

# **Cell-based IC Design Flow**

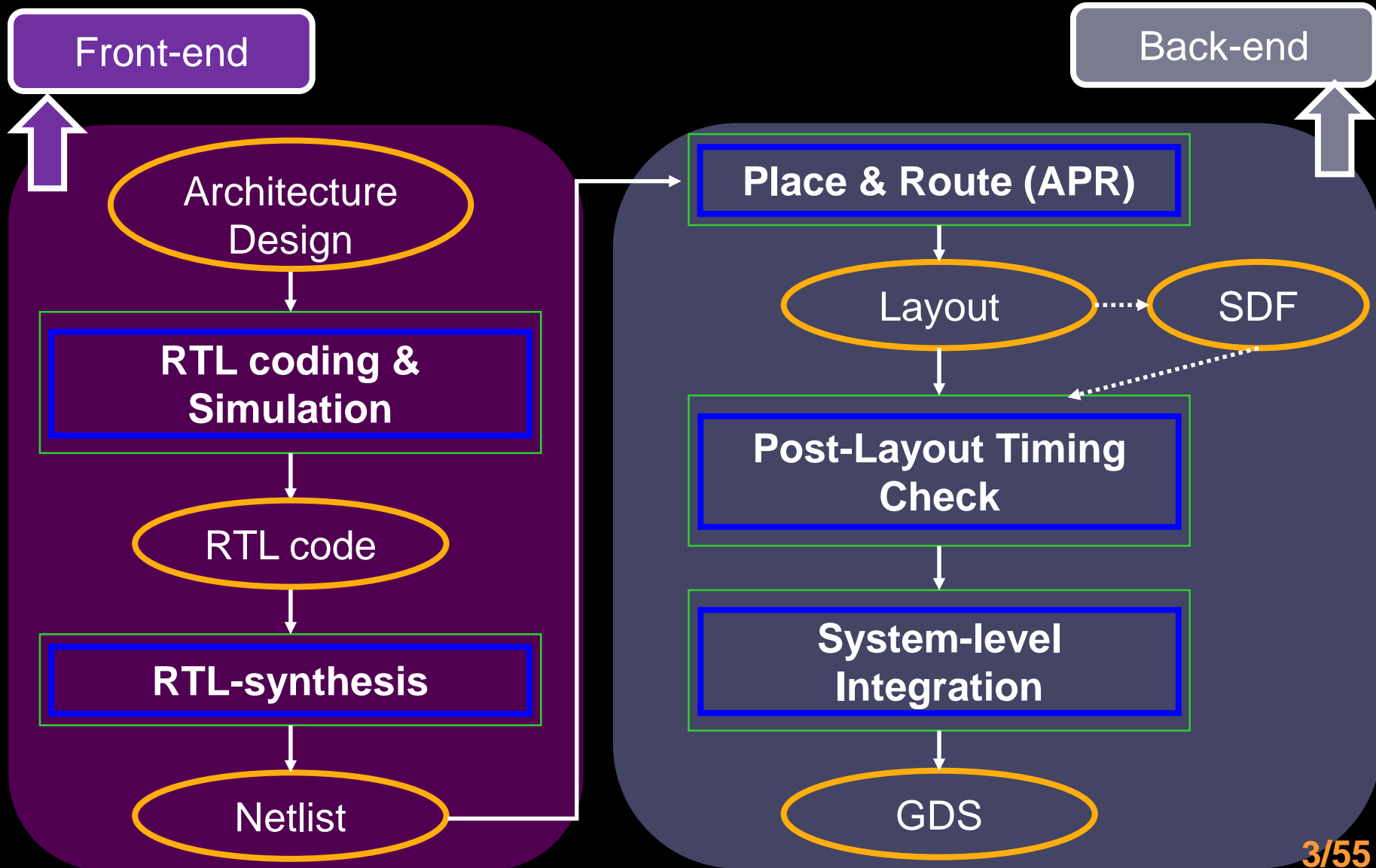
## **Front-end: Synthesis**

**CAD Lab.**

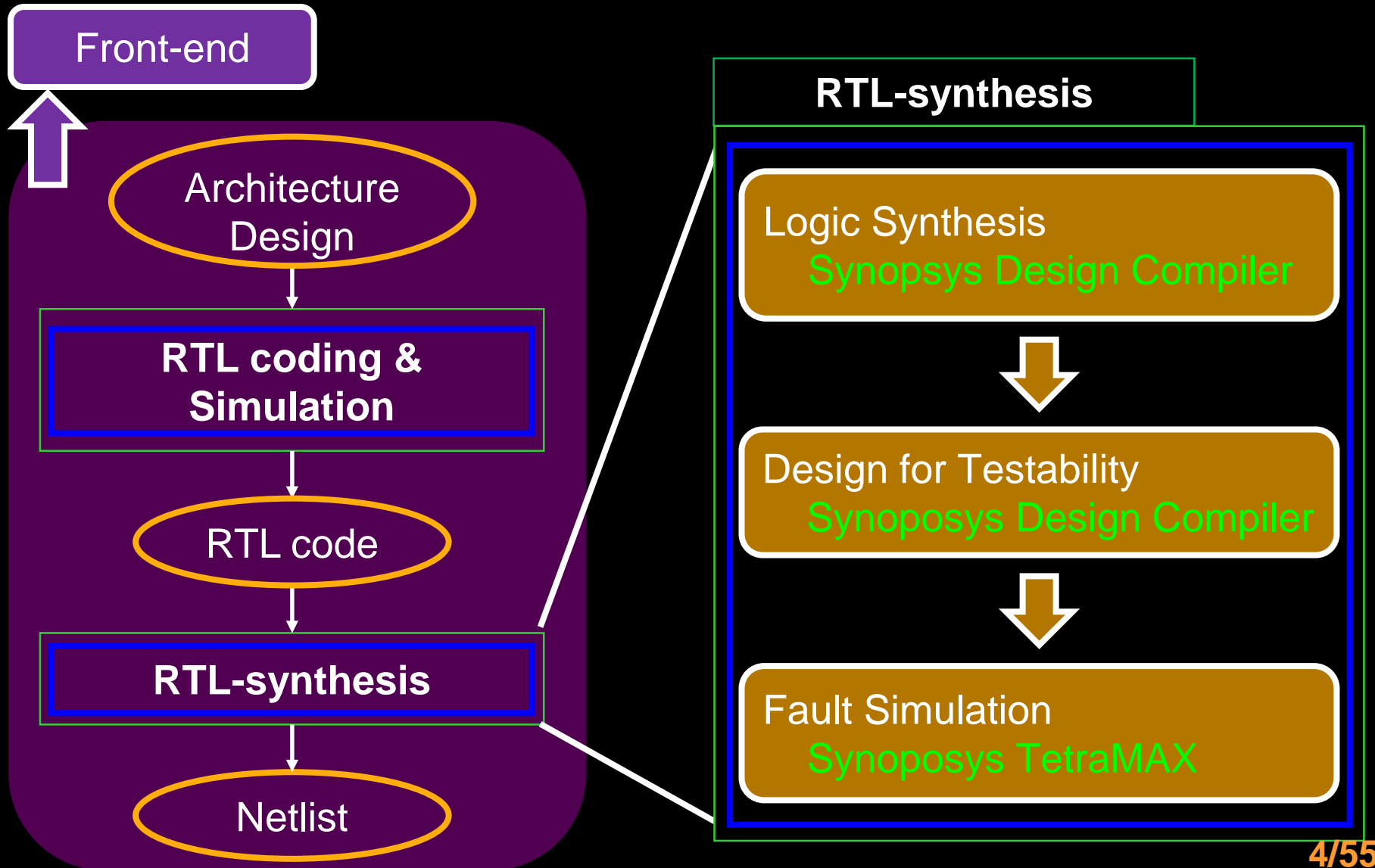
# Outline

- **IC Design Flow**
  - Cell-based Design Flow (Front-end)
  - Environment Setup
- **Logic Synthesis with Design Compiler**
- **Design for Testability with Design Compiler**
- **Fault Simulation with TetraMAX**

# Cell-based Design Flow



# Cell-based Design Flow – Front-end



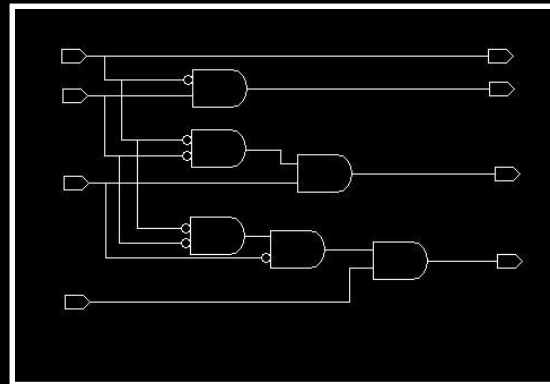
# What is Synthesis

- **Synthesis = translation + optimization + mapping**

```
Residue = 16'h0000;  
If ( high_bits == 2'b10 )  
    residue = state_table[i] ;  
Else state_table[i] = 16'h0000;
```

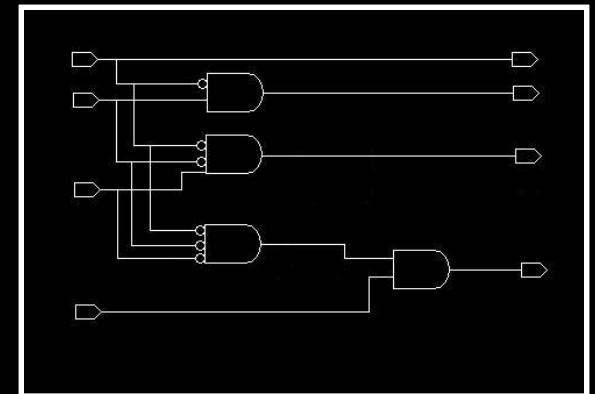
**HDL Source  
( RTL )**

**Translate (HDL Compiler)**



**Generic Boolean  
( GTECH )**

**Optimize + Map  
(Design Compiler)**



**Target Technology**

# Behavioral HDL

```
module lab1(a,b,c,sel,z);  
input [7:0] a,b,c;  
input sel;  
output [8:0] z;  
  
always @(a or b or c) begin  
if (sel) z = a + b;  
else z = a + c;  
End  
  
endmodule
```

# Gate-level HDL (1/2)

```
module lab1_DW01_add_0 ( A, B, CI, SUM, CO );
  input [8:0] A;
  input [8:0] B;
  output [8:0] SUM;
  input CI;
  output CO;
  wire \carry[7] , \carry[6] , \carry[5] , \carry[4] , \carry[3] ,
      \carry[2] , \carry[1] ;

  ADDFX2 U1_7 ( .A(A[7]), .B(B[7]), .CI(\carry[7] ), .CO(SUM[8]), .S(SUM[7])
  );
  ADDFX2 U1_6 ( .A(A[6]), .B(B[6]), .CI(\carry[6] ), .CO(\carry[7] ), .S(
  SUM[6]) );
  ADDFX2 U1_5 ( .A(A[5]), .B(B[5]), .CI(\carry[5] ), .CO(\carry[6] ), .S(
  SUM[5]) );
  ADDFX2 U1_4 ( .A(A[4]), .B(B[4]), .CI(\carry[4] ), .CO(\carry[5] ), .S(
  SUM[4]) );
  ADDFX2 U1_3 ( .A(A[3]), .B(B[3]), .CI(\carry[3] ), .CO(\carry[4] ), .S(
  SUM[3]) );
  ADDFX2 U1_2 ( .A(A[2]), .B(B[2]), .CI(\carry[2] ), .CO(\carry[3] ), .S(
  SUM[2]) );
  ADDFX2 U1_1 ( .A(A[1]), .B(B[1]), .CI(\carry[1] ), .CO(\carry[2] ), .S(
  SUM[1]) );
  AND2X1 U1 ( .A(A[0]), .B(B[0]), .Y(\carry[1] ) );
  XOR2X1 U2 ( .A(B[0]), .B(A[0]), .Y(SUM[0]) );
endmodule
```

