Training Course of Design Compiler

- T. –W. Tseng, “ARES Lab 2008 Summer Training Course of Design Compiler”

REF:

- CIC Training Manual – Logic Synthesis with Design Compiler, July, 2006
- TSMC 0.18um Process 1.8-Volt SAGE-X™ Stand Cell Library Databook, September, 2003
- TPZ973G TSMC 0.18um Standard I/O Library Databook, Version 240a, December 10, 2003
- Artisan User Manual

Speaker: T. –J. Chen

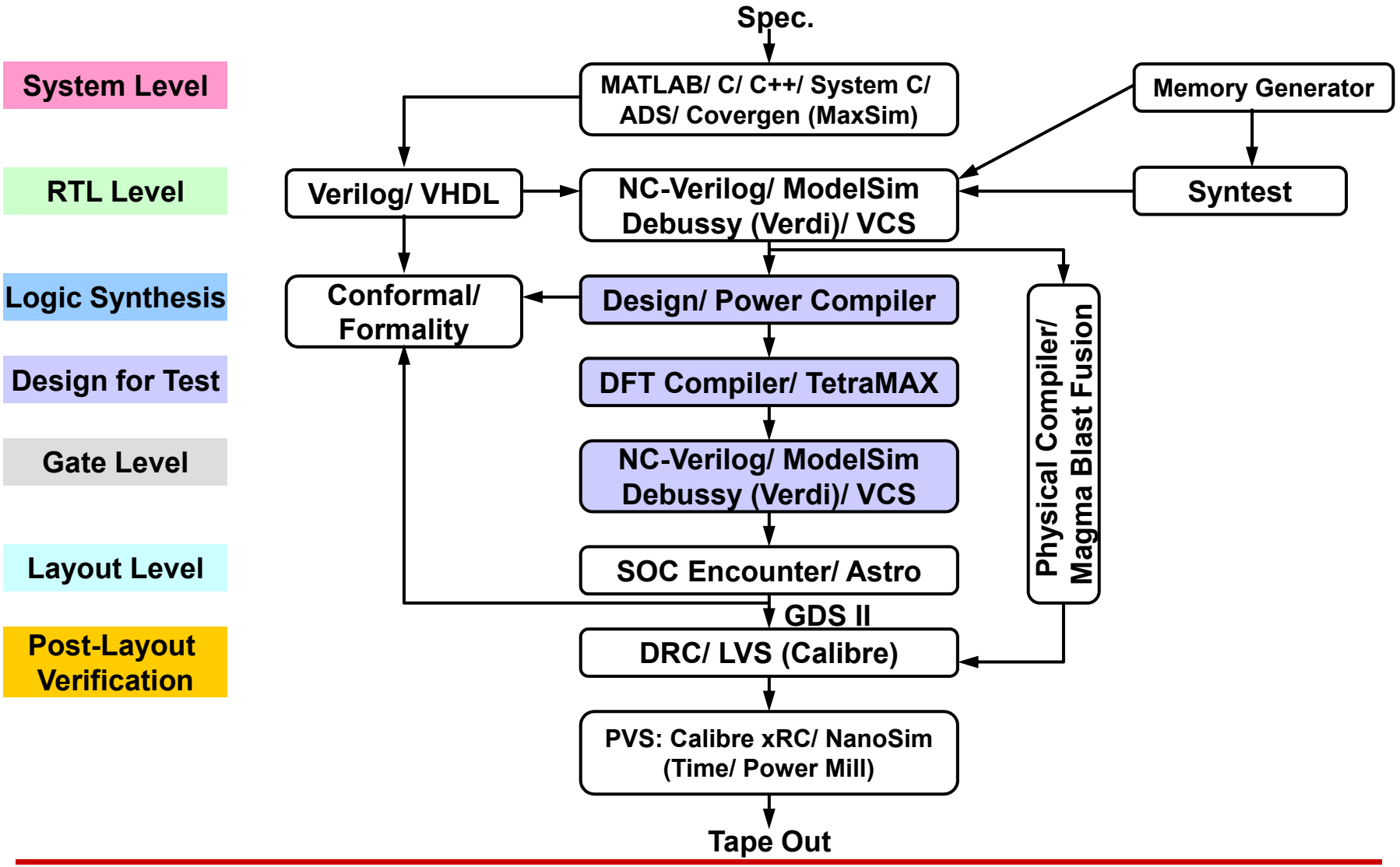
ADVANCED RELIABLE SYSTEMS (ARES) LAB.

Outline

- Basic Concept of the Synthesis
- Synthesis Using Design Compiler

Basic Concept of the Synthesis

Cell-Based Design Flow



What is Synthesis

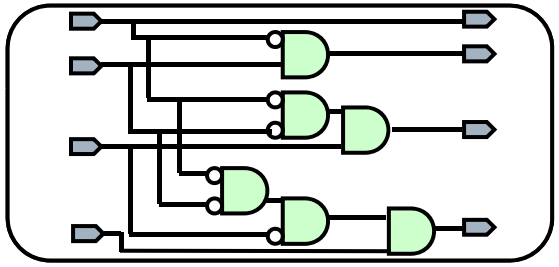
- Synthesis = translation + optimization + mapping

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

HDL Source (RTL)

Translate (HDL Compiler)

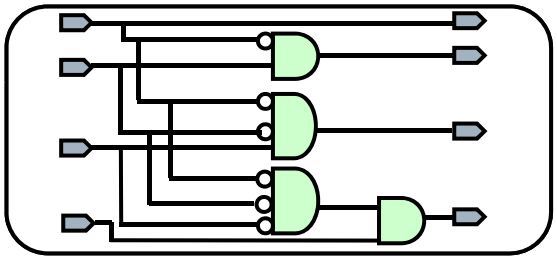
No Timing Info.



Generic Boolean (GTECT)

Optimize + Mapping (Design Compiler)

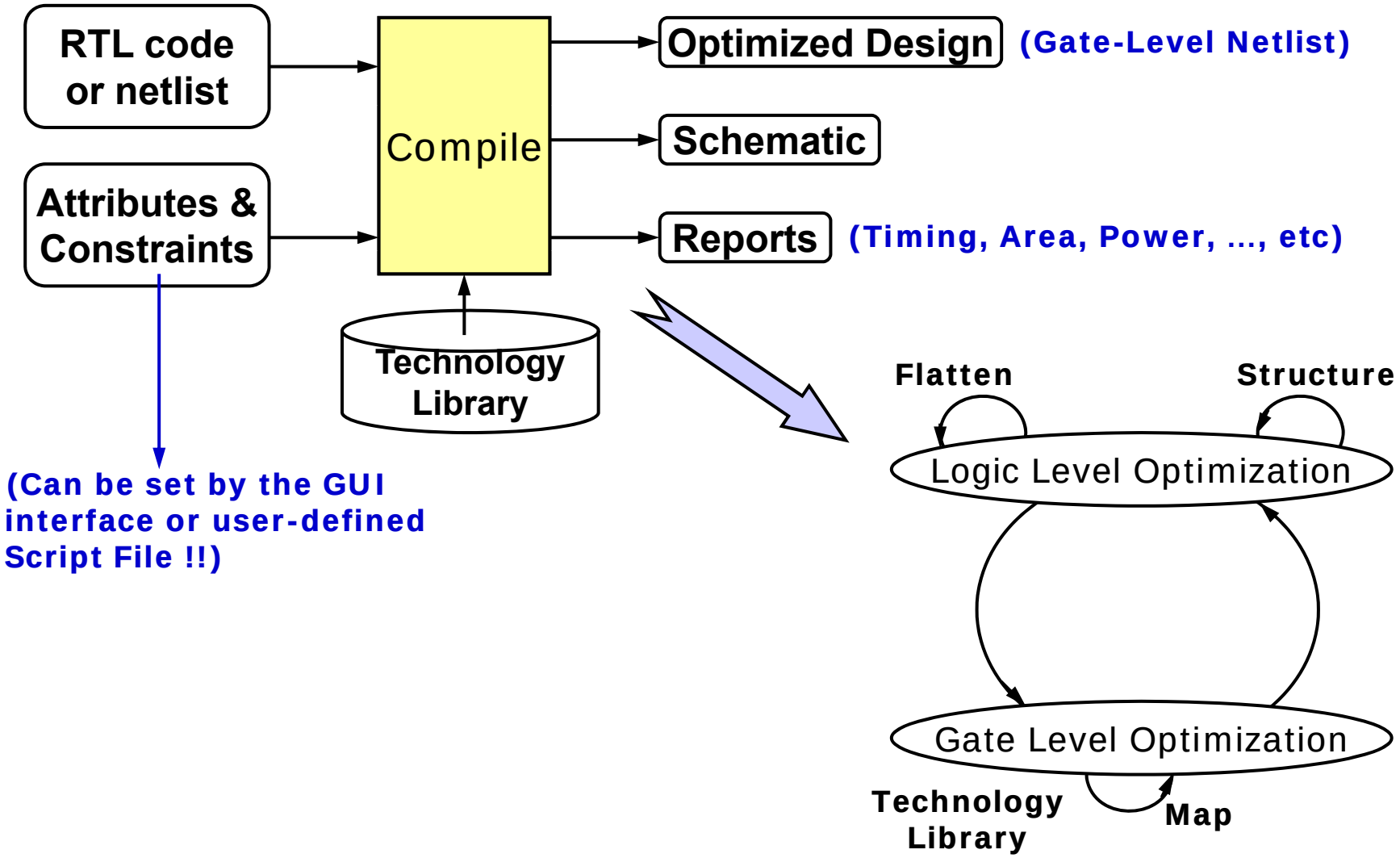
Timing Info.



Target Technology

The synthesis is constraint driven and technology independent !!

