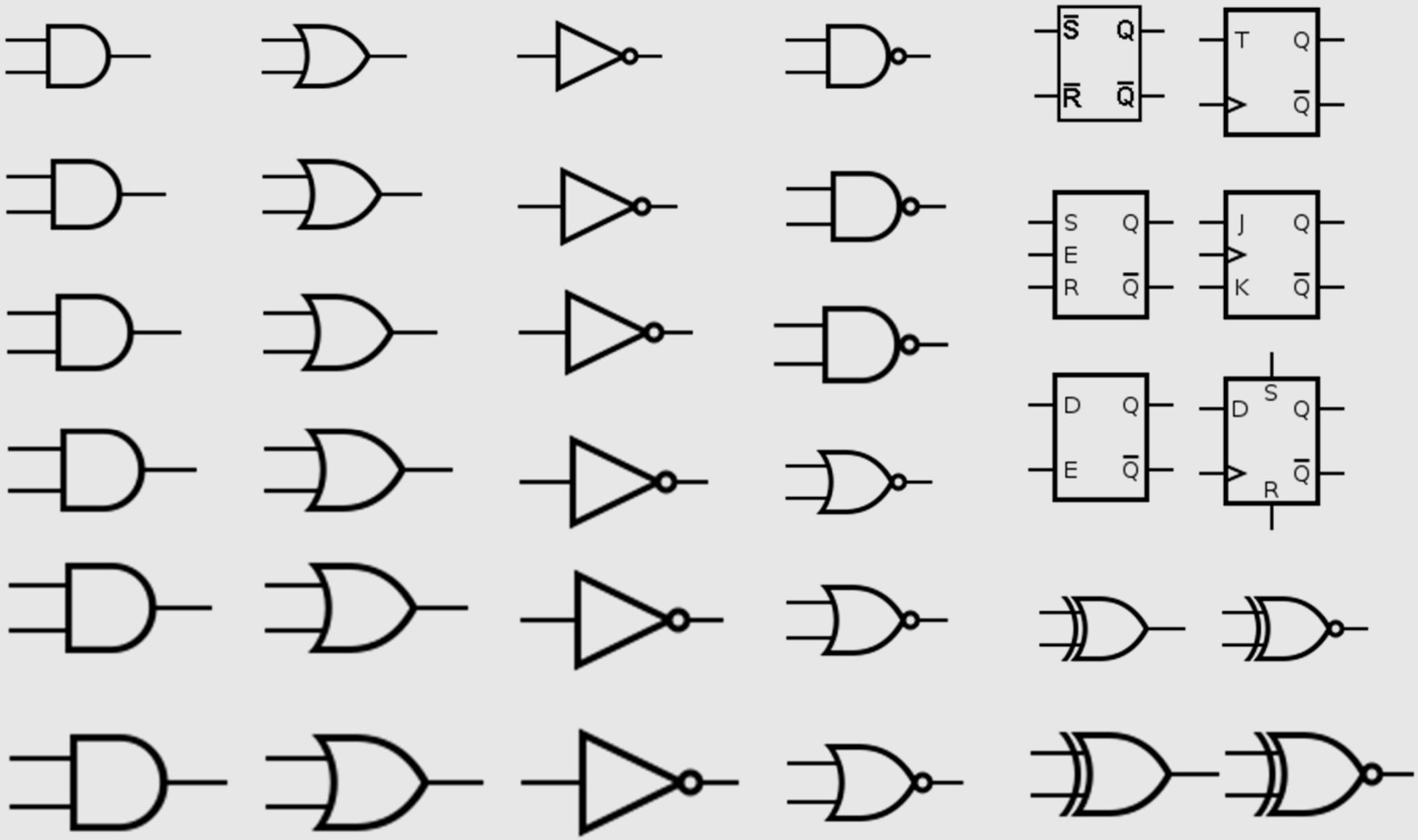
RTL

- Logic Synthesis translates RTL design to gate-level design.

Gate-level Design

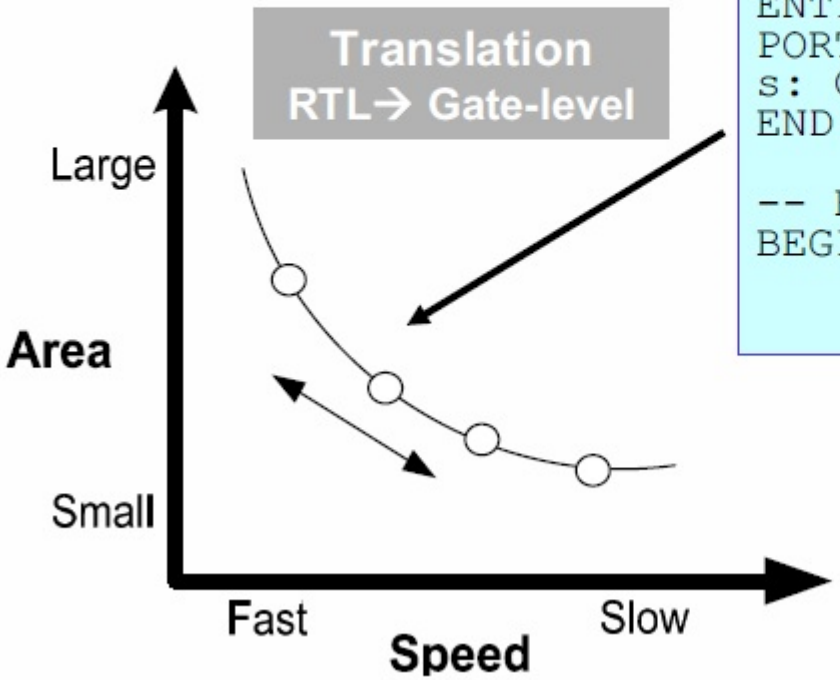


Cell Library (Logic Gate v.s. Lego)



Synthesis is Constraint Driven

- Logic Synthesizer first translates RTL design to an intermediate gate-level design , then optimize according to the area and timing constraint.




```
--Interface
ENTITY adder IS
PORT (a, b:IN unsigned(0 to 31);
s: OUT unsigned (0 to 32));
END adder;

-- Behavioral representation
BEGIN
        s <= a + b;
        END behavioral;
```

Environment Setup (1/3)

- Set the environmental files
  - Open Terminal
  - Key in " cd ~“
  - Key in "cp /home/standard/Environment_Setup_File/cshrc .cshrc"
  - Key in " ls –a .cs* "

```
[t00418099@is1abx2 ~]$ ls -a .cs*
.cshrc
```

  - Key in " source .cshrc“

Environment Setup (2/3)

Create a New Folder for Synthesis

Ex:

mkdir Synthesis　　建立欲用來合成的資料夾

cd Synthesis　　進入欲用來合成的資料夾

Copy File for Environment Setting

cp Δ /home/standard/Environment_Setup_File/synthesis_setup_for_18/synopsys_dc.setup Δ .synopsys_dc.setup

Δ處記得要按一次空白鍵

Environment Setup (3/3)

.synopsys_dc.setup

```
set company "CIC"
set designer "Student"

set#search_path       " .  $Your_path/CBDK_TSMC018_Arm/CIC/SynopsysDC $search_path "
set#target_library    "    #slow.db  fast.db tpz973gvwc.db#tpz973gvbc.db "
set#link_library      " *#$target_library  dw_foundation.sldb "
set#symbol_library    "#tsmc18.sdb  generic.sdb "
set#synthetic_library "#dw_foundation.sldb "

set#verilogout_no_tri true
set#hdlin_enable_presto_for_vhdl "TRUE"
set#sh_enable_line_editing true
history keep 100
alias h history
```

- link_library : The library used for interpreting input description.
- target_library : The ASIC technology that the design is mapped to.
- symbol_library : Used during schematic generation.
- search_path : The path to search for unsolved reference library or design.
- synthetic_library : Designware to be used.

Synthesizable Code

Copy File for Synthesis

cp /home/standard/summer2013/dv/* .　　千萬別忘了這個點

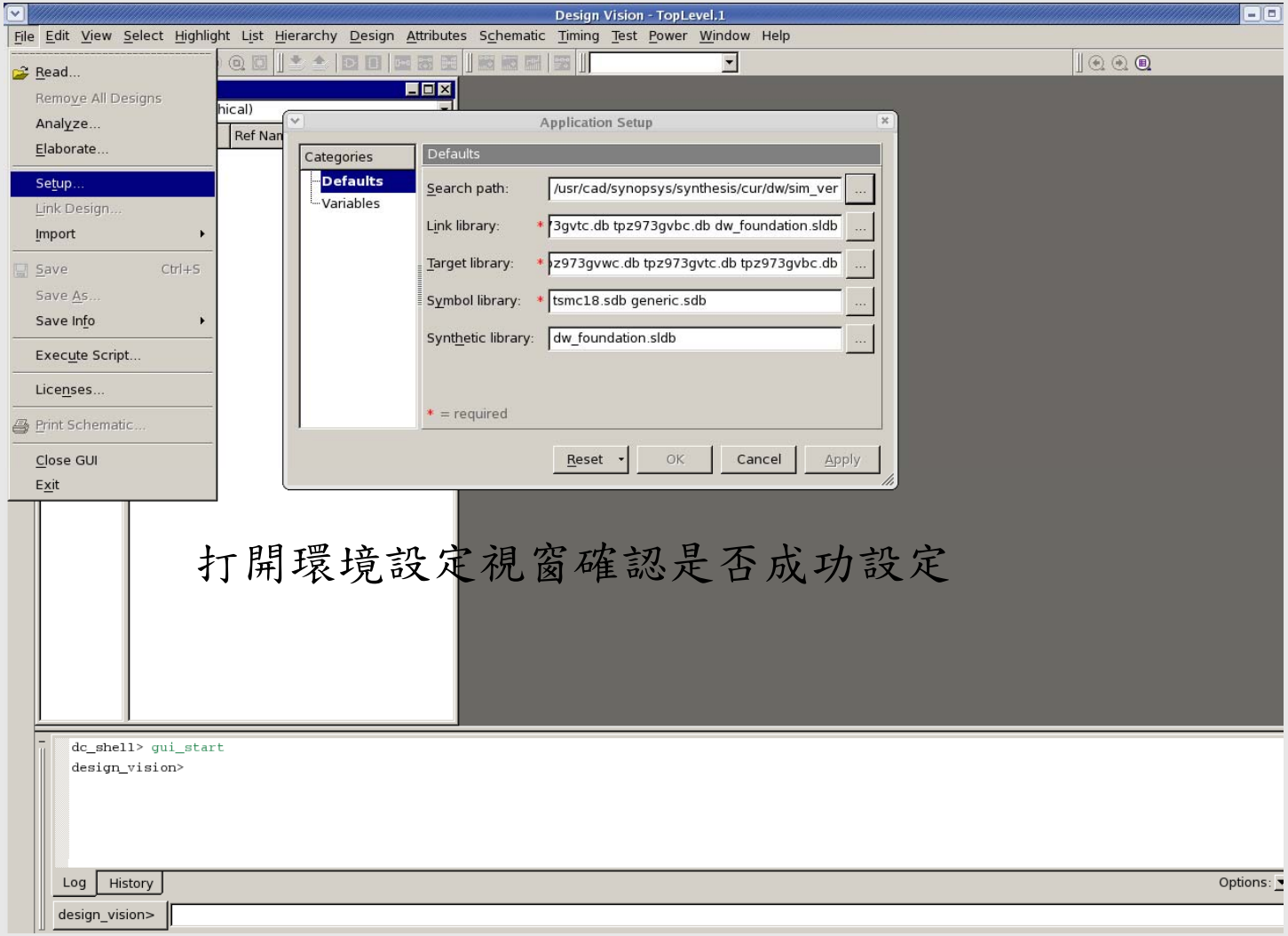
處記得要按一次空白鍵

Open Design Vision

```
終端機
檔案(F)  編輯(E)  顯示(V)  終端機(T)  分頁(B)  求助(H)
— Now source Synopsys Nanosim
— Now source Spring soft nWave(Verdi) —
Platform = LINUX
#############################################################
#          32BIT is the default mode                        #
#     If you want to run 64BIT mode,                        #
#     please set the LD_LIBRARY_PATH and SHLIB_PATH         #
#     to path of 64BIT by yourself.                         #
#############################################################
— Now Plug-in Calibre for Laker —
[t7418082@islabx2 ~]$ ls
Desktop  synthesis
[t7418082@islabx2 ~]$ cd synthesis/
[t7418082@islabx2 ~/synthesis]$ ls
command.log  fadder.v
[t7418082@islabx2 ~/synthesis]$ dv

                        DC Professional (TM)
                           DC Expert (TM)
                            DC Ultra (TM)
                       FloorPlan Manager (TM)
                          HDL Compiler (TM)
                         VHDL Compiler (TM)
                        Library Compiler (TM)
                      DesignWare Developer (TM)
                           DFT Compiler (TM)
                             BSD Compiler
                         Power Compiler (TM)

            Version B-2008.09-SP2 for amd64 — Nov 28, 2008
               Copyright (c) 1988-2008 by Synopsys, Inc.
                          ALL RIGHTS RESERVED

This software and the associated documentation are confidential and
proprietary to Synopsys, Inc. Your use or disclosure of this software
is subject to the terms and conditions of a written license agreement
between you, or your company, and Synopsys, Inc.

The above trademark notice does not imply that you are licensed to use
all of the listed products. You are licensed to use only those products
for which you have lawfully obtained a valid license key.

Initializing...
design_vision> design_vision>
```