## Gate-level HDL (2/2)

```
module lab1 ( a, b, c, sel, z );
  input [7:0] a;
  input [7:0] b;
  input [7:0] c;
  output [8:0] z;
  input sel;
  wire \U1/U1/Z_0 , \U1/U1/Z_1 , \U1/U1/Z_2 , \U1/U1/Z_3 , \U1/U1/Z_4 ,
        \U1/U1/Z_5 , \U1/U1/Z_6 , \U1/U1/Z_7 ;

  lab1_DW01_add_0 r248 ( .A({1'b0, a}), .B({1'b0, \U1/U1/Z_7 , \U1/U1/Z_6 ,
        \U1/U1/Z_5 , \U1/U1/Z_4 , \U1/U1/Z_3 , \U1/U1/Z_2 , \U1/U1/Z_1 ,
        \U1/U1/Z_0 }), .CI(1'b0), .SUM(z) );
  MX2X1 U20 ( .A(c[7]), .B(b[7]), .S0(sel), .Y(\U1/U1/Z_7 ) );
  MX2X1 U21 ( .A(c[6]), .B(b[6]), .S0(sel), .Y(\U1/U1/Z_6 ) );
  MX2X1 U22 ( .A(c[5]), .B(b[5]), .S0(sel), .Y(\U1/U1/Z_5 ) );
  MX2X1 U23 ( .A(c[4]), .B(b[4]), .S0(sel), .Y(\U1/U1/Z_4 ) );
  MX2X1 U24 ( .A(c[3]), .B(b[3]), .S0(sel), .Y(\U1/U1/Z_3 ) );
  MX2X1 U25 ( .A(c[2]), .B(b[2]), .S0(sel), .Y(\U1/U1/Z_2 ) );
  MX2X1 U26 ( .A(c[1]), .B(b[1]), .S0(sel), .Y(\U1/U1/Z_1 ) );
  MX2X1 U27 ( .A(c[0]), .B(b[0]), .S0(sel), .Y(\U1/U1/Z_0 ) );
endmodule
```

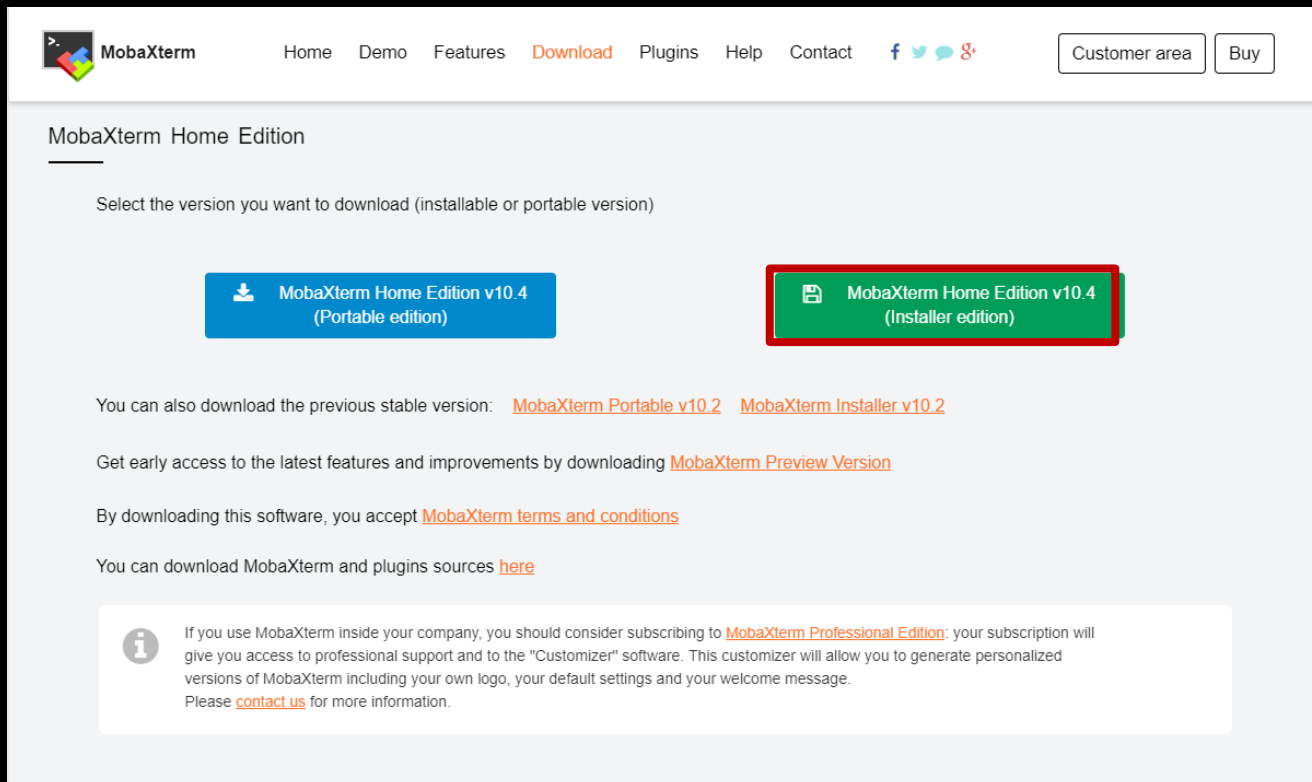


# Outline

- **IC Design Flow**
  - Cell-based Design Flow (Front-end)
  - **Environment Setup**
- **Logic Synthesis with Design Compiler**
- **Design for Testability with Design Compiler**
- **Fault Simulation with TetraMAX**

# MobaXterm (1/4)


- <https://mobaxterm.mobatek.net/download-home-edition.html>
- Download installer edition




The screenshot shows the MobaXterm website's download page for the Home Edition. The navigation bar includes links for Home, Demo, Features, Download (highlighted in red), Plugins, Help, and Contact, along with social media icons and buttons for 'Customer area' and 'Buy'. The main heading is 'MobaXterm Home Edition'. Below it, a prompt asks the user to 'Select the version you want to download (installable or portable version)'. Two buttons are presented: a blue button for 'MobaXterm Home Edition v10.4 (Portable edition)' and a green button for 'MobaXterm Home Edition v10.4 (Installer edition)', which is highlighted with a red border. Further down, links are provided for previous stable versions (v10.2) and a preview version. A disclaimer states that downloading the software implies acceptance of the terms and conditions. At the bottom, an information box suggests subscribing to the Professional Edition for company use, offering professional support and customization options.

MobaXterm Home Edition

Select the version you want to download (installable or portable version)

 MobaXterm Home Edition v10.4 (Portable edition)


 MobaXterm Home Edition v10.4 (Installer edition)

You can also download the previous stable version: [MobaXterm Portable v10.2](#) [MobaXterm Installer v10.2](#)

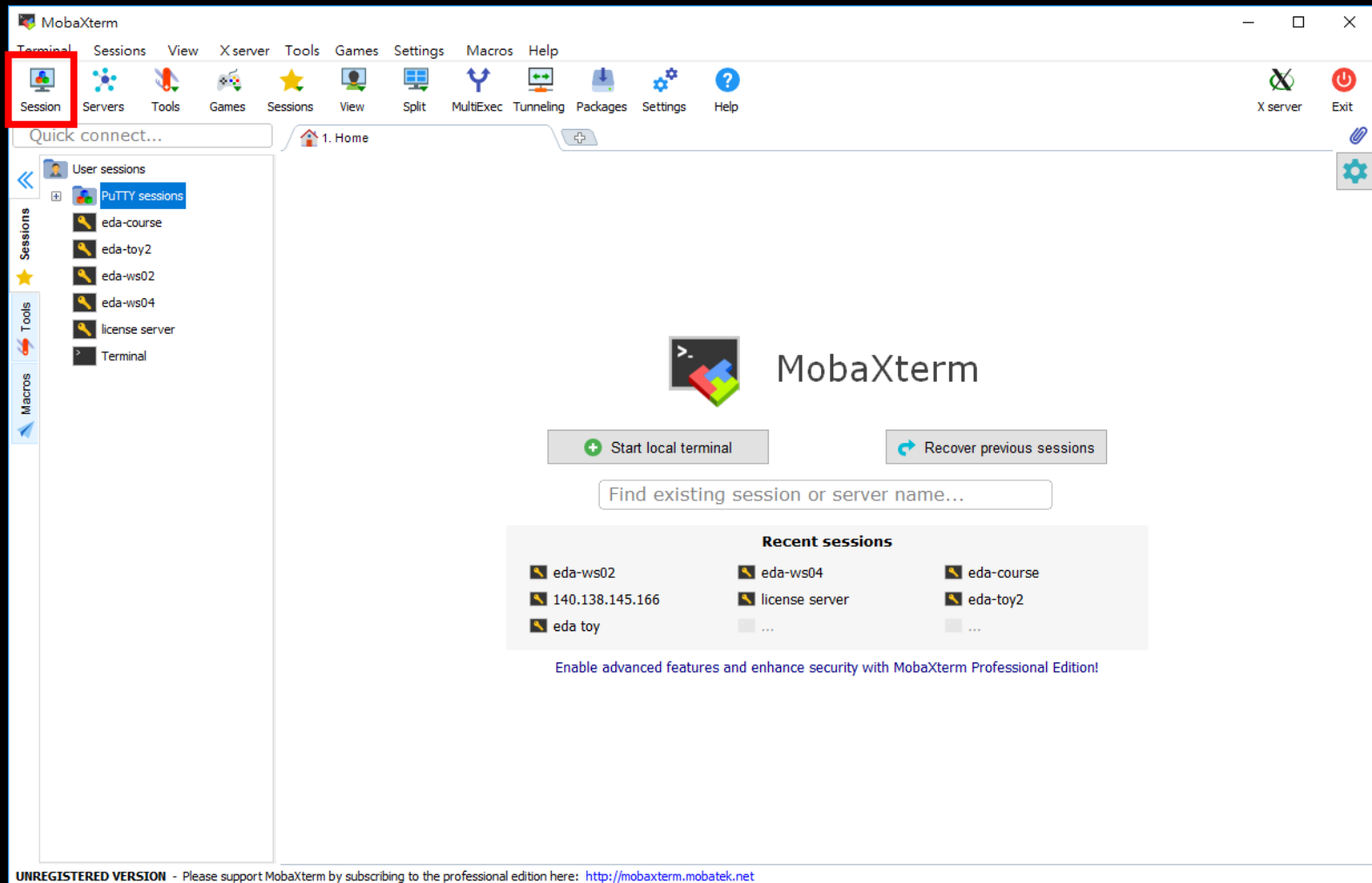
Get early access to the latest features and improvements by downloading [MobaXterm Preview Version](#)

By downloading this software, you accept [MobaXterm terms and conditions](#)

