Logic Synthesis With Design Compiler

ADVISOR：YU-CHENG FAN

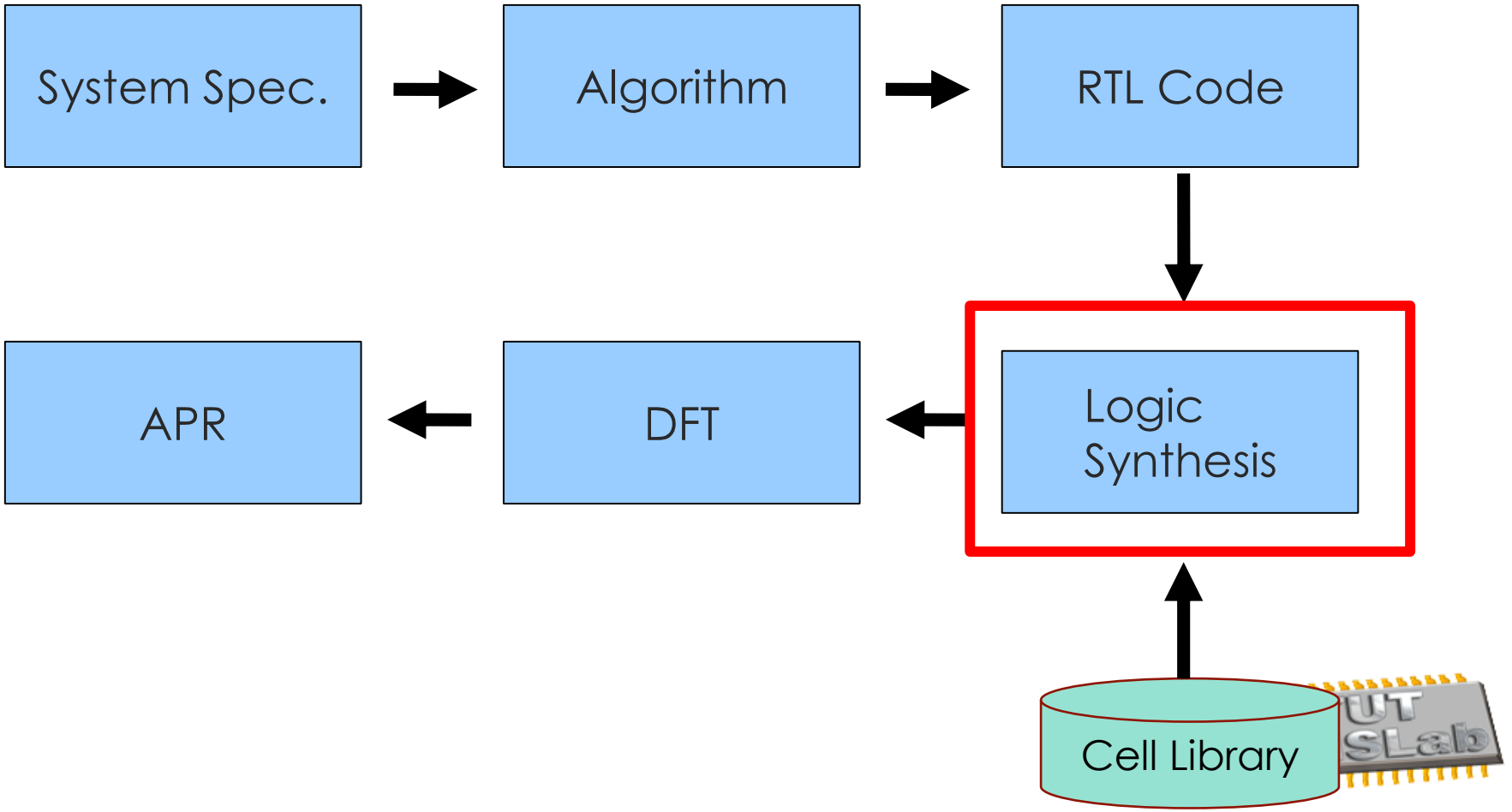
PRESENTER：ZHENG-AN YANG

OUTLINE

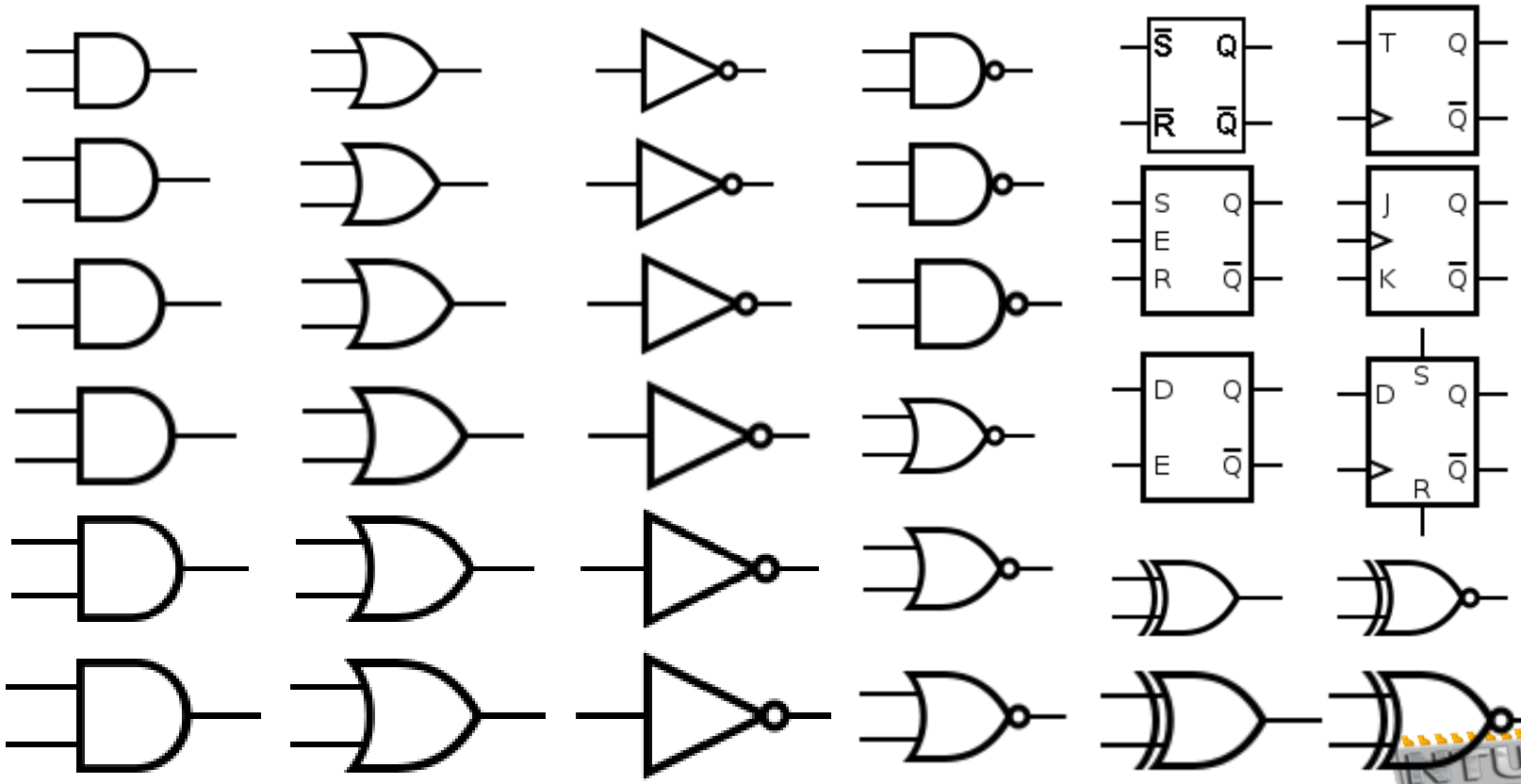
- Introduction
- Synthesis Object
- Synthesis GUI-Design Vision
- Setting Design Constraints
- Setting Design Environment
- Reports and Save file

INTRODUCTION

Cell-Based Design flow



Cell Library (Logic Gate v.s. Lego)

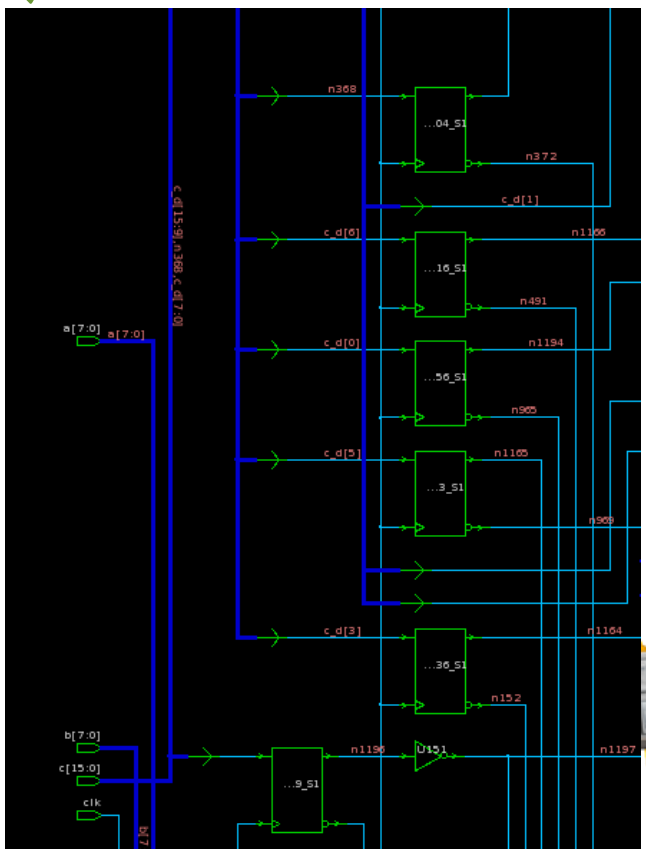


Logic Synthesis

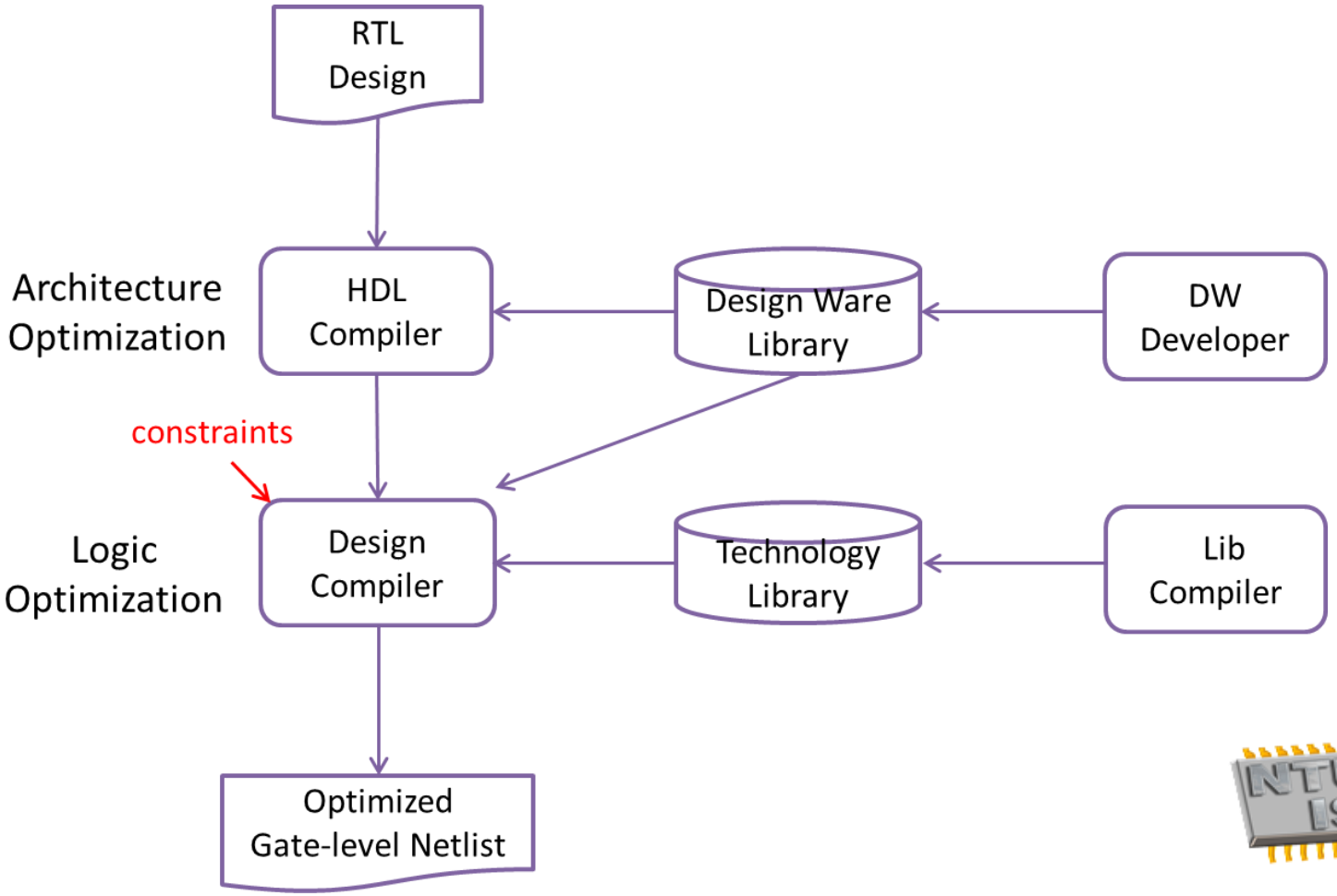
Behavior code

```
module top(clk,out,a,b,c);
output [15:0]  out;
input  [7:0]  a,b;
input  [15:0]  c;
input          clk;
reg    [15:0]  mul_out;
reg    [15:0]  out;
reg    [15:0]  c_d;

always @(posedge clk) mul_out=a*b;
always @(posedge clk) c_d=c;
always @(posedge clk) out=mul_out/c_d;

endmodule
```