Please key in **dv**

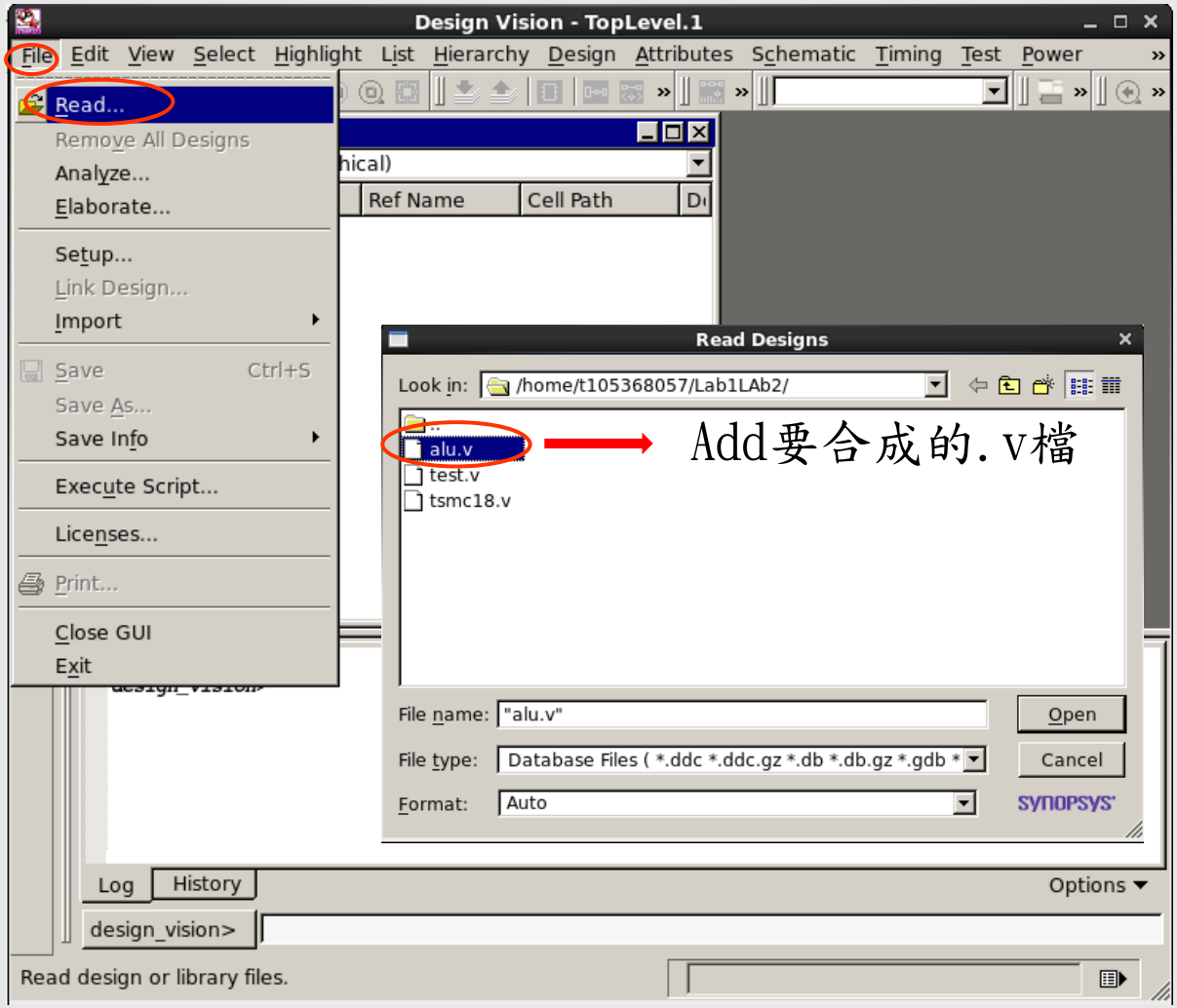
Environment Setup Check



Method 1: Read File

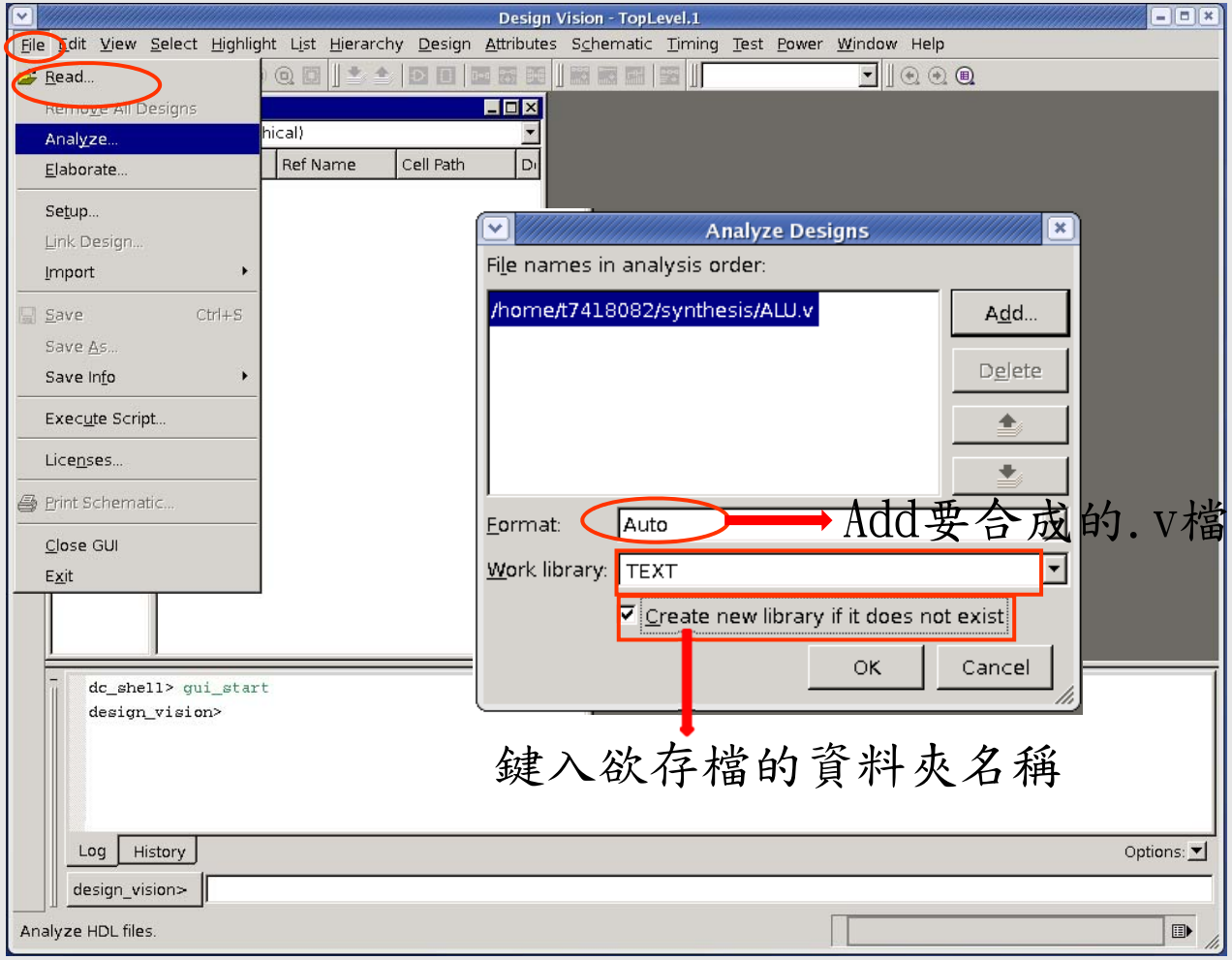
1. Read netlist or other design descriptions into Design Compiler
2. File/Read
3. Support many different formats

- Verilog: .v
- VHDL: .vhd
- System Verilog: .sv
- EDIF
- PLA(Berkeley Espresso): .pla
- Synopsys formats:
  - DB(binary): .db
  - Enhance db file: .ddc
  - Equation: .eqn
  - State table: .st



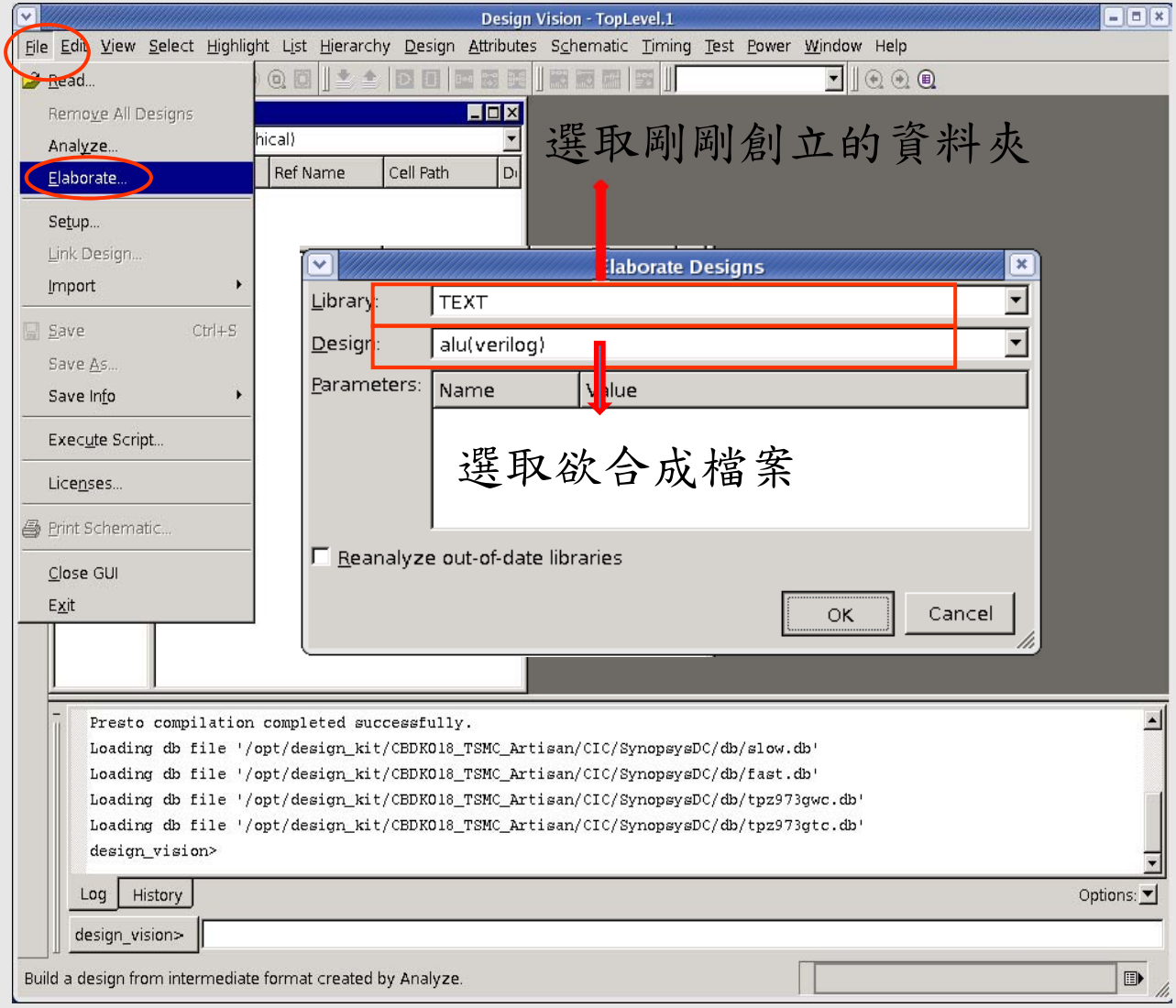
Method 2: Analysis The Design(1/2)

1. Check VHDL & Verilog for syntax and synthesizability
2. Creat intermediate .mr and .pvl and .syn files and places them in library
3. Specified – design library