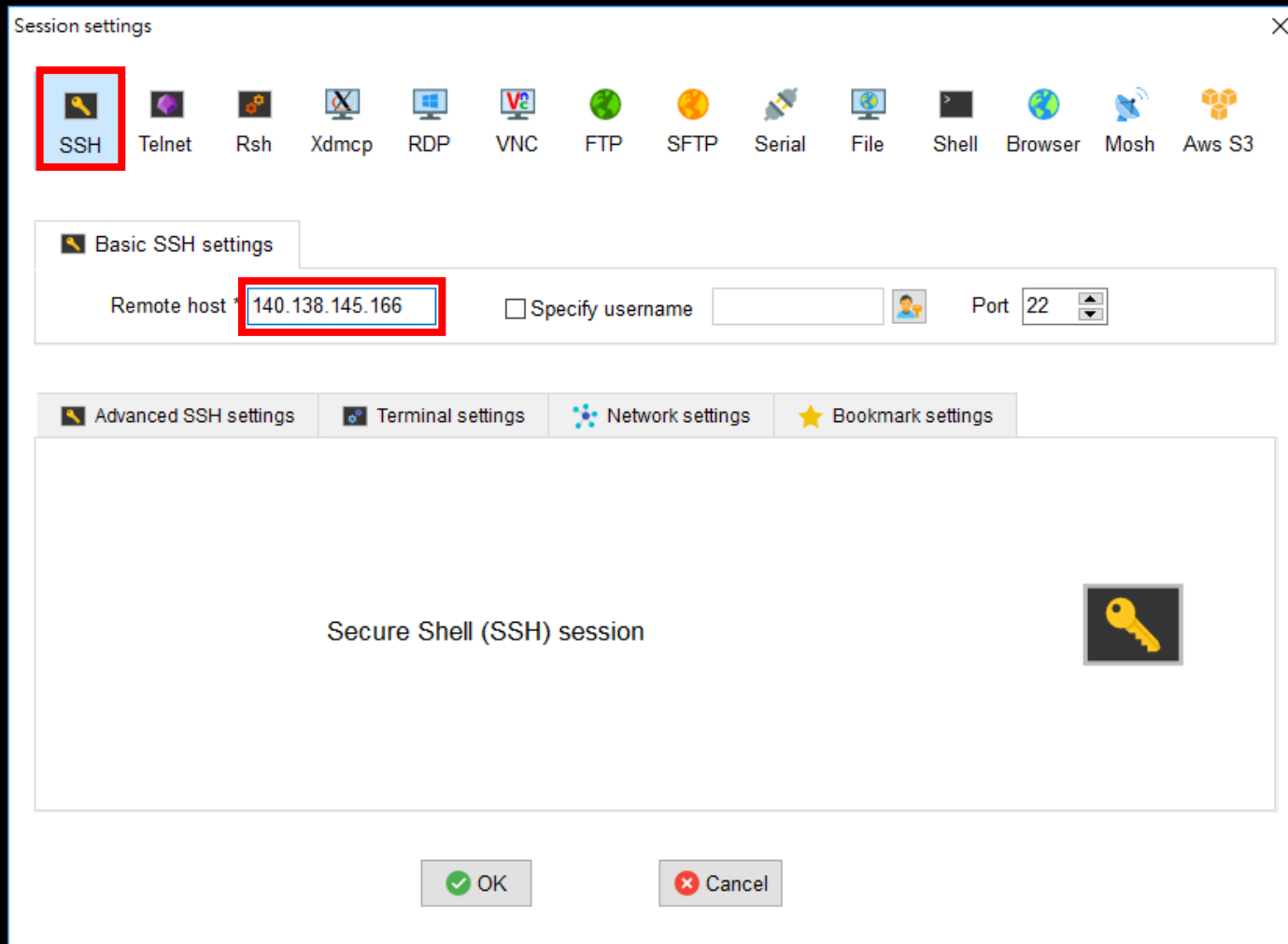
You can download MobaXterm and plugins sources [here](#)

 If you use MobaXterm inside your company, you should consider subscribing to [MobaXterm Professional Edition](#): your subscription will give you access to professional support and to the "Customizer" software. This customizer will allow you to generate personalized versions of MobaXterm including your own logo, your default settings and your welcome message. Please [contact us](#) for more information.

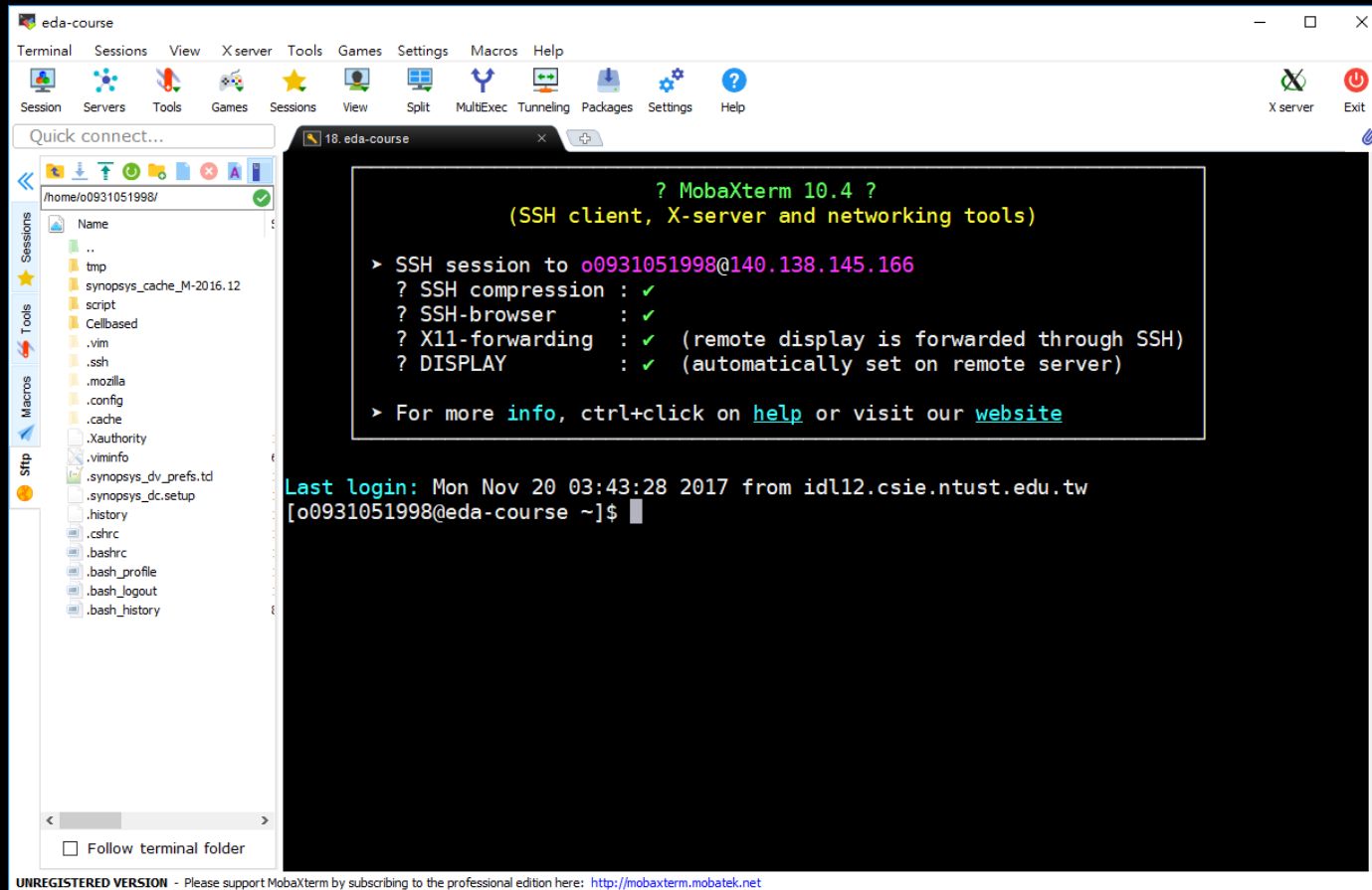
# MobaXterm (2/4)



# MobaXterm (3/4)



# MobaXterm (4/4)



# Linux Commands

- **\$> cd: change directory**
- **\$> ls: list directory contents**
- **\$> mv: move (rename) files**
- **\$> cp: copy files and directories**
- **\$> man: manual page**
  - **\$> man cd**
  - **\$> man ls**

# Linux Commands (cont.)

- **\$> pwd: print name of current/working directory**
  - . → current directory
  - .. → parent directory
  - ~ → home directory

# Environment Setting

- **CAD Tool (Synopsys - Design Vision)**
  - `$> csh`
  - `$> source /usr/cad/synopsys/CIC/synthesis.cshrc`
  - `$> source /usr/cad/synopsys/CIC/tmax.cshrc`
- **Directory and files**
  - `$> cp /tmp/Cellbased_frontend_32nm.tar .`
  - `$> tar xvf Cellbased_frontend_32nm.tar`
  - `$> cd Cellbased_frontend_32nm`
  - `$> cd run/`

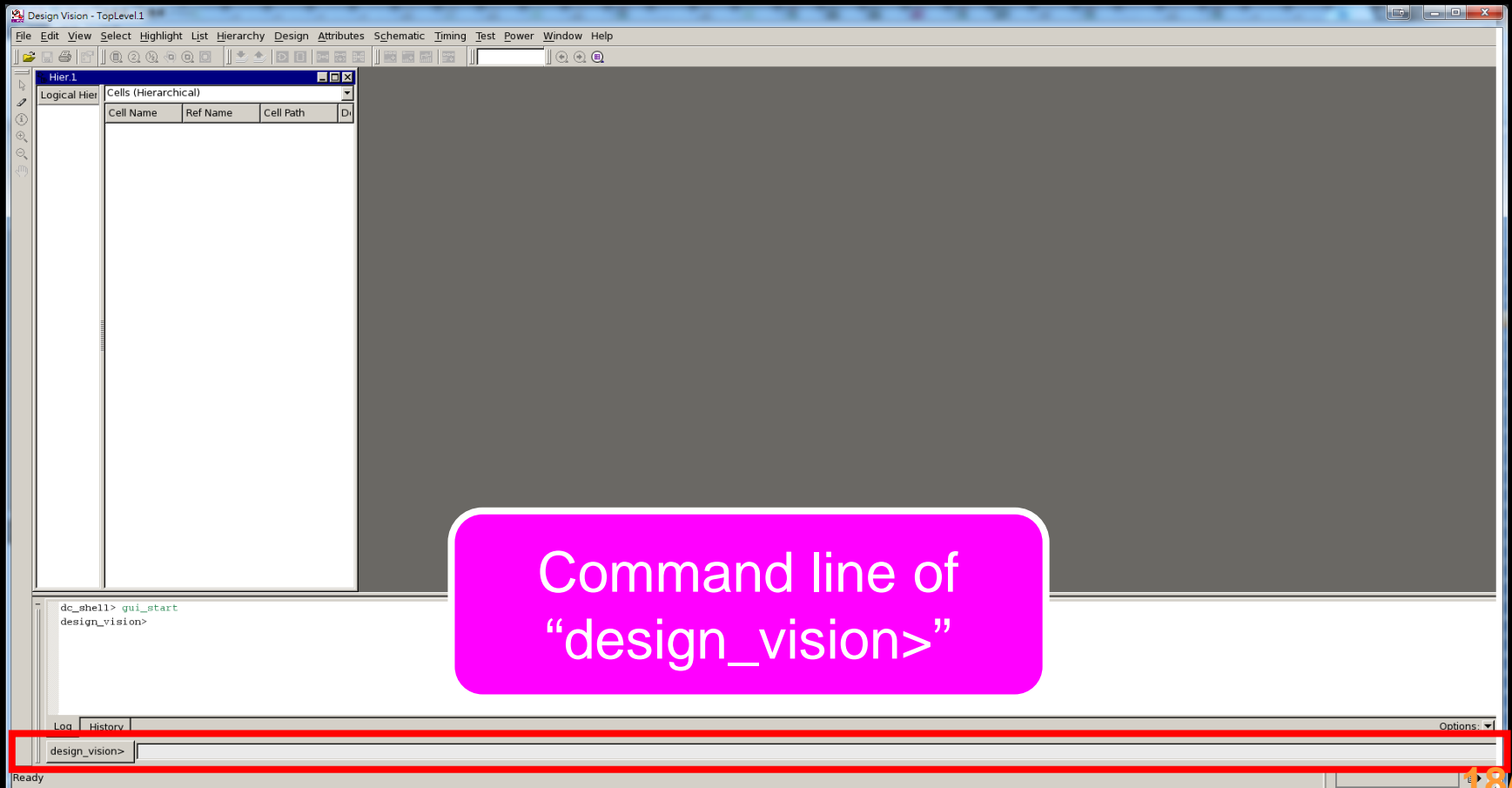


# Outline

- **IC Design Flow**
  - Cell-based Design Flow (Front-end)
  - Environment Setup
- **Logic Synthesis with Design Compiler**
- **Design for Testability with Design Compiler**
- **Fault Simulation with TetraMAX**