Gate level code

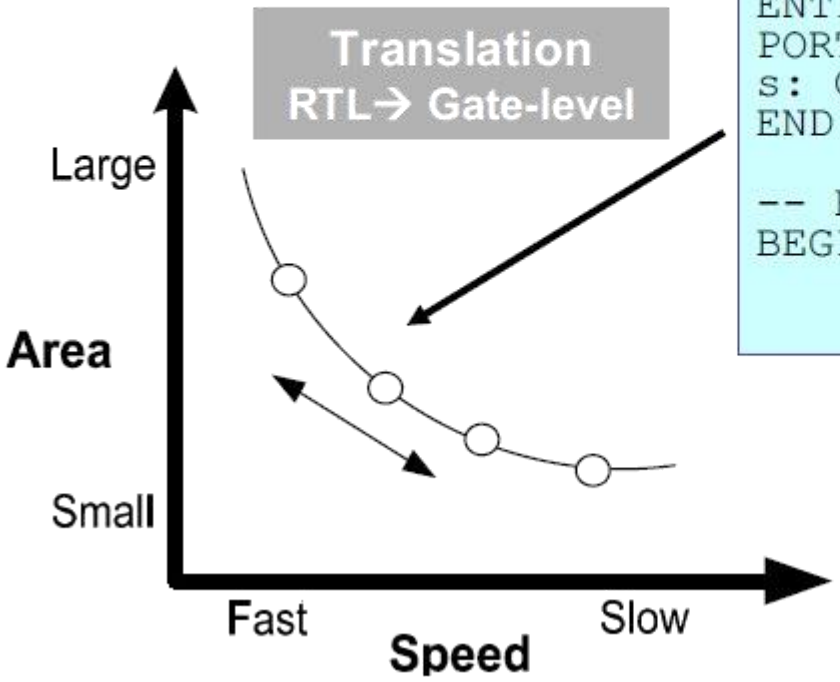


Logic Synthesis Overview



Synthesis is Constraint Driven

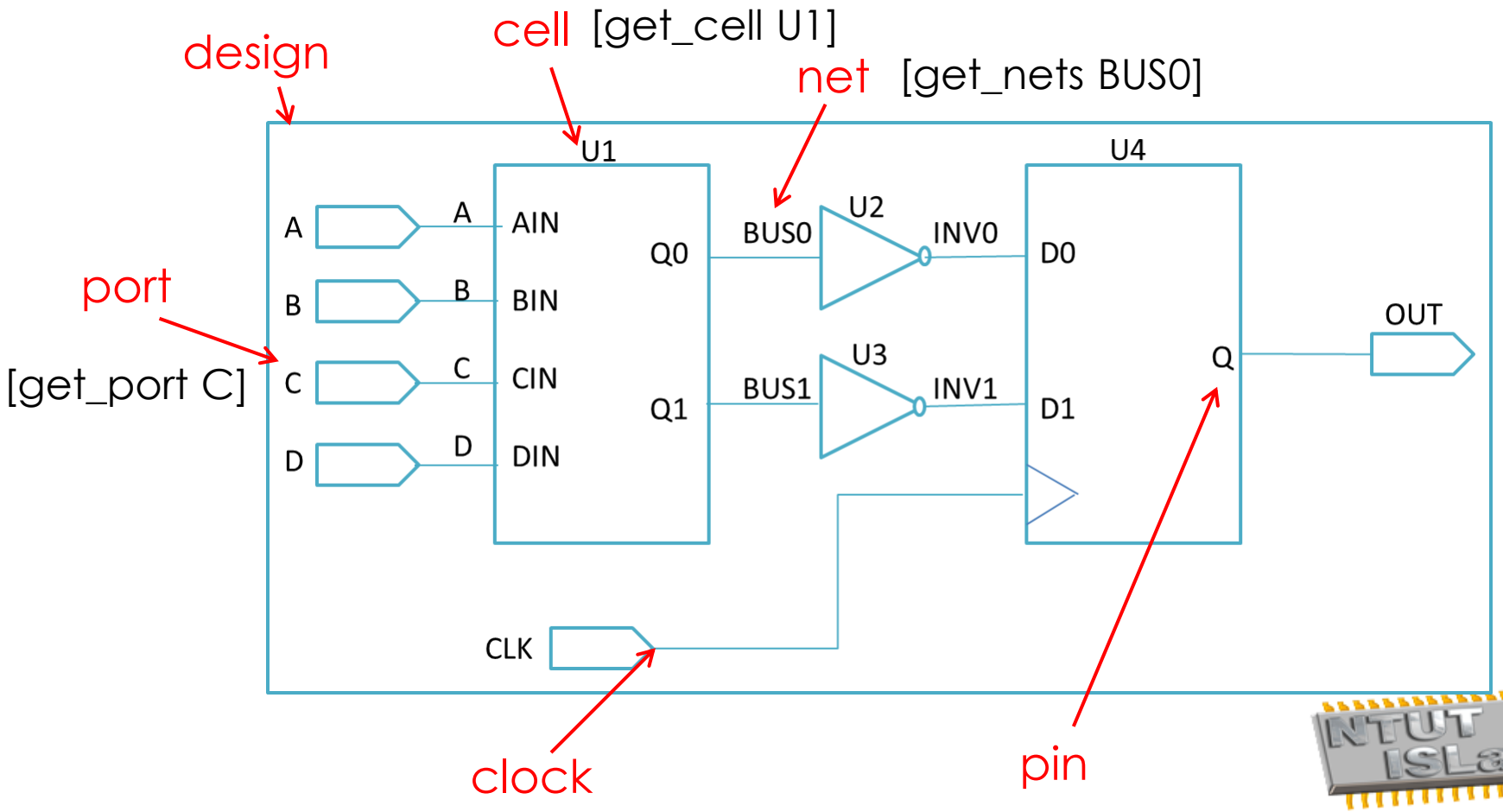
- Logic Synthesizer first translates RTL design to an intermediate gate-level design , then optimize according to the area and timing constraint.




```
--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;
```

SYNTHESIS OBJECT

Design Objects (1/2)



Design Objects (2/2)

- Design ：A circuit that performs one or more logical functions.
- Cell：An instantiation of a design within another design.
- Port：The input or output of a design.
- Pin : The input or output of a cell.
- Net ：The wire that connects ports to pints and/or pints to other each other.
- Clock ：Waveform applied to a port or pin identified as a clock source.

SYNTHESIS GUI-DESIGN VISION

Environment Setup (1/3)

- Set the environmental files

  Open Terminal

  Key in Key in “ ls –a .cs* ”

  “cp /home/standard/Environment_Setup_File/cshrc .cshrc”

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

  Key in “ source .cshrc”

Environment Setup (2/3)

t107368062@islabx5:~/Desktop

檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
[t107368062@islabx5 ~/Desktop]$ ls -a .cs*
.cshrc
[t107368062@islabx5 ~/Desktop]$ cp /home/standard/Environment_Setup_File/cshrc .
cshrc
[t107368062@islabx5 ~/Desktop]$ source .cshrc
-- Now source Cadence NC-Verilog --
-- Now source Synopsys Design Compiler --
Platform = amd64
-- Now source Synopsys Formality --
-- Now source Mentor MBISTArchitect --
-- Now source Synopsys TetraMAX --
-- Now source Cadence INNOVUS --
-- Now source Mentor Calibre --
-- Now source Synopsys PrimeTime --
Platform = amd64
-- Now source Synopsys Nanosim --
-- Now source Synopsys 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.                          #
#############################################################
[t107368062@islabx5 ~/Desktop]$
```

ls –a .cs* (*後面有空格)

cp /home/standard/Environment_Setup_File/cshrc .cshrc

∧ 空格 ∧ 空格

source .cshrc

∧ 空格

代表終端機(Terminal)，成功開啟

Environment Setup (3/3)

Create a New Folder for Synthesis

Ex:

1. mkdir Synthesis (terminal command)
2. cd Synthesis (terminal command)

Copy File for Environment Setting

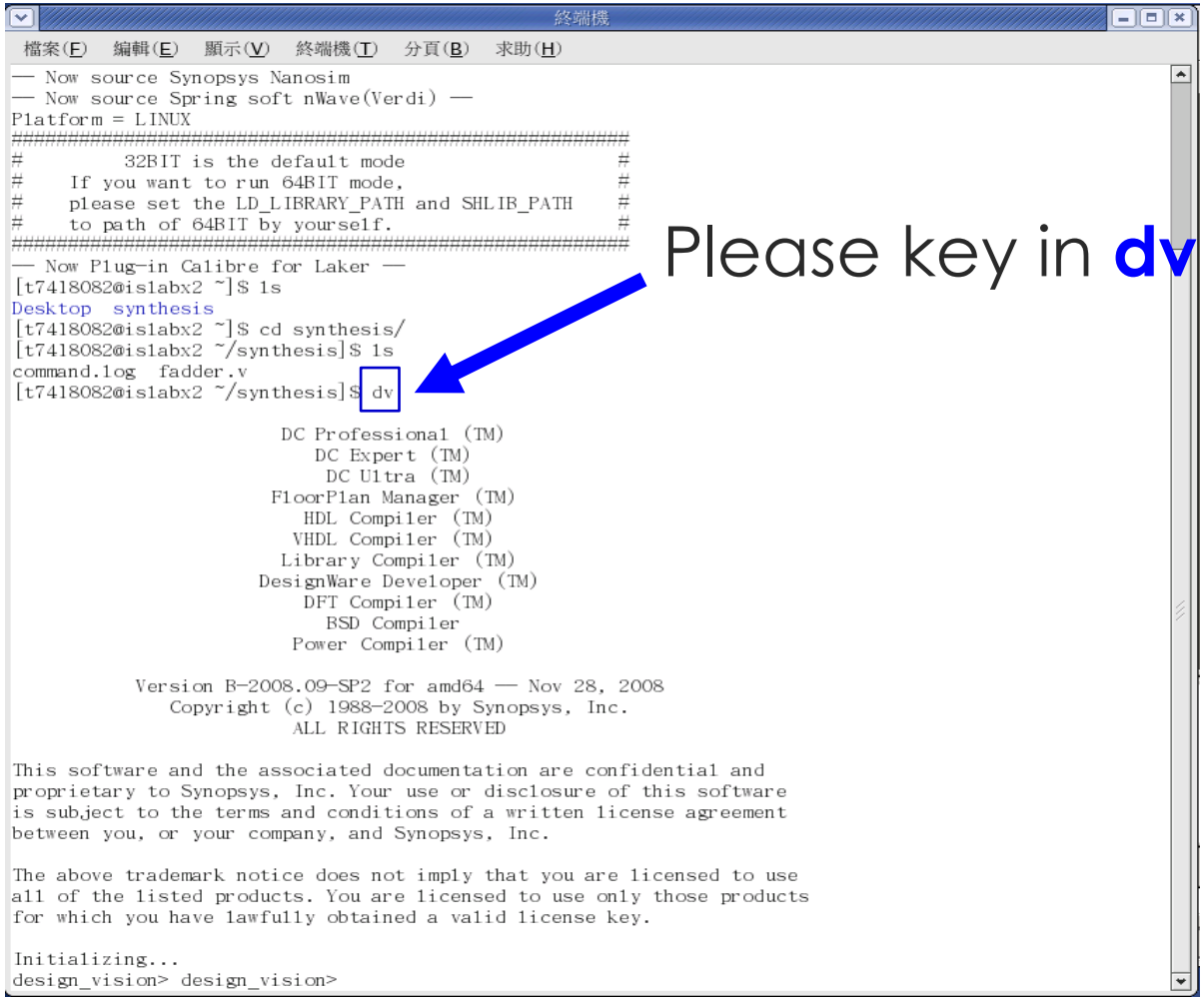
Δ 代表空格

cpΔ
/home/standard/Environment_Setup_File/synthesis_setup_for_18/synopsys_dc.setup ΔΔ.synopsys_dc.setup

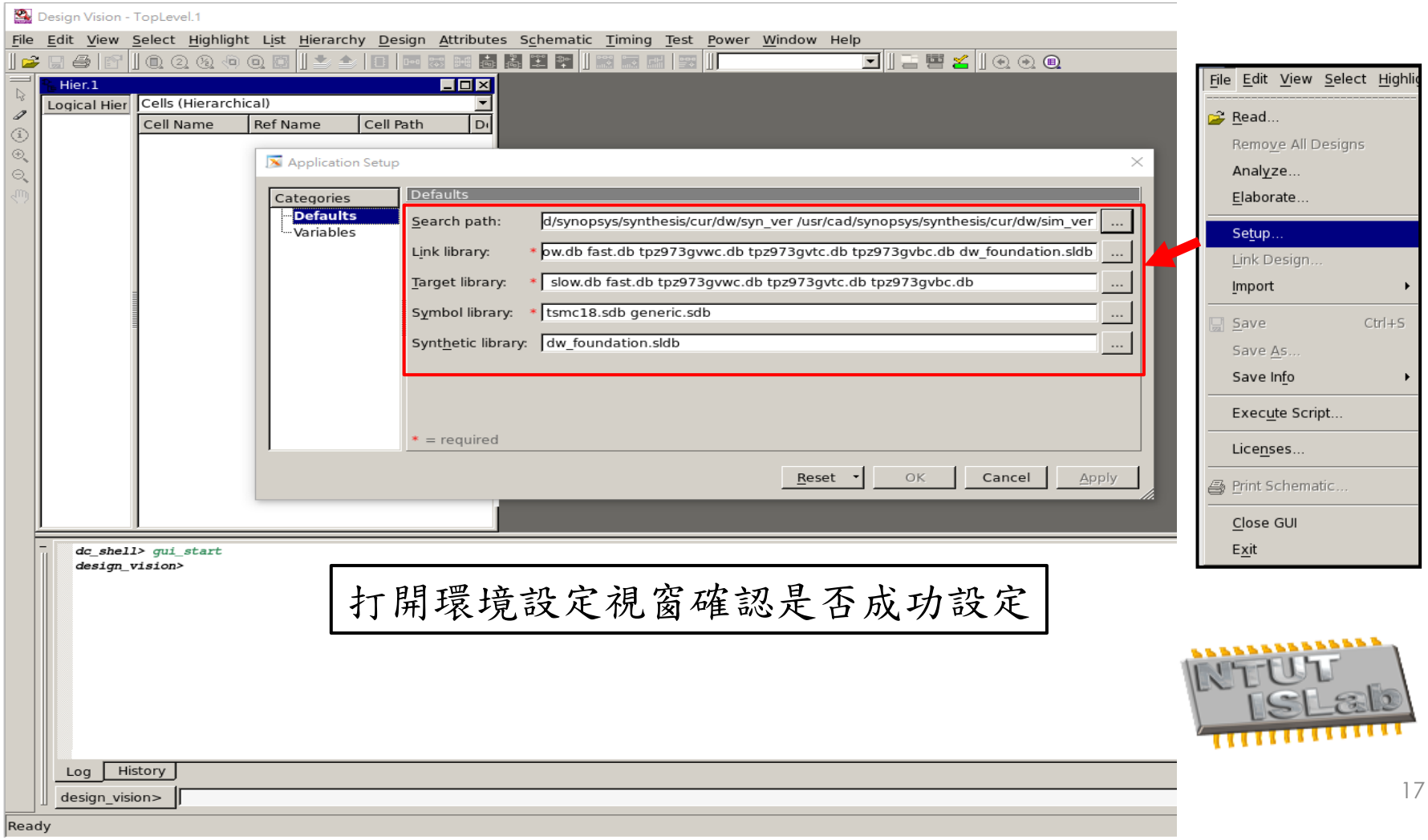
Copy File for Synthesis

cpΔΔ/home/standard/vlsi_2022/synthesis/* .Δ

Open Design Vision

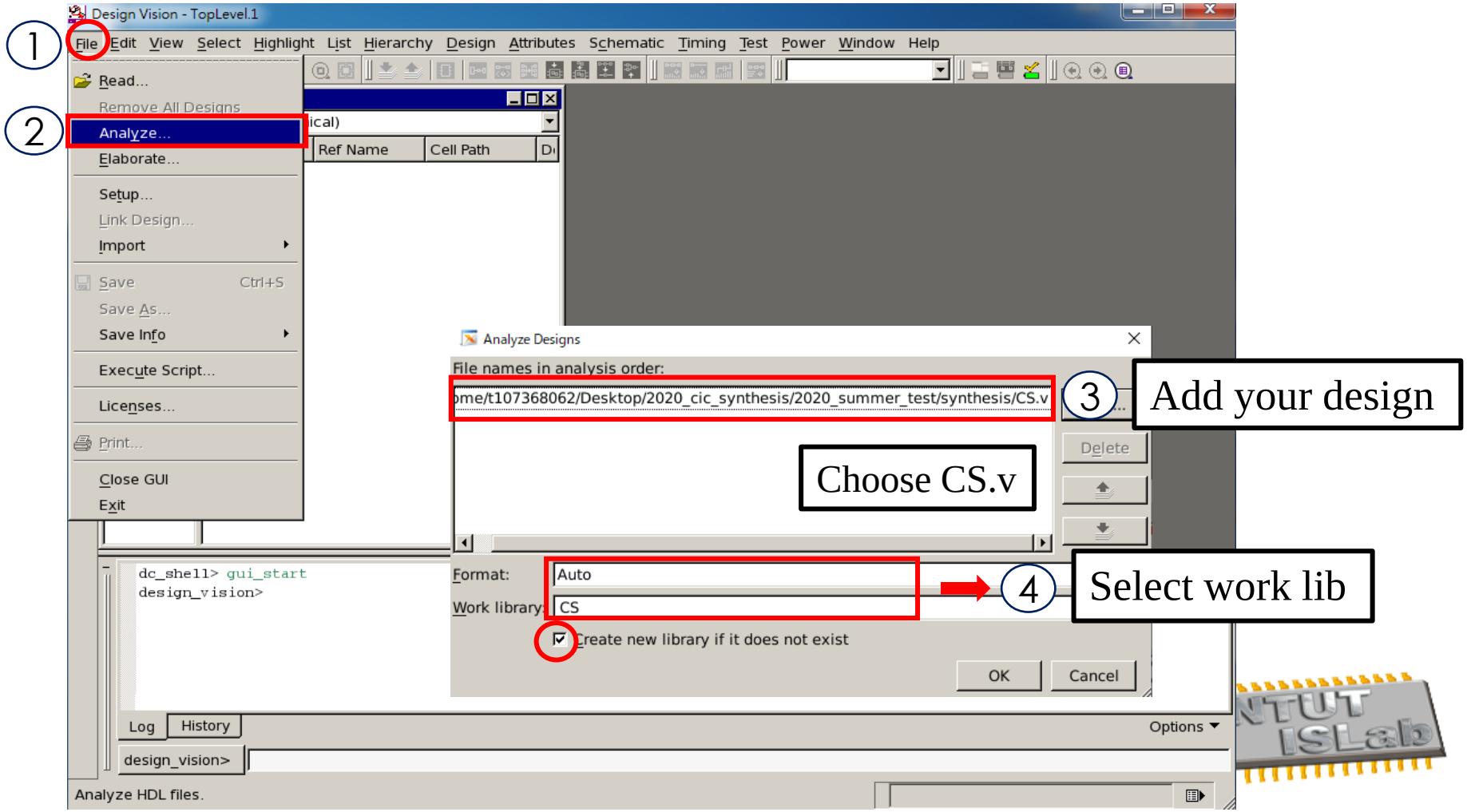


Environment Setup Check



打開環境設定視窗確認是否成功設定

Analysis the Design(1/2)



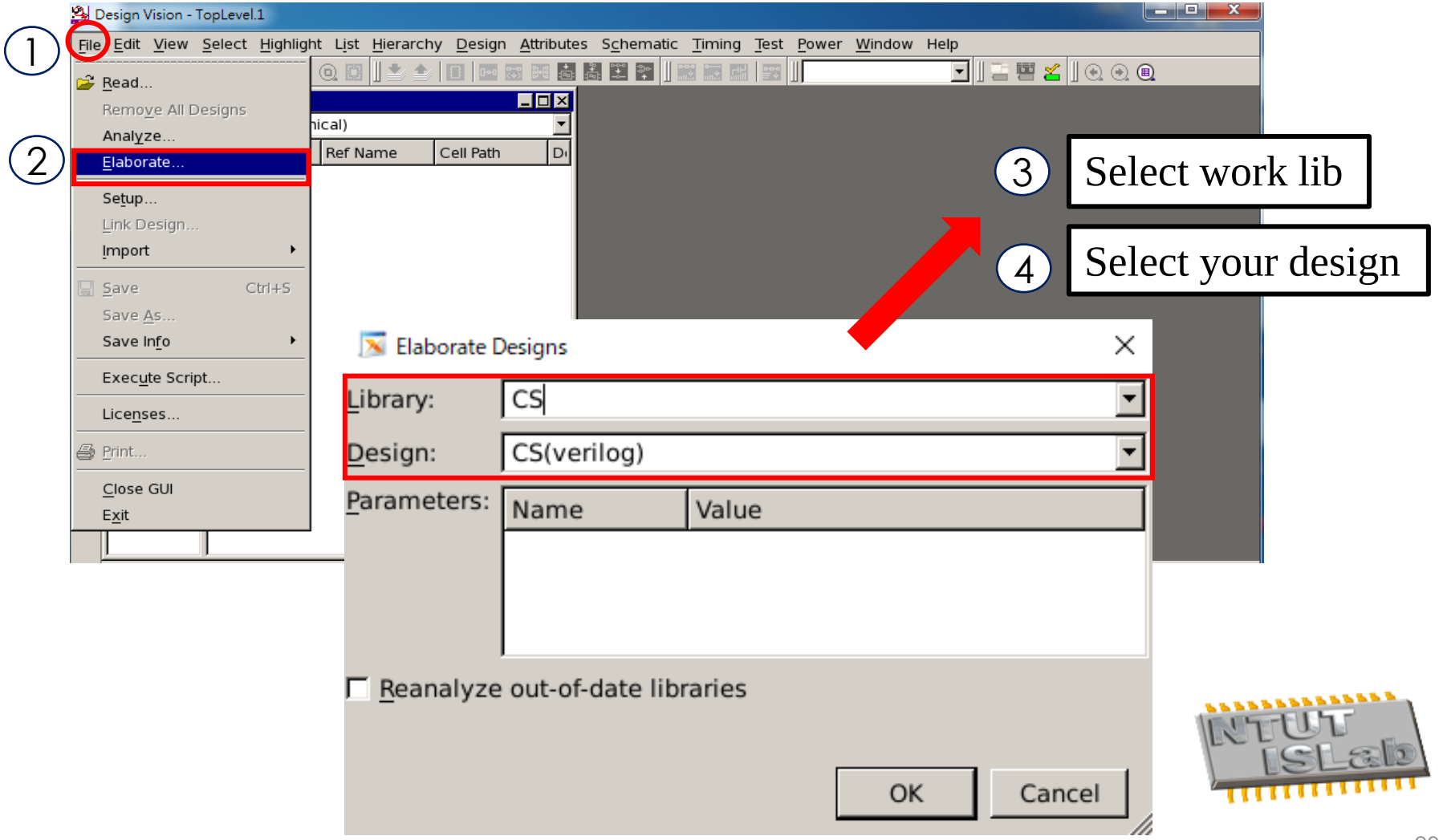
Analysis the Design(2/2)

