

# **Logic Synthesis (Synopsys)**

王旭昇

[hotline@cic.narl.org.tw](mailto:hotline@cic.narl.org.tw)

(03)577-3693 ext 885

# Outline

---

- ◆ Introduction
- ◆ Prepare your Verilog Code
- ◆ Process Setting
- ◆ Design Compiler GUI - Design Vision
- ◆ Setting Constraints
- ◆ Compiler
- ◆ Report and Analysis
- ◆ Save & Run Simulation



---

# 1. Introduction



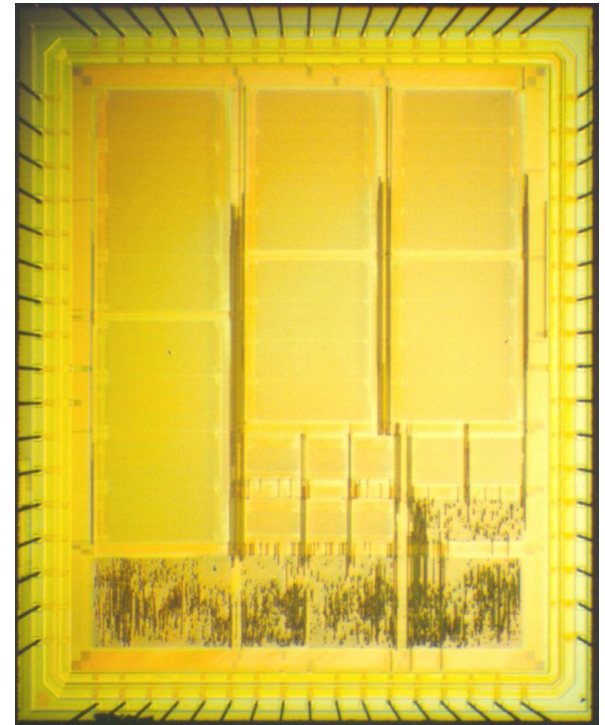
# IC Design and Implementation



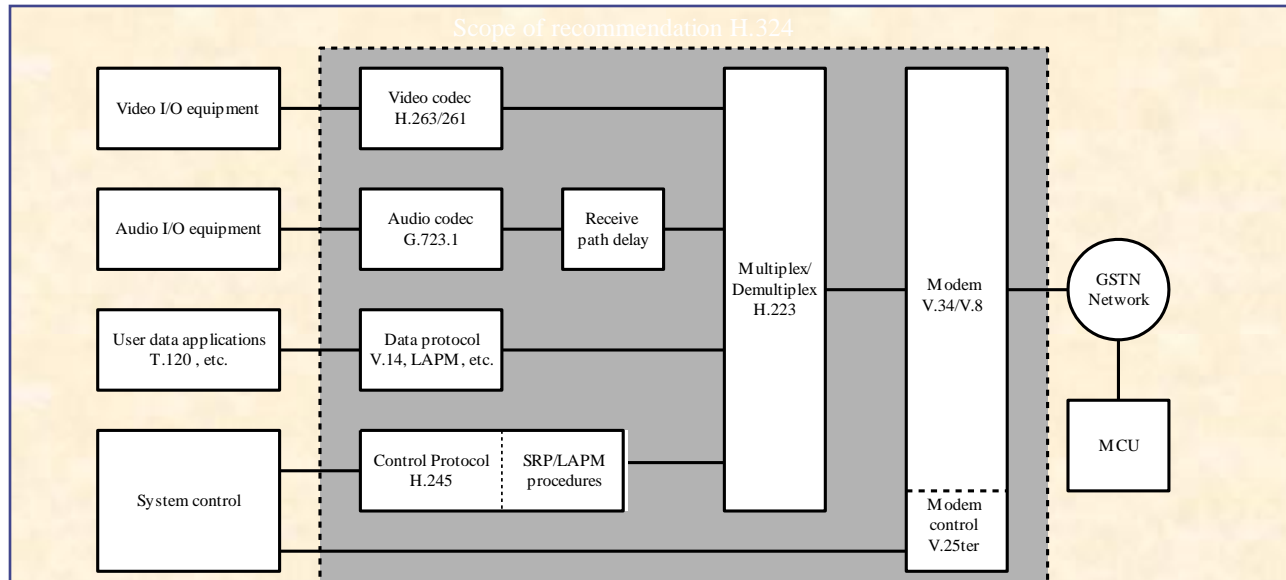
Idea



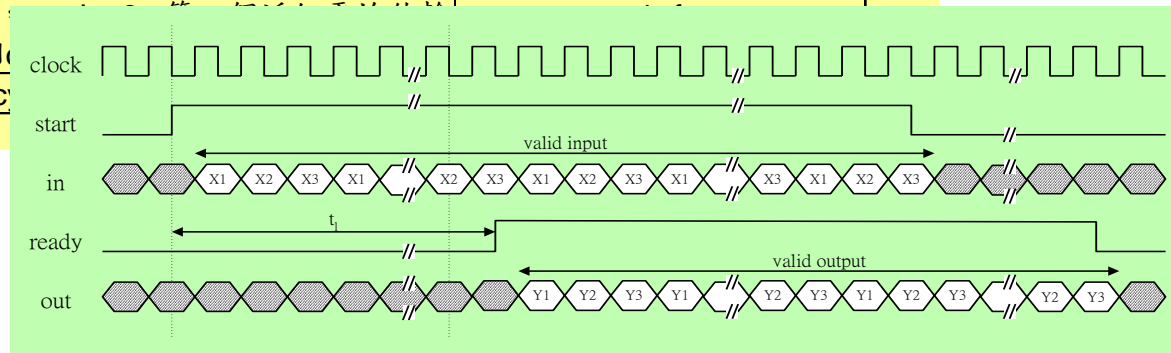
Chip



# System Spec.



腳位名稱		描述	Drive Strength/Output Load
clk	輸入	系統時脈	assume infinite
reset	輸入	系統重置訊號，high active	1 ns/pf
din	輸入	每個clock cycle輸入一個16-bit 正整數	1 ns/pf
ready	輸出	reset為1時，輸出前的半個clock cycle	
dout	輸出	每個clock cycle輸出一個16-bit 正整數	

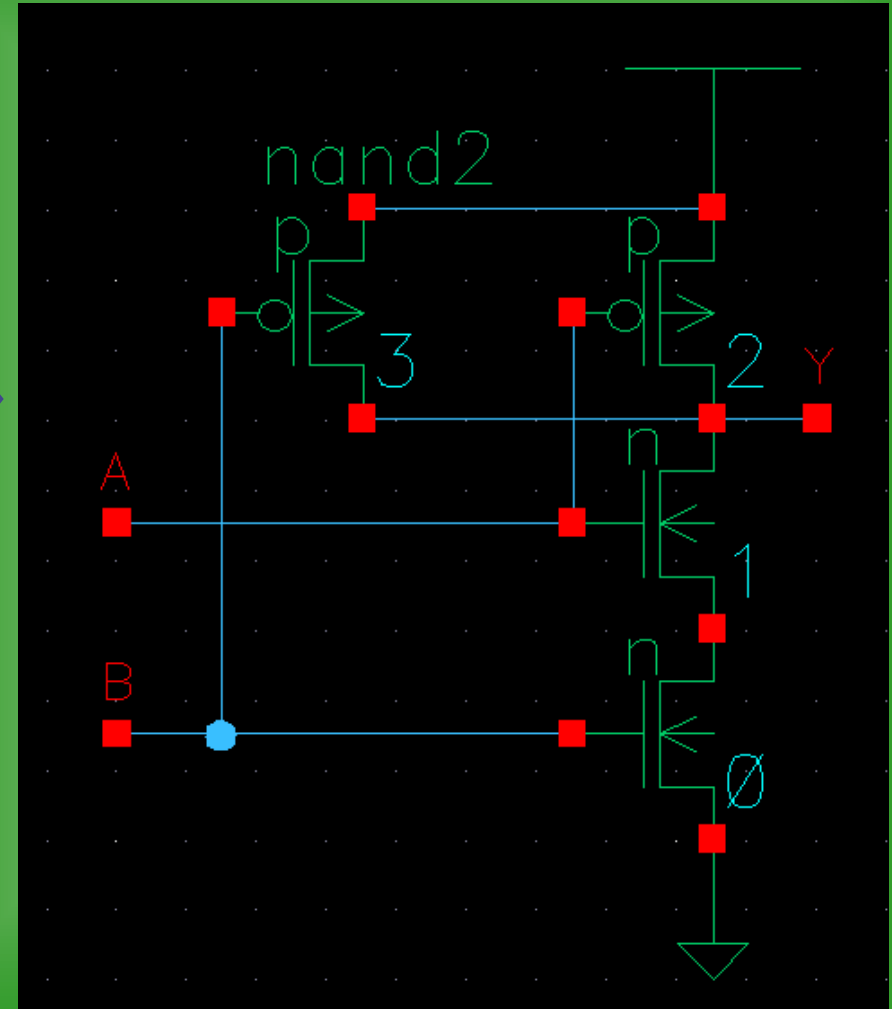
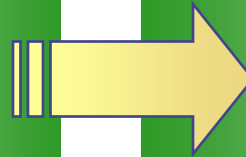
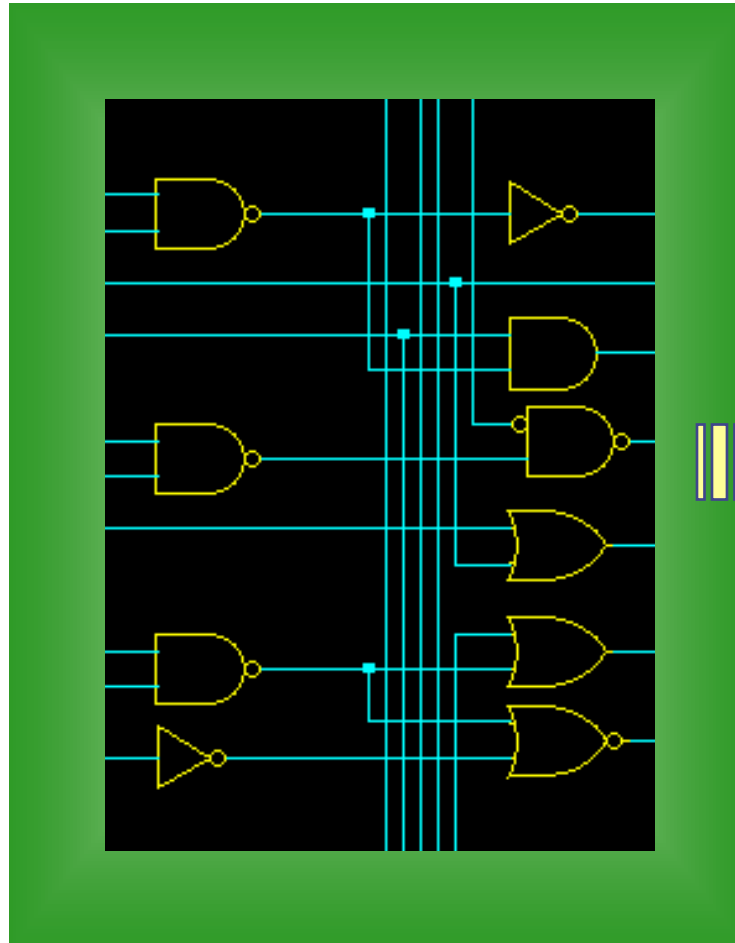


# RTL Level

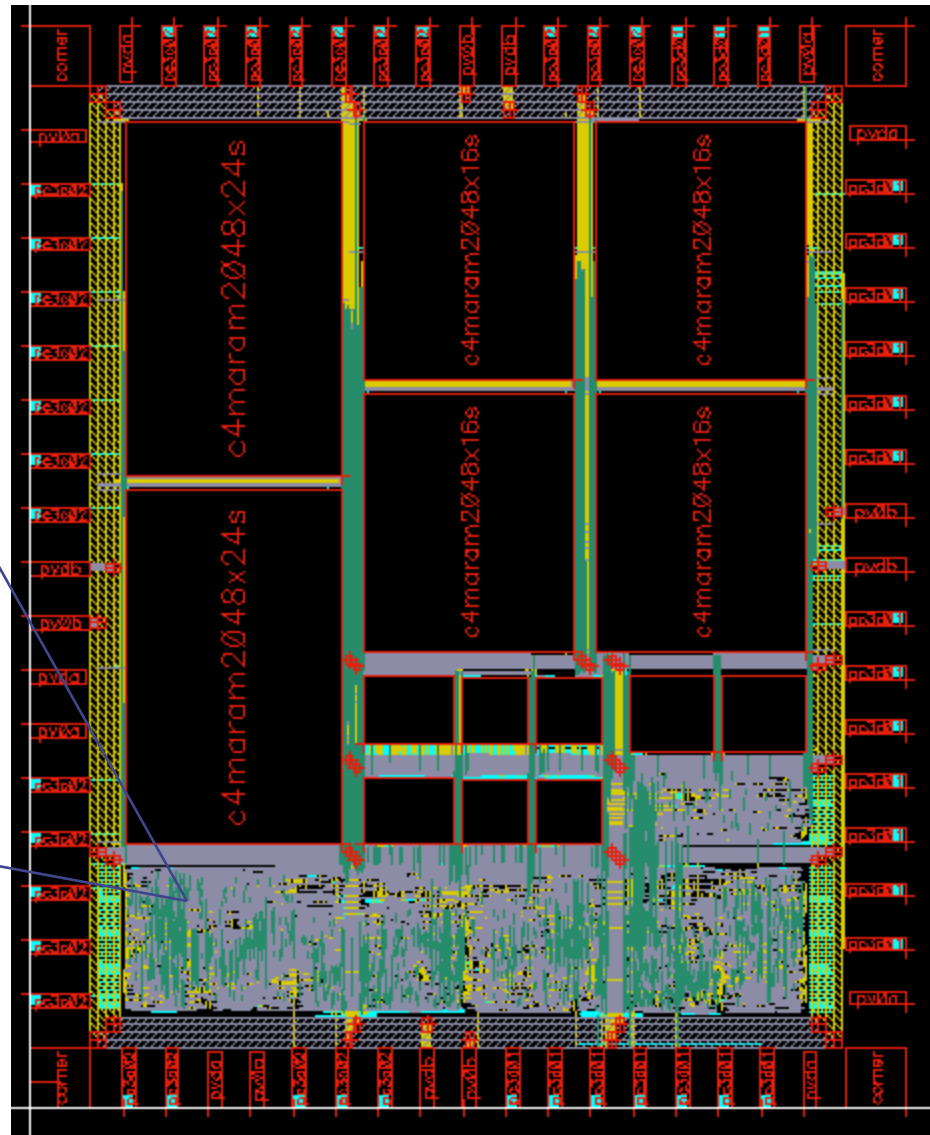
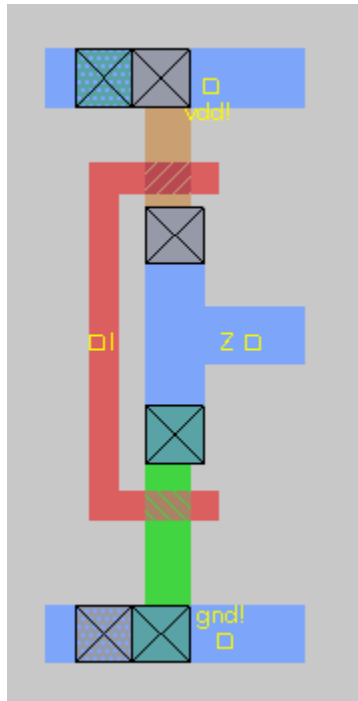


The diagram illustrates a 4x4 crossbar switch architecture. It consists of an input stage, a crossbar switch, and an output stage. The input stage has four 32-bit inputs: two solid white and two hatched. The output stage has four 32-bit outputs: two solid white and two hatched. The crossbar switch connects these inputs to the outputs. The outputs of the switch are then fed into two pairs of 32-bit ALUs, labeled AU0 and AU1. Each ALU pair has two 32-bit inputs and produces two 32-bit outputs. The AU0 outputs are connected to the AU1 inputs.