# GUI Interface of Design Compiler

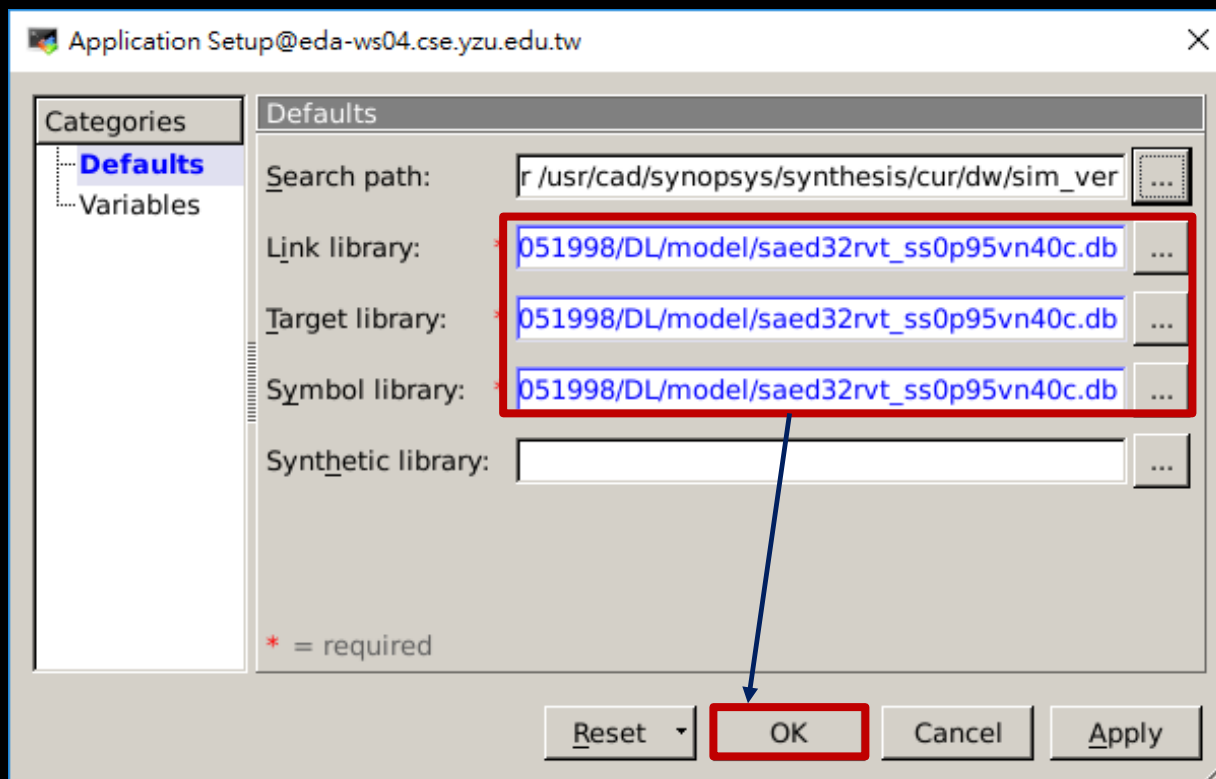
- GUI of Design Compiler is called Design Vision
- `$> dv`



# Design Vision Setup (1/2)

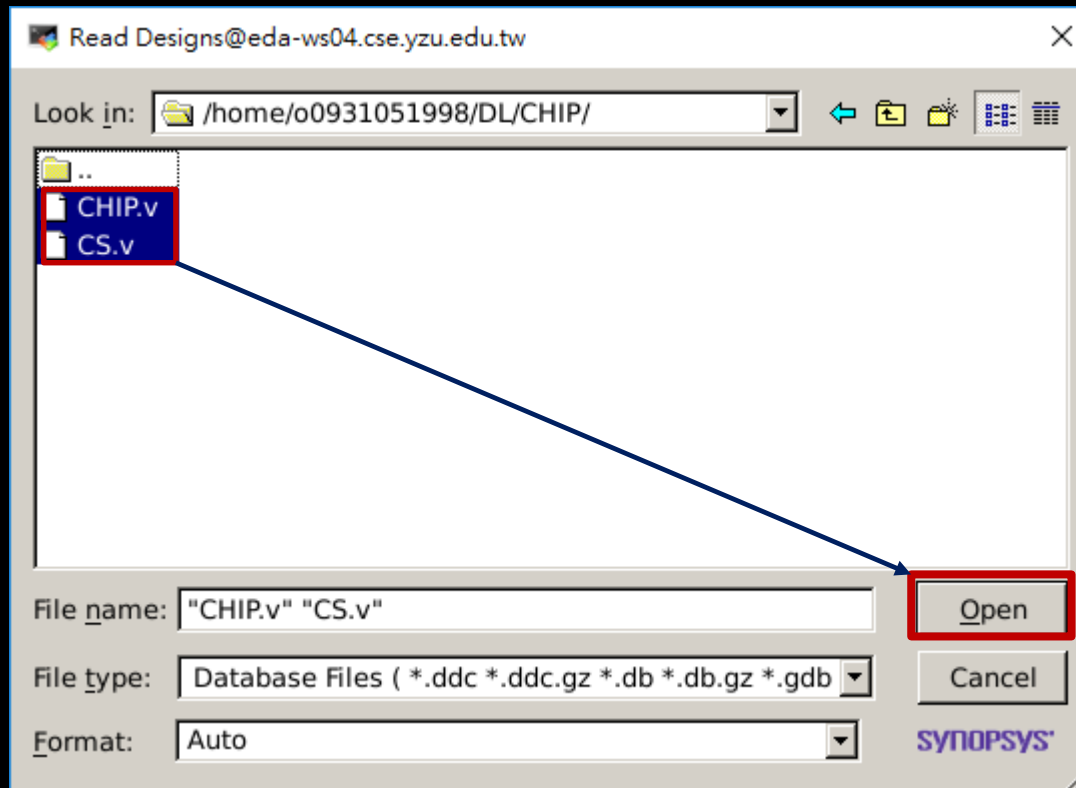
- File → Setup
- Delete the default model “\*”, “your\_library.db”, “your\_library.sdb”
- Choose the following four models for **link library**, **target library**, and **symbol library**
  - saed32io\_wb\_ff1p16v25c\_2p75v.db
  - saed32io\_wb\_ss0p95vn40c\_2p25v.db
  - saed32rvt\_ff1p16v25c.db
  - saed32rvt\_ss0p95vn40c.db
- Press “OK”