```
design_vision> sh mkdir CS
design_vision> define_design_lib CS -path ./CS
design_vision> analyze -library CS -format verilog {/home/t107368062/Desktop/2020_cic_synthesis/2020_summer_test/synthesis/CS.v}
Running PRESTO HDLC
Compiling source file /home/t107368062/Desktop/2020_cic_synthesis/2020_summer_test/synthesis/CS.v
Presto compilation completed successfully.
Loading db file '/home/cell_lib/CBDK_TSMC018_Arm_v4.0/CIC/SynopsysDC/db/slow.db'
Loading db file '/home/cell_lib/CBDK_TSMC018_Arm_v4.0/CIC/SynopsysDC/db/fast.db'
Loading db file '/home/cell_lib/CBDK_TSMC018_Arm_v4.0/CIC/SynopsysDC/db/tpz973gvwc.db'
Loading db file '/home/cell_lib/CBDK_TSMC018_Arm_v4.0/CIC/SynopsysDC/db/tpz973gvtc.db'
Loading db file '/home/cell_lib/CBDK_TSMC018_Arm_v4.0/CIC/SynopsysDC/db/tpz973gvbc.db'
Loading db file '/usr/cad/synopsys/synthesis/cur/libraries/syn/dw_foundation.sldb'
design_vision>
```

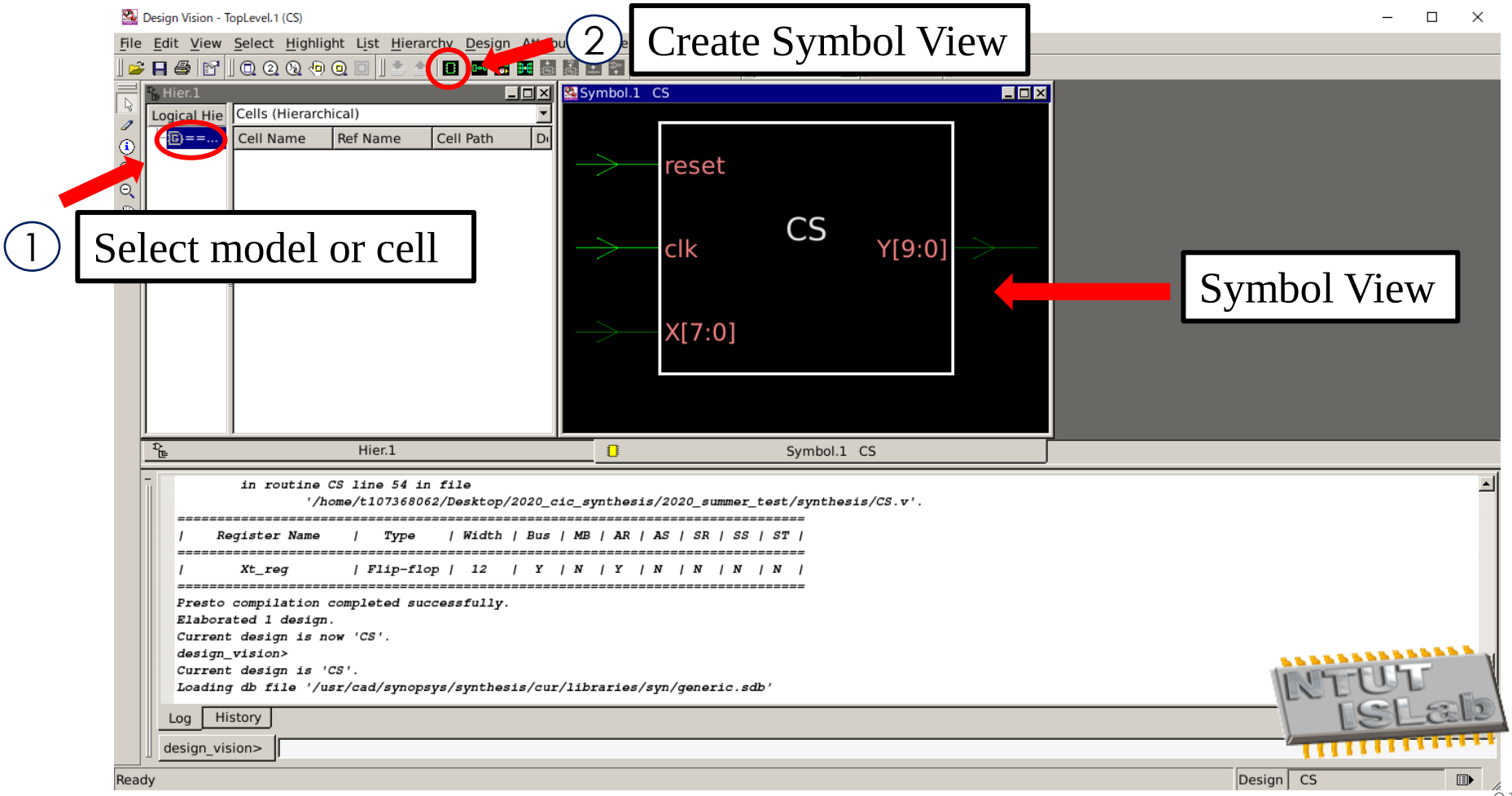
Log | History

design_vision>

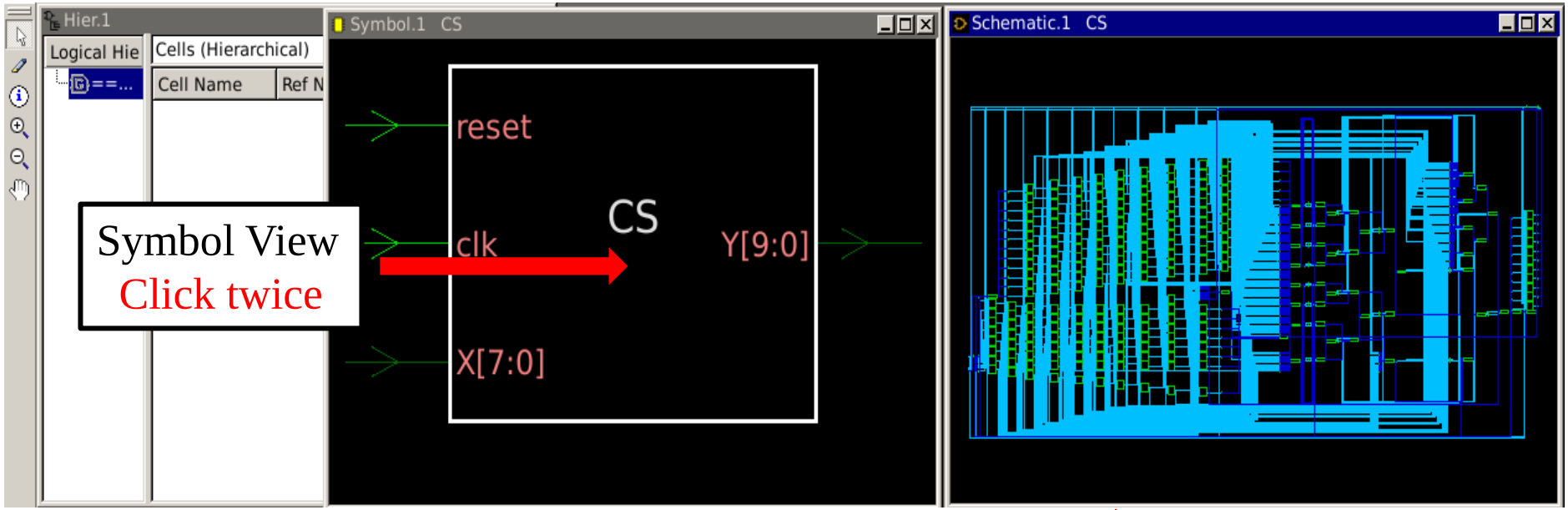
Elaborate the Design



Symbol View


Design Vision - TopLevel.1 (CS)
File Edit View Select Highlight List Hierarchy Design
2
Create Symbol View
Hier.1
Logical Hie
Cells (Hierarchical)
Cell Name
Ref Name
Cell Path
1
Select model or cell
Symbol.1 CS
reset
clk
X[7:0]
CS
Y[9:0]
Symbol View
Hier.1
Symbol.1 CS
in routine CS line 54 in file
'/home/t107368062/Desktop/2020_cic_synthesis/2020_summer_test/synthesis/CS.v'.
| Register Name | Type | Width | Bus | MB | AR | AS | SR | SS | ST |
| Xt_reg | Flip-flop | 12 | Y | N | Y | N | N | N | N |
Presto compilation completed successfully.
Elaborated 1 design.
Current design is now 'CS'.
design_vision>
Current design is 'CS'.
Loading db file '/usr/cad/synopsys/synthesis/cur/libraries/syn/generic.sdb'
Log
History
design_vision>
NTUT ISLab
Ready
Design
CS

Schematic View



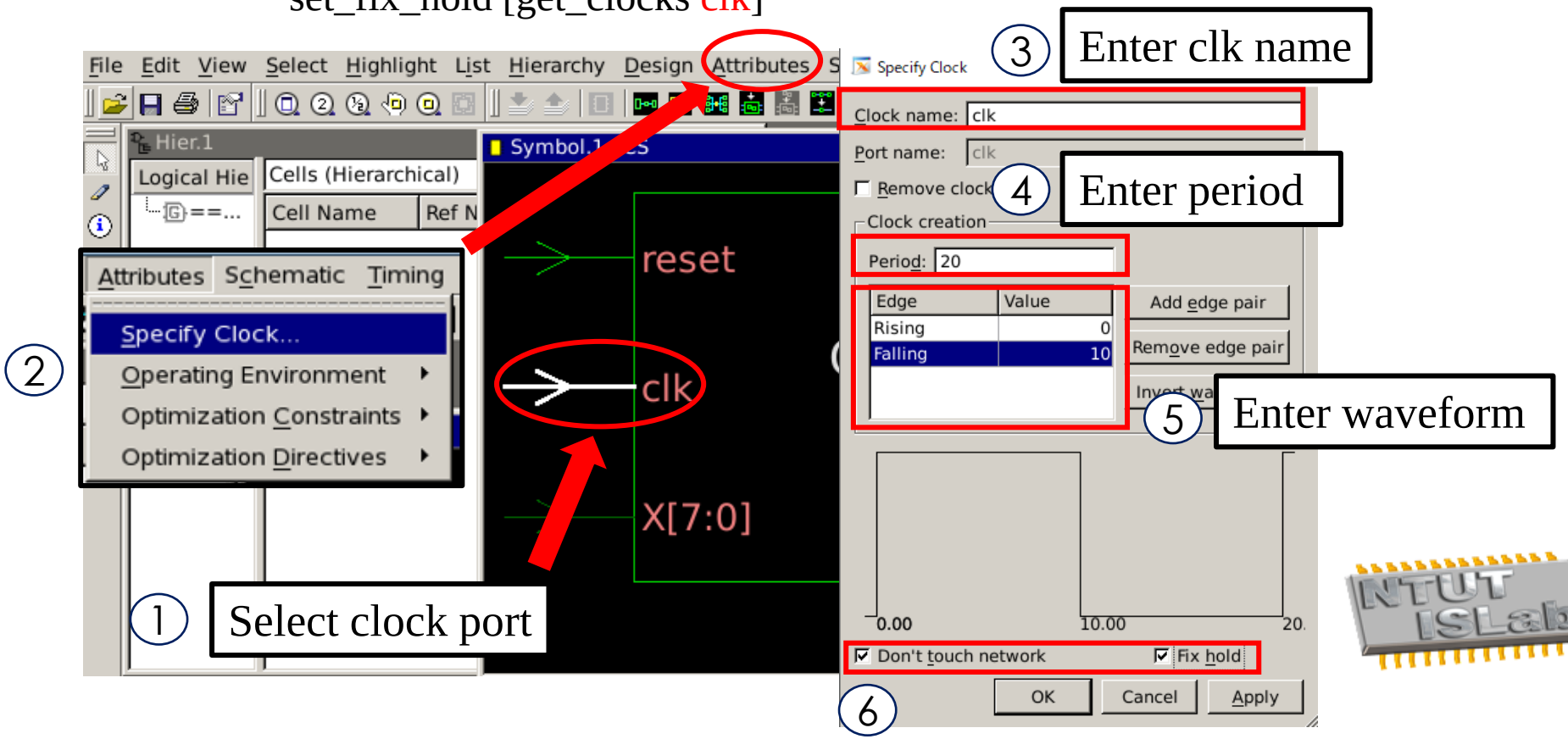
SETTING DESIGN CONSTRAINTS

Define Clock specification

- We need to accurately specify the clock including the clock routing details in the early design stage in order to achieve timing convergence.
- What should be defined?
  - ✓ Period
  - ✓ Waveform
  - ✓ Uncertainty
    - skew
  - ✓ Latency
    - Source latency
    - Network latency
  - ✓ Transition
    - Input transition
    - Clock transition
  - ✓ Area
  - ✓ Max fanout

Specify Clock

Command: create_clock -name clk -period 20 -waveform {0 10} [get_ports clk]
set_dont_touch_network [get_clocks clk]
set_fix_hold [get_clocks clk]



Report Clock

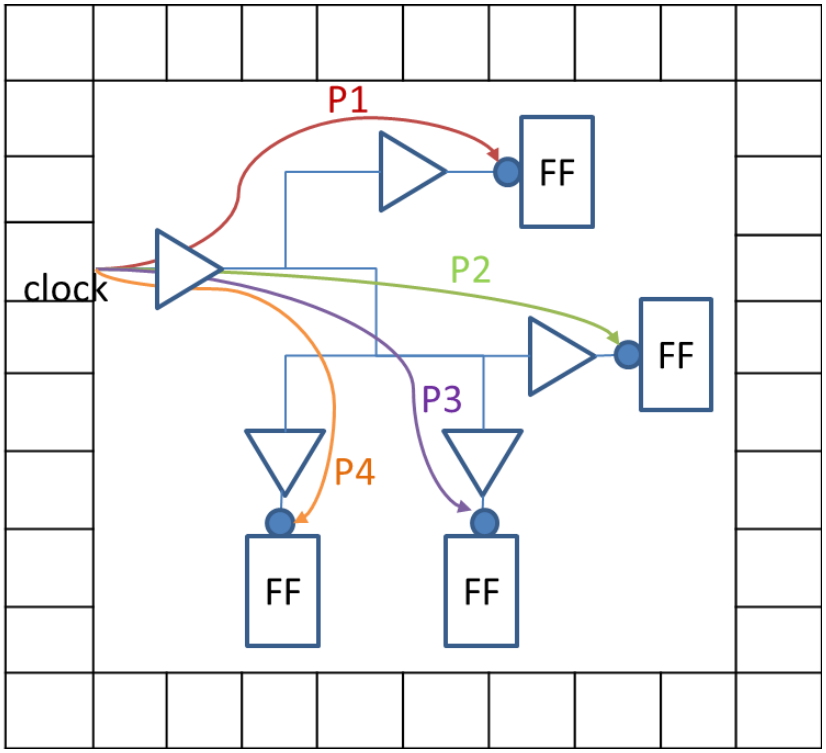
- Design → Report_Clocks