Compile


RTL code or netlist
Attributes & Constraints
Compile
Optimized Design
(Gate-Level Netlist)
Schematic
Reports
(Timing, Area, Power, ..., etc)
Technology Library
(Can be set by the GUI interface or user-defined Script File !!)
Flatten
Structure
Logic Level Optimization
Gate Level Optimization
Technology Library
Map

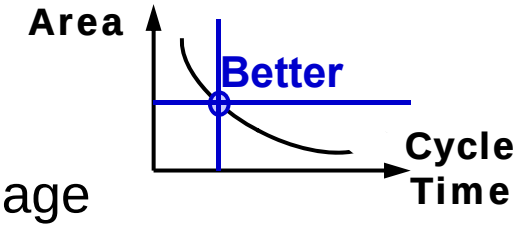
Synthesizable Verilog

- Verilog Basis
  - parameter declarations
  - wire, wand, wor declarations
  - reg declarations
  - input, output, inout declarations
  - continuous assignments
  - module instructions
  - gate instructions
  - always blocks
  - task statements
  - function definitions
  - for, while loop
- Synthesizable Verilog primitives cells
  - and, or, not, nand, nor, xor, xnor
  - bufif0, bufif1, notif0, notif1

Synthesizable Verilog (Cont')

- Operators
  - Binary bit-wise (~, &, |, ^, ~^)
  - Unary reduction (&, ~&, |, ~|, ^, ~^)
  - Logical (!, &&, ||)
  - 2's complement arithmetic (+, -, *, /, %)
  - Relational (>, <, >=, <=)
  - Equality (==, !=)
  - Logic shift (>>, <<)
  - Conditional (?:)
  - Concatenation ({ })

Notice Before Synthesis



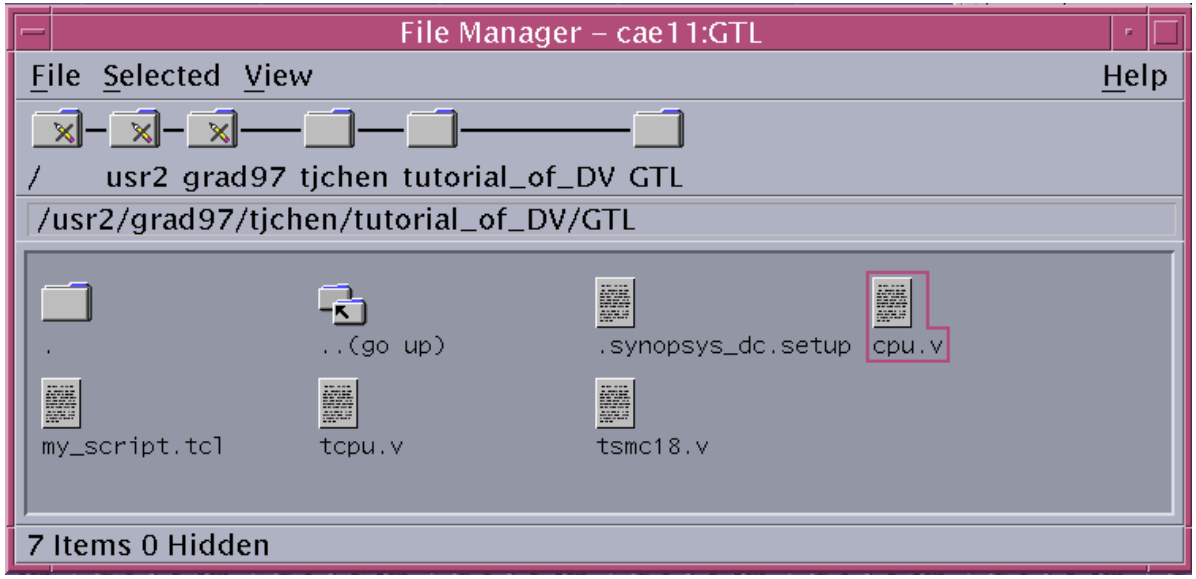
- Your RTL design
  - Functional verification by some high-level language
    - Also, the code coverage of your test benches should be verified (i.e. VN)
  - Coding style checking (i.e. n-Lint)
    - Good coding style will reduce most hazards while synthesis
    - Better optimization process results in better circuit performance
    - Easy debugging after synthesis
- Constraints
  - The area and timing of your circuit are mainly determined by your circuit architecture and coding style
  - There is always a trade-off between the circuit timing and area
  - In fact, a super tight timing constraint may be worked while synthesis, but failed in the Place & Route (P&R) procedure

Synthesis Using Design Compiler

Related Files

Folder	Name	Description
GTL	.synopsys_dc.setup	Design compiler setup file
	my_script.tcl	Synthesis script file
	my_design.v	Verilog files
	tmy_design.v	Test bench
	tsmc18.v	Verilog model of standard cells

Ex:



<.synopsys_dc.setup> File

- **link_library** : the library used for interpreting input description
  - Any cells instantiated in your HDL code
  - Wire load or operating condition modules used during synthesis
- **target_library** : the ASIC technology which the design is mapped
- **symbol_library** : used for schematic generation
- **search_path** : the path for unsolved reference library
- **synthetic_path** : designware library

<.synopsys_dc.setup> File (Cont')

- MEMs libraries are also included in this file

Ex:

Text Editor – .synopsys_dc.setup

File Edit Format Options Help

```
########0.18um
set search_path         "/APP/cell_lib/CBDK018_TSMC_Artisan/CIC/SynopsysDC/db $search_path"
set search_path         "/usr/cad/synopsys/synthesis/cur/libraries/syn  $search_path"
set search_path         "/APP/cell_lib/CBDK018_TSMC_Artisan/orig_lib/aci/sc/symbols/synopsys $search_path"
set search_path         "/usr2/grad97/tjchen/tutorial_of_DV/MEM $search_path"
######with MEM#####
set link_library        "RAM_64B_fast@-40C_syn.db RAM_64B_fast@0C_syn.db RAM_64B_slow_syn.db RAM_64B_typical_syn.db typical.
set target_library      "RAM_64B_fast@-40C_syn.db RAM_64B_fast@0C_syn.db RAM_64B_slow_syn.db RAM_64B_typical_syn.db typical.
set symbol_library      "tsmc18.sdb"

set hdlin_translate_off_skip_text "TRUE"
set edifout_netlist_only "TRUE"
set verilogout_no_tri true
set plot_command {lpr -Plp}
```

Add a "search path"

MEM Libraries (.db file)

(.synopsys_dc.setup File)

Note that the MEM DB files are converted from the LIB files which are generated from the Artisan !!

Settings for Using Memory

- Convert *.lib to *.db
  - *%> dc_shell –t*
  - *dc_shell-t> read_lib t13spsram512x32_slow_syn.lib*
  - *dc_shell-t> write_lib t13spsram512x32 -output \ t13spsram512x32_slow_syn.db*

any memory LIB file

user library name, which should be the same as the library name in the Artisan

memory DB file

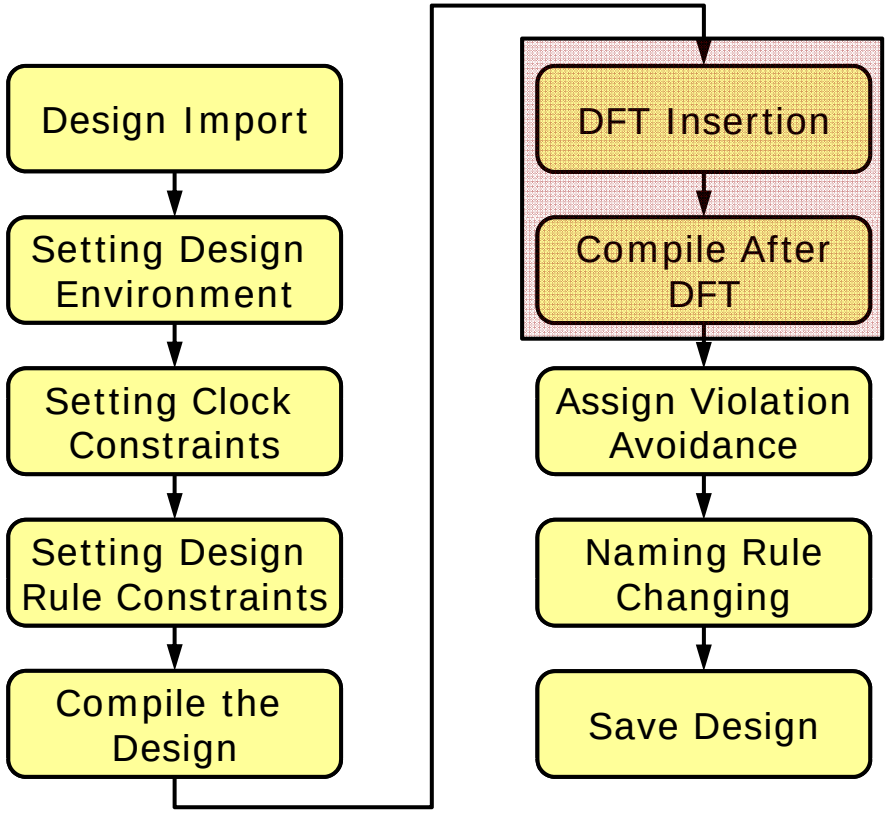
- Modify <.synopsys_dc.setup> File:
  - set link_library "* slow.db t13spsram512x32_slow.db dw_foundation.sldb"
  - set target_library "slow.db t13spsram512x32_slow.db"
  - add a "search path" to this file

add to the file

- Before the synthesis, the memory HDL model should be blocked in your netlist

```
//`include "sr_memory_1k.v"

module bisr_mem(clk,rst,ams,CSI
,bisr_mode,cmd_done,BGO,CSO,shi

parameter WORD_LENGTH = 64;
parameter ADR_LEN   = 13;
```

