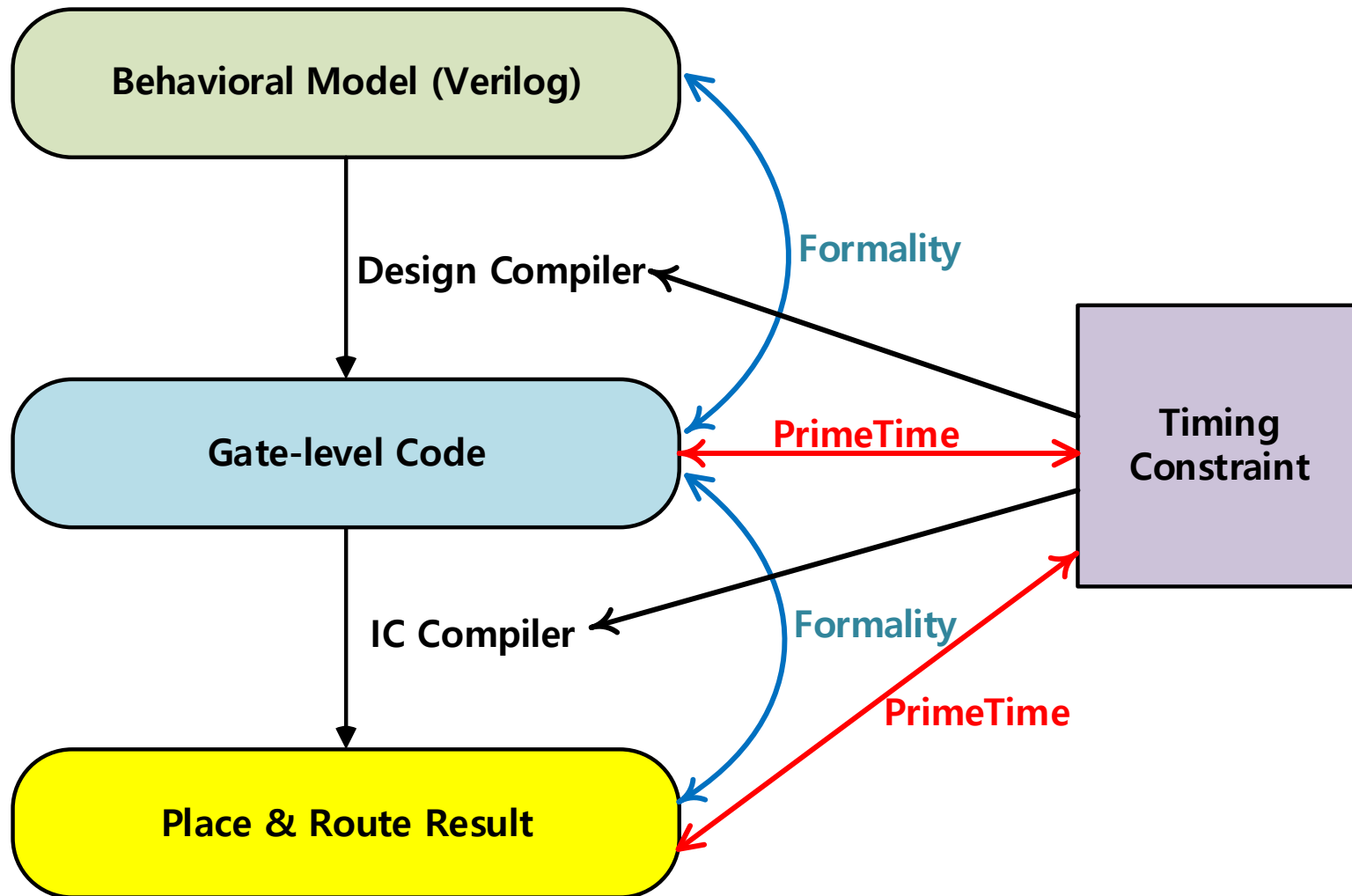


# Digital-Loop Filter Synthesis and Verification

**Jaeha Kim, Sung-Joon Lee, and Eunseo Kim**  
**Mixed-Signal IC and System Group**  
**Seoul National University**  
**May 19. 2016**