# Gate and Circuit Level Design

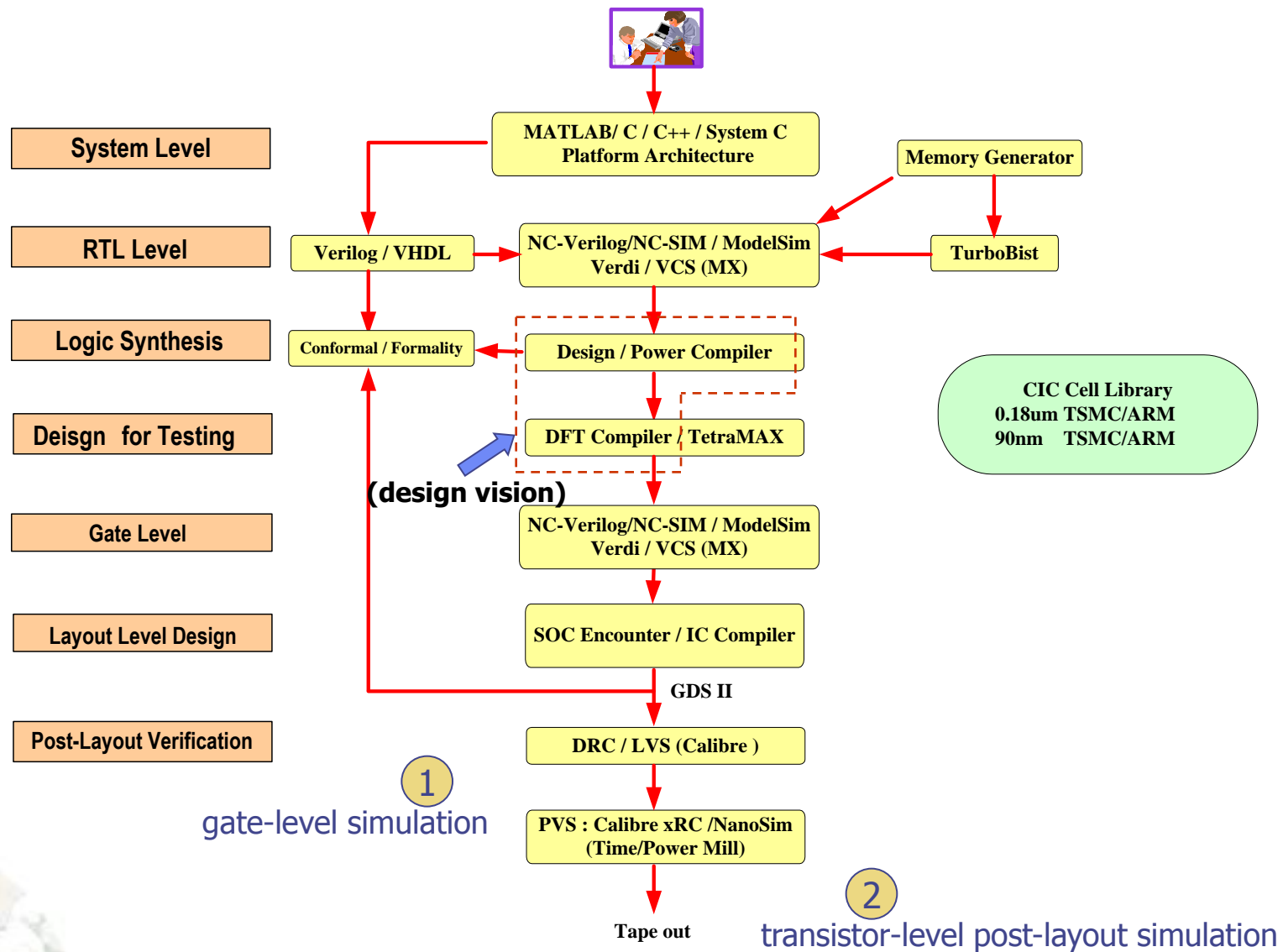


# Physical Design





# Cell-Based Design Flow



# What is Synthesis

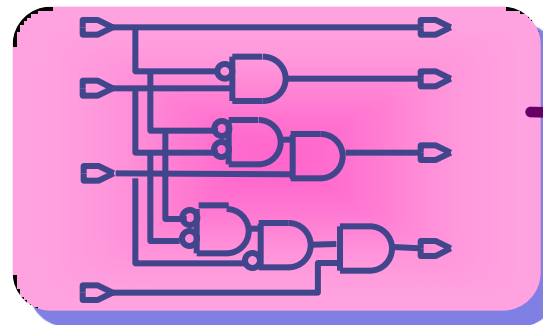
◆ Synthesis = translation + optimization + Mapping

```
residue = 16'h0000;  
  
if (high_bits == 2'b10)  
    residue = state_table[i];  
else state_table[i] = 16'h0000;
```