```
****************************************

Attributes:
    d - dont_touch_network
    f - fix_hold
    p - propagated_clock
    G - generated_clock
    g - lib_generated_clock

Clock          Period   Waveform            Attrs     Sources
-------------------------------------------------------------
clk             20.00   {0 10}              d f       {clk}
-------------------------------------------------------------
Current design is 'CS'.

***** End Of Report *****
```

Clock skew

- The maximum difference between the arrival of clock signals at sequential cells in one clock domain or between domains.

Command: set_clock_uncertainty 0.8 [get_clocks clock]



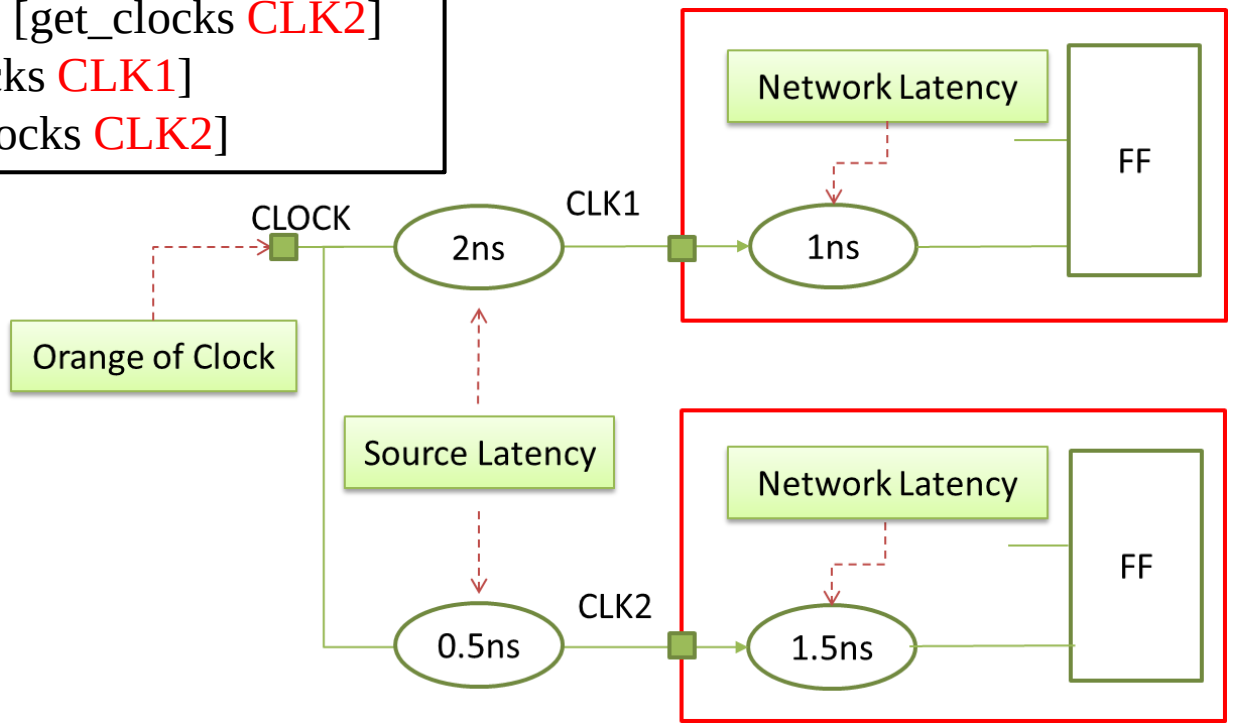
Arrival(P1)=0.5ns
Arrival(P2)=1ns
Arrival(P3)=1.2ns
Arrival(P4)=1.3ns

Uncertainty = 1.3 – 0.5
= 0.8ns clock skew!!

Model Source Latency

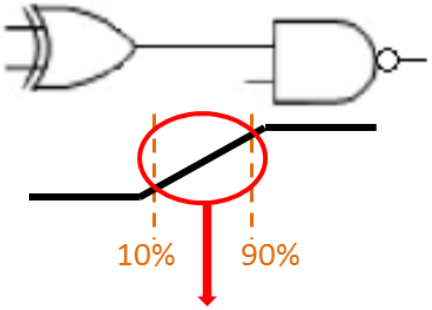
- Source latency is the propagation time from the actual clock origin to the clock definition point in the design.

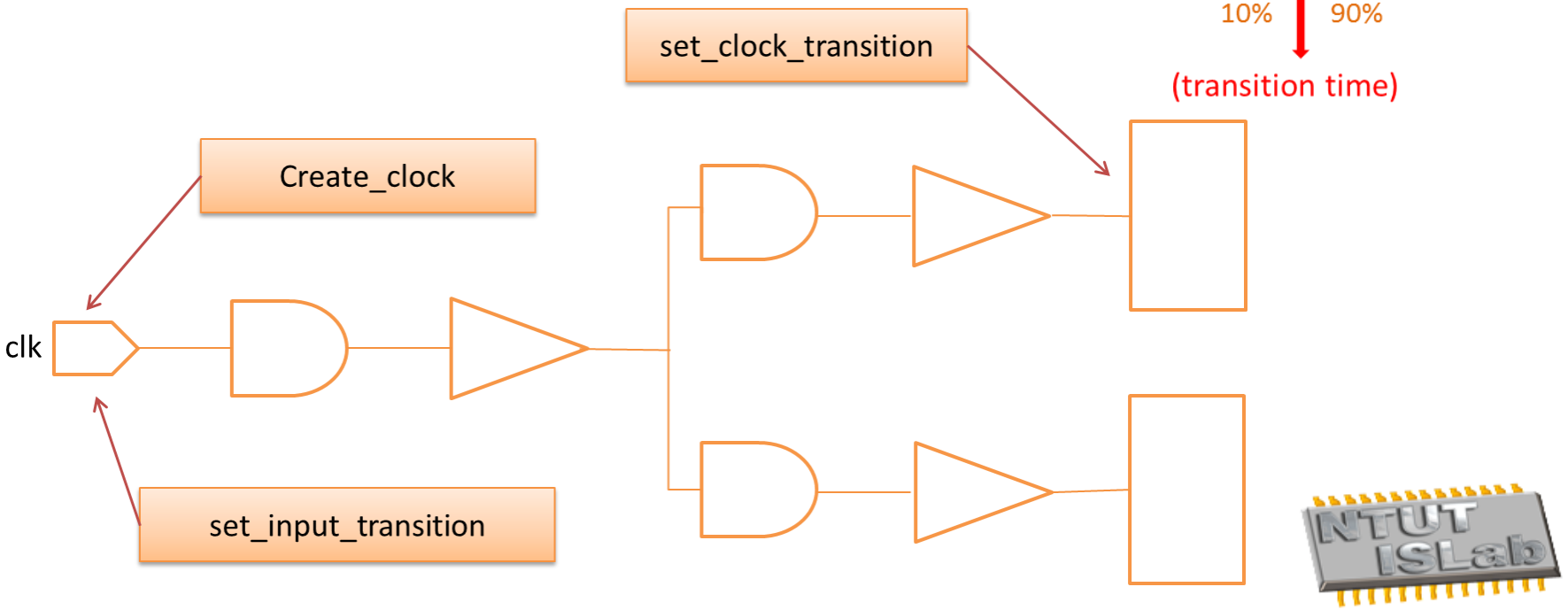
Command:
set_clock_latency -source 2 [get_clocks CLK1]
set_clock_latency -source 0.5 [get_clocks CLK2]
set_clock_latency 1 [get_clocks CLK1]
set_clock_latency 1.5 [get_clocks CLK2]



Clock Transition Time

Command: set_input_transition 0.5 [get_clocks clk]
set_clock_transition 0.1 [get_clocks clk]



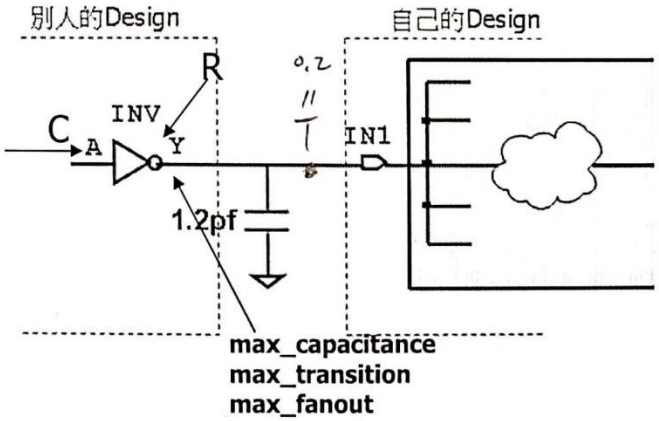


Design Rule Constraints

- Vendors impose design rules that restrict how many cells are connected to one another based on capacitance, transition and fanout
- You may apply more conservative design rules to:
  - Anticipate the interface environment your block will see
  - Prevent the design from operating cells close to their limits, where performance degrades rapidly

DRC Syntax:

```
set_max_capacitance 0.1 [get_ports IN1]
set_max_transition 0.2 [get_ports IN1]
set_max_fanout 10 [get_ports IN1]
```



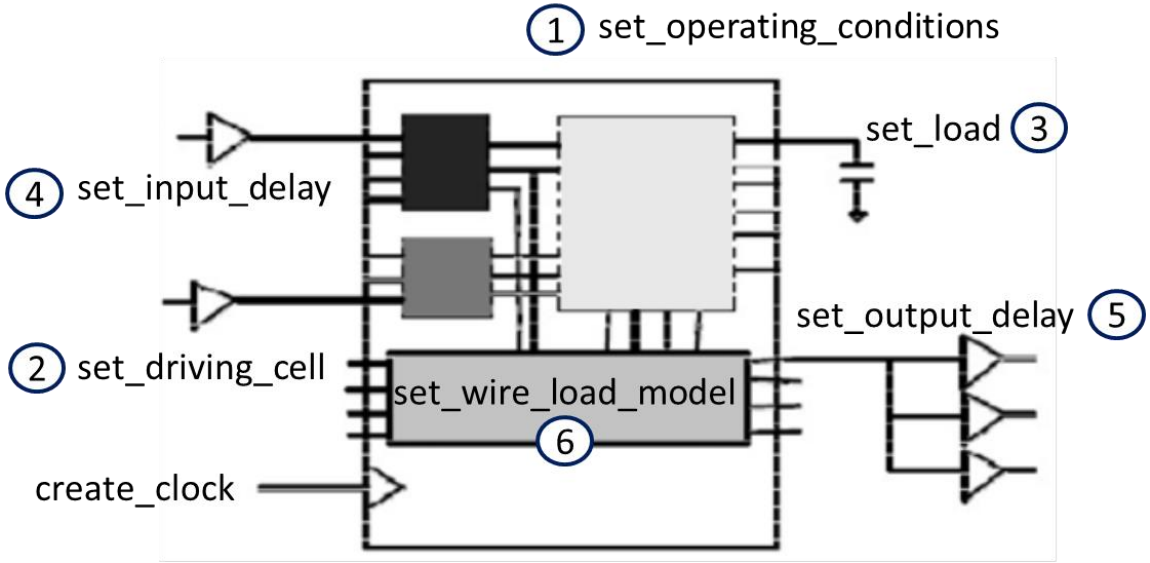
S.S. WANG / 2022.01

3 - 73

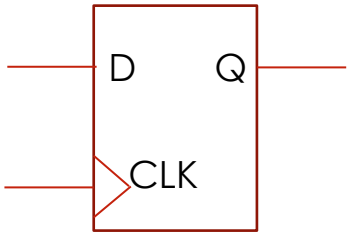
SETTING DESIGN ENVIRONMENT

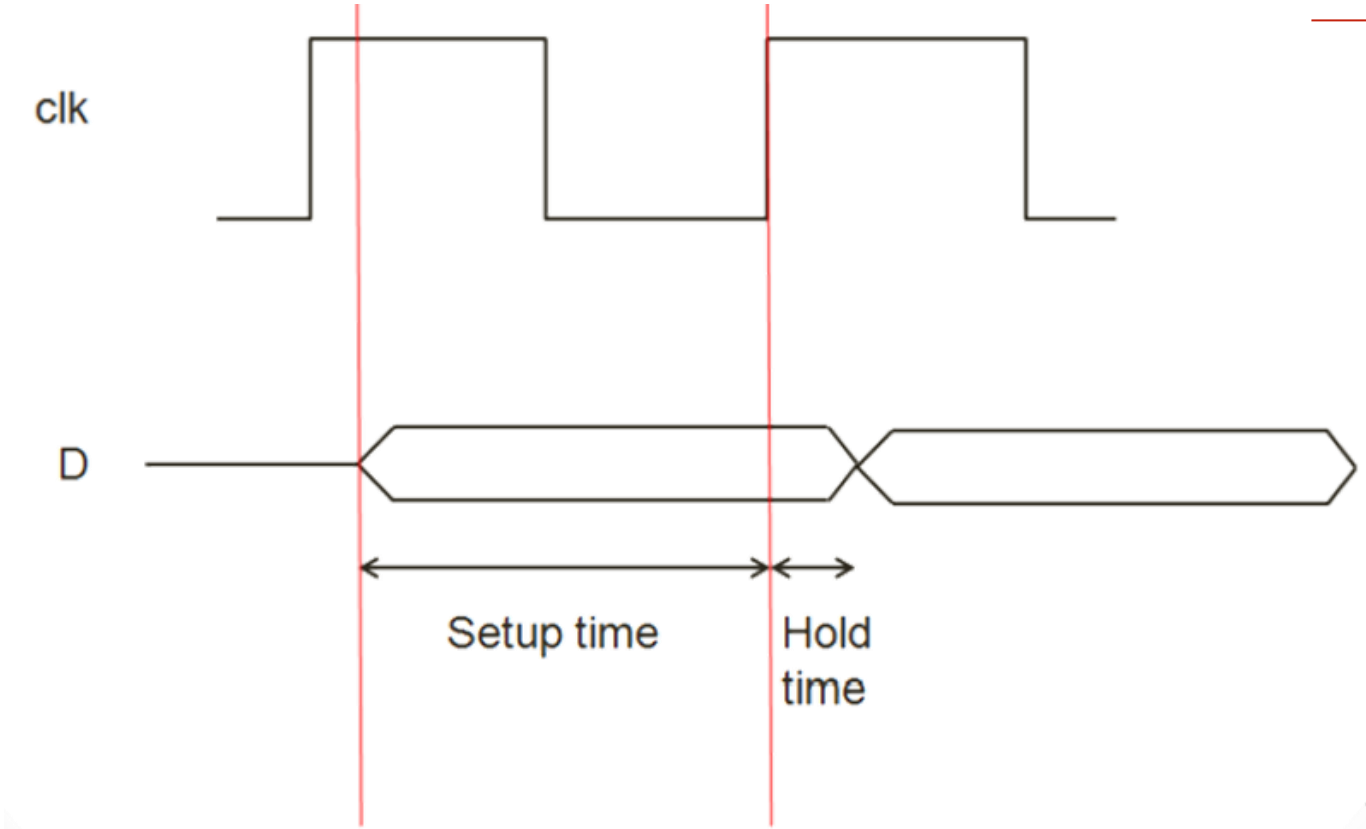
Real World Environmen

- Beware the defaults are not realistic conditions
  - ✓ Input drive is not infinite
  - ✓ Capacitive loading is usually not zero
  - ✓ Consider process, voltage, temperature (PVT) variation



Setup Time & Hold Time

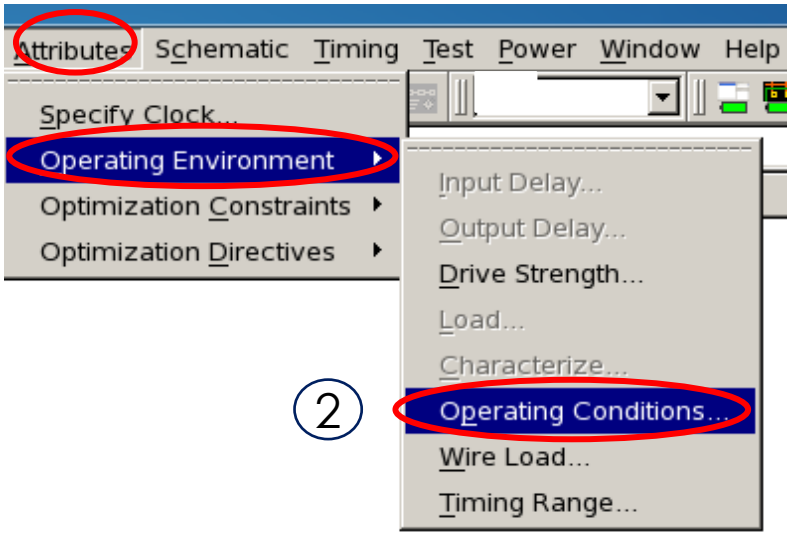


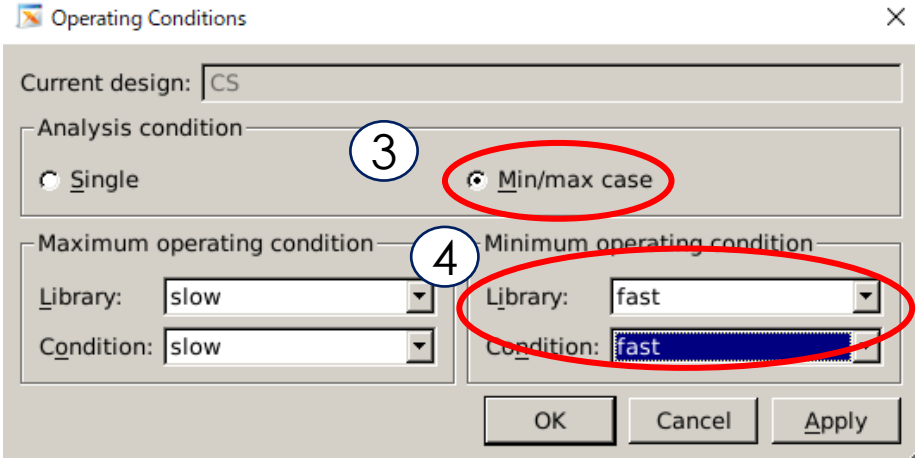


Setting Operating Condition

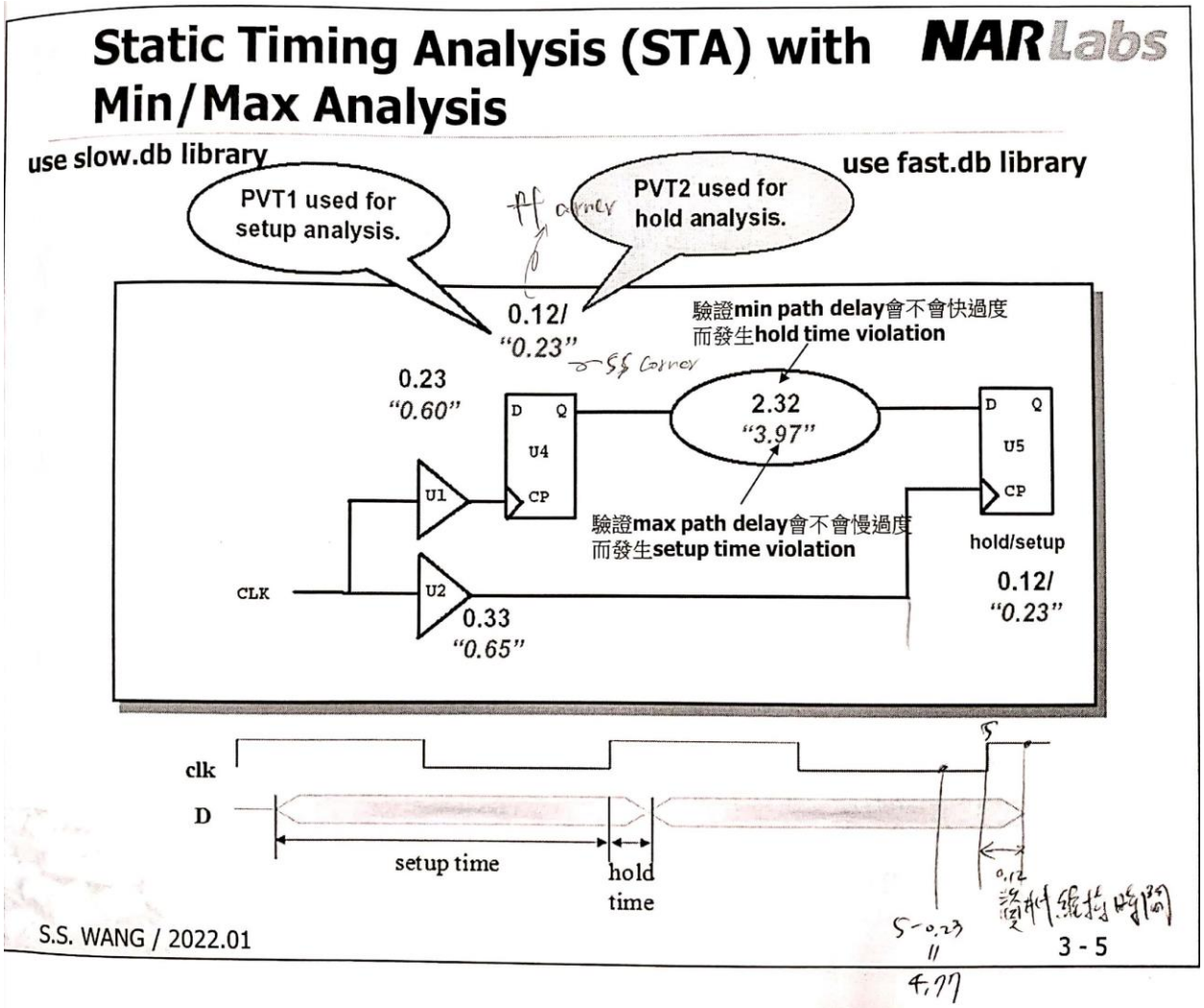
Command: set_operating_conditions -min_library fast -min fast -max_library slow -max slow

1





Setting Operating Condition



Setting driving Cell and output load

1.Setting input driving strength for clk port

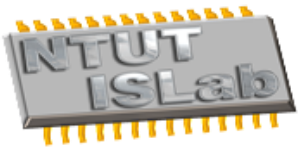
-> set_driving_cell -library slow -lib_cell BUFX4 -pin {Y}
[get_ports clk]

2.Setting input driving strength for all input port except clk

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

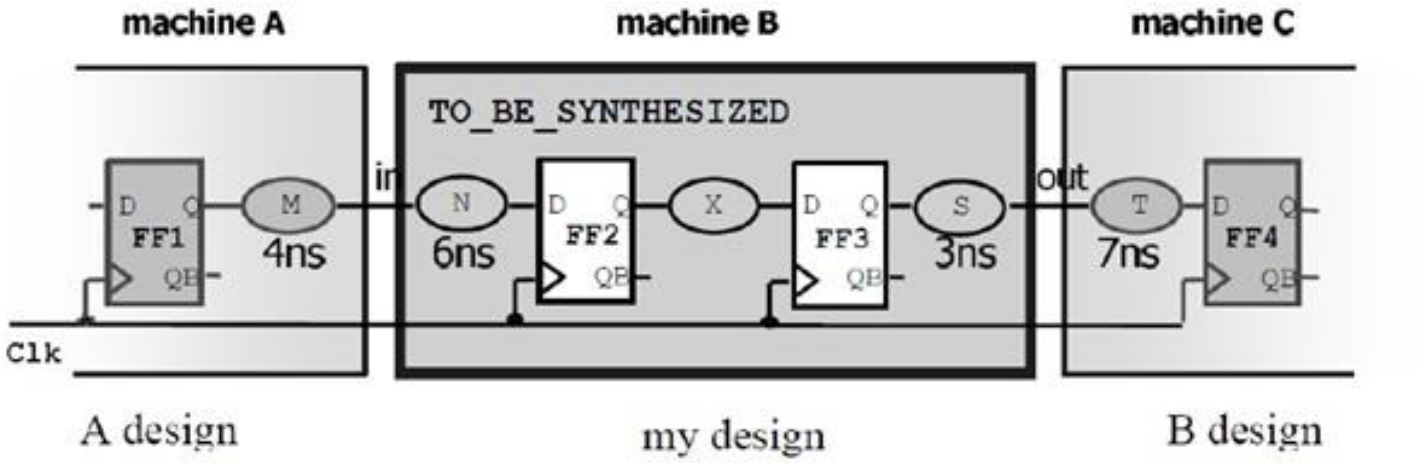
3.Setting output load 0.001913

-> set_load 0.001913 [all_outputs]



Input Delay and Output Delay

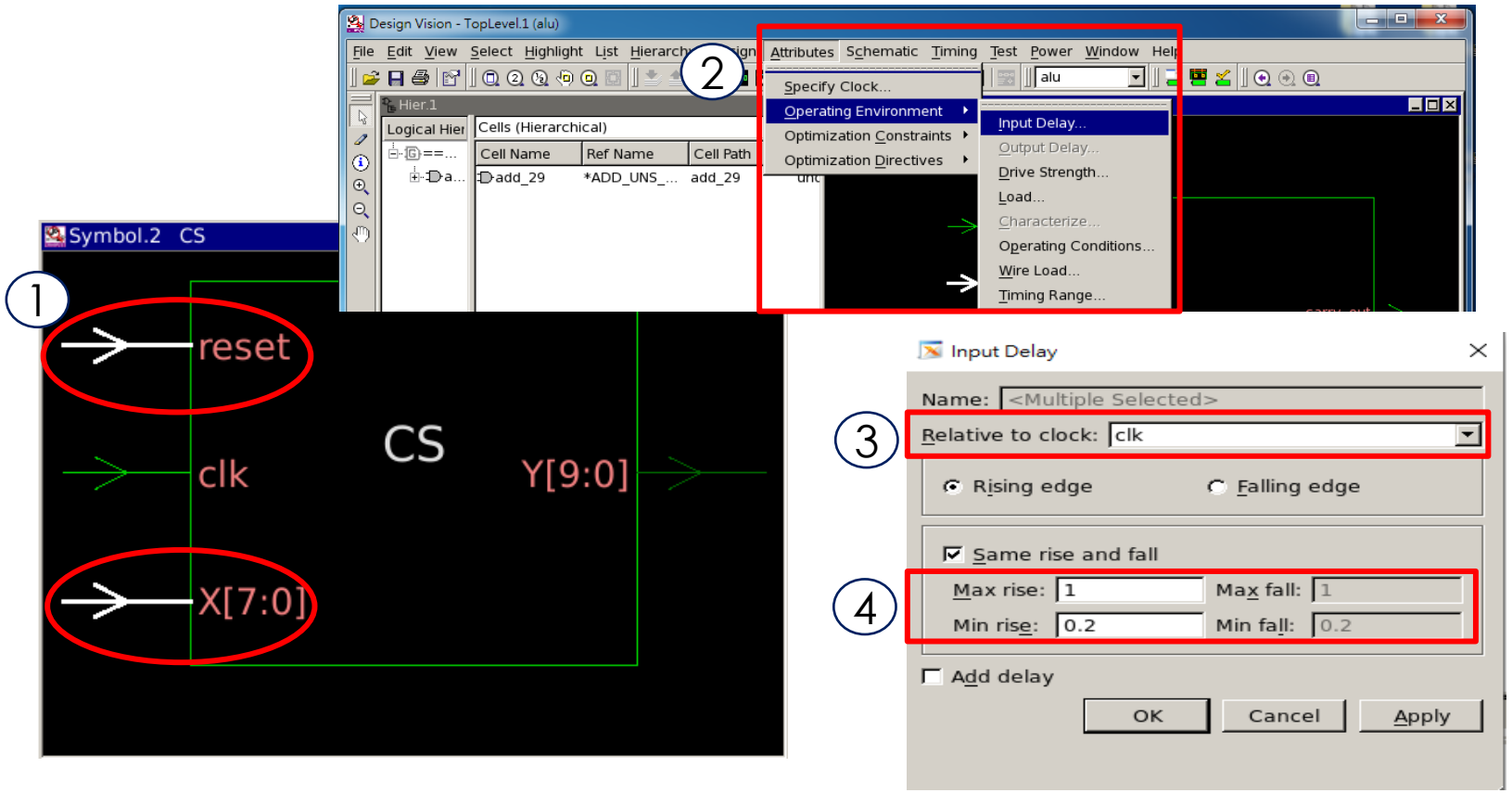