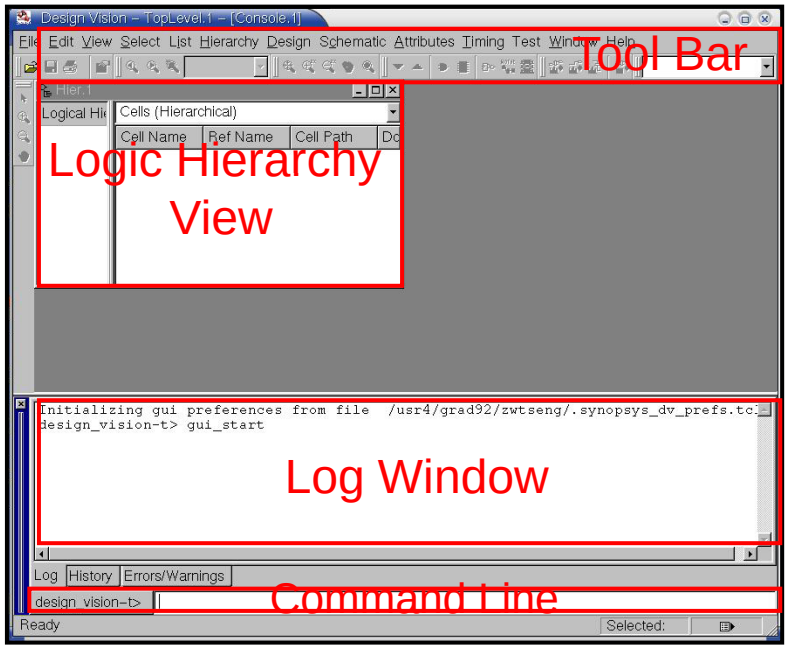
Synthesis Flow



Getting Started

- Prepare Files:
  - *.v files
  - *.db files (i.e. memory is used)
  - Synthesis script file (i.e. described later)
-

linux %> dv& (XG Mode)

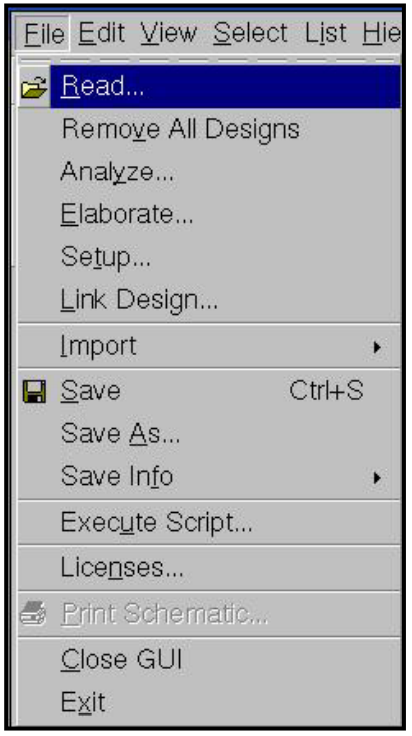


(GUI view of the Design Vision)

Read File

Design Import

- Read netlists or other design descriptions into Design Compiler
- ***File/Read***
- Supported 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

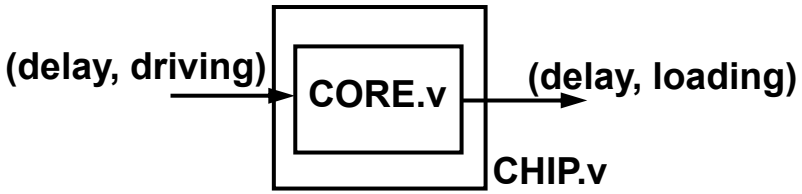


{ Command Line }

read_file –format verilog **file name**

PAD Parameters Extraction

- Input PAD
  - Input delay
  - Input driving
- Output PAD
  - Output delay
  - Output loading




```
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {ADDR_S[0]}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {bira_en}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {test_done}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {bisr_mode[1]}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {bisr_mode[0]}]
set_load -pin_load 0.06132 [get_ports {cmd_done}]
set_load -pin_load 0.06132 [get_ports {BGO}]
set_load -pin_load 0.06132 [get_ports {CSO}]
set_load -pin_load 0.06132 [get_ports {shift_en}]
set_load  pin_load 0.06132 [get_ports {unrcpaircd}]
set_load -pin_load 0.06132 [get_ports {done}]
set_load -pin_load 0.06132 [get_ports {bira_out_valid}]
set_load -pin_load 0.06132 [get_ports {addr_change}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {se}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {si}]
set_driving_cell -lib_cell PDIDGZ -library tpz973gbc -pin C -from_pin PAD \
-no_design_rule [get_ports {scantest}]
set_load -pin_load 0.06132 [get_ports {so}]
```

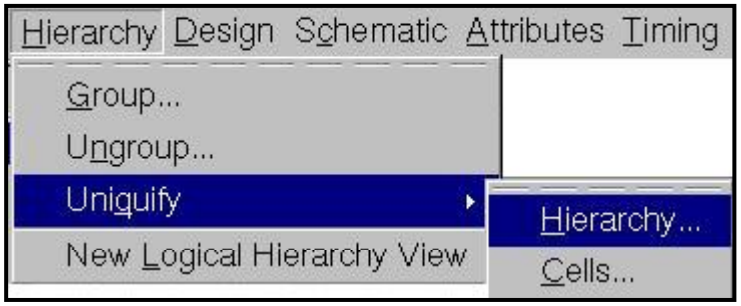
(chip_const.tcl)

{ Command Line }

```
current_design CHIP
characterize [get_cells CORE]
current_design CORE
write_script –format dctcl –o chip_const.tcl
```

Uniquify

- Select the most top design of the hierarchy
- ***Hierarchy/Uniquify/Hierarchy***



Name	Design Area	Dont Touch
MEM	0	undefined
ROM	0	undefined
SES_ID	0	undefined
SYN_DEC_8_0	0	undefined
SYN_DEC_8_1	0	undefined
SYN_DEC_8_2	0	undefined
SYN_DEC_8_3	0	undefined
SYN_DEC_8_4	0	undefined
SYN_DEC_8_5	0	undefined
SYN_DEC_8_6	0	undefined
SYN_DEC_8_7	0	undefined
addr_present	0	undefined
addr_previous1	0	undefined
addr_previous2	0	undefined
b_to_g_0	0	undefined
b_to_g_1	0	undefined

(Design View)

```
design_vision-xg-t> uniquify
Removing uniquified design 'b_to_g'.
Removing uniquified design 'SYN_DEC_8'.
  Uniquified 2 instances of design 'b_to_g'.
  Uniquified 8 instances of design 'SYN_DEC_8'.
```

(Log Window)

{ Command Line }

uniquify

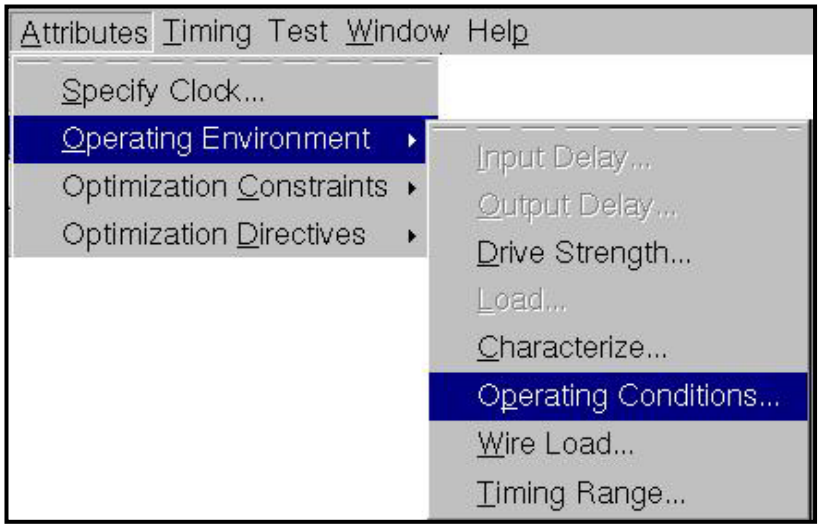
Design Environment

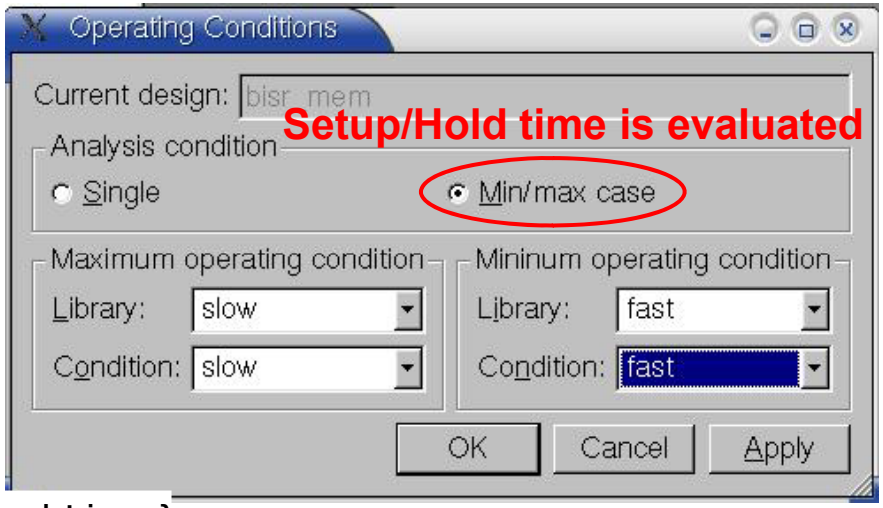
Setting Design Environment

- Setting Operating Environment
- Setting Input Driving Strength
- Setting Output Loading
- Setting Input/Output Delay
- Setting Wire Load Model