# Designing Digital System by Synopsys Tool



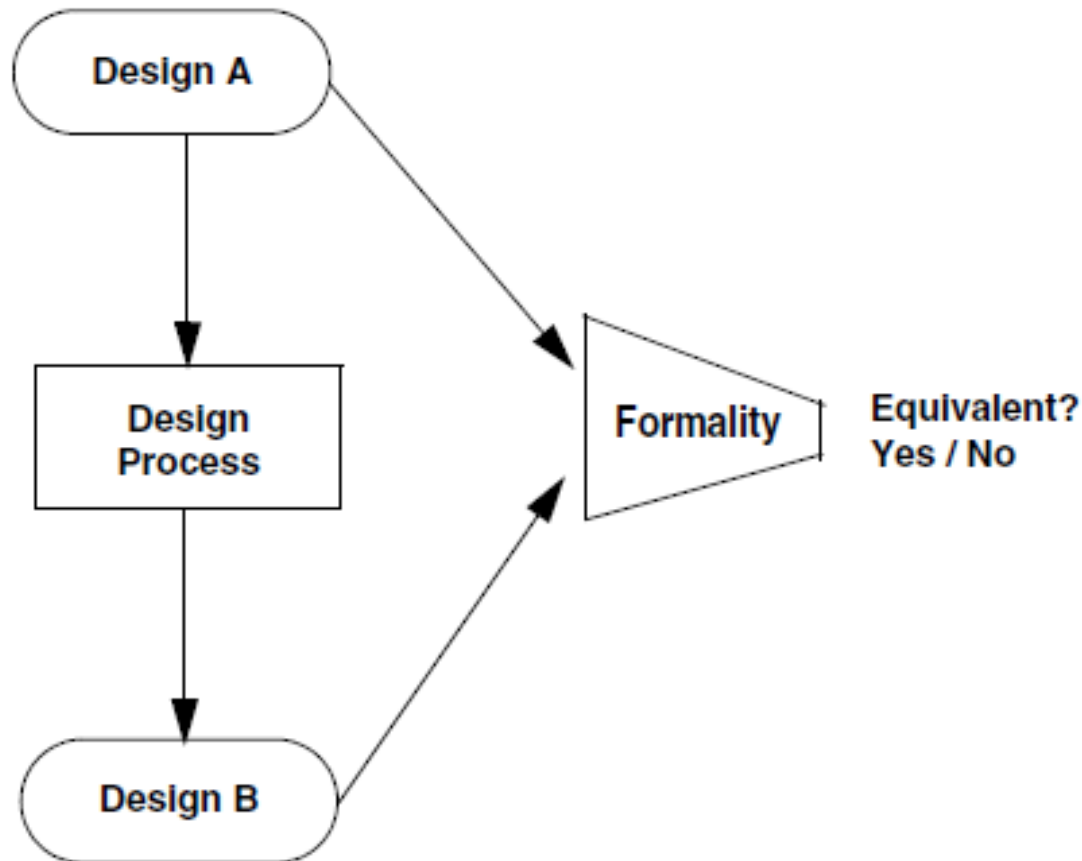
# Design Compiler

- ▶ **Design Compiler** is the core of the Synopsys synthesis tool suite
  - ▶ Design Compiler optimizes designs to provide the smallest and fastest logical representation of a given function
  - ▶ It comprises of tools that synthesize the HDL designs into optimized technology-dependent, gate-level designs

# Formality : Equivalence Checker

- ▶ The purpose of Formality is to detect unexpected differences that might have been introduced into a design during development by using **formal verification**
- ▶ **Formal verification** uses mathematical techniques to compare the logic to be verified against either a logical specification or a reference design
  - ▶ **reveals unexpected differences without relying on input vector sets**, which verifies large designs faster than simulation while **providing 100 percent coverage**

