Design Compiler

Logic Synthesis (Synopsys)

王旭昇
hotline@cic.narl.org.tw
(03)577-3693 ext 885

Outline

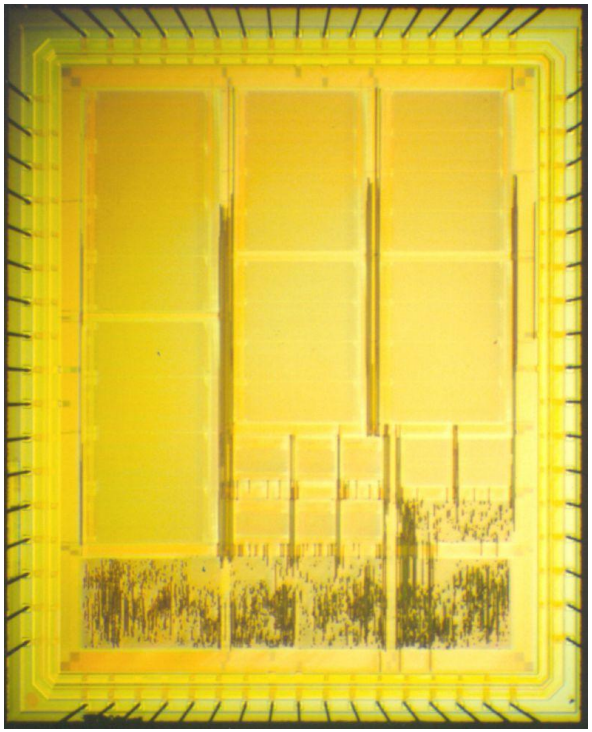
- Introduction
- Prepare your Verilog Code
- Process Setting
- Design Compiler GUI - Design Vision
- Setting Constraints
- Compiler
- Report and Analysis
- Save & Run Simulation

1. Introduction

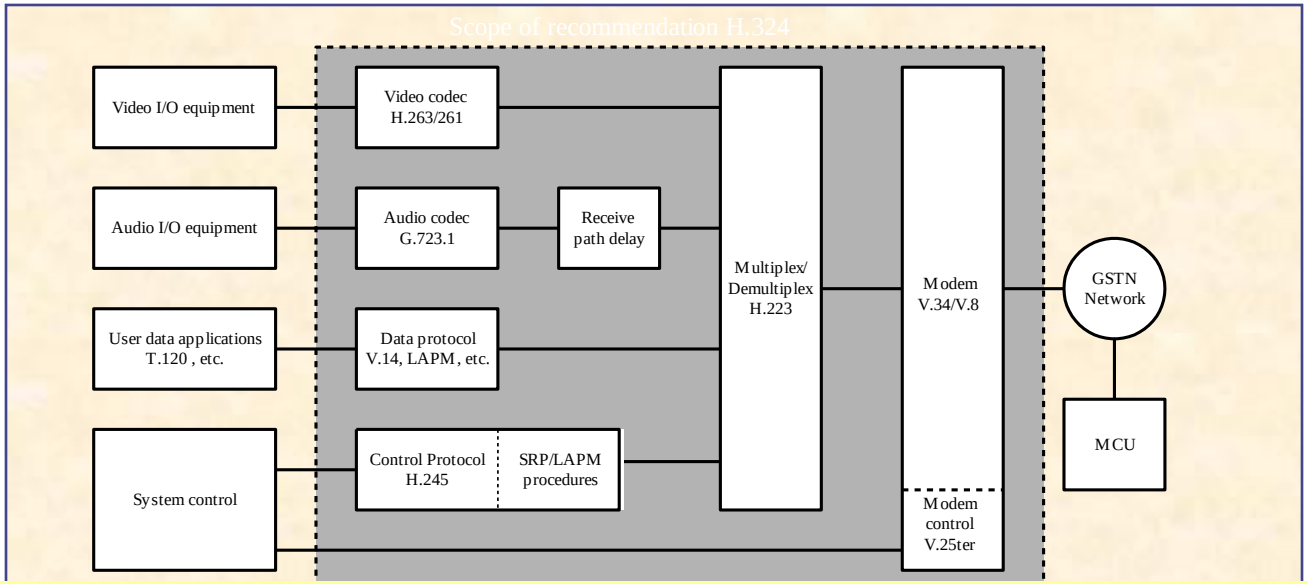
IC Design and Implementation

Idea

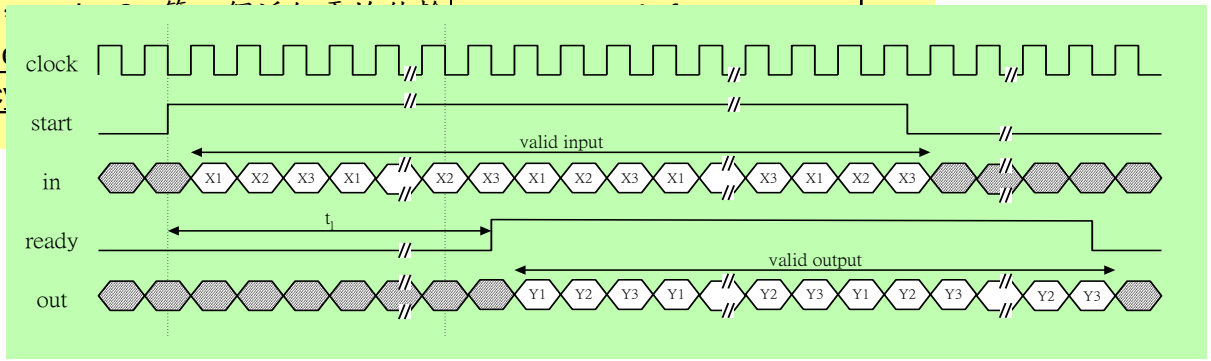
Chip



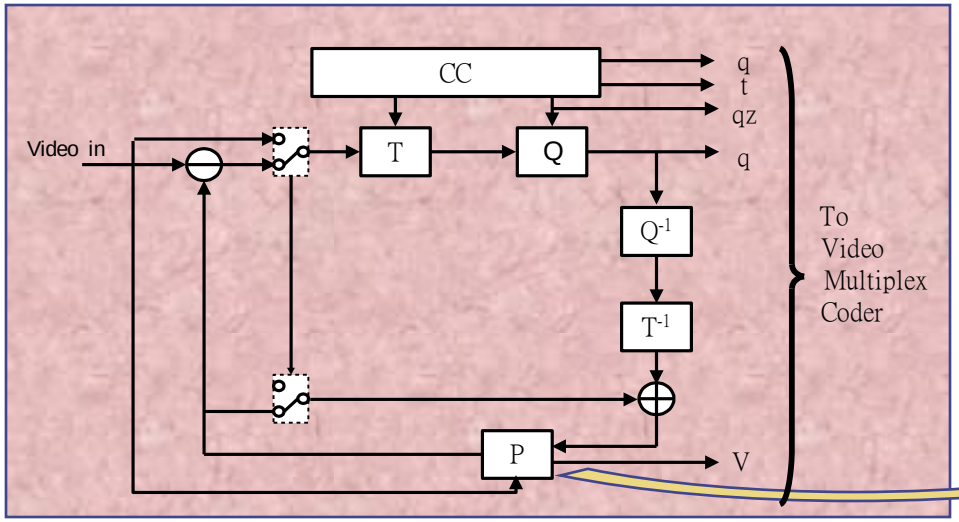
System Spec.



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 出前的半個	
dout	輸出	每個clock c	



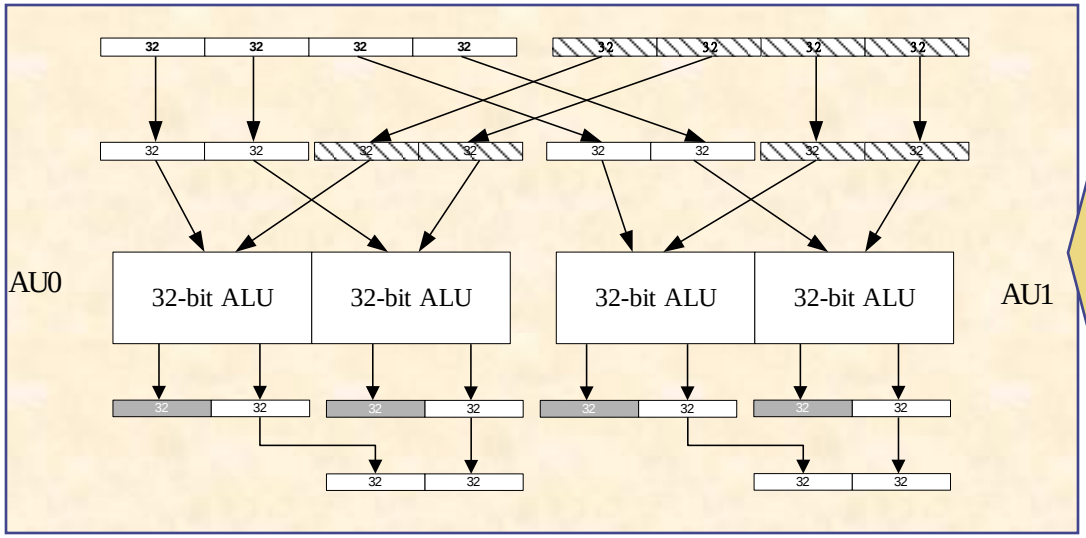
Algorithm Analysis



System Level

$$sad = \min SAD(k,l) = \min \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \left| I_c(i,j) - I_r(i+k, j+l) \right|$$

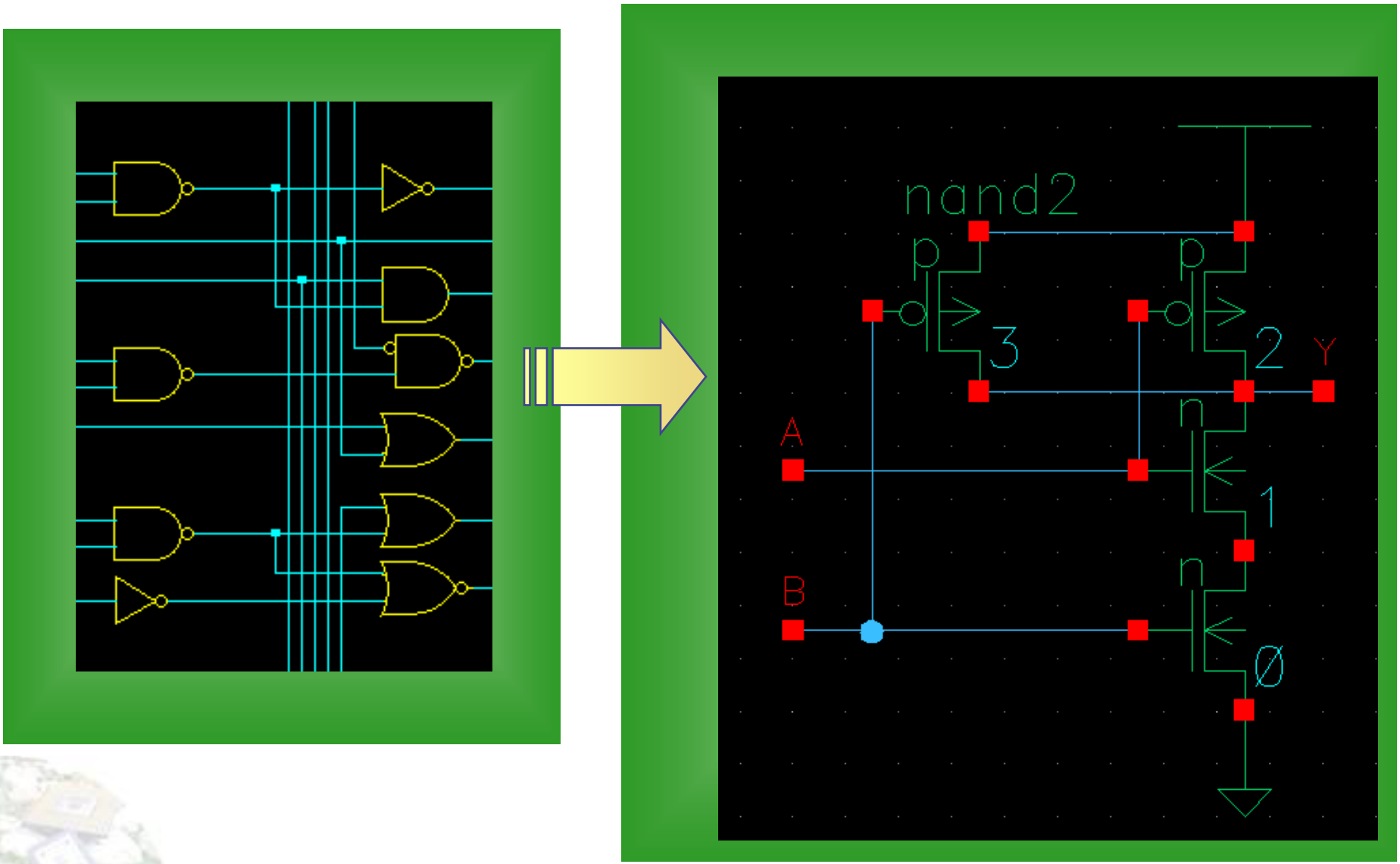
RTL Level



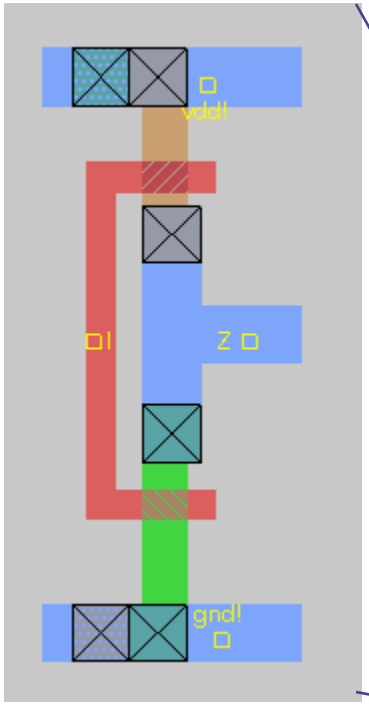
VLSI Architecture 推薦書籍:

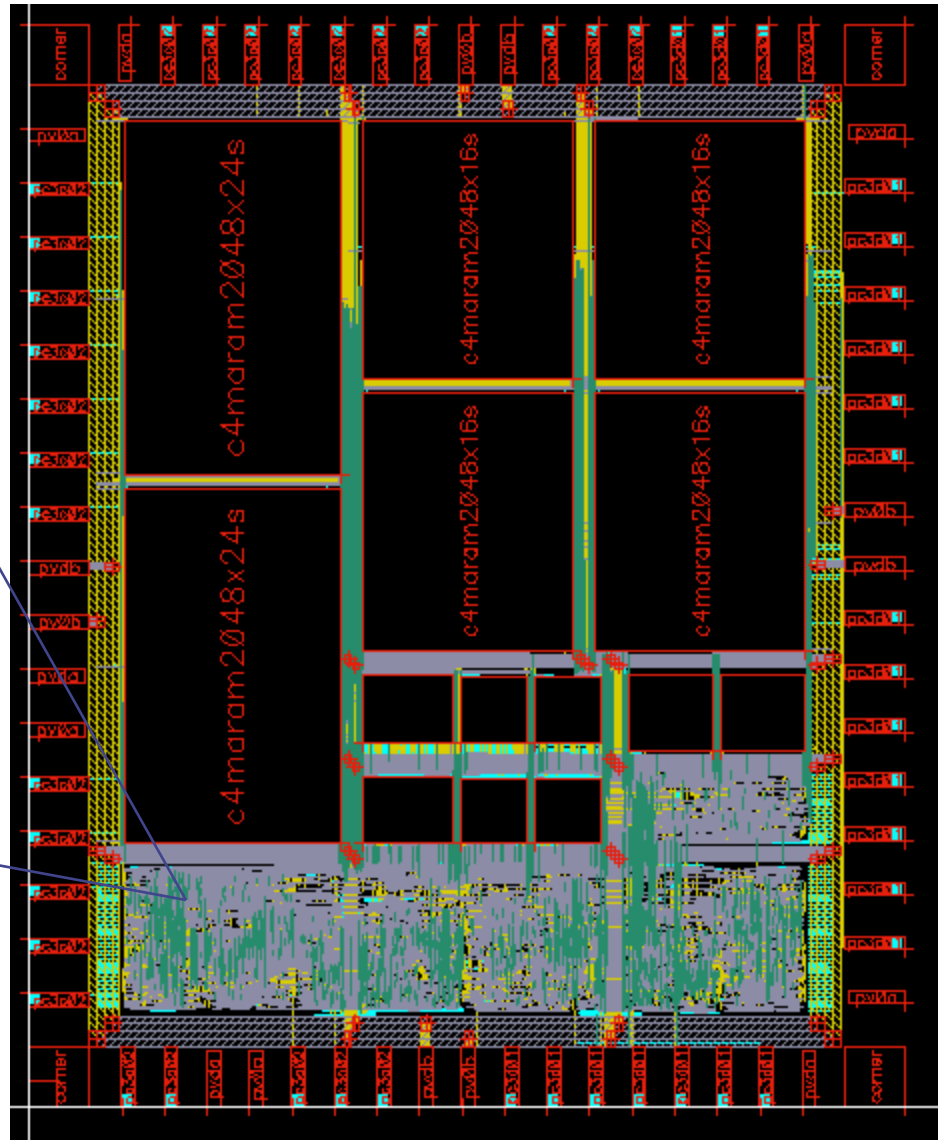
Parhi, "VLSI Digital Signal Processing Systems Design and Implementation"

Gate and Circuit Level Design



Physical Design





Cell-Based Design Flow

System Level

MATLAB/ C / C++ / System C
Platform Architecture

Memory Generator

RTL Level

Verilog / VHDL

NC-Verilog/NC-SIM / ModelSim
Verdi / VCS (MX)

TurboBist

Logic Synthesis

Conformal / Formality

Design / Power Compiler

CIC Cell Library
0.18um TSMC/ARM
90nm TSMC/ARM

Deisgn for Testing

DFT Compiler / TetraMAX

(design vision)

Gate Level

NC-Verilog/NC-SIM / ModelSim
Verdi / VCS (MX)

Layout Level Design

SOC Encounter / IC Compiler

GDS II

Post-Layout Verification

DRC / LVS (Calibre)

1

gate-level simulation

PVS : Calibre xRC /NanoSim
(Time/Power Mill)

Tape out

2

transistor-level post-layout simulation

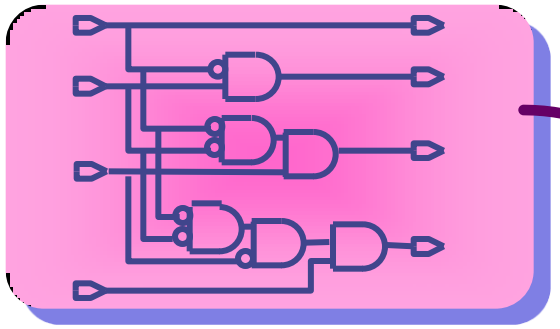
What is Synthesis

- 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)

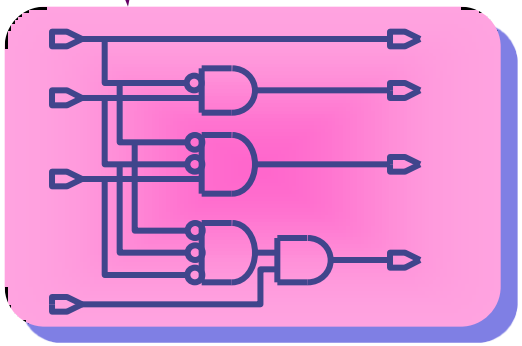


Generic Boolean
(GTECH)

no timing info.

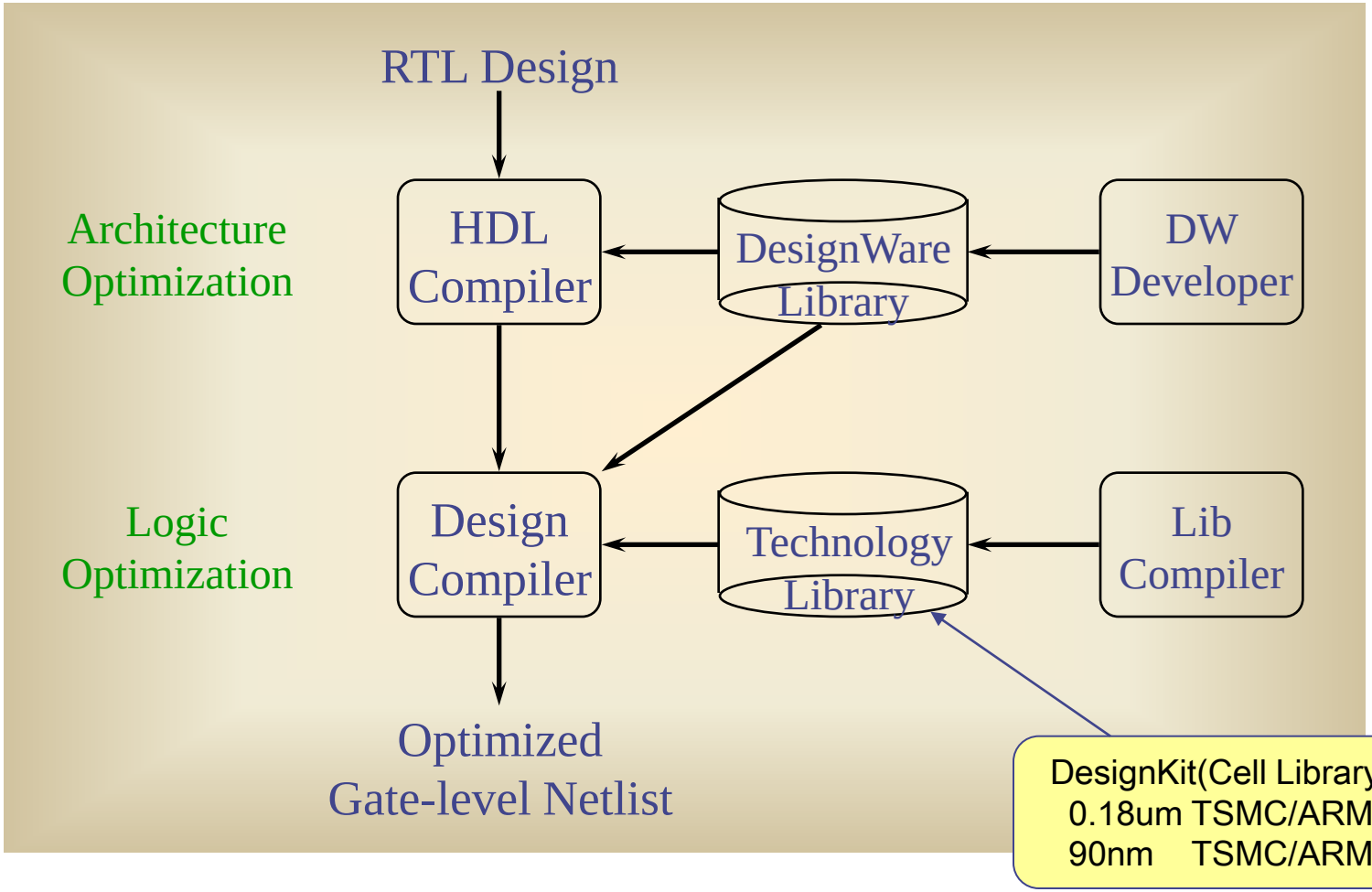
Optimize + Map
(Design Compiler)

timing info.