Setting Operating Condition

- ***Attributes/Operating Environment/Operating Conditions***



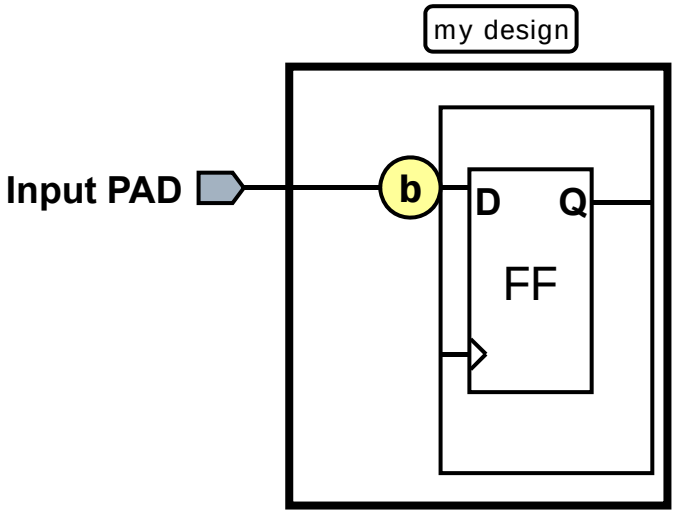


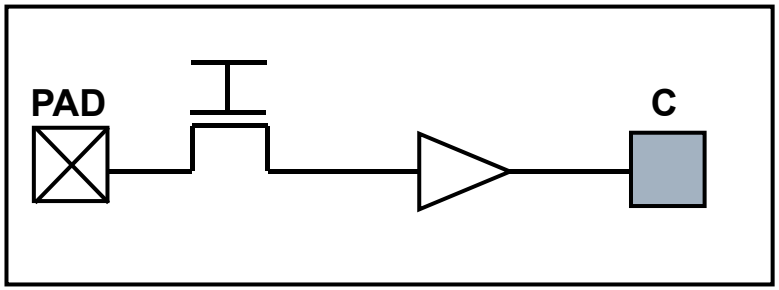
{ Command Line }

```
set_operating_conditions –max “slow” –max_library “slow” –min “fast”\
-min_library “fast”
```

Setting Drive Strength / Input Delay for PADs

- Assume that we use the input PAD “PDIDGZ”





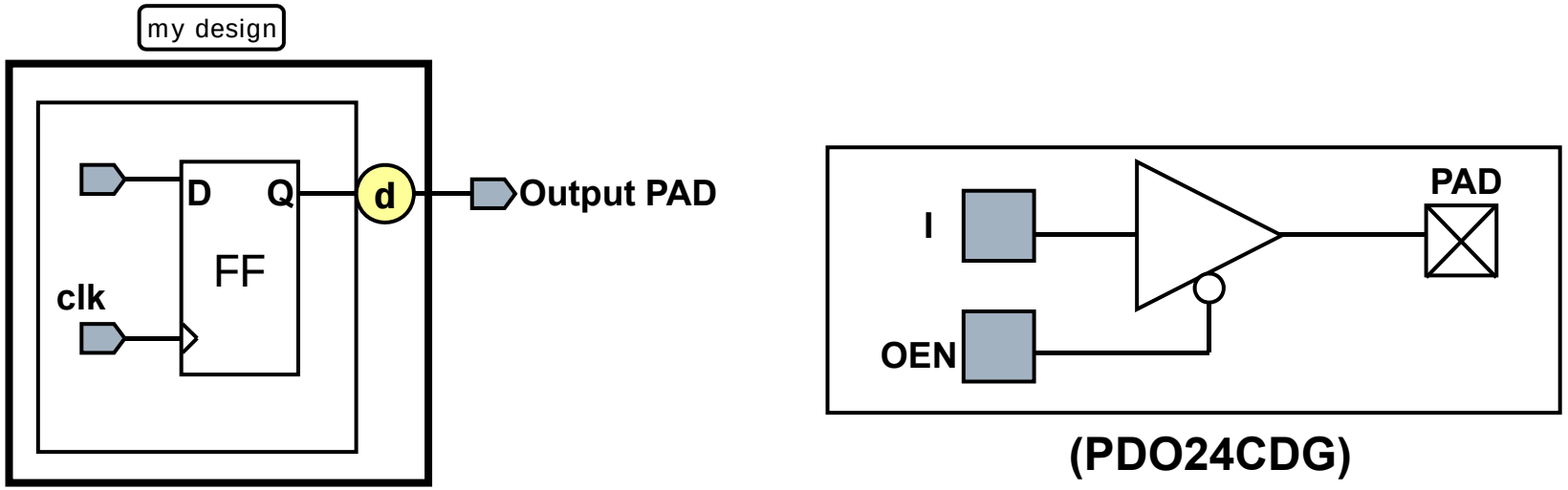
(PDIDGZ)

{ Command Line }

```
set_drive [expr 0.288001] [all_inputs]
set_input_delay [expr 0.34] –clock clk [all_inputs]
```

Setting Load/Output Delay for PADs

- Assume that we use the output PAD “PDO24CDG”

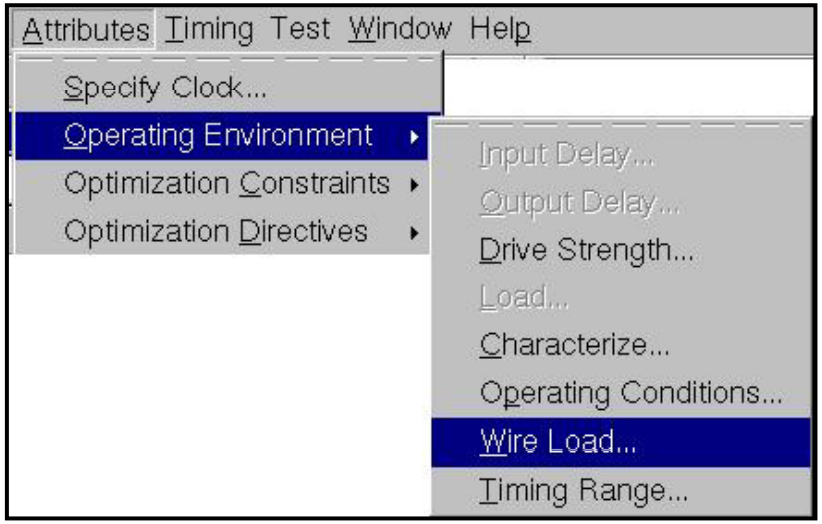


{ Command Line }

```
set_load [expr 0.06132] [all_outputs]
set_output_delay [expr 2] [all_outputs]
```

Setting Wire Load Model

- ***Attributes/Operating Environment/Wire Load***



(Worst Case)

Recommend

{ Command Line }

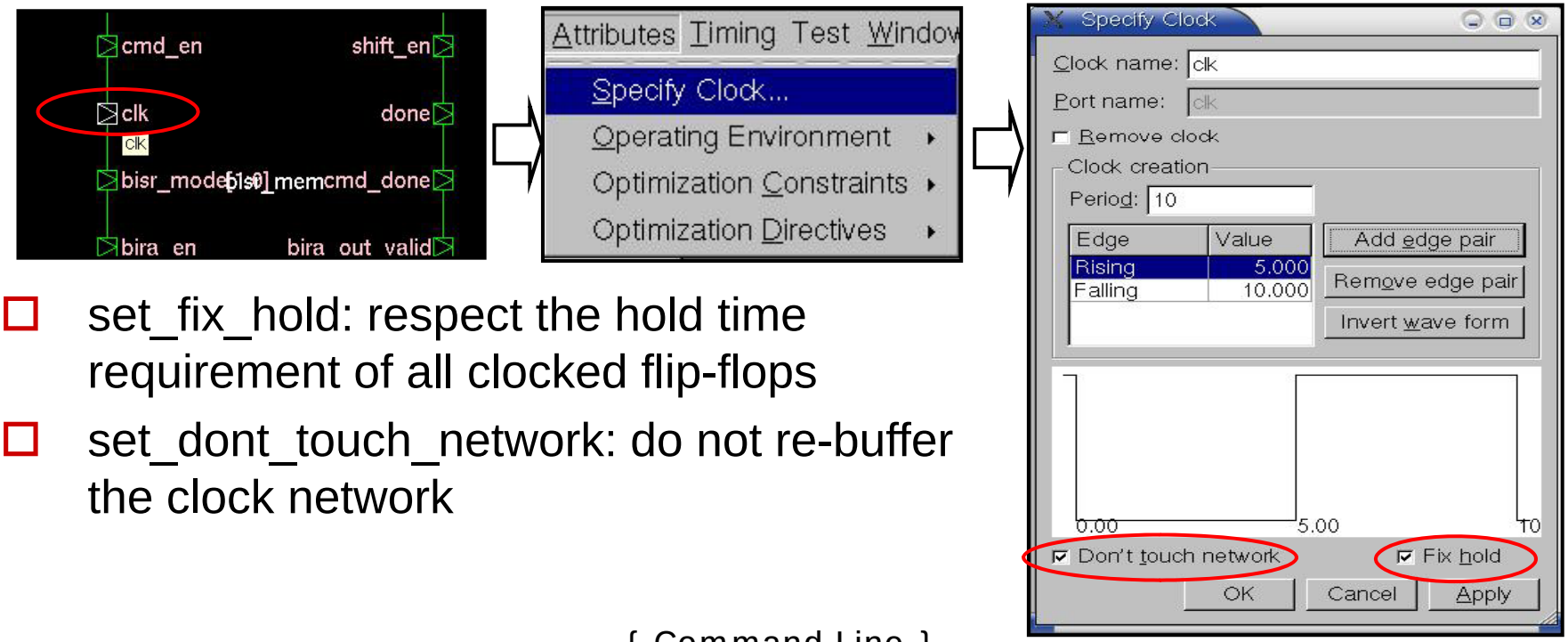
```
set_wire_load_model –name “tsmc18_wl10” –library “slow”
set_wire_lode_mode “top”
```

Clock Constraints

- Period
- Waveform
- Uncertainty
  - Skew
- Latency
  - Source latency
  - Network latency
- Transition
  - Input transition
  - Clock transition
- Combination Circuit – Maximum Delay Constraints

Sequential Circuit → Specify Clock

- ***Select the "clk" pin on the symbol***
- ***Attributes/Specify Clock***



- set_fix_hold: respect the hold time requirement of all clocked flip-flops
- set_dont_touch_network: do not re-buffer the clock network