# Design Vision Setup (2/2)

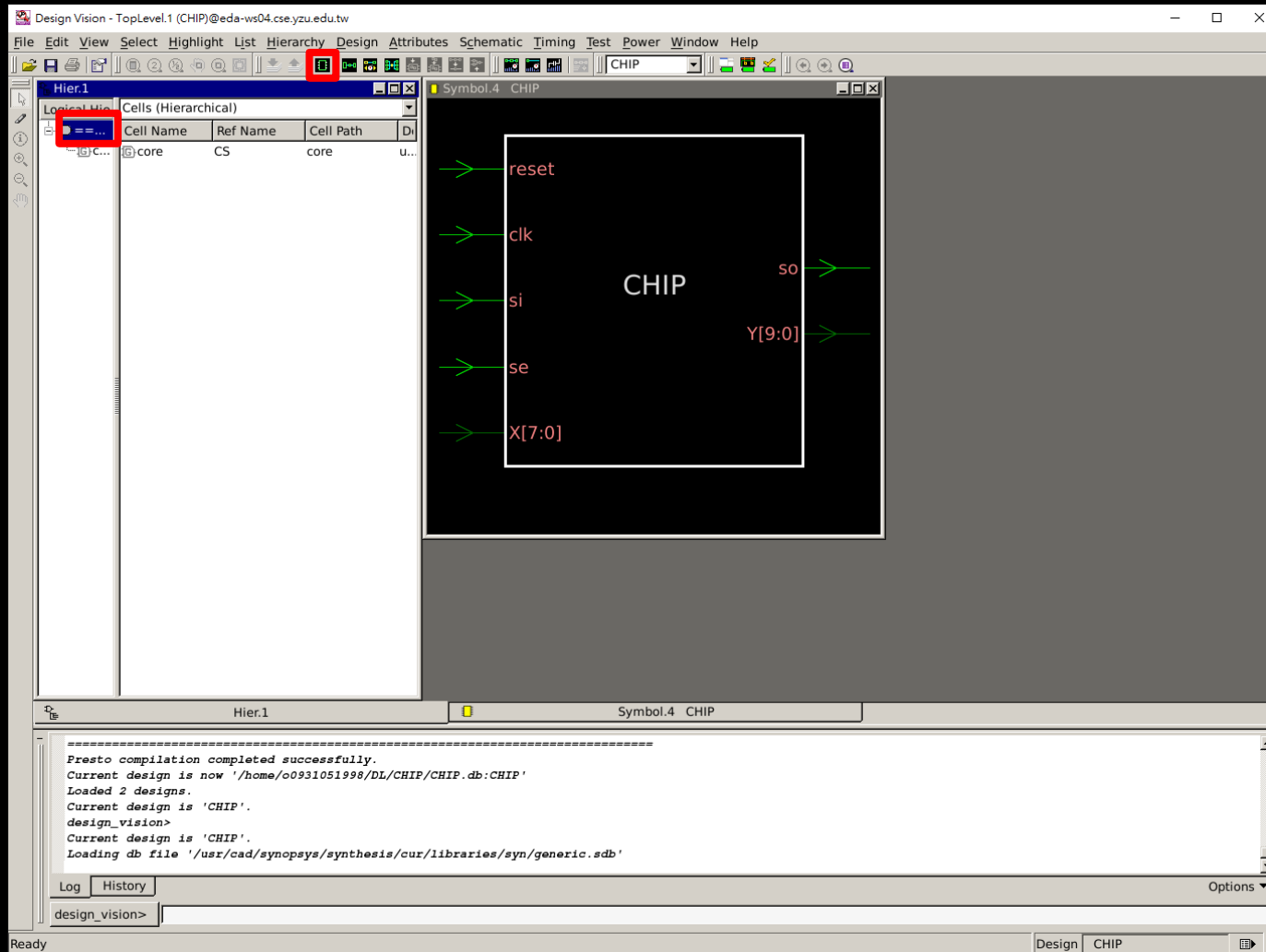


# Read File

- File → Read

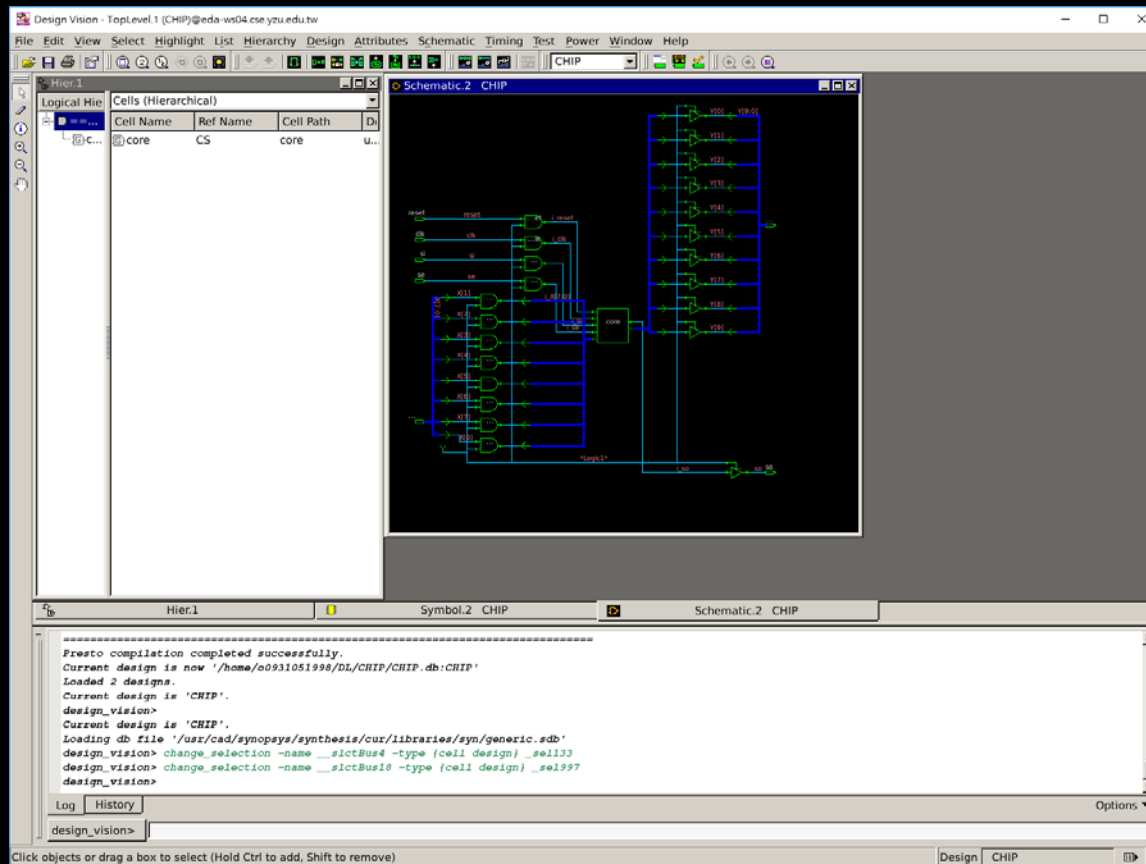


# Symbol View



# Schematic View

- Double click the symbol view



# Setting Operating Condition (1/2)

- **Attributes → Operating Environment → Operating Conditions**
- **Select “Min/max case”**
- **Set **Maximum** operating condition**
  - Library: select “saed32rvt\_ss0p95vn40c”
  - Condition: select “ss0p95vn40c”
- **Set **Minimum** operating condition**
  - Library: select “saed32rvt\_ff1p16v25c”
  - Condition: select “ff1p16v25c”



# Setting Operating Condition (2/2)

Operating Conditions@eda-course

Current design:

Analysis condition

☐ Single ☒ Min/max case

Maximum operating condition

Library:

Condition:

Minimum operating condition

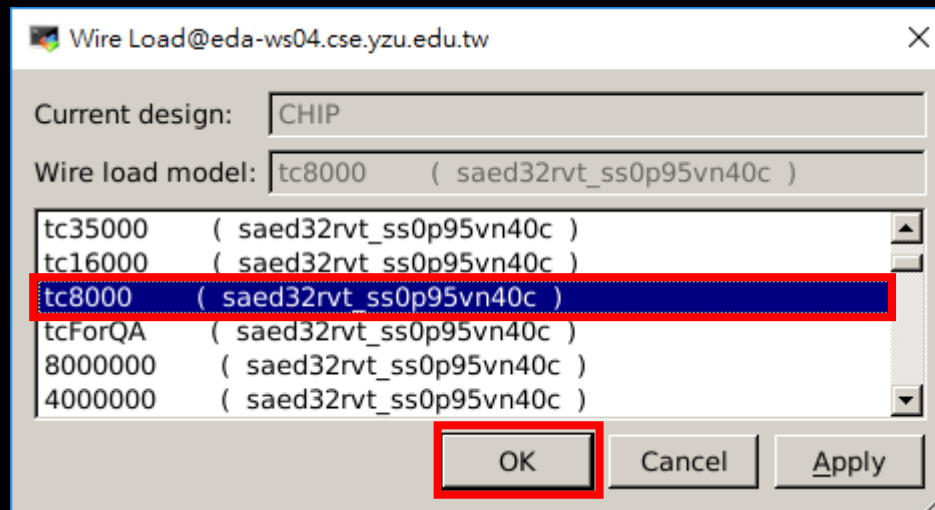
Library:

Condition:

OK Cancel Apply

# Setting Wire Load

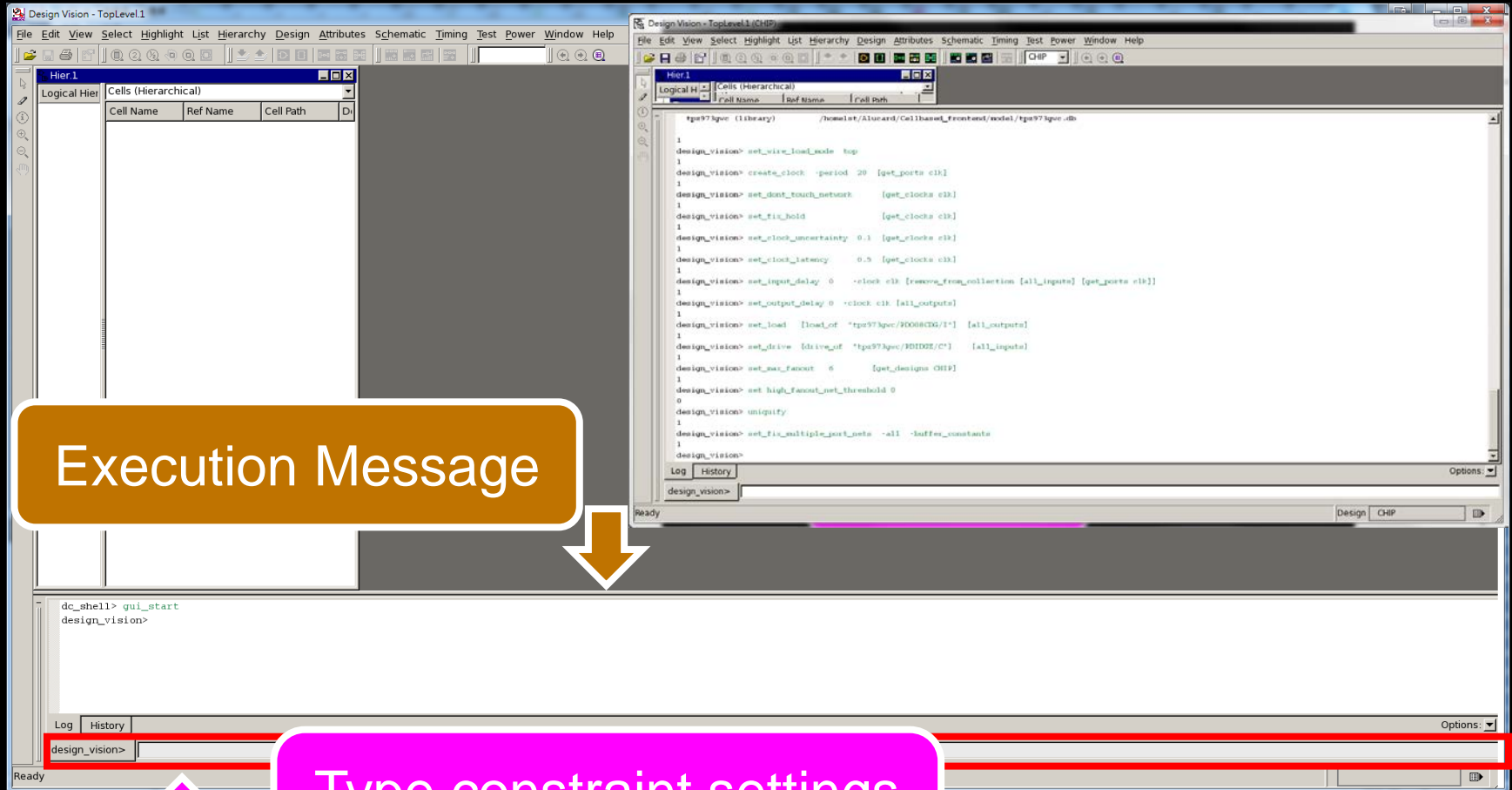
- Attributes → Operating Environment → Wire Load
- Select “tc8000 (saed32rvt\_ss0p95vn40c)”



# Setup Constraints (Command)

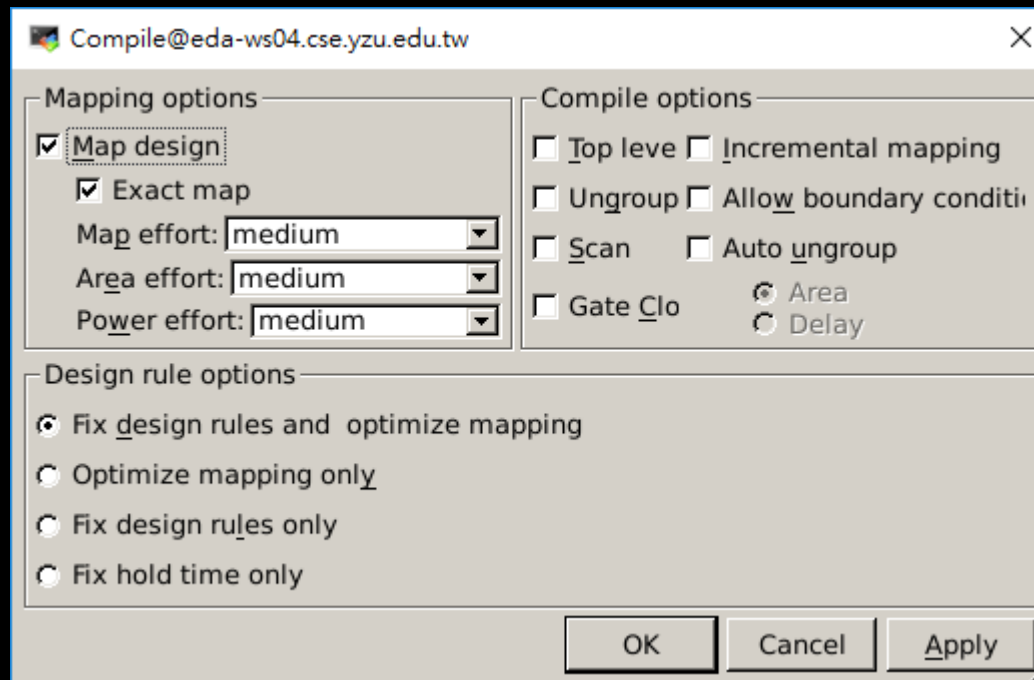
- **design\_vision>**
  - current\_design [get\_designs **CHIP**]
  - link
  - set\_wire\_load\_mode top
  - create\_clock -period 20 [get\_ports **clk**]
  - set\_dont\_touch\_network [get\_clocks **clk**]
  - set\_fix\_hold [get\_clocks **clk**]
  - set\_clock\_uncertainty 0.1 [get\_clocks **clk**]
  - set\_clock\_latency 0.5 [get\_clocks **clk**]
  - set\_input\_delay 0 -clock **clk** [remove\_from\_collection [all\_inputs] [get\_ports **clk**]]
  - set\_output\_delay 0 -clock **clk** [all\_outputs]
  - set\_load [load\_of "saed32io\_wb\_ss0p95vn40c\_2p25v/D8I1025\_EW/DIN"] [all\_outputs]
  - set\_drive [drive\_of "saed32io\_wb\_ss0p95vn40c\_2p25v/I1025\_NS/DOUT"] [all\_inputs]
  - set\_max\_fanout 6 [get\_designs **CHIP**]
  - set\_high\_fanout\_net\_threshold 0
  - uniquify
  - set\_fix\_multiple\_port\_nets -all -buffer\_constants