**HDL Source  
(RTL)**

no timing info. →

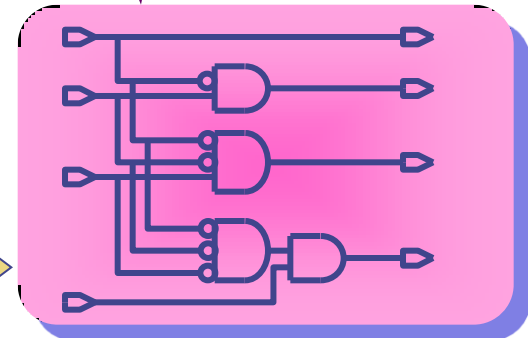
**Translate (HDL Compiler)**



**Generic Boolean  
(GTECH)**

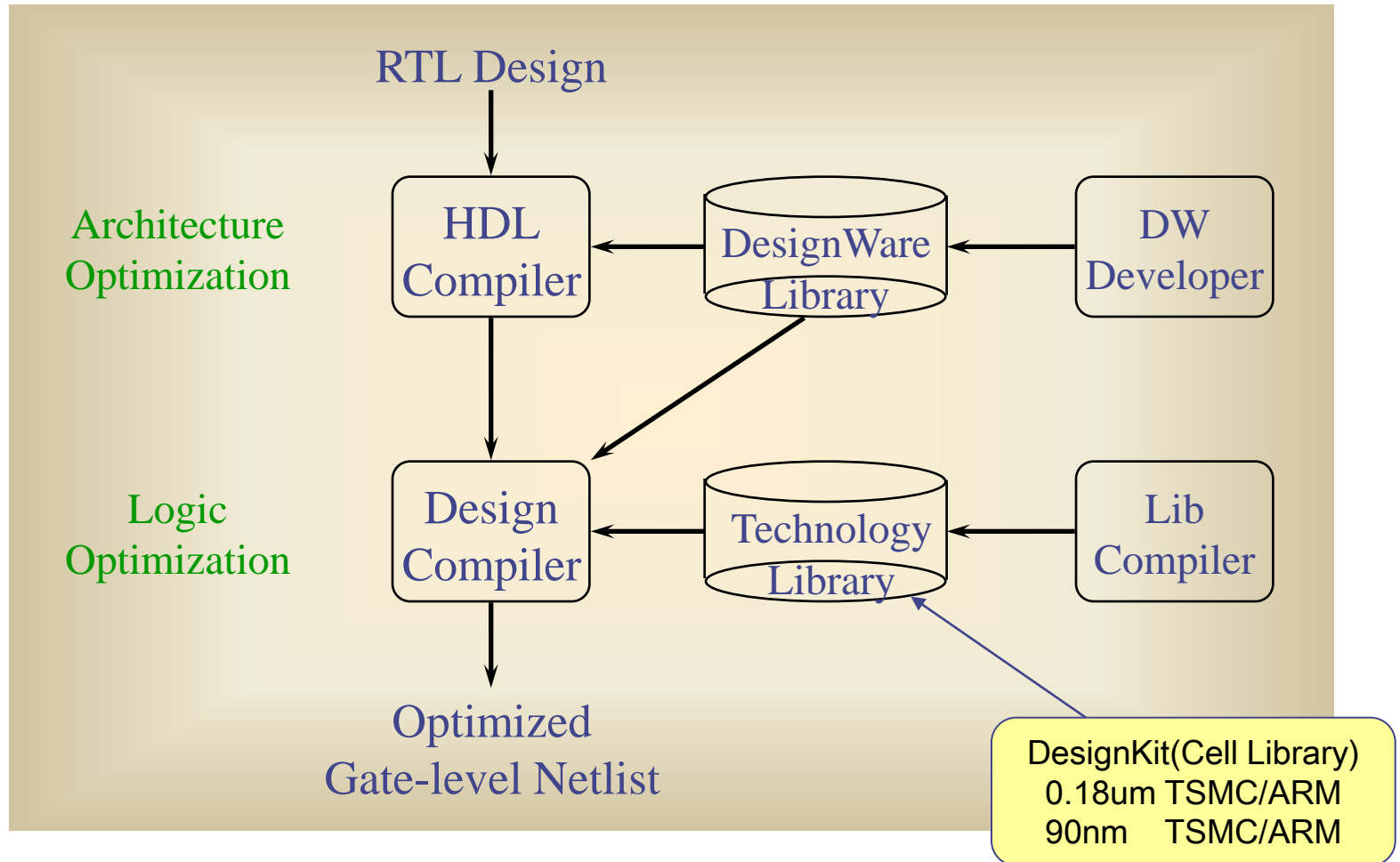
**Optimize + Map  
(Design Compiler)**

timing info. →



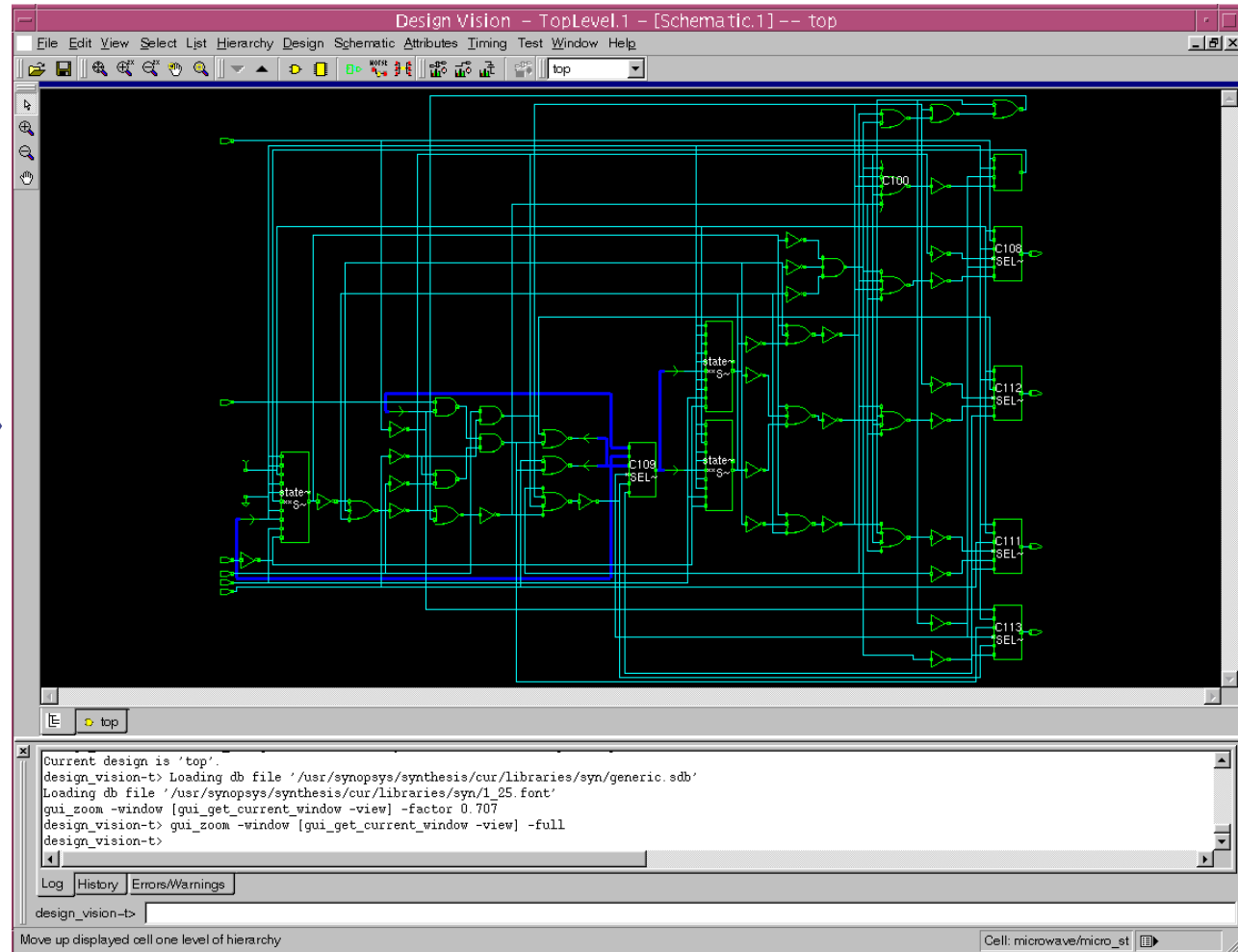
**Target Technology**

# Logic Synthesis Overview



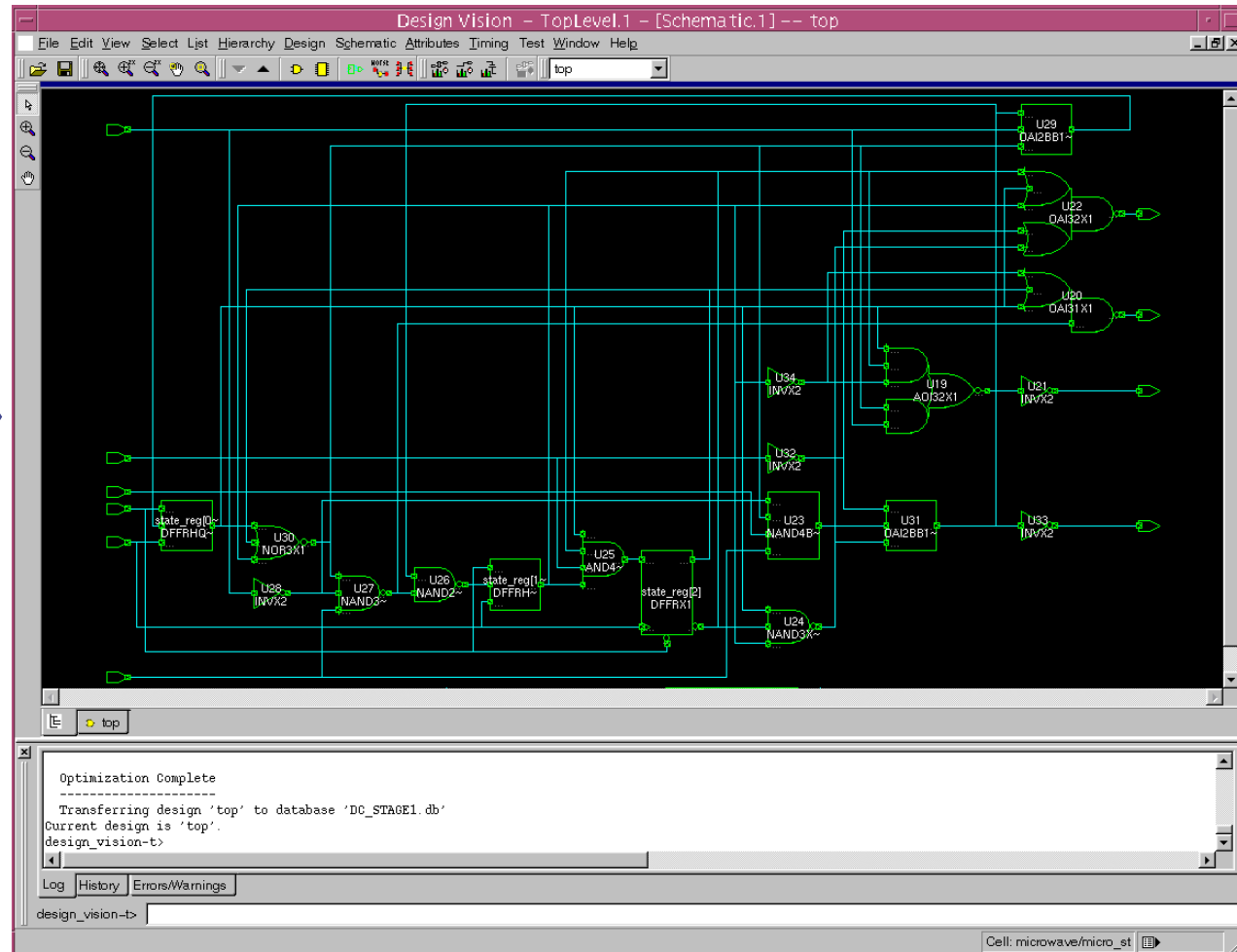
# HDL Compiler

- ◆ In schematic view, we can see the Verilog file is translated with a GTECH library (the Synopsys default)



# Design Compiler

- ◆ Design Compiler maps Synopsys design block to gate level design with a user specified library



# Synopsys On-Line Document (SOLD)

**SYNOPSYS** SOLVNET HOME | SYNOPSYS.COM | FEEDBACK | SITE MAP | HELP | SIGN OUT

**SolvNet®** Documentation Support Downloads Training Methodology My Profile Search SolvNet

HOME > DOCUMENTATION

## Documentation - Version E-2010.09

<p><b>A</b></p> <p>Astro® Astro® Interactive Ultra</p> <p><b>B</b></p> <p>BSD Compiler</p> <p><b>C</b></p> <p>Cadabra Certify® Confirma™ CHIPit™ CosmosScope™ CRITIC™ Custom Designer CustomExplorer / Custom WaveView</p> <p><b>D</b></p> <p>Design Compiler® Design Vision™ DesignWare® Library DesignWare® Memory Models DFT Compiler / DFTMAX™</p> <p><b>E</b></p> <p>ESP</p> <p><b>F</b></p> <p>Formality®</p>	<p><b>H</b></p> <p>HAPS HDL Compiler™ Hercules™ HSIM® &amp; HSIM<sup>plus</sup>™ HSPICE®</p> <p><b>I</b></p> <p>IC Compiler IC Validator Identify® Identify Pro</p> <p><b>J</b></p> <p>JupiterXT™</p> <p><b>L</b></p> <p>Leda® Library Compiler™ Liberty™ NCX Lynx Design System</p> <p><b>M</b></p> <p>Magellan™ Milkyway Environment™ MVTools</p> <p><b>N</b></p> <p>NanoSim® NanoTime</p>	<p><b>P</b></p> <p>Power Compiler™ PrimeRail PrimeTime® Suite</p> <p><b>S</b></p> <p>Saber SmartModels and FlexModels SpiceCheck StarRC™ Synplify® / Synplify Pro® / Synplify Premier Symphony C Compiler Symphony HLS System Studio</p> <p><b>T</b></p> <p>TCAD TetraMAX®</p> <p><b>V</b></p> <p>VCS® VCS® MX Vera® VMC</p> <p><b>X</b></p> <p>XA Option for NanoSim® and HSIM<sup>plus</sup>™</p>	<p><b>Local Disk Documentation Alternative</b> Available Now for the E-2010.09 release</p> <p><b>PrimeTime Recommended Reading List</b> Expert Advice on PrimeTime</p> <p><b>Download the Latest Release</b> 2010.09 Now Available</p> <p><input type="checkbox"/> DOCUMENTATION FOR OTHER PRODUCTS</p> <p><input type="checkbox"/> MAN PAGES AND ERROR MESSAGES</p> <p><input type="checkbox"/> JAPANESE LANGUAGE DOCUMENTATION</p> <p><input type="checkbox"/> DOCUMENT ARCHIVES</p> <p><input type="checkbox"/> LICENSING</p> <p><input type="checkbox"/> INSTALLATION</p> <p><input type="checkbox"/> COPYRIGHT INFORMATION</p> <p><input type="checkbox"/> SNUG PROCEEDINGS LIBRARY</p>
---	--	---	--

資料來源:  
<http://solvnet.synopsys.com>

# How to get the latest User Guide?

財團法人國家實驗研究院  
國家晶片系統設計中心  
NATIONAL CHIP IMPLEMENTATION CENTER  
晶片中心 > 設計服務 > Design Compiler

關於 CIC 設計服務 製程服務 量測服務 教育訓練 技術推廣 網路資源 即時訊息 會員專區

申請狀態 軟體簡介 軟體申請 軟體取得 IP修改 實作範例 聯絡窗口

檔名	檔案大小	版本	開放下載
synthesis_2008.09-sp2_linux.tgz	261832951	2008.09-sp2	NO
說明：Synopsys Synthesis for Red Hat Enterprise Linux 4 or compatible			
synthesis_2008.09-sp2_sun.tgz	196779050	2008.09-sp2	NO
說明：Synopsys Synthesis for SUN OS 5.9 or later			
synthesis_2010.03-sp5_linux.tgz	353187103	2010.03-sp5	NO
說明：Synopsys Synthesis for Red Hat Enterprise Linux 4 or compatible			
synthesis_2010.03-SP5-2_sun.tgz	436586640	2010.03-sp5	NO
說明：Synopsys Synthesis for SUN OS 5.9 or later			
DFT-Compiler.zip	29574310	共用檔案	NO
說明：DFT Compiler User Guide			
Design-Compiler.zip	28563484	共用檔案	NO
說明：Design Compiler User Guide			

Synopsys SOLD

---

## 2. Prepare your Verilog Code





# Verilog Module

## Module

Module Name &  
Port List

Definitions  
Port, Wire, Register  
Parameter, Integer, Function

Module Instantiations

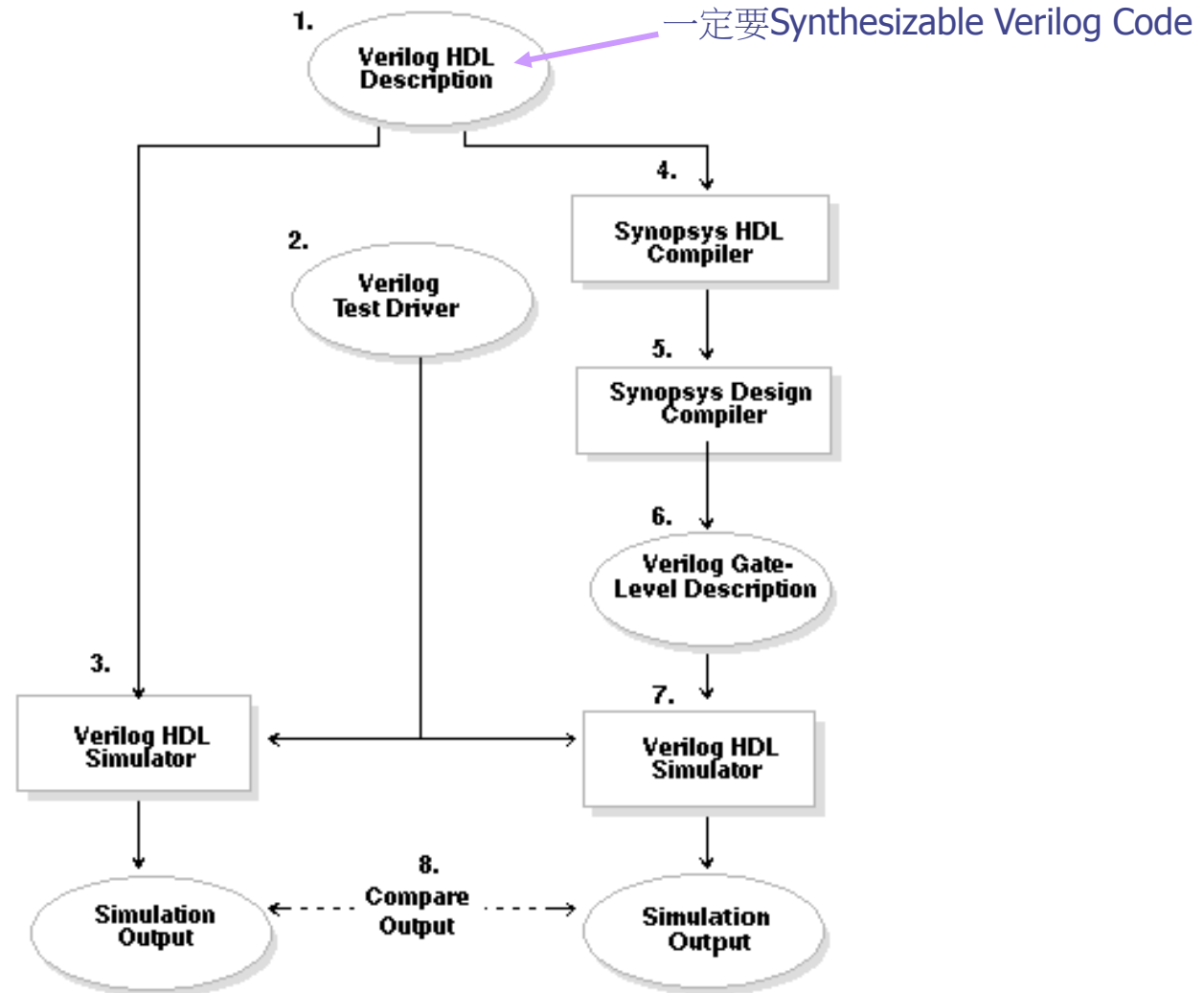
Module Statements &  
Constructs

```
module test(a,b,c,d,z,sum);
    input a,b;           --Inputs to nand gate
    input[3:0] c,d;      --Bused Input
    output z;            --Output from nand gate
    output[3:0] sum;     --Output from adder
    wire and_out;        --Output from and gate
    reg [3:0] sum;       --Bused Output

    AND instance1(a,b,and_out);
    INV instance2(and_out, z);

    always @(c or d)
    begin
        sum = c + d;
    end
endmodule
```

# Design Methodology



# Synthesizable Verilog Code

---

- ◆ Synopsys can't accept all kinds of Verilog and VHDL constructs
- ◆ Synopsys can only accept a subset of Verilog syntax and this subset is called "Synthesizable Verilog Code"
- ◆ This chapter will introduce synthesizable verilog coding style to you, and this is the first challenge when you use Synopsys to convert your RTL code to Gate level netlist