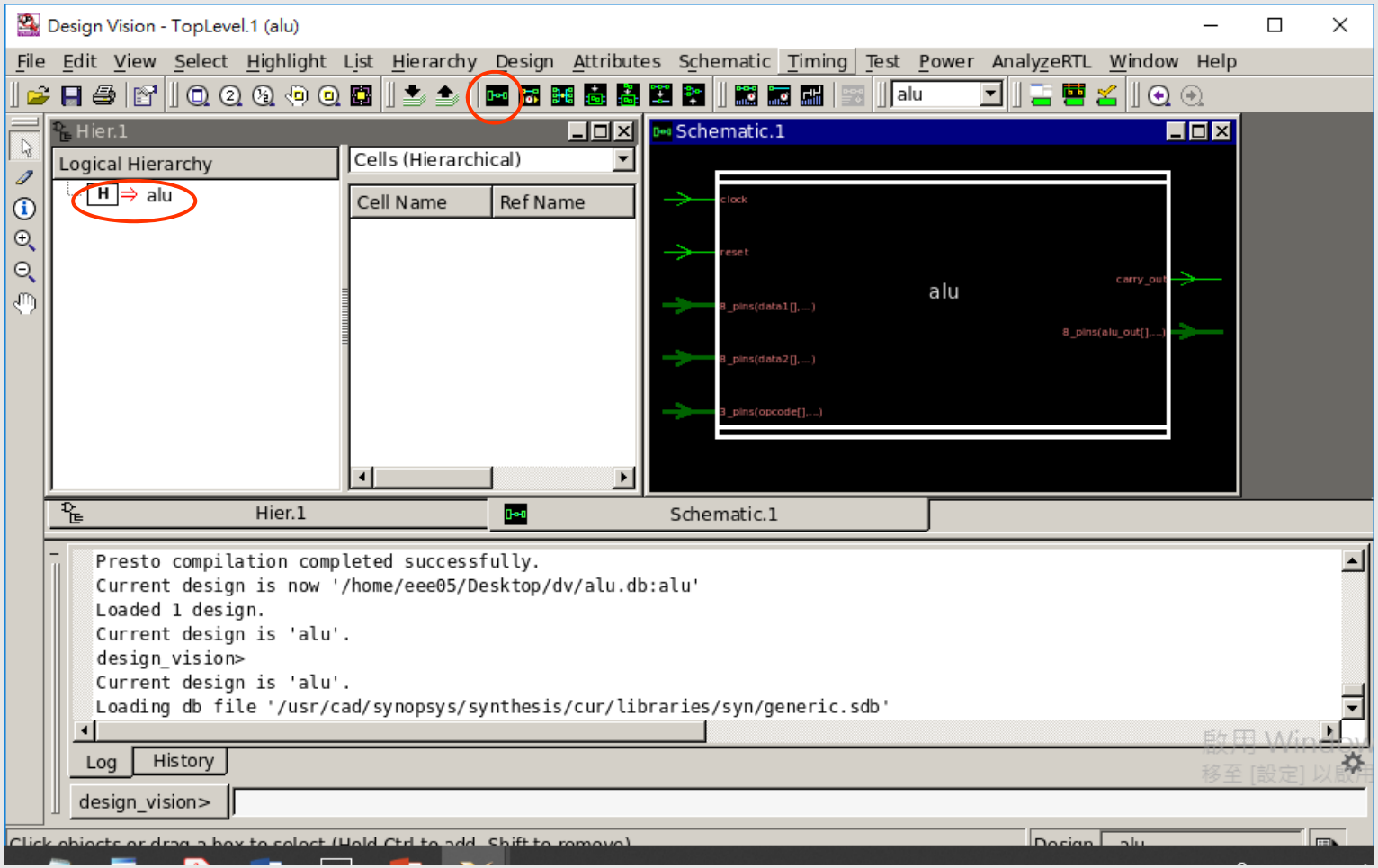


Method 2: Elaborate The Design(2/2)

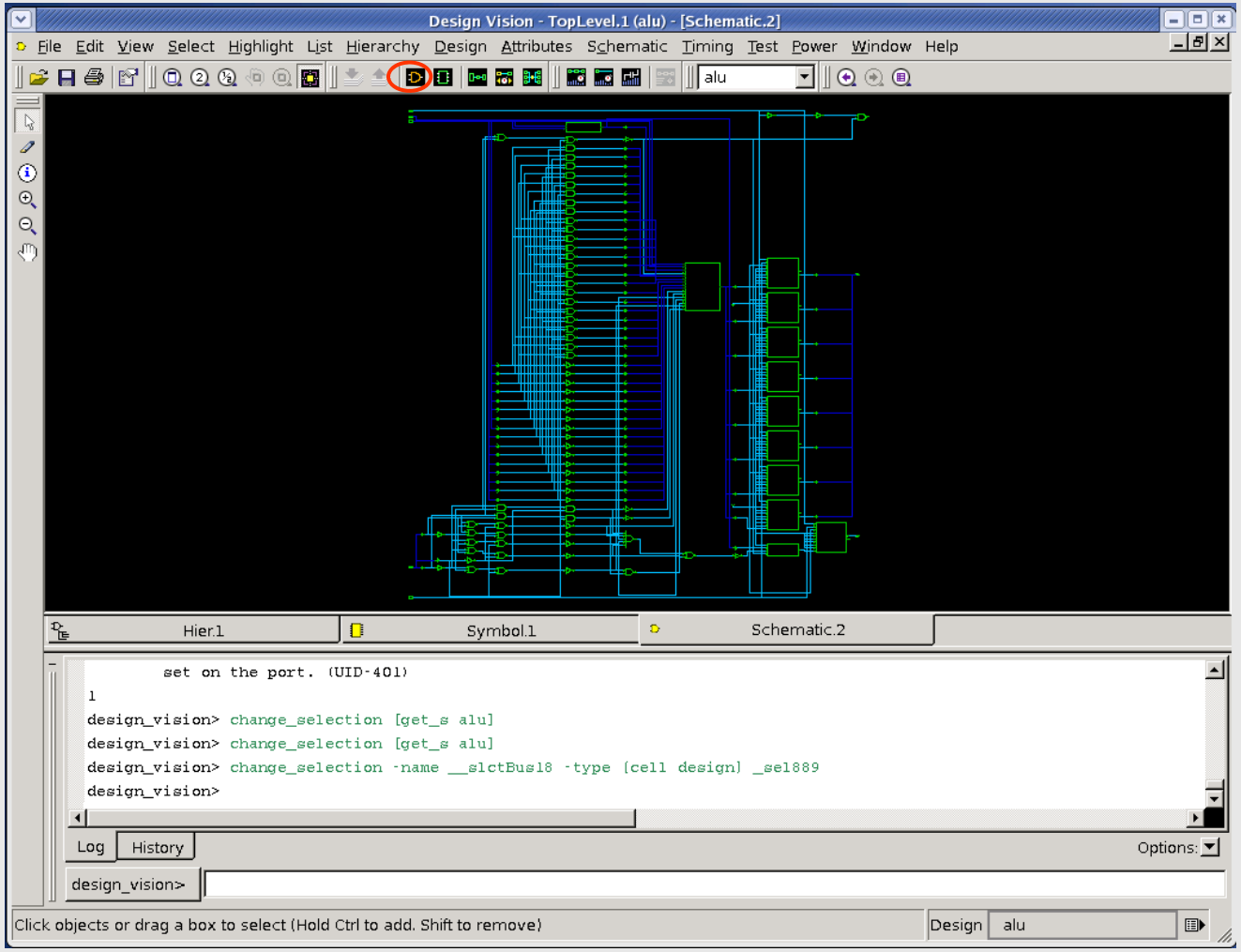
1. Elaborate after analyze to bring design into Design Compiler
2. Look in the design library for intermediate file for design specified



Symbol View


Design Vision - TopLevel.1 (alu)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power AnalyzeRTL Window Help
alu
Hier.1
Logical Hierarchy
alu
Cells (Hierarchical)
Cell Name
Ref Name
Schematic.1
clock
reset
alu
carry_out
Hier.1
Schematic.1
Presto compilation completed successfully.
Current design is now '/home/eee05/Desktop/dv/alu.db:alu'
Loaded 1 design.
Current design is 'alu'.
design_vision>
Current design is 'alu'.
Loading db file '/usr/cad/synopsys/synthesis/cur/libraries/syn/generic.sdb'
Log
History
design_vision>

Schematic View



Define Clock Specification

■Setting Design Constraints

1. set period
2. set waveform
3. set clock skew

   clock訊號至F.F.最大的路徑時間，約0.1~0.3ns
4. set source latency

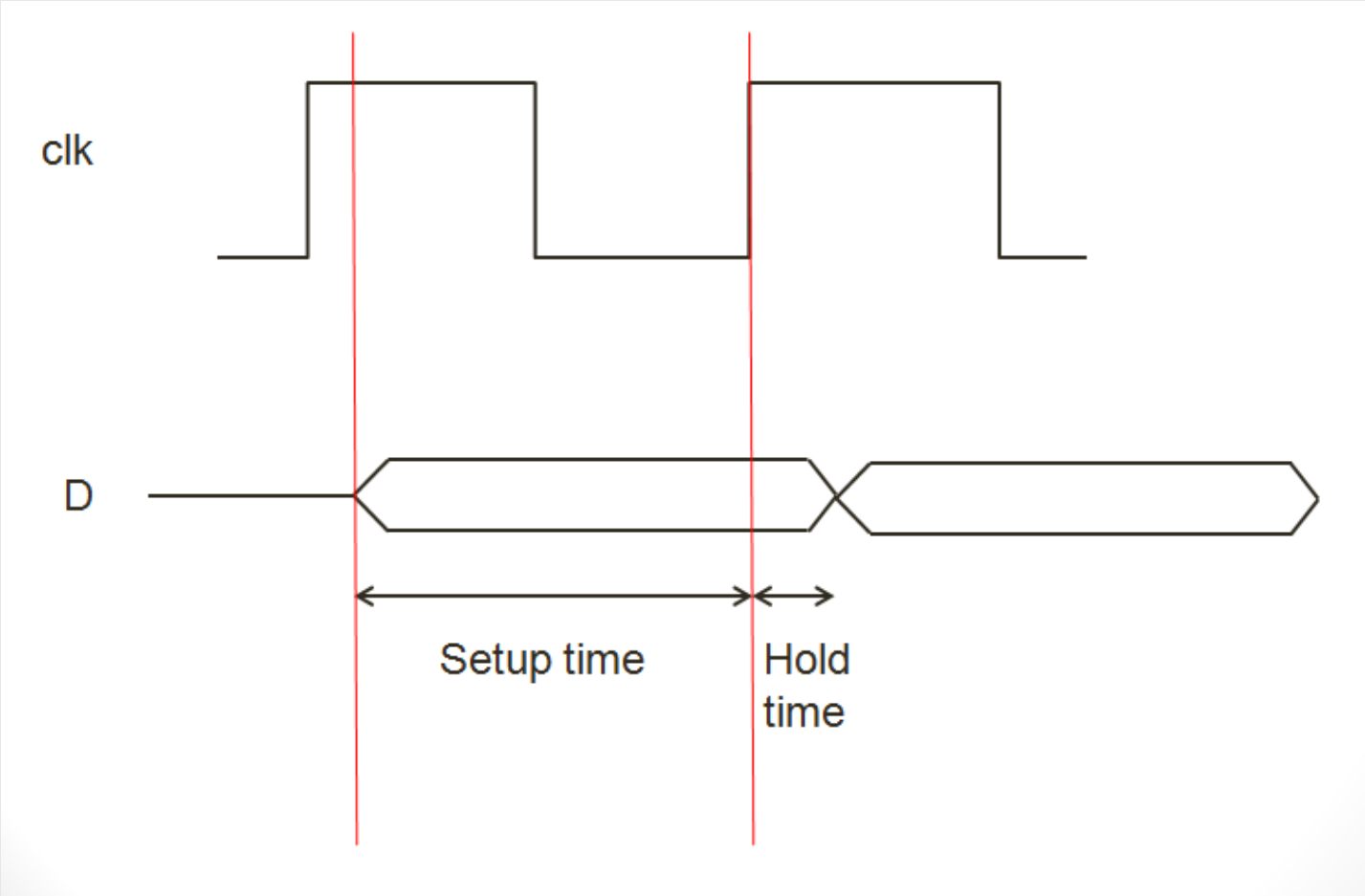
   原始clock至自己定義clock的傳輸時間，有除頻電路或倍頻電路才需要設定
5. set latency