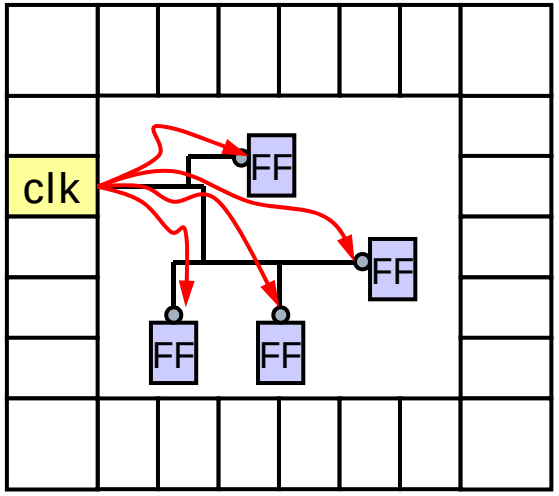
{ Command Line }

```
creat_clock –period 10 [get_ports clk]
set_dont_touch_network [get_clocks clk]
set_fix_hold [get_clocks clk]
```

Setting Clock Skew

- Different clock arrival time

Ex:




```
memory_8k_64_2r_2c/aru/U488/Y (MXI4X1)                  0.40      10.69 r
memory_8k_64_2r_2c/aru/U668/Y (NOR2X1)                  0.09      10.77 f
memory_8k_64_2r_2c/aru/data_in_sc[0] (aru)              0.00      10.77 f
memory_8k_64_2r_2c/sc_memory/D[0] (sc_memory)           0.00      10.77 f
data arrival time                                                 10.77

clock clk (rise edge)                                  10.00      10.00
clock network delay (ideal)                             1.00      11.00
clock uncertainty                                      -0.10      10.90
memory_8k_64_2r_2c/sc_memory/CLK (sc_memory)            0.00      10.90 r
library setup time                                     -0.12      10.78
data required time                                                10.78
-----------------------------------------------------------------------
data required time                                                10.78
data arrival time                                                -10.77
-----------------------------------------------------------------------
slack (MET)                                                       0.00
```

(Timing Report)

- experience
  - Small circuit: 0.1 ns
  - Large circuit: 0.3 ns

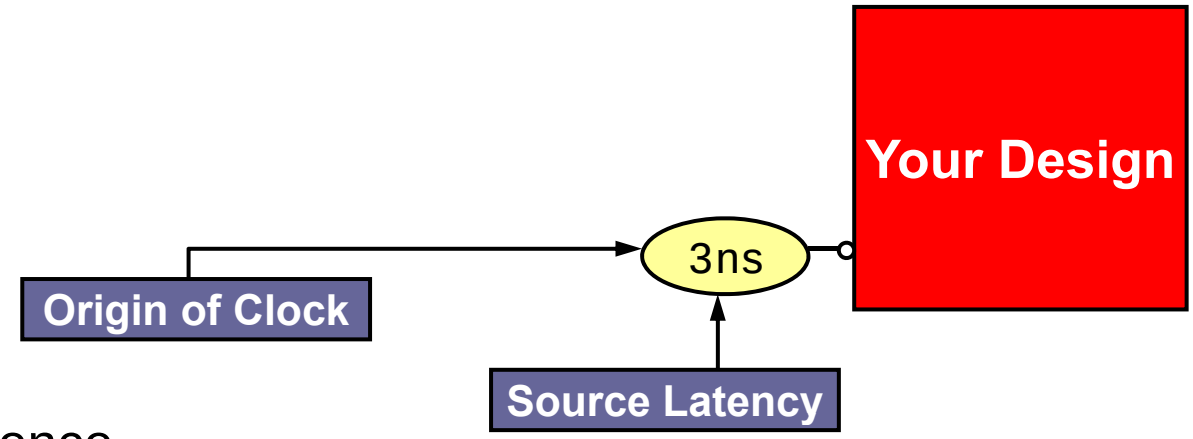
{ Command Line }

```
set_clock_uncertainty 0.1 [get_clocks clk]
```

Setting Clock Latency

- *Source latency* is the propagation time from the actual clock origin to the clock definition point in the design
- This setting can be avoid if the design is without the clock generator

Ex:



- experience
  - Small circuit: 1 ns
  - Large circuit: 3 ns

{ Command Line }

```
set_clock_latency 1 [get_clocks clk]
```

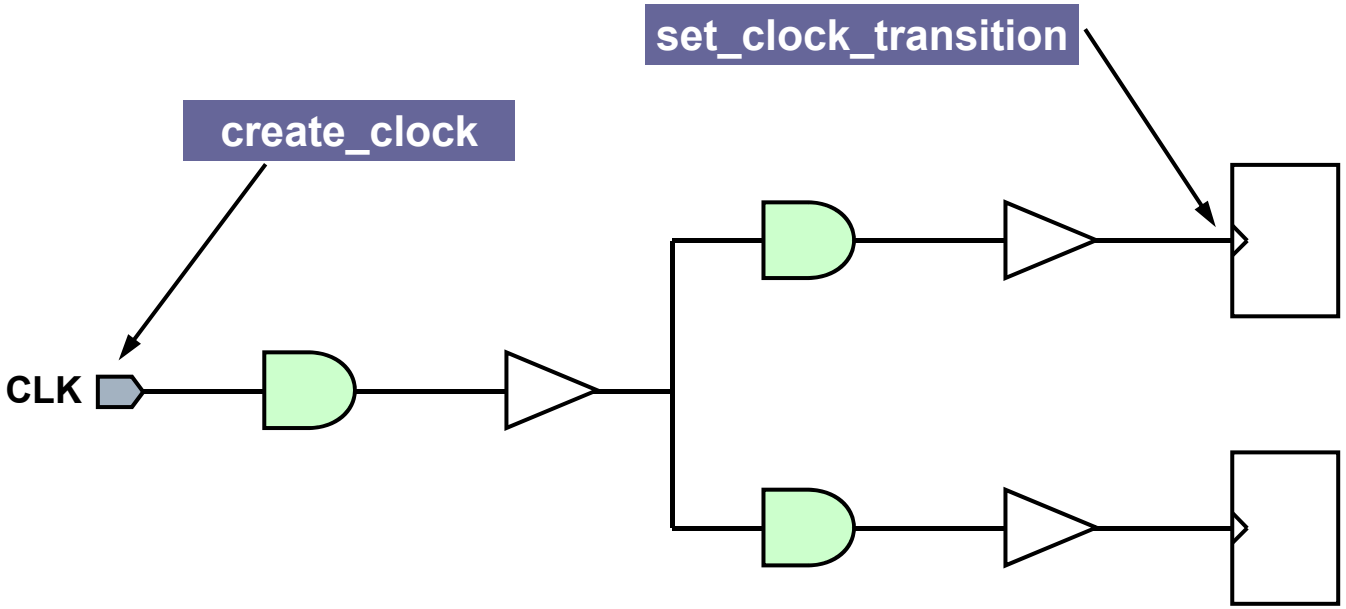
Setting Ideal Clock

- Since we usually let the clock tree synthesis (CTS) procedure performed in the P&R (i.e. set_dont_touch_network), the clock source driving capability is poor
- Thus, we can set the clock tree as an ideal network without driving issues
  - Avoid the hazard in the timing evaluation

{ Command Line }

```
set_ideal_network [get_ports clk]
```

Setting Clock Transition



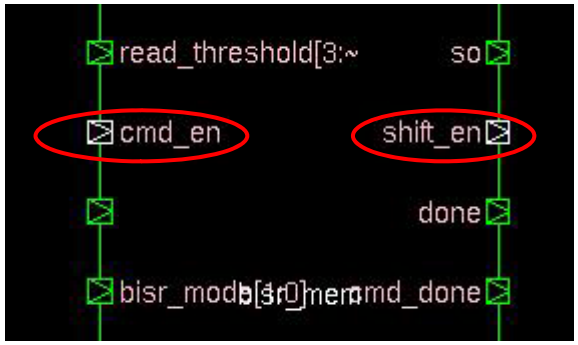
- experience
  - < 0.5ns
  - CIC tester: 0.5 ns

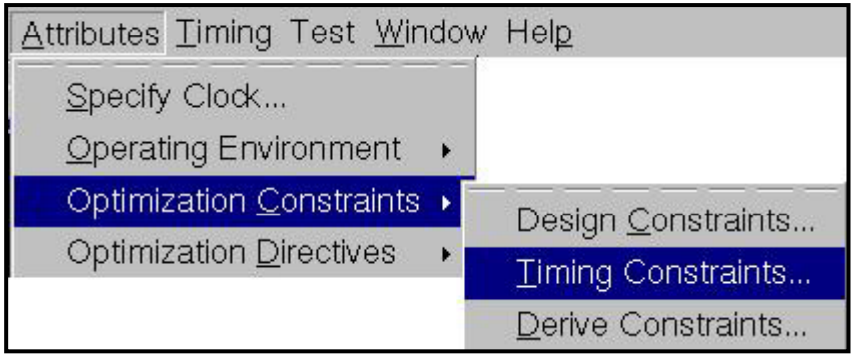
{ Command Line }

```
set_input_transition –max 0.5 [all_inputs]
```

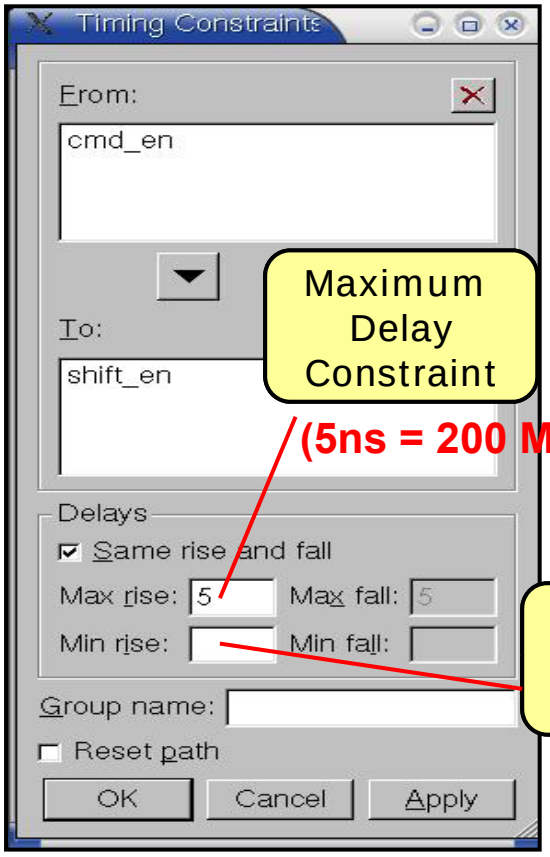
Combination Circuit – Maximum Delay Constraints