- Clock cycle > = (DFF clk - Q delay) + X + DFF setup
- Input Delay = (DFF clk - Q delay) + M
- Output delay = T + DFF setup




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

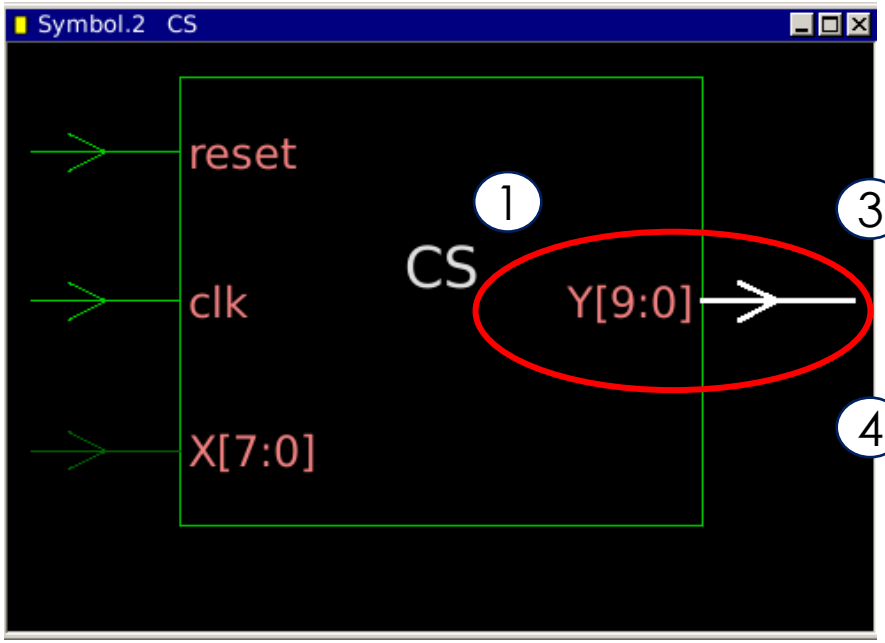
Setting Input Delay

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]]



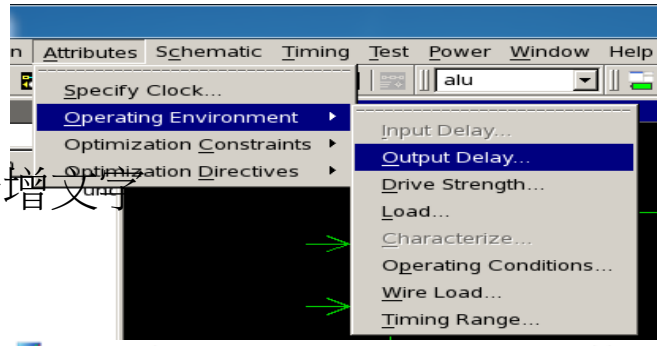
Setting Output Delay

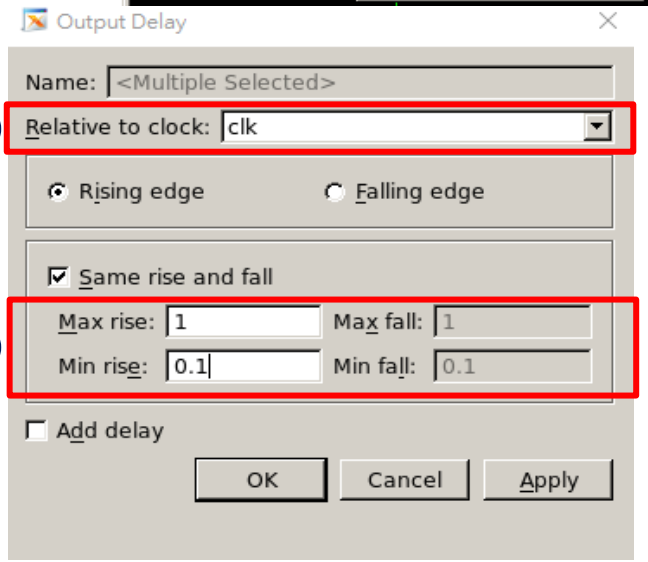
Command: set_output_delay -clock clk -max 1 [all_outputs]
set_output_delay -clock clk -min 0.1 [all_outputs]



2

按一下以新增文字



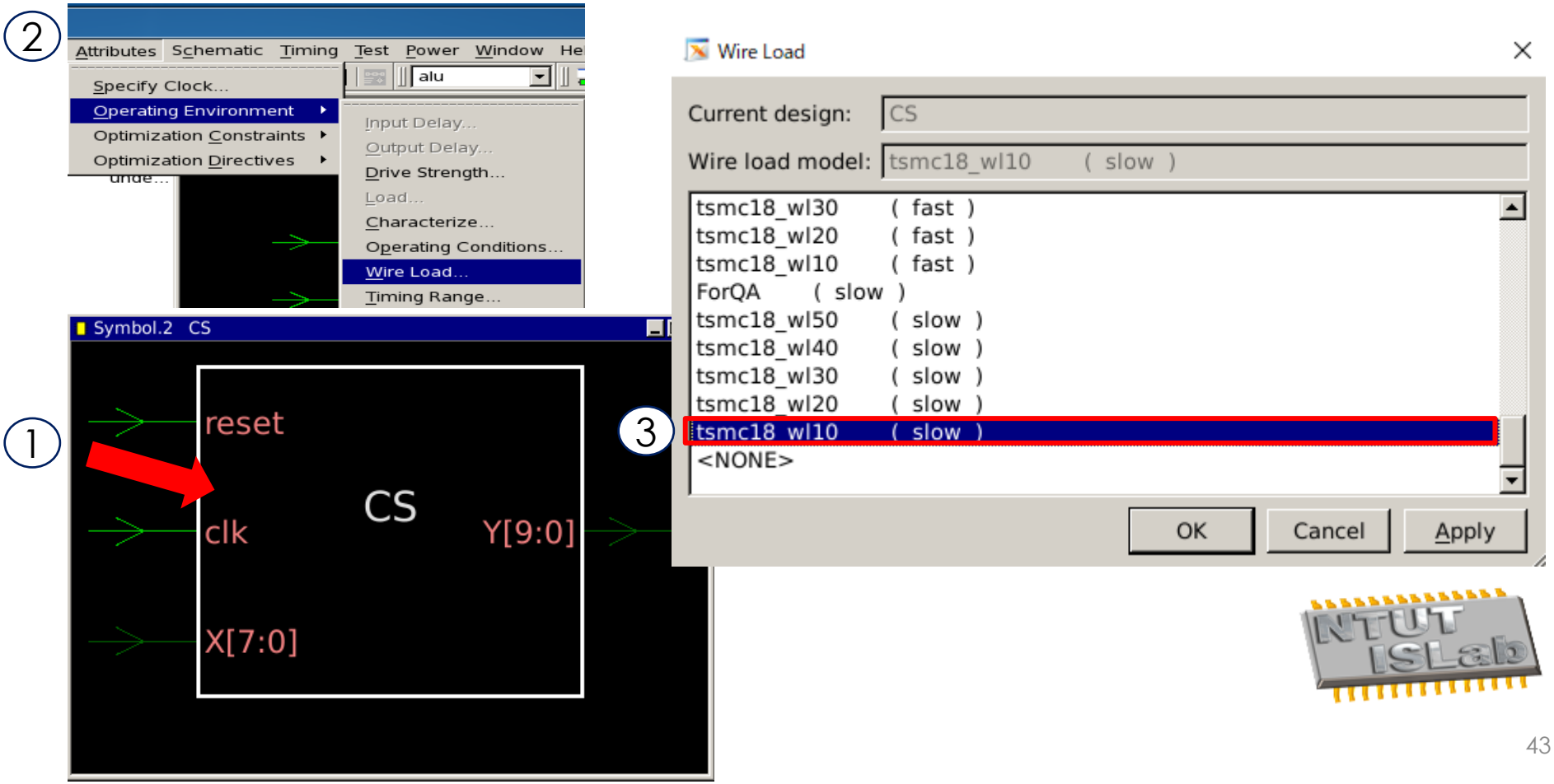


3

4

Setting Wire Load Model for Net Delay

Command: set_wire_load_model -name tsmc18_wl10 -library slow



Check & uniquify Design

- You have to check design after setting.

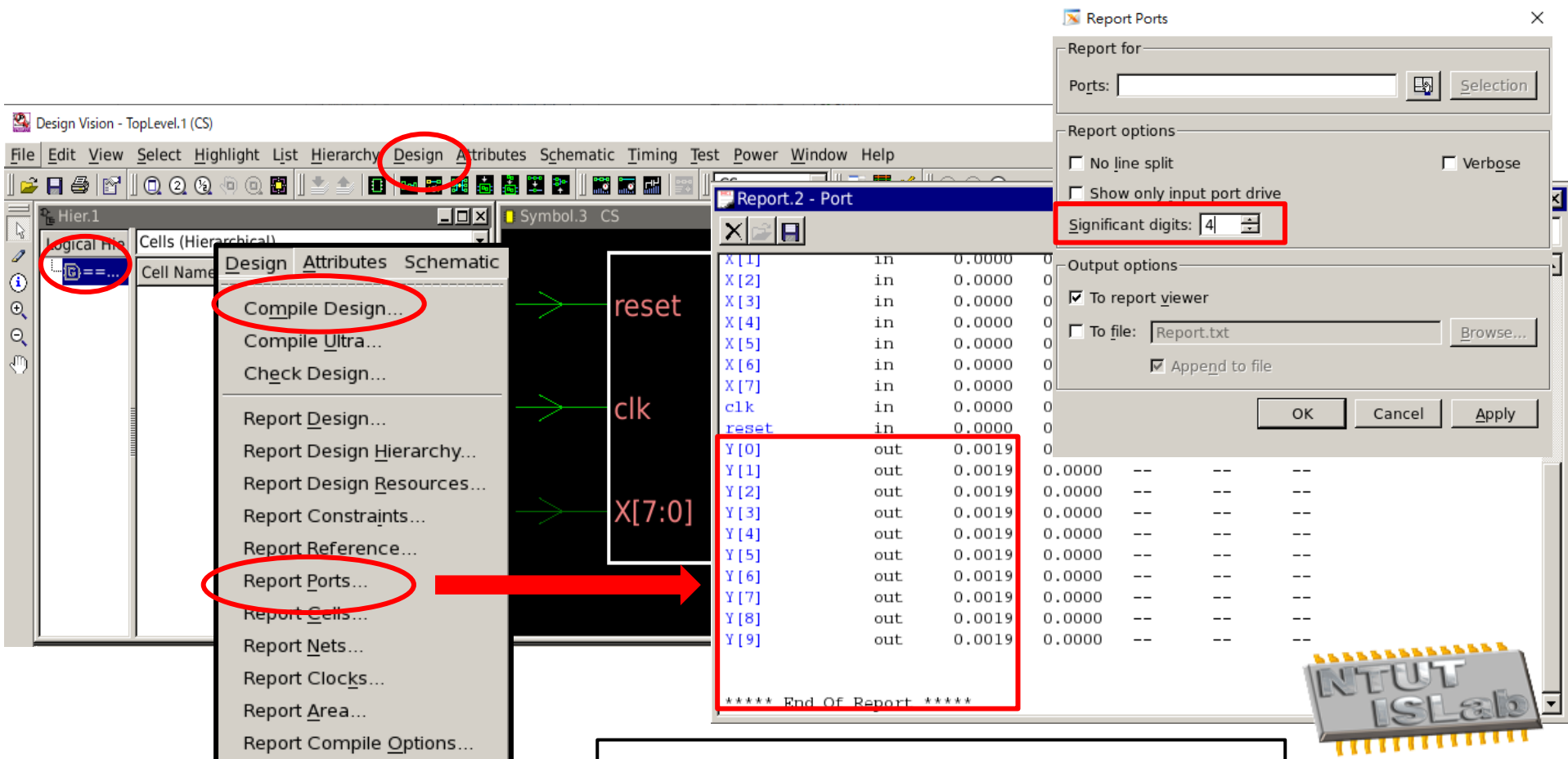
Command:

->check_design -multiple_designs

->uniquify

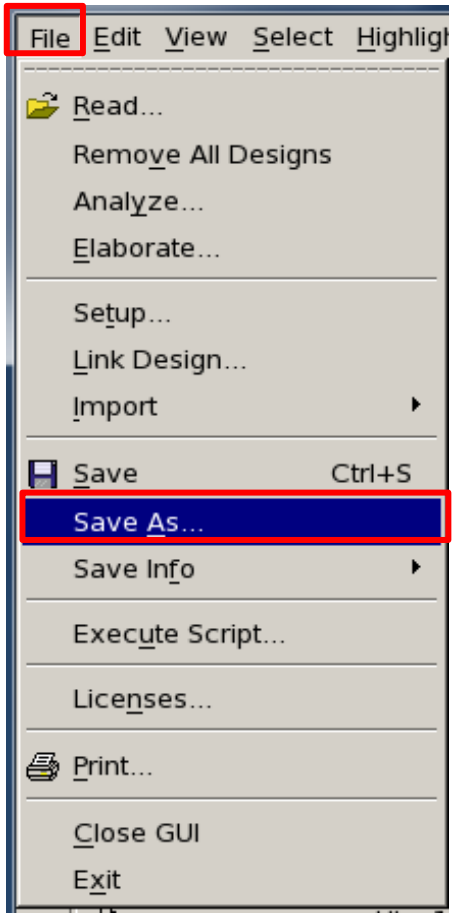
REPORTS AND SAVE FILE

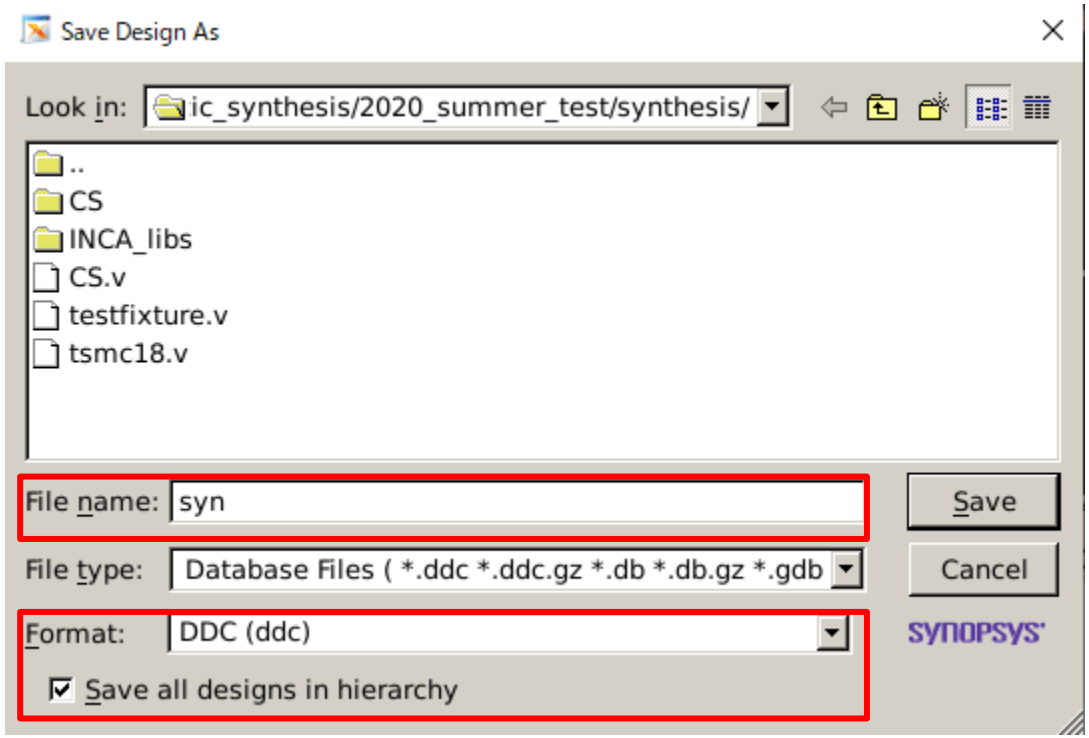
Reports Before Compiler



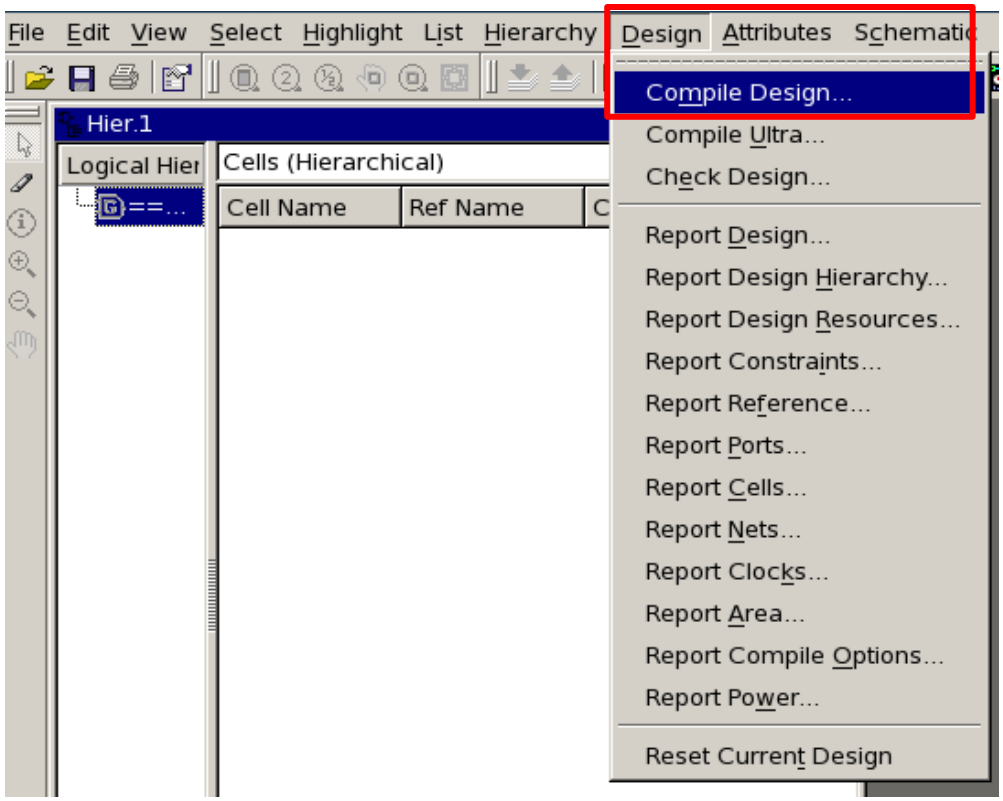
Make sure the output loading is 0.001913 in Report Ports.

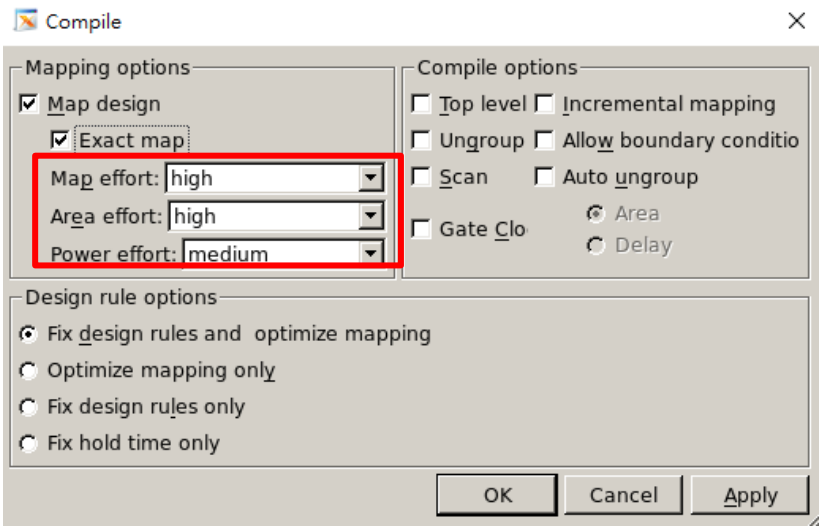
Save Design as .DDC File





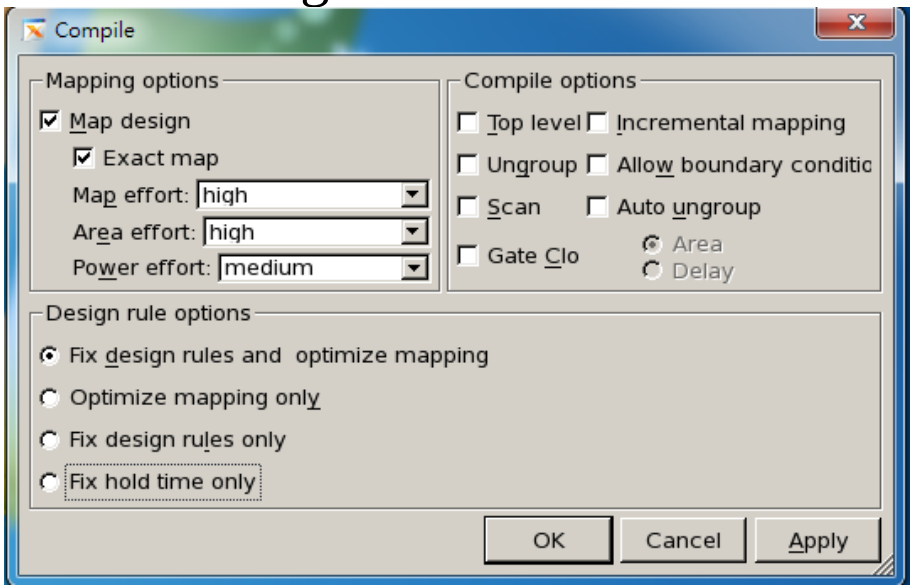
Compile Design



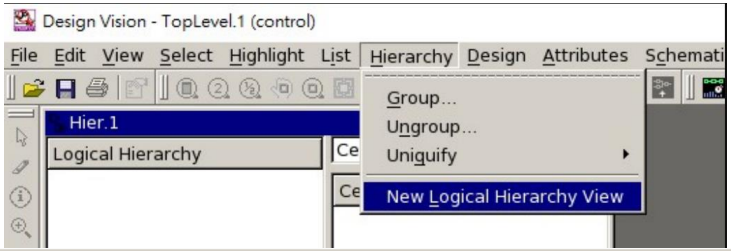


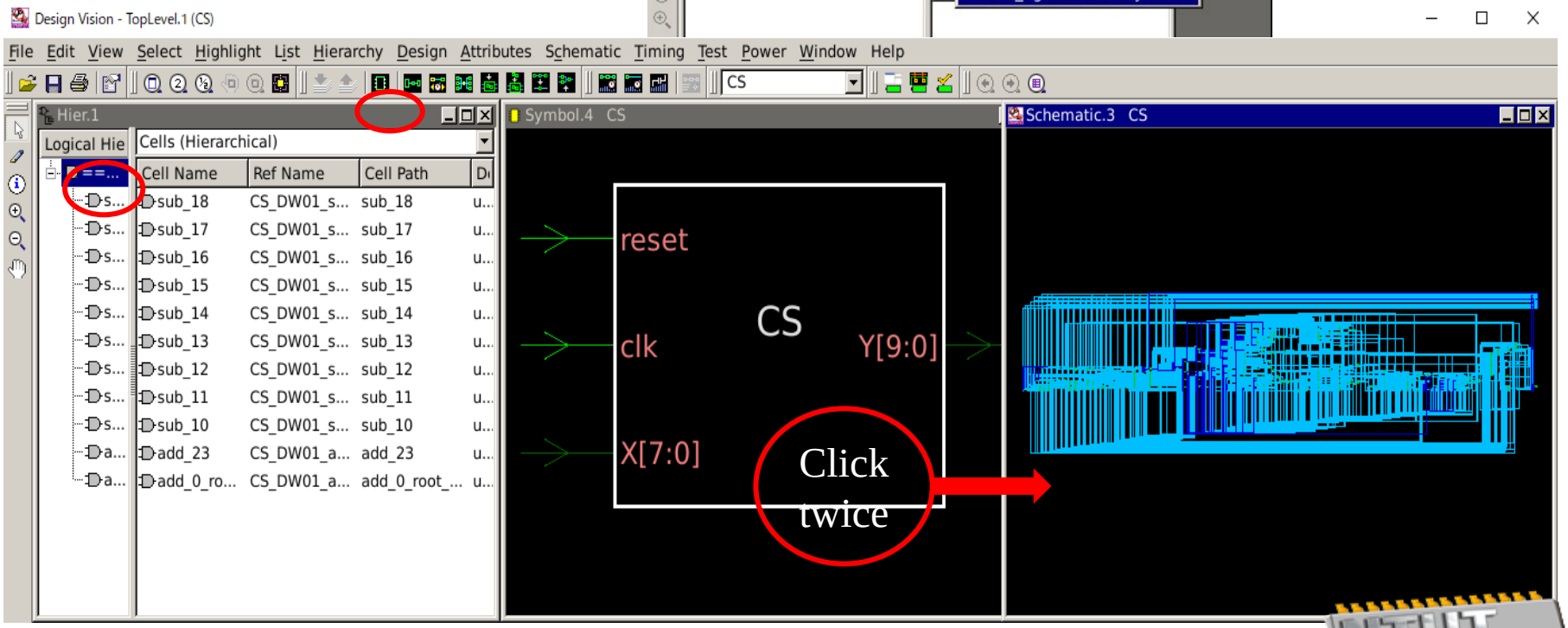
Mapping Effort

- Three effort levels, low, medium, high, determine relative amount of CPU time spent during mapping phase of compile.