Target Technology

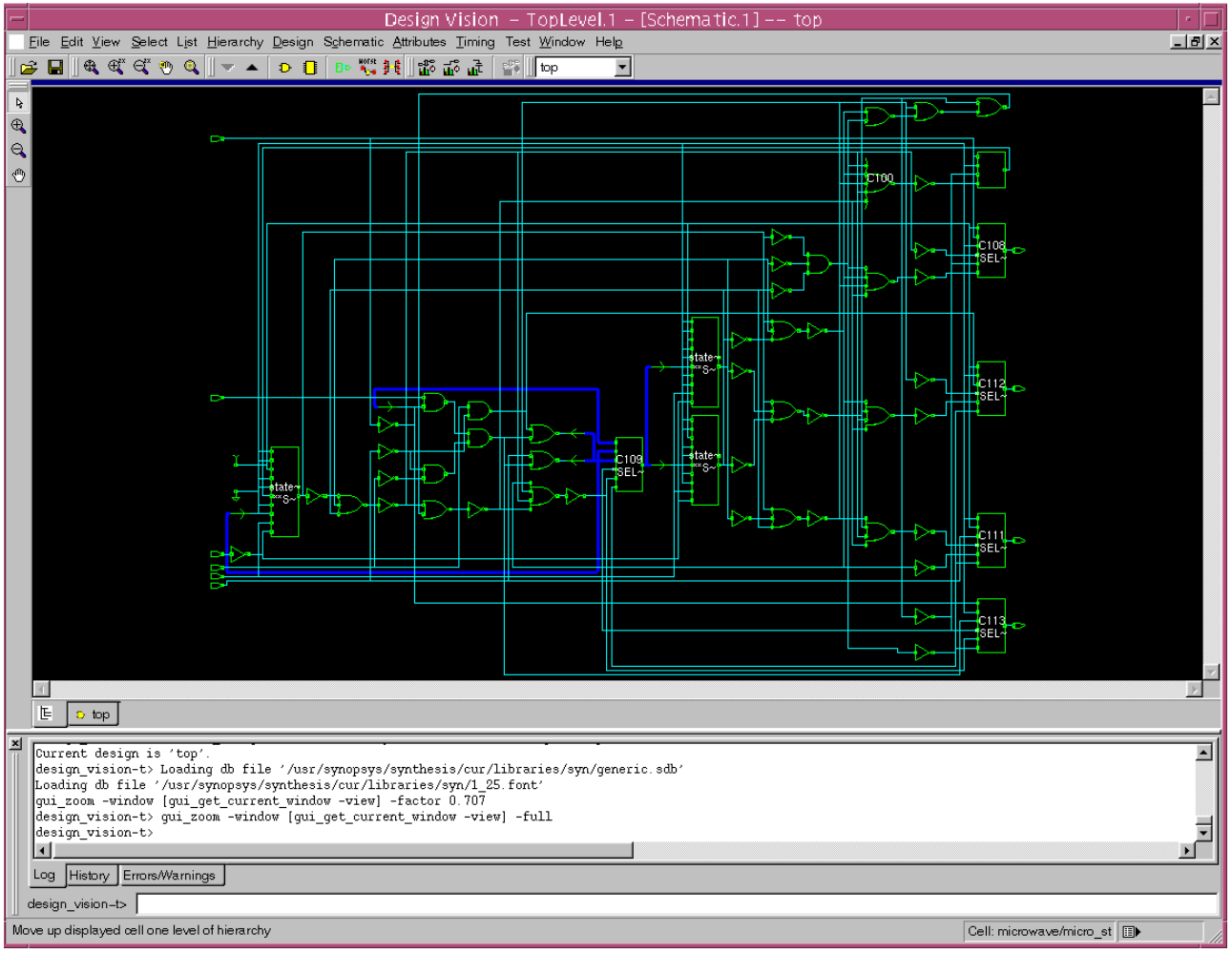
Logic Synthesis Overview



HDL Compiler

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

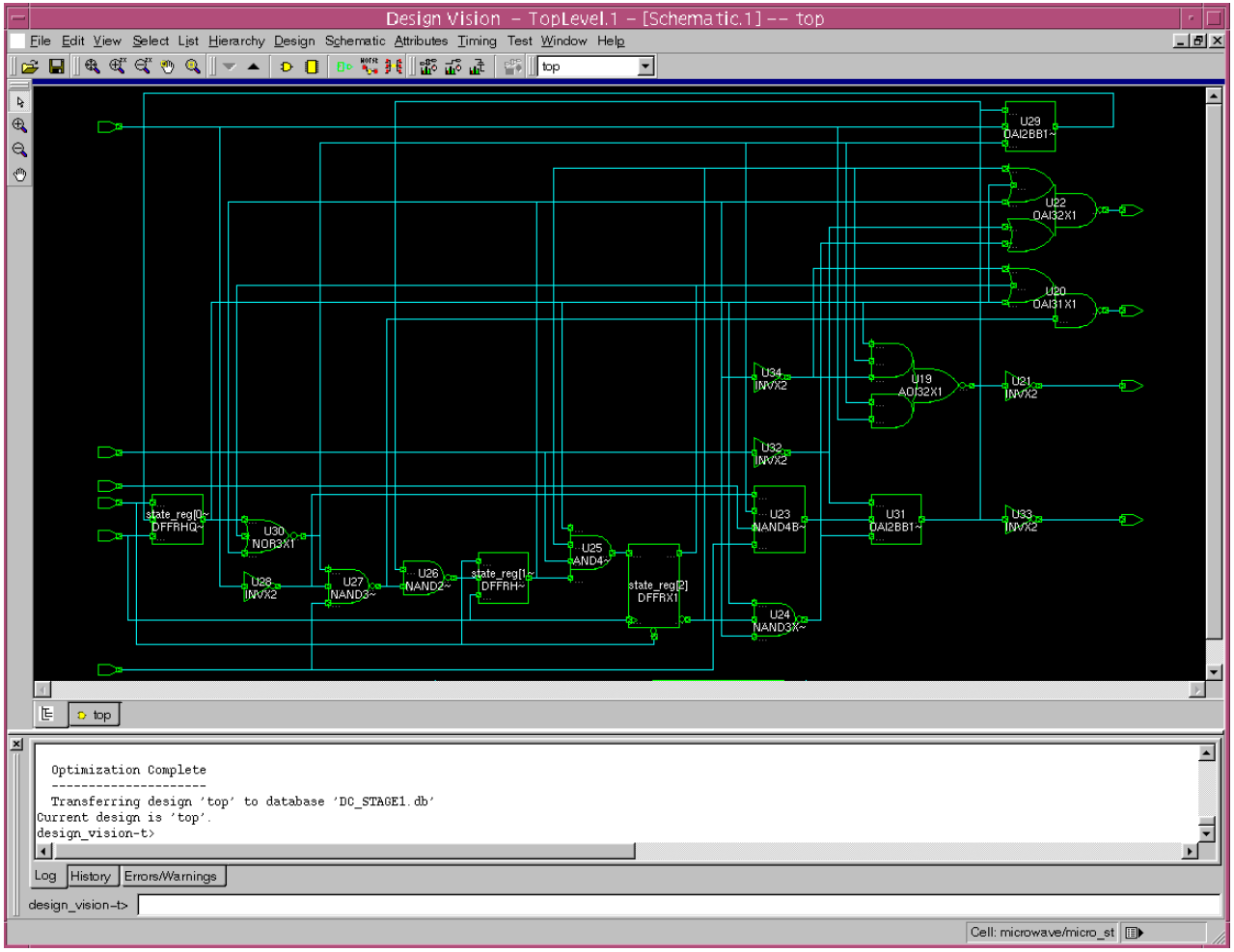
no timing info.



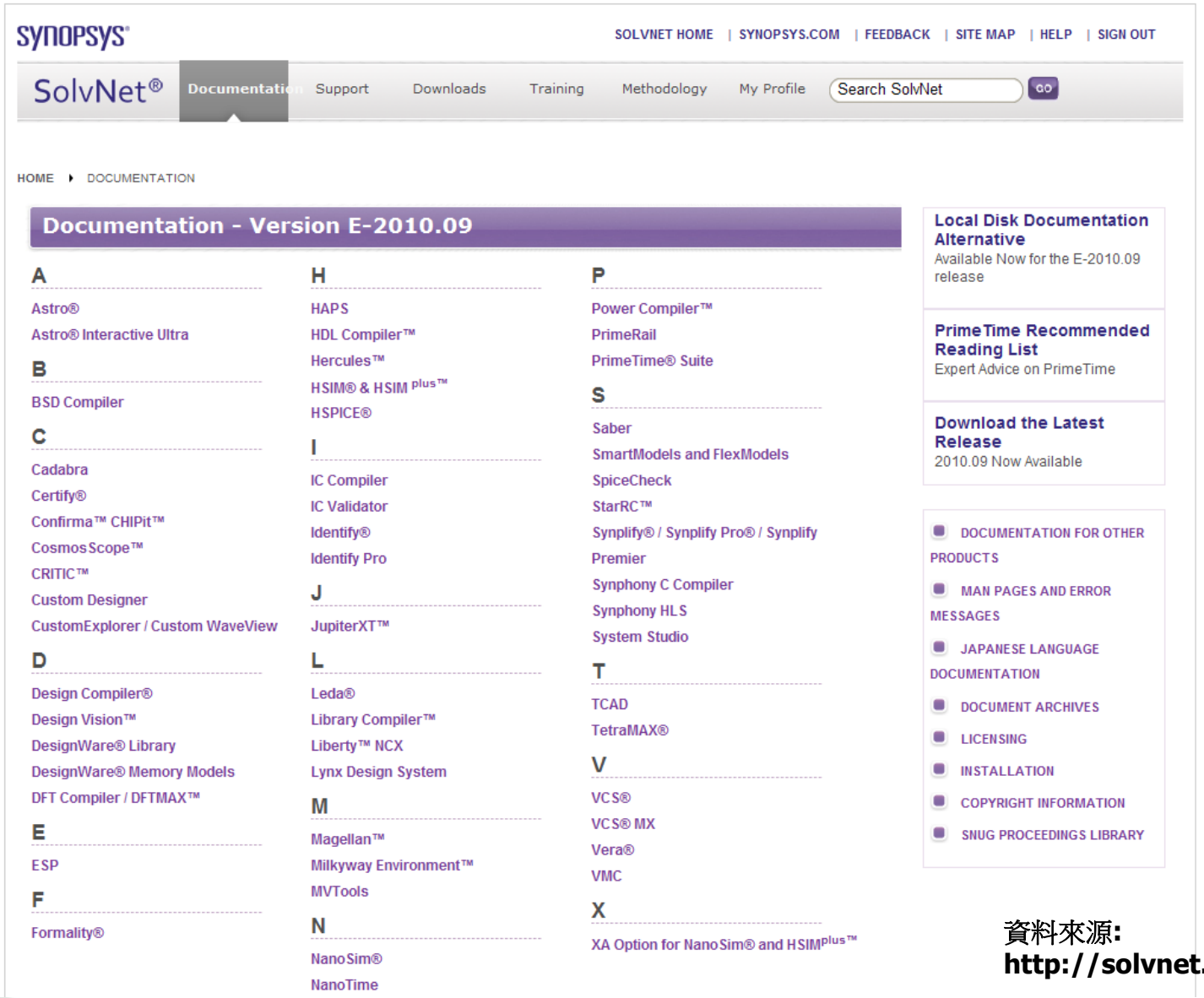
Design Compiler

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

timing info.



Synopsys On-Line Document (SOLD)

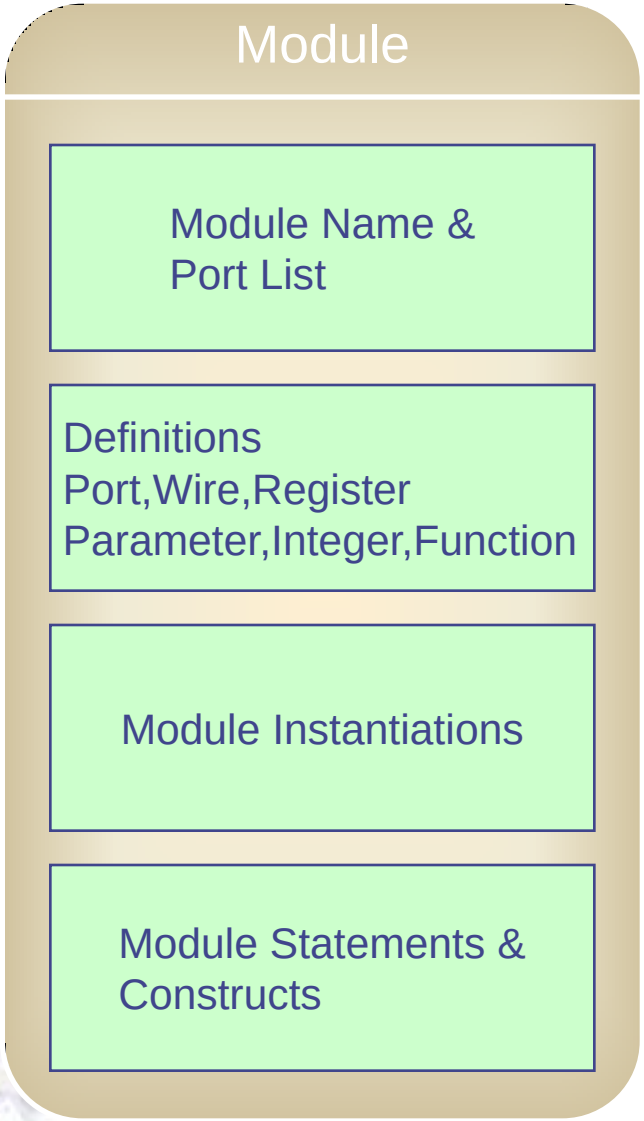


資料來源:
http://solvnet.synopsys.com

How to get the latest User Guide?