- For combinational circuits primarily (i.e. design with no clock)
  - Select the start & end points of the timing path
  - ***Attributes/Optimization Constraints/Timing Constraints***





Ex:



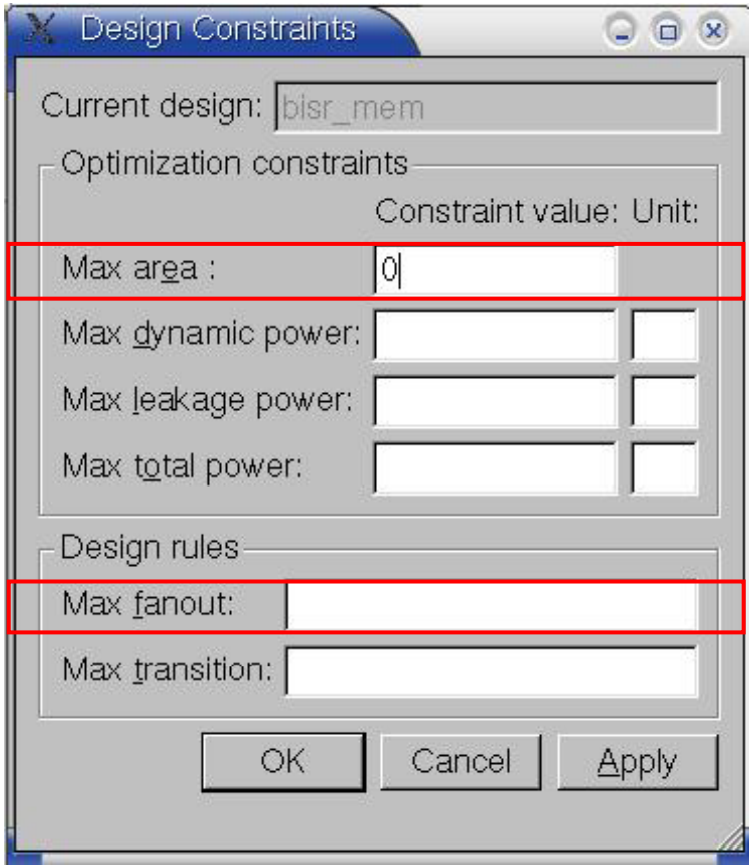
Design Rule Constraints

Setting Design Rule Constraints

- Area Constraint
- Fanout Constraint

Setting Area/Fanout Constraint

- ***Attributes/Optimization Constraints/Design Constraints***
- If you only concern the circuit area, but don't care about the timing
  - You can set the max area constraints to 0

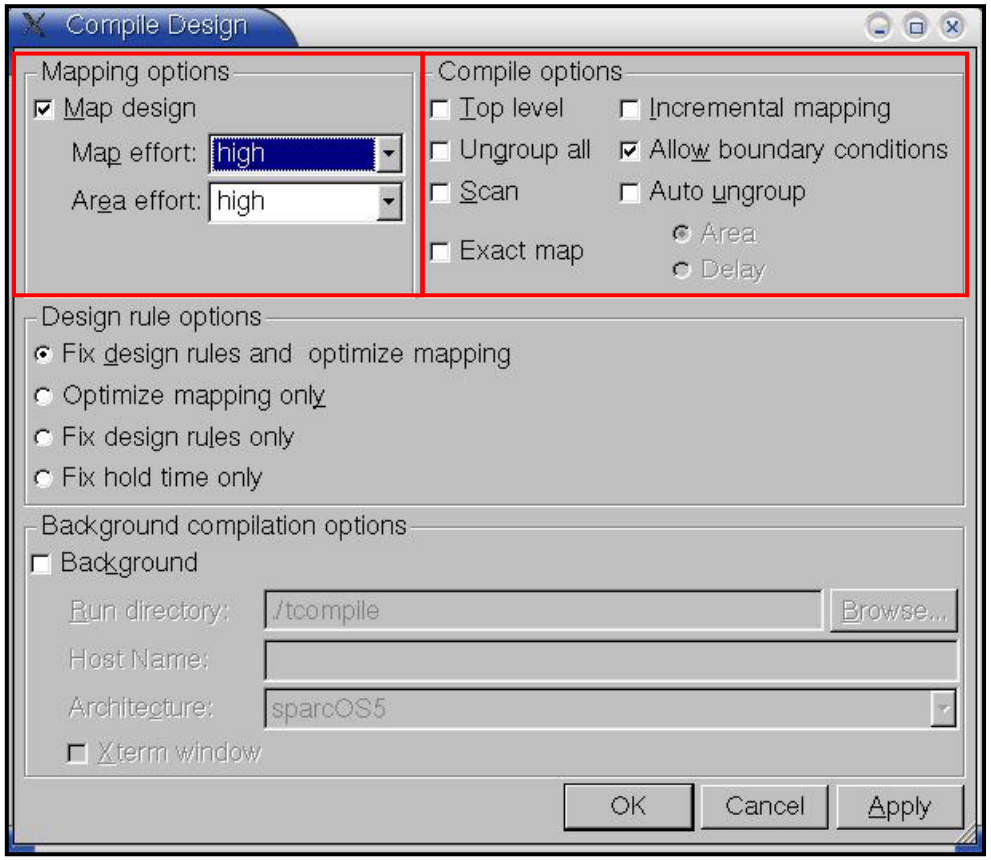


{ Command Line }

```
set_max_area 0
set_max_fanout 50 [get_designs CORE]
```

Compile the Design

Design/Compile Design



{ Command Line }

compile –map_effort high –boundary_optimization

Assign Problem

Assign Violation Avoidance

- The syntax of “assign” may cause problems in the LVS

```
assign \A[19] = A[19];
assign \A[18] = A[18];
assign \A[17] = A[17];
assign \A[16] = A[16];
assign \A[15] = A[15];
assign ABSVAL[19] = \A[19];
assign ABSVAL[18] = \A[18];
assign ABSVAL[17] = \A[17];
assign ABSVAL[16] = \A[16];
assign ABSVAL[15] = \A[15];
```

```
BUFX1 X37X( .I(A[19]), .Z(ABSVAL[19]) );
BUFX1 X38X( .I(A[18]), .Z(ABSVAL[18]) );
BUFX1 X39X( .I(A[17]), .Z(ABSVAL[17]) );
BUFX1 X40X( .I(A[16]), .Z(ABSVAL[16]) );
BUFX1 X41X( .I(A[15]), .Z(ABSVAL[15]) );
```

{ Command Line }

```
set_fix_multiple_port_nets –all –constants –buffer_constants [get_designs *]
```

Floating Port Removing

- Due to some ports in the standard cells are not used in your design

{ Command Line }

```
remove_unconnected_ports –blast_buses [get_cells –hierarchical *]
```

Chang Naming Rule Script

Naming Rule Changing

- Purpose: Let the naming-rule definitions in the gate-level netlist are the same as in the timing file (e.g. *.sdf file)
  - Also, the wrong naming rules may cause problems in the LVS

{ Command Line }

```
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
```

Save Design

Save Design

- Five design files:
  - *.spf: test protocol file for ATPG tools (i.e. TetraMax)
  - *.sdc: timing constraint file for P&R
  - *.vg: gate-level netlist for P&R
  - *.sdf: timing file for Verilog simulation
  - *.db: binary file (i.e. all the constraints and synthesis results are recorded)

{ Command Line }

```
write_test_protocol -f stil -out "CHIP.spf"
write_sdc CHIP.sdc
write -format verilog -hierarchy -output "CHIP.vg"
write_sdf -version 1.0  CHIP.sdf
write -format db    -hierarchy -output "CHIP.db"
```

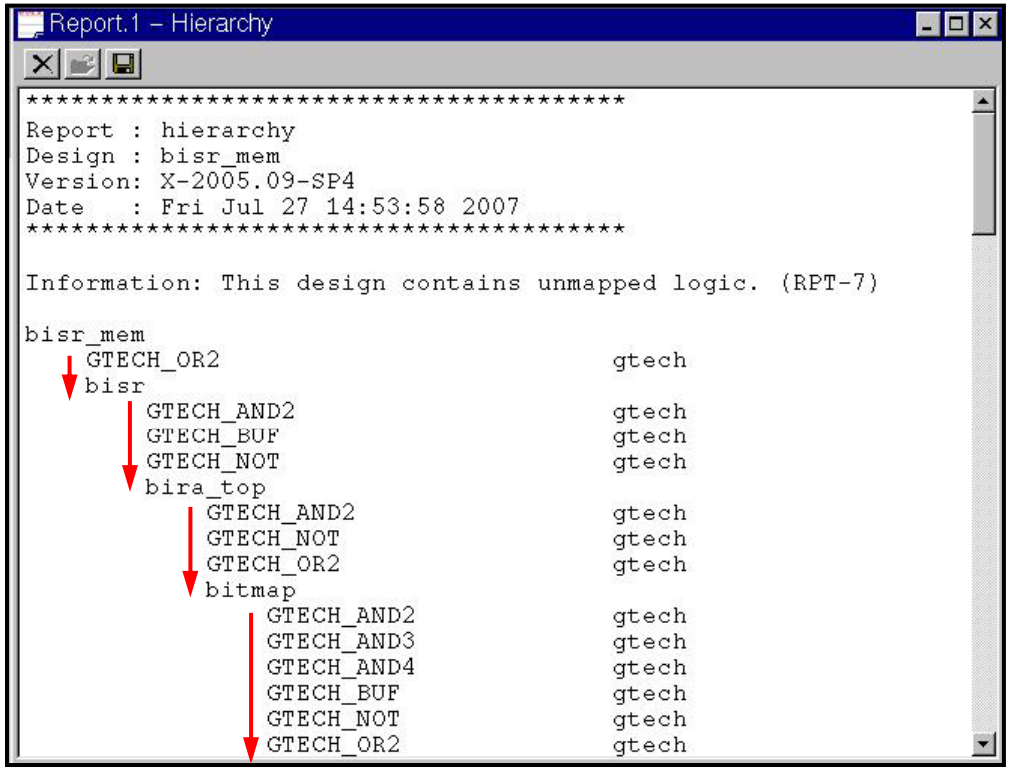
Synthesis Report

- Report Design Hierarchy
- Report Area
- Design View
- Report Timing
- Critical Path Highlighting
- Timing Slack Histogram