# How to Setup Constraints



# Compile Design

- Optimizes and maps the current\_design
- Design → Compile Design
  - default

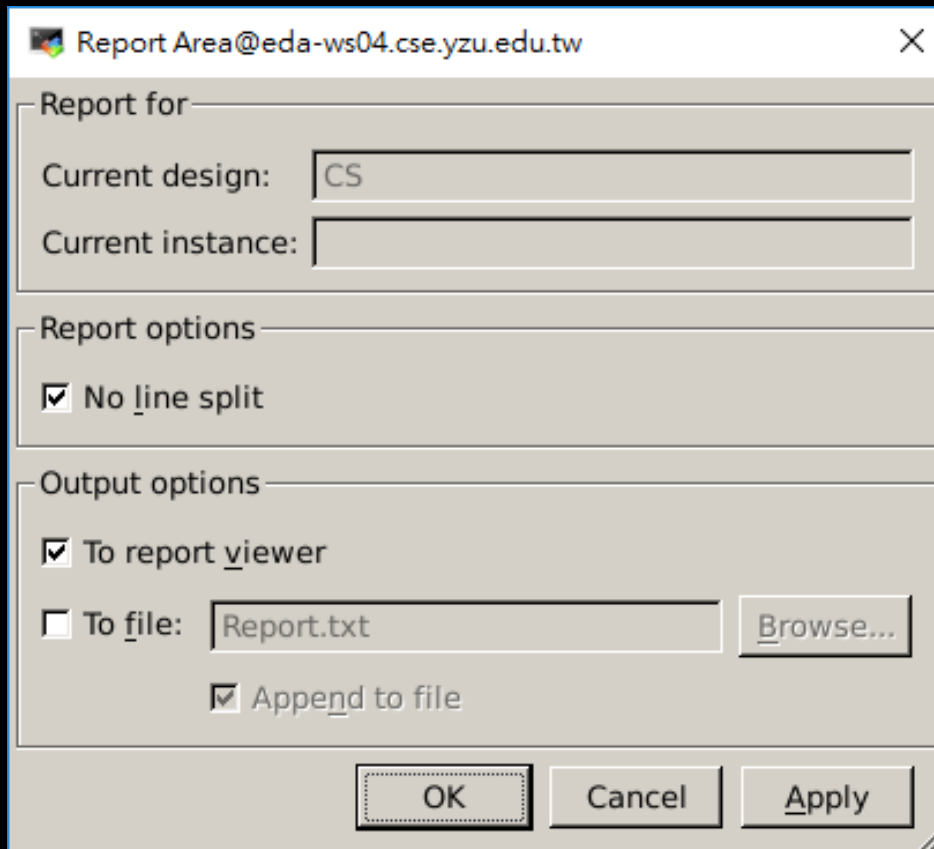


# Report Timing & Area

- **Report**
  - Area
  - Timing

# Report Area (1/2)

- Design → Report Area → OK



The screenshot shows a dialog box titled "Report Area@eda-ws04.cse.yzu.edu.tw". It contains three main sections: "Report for", "Report options", and "Output options".

**Report for:**

- Current design: CS
- Current instance: (empty text box)

**Report options:**

- ☒ No line split

**Output options:**

- ☒ To report viewer
- ☐ To file: Report.txt (with a "Browse..." button next to it)
- ☒ Append to file

At the bottom, there are three buttons: "OK", "Cancel", and "Apply". The "OK" button is highlighted with a dashed border.

# Report Area (2/2)

The screenshot shows the Design Vision software interface. The main window displays a report titled "Report : area". The report content is as follows:

```
*****
Report : area
Design : CS
Version: H-2013.03-SP5
Date   : Mon Sep 11 17:39:06 2017
*****

Library(s) Used:

saed32rvt_fflp16v25c (File: /home/o0931051998/DL/model/saed32rvt_fflp16v25c.db)

Number of ports:      23
Number of nets:      1040
Number of cells:      845
Number of combinational cells: 672
Number of sequential cells: 173
Number of macros/black boxes: 0
Number of buf/inv:    199
Number of references: 34

Combinational area:   1959.704374
Buf/Inv area:         428.486788
Noncombinational area: 1582.808871
Macro/Black Box area: 0.000000
Net Interconnect area: 758.160472

Total cell area:      3542.513246
Total area:           4300.673718

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

A red rectangular box highlights the statistics section of the report, from "Number of ports:" to "Total area:". Below the main report window, there is a "Report.1" tab. The content of this tab is a summary of the statistics:

```
Number of references:      34

Combinational area:        1959.704374
Buf/Inv area:              428.486788
Noncombinational area:     1582.808871
Macro/Black Box area:      0.000000
Net Interconnect area:     758.160472

Total cell area:           3542.513246
Total area:                4300.673718
design_vision>
```

At the bottom of the interface, there is a "Log" button and a "History" button. The status bar at the very bottom indicates "Ready".



# Report Timing (1/2)

- Timing → Report Timing Path → OK

Report Timing Paths@eda-ws04.cse.yzu.edu.tw

From: pin [ ] Selection[1]

Through: pin [ ] Selection[2]

To: pin [ ] Selection[3]

Report options

Worst paths per endpoint: 1 Maximum path delay: [ ]

Max paths per group: 1 Minimum path delay: [ ]

Path type: full [ ]

Delay type: max [ ]

Sort by: group [ ]

Significant digits: 2 [ ]

☐ Report timing loops

☐ Justify paths with input vector

☐ Find true path

Path delay threshold: 0 [ ]

☐ No line split

☐ Enable asynchronous arcs

☐ Show nets in combinational path

☐ Show net transition time

☐ Show input pins in combinational path

☐ Show net capacitance

☐ Show dont\_touch, size\_only attributes for nets and cells

Output options

☒ To report viewer

☐ To file: Report.txt [ ] Browse...

☒ Append to file

OK Cancel Apply

# Report Timing (2/2)

Design Vision - TopLevel.1 (CHIP) - [Report.2 - Timing]@eda-ws04.cse.yzu.edu.tw

File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power Window Help

CHIP

core/U469/Y (AO222X1_RVT)	0.05	3.04 r
core/U471/Y (AO222X1_RVT)	0.05	3.09 r
core/U56/Y (AO222X1_RVT)	0.05	3.14 r
core/U57/Y (AO222X1_RVT)	0.04	3.18 r
core/U527/Y (MUX21X1_RVT)	0.04	3.21 f
core/U532/Y (HAND2X0_RVT)	0.02	3.23 r
core/U533/Y (AO222X1_RVT)	0.04	3.27 r
core/U534/Y (AO222X1_RVT)	0.05	3.33 r
core/intadd_10/U10/CO (FADDX1_RVT)	0.04	3.36 r
core/intadd_10/U9/CO (FADDX1_RVT)	0.03	3.40 r
core/intadd_10/U8/CO (FADDX1_RVT)	0.03	3.43 r
core/intadd_10/U7/CO (FADDX1_RVT)	0.03	3.46 r
core/intadd_10/U6/CO (FADDX1_RVT)	0.03	3.49 r
core/intadd_10/U5/CO (FADDX1_RVT)	0.03	3.53 r
core/intadd_10/U4/CO (FADDX1_RVT)	0.03	3.56 r
core/intadd_10/U3/CO (FADDX1_RVT)	0.03	3.59 r
core/intadd_10/U2/S (FADDX1_RVT)	0.05	3.65 f
core/U67/Y (INVX4_RVT)	0.02	3.67 r
core/Y[8] (CS)	0.00	3.67 r
opad_Y8/PADIO (D8I1025_EW)	0.79	4.46 r
Y[8] (out)	0.00	4.46 r
data arrival time		4.46
clock clk (rise edge)	20.00	20.00
clock network delay (ideal)	0.50	20.50
clock uncertainty	-0.10	20.40
output external delay	0.00	20.40
data required time		20.40
data required time		20.40
data arrival time		-4.46
-----		
slack (MET)		15.94

\*\*\*\*\* End Of Report \*\*\*\*\*

Hier.1 Report.2

clock uncertainty	-0.10	20.40
output external delay	0.00	20.40
data required time		20.40
-----		
data required time		20.40
data arrival time		-4.46
-----		
slack (MET)		15.94

design\_vision>

Log History Options

design\_vision>

Ready

# Modify Timing Constrain

- **design\_vision>**
  - create\_clock -period 10 [get\_ports clk]
- **Compile Design again and observe the difference.**
  - Tradeoff timing and area.
  - Let the SLACK MET.

# Save File – Gate-level Netlist

- **File → Save as**
  - Type the file\_name (ex: **CHIP\_synth.v**)
  - Choose Verilog as file format
- **design\_vision>**
  - write\_sdc -version 1.2 **CHIP\_synth.sdc**

# Outline

- **IC Design Flow**
  - Cell-based Design Flow (Front-end)
  - Environment Setup
- **Logic Synthesis with Design Compiler**
- **Design for Testability with Design Compiler**
- **Fault Simulation with TetraMAX**

# Create Test Protocol (1/2)

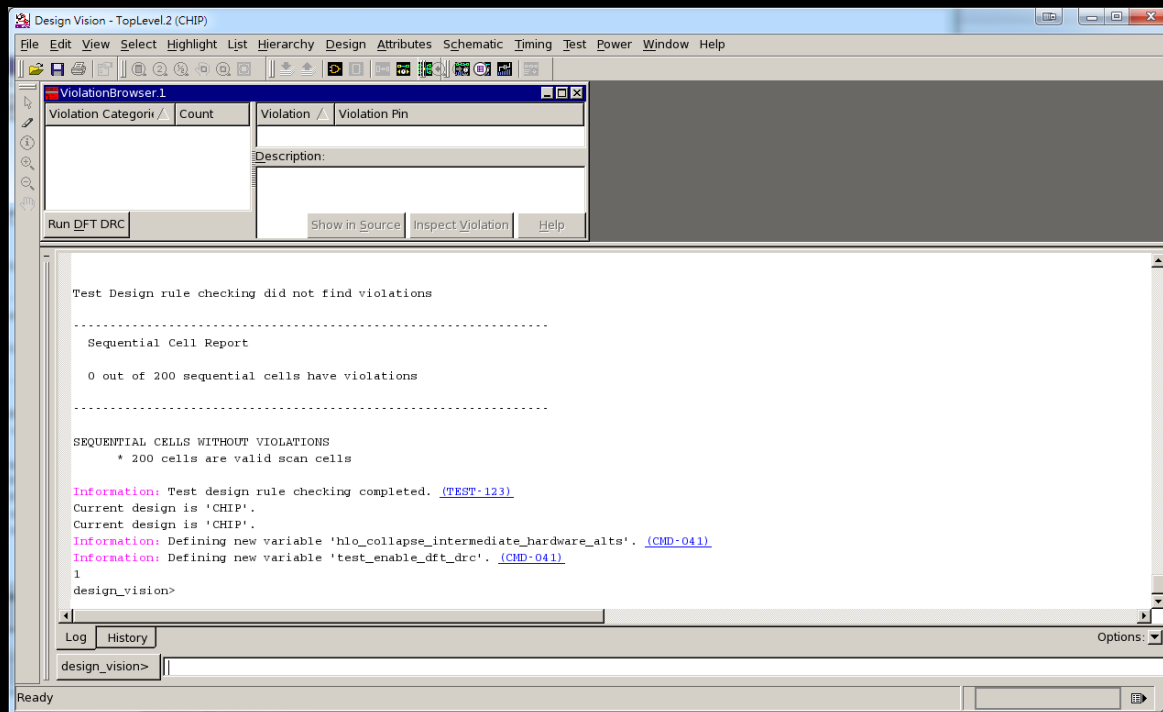
- **design\_vision>**
  - current\_design [get\_designs CS]
  - set\_scan\_configuration -style multiplexed\_flip\_flop
  - set\_test\_default\_period 100
  - set\_dft\_signal -view existing\_dft -type ScanClock -timing {45 55} -port clk
  - set\_dft\_signal -view existing -type Reset -active\_state 1 -port reset
  - set\_scan\_configuration -chain\_count 1
  - set\_scan\_configuration -clock\_mixing no\_mix
  - set\_dft\_signal -view spec -type ScanDataIn -port si

## Create Test Protocol (2/2)

- **design\_vision>**
  - set\_dft\_signal -view spec -type ScanDataOut  
-port **so**
  - set\_dft\_signal -view spec -type ScanEnable  
-port **se** -active\_stat 1
  - set\_scan\_path **chain1** -scan\_data\_in **si**  
-scan\_data\_out **so**
  - create\_test\_protocol

# DFT Check (1/2)

- Pre-DFT DRC
- Check scan design rule before scan chain synthesis
- `design_vision>`
  - `dft_drc`





# DFT Check (2/2)

DRC Report

Total violations: 0

Test Design rule checking did not find violations

Sequential Cell Report

0 out of 200 sequential cells have violations

SEQUENTIAL CELLS WITHOUT VIOLATIONS

\* 200 cells are valid scan cells

# Scan Preview

- Check scan-path consistency
- Determine the chain count
- Allocate and orders scan cells
- Add connecting hardware
  - design\_vision>
    - preview\_dft -show all
    - preview\_dft -test\_points all

# Scan Preview Report