  - Low - quicker synthesis, does not do all algorithms.
  - Medium – default, good, for many design.
  - High – it does critical path re-synthesis, but it will use more CPU time; in some cases the action of compile will not terminate.



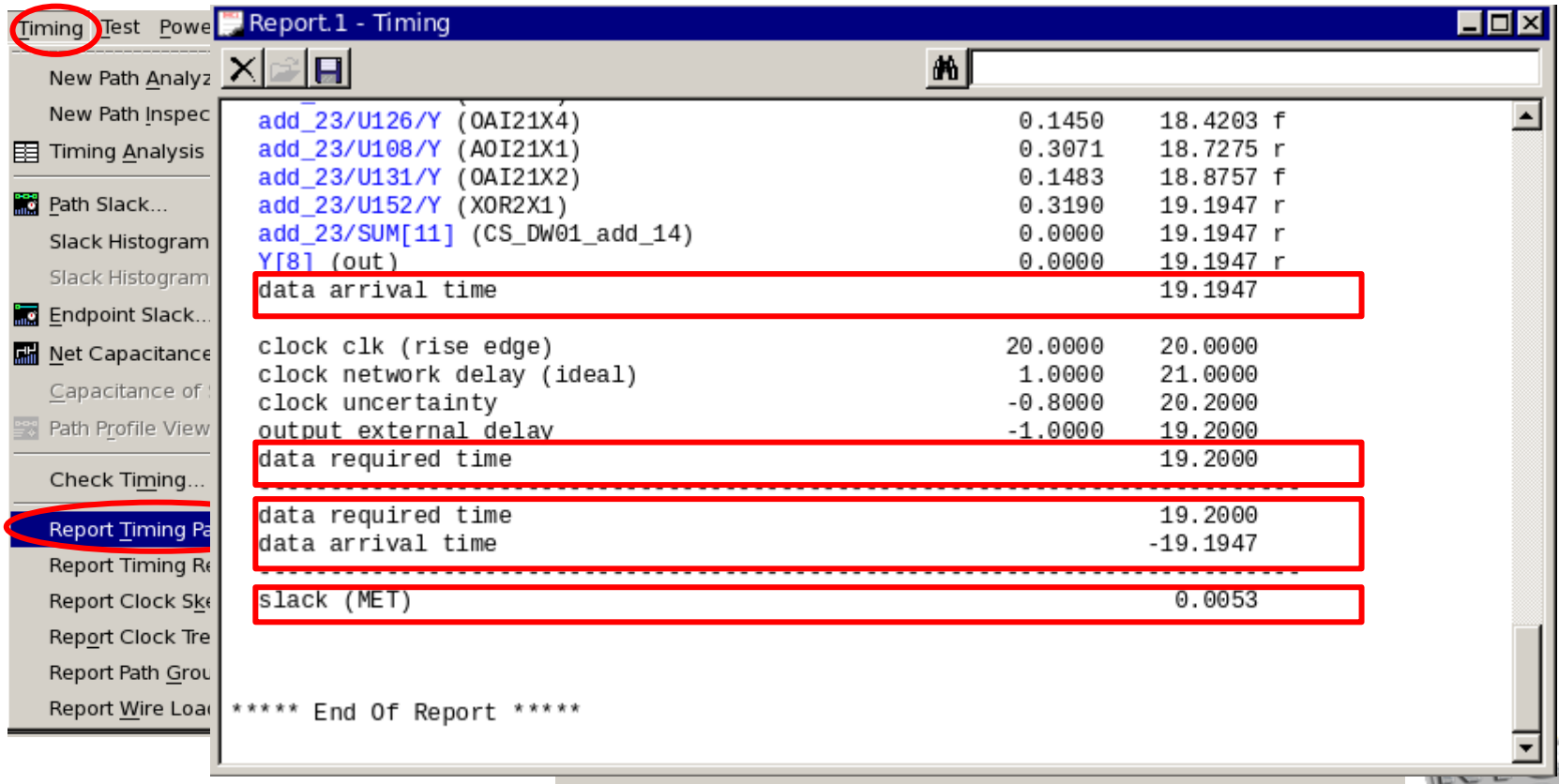
Explore The Schematic View





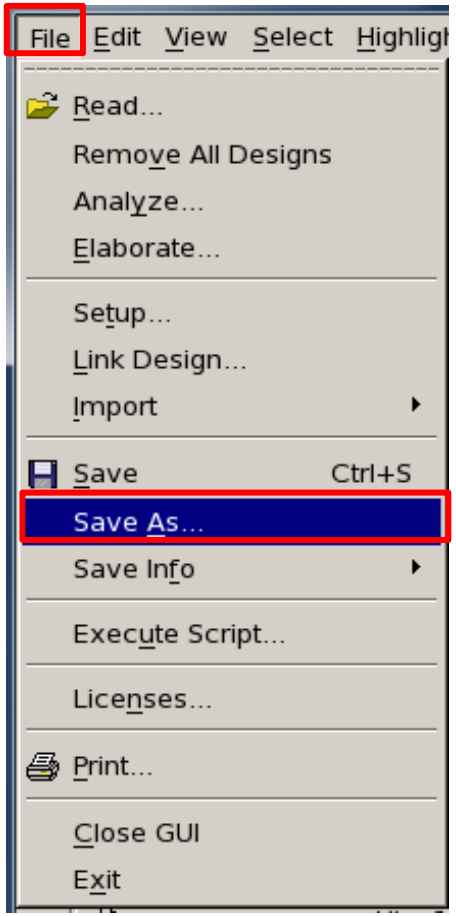
Select CS and save again

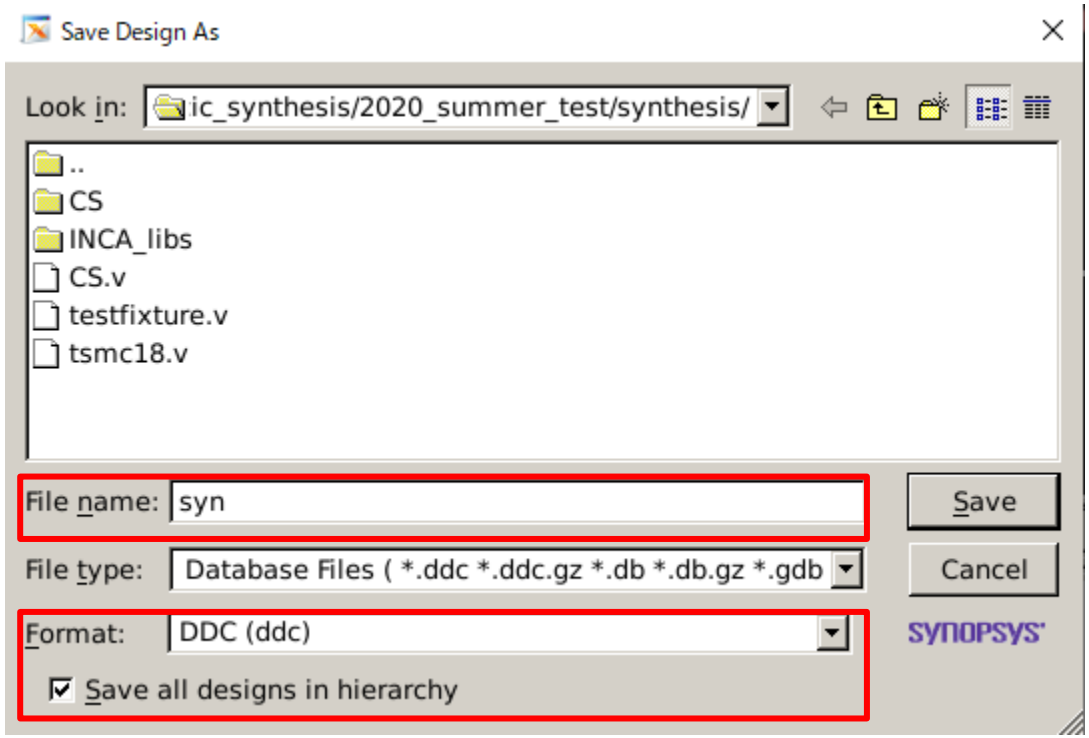
Report Area & Timing



Reports Area (μm^2) : Total Area
Gate Count Gate Count: Total cell area / 10

Save Design as .DDC File





Write SDF File for Pre-Sim

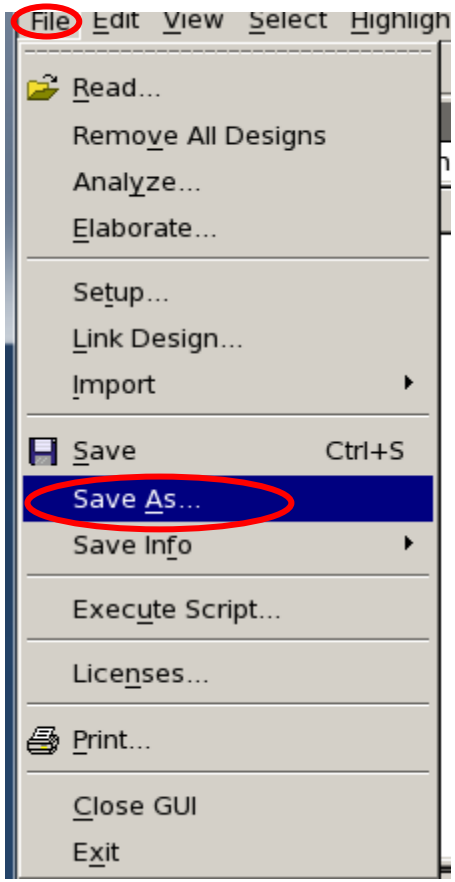
- The file includes information of delay of Gate-Level circuit.

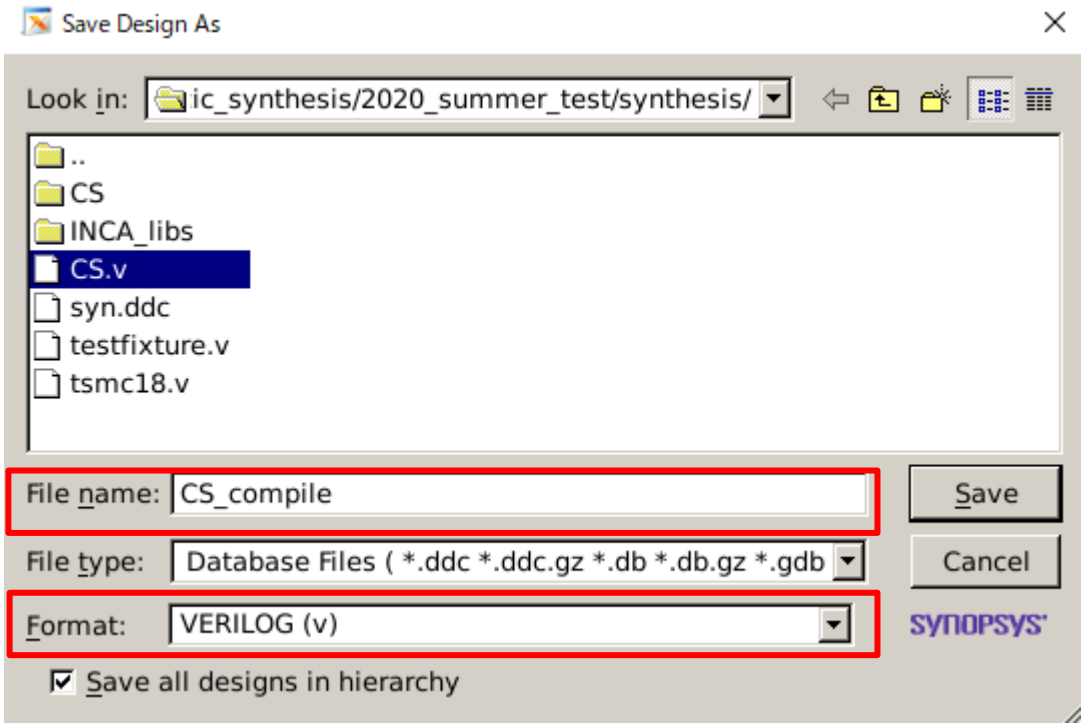
Command:

-> write_sdf -version 1.0 -context verilog chip_syn.sdf

-> write_sdc -version 1.7 CHIP.sdc

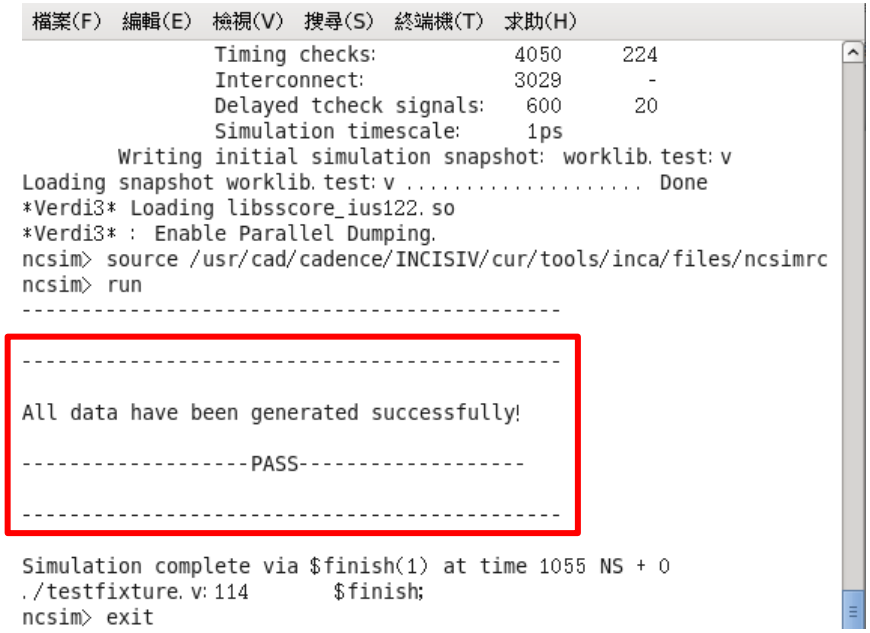
Save Design as .V File : Netlist





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。



Command (Terminal , not Design Compiler):
ncverilog testfixture.v CS_compile.v –v tsmc18.v +access+r


Thank
You!

LAB2 PROCEDURE

- Environment setup
- Environment setup check
- Analysis the design, Elaborate the design
- Constraints(Reference command)
- Compile design
- Reports and Save file
- Gate-Level simulation