Report Design Hierarchy

- Hierarchy report shows the component used in your each block & its hierarchy
- ***Design/Report Design Hierarchy***

Ex:



Report Area

Design/Report Area

Ex:

```
****************************************
Report : area
Design : bisr_mem
Version: X-2005.09-SP4
Date   : Fri Jul 27 15:31:16 2007
****************************************

Library(s) Used:
    gtech (File: /usr/cad/synopsys/synthesis/cur/libraries/syn/gtech.db)
    USERLIB (File: /usr4/grad92/zwtseng/dv_training/RTL/MEM/DB/memory_8k_32_fast@-40C_syn.db)
    USERLIB (File: /usr4/grad92/zwtseng/dv_training/RTL/MEM/DB/sc_memory_fast@-40C_syn.db)
    USERLIB (File: /usr4/grad92/zwtseng/dv_training/RTL/MEM/DB/sr_memory_fast@-40C_syn.db)

Number of ports:              105
Number of nets:               248
Number of cells:                3
Number of references:           3

Combinational area:         0.000000
Noncombinational area:    3271507.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          3271507.000000
Total area:                 undefined

Information: This design contains unmapped logic. (RPT-7)
Information: This design contains black box (unknown) components. (RPT-8)

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

(0.18um Cell-Library: 1 gate ≈ 10 um^2)
(0.13um Cell-Library: 1 gate ≈ 5 um^2)

(um^2)

Design View

- ***List/Design View***

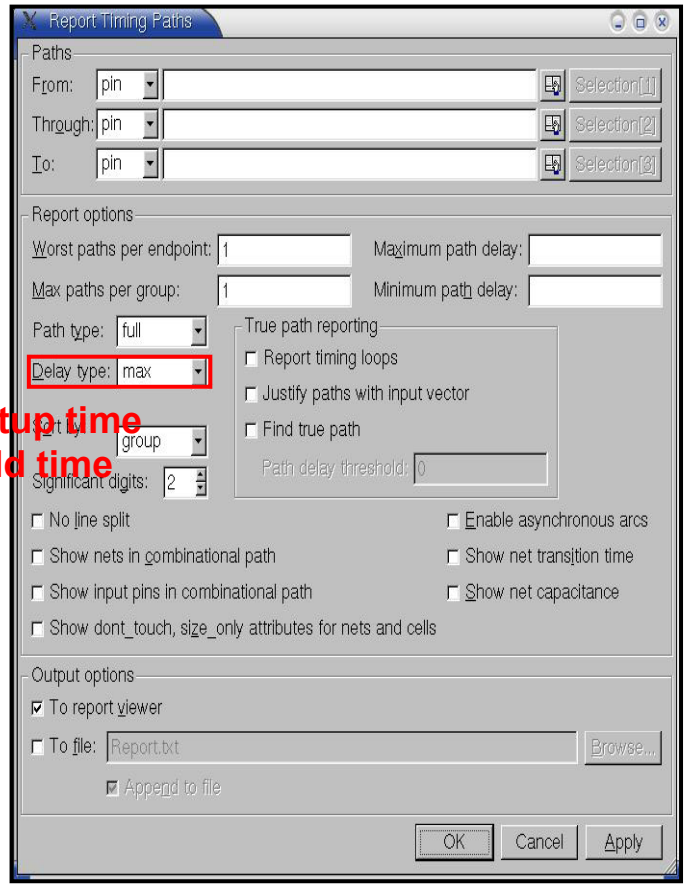
All the block area are listed !!

Ex:

array_or	851.558	undefined	undefined	undefined
bitmap	15501	undefined	undefined	undefined
bisr_mem	3.34175e+06	undefined	undefined	undefined
memory_8k...	3.29505e+06	undefined	undefined	undefined
aru	7717.25	undefined	undefined	undefined
remapping_...	4440.74	undefined	undefined	undefined
address_de...	8276.08	undefined	undefined	undefined
finj	1293.97	undefined	undefined	undefined
bisr	46686	undefined	undefined	undefined
bira_top	26953.8	undefined	undefined	undefined
remap	8016.62	undefined	undefined	undefined
remap_DW...	192.931	undefined	true	false
multi_bit	1593.35	undefined	undefined	undefined
bitmap_DW...	83.16	undefined	undefined	undefined
fsm	1816.21	undefined	undefined	undefined
bist	17413.7	undefined	undefined	undefined
tpg	15151.8	undefined	undefined	undefined
ADDR	2421.62	undefined	undefined	undefined
ADDR_DW...	435.758	undefined	true	false
ADDR_DW...	472.349	undefined	true	false
CMP	9433.67	undefined	undefined	undefined
DATA	219.542	undefined	undefined	undefined
DECO	804.989	undefined	undefined	undefined
ROM	1147.61	undefined	undefined	undefined
ctr	2261.95	undefined	undefined	undefined

Report Timing

- *Timing/Report Timing*



max: setup time
min: hold time

Ex:

setup time

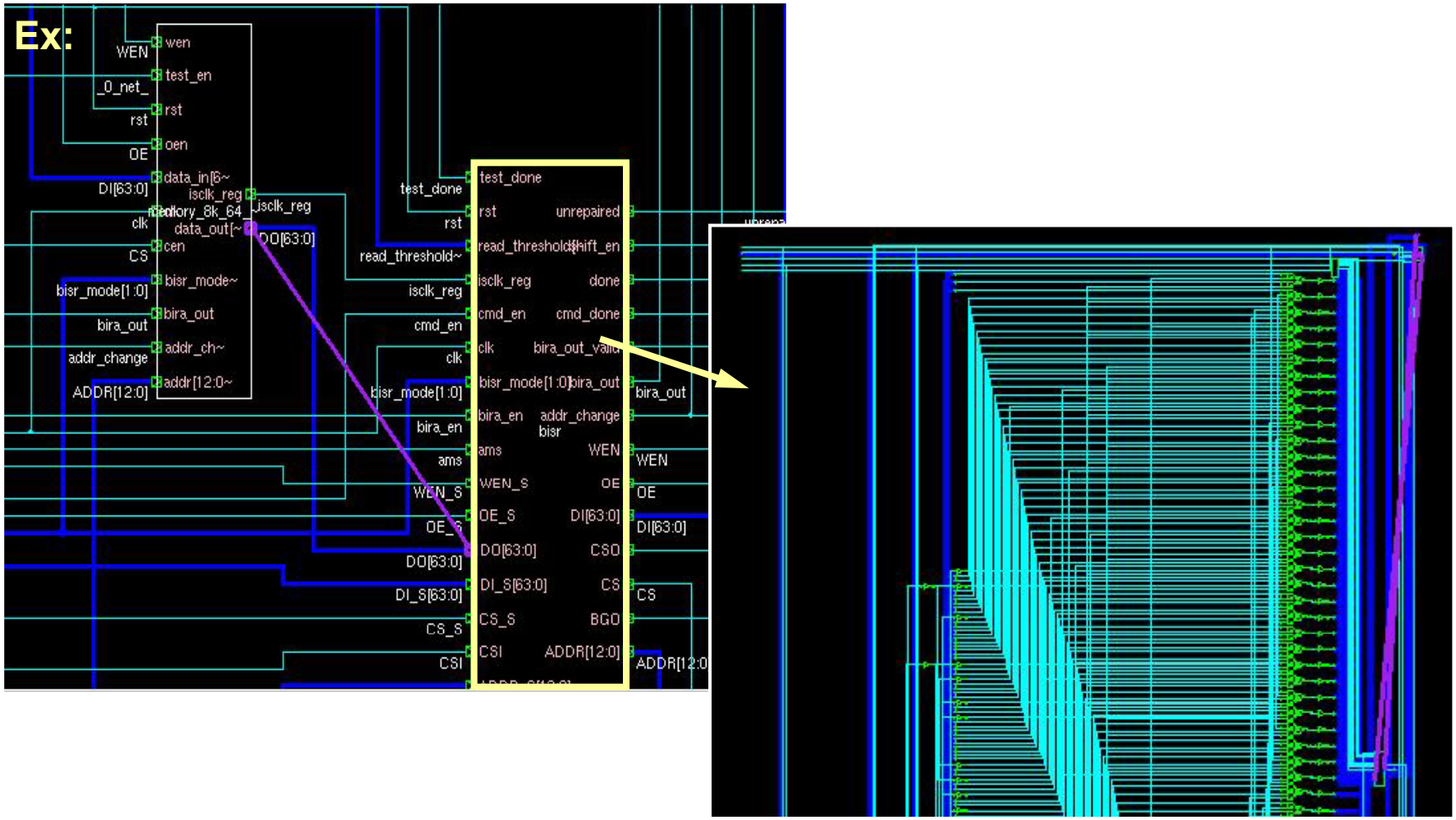
Critical Path