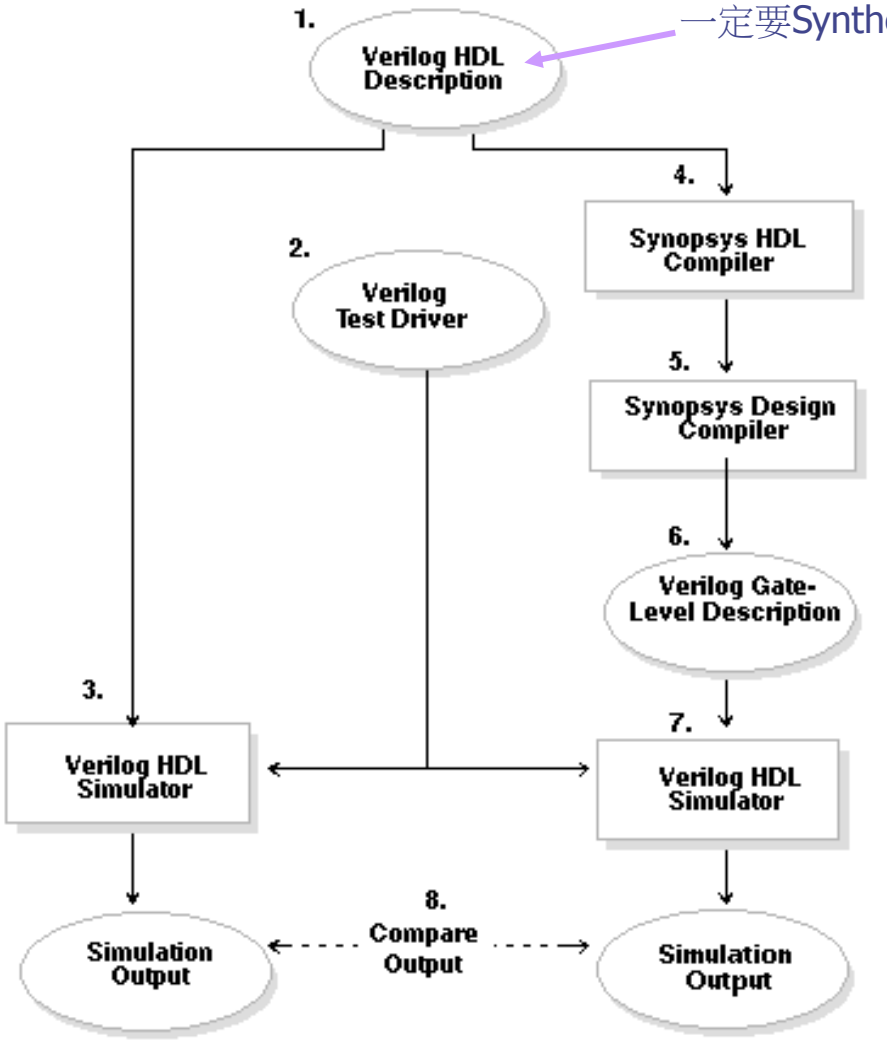
2. Prepare your Verilog Code

Verilog Module




```
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

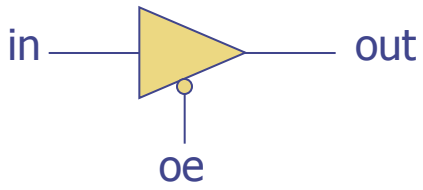
HDL Compiler Unsupported

- 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 (~,&,|,^,~^)
- Unary reduction (&,~&,|,~|,^,~^)
- 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
  - (6) if … else if … else
  - 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 declaraions
  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

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	Logic Synthesis with Design Compiler	王旭昇	2011-07-13~2011-07-15	新竹CIC	2,100	4,200	2011-07-01 23:59	已額滿 建議增開
100C104-F	Logic Synthesis with Design Compiler	王旭昇	2011-07-27~2011-07-29	成大自強校區電機館	2,100	4,200	2011-07-08 23:59	已額滿 建議增開
100C104-G	Logic Synthesis with Design Compiler	王旭昇	2011-08-08~2011-08-10	新竹CIC	2,100	4,200	2011-07-28 23:59	已額滿 建議增開
100C104-H	Logic Synthesis with Design Compiler	王旭昇	2011-08-17~2011-08-19	興大電機館孟堯晶片中心	2,100	4,200	2011-07-31 23:59	已額滿 建議增開

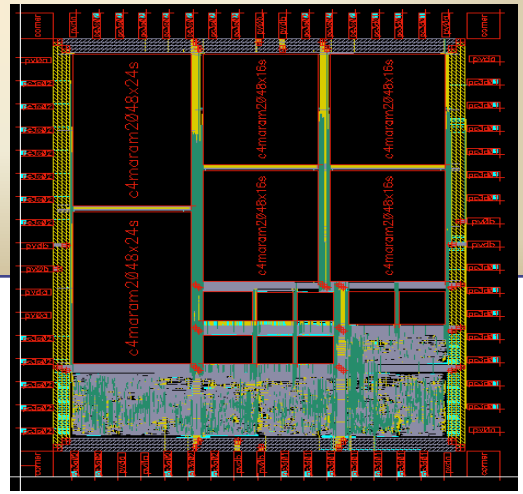
3. Process Setting for Synthesis Tool

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
```



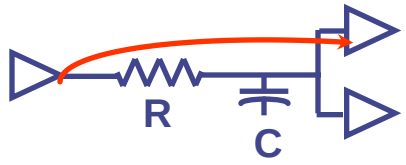
補充:

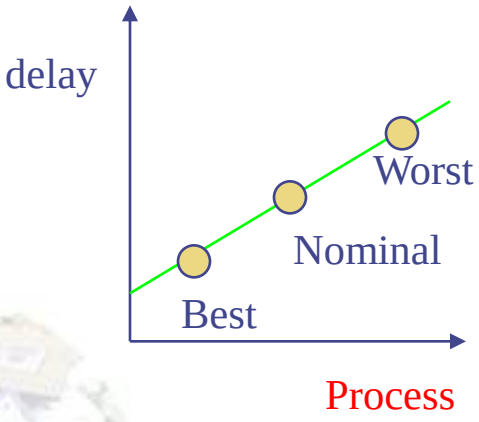
ex:
進入一個synopsys目錄
unix% cd synTab
=> cd synopsys (自動秀出)

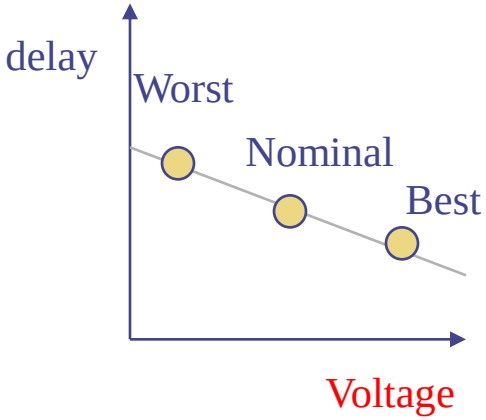
Slow v.s. Fast Library

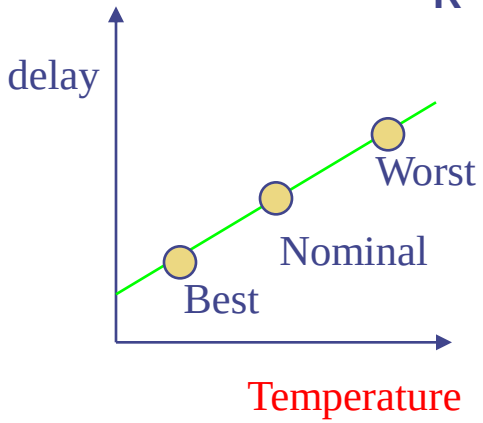
- Operating condition model scales components delay, directs the optimizer to simulate variations in process, temperature, and voltage

Name	*Process*	*Temp*	*Volt*	*Interconnection Model*
slow	1	125	1.62	balance_case_tree
typical	1	25	1.8	balance_case_tree
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

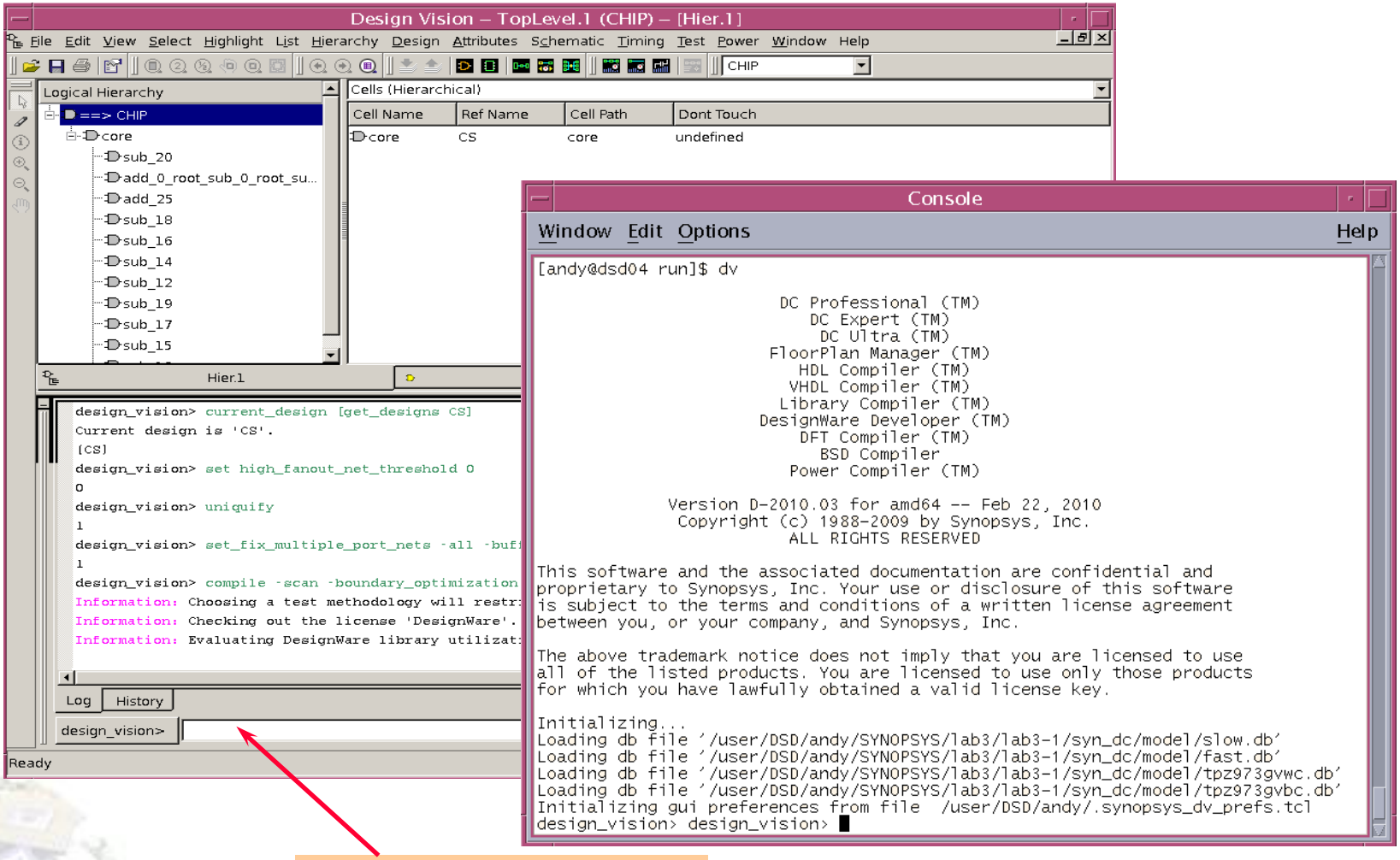
.synopsys_dc.setup **TCL 語法**

```
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 verilogout_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

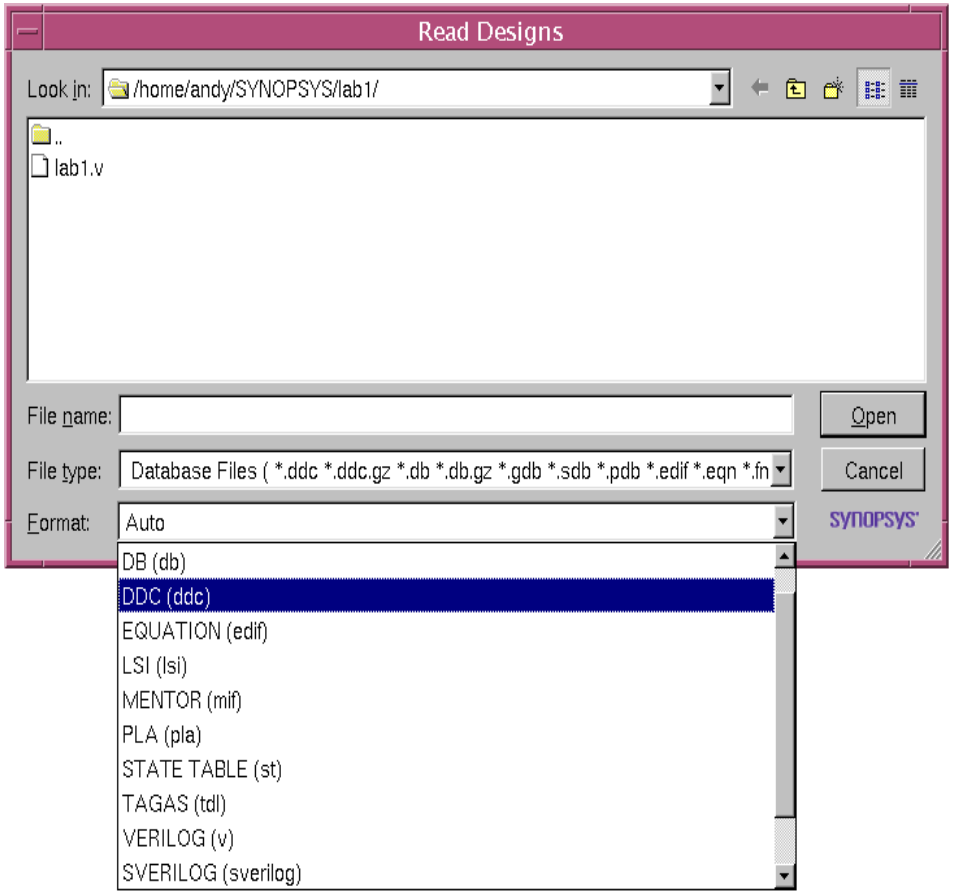
Invoke Design Vision

- *Unix%> dv* (Only command mode: dc_shell)



Method 1:Read File

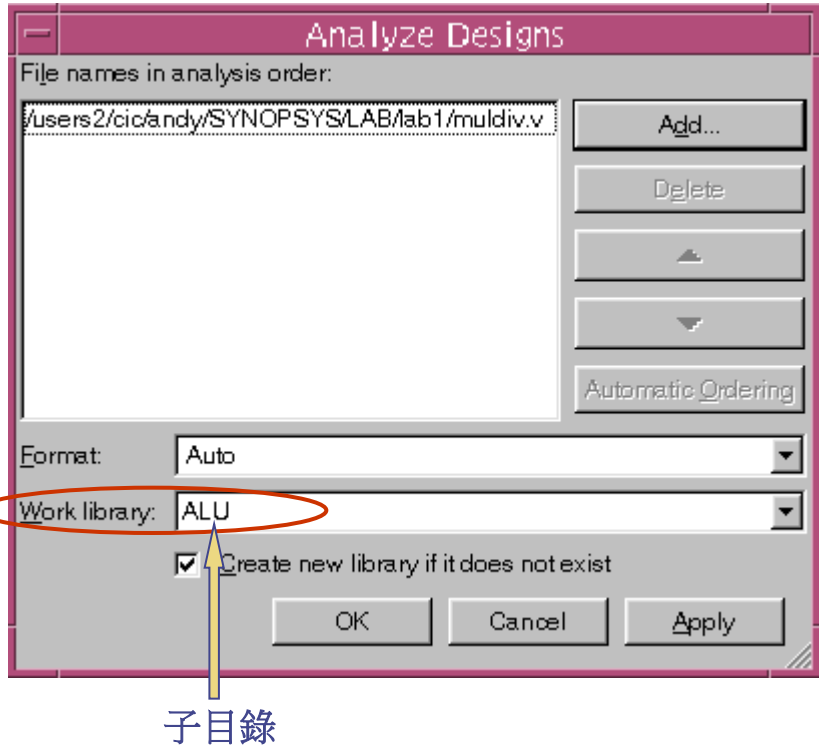
- Read netlists or other design descriptions into Design Compiler
- *File/Read*
- Support many different formats
  - Verilog: .v
  - VHDL: .vhd
  - System Verilog: .sv
  - EDIF
  - PLA(Berkeley Espresso): .pla
  - Synopsys internal formats:
    - DB(binary): .db
    - Enhance db file: .ddc
    - equation: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

(File/Analyze)



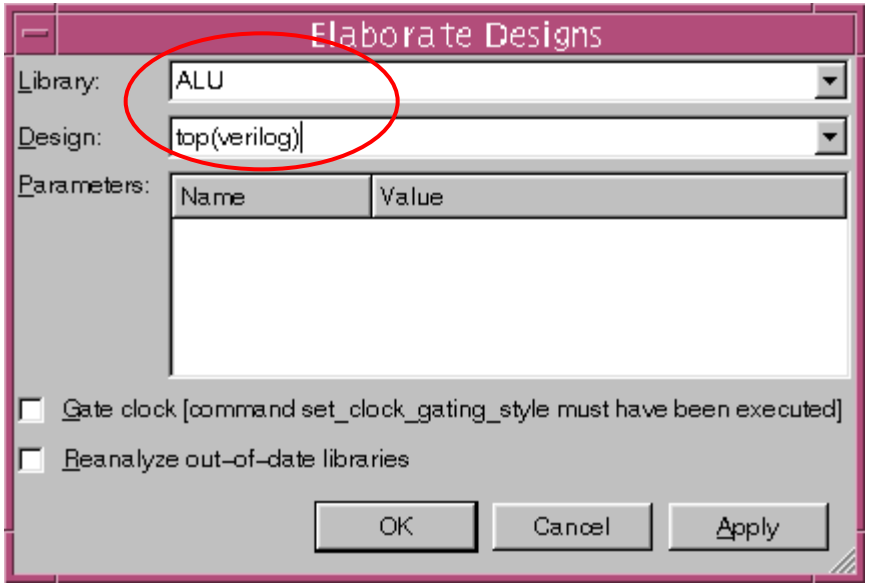
- Equivalent to design_vision command