   因為clock後面負載很大，所以加Buffer後產生latency，值大約1~3ns
6. set input transition

   輸入訊號的轉態時間，CIC測試機台環境為0.5ns
7. set clock transition

   F.F.內部clk到Q之轉態時間，值越小、F.F.速度越快，但Power消耗越多，一般設0.1ns

Setup Time & Hold Time

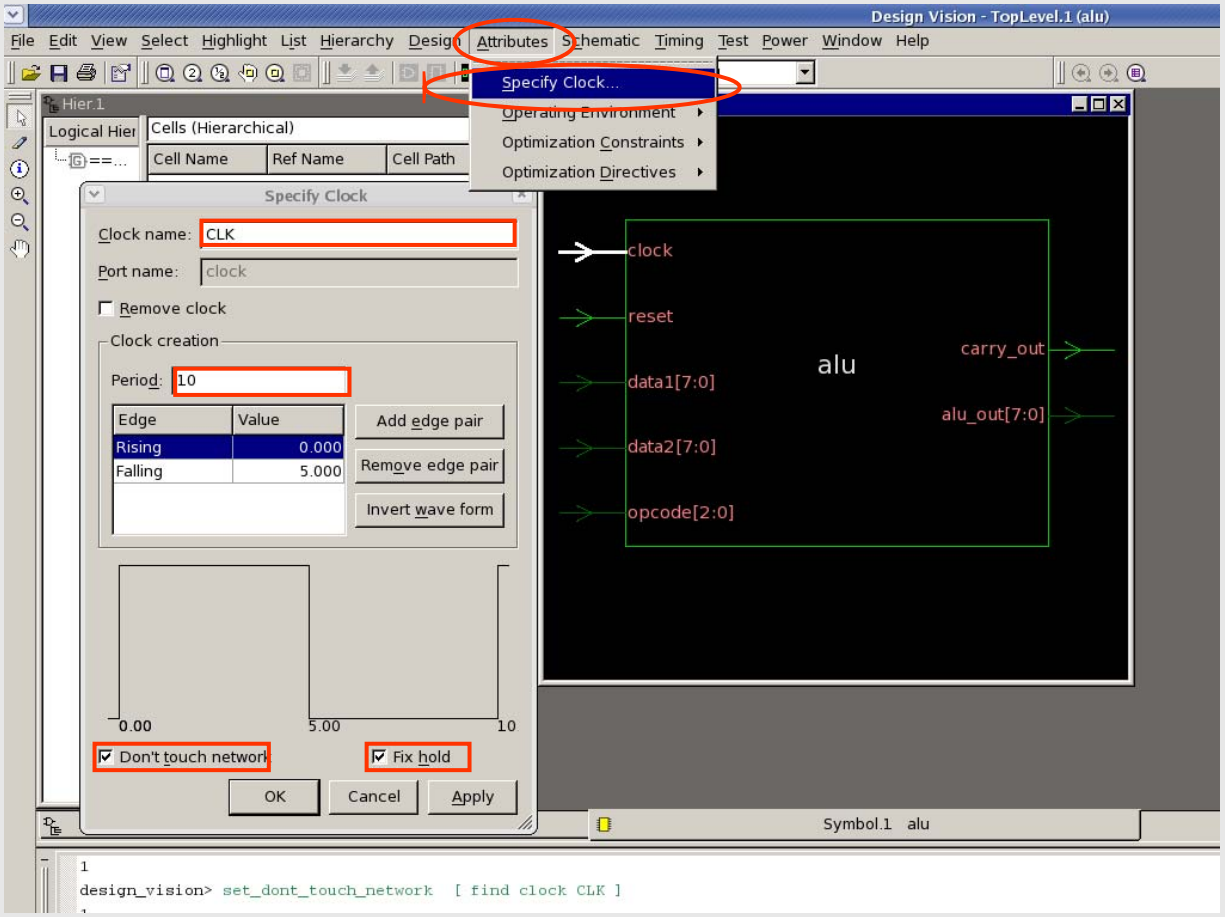


Specify Clock (Period & Waveform)

Command: create_clock -name CLK -period 10 -waveform {0 5} [get_ports clock]

set_dont_touch_network [get_clocks CLK]

set_fix_hold [get_clocks CLK]



Setting Clock Command

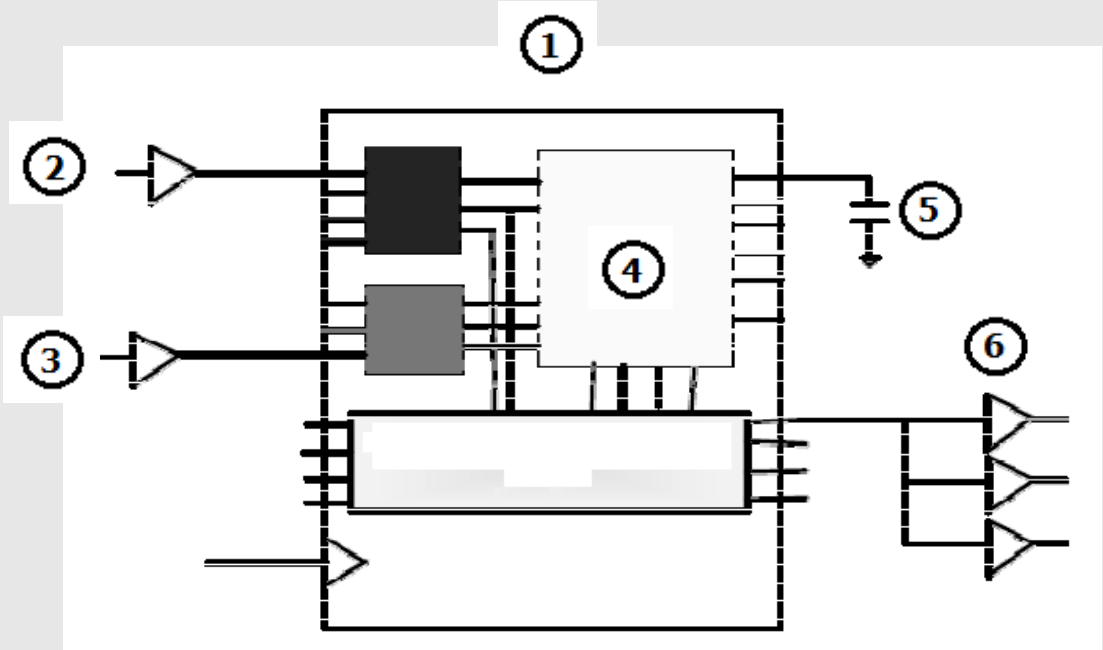
Command:

- set_clock_uncertainty 0.1 [get_clocks CLK]
- set_clock_latency -source 0 [get_clocks CLK]
- set_clock_latency 1 [get_clocks CLK]
- set_input_transition 0.5 [all_inputs]
- set_clock_transition 0.5 [all_clocks]

Design Constraints Setting

Setting Design Environment

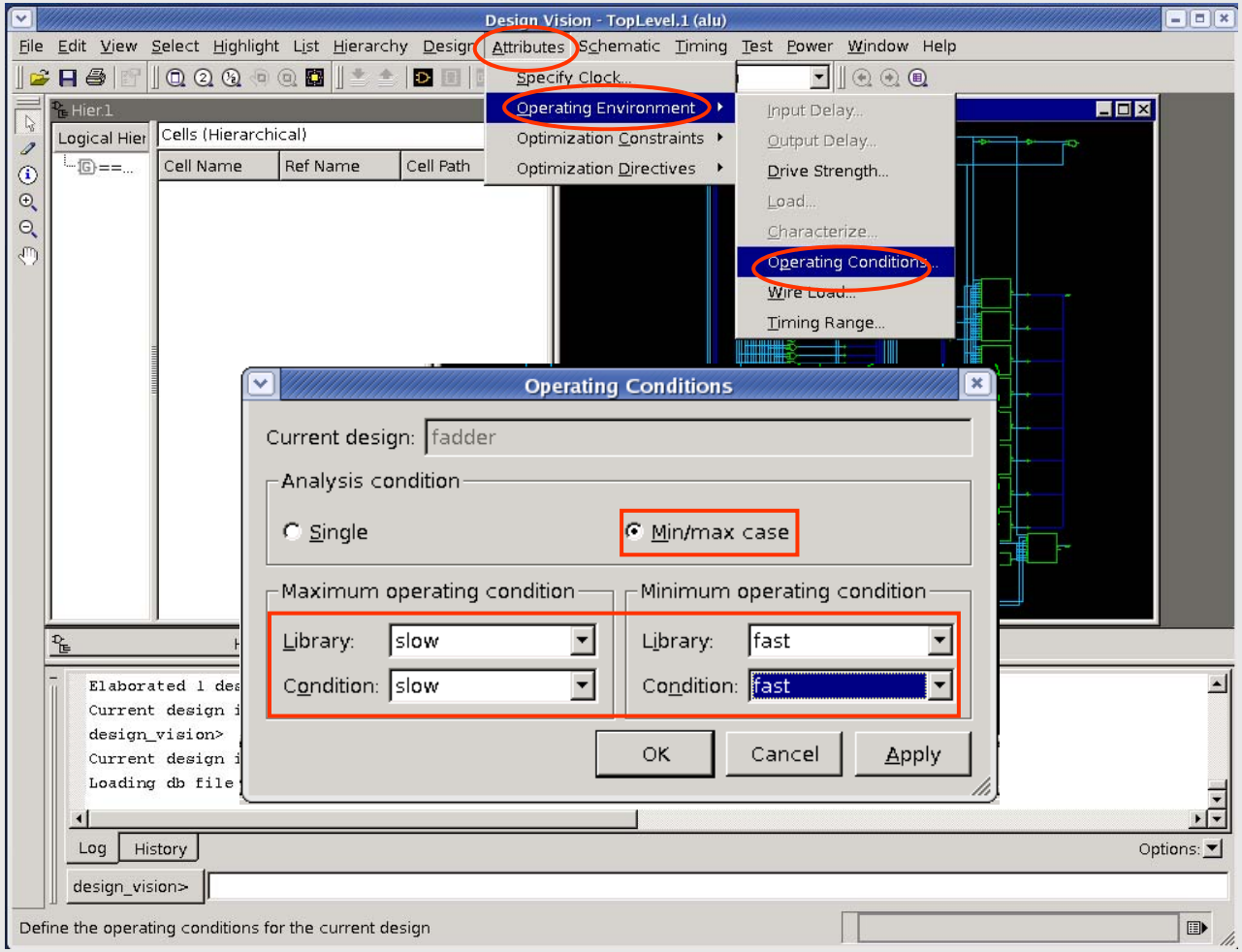
1. set_operating_conditions
2. set_input_delay
3. set_driving_cell
4. set_wire_load_model
5. set_load
6. set_output_delay



Setting Operating Condition

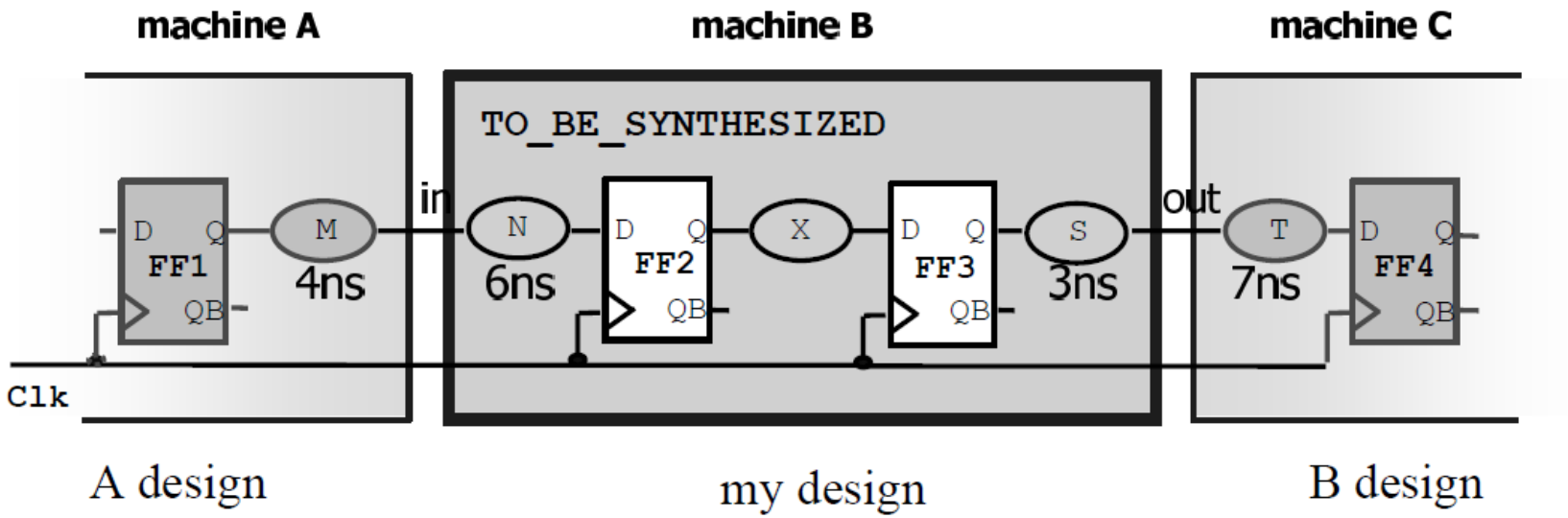
Command:

set_operating_conditions -min_library fast -min fast -max_library slow -max slow



Input Delay and Output Delay (1/3)

- Clock cycle > = $DFF_{clk\text{-}Qdelay}$ + X + DFF_{setup}
- Input Delay = $DFF_{clk\text{-}Qdelay}$ + M
- Output delay = T + DFF_{setup}