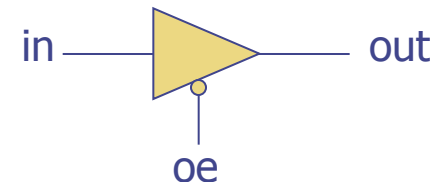
# HDL Compiler Unsupported

- ◆ delay
- ◆ initial
- ◆ repeat
- ◆ wait
- ◆ fork ... join
- ◆ event
- ◆ deassign
- ◆ force
- ◆ release
- ◆ **primitive** -- User defined primitive
- ◆ time
- ◆ triand, trior, tri1, tri0, trireg
- ◆ nmos, pmos, cmos, rnmos, rpmos, rcmos
- ◆ pullup, pulldown
- ◆ rtran, tranif0, tranif1, rtranif0, rtranif1
- ◆ case identity and not identity operators (**===** and **!==**)
- ◆ **forever**
- ◆ **Division and modulus operators**
  - division can be done using designware instantiation

技巧

```
ex:
`define del 1 //define global variable
always @(posedge clk)begin
    #`del;
    if(!rst) q=0;
    else    q=d;
end
```

example: wire out=(!oe)? in : 'hz;



# Verilog Operators Supported

---

- ◆ Binary bit-wise ( $\sim$ ,  $\&$ ,  $|$ ,  $\wedge$ ,  $\sim\wedge$ )
- ◆ Unary reduction ( $\&$ ,  $\sim\&$ ,  $|$ ,  $\sim|$ ,  $\wedge$ ,  $\sim\wedge$ )
- ◆ Logical ( $!$ ,  $\&\&$ ,  $||$ )
- ◆ 2's complement arithmetic ( $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ )
- ◆ Relational ( $>$ ,  $<$ ,  $>=$ ,  $<=$ )
- ◆ Equality ( $==$ ,  $!=$ )
- ◆ Logical shift ( $>>$ ,  $<<$ )
- ◆ Conditional ( $?:$ )

# Bit-wise,Unary,Logical Operator

a = 1011  
b = 0010

bit-wise	unary reduction	logical
a   b = 1011	a = 1	a    b = 1
a & b = 0010	& a = 0	a && b = 1

# Synthesizable Code for Verilog-1995

---

## ◆ Verilog Basis

- parameter declarations
- ① ■ wire, wand, wor declarations
- reg declarations
- input, ouput, inout declarations
- ② ■ continuous assignments
- module instantiations
- gate instantiations
- ③ ■ always blocks
- task statements
- function definitions
- ④ ■ for, while loop
- ⑤ ■ disable
- if ... else if ... else
- ⑥ ■ case, caseX, caseZ

## ◆ Synthesizable Verilog primitives cells

- and, or, not, nand, nor, xor, xnor
- bufif0, bufif1, notif0, notif1



# Verilog 2001/2005

## ◆ Supported Verilog 2001/2005 Constructs

- 1 ■ ANSI-C-style port declarations
- 2 ■ Power operator (**\*\***)
- 3 ■ Arithmetic shift operators (**<<<** and **>>>**)
- 4 ■ Multidimensional arrays
- 5 ■ Part select addressing (**[+:]** and **[-:]** operators)
- 6 ■ **generate** statement
- 7 ■ signed quantities

### 補充1

如何告知**DC**要以哪一版本的**Verilog**語法為主:

```
design_vision> set  hdlin_vrlg_std 2001      (1995 / 2001 / 2005)
```

### 補充2

如何告知**DC**要以哪一版本的**System Verilog**語法為主:

```
design_vision> set  hdlin_sverilog_std 2005  (2005 / 2009)
```



# CIC 寒暑假課程報名網頁



財團法人國家實驗研究院  
**國家晶片系統設計中心**  
NATIONAL CHIP IMPLEMENTATION CENTER

[晶片中心](#) > [教育訓練](#) > [報名服務](#)



[關於 CIC](#)
[設計服務](#)
[製程服務](#)
[量測服務](#)
[教育訓練](#)
[技術推廣](#)
[網路資源](#)
[即時訊息](#)
[會員專區](#)

[線上報名](#)
[開課課程](#)
[增購講義](#)
[繳費單列印](#)
[點選與免費課程取](#)

[最新消息](#)
[學術界課程](#)
[半導體學院課程](#)
[研討會](#)
[e-Learning](#)

[報名資料查詢](#)
[請假登錄](#)
[問卷調查](#)
[常見問題](#)
[連絡窗口](#)

[您的課程清單](#)

梯次課程代號	課程名稱	講師		上課地點	費用(NTD)		報名截止時間	狀態
					學界	一般人士		
100C104-E	<a href="#">Logic Synthesis with Design Compiler</a>	<a href="#">王旭昇</a>	2011-07-13~2011-07-15	<a href="#">新竹CIC</a>	2,100	4,200	2011-07-01 23:59	已額滿 建議增開
100C104-F	<a href="#">Logic Synthesis with Design Compiler</a>	<a href="#">王旭昇</a>	2011-07-27~2011-07-29	<a href="#">成大自強校區電機館</a>	2,100	4,200	2011-07-08 23:59	已額滿 建議增開
100C104-G	<a href="#">Logic Synthesis with Design Compiler</a>	<a href="#">王旭昇</a>	2011-08-08~2011-08-10	<a href="#">新竹CIC</a>	2,100	4,200	2011-07-28 23:59	已額滿 建議增開
100C104-H	<a href="#">Logic Synthesis with Design Compiler</a>	<a href="#">王旭昇</a>	2011-08-17~2011-08-19	<a href="#">興大電機館孟堯晶片中心</a>	2,100	4,200	2011-07-31 23:59	已額滿 建議增開

---

## 3. Process Setting for Synthesis Tool



# 0.18um TSMC/ARM (ver 3.2)

- ◆ In CIC Cell-Based flow, we support **ARM 0.18um** cell library, the .synopsys\_dc.setup file in DB/XG mode is as follows

## .synopsys\_dc.setup

TCL 語法

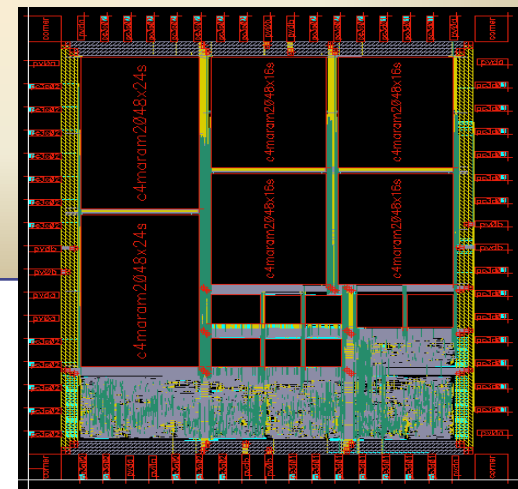
```
set company "CIC"
set designer "Student"

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
```

補充:

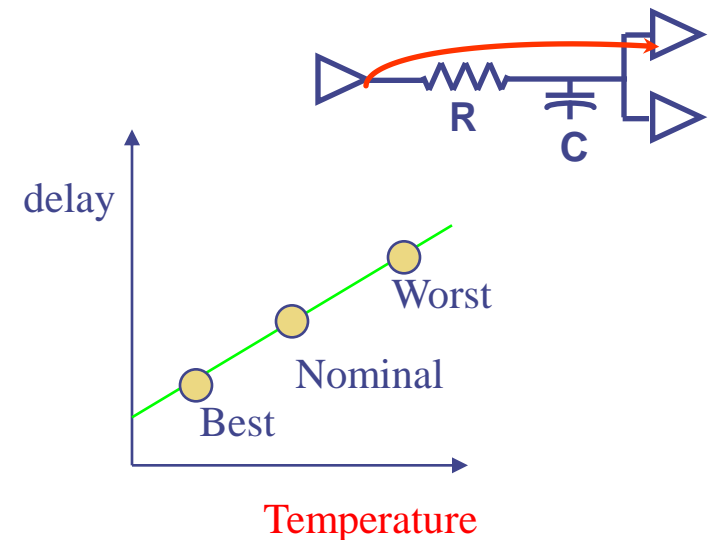
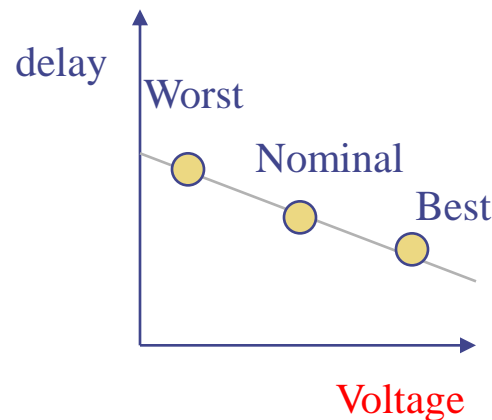
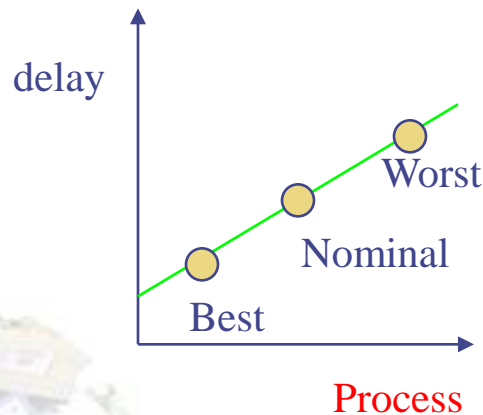
ex:  
進入一個synopsys目錄  
unix% cd synTab  
=> cd synopsys (自動秀出)



# Slow v.s. Fast Library

- Operating condition model scales components delay, directs the optimizer to simulate variations in process, temperature, and voltage

<i>Name</i>	<i>Process</i>	<i>Temp</i>	<i>Volt</i>	<i>Interconnection Model</i>
slow	1	125	1.62	balance_case_tree
<b>typical</b>	<b>1</b>	<b>25</b>	<b>1.8</b>	<b>balance_case_tree</b>
fast	1	-40	1.98	balance_case_tree



# 90nm TSMC/ARM

- ◆ In CIC Cell-Based flow, we support ARM 90nm cell library, the .synopsys\_dc.setup file in DB/XG mode is as follows

## **.synopsys\_dc.setup**

TCL 語法

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

set search_path      " . $Your_path/CBDK_TSMC90G_Arm/CIC/SynopsysDC $search_path "
set target_library   "      slow.db fast.db tpzn90gv3wc.db tpzn90gv3lt.db "
set link_library     " * $target_library dw_foundation.sldb "
set symbol_library   " tsmc090.sdb generic.sdb "
set synthetic_library " dw_foundation.sldb "

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

---

## 4. Invoke Design Compiler GUI - Design Vision



# Invoke Design Vision

◆ *Unix%> dv* ( Only command mode: *dc\_shell* )

The screenshot displays the Design Vision software interface. The main window is titled "Design Vision - TopLevel.1 (CHIP) - [Hier.1]". It features a menu bar (File, Edit, View, Select, Highlight, List, Hierarchy, Design, Attributes, Schematic, Timing, Test, Power, Window, Help) and a toolbar. The "Logical Hierarchy" pane on the left shows a tree structure with "CHIP" selected. The "Cells (Hierarchical)" pane on the right shows a table with columns: Cell Name, Ref Name, Cell Path, and Dont Touch. The "Console" window is open in the foreground, showing the command prompt [andy@dbsd04 run]\$ dv and the output of the Design Vision initialization process.

Cell Name	Ref Name	Cell Path	Dont Touch
core	CS	core	undefined

```

[andy@dbsd04 run]$ 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 D-2010.03 for amd64 -- Feb 22, 2010
Copyright (c) 1988-2009 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.

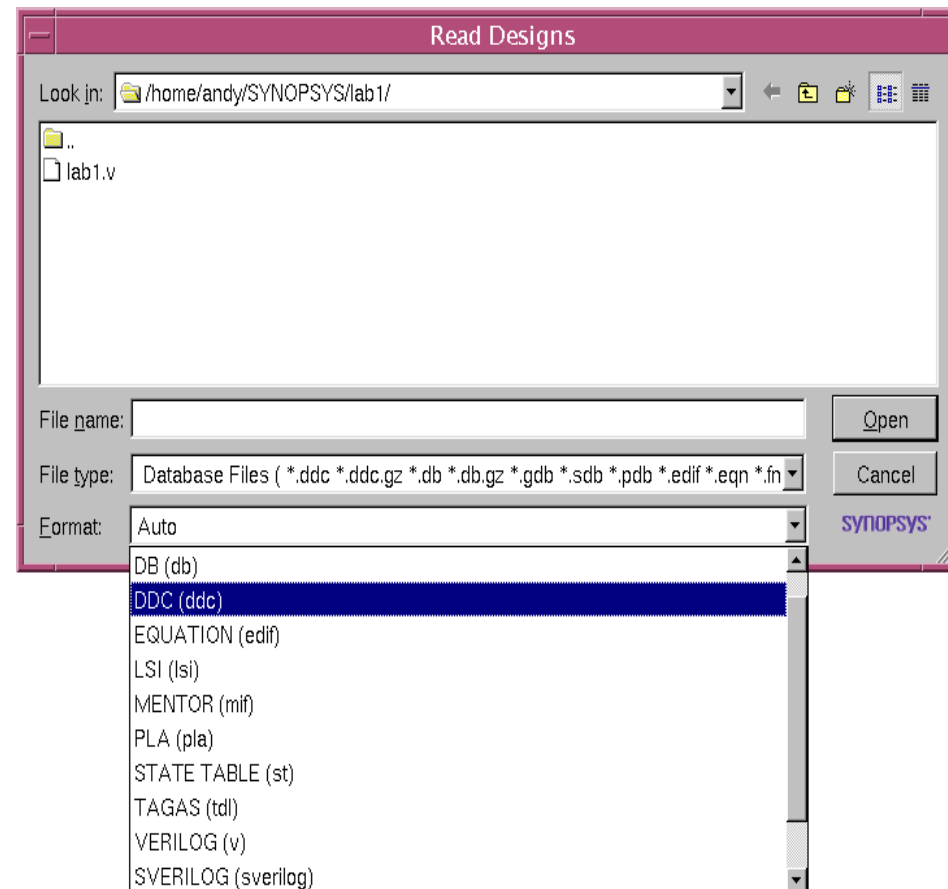
Initializing...
Loading db file '/user/DSD/andy/SYNOPSYS/lab3/lab3-1/syn_dc/model/slow.db'
Loading db file '/user/DSD/andy/SYNOPSYS/lab3/lab3-1/syn_dc/model/fast.db'
Loading db file '/user/DSD/andy/SYNOPSYS/lab3/lab3-1/syn_dc/model/tpz973gvwc.db'
Loading db file '/user/DSD/andy/SYNOPSYS/lab3/lab3-1/syn_dc/model/tpz973gvbc.db'
Initializing gui preferences from file /user/DSD/andy/.synopsys_dv_prefs.tcl
design_vision> design_vision>
  
```

A red arrow points to the command prompt in the Design Vision console window, indicating where to enter the command.