```
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
```

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

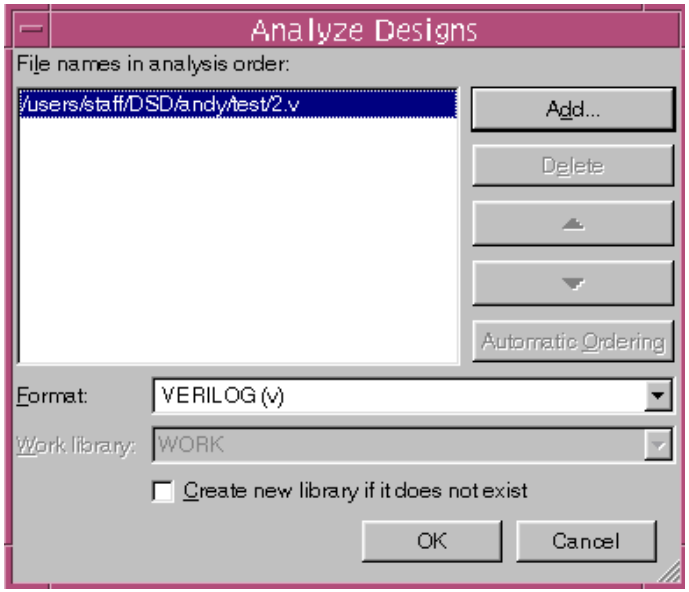
```
Warning: Design 'top' has '2' unresolved references. For more detailed information, use the "link" command. UID-341
Warning: In design 'top', there are 16 multiple-driver nets with unknown wired-logic type. LINT-30
Warning: Cell 'U1' (mul_param_1) not translated. TRANS-1
Warning: Cell 'U2' (mul_param_2) 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

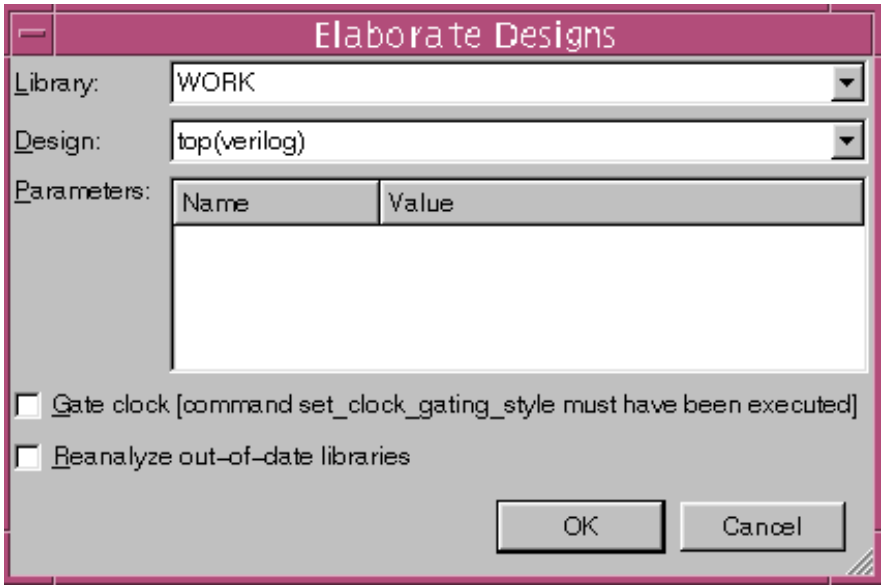
design_vision-t>

Example: Analyze/Elaborate (2/2)