# Formality : Equivalence Checker

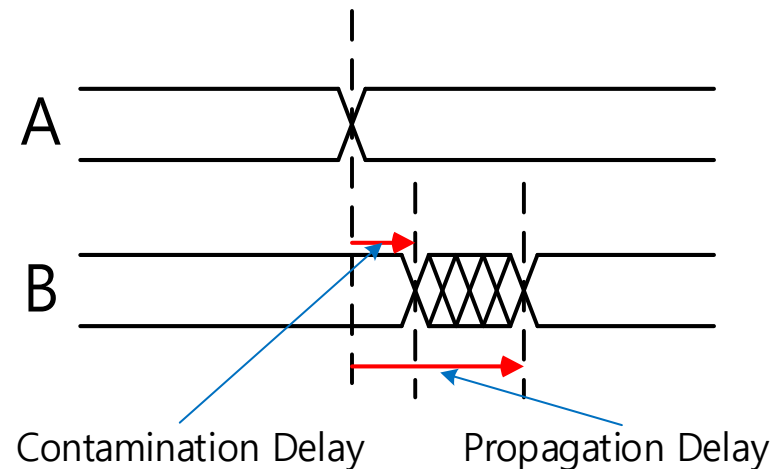
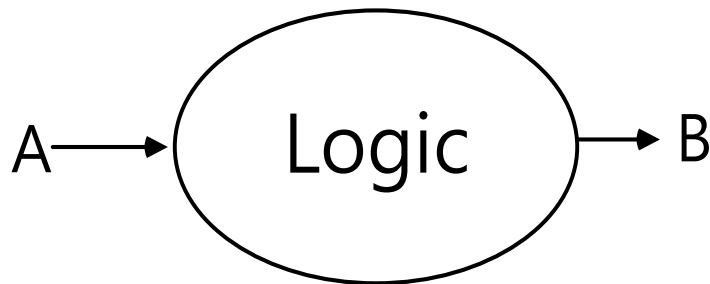


# PrimeTime : Timing Checker

- ▶ PrimeTime performs full-chip, gate-level **static timing analysis (STA)**
- ▶ **Static timing analysis** is a method of computing the expected timing of a digital circuit **without requiring simulation**

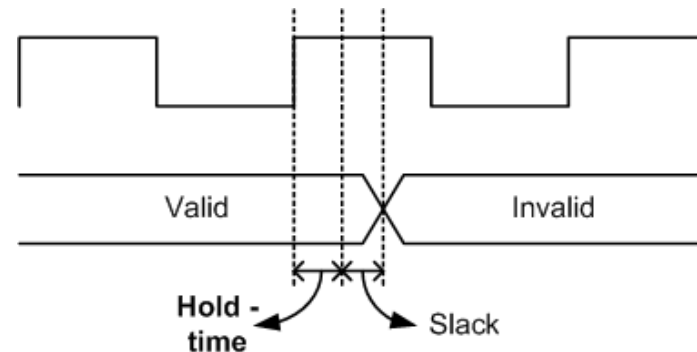
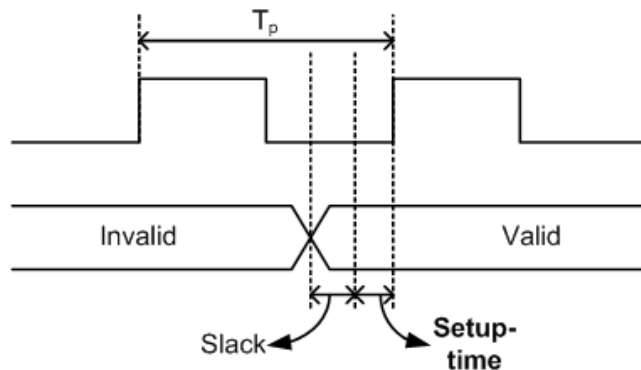
# Delay in Digital Systems

- ▶ Propagation Delay : time after the input that the output is guaranteed to be stable
- ▶ Contamination Delay : time after the input that the output becomes unstable