Enter command here

# Method 1: Read File

- ◆ Read netlists or other design descriptions into Design Compiler
- ◆ *File/Read*
- ◆ Support many different formats
  - Verilog: .v
  - VHDL: .vhd
  - System Verilog: .sv
  - EDIF
  - PLA(Berkeley Espresso): .pla
  - Synopsys internal formats:
    - ◆ DB(binary): .db
    - ◆ Enhance db file: .ddc
    - ◆ equation: .eqn
    - ◆ state table: .st





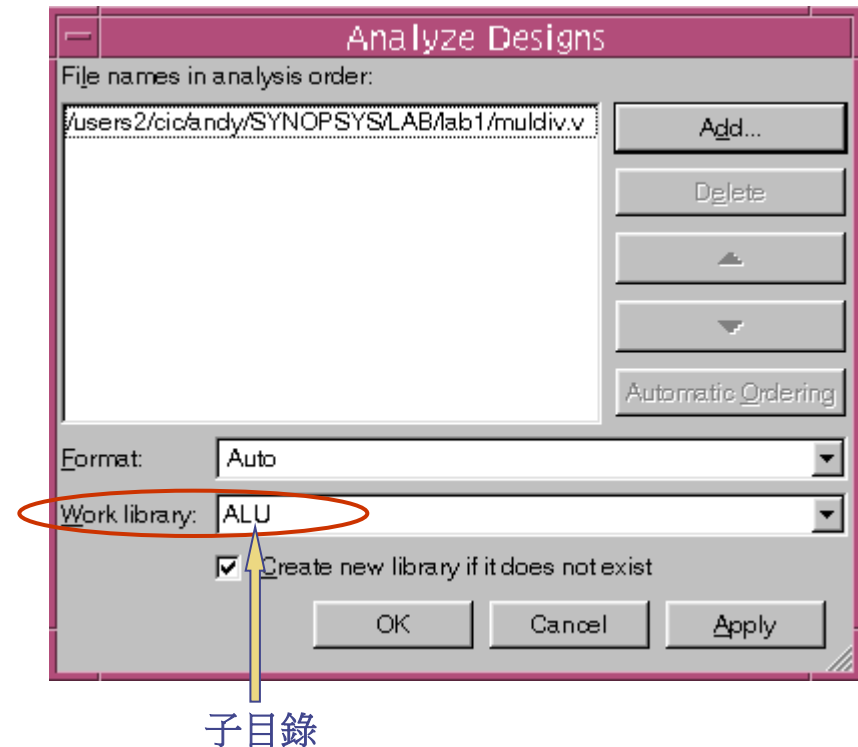
## Method 2: Analyze (1/2)

- ◆ Check VHDL & Verilog for **syntax** and **synthesizability**
- ◆ Create intermediate **.mr** and **.pvl** and **.syn** files and places them in library specified -- design library

- ◆ Equivalent to design\_vision command

1. design\_vision> **sh** mkdir ALU
2. design\_vision> **define\_design\_lib** ALU -path ./ALU
3. design\_vision> **analyze -library** ALU -format verilog muldiv.v

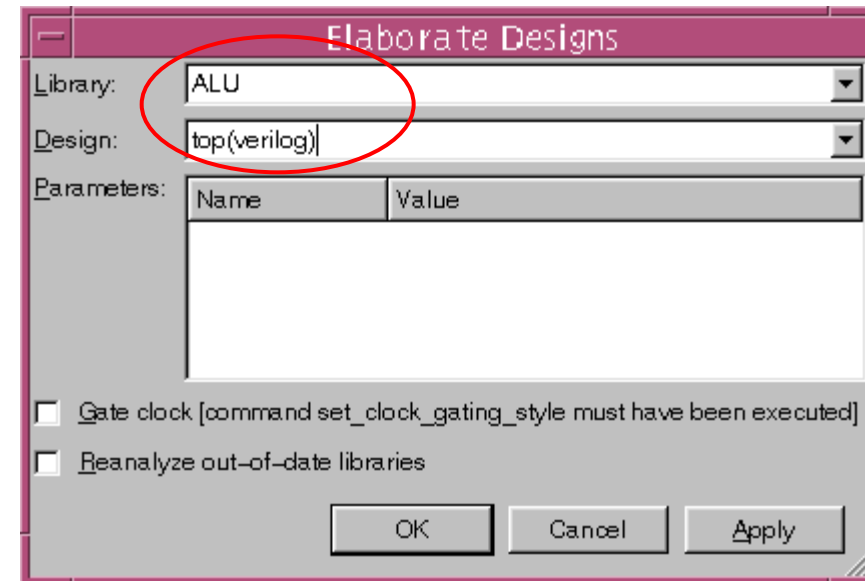
(File/Analyze)



## Method 2:Elaborate (2/2)

- ◆ Elaborate after analyze to bring design into Design Compiler memory using *generic* components (*GTECH*)
- ◆ Look in the design library for *intermediate file* for design specified
- ◆ Equivalent `dc_shell-xg-t` command

(File/Elaborate)



```
design_vision> elaborate top -architecture verilog -library ALU
```

3

新版DC提供**AutoRead**功能, 例如 :

```
design_vision> read_file -autoread -top top -recursive {./ ./dir1 ./dir2}
```

your top module name

# Example: Analyze/Elaborate (1/2)

```

module mul(P,A,B);
  parameter A_width=16;
  parameter B_width=16;
  input [A_width-1:0]    A;
  input [B_width-1:0]    B;
  output [A_width+B_width-1:0] P;
  assign                 P=A*B;
endmodule

module top(P1,P2,IN1,IN2,IN3,IN4);
  input [3:0] IN1,IN2;
  input [4:0] IN3;
  input [2:0] IN4;
  output [7:0] P1,P2;
  mul #(4,4) U1(P1,IN1,IN2);
  mul #(5,3) U2(P2,IN3,IN4);
endmodule

```

File / Read

```

Warning: Design 'top' has '2' unresolved references. For more detailed information, use the "link" command. UID-341
Warning: In design 'top', there are 16 multiple-driver nets with unknown wired-logic type. LINT-30
Warning: Cell 'U1' (mul_param_1) not translated. TRANS-1
Warning: Cell 'U2' (mul_param_2) not translated. TRANS-1
Warning: File /users/staff/DSD/andy/test/mul-verilog.pvl for module mul cannot be found. ELAB-320
Warning: Parameter mismatch in linking reference 'mul' by name.
Can't find design. LINK-18
Warning: File /users/staff/DSD/andy/test/mul-verilog.pvl for module mul cannot be found. ELAB-320
Error: Module 'mul' cannot be found for elaboration. ELAB-357
Error: 'mul' was not identified as a synthetic library module
and could not be successfully elaborated from design library 'WORK'. LINK-10
Warning: Unable to resolve reference 'mul' in 'top'. LINK-5
Warning: File /users/staff/DSD/andy/test/mul-verilog.pvl for module mul cannot be found. ELAB-320
Warning: Parameter mismatch in linking reference 'mul' by name.
Can't find design. LINK-18
Warning: File /users/staff/DSD/andy/test/mul-verilog.pvl for module mul cannot be found. ELAB-320
Error: Module 'mul' cannot be found for elaboration. ELAB-357
Error: 'mul' was not identified as a synthetic library module
and could not be successfully elaborated from design library 'WORK'. LINK-10
Warning: Unable to resolve reference 'mul' in 'top'. LINK-5

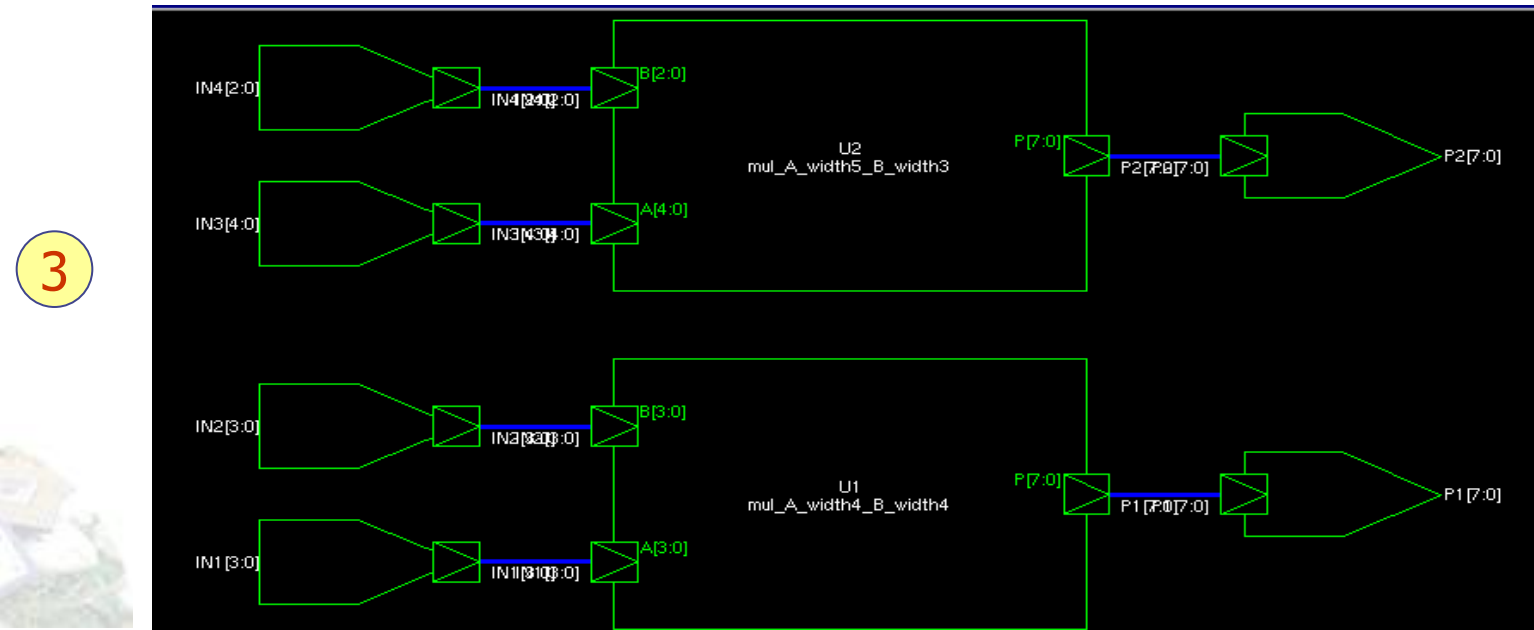
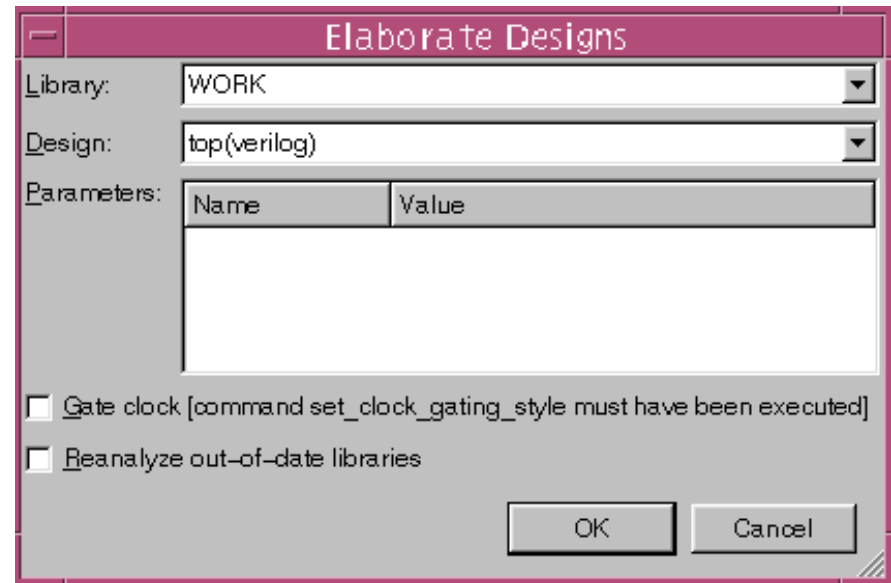
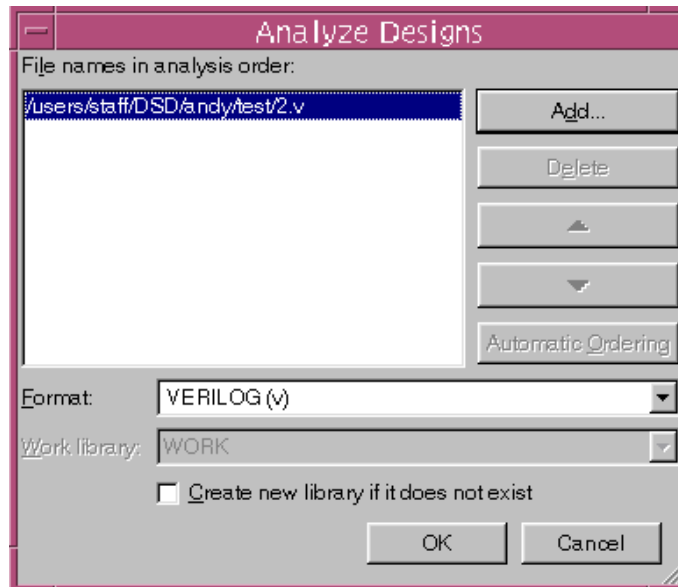
```