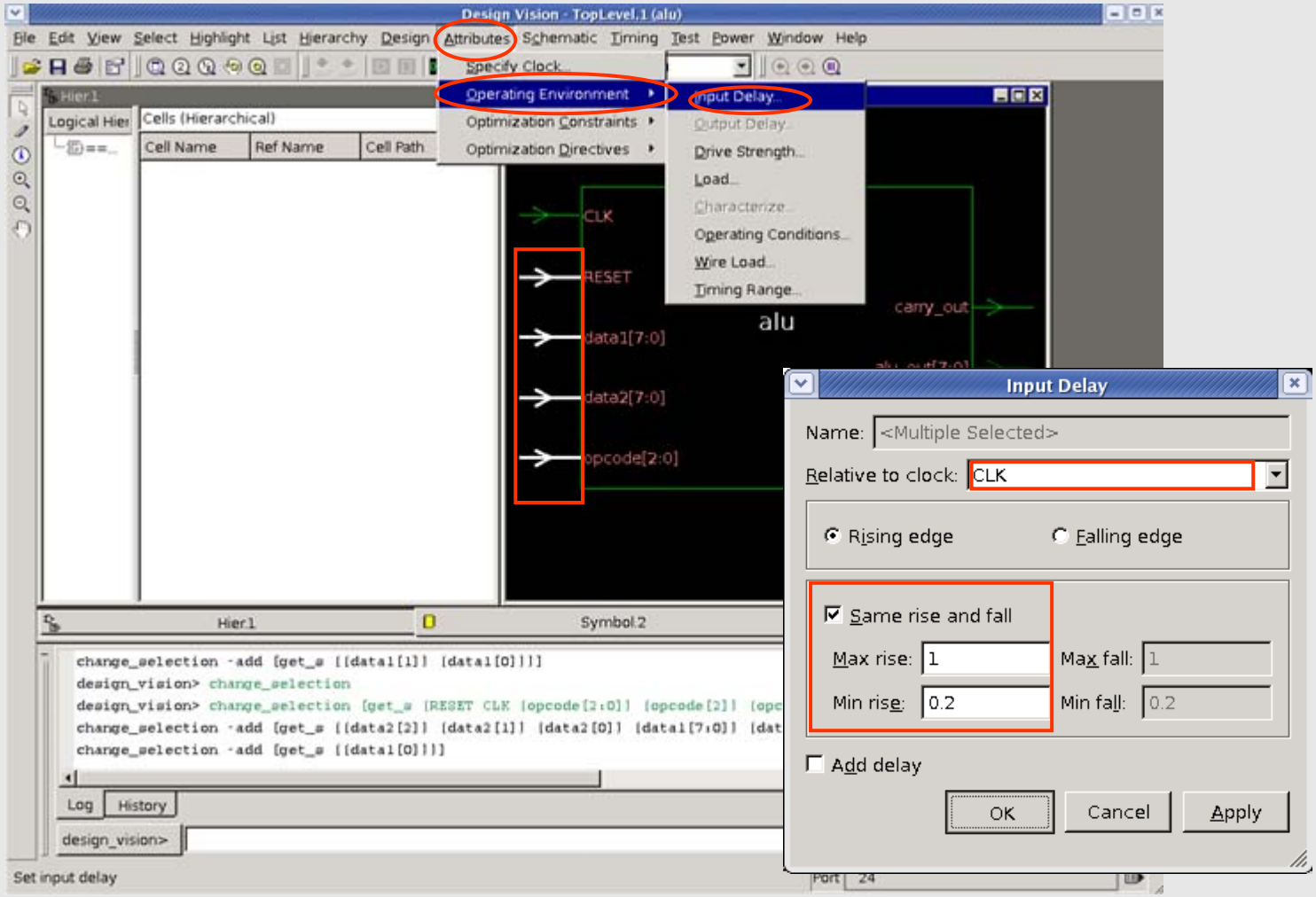

```
design_vision>set_input_delay  -clock#clk -max 4 [get_ports in]
design_vision>set_output_delay -clock#clk -max 7 [get_ports out]
```

Input Delay and Output Delay (2/3)

Command:

set_input_delay -clock CLK -max 1 [remove_from_collection [all_inputs] [get_clocks CLK]]

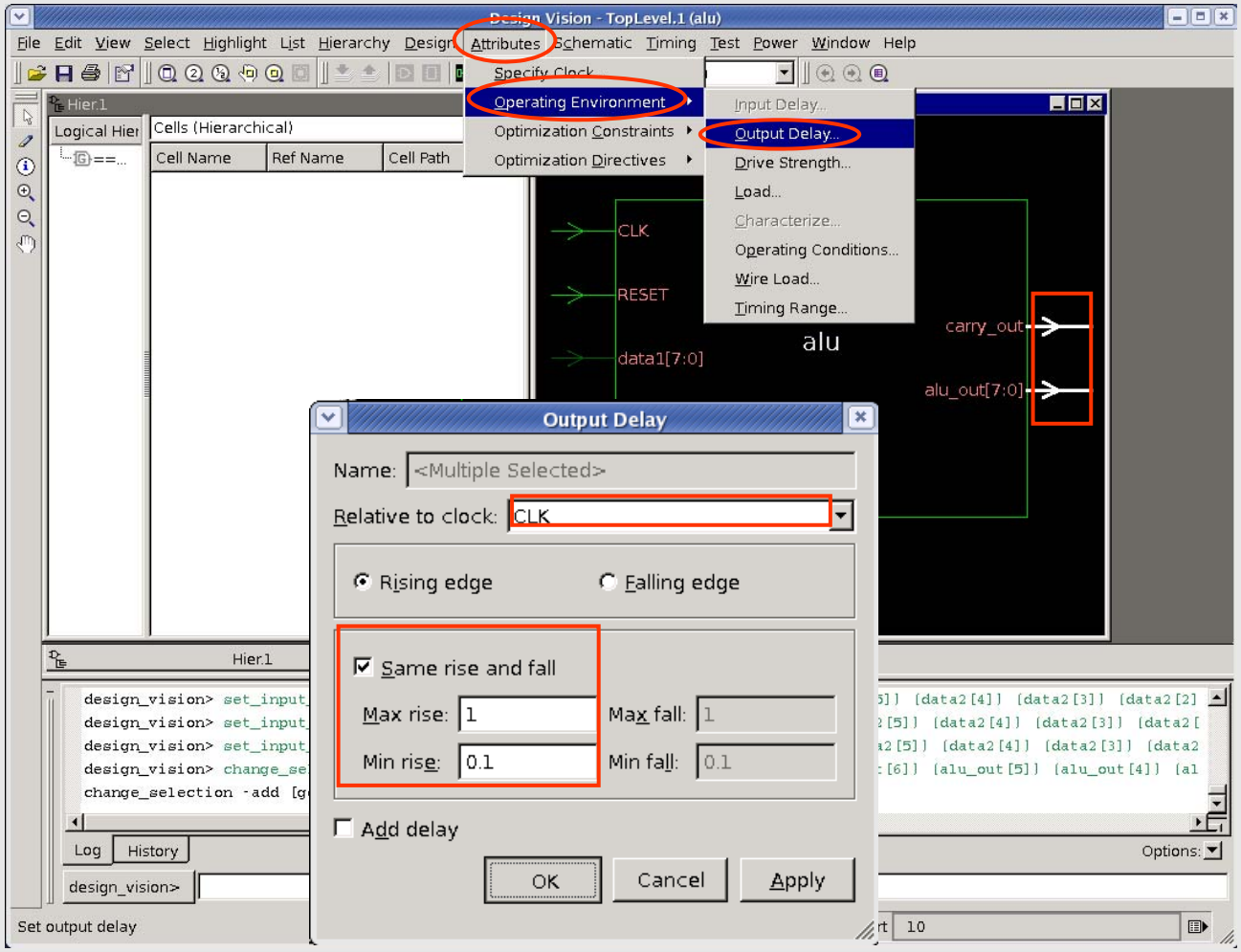
set_input_delay -clock CLK -min 0.2 [remove_from_collection [all_inputs] [get_clocks CLK]]



Input Delay and Output Delay (3/3)

Command:

set_output_delay -clock CLK -max 1 [all_outputs]
set_output_delay -clock CLK -min 0.1 [all_outputs]



Input Drive Strength for Pads

設定Driving Strength時可以直接輸入值，或是指定不同的Cell，一般來說我們會假設驅動input腳位的Cell為D型正反器，而DFF的值為6.60925， Driving Strength的值越小，推動力愈大，如設0則推動力為無限大

Setting input driving strength for clk port

Command:

set_driving_cell -library slow -lib_cell BUFX4 -pin {Y} [get_ports clock]

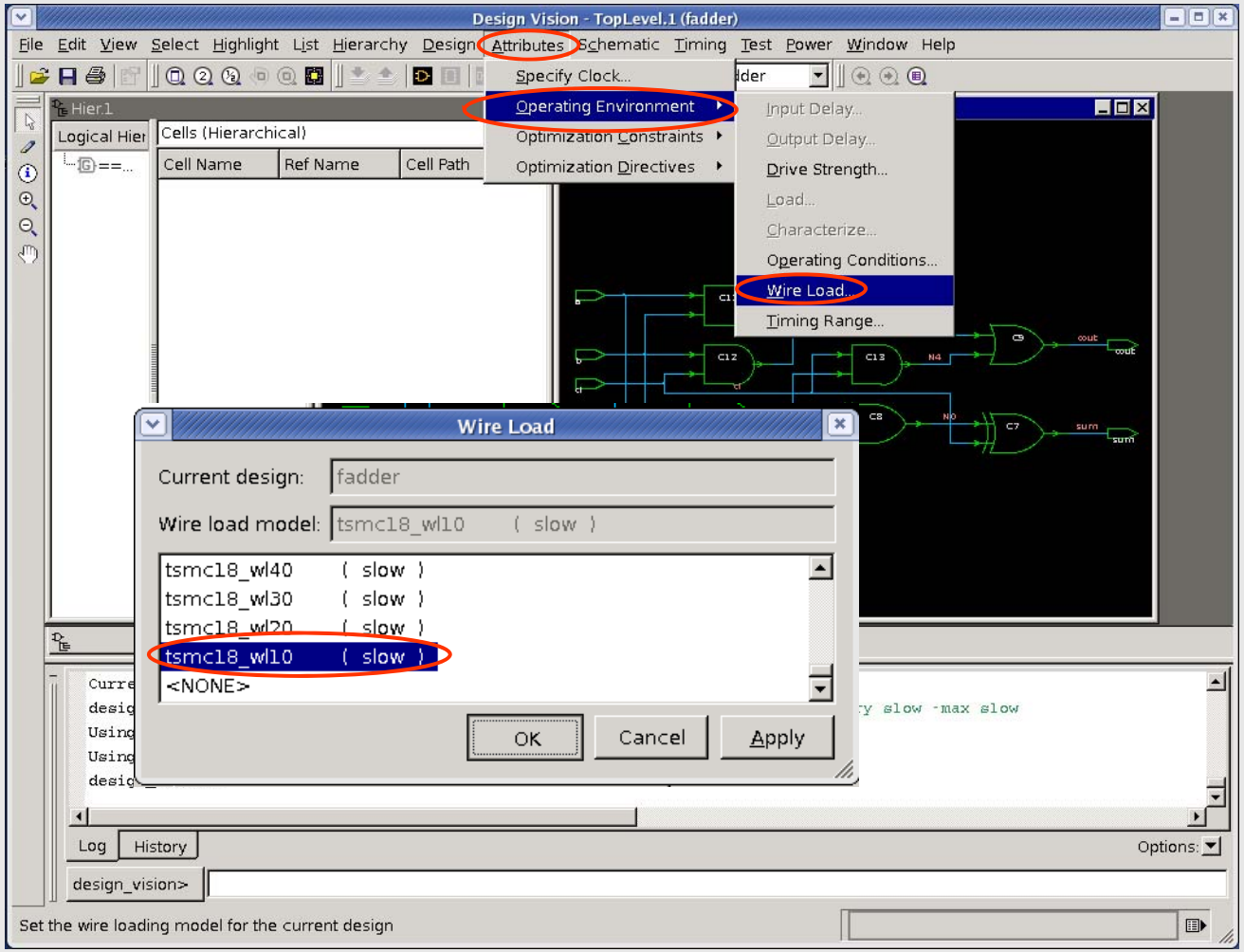
Setting input driving strength for all input port except clk

Command:

set_driving_cell -library slow -lib_cell DFFX1 -pin {Q} [remove_from_collection [all_inputs] [get_ports clock]]