# Timing Constraint

- ▶ To make sure the data is sampled safely, data should be stable before and after the edge of the clock
  - ▶ Setup time : the region of time that data should not change “before” the edge of clock
  - ▶ Hold time : the region of time that data should not change “after” the edge of clock

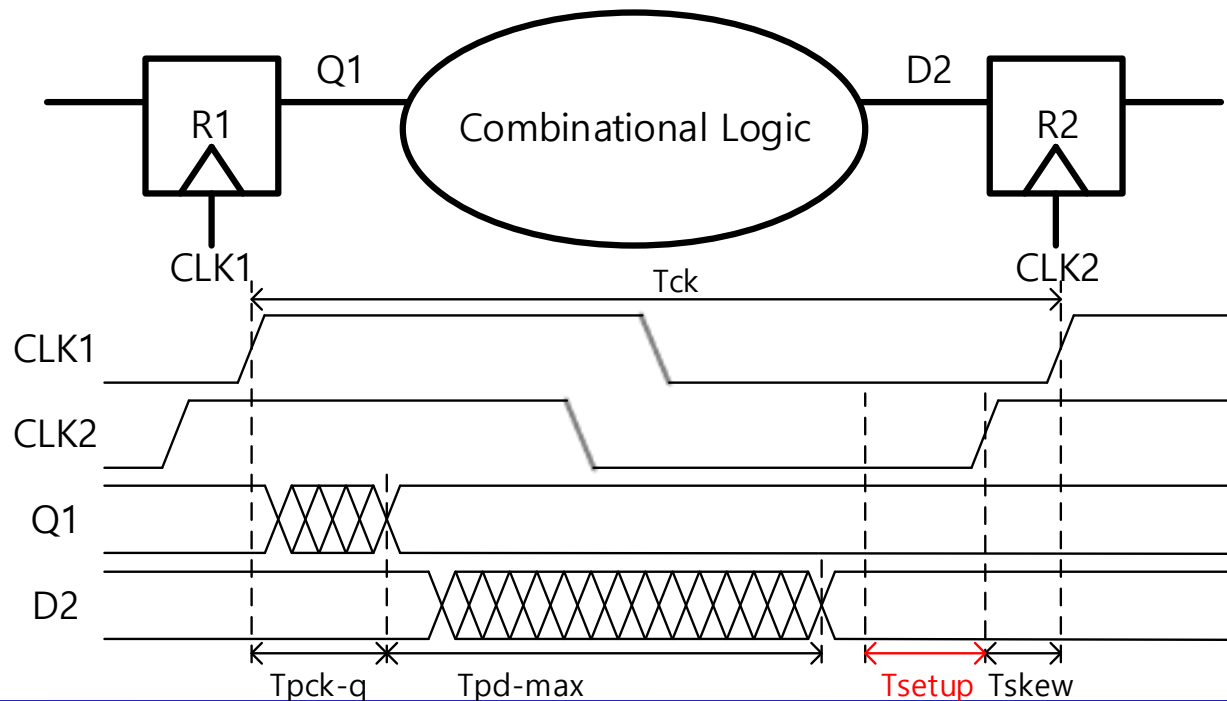




# Setup Time Constraint

- ▶ depend on **the maximum propagation delay of the combinational logic** between the registers
- ▶ In the worst case, clk2 is earlier than clk1 due to skew