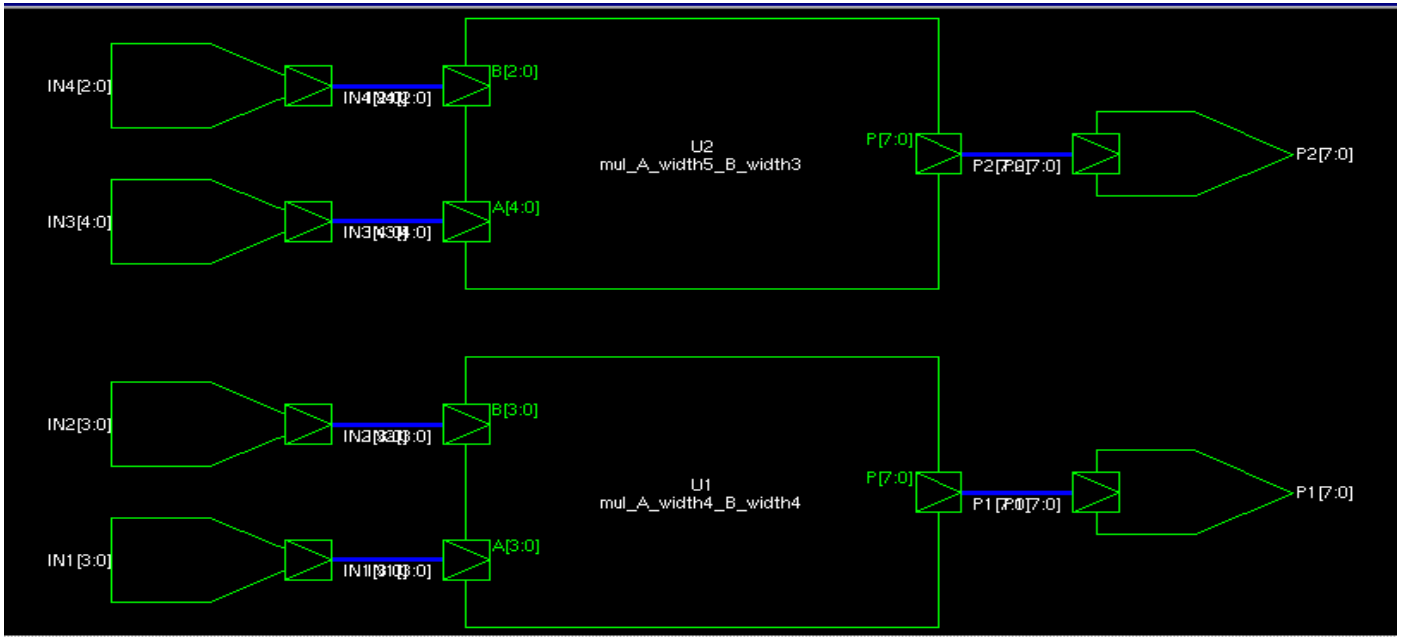
1



2

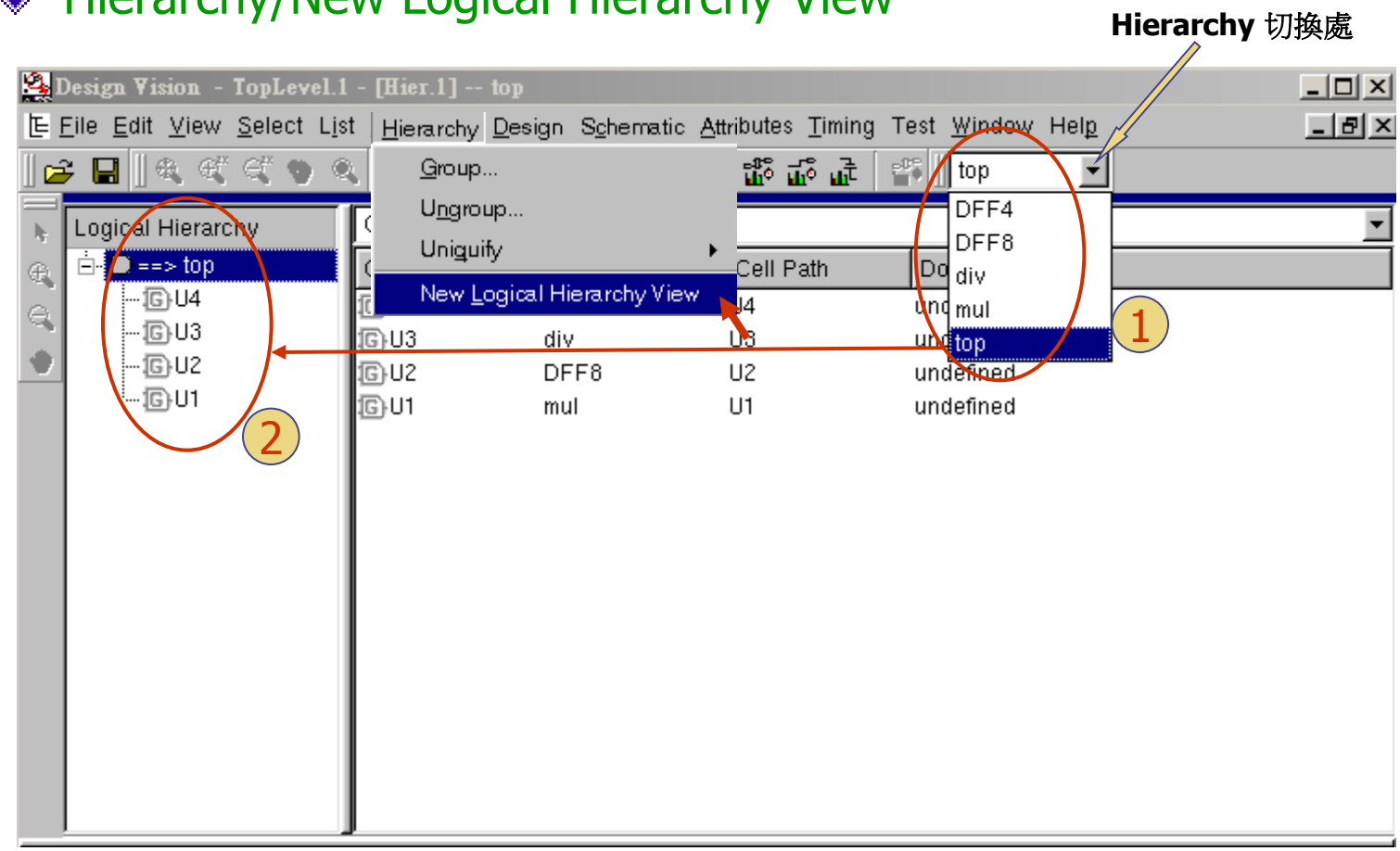


3

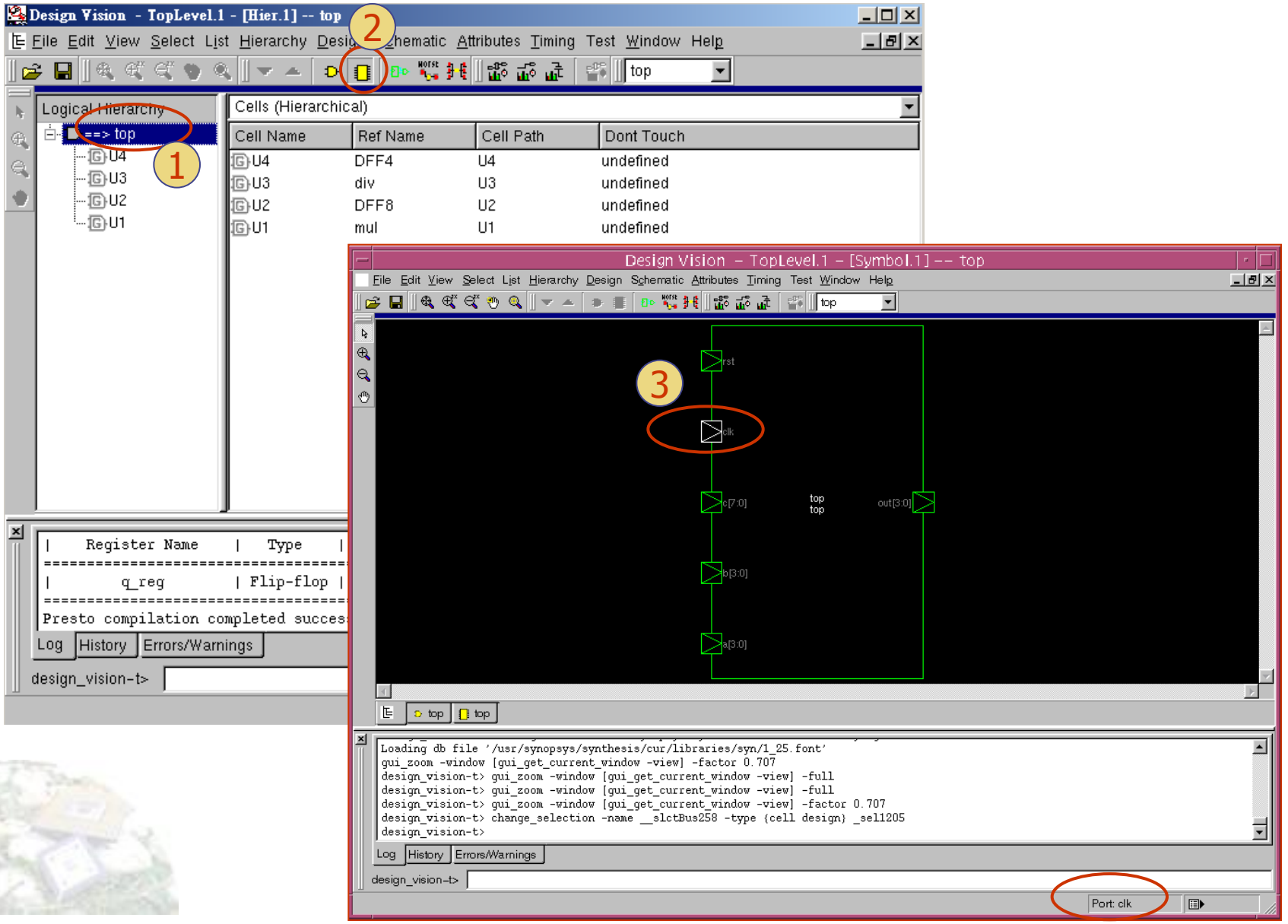


Hierarchy View

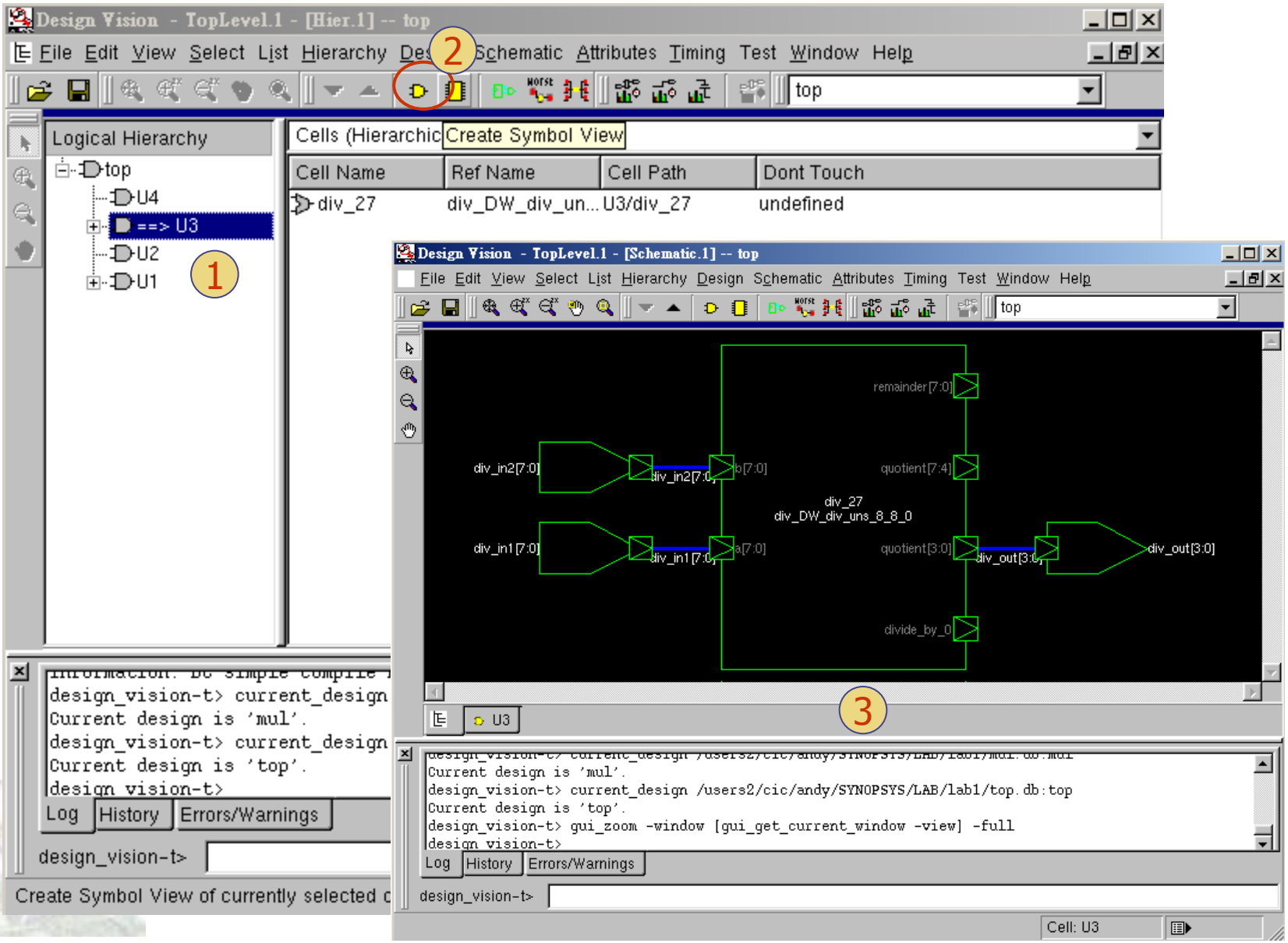
- Hierarchy/New Logical Hierarchy View



Symbol View


Design Vision - TopLevel.1 - [Hier.1] -- top
File Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help
top
Logical Hierarchy
==> top
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
1
2
Register Name
Type
q_reg
Flip-flop
Presto compilation completed succes
Log
History
Errors/Warnings
design_vision-t>
Design Vision - TopLevel.1 - [Symbol.1] -- top
File Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help
top
3
rst
clk
c[7:0]
top
top
out[3:0]
b[3:0]
a[3:0]
top
top
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 __sltBus258 -type (cell design) _sel1205
design_vision-t>
Log
History
Errors/Warnings
design_vision-t>
Port: clk

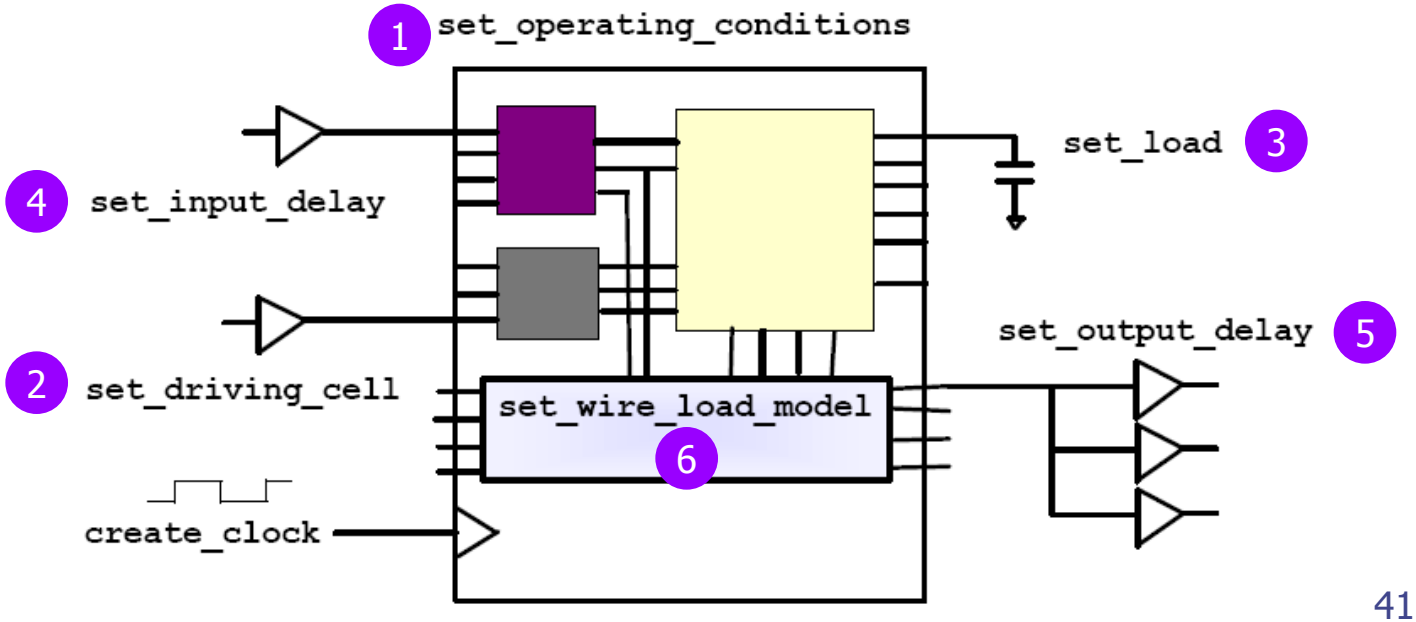
Schematic View



5. Setting Design Environment & Constraints

Setting Design Environment

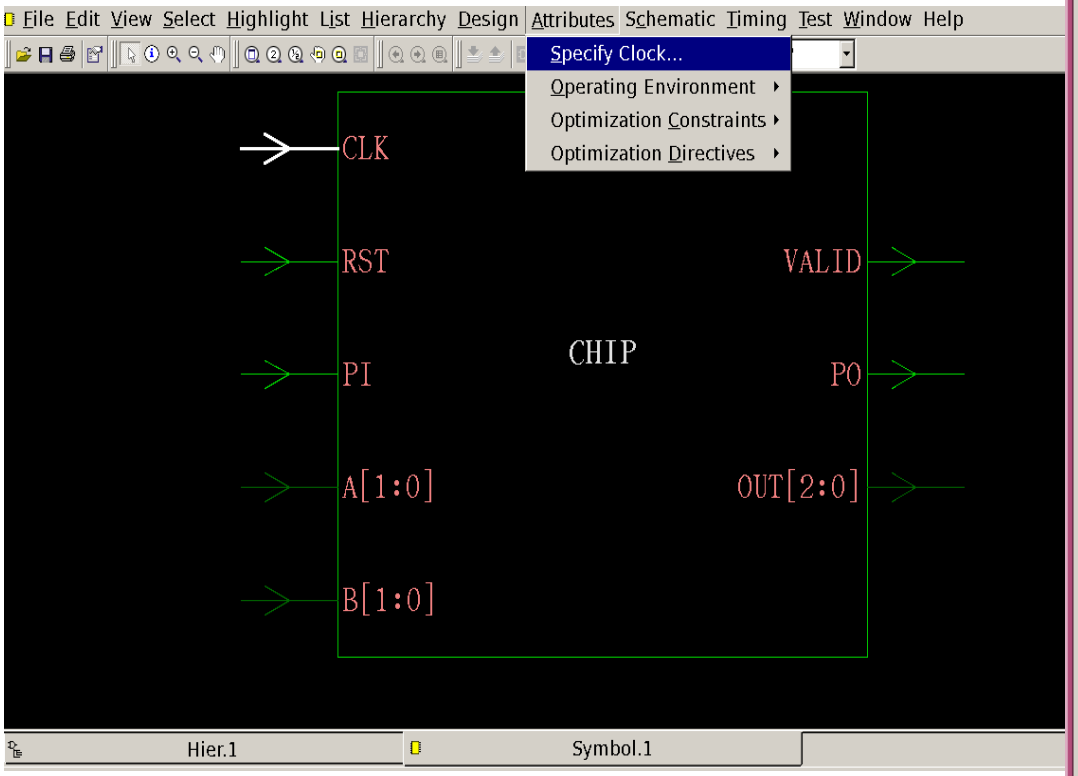
- Beware the defaults are not realistic conditions
  - Input drive is not infinite
  - Capacitive loading is usually not zero
  - Consider process, voltage, temperature (**PVT**) variation
- The operating environment affects the components selected from target library and timing through your design
-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