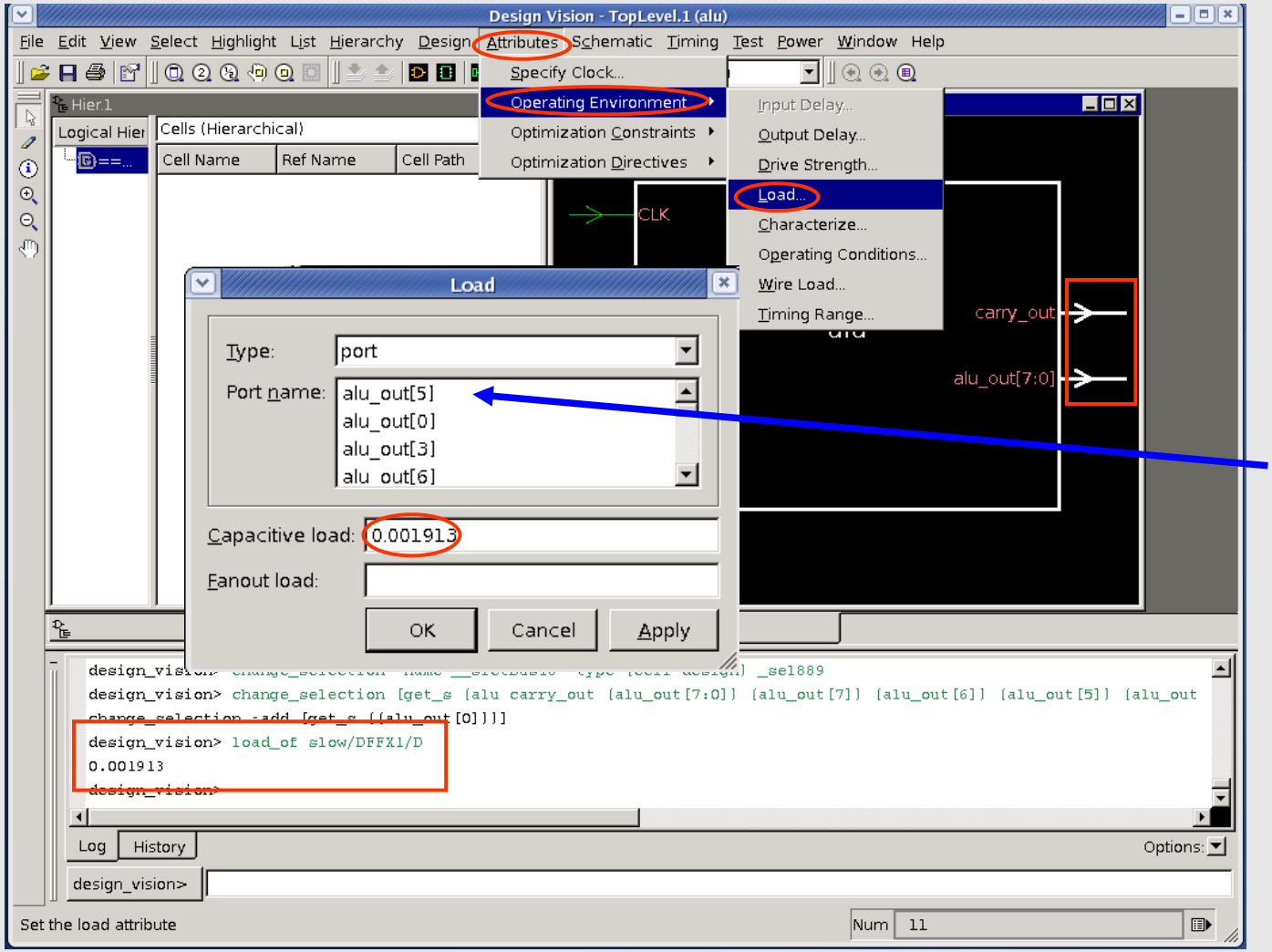
Setting Wire Load Model for Net Delay

Command:
set_wire_load_model -name tsmc18_wl10 -library slow



Output Load for Pad

Command:
set_load 0.001913 [all_outputs]



輸出腳位要一個一個去設定0.001913

Setting Area And Design Rule Constraints

In general, design rule constraints reflect technology-specific constraints that must be met for a design to function correctly.

Setting area constraint

Command: set_max_area 0

→To create a design that has been optimized for the smallest possible size.

→If used **-ignore_tns** option, then the compiler prioritizes **area** above **TNS**.

Setting design rule constraints

Command: set_max_fanout 6 [all_inputs]

→Ensure that the sum of the fanout_load attributes for input pins in the specified design is less than the given value.

Command: set_max_transition 0.3 [all_inputs]

→ Attempts to ensure that the transition time for a net is less than the specified value.

→ By default, no restriction is placed on the transition time.

Check & uniquify Design

在設定完後要做check design

Check design

Command:

check_design -multiple_designs

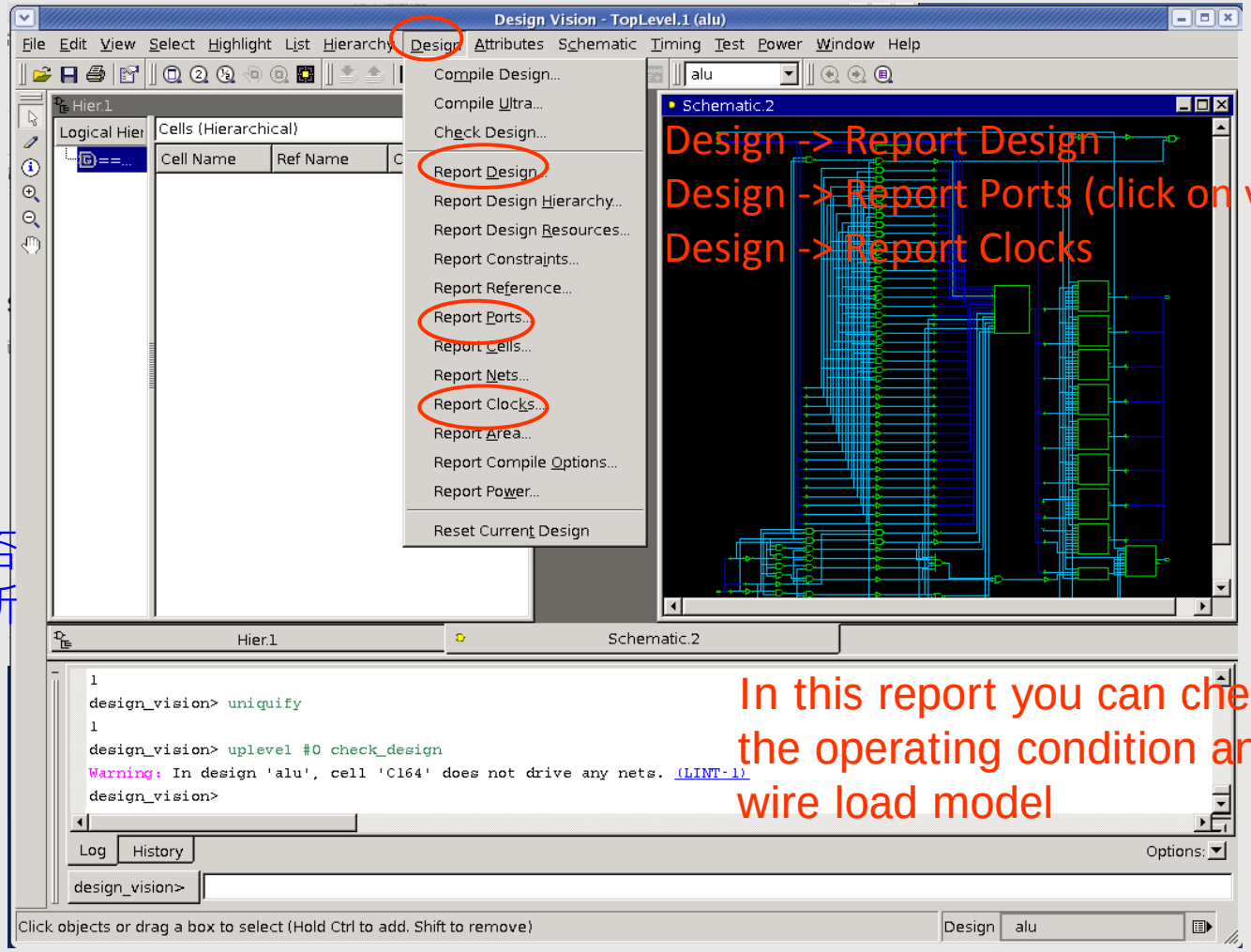
Uniquify the design

Command:

uniquify

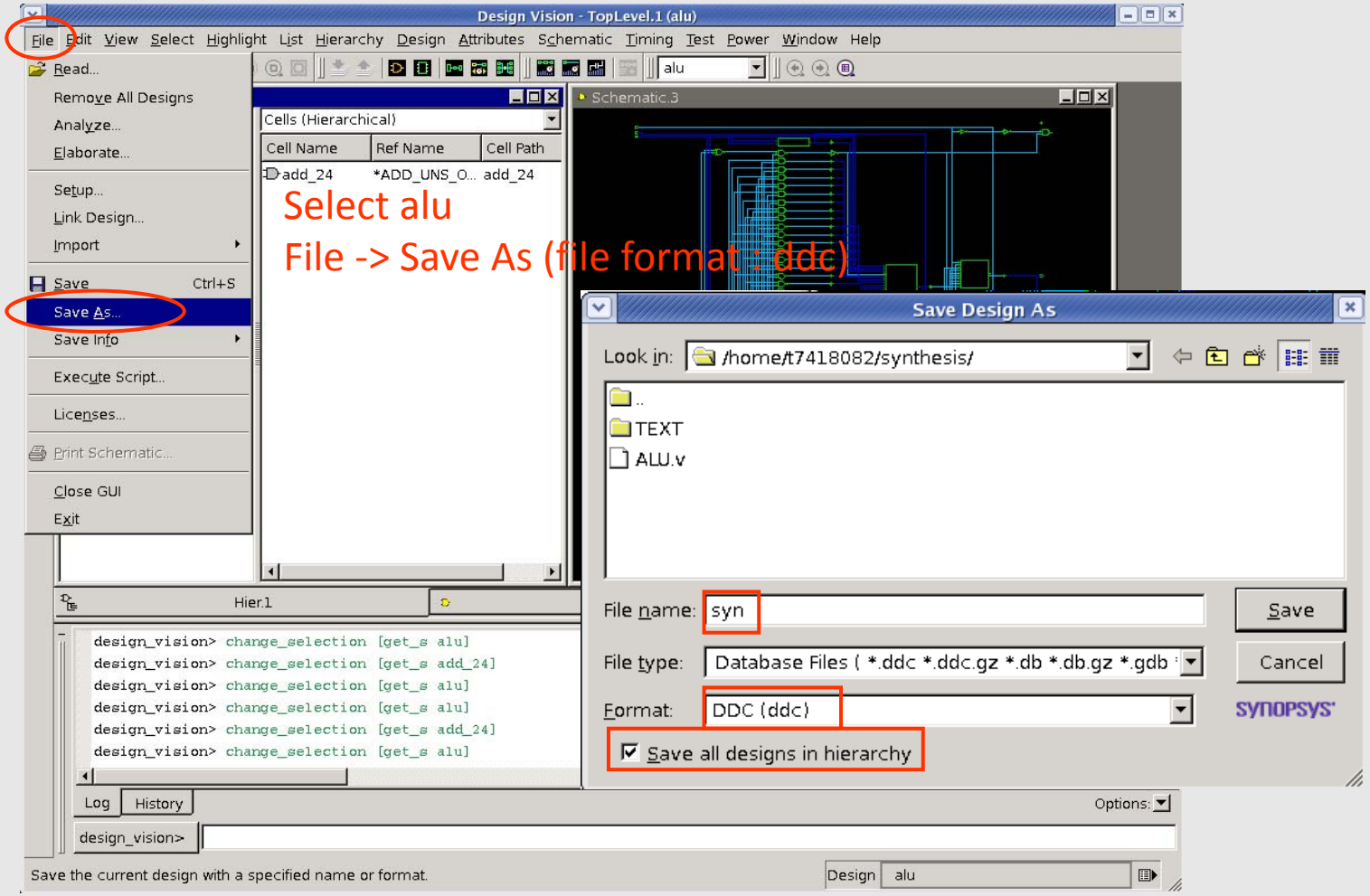
Reports Before Compiler

Report Ports 可以確認是否正確設定了所有的Output Loading為0.001913

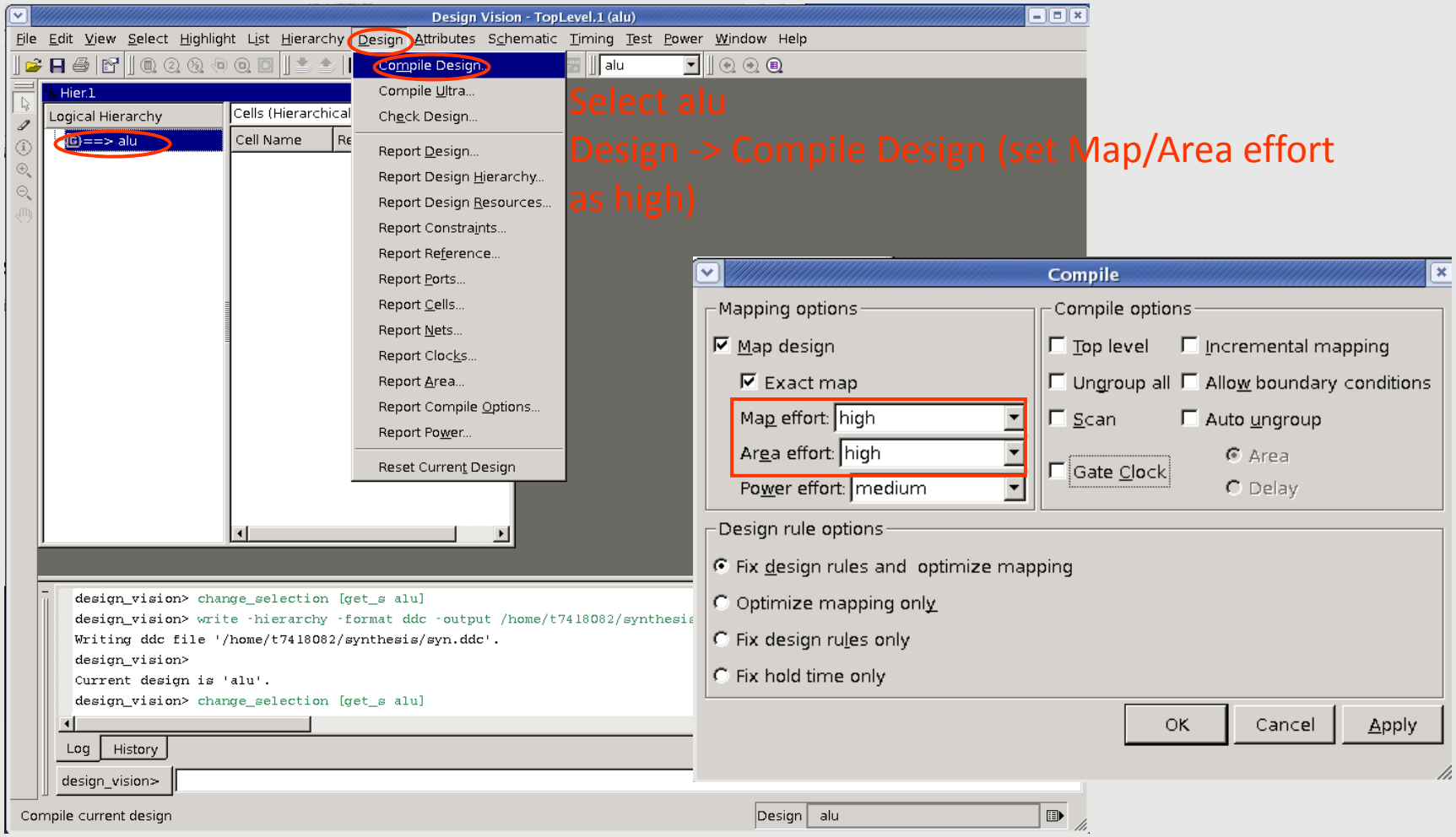


In this report you can check the operating condition and wire load model

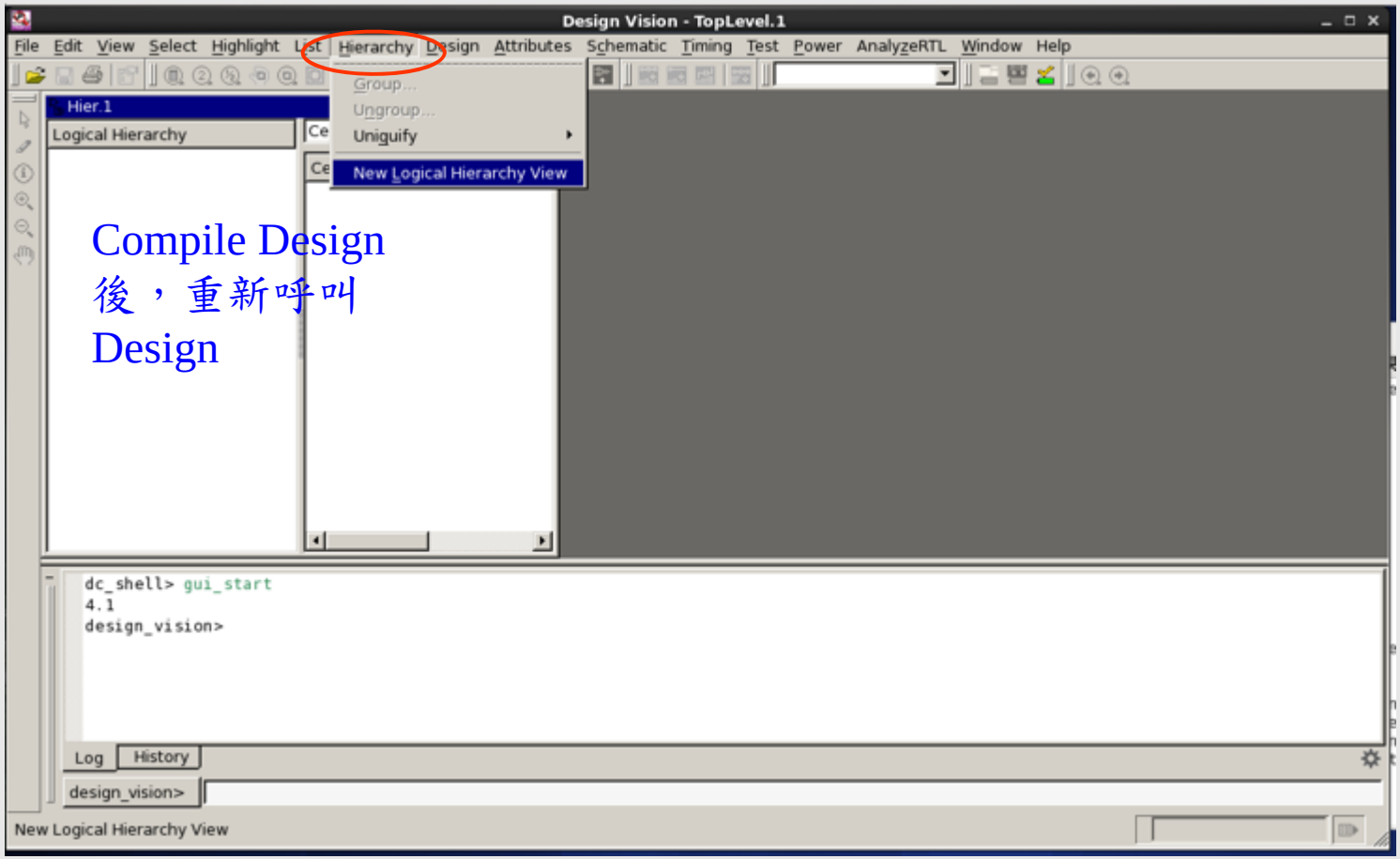
Save Design as .DDC File



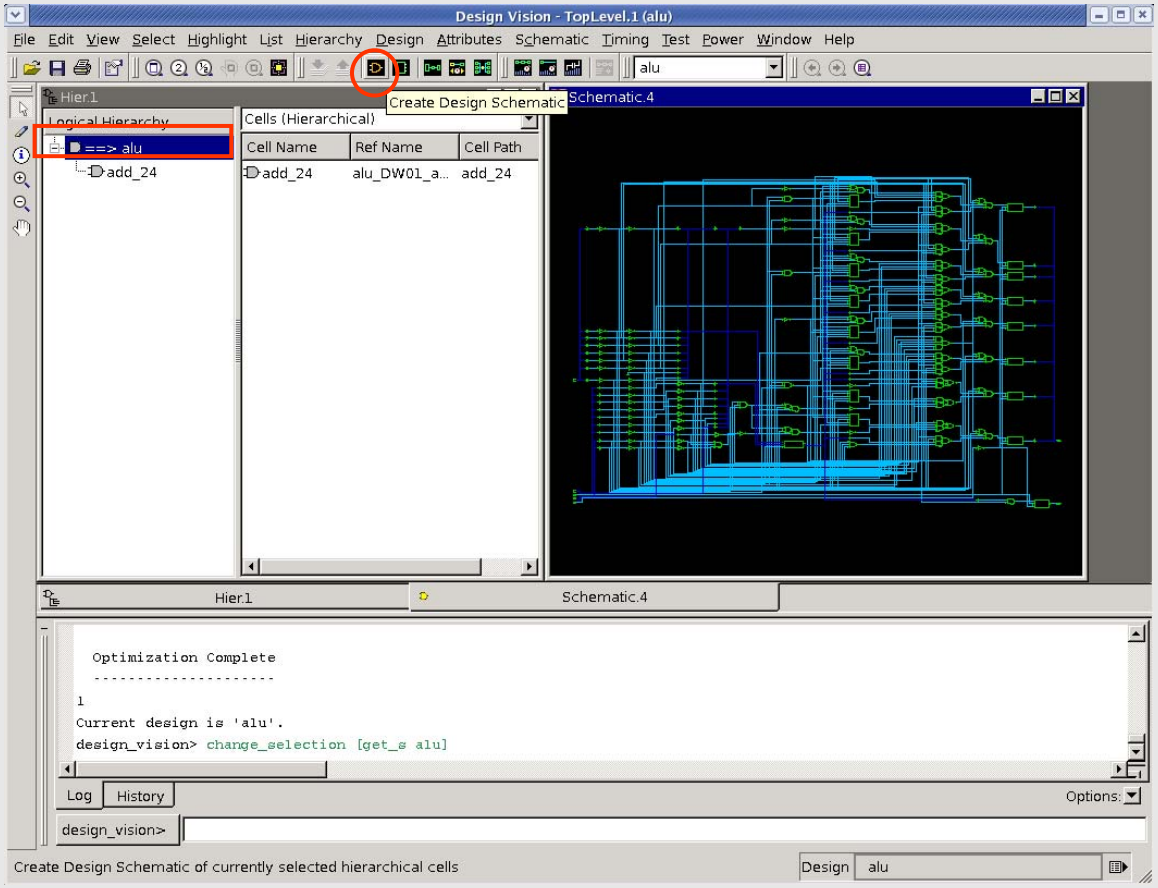
Compile Design



Compile Design



Explore The Schematic View

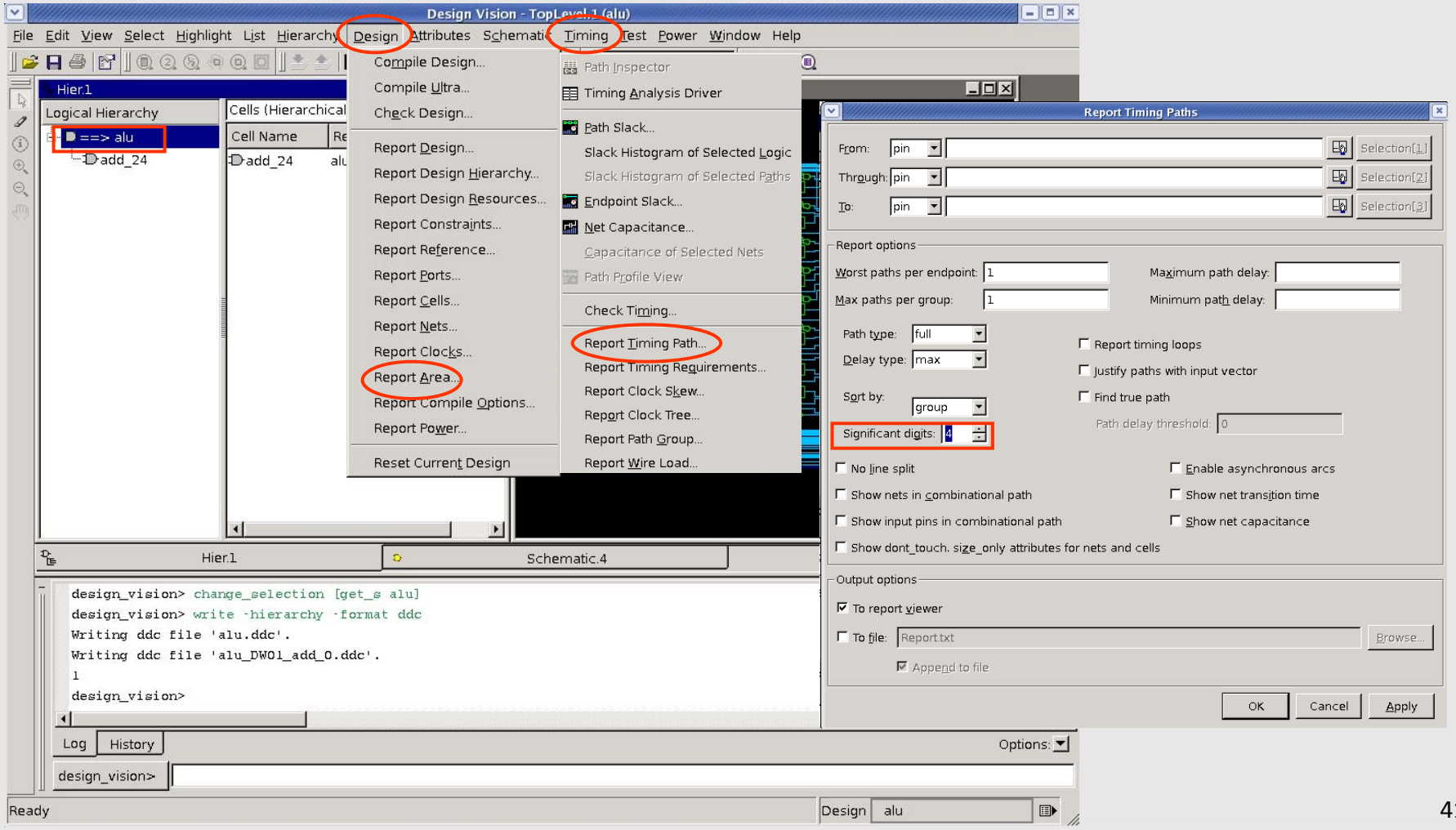


Select alu and save again

Report Area & Timing

Reports Area (μm^2) : **Total Area**

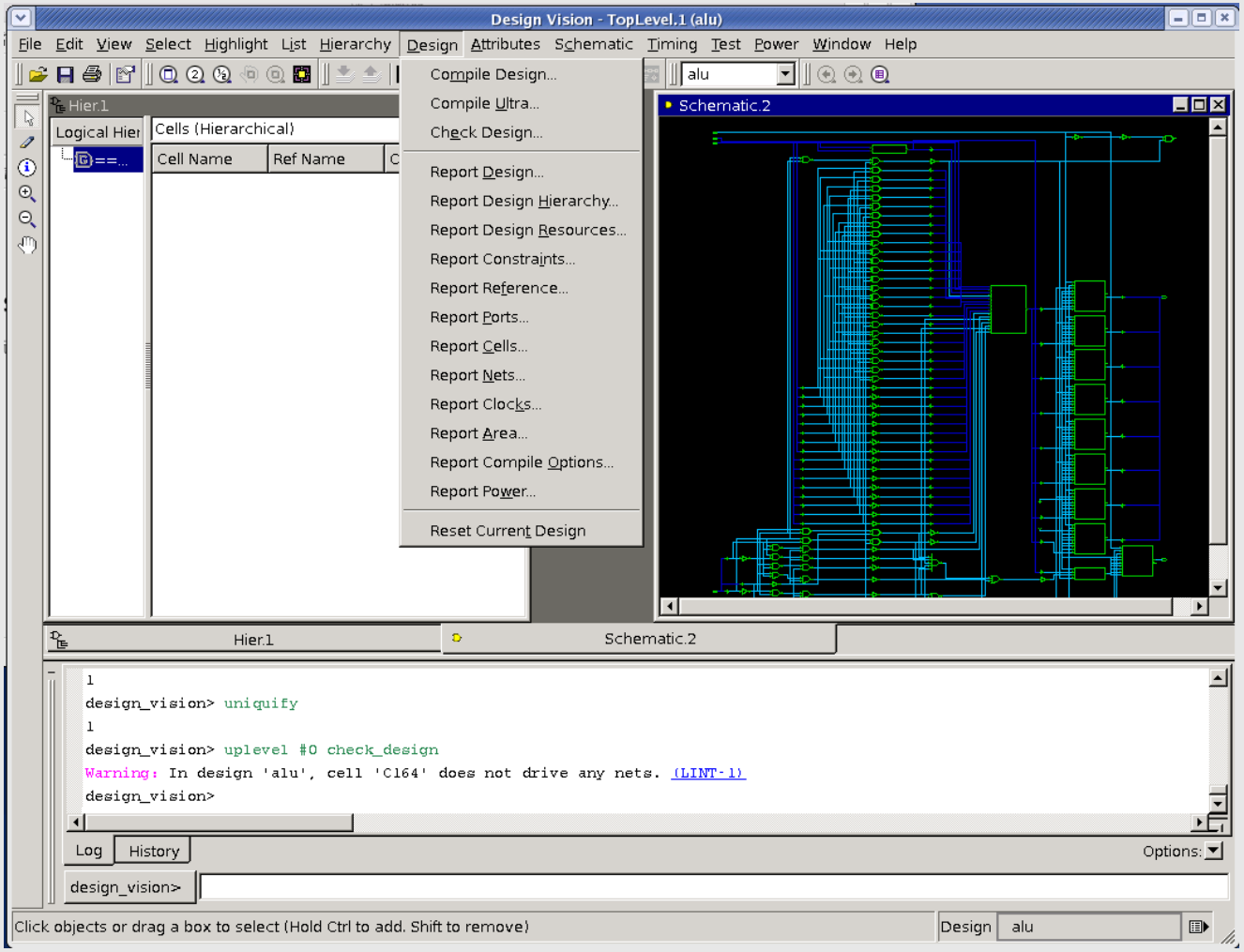
Gate Count : **Total Area / 10**



Report Area & Timing (Homework)

- Use "Design/Report Area & Timing/Report Timing"
- Area (total cell area)=_____ μm^2
- Timing (data arrival time)=_____ ns
- Slack =_____ ns

Reports After Compiler (Homework)



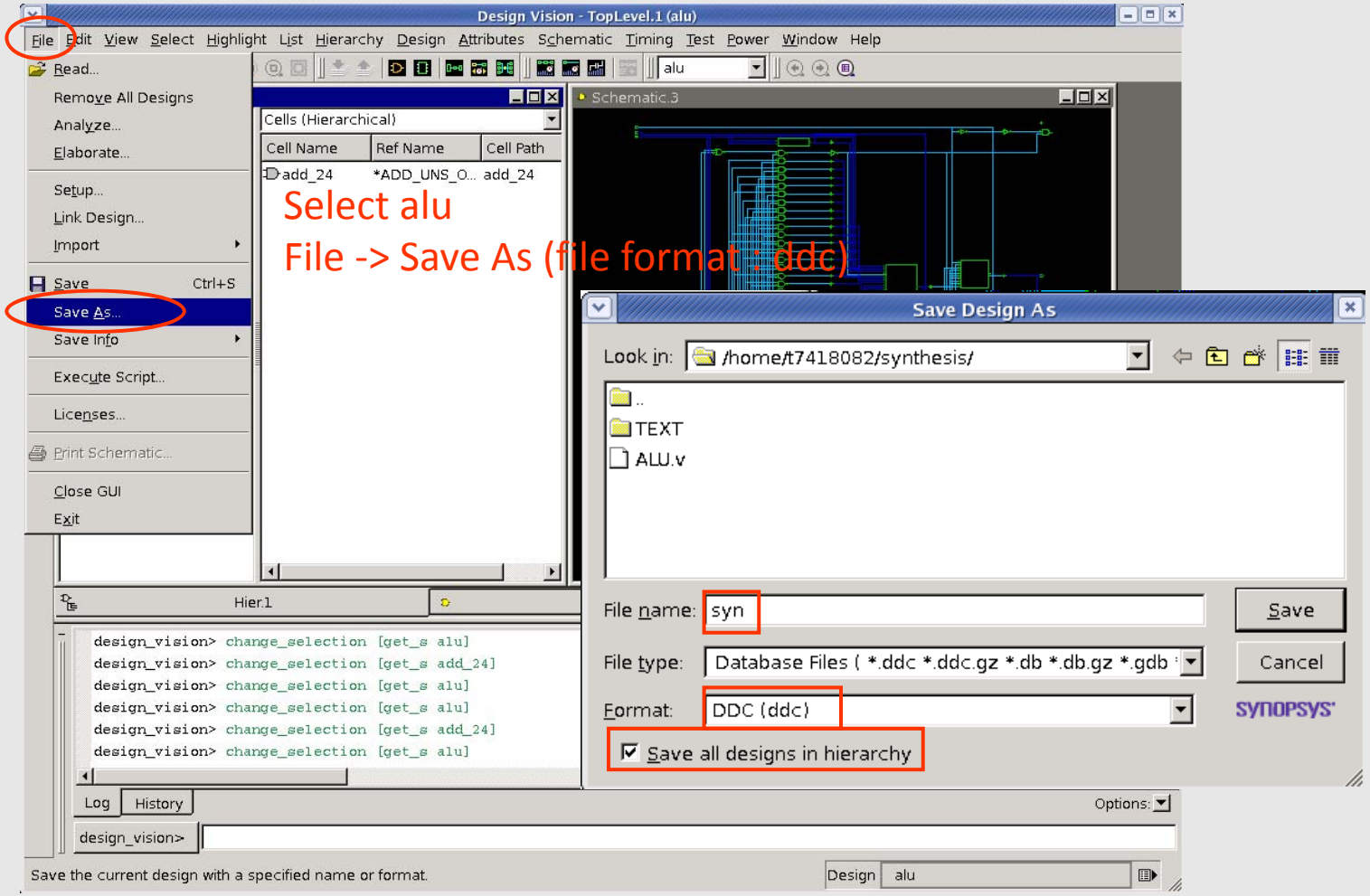
Reports After Compiler (Homework)

Report Design
Report Design Hierarchy
Report Constraints