```
preview_dft --show all
```

```
Number of chains: 1
Scan methodology: full_scan
Scan style: multiplexed_flip_flop
Clock domain: no_mix
Scan enable: SE (no hookup pin)
```

```
Scan chain 'chain1' (SI --> SO) contains 200 cells
```

```
***** Test Point Plan Report *****
```

```
Total number of test points : 0
Number of Autofix test points: 0
Number of Wrapper test points: 0
Number of test modes : 0
Number of test point enables : 0
Number of data sources : 0
Number of data sinks : 0
```

```
*****
```

```
No test points.
```

# Scan Chain Synthesis

- Scan replacement and insert test points
- Optimized the logic
  - design\_vision>
  - insert\_dft

```
design_vision> insert_dft
Information: Starting test design rule checking. (TEST-222)
Test Design rule checking did not find violations
Information: Test design rule checking completed. (TEST-123)
Architecting Scan Chains
Routing Scan Chains
Routing Global Signals
Mapping New Logic
Resetting current test mode
Beginning Mapping Optimizations
-----
```

ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL NEG SLACK	DESIGN RULE COST	ENDPOINT	MIN DELAY COST
-----						
Beginning Phase 1 Design Rule Fixing (max_fanout)						
-----						
ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL NEG SLACK	DESIGN RULE COST	ENDPOINT	MIN DELAY COST
-----						
0:09:12	376920.3	0.06	0.1	176.0 se		0.00
0:09:12	376920.3	0.06	0.1	176.0 se		0.00
0:09:12	376920.3	0.06	0.1	176.0 se		0.00
0:09:12	377635.9	0.06	0.1	158.0 net12986		0.00
0:09:12	377635.9	0.06	0.1	158.0 net12986		0.00
0:09:12	377635.9	0.06	0.1	158.0 net12986		0.00
0:09:12	378351.6	0.06	0.1	140.0 net12985		0.00
0:09:12	378351.6	0.06	0.1	140.0 net12985		0.00
0:09:12	378351.6	0.06	0.1	140.0 net12985		0.00
0:09:12	379067.2	0.06	0.1	122.0 net12984		0.00
0:09:12	379067.2	0.06	0.1	122.0 net12984		0.00
0:09:12	379067.2	0.06	0.1	122.0 net12984		0.00
0:09:12	379782.9	0.06	0.1	104.0 net12983		0.00
0:09:12	379782.9	0.06	0.1	104.0 net12983		0.00

# Handoff Design

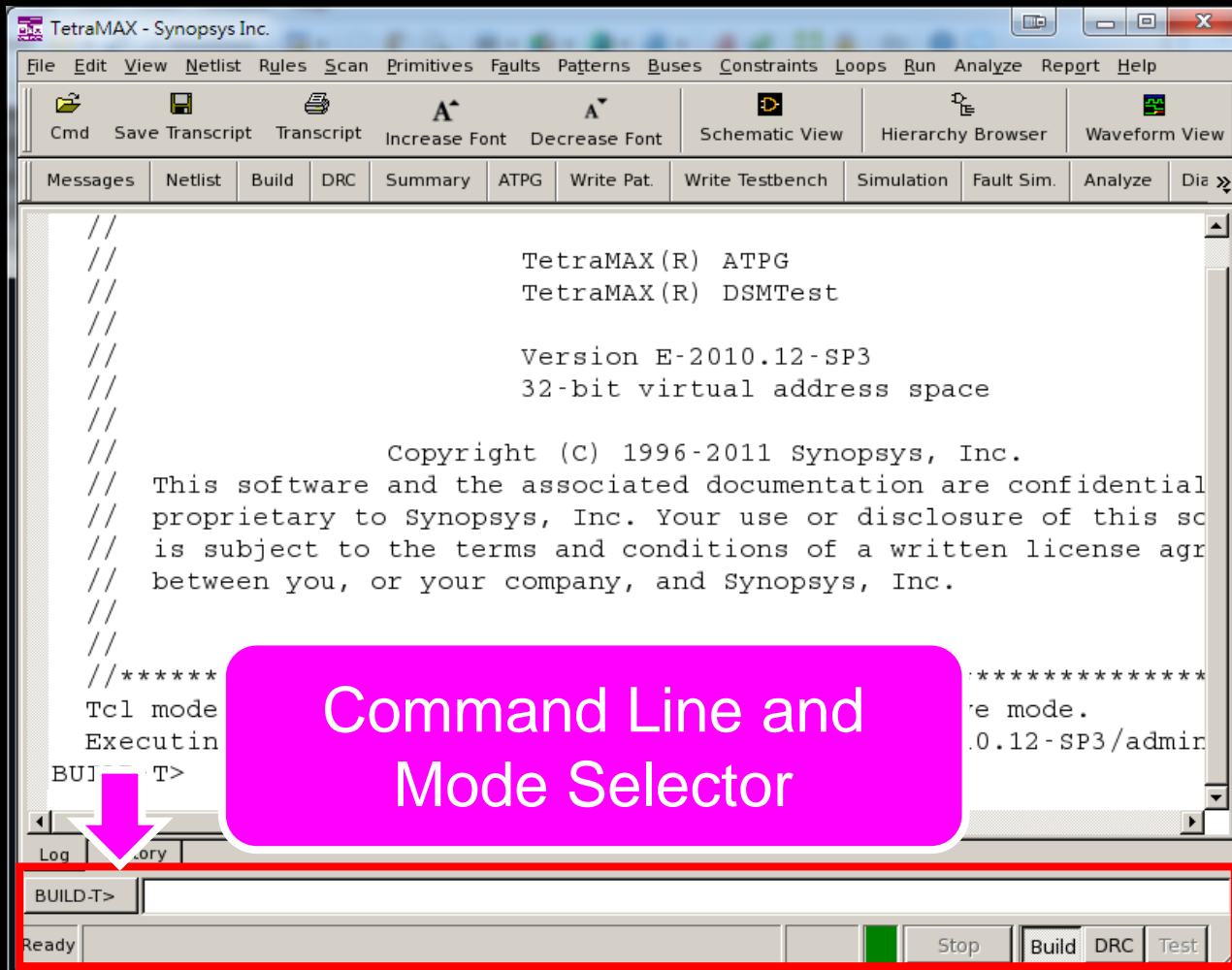
- **Report scan information**
  - design\_vision>
    - report\_scan\_path -view existing\_dft -chain all
    - report\_scan\_path -view existing\_dft -cell all
- **Prepare TetraMax script**
  - design\_vision>
    - change\_names -hierarchy -rule verilog