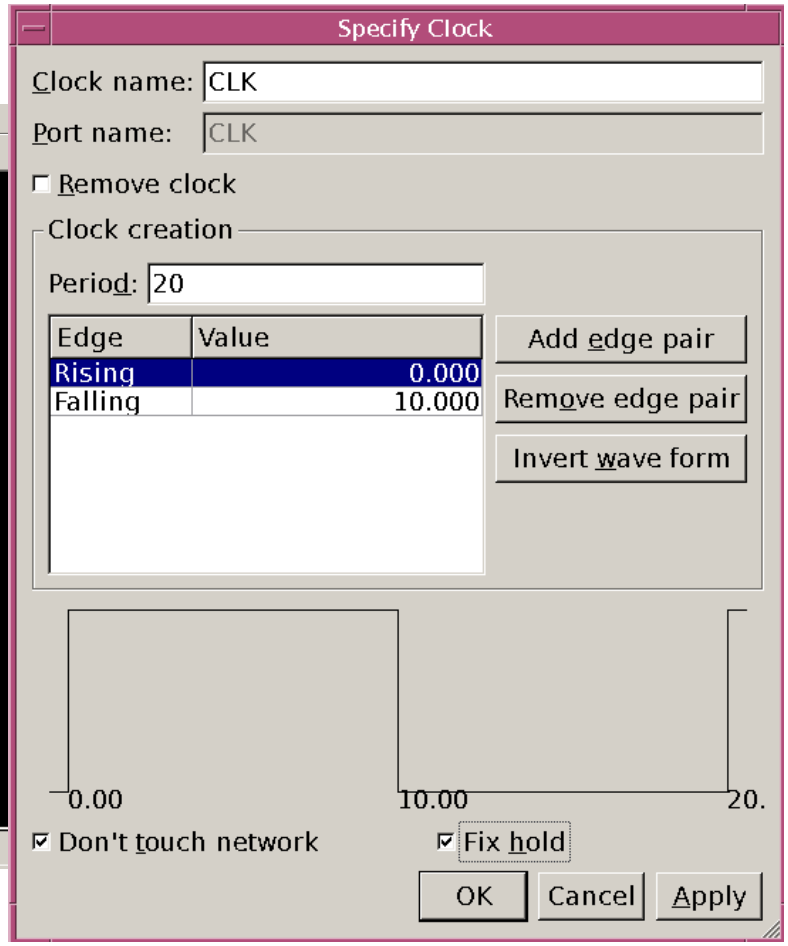


- Design Constraints
  - Timing 
  - Area
  - Power

Sequential Circuit ➔ Specify Clock

- Select clock port
- *Attributes/Specify Clock*





Command: create_clock –period 20 [get_ports clk]

Define Clock Specification

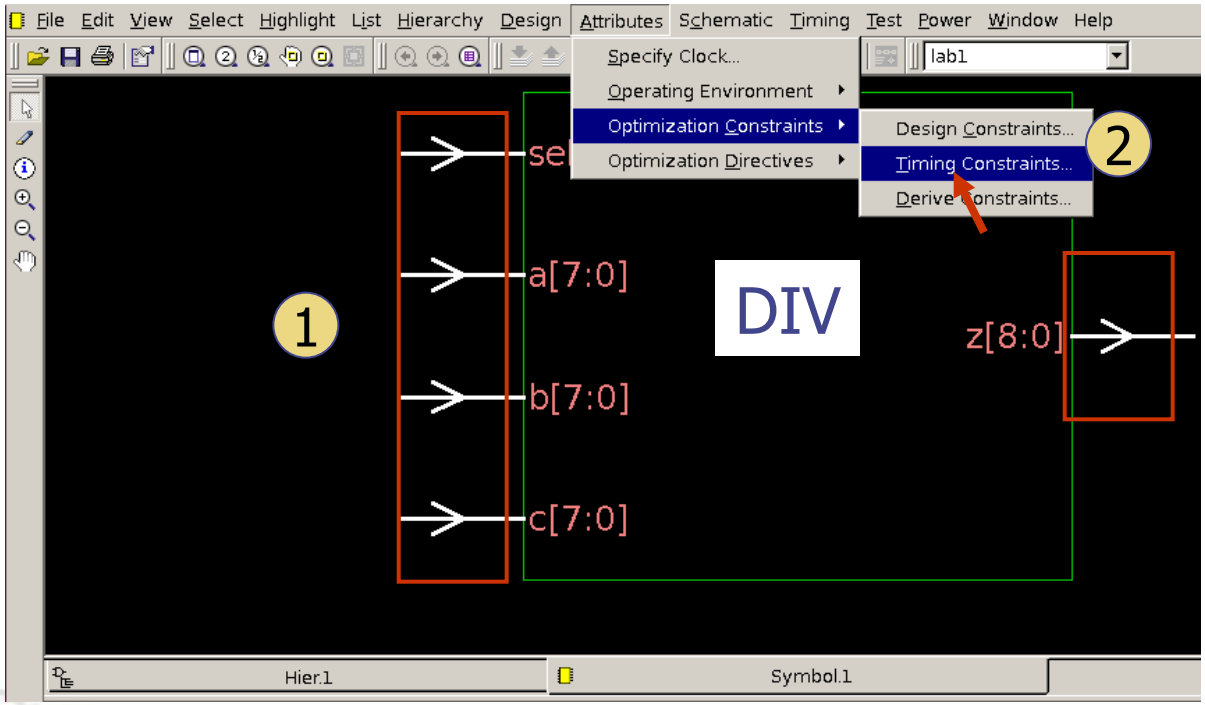
- We need to accurately specify the clock including the clock routing details in the early design stage in order to achieve timing convergence

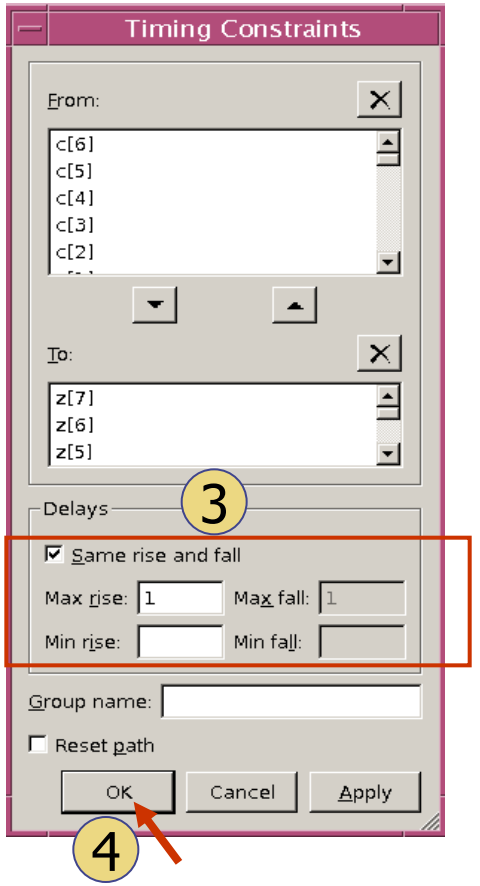
- What should be defined?
  - Period
  - Waveform
  - Uncertainty
    - Skew
  - Latency
    - Source latency (option)
    - Network latency
  - Transition
    - Input transition
    - Clock transition

- 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*





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?

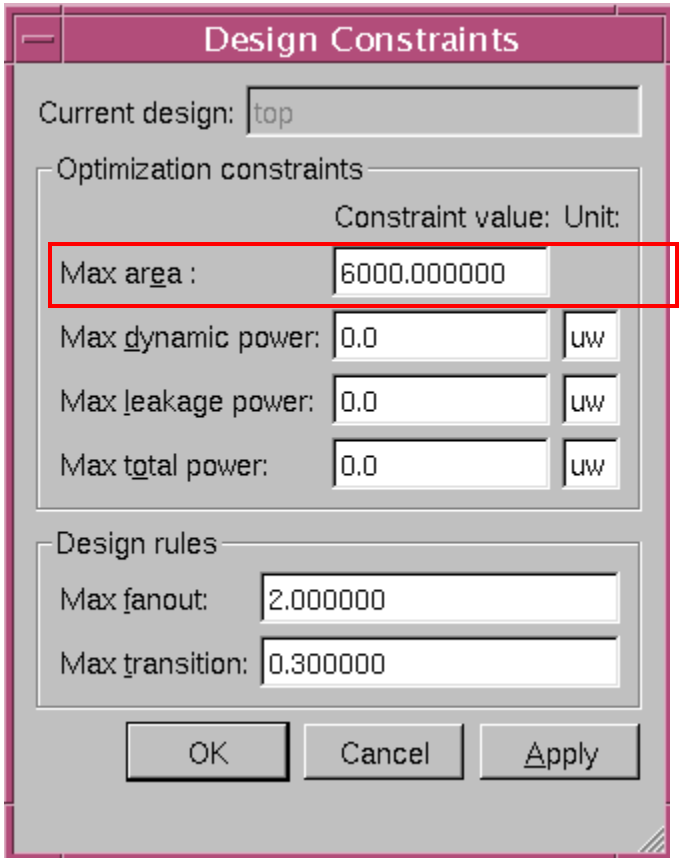
- Design Constraints
  - Timing
  - Area
  - Power