Report Reference
Report Ports
Report Cells
Report Nets
Report Clocks
Report Compile Options
Report Power

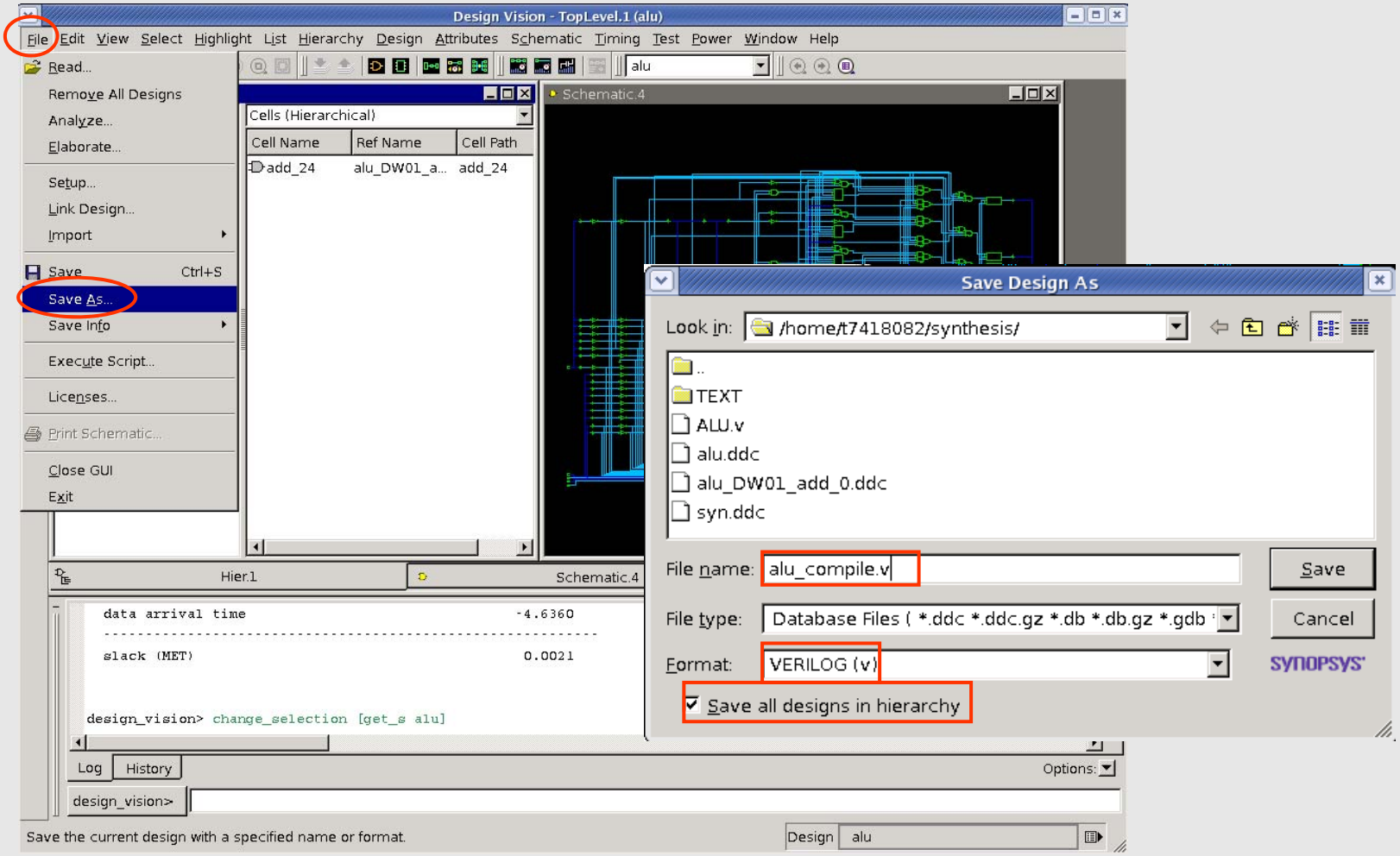
Compare the Difference between alu.v and alu_compile.v (Homework)

- Using vi editor to compare the difference between alu.v and alu_compile.v
- What is the function of logic synthesis?
- Please use Cell-based Design Flow to describe the difference between before and after logic synthesis

Save Design as .DDC File (Design Compiler Database)



Save Design as .V File : Netlist



Write SDF File for Pre-Sim

The file includes information of delay of Gate-Level circuit.

Command in design_vision:

```
write_sdf -version 1.0 -context verilog alu_syn.sdf

write_sdc -version 1.7 CHIP.sdc
```

Pre-Layout Simulation

以剛剛產生出的**Netlist**檔進行Gate-Level Simulation也常稱作Pre-Sim用以驗證確認我們合成出來的邏輯電路是否符合System Spec.所要求的功能

於終端機(Terminal)輸入:

ncverilog test.v alu_compile.v tsmc18.v +access+r

Thanks for your attention!