```
Path Type: max

Point                                                   Incr       Path
-----------------------------------------------------------------------
clock clk (rise edge)                                   5.00       5.00
clock network delay (ideal)                             0.00       5.00
memory_8k_64_2r_2c/sc_memory/CLK (sc_memory)            0.00       5.00 r
memory_8k_64_2r_2c/sc_memory/Q[0] (sc_memory)           0.71       5.71 r
memory_8k_64_2r_2c/aru/Q_sc[0] (aru)                    0.00       5.71 r
memory_8k_64_2r_2c/aru/U825/Y (AOI22X1)                 0.06       5.78 f
memory_8k_64_2r_2c/aru/U447/Y (INVX1)                   0.70       6.47 r
memory_8k_64_2r_2c/aru/U643/Y (AOI22X1)                 0.06       6.53 f
memory_8k_64_2r_2c/aru/U642/Y (OAI2BB1X1)               0.12       6.64 r
memory_8k_64_2r_2c/aru/data_out[47] (aru)               0.00       6.64 r
memory_8k_64_2r_2c/data_out[47] (memory_8k_64_2r_2c)
                                                        0.00       6.64 r
bisr/DO[47] (bisr)                                      0.00       6.64 r
bisr/bist/DO[47] (bist)                                 0.00       6.64 r
bisr/bist/m1/DO[47] (tpg)                               0.00       6.64 r
bisr/bist/m1/CMP/DO[47] (CMP)                           0.00       6.64 r
bisr/bist/m1/CMP/U207/Y (XNOR2X1)                       0.20       6.85 r
bisr/bist/m1/CMP/U185/Y (NAND4X1)                       0.07       6.92 f
bisr/bist/m1/CMP/U204/Y (NOR4X1)                        0.20       7.11 r
bisr/bist/m1/CMP/U202/Y (NAND4X1)                       0.06       7.17 f
bisr/bist/m1/CMP/U171/Y (AND2X2)                        0.13       7.30 f
bisr/bist/m1/CMP/U174/Y (INVX1)                         1.43       8.74 r
bisr/bist/m1/CMP/U170/Y (NAND2X2)                       0.64       9.38 f
bisr/bist/m1/CMP/U334/Y (OAI22X1)                       0.32       9.70 r
bisr/bist/m1/CMP/syndrome_reg[0]/D (DFFNRX1)            0.00       9.70 r
data arrival time                                                  9.70

clock clk (fall edge)                                  10.00      10.00
clock network delay (ideal)                             0.00      10.00
bisr/bist/m1/CMP/syndrome_reg[0]/CKN (DFFNRX1)          0.00      10.00 f
library setup time                                     -0.09       9.91
data required time                                                 9.91

data required time                                                 9.91
data arrival time                                                 -9.70
-----------------------------------------------------------------------
slack (MET)                                                        0.21

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

Slack = Data Require Time – Data Arrival Time

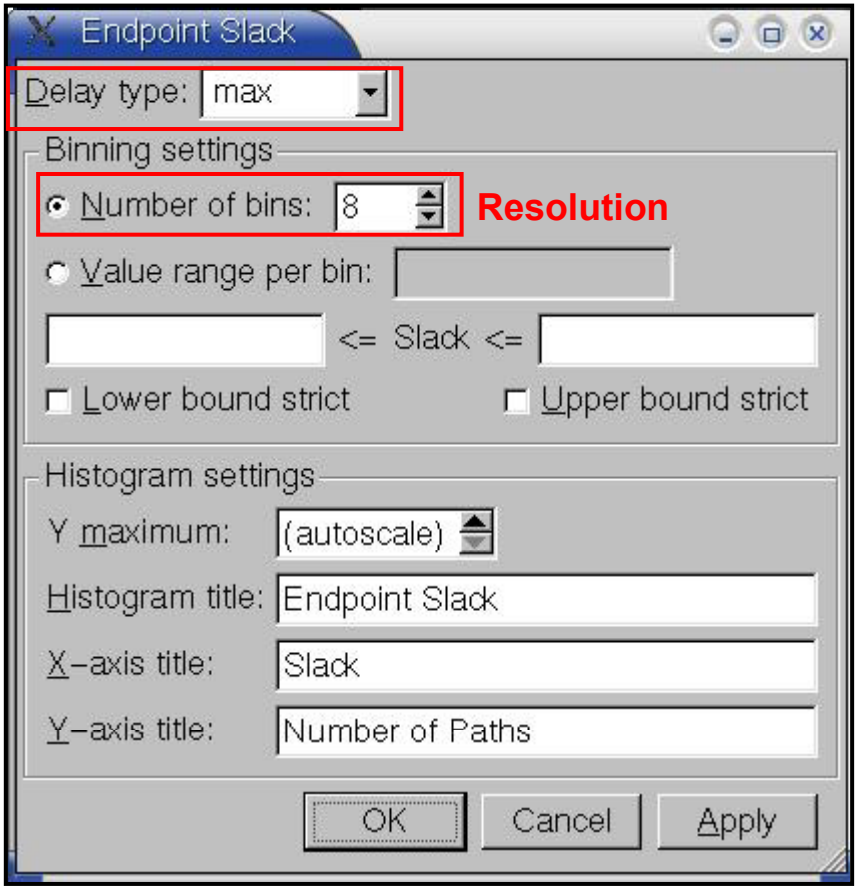
Critical Path Highlighting

- ***View/Highlight/Critical Path***

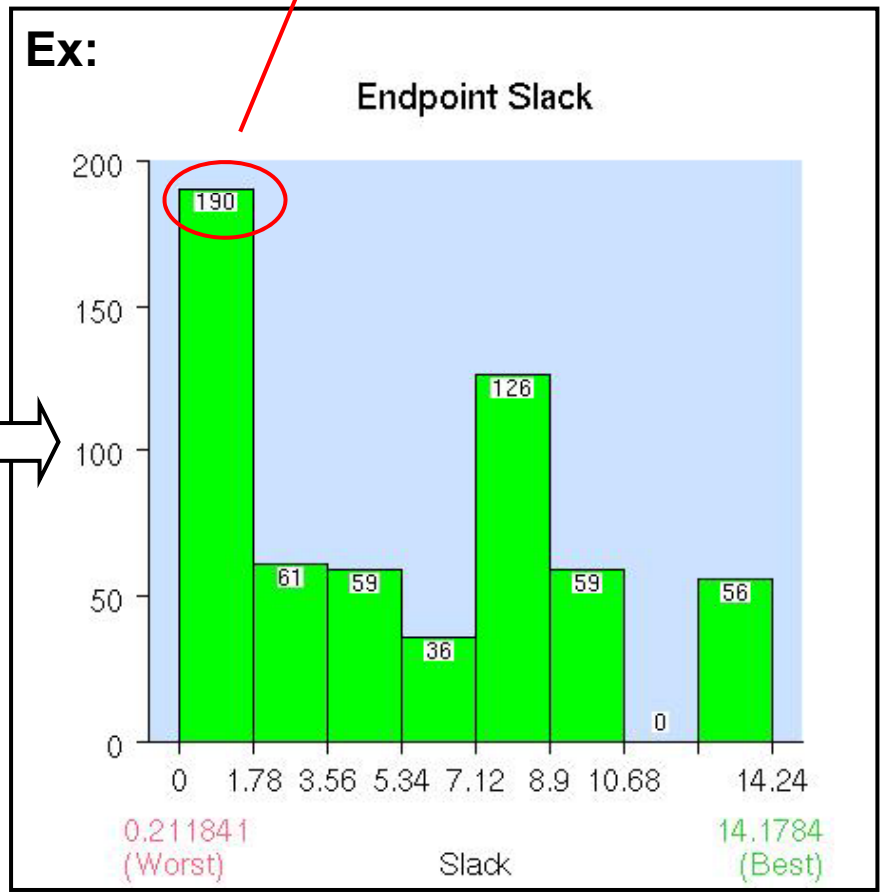


Timing Slack Histogram

- ***Timing/Endpoint Slack***



Totally 190 paths are in the slack range between 0 to 1.78



Edit Your Own Script File

- For convenient, you should edit your own synthesis script file. Whenever you want to synthesis a new design, you just only change some parameters in this file.
- Execute Script File
  - ***File/Execute Script***
  - ***Or use "source your_script.dc" in dc_shell command line***

Ex:

```
####################Parameters#######################################
#set cycle 10
#set t_in 5
#set t_out 0.5
#set in_pad_delay 0.34
#set out_pad_delay 0.96
#################Set Current Design#################################
current_design bisr_mem
uniquify
#################Operating Evironment###############################
set_operating_conditions -max "slow" -max_library "slow" -min "fast"\
-min_library "fast"
set_wire_load_model  -name "tsmc18_wl10" -library "slow"
set_wire_load_mode "top"

#################Clock Settings#####################################
create_clock -period 10 [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  1 [get_clocks clk];
set_ideal_network [get_ports clk]
set_input_transition -max 0.5 [all_inputs];

#################In/Out Delay, Driving, and Loading Settings#######
set_input_delay  [expr 0.34] -clock clk [all_inputs]
set_input_delay  [expr 0.34] -clock clk [get_ports clk]
#set_input_delay  [expr 1+0.34] -clock clk [get_ports clk]
set_output_delay [expr 0.5+1.5] -clock clk [all_outputs]
set_load [expr 0.06132] [all_outputs]
set_drive [expr 0.288001] [all_inputs]

#################Design Rule Constraint#############################
set_max_fanout 50 [get_designs bisr_mem]
#set_max_area 0

#################Avoid Multi-Instance Wrning########################
```

Gate-Level Simulation

- Include the Verilog model of standard cell and gate-level netlist to your test bench

```
`include "tsmc18.v"
`include "bisr_mem.vg"
`include "memory_8K_32.v"
`include "sc_memory.v"
`include "sr_memory.v"
```

Standard Cell Library

Gate- Level Netlist

- Add the following Synopsys directives to the test bench

```
initial
begin
$fsdbDumpfile("bisr_mem.fsdb");
$fsdbDumpfile("bisr.fsdb");
$fsdbDumpvars;
end

initial
begin
$sdf_annotate("bisr_mem.sdf",bisr_mem);
end

initial begin
  #0              clk = 1'b0;
  forever #50.0000    clk = ~clk;
end
```

***.sdf File**

Instance Name

Lab.

- cp –r –f /usr2/grad97/tjchen/tutorial_of_DV/Lab .