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
```



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
0:02:00	10066.2	0.00	0.0	0.0
0:02:00	10029.6	0.00	0.0	0.0
0:02:00	10019.6	0.00	0.0	0.0
0:02:00	10006.3	0.00	0.0	0.0
0:02:00	9973.0	0.00	0.0	0.0
0:02:00	9949.8	0.00	0.0	0.0
0:02:00	9923.1	0.00	0.0	0.0
0:02:01	9869.9	0.00	0.0	0.0
0:02:01	9859.9	0.00	0.0	0.0
0:02:01	9820.0	0.00	0.0	0.0
0:02:01	9820.0	0.00	0.0	0.0
0:02:01	9806.7	0.00	0.0	0.0
0:02:01	9793.4	0.00	0.0	0.0
0:02:01	9780.1	0.00	0.0	0.0
0:02:01	9753.5	0.00	0.0	0.0
0:02:01	9740.2	0.00	0.0	0.0
0:02:01	9710.2	0.00	0.0	0.0
0:02:01	9683.6	0.00	0.0	0.0
0:02:02	9657.0	0.00	0.0	0.0
0:02:02	9630.4	0.00	0.0	0.0
0:02:02	9603.8	0.00	0.0	0.0
0:02:02	9590.5	0.00	0.0	0.0
0:02:02	9563.8	0.00	0.0	0.0
0:03:08	7784.2	0.00	0.0	0.0
0:03:22	7770.9	0.00	0.0	0.0
0:03:22	7770.9	0.00	0.0	0.0
0:03:23	7764.2	0.00	0.0	0.0
0:03:25	7757.5	0.00	0.0	0.0
0:03:25	7757.5	0.00	0.0	0.0
0:03:27	7747.6	0.00	0.0	0.0
0:03:30	7724.3	0.00	0.0	0.0
0:03:31	7714.3	0.00	0.0	0.0
0:03:31	7711.0	0.00	0.0	0.0
0:03:31	7694.3	0.00	0.0	0.0
0:03:32	7691.0	0.00	0.0	0.0
0:03:32	7684.4	0.00	0.0	0.0
0:03:35	7667.7	0.00	0.0	0.0
0:03:44	7664.4	0.00	0.0	0.0
0:03:44	7651.1	0.00	0.0	0.0
0:03:48	7644.4	0.00	0.0	0.0
0:03:51	7627.8	0.00	0.0	0.0
0:03:56	7621.1	0.00	0.0	0.0
0:04:00	7611.2	0.00	0.0	0.0
0:04:00	7604.5	0.00	0.0	0.0
0:04:00	7601.2	0.00	0.0	0.0
0:04:02	7597.8	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

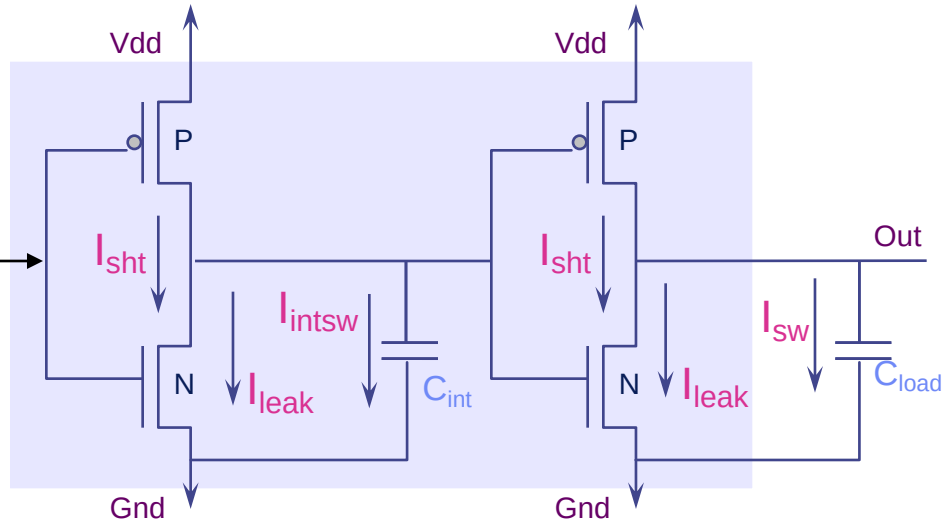
Optimization Complete

Area

- Design Constraints
  - Timing
  - Area
  - Power

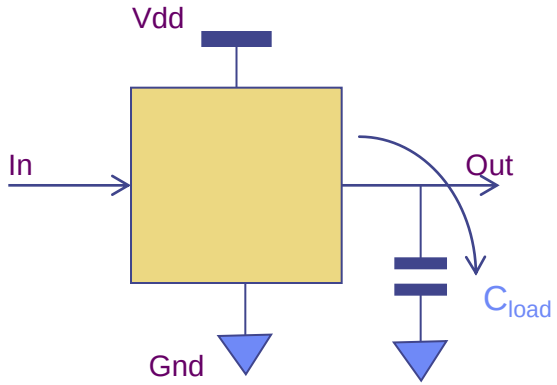
Power Model

- 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%)
                         ---------
Total Dynamic Power    = 553.5076 uW  (100%)

Cell Leakage Power     =   2.6536 uW
```

```
Beginning Dynamic Power Optimization  (max_total_power 0 uw)
-----------------------------------

 ELAPSED            WORST NEG TOTAL NEG  DESIGN                              TOTAL
  TIME      AREA      SLACK     SLACK   RULE COST         ENDPOINT           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            WORST NEG TOTAL NEG  DESIGN                              TOTAL
  TIME      AREA      SLACK     SLACK   RULE COST         ENDPOINT           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

w/ Power Optimization:

```
  Cell Internal Power  = 225.5084 uW   (43%)
  Net Switching Power  = 293.5588 uW   (57%)
                         ---------
Total Dynamic Power    = 519.0672 uW  (100%)

Cell Leakage Power     =   2.4994 uW
```

Power Improve:
(553.5 - 519) / 553.5 = 6.23%

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)

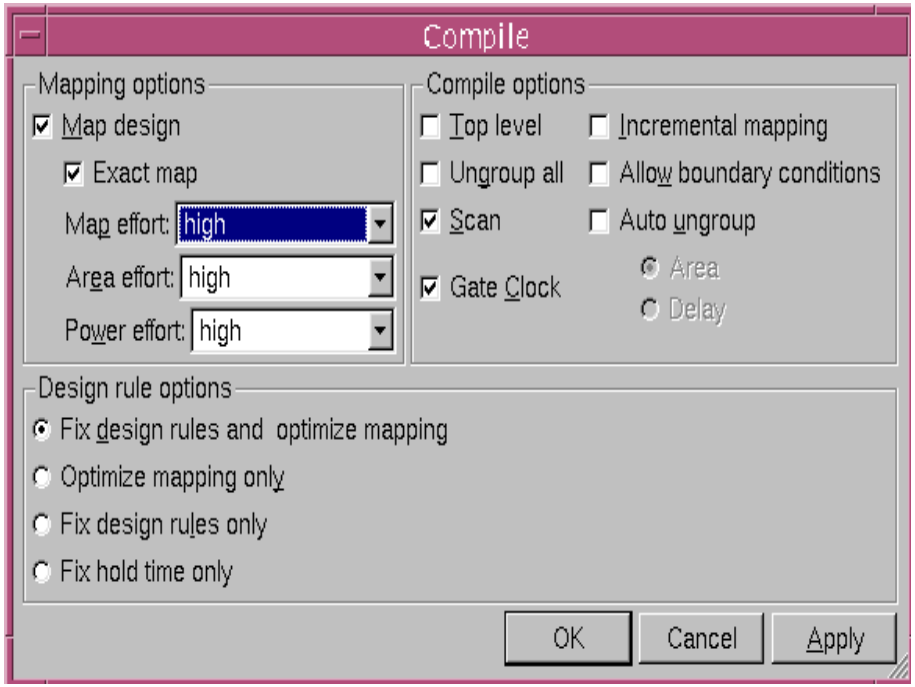
5. Leakage power optimization:
   - EDA : Leakage Power optimization (20 - 80%)
     (90nm UMC/Faraday Support Multi-V_T Cell)

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

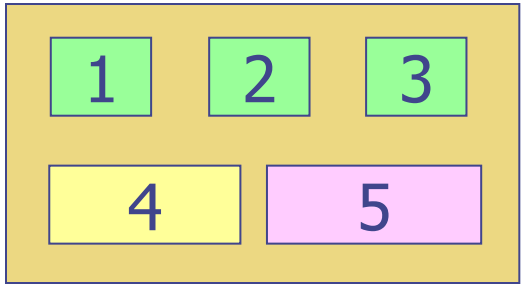
2010.03-sp5



Ultra high:

compile_ultra

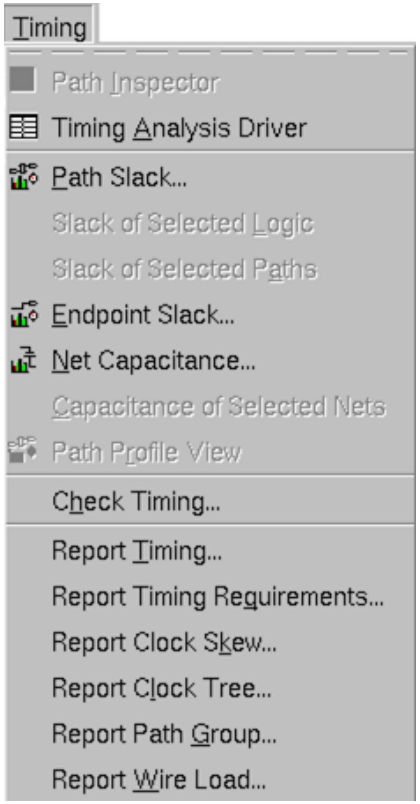
ex:



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

start point

Net_delay + cell_delay are combined

end point

```
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
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 (BUFX10TH)                      0.13       2.24 f
microwave/micro_st/U5/Y (CLKNAND2X8TH)                  0.05       2.29 r
microwave/micro_st/U20/Y (CLKINVX4TH)                   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 (BUFX18TH)                                  0.11       2.88 f
timer/U50/Y (CLKINVX6TH)                                0.08       2.96 r
timer/U43/Y (CLKINVX24TH)                               0.13       3.09 f
timer/U23/Y (OAI22X1TH)                                 0.29       3.38 r
timer/U40/Y (BUFX6TH)                                   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 (CLKINVX12TH)                               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
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
```

Report Area

- Area report shows the um^2 of the design
- Design / Report Area

report_area -hier

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

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
------------------------------

                                 Global cell area          Local cell area
                                 ------------------  ----------------------------
Hierarchical cell                Absolute   Percent  Combi-     Noncombi-  Black
                                 Total      Total    national   national   boxes   Design
-------------------------------- ---------  -------  ---------  ---------  ------  -----------
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/convml                     82.5552      5.0    82.5552     0.0000  0.0000  converter_2
display/convmm                     82.5552      5.0    82.5552     0.0000  0.0000  converter_3
display/convsl                     82.5552      5.0    82.5552     0.0000  0.0000  converter_0
display/convsm                     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
```

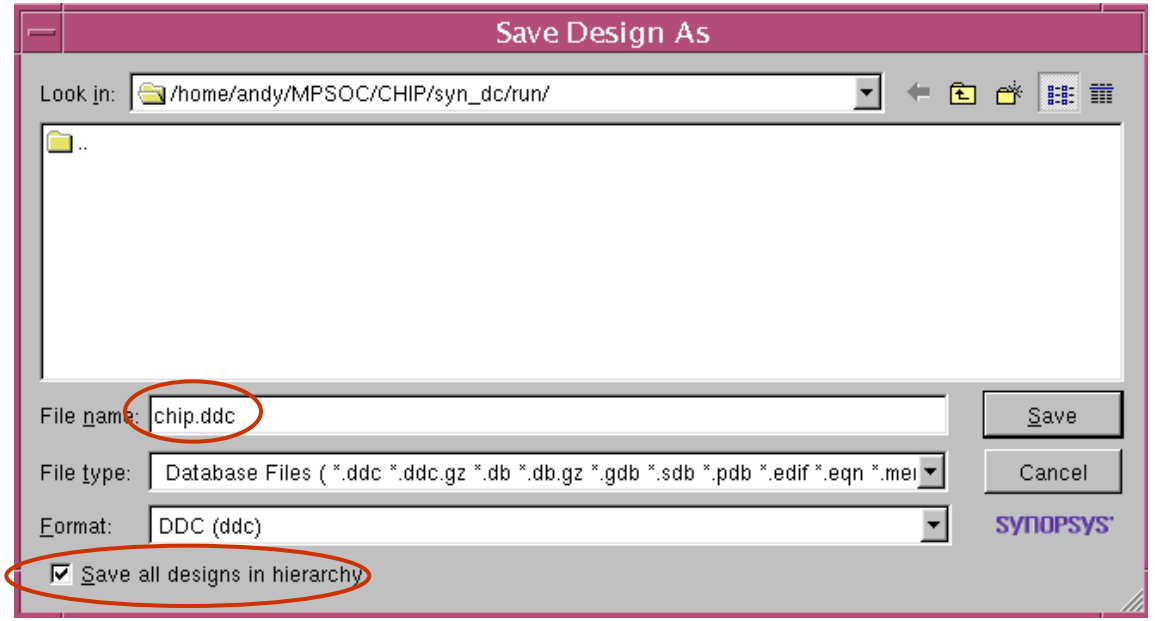
2. report_power -hier

```
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


--------------------------------------------------------------------------------
                                   Switch   Int      Leak     Total
Hierarchy                          Power    Power    Power    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]}

Bus[5]

- 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

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***

  1.0 or 2.1

- Modifiy your testfixture file

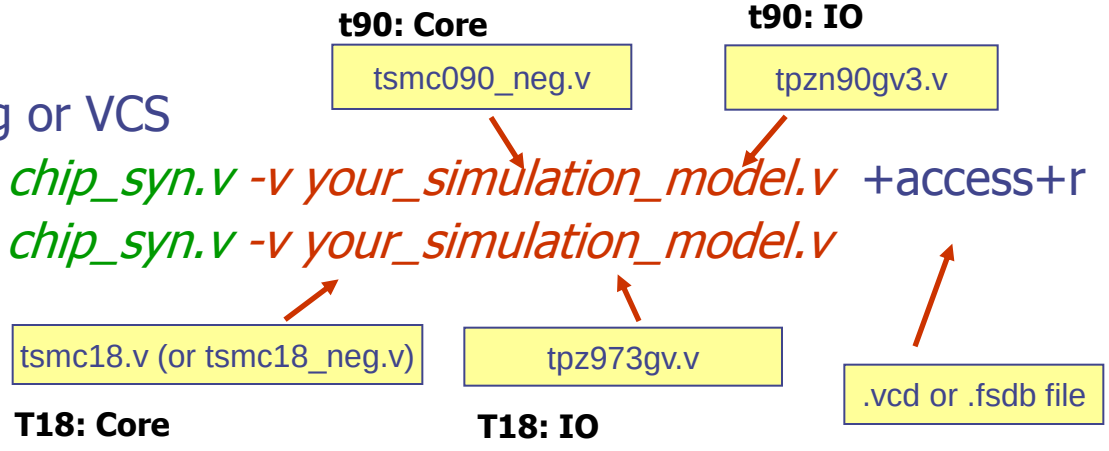
  $sdf_annotate("the_SDF_file_name",
  the_top_level_module _instance_name);

  For example: $sdf_annotate("chip_syn.sdf", chip);

- 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% ncverilog 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 &**