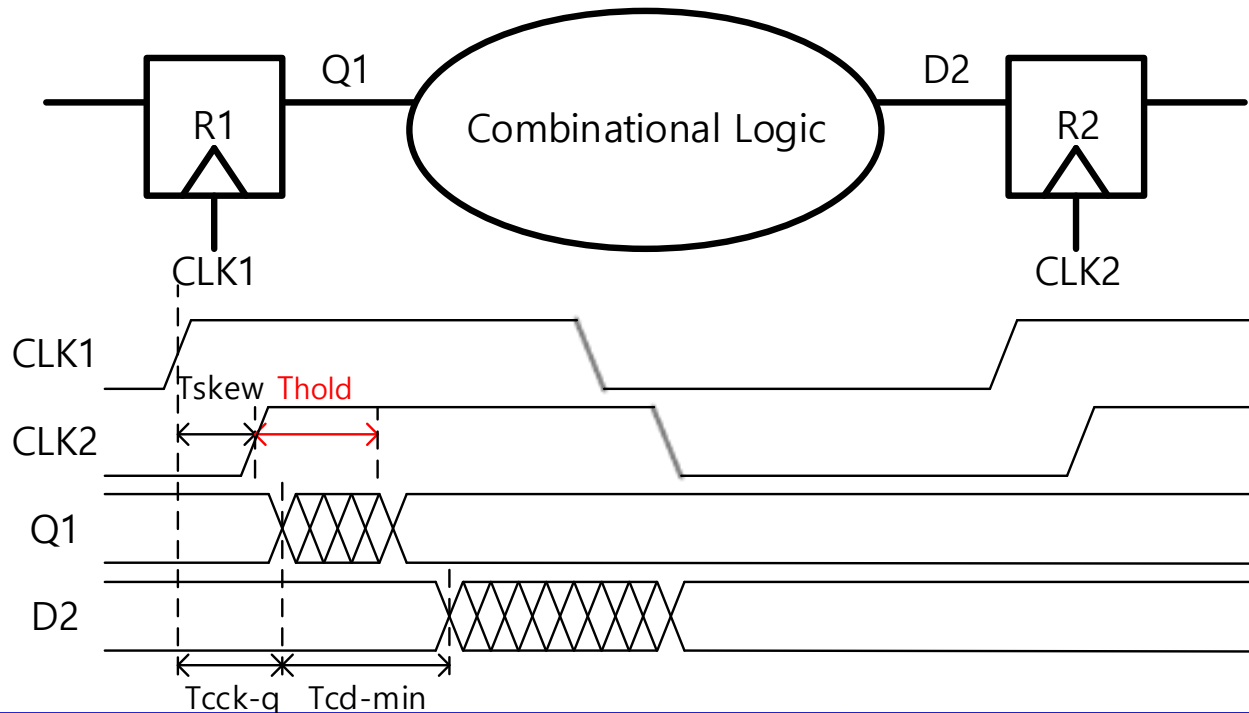
$$T_{ck} \geq T_{pck-q} + T_{pd-max} + T_{setup} + T_{skew}(\pm T_{jitter})$$



# Hold Time Constraint

- ▶ depend on **the minimum contamination delay of the combinational logic** between the registers
- ▶ In the worst case, clk1 is earlier than clk2 due to skew

$$T_{cck-q} + T_{cd-min} \geq T_{hold} + T_{skew}(\pm T_{jitter})$$



# Automation Scripts for Synthesis

- ▶ Let's see the scripts we have:

```
$~/IDEC_CBDF> cd xmodel_dp11/dp11/tb/tb_syn  
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn> ls
```

- ▶ You'll see
  - ▶ Makefile : Running files for scripts
  - ▶ tcl : Folder that contains scripts

# Automation Scripts for Synthesis

- ▶ Let's see the scripts we have:

```
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn> cd tcl  
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn/tcl> ls
```

- ▶ You'll see
  - ▶ digital\_lf.tcl : common setup file for synthesis & verification
  - ▶ {dc,fm,pt}\_digital\_lf.tcl : Scripts for running Design Compiler / Formality / PrimeTime

# Constraint File for Timing

- ▶ When running Design Compiler, the tool gets the constraint of the timing of design from the constraint file, which is located in :

```
$ cd ~/IDEC_CBDF/Synthesis/constraints/  
$~/IDEC_CBDF/Synthesis/constraints> ls
```

- ▶ constraints\_digital\_lf.tcl : collection of commands related to timing

# Constraint Commands in the File

► \$~/IDEC\_CBDF/Synthesis/constraints> vi constraints\_digital\_lf.tcl

```
set_max_capacitance 1.000 [current_design]
```

```
set_max_transition 0.250 [current_design]
```