Environment Setup (for control.v)

Create a New Folder for Synthesis

Ex:

1. mkdir Synthesis (terminal command)
2. cd Synthesis (terminal command)

Copy File for Environment Setting

Δ 代表空格

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

Copy File for Synthesis

cp Δ/home/standard/vlsi_2022/synthesis/* . Δ

除了以上cp指令，Lab1寫完的control相關檔案也要複製到Synthesis的資料夾中

Reference command_1

- Command:

create_clock -name clk -period 20 -waveform {0 10} [get_ports clk]
set_dont_touch_network [get_clocks clk]
set_fix_hold [get_clocks clk]

set_clock_uncertainty 0.8 [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]

set_max_area 0

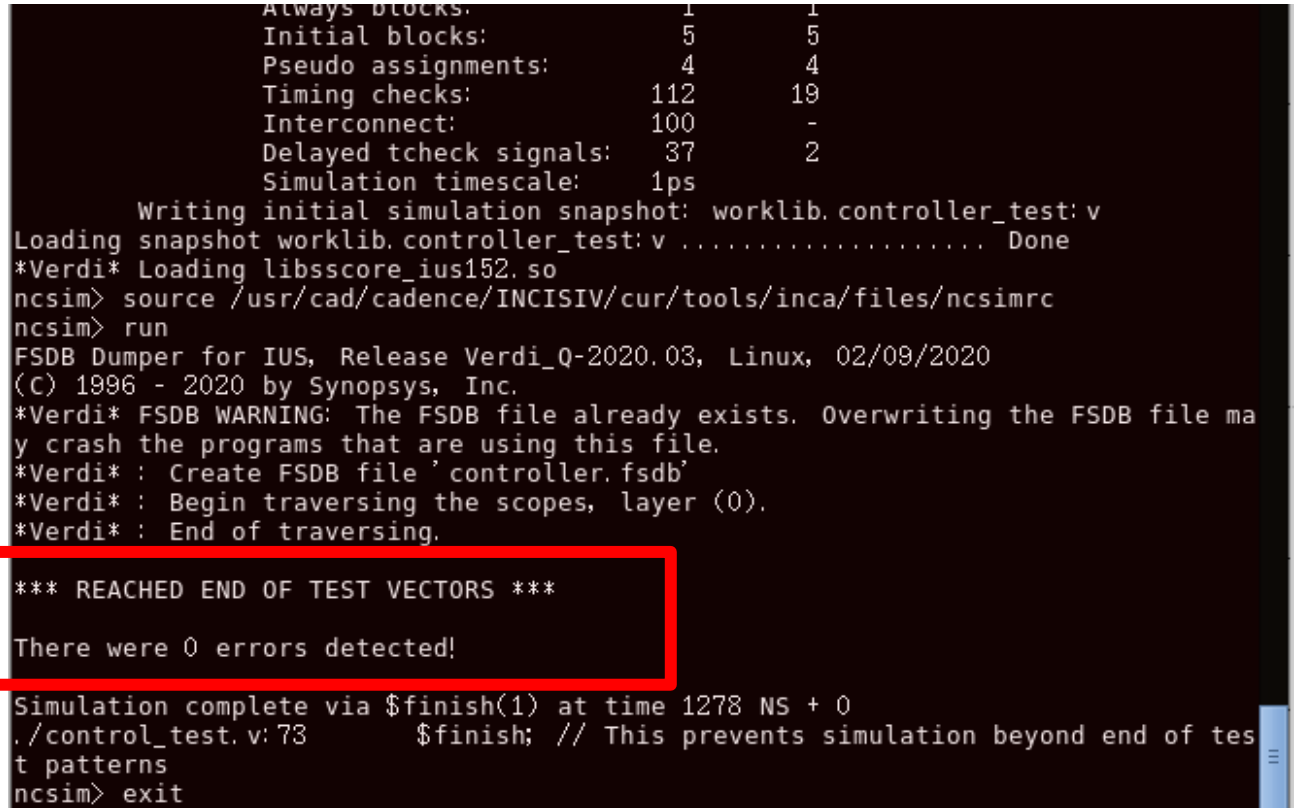
set_max_fanout 6 [all_inputs]
set_max_transition 0.3 [all_inputs]

Reference command_2

- Command:

```
set_driving_cell -library slow -lib_cell BUFX4 -pin {Y}  [get_ports clk]
set_driving_cell -library slow -lib_cell DFFX1 -pin {Q} [remove_from_collection [all_inputs] [get_ports clk]]
set_load 0.001913 [all_outputs]

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]]

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

 set_wire_load_model -name tsmc18_wl10 -library slow

check_design -multiple_designs
uniquify
```

Gate-Level simulation



Command (Terminal , not Design Compiler):

ncverilog control_test_syn.v control_syn.v –v tsmc18.v +access+r +define+SDF

合成完的NETLIST檔