Clear

Log History Errors/Warnings

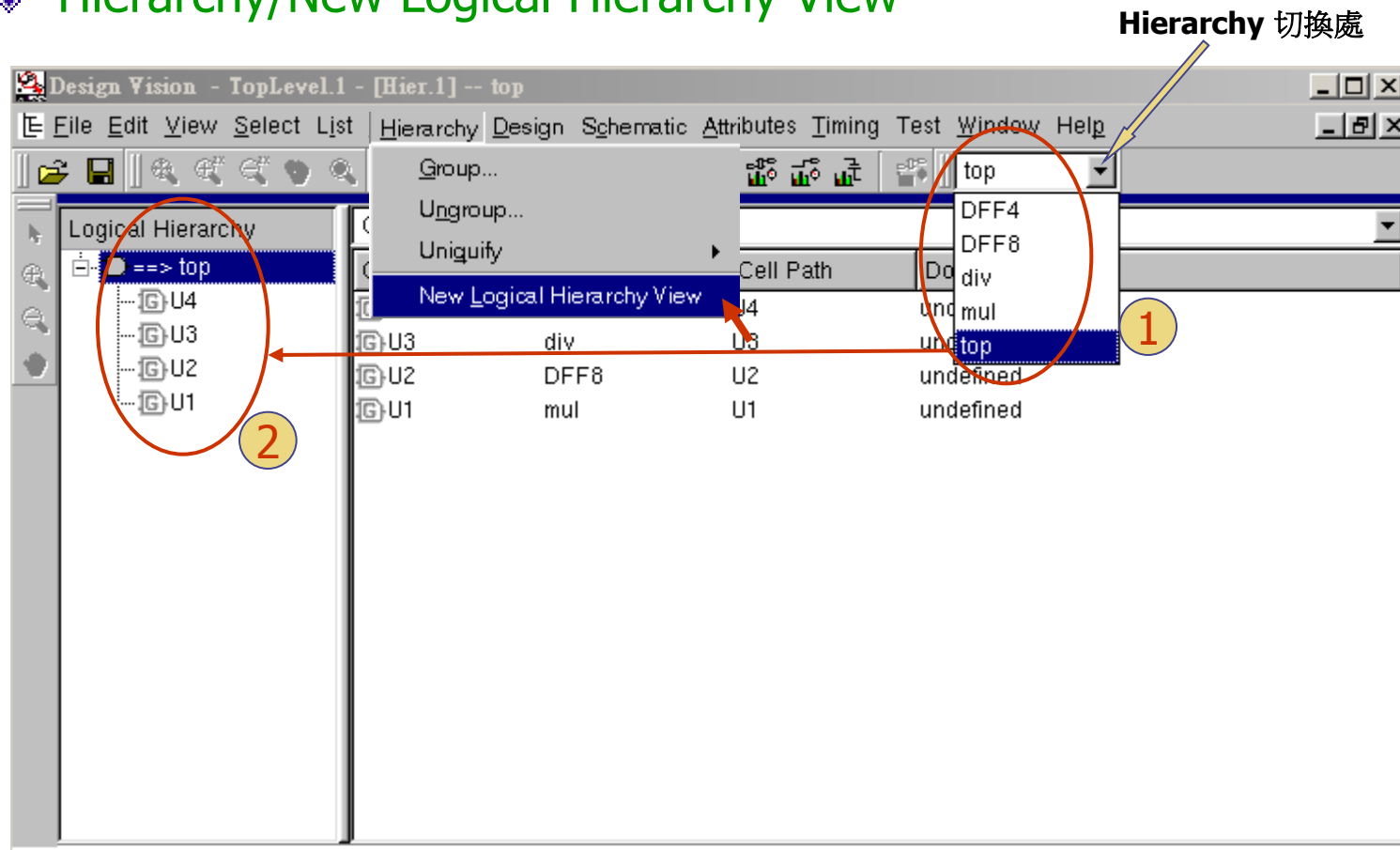
design\_vision-t>

# Example: Analyze/Elaborate (2/2)



# Hierarchy View

## ◆ Hierarchy/New Logical Hierarchy View



# Symbol View

**Design Vision - TopLevel.1 - [Hier.1] -- top**

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

Logical Hierarchy

- ==> top (1)
- U4
- U3
- U2
- U1

Cells (Hierarchical)

Cell Name	Ref Name	Cell Path	Dont Touch
U4	DFF4	U4	undefined
U3	div	U3	undefined
U2	DFF8	U2	undefined
U1	mul	U1	undefined

**Design Vision - TopLevel.1 - [Symbol.1] -- top**

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

Schematic Diagram:

- rst
- clk (3)
- c[7:0]
- b[3:0]
- a[3:0]
- top top
- out[3:0]

Register Name | Type |

q\_reg | Flip-flop |

Presto compilation completed success

Log History Errors/Warnings

design\_vision-t>

Loading db file '/usr/synopsys/synthesis/cur/libraries/syn/1\_25.font'

```

gui_zoom -window [gui_get_current_window -view] -factor 0.707
design_vision-t> gui_zoom -window [gui_get_current_window -view] -full
design_vision-t> gui_zoom -window [gui_get_current_window -view] -full
design_vision-t> gui_zoom -window [gui_get_current_window -view] -factor 0.707
design_vision-t> change_selection -name __slctBus258 -type {cell design} _sel1205
design_vision-t>
  
```

Log History Errors/Warnings

design\_vision-t>

Port: clk

# Schematic View

**Design Vision - TopLevel.1 - [Hier.1] -- top**

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

Logical Hierarchy

- top
  - U4
  - U3** (1)
  - U2
  - U1

Cells (Hierarchic) Create Symbol View

Cell Name	Ref Name	Cell Path	Dont Touch
div_27	div_DW_div_un...	U3/div_27	undefined

**Design Vision - TopLevel.1 - [Schematic.1] -- top**

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

Schematic View

div\_in2[7:0] → div\_in2[7:0] → b[7:0]

div\_in1[7:0] → div\_in1[7:0] → a[7:0]

div\_27  
div\_DW\_div\_uns\_8\_8\_0

remainder[7:0]

quotient[7:4]

quotient[3:0]

div\_out[3:0]

divide\_by\_0

**U3** (3)

design\_vision-t> current\_design  
Current design is 'mul'.  
design\_vision-t> current\_design  
Current design is 'top'.  
design\_vision-t>  
Log History Errors/Warnings

design\_vision-t> Create Symbol View of currently selected c

design\_vision-t> /users2/cic/andy/SYNOPSYS/LAB/lab1/top.db:mul  
Current design is 'mul'.  
design\_vision-t> current\_design /users2/cic/andy/SYNOPSYS/LAB/lab1/top.db:top  
Current design is 'top'.  
design\_vision-t> gui\_zoom -window [gui\_get\_current\_window -view] -full  
design\_vision-t>  
Log History Errors/Warnings

Cell: U3

---

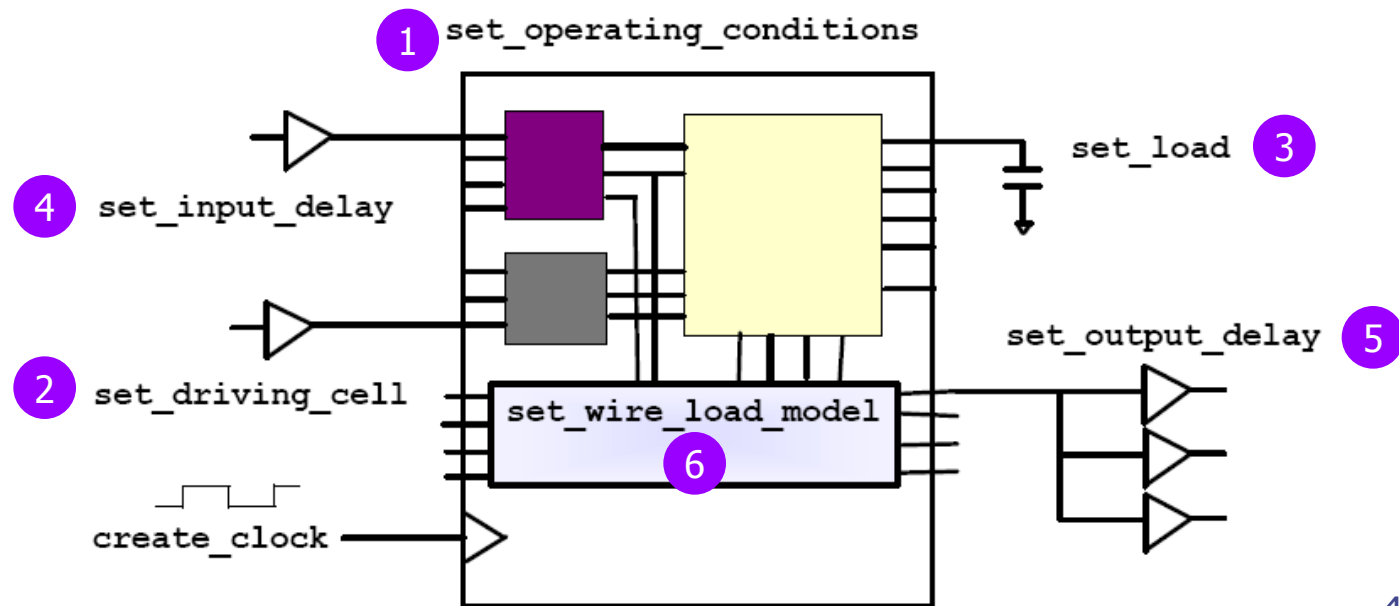
## 5. Setting Design Environment & Constraints





# Setting Design Environment

- ◆ Beware the defaults are not realistic conditions
  - Input **drive** is **not infinite**
  - Capacitive **loading** is usually **not zero**
  - Consider **process, voltage, temperature (PVT)** variation
- ◆ The operating environment affects the components selected from target library and timing through your design
- ◆ The real world environment you define describes the conditions that the circuit will operate within



## ◆ Design Constraints

- Timing



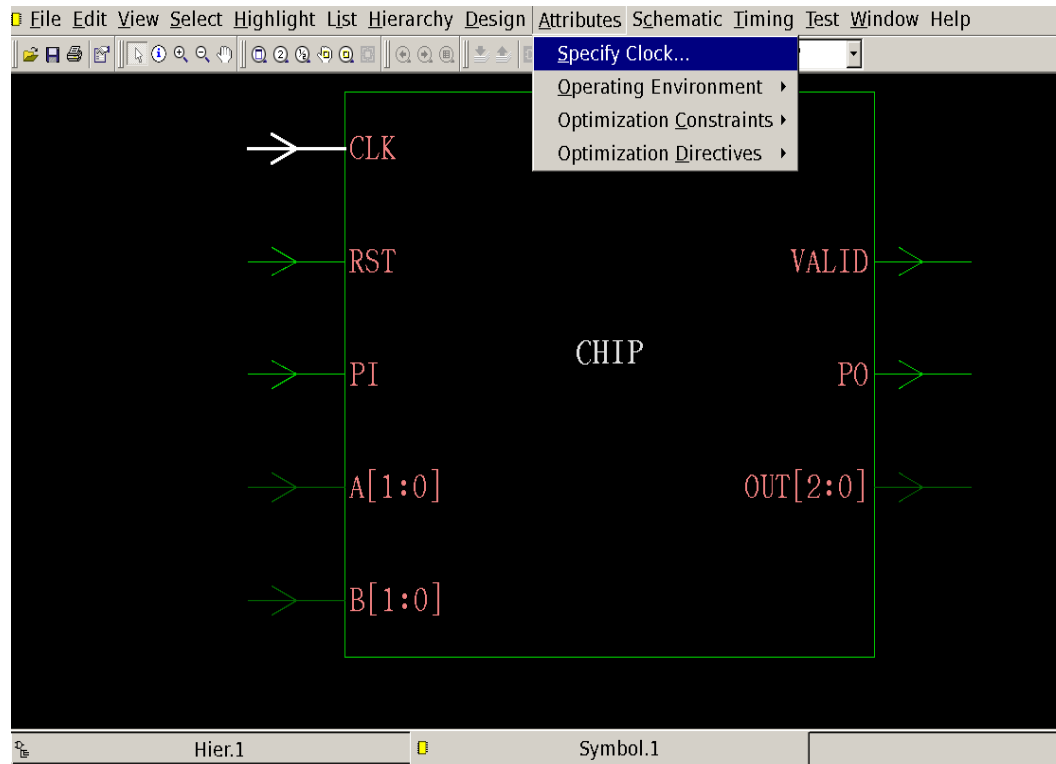
- Area

- Power



# Sequential Circuit → Specify Clock

- ◆ Select clock port
- ◆ *Attributes/Specify Clock*



**Specify Clock**

Clock name:

Port name:

☐ Remove clock

Clock creation

Period:

Edge	Value
Rising	0.000
Falling	10.000

☐ Don't touch network

☒ Fix hold

**Command: create\_clock -period 20 [get\_ports clk]**

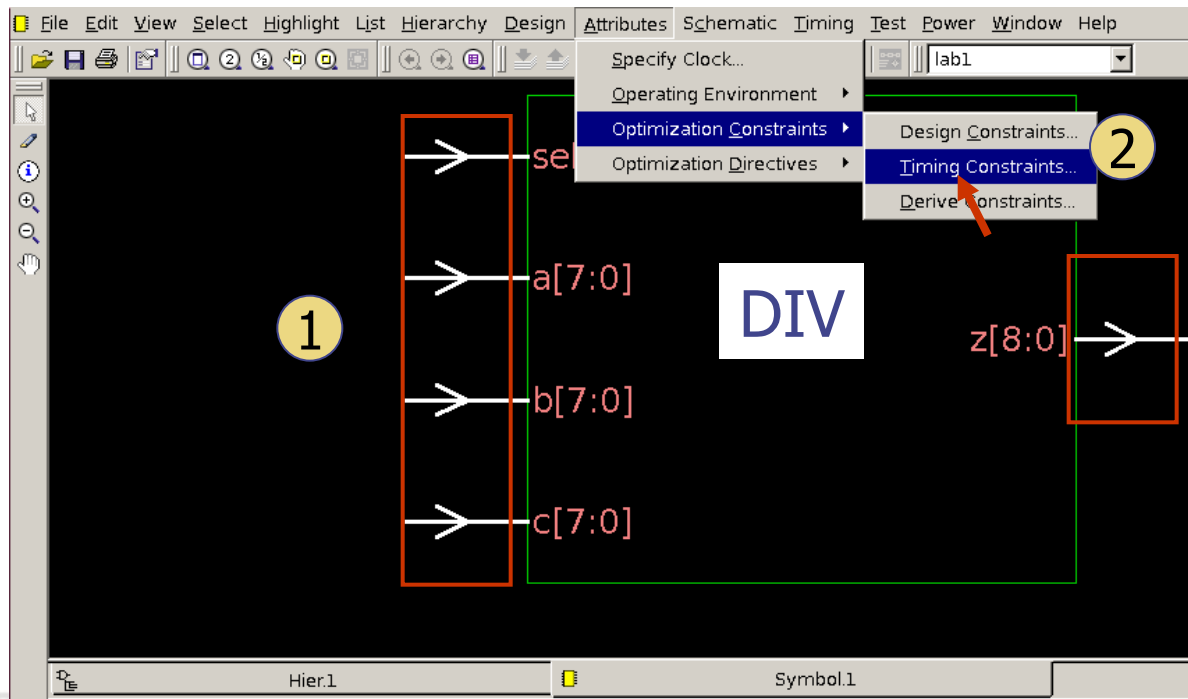
# Define Clock Specification

---

- ◆ 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 (option)
    - Network latency
  - Transition
    - Input transition
    - Clock transition
  
- ◆ All register-to-register path are constrained now

# Combinational Circuit - Maximum Delay Constraints

- ◆ For combinational circuits primarily
  - Select the start & end points of the timing path
  - *Attributes/Optimization Constraints/Timing Constraints*



The 'Timing Constraints' dialog box is shown. The 'From' field contains a list of signals: c[6], c[5], c[4], c[3], c[2]. The 'To' field contains a list of signals: z[7], z[6], z[5]. The 'Delays' section has a checked box for 'Same rise and fall', with 'Max rise' and 'Max fall' both set to '1'. The 'Group name' field is empty. The 'Reset path' checkbox is unchecked. A yellow circle '3' is next to the 'Delays' section, and a yellow circle '4' is next to the 'OK' button.