    - write -format verilog -hierarchy -out CS\_dft.v
    - write -format ddc -hierarchy -output CS.ddc
    - write\_scan\_def -output CS\_scan.def
    - set test\_stil\_netlist\_format verilog
    - write\_test\_protocol -output CS.spf

# Outline

- **IC Design Flow**
  - Cell-based Design Flow (Front-end)
  - Environment Setup
- **Logic Synthesis with Design Compiler**
- **Design for Testability with Design Compiler**
- **Fault Simulation with TetraMAX**

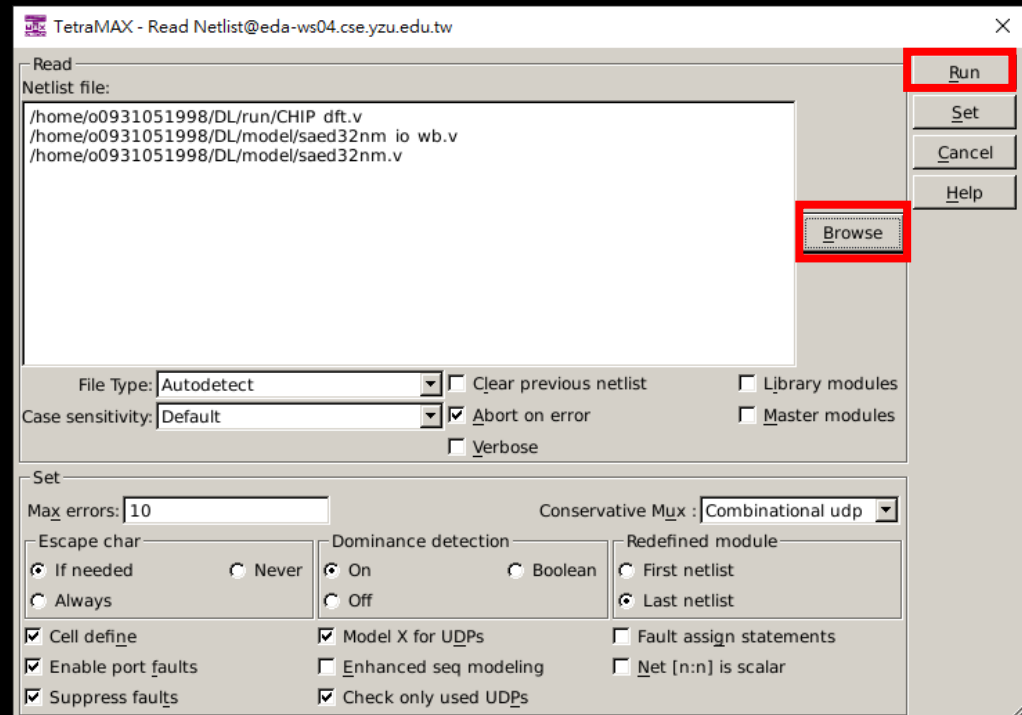
# TetraMAX Interface

- `$> tmax`



# Build Mode – Read Netlists

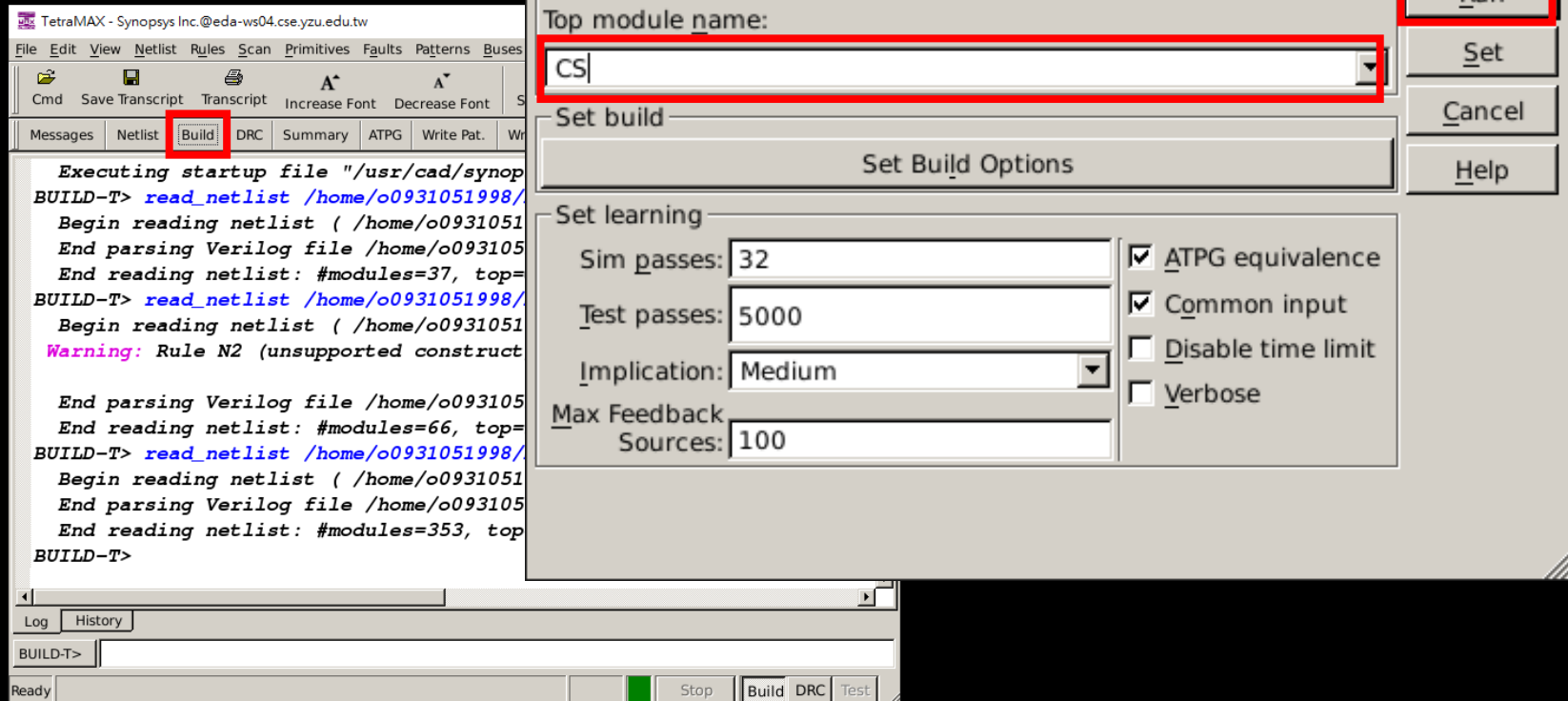
- Netlist → Read Netlist
- Click “Browse” →  
Select CS\_dft.v & saed32nm\_io\_wb.v &  
saed32nm.v
- Press “Run”





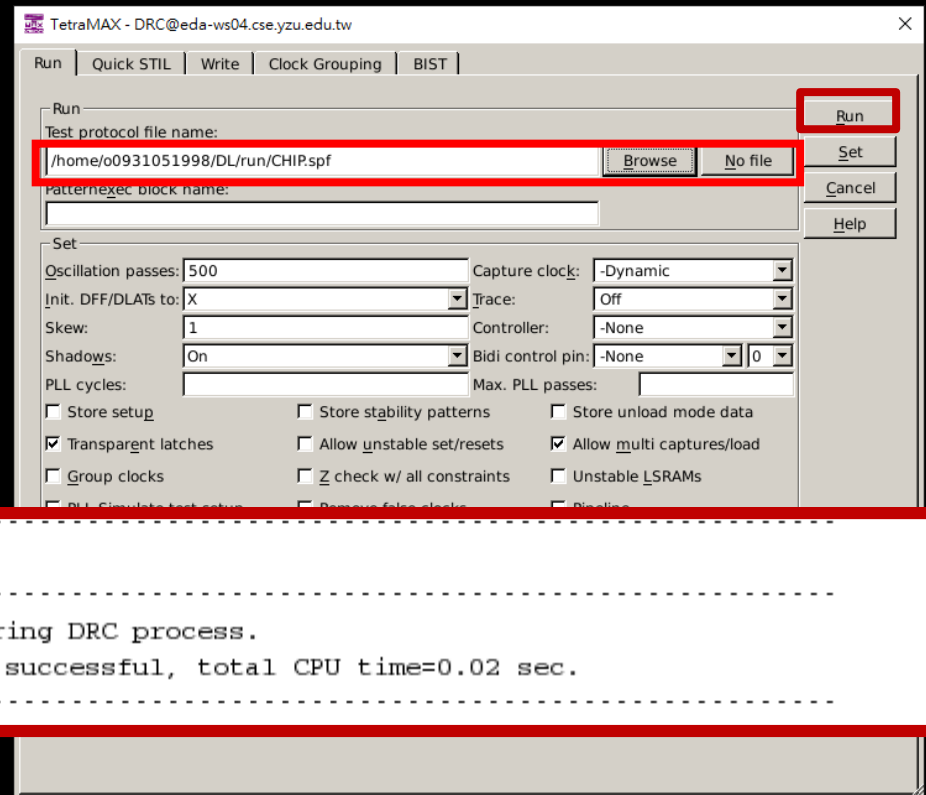
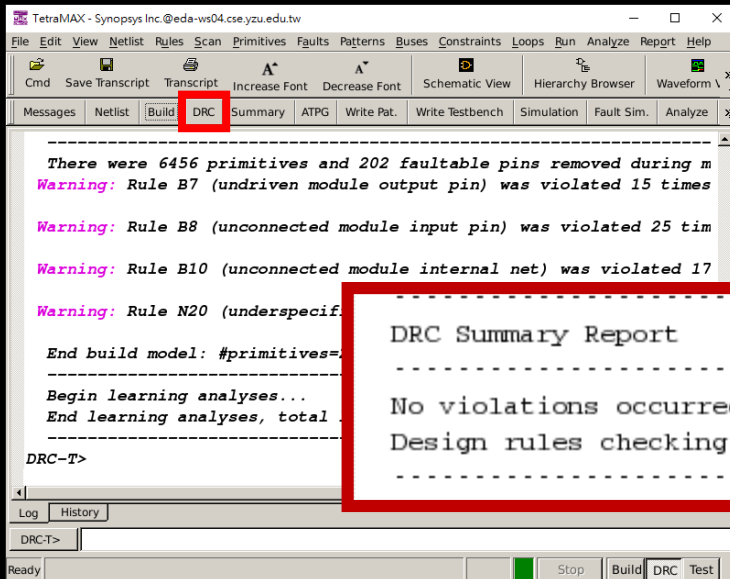
# Build Mode – Choose Top Module

- Click “Build” on main window
- Top module name: Choose “CS”
- Press “Run”



# DRC Mode – Choose Test Protocol

- Click “DRC” on main window
- Test protocol file name
  - Click “Browse”
  - Choose “CS.spf”
- Press “Run”



## DRC Summary Report

-----  
No violations occurred during DRC process.  
Design rules checking was successful, total CPU time=0.02 sec.  
-----

# ATPG (1/3)

- **Populate the design with faults**
  - TEST>
    - add\_faults -all
- **Generate ATPG Patterns**
  - TEST>
    - set\_patterns -internal
    - run\_atpg -auto
- **Report total faults, test pattern count, test coverage**
  - TEST>
    - report\_summaries

# ATPG (2/3)

## Uncollapsed Stuck Fault Summary Report

<i>fault class</i>	<i>code</i>	<i>#faults</i>
<i>Detected</i>	<i>DT</i>	<i>14918</i>
<i>Possibly detected</i>	<i>PT</i>	<i>0</i>
<i>Undetectable</i>	<i>UD</i>	<i>28</i>
<i>ATPG untestable</i>	<i>AU</i>	<i>0</i>
<i>Not detected</i>	<i>ND</i>	<i>96</i>
<i>total faults</i>		<i>15042</i>
<i>test coverage</i>		<i>99.36%</i>

## Pattern Summary Report

<i>#internal patterns</i>	<i>342</i>
<i>#basic_scan patterns</i>	<i>342</i>

# ATPG (3/3)

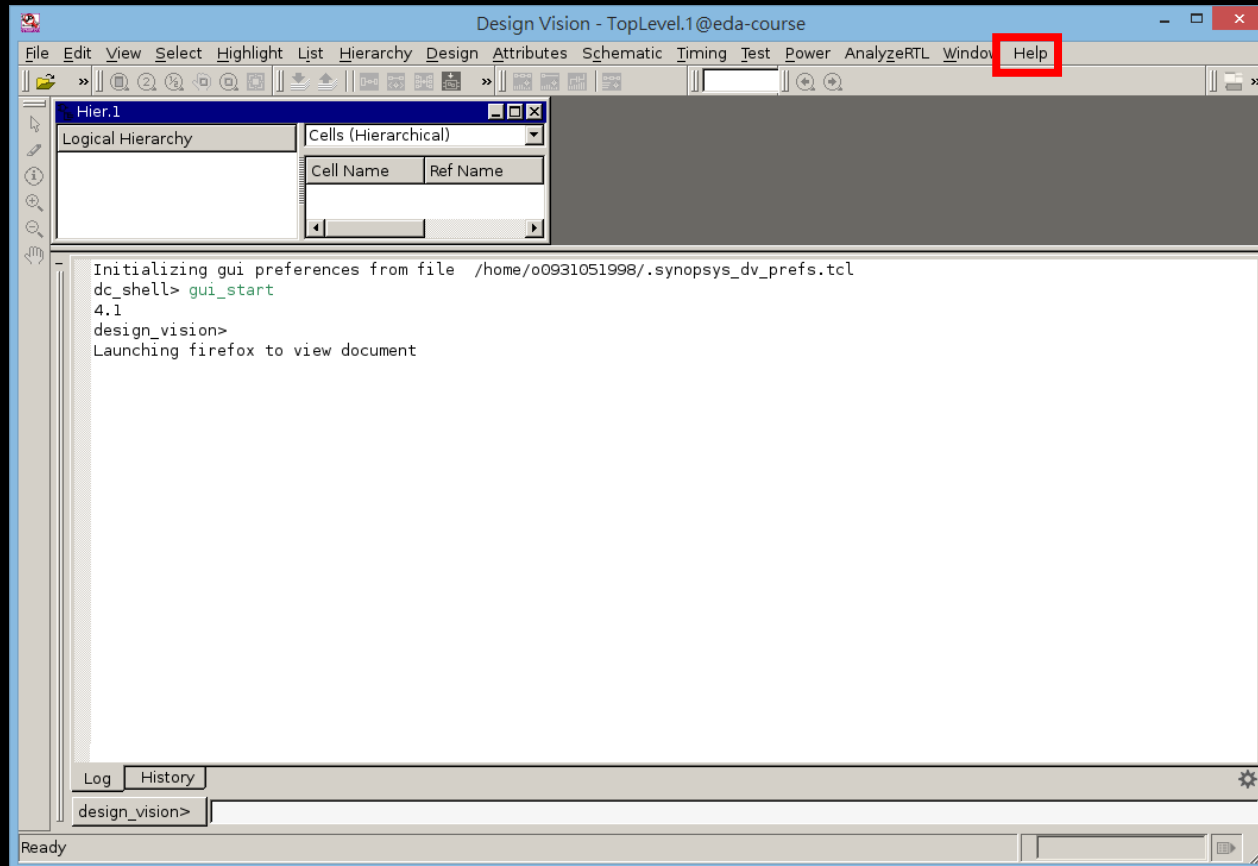
- **Saving ATPG Patterns**

- TEST>

- write\_patterns CS\_atpg.v -replace -internal -format stil -serial -vcs

# Help for Design Compiler

- Man page, report hotkey bindings, online help



# Help for TetraMAX

- Command summary, getting started, ...