# Advanced Clock Constraints

---

- Multiple Clocks Domain Design
- Positive & Negative edge Clock Trigger
- Asynchronous Clocks Domain Design
- Clock Gating
- Multicycle Path
- How to Disable Timing Analysis for Special Path?

⋮



## ◆ Design Constraints

- Timing
- Area
- Power



# Setting Area Constraint

- ◆ *Attributes/Optimization Constraints/Design Constraints*
- ◆ If you only want to concern the area, but don't care the timing. You can use the following constraint script

**Area unit :**

(1) equivalent gate counts

(2) *um x um*

(3) *Transistors*

```
set max_area 0
```

Design Constraints

Current design: top

Optimization constraints

Constraint value: Unit:

Max area : 6000.000000

Max dynamic power: 0.0 uW

Max leakage power: 0.0 uW

Max total power: 0.0 uW

Design rules

Max fanout: 2.000000

Max transition: 0.300000

OK Cancel Apply



# Area Optimization

Beginning Area-Recovery Phase

ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL NEG SLACK	DESIGN RULE COST					
0:02:00	10086.2	0.00	0.0	0.0	0:03:08	7784.2	0.00	0.0	0.0
0:02:00	10066.2	0.00	0.0	0.0	0:03:22	7770.9	0.00	0.0	0.0
0:02:00	10029.6	0.00	0.0	0.0	0:03:22	7770.9	0.00	0.0	0.0
0:02:00	10019.6	0.00	0.0	0.0	0:03:23	7764.2	0.00	0.0	0.0
0:02:00	10006.3	0.00	0.0	0.0	0:03:25	7757.5	0.00	0.0	0.0
0:02:00	9973.0	0.00	0.0	0.0	0:03:25	7757.5	0.00	0.0	0.0
0:02:00	9949.8	0.00	0.0	0.0	0:03:27	7747.6	0.00	0.0	0.0
0:02:00	9923.1	0.00	0.0	0.0	0:03:30	7724.3	0.00	0.0	0.0
0:02:01	9869.9	0.00	0.0	0.0	0:03:31	7714.3	0.00	0.0	0.0
0:02:01	9859.9	0.00	0.0	0.0	0:03:31	7711.0	0.00	0.0	0.0
0:02:01	9820.0	0.00	0.0	0.0	0:03:31	7694.3	0.00	0.0	0.0
0:02:01	9820.0	0.00	0.0	0.0	0:03:32	7691.0	0.00	0.0	0.0
0:02:01	9806.7	0.00	0.0	0.0	0:03:32	7684.4	0.00	0.0	0.0
0:02:01	9793.4	0.00	0.0	0.0	0:03:35	7667.7	0.00	0.0	0.0
0:02:01	9780.1	0.00	0.0	0.0	0:03:44	7664.4	0.00	0.0	0.0
0:02:01	9753.5	0.00	0.0	0.0	0:03:44	7651.1	0.00	0.0	0.0
0:02:01	9740.2	0.00	0.0	0.0	0:03:48	7644.4	0.00	0.0	0.0
0:02:01	9710.2	0.00	0.0	0.0	0:03:51	7627.8	0.00	0.0	0.0
0:02:01	9683.6	0.00	0.0	0.0	0:03:56	7621.1	0.00	0.0	0.0
0:02:02	9657.0	0.00	0.0	0.0	0:04:00	7611.2	0.00	0.0	0.0
0:02:02	9630.4	0.00	0.0	0.0	0:04:00	7604.5	0.00	0.0	0.0
0:02:02	9603.8	0.00	0.0	0.0	0:04:00	7601.2	0.00	0.0	0.0
0:02:02	9590.5	0.00	0.0	0.0	0:04:02	7597.8	0.00	0.0	0.0
0:02:02	9563.8	0.00	0.0	0.0	0:04:05	7587.9	0.00	0.0	0.0
					0:04:07	7577.9	0.00	0.0	0.0
					0:04:07	7564.6	0.00	0.0	0.0
					0:04:07	7551.3	0.00	0.0	0.0

Optimization Complete

Area

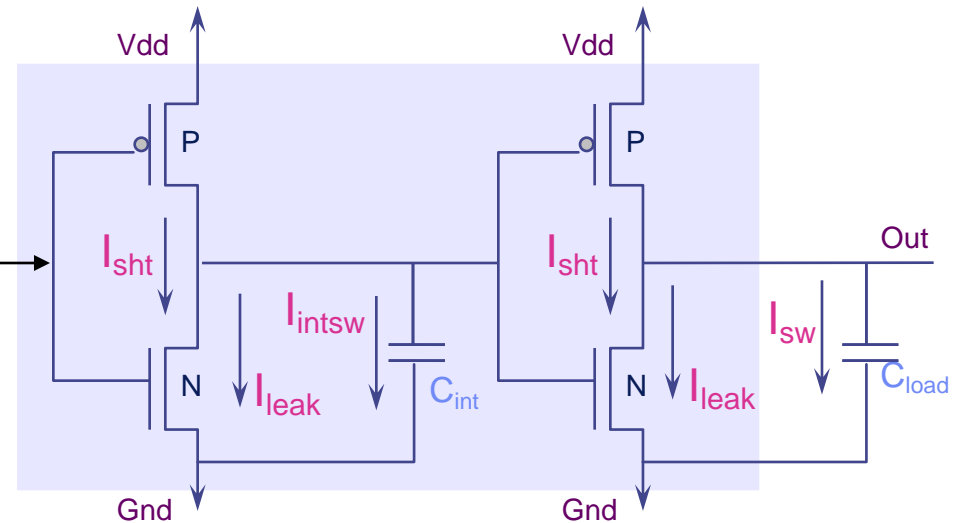
## ◆ Design Constraints

- Timing
- Area
- Power



# Power Model

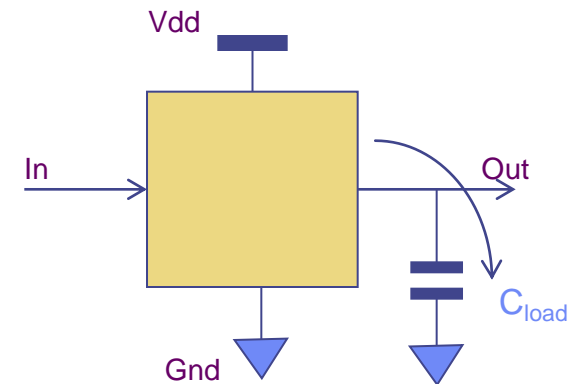
- ◆ CMOS cell power model
- ◆ Switch power (dynamic):
  - Charging output load
- ◆ Internal power (dynamic):
  - Short circuit;
  - Charging internal load
- ◆ Leakage power (static):
  - Stable state



**Internal power != Short Circuit Power**



**Who consumes switching power?**



# Power Optimization

```
dc_shell-xg-t> compile
dc_shell-xg-t> report_power
```

...

```
dc_shell-xg-t> set_max_total_power 0 uw
dc_shell-xg-t> compile -inc
```

...

```
dc_shell-xg-t> report_power
```

## w/o Power Optimization:

Cell Internal Power	=	247.2959 uW	(45%)
Net Switching Power	=	306.2117 uW	(55%)
<hr/>			
Total Dynamic Power	=	553.5076 uW	(100%)
Cell Leakage Power	=	2.6536 uW	

## w/ Power Optimization:

Cell Internal Power	=	225.5084 uW	(43%)
Net Switching Power	=	293.5588 uW	(57%)
<hr/>			
Total Dynamic Power	=	519.0672 uW	(100%)
Cell Leakage Power	=	2.4994 uW	

## Power Improve:

$$(553.5 - 519) / 553.5 = 6.23\%$$

Beginning Dynamic Power Optimization (max\_total\_power 0 uw)

ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL NEG SLACK	DESIGN RULE COST	ENDPOINT	TOTAL POWER
0:00:02	37786.6	0.00	0.0	0.0		550289024.0000
0:00:04	37671.2	0.00	0.0	0.0		538394432.0000
0:00:04	37647.4	0.00	0.0	0.0		532568320.0000

Beginning Leakage Power Optimization (max\_total\_power 0 uw)

ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL NEG SLACK	DESIGN RULE COST	ENDPOINT	TOTAL POWER
0:00:04	37647.4	0.00	0.0	0.0		532568320.0000
0:00:06	37628.8	0.00	0.0	0.0		529273184.0000

Power Optimization



# Advanced Power Optimization

---

## Dynamic power optimization:

- ① ♦ RTL: Clock-Gating (20% - 40% , depends on your design)
- ② ♦ RTL: Operand Isolation (5% - 20% , depends on your design)
- ③ ♦ EDA: Gate level Dynamic power optimization (2 - 6% )
- ④ ♦ Library: MVMS (Multi-VDD Multi-Supply) <-- Library must support PMK  
(MSV Design: Synopsys UPF / Cadence CPF)

## ⑤ Leakage power optimization:

- ♦ EDA : Leakage Power optimization (20 - 80% )  
(90nm UMC/Faraday Support Multi- $V_T$  Cell)



---

## 6. Compiler Design (Synthesis)



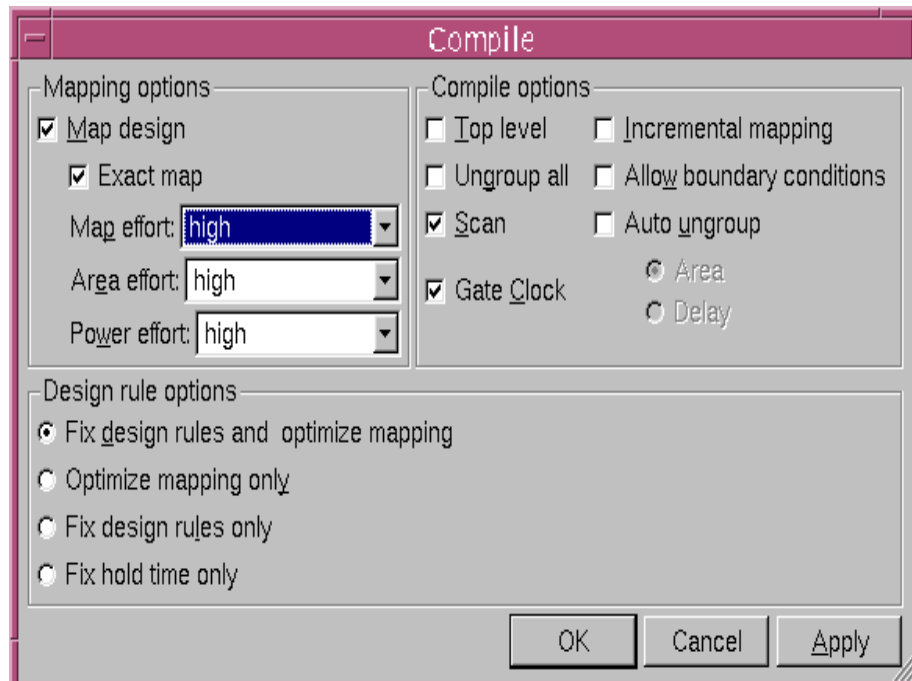
# Mapping Effort

- ◆ 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

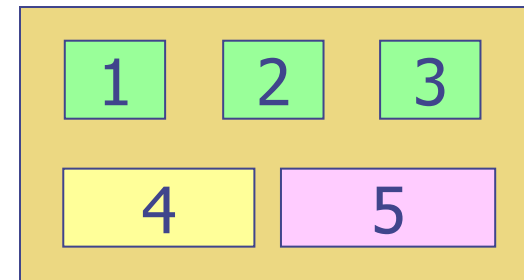
Ultra high:

compile\_ultra

2010.03-sp5



ex:



---

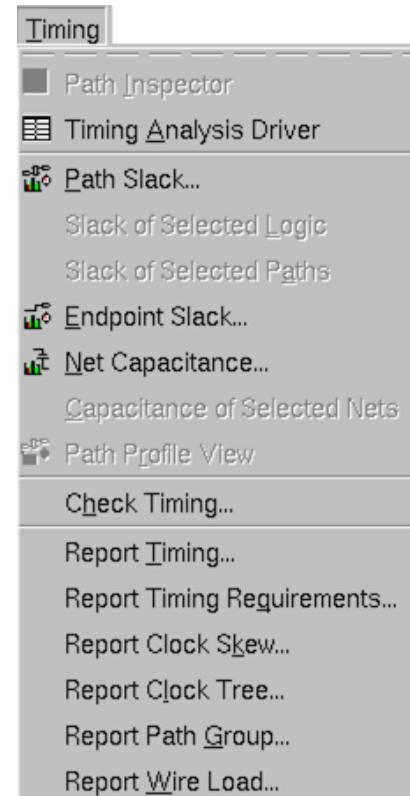
## 7. Synthesis Report and Analysis





# Analyze the Design

- ◆ In Design Vision, you can find the set attributes and the results after optimization from Design report and Timing analysis



# Report Timing (2/2)

- Timing report shows maximum or minimum delay path of design, the default is to display one maximum delay path

	Point	Incr	Path
-----			
	clock clk (rise edge)	0.00	0.00
	clock network delay (ideal)	1.00	1.00
	input external delay	1.00	2.00 f
start point	set_time (in)	0.10	2.10 f
	microwave/set_time (microwave)	0.00	2.10 f
	microwave/micro_st/set_time (micro_st)	0.00	2.10 f
	microwave/micro_st/U4/Y (BUF10TH)	0.13	2.24 f
	microwave/micro_st/U5/Y (CLKNAND2X8TH)	0.05	2.29 r
	microwave/micro_st/U20/Y (CLKINX4TH)	0.08	2.37 f
	microwave/micro_st/U19/Y (NAND2X6TH)	0.09	2.46 r
	microwave/micro_st/U21/Y (INVX3TH)	0.07	2.53 f
	microwave/micro_st/U24/Y (OR2X8TH)	0.13	2.65 f
	microwave/micro_st/load_clk (micro_st)	0.00	2.65 f
	microwave/loader/load_clk (loader)	0.00	2.65 f
	microwave/loader/U19/Y (NAND2BX8TH)	0.12	2.77 f
	microwave/loader/load (loader)	0.00	2.77 f
	microwave/load (microwave)	0.00	2.77 f
	timer/load (timer)	0.00	2.77 f
	timer/U51/Y (BUF18TH)	0.11	2.88 f
	timer/U50/Y (CLKINX6TH)	0.08	2.96 r
	timer/U43/Y (CLKINX24TH)	0.13	3.09 f
	timer/U23/Y (OAI22X1TH)	0.29	3.38 r
	timer/U40/Y (BUF6TH)	0.19	3.57 r
	timer/U101/Y (NAND4X8TH)	0.21	3.79 f
	timer/U64/Y (OR2X4TH)	0.21	4.00 f
	timer/U25/Y (CLKINX12TH)	0.07	4.07 r
	timer/U53/Y (NAND4X8TH)	0.18	4.25 f
	timer/U34/Y (NAND2BX4)	0.20	4.45 f
	timer/U75/Y (OAI32X2TH)	0.31	4.77 r
end point	timer/min_msb_next_reg[2]/D (DFFQX4)	0.00	4.77 r
	data arrival time		4.77
-----			
	clock clk (rise edge)	4.00	4.00
	clock network delay (ideal)	1.00	5.00
	clock uncertainty	-0.10	4.90
	timer/min_msb_next_reg[2]/CK (DFFQX4)	0.00	4.90 r
	library setup time	-0.12	4.78
	data required time		4.78
-----			
	data required time		4.78
	data arrival time		-4.77
-----			
	slack (MET)		0.01

Net\_delay + cell\_delay  
are combined

# Report Area

- ◆ Area report shows the  $\mu\text{m}^2$  of the design
- ◆ Design / Report Area

```
*****
Report : area
Design : top
Version: D-2010.03-SP5
Date   : Wed Dec 1 13:57:38 2010
*****
```

report\_area -hier

Library(s) Used:

```
slow (File: /user/DSD/andy/SYNOPSYS/core/slow_hvt.db)
slow (File: /user/DSD/andy/SYNOPSYS/core/slow.db)
```

```
Number of ports:      49
Number of nets:       86
Number of cells:      5
Number of references: 5
```

```
Combinational area:   1306.065621
Noncombinational area: 342.215997
Net Interconnect area: 31873.492676
```

```
Total cell area:      1648.281618
Total area:            33521.774294
```

don't care net area

Hierarchical area distribution

Hierarchical cell	Global cell area		Local cell area			Design
	Absolute Total	Percent Total	Combi-national	Noncombi-national	Black boxes	
top	1648.2815	100.0	4.2336	0.0000	0.0000	top
display	330.2209	20.0	0.0000	0.0000	0.0000	display
display/convm1	82.5552	5.0	82.5552	0.0000	0.0000	converter_2
display/convm2	82.5552	5.0	82.5552	0.0000	0.0000	converter_3
display/convs1	82.5552	5.0	82.5552	0.0000	0.0000	converter_0
display/convs2	82.5552	5.0	82.5552	0.0000	0.0000	converter_1
microwave	299.1744	18.2	0.0000	0.0000	0.0000	microwave
microwave/loader	112.1904	6.8	112.1904	0.0000	0.0000	loader
microwave/micro_st	186.9840	11.3	129.1248	57.8592	0.0000	micro_st
timer	1014.6529	61.6	730.2960	284.3568	0.0000	timer
Total			1306.0657	342.2160	0.0000	

# Report Power

## ◆ Design / Report Power

1

report\_power

Operating Conditions: slow Library: slow  
Wire Load Model Mode: top

Design	Wire Load Model	Library
top	tsmc090_wl10	slow

Global Operating Voltage = 0.9  
Power-specific unit information :  
Voltage Units = 1V  
Capacitance Units = 1.000000pf  
Time Units = 1ns  
Dynamic Power Units = 1mW (derived from V,C,  
Leakage Power Units = 1pW

Power Model

Cell Internal Power	=	86.2410 uW	(58%)
Net Switching Power	=	62.9489 uW	(42%)
-----			
Total Dynamic Power	=	149.1898 uW	(100%)
Cell Leakage Power	=	2.5576 uW	

2

report\_power -hier

Operating Conditions: slow Library: slow  
Wire Load Model Mode: top

Design	Wire Load Model	Library
top	tsmc090_wl10	slow

Global Operating Voltage = 0.9  
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

Hierarchy	Switch Power	Int Power	Leak Power	Total Power	%
top	6.29e-02	8.62e-02	2.56e+06	0.152	100.0
display (display)	1.72e-02	2.29e-03	1.76e+05	1.96e-02	12.9
convsl (converter_0)	8.04e-03	1.06e-03	4.53e+04	9.15e-03	6.0
convsm (converter_1)	3.39e-03	4.53e-04	4.43e+04	3.89e-03	2.6
convml (converter_2)	2.86e-03	3.85e-04	4.39e+04	3.29e-03	2.2
convmm (converter_3)	2.88e-03	3.88e-04	4.29e+04	3.31e-03	2.2
timer (timer)	3.10e-02	6.77e-02	1.93e+06	0.101	66.3
microwave (microwave)	1.43e-02	1.62e-02	4.53e+05	3.09e-02	20.3
loader (loader)	4.41e-03	1.52e-03	1.87e+05	6.12e-03	4.0
micro_st (micro_st)	9.84e-03	1.47e-02	2.66e+05	2.48e-02	16.3

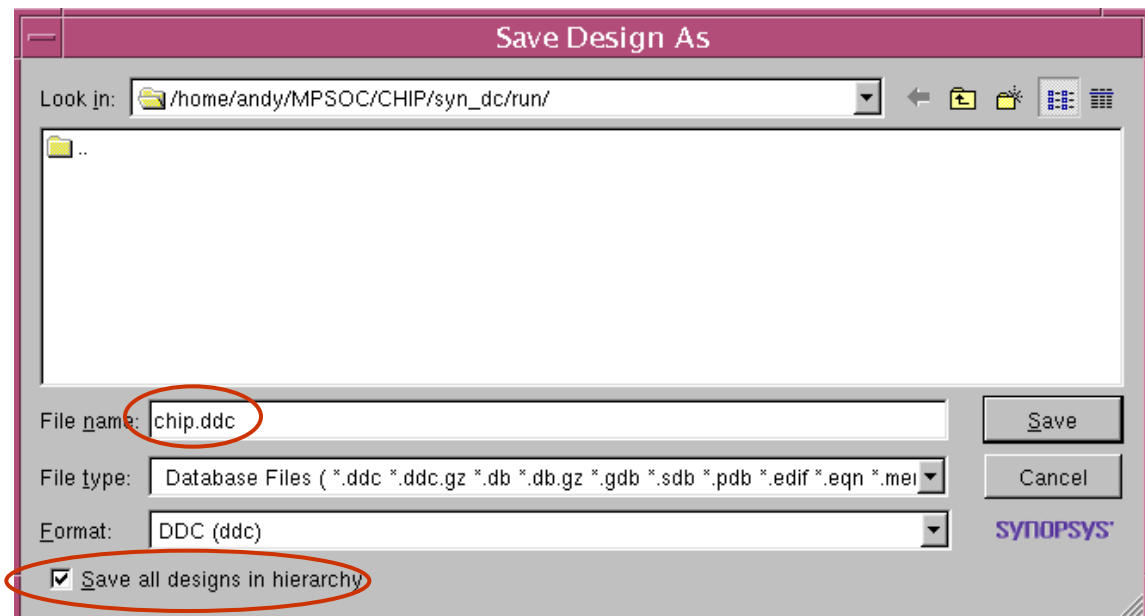
---

## 8. Save All Design & Run Simulation



# Save Design

- ◆ Save your design to file before you quit Design Compiler
- ◆ *File/Save* saves your design in the **ddc (or db)** format
- ◆ *File/Save As* can save your design in other Write formats:
  - PLA( Berkeley Espresso): .pla
  - Synopsys formats
    - ◆ equation: .eq
    - ◆ state table: .st
  - EDIF
  - Verilog: .v
  - VHDL: .vhd



# Change Naming Rule Script

- ◆ set bus\_inference\_style {%s[%d]}
- ◆ set bus\_naming\_style {%s[%d]}
- ◆ set hdlout\_internal\_busses true
- ◆ change\_names -hierarchy -rule verilog
- ◆ define\_name\_rules name\_rule -allowed "A-Z a-z 0-9 \_" -max\_length 255 -type **cell**
- ◆ define\_name\_rules name\_rule -allowed "A-Z a-z 0-9 \_[]" -max\_length 255 -type **net**
- ◆ define\_name\_rules name\_rule -map {"\\\*cell\\" "\*" "cell"}
- ◆ define\_name\_rules name\_rule -case\_insensitive
- ◆ change\_names -hierarchy -rules name\_rule

Bus[5]

## Example

```
input Tclk;
input BistMode;
output [8:0] S0;
input S1;
output S2;
output S3;
input S4;
input test_si;
output test_so;
input test_se;
wire n61, n62, n63, n64, n65, n66, n67, n68, N15, N16, N17, N18, N19, N20,
N21, N22, N23, N27, N28, N29, N30, N31, N32, N33, N34, N35, N53, n16,
n17, n19, n21, n23, n25, n27, n29, n31, n33, n35, n36, n46, n47, n48,
n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n69;

BUFX3 U50 ( .A(test_se), .Y(n69) );
ST_MAG_mbist_DW01_incdec_9_0 r521 ( .A({test_so, n61, n62, n63, n64, n65,
n66, n67, n68}), .INC_DEC(n36), .SUM({N35, N34, N33, N32, N31, N30,
N29, N28, N27}));
SDFFNXL S5_reg_8 ( .SI(n61), .SE(test_se), .D(n54), .CKN(Tclk), .Q(test_so),
.QN(n33) );
```

# Gate-Level Simulation (Verilog)

- ◆ Write out gate-level netlist (two methods)
  1. *File/Save As* → *Verilog (for File format)*
  2. **design\_vision> write -format verilog -hierarchy -output *chip\_syn.v***

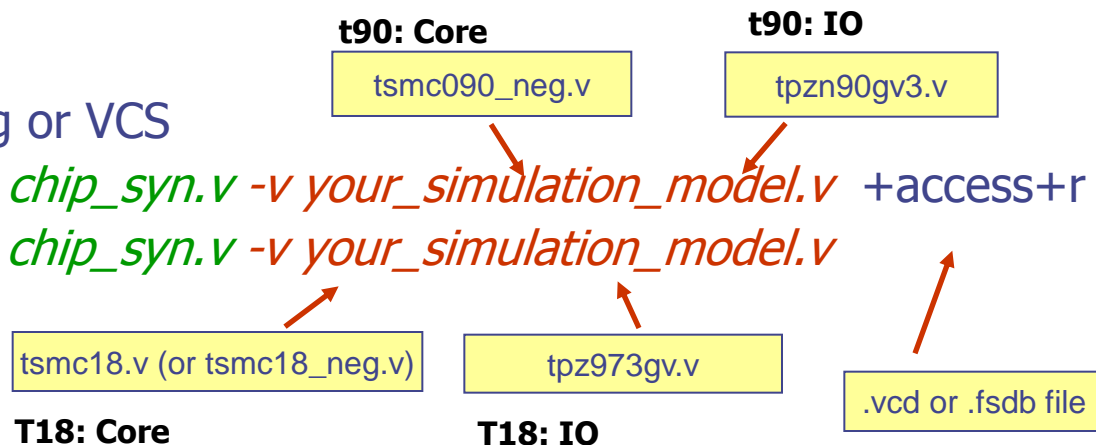
- ◆ Get SDF (Standard Delay Format)  
**design\_vision> write\_sdf -version 1.0 chip\_syn.sdf**

- ◆ Modify your testfixture file  
`$sdf_annotate("the_SDF_file_name",  
the_top_level_module_instance_name);`  
 For example: `$sdf_annotate("chip_syn.sdf", chip);`

1.0 or 2.1

- ◆ Simulation using NC-Verilog or VCS

unix% **ncverilog** *testfixture.v chip\_syn.v -v your\_simulation\_model.v +access+r*  
 unix% **vcs** *-R testfixture.v chip\_syn.v -v your\_simulation\_model.v*





# Timing Simulation

- ◆ Begin to simulate synthetic result. Is this function PASS?
- `unix% nverilog testfixture.v chip_syn.v -v tsmc18.v -v tpz973gv.v +access+r`

```

Loading snapshot worklib.test:v ..... Done
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

```

```

-----
All data have been generated successfully!
-----

```

PASS

```

-----
Simulation complete via $finish(1) at time 10110 NS + 0
./testfixture.v:128      $finish;
ncsim> exit

```

- `unix% nWave &`