Setting max capacitance & transition of design

```
create_clock clk -name "clk" -period 8.00 -waveform {0.100 4.100}
```

```
set_propagated_clock [all_clocks]
```

```
set_clock_uncertainty -setup 0.2 clk
```

```
set_clock_uncertainty -hold 0.2 clk
```

Defining clock signal and its uncertainty

```
set_max_delay 3.0 -from up -to [get_clocks *]
```

```
set_min_delay 0.2 -from up -to [get_clocks *]
```

Constraint in Delay of I/O

```
...
```

```
set_fix_hold clk
```

```
set_load 0.02 [all_outputs]
```

```
set_driving_cell -lib_cell INV_X16 -library cello.slow [all_inputs]
```

Miscellaneous constraints

# Running Design Compiler

- ▶ Move to directory containing Makefile, and type “make dc”

```
$ cd ~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn/  
$ make dc
```

# Synthesis Result

## ► Let's check the result

```
$ cd ~/IDEC_CBDF/Synthesis/output/digital_lf
$~/IDEC_CBDF/Synthesis/output/digital_lf> ls
digital_lf_mapped.sdf  digital_lf_mapped.sdc  digital_lf_mapped.svf
digital_lf_mapped.rpt  digital_lf_mapped.v    ....
```

Standard Delay Format  
: Delay information of  
gates and interconnect

Synopsys Design Constraint  
: Timing constraint of Design


Synopsys Formality Guide File  
: automation setup file for  
formality



# Synthesis Result

## ► Let's check the result

```
$ cd ~/IDEC_CBDF/Synthesis/output/digital_lf
$~/IDEC_CBDF/Synthesis/output/digital_lf> ls
digital_lf_mapped.sdf  digital_lf_mapped.sdc  digital_lf_mapped.svf
digital_lf_mapped.rpt  digital_lf_mapped.v    ....
```



Report File  
: Detailed results of  
synthesis



Gate-level code  
: Synthesized code of digital loop filter

# Synthesis Result

- Open the \*.rpt file

```
$~/IDEC_CBDF/Synthesis/output/digital_lf> vi digital_lf_mapped.rpt
```

- The report file contains expected power consumption of the digital loop filter

```
Global Operating Voltage = 1.08
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 = 1uW
```

Hierarchy	Switch Power	Int Power	Leak Power	Total Power	%
digital_lf	7.72e-03	2.79e-02	0.464	3.61e-02	100.0

# Synthesis Result (2)

- ▶ Report file contains area of digital loop filter (wires are not included)

```
Number of ports:                28
Number of nets:                 124
Number of cells:                94
Number of combinational cells:  84
Number of sequential cells:     10
Number of macros/black boxes:   0
Number of buf/inv:              64
Number of references:           8

Combinational area:             481.375996
Buf/Inv area:                   157.584000
Noncombinational area:          258.719997
Macro/Black Box area:           0.000000
Net Interconnect area:          undefined (No wire load specified)

Total cell area:                740.095993
Total area:                     undefined
```

# Synthesis Result (3)

- ▶ Timing information is also represented in report file
- ▶ “MET” means timing constraint is satisfied
- ▶ “VIO” means violation of timing

Startpoint: up[0] (input port)  
Endpoint: out\_reg\_9\_ (rising edge-triggered flip-flop clocked by clk)  
Path Group: clk  
Path Type: max

Point	Incr	Path
-----		
input external delay	0.00	0.00 r
up[0] (in)	0.00	0.00 r
U101/Y (INV_X1)	0.03	0.03 f
U106/Y (NAND2_X1)	0.06	0.08 r
sub_45_U2_1/SUM (ADDF_X1)	0.22	0.30 f
U117/CO (ADDF_X1)	0.16	0.46 f
U116/CO (ADDF_X1)	0.15	0.61 f
U115/CO (ADDF_X1)	0.15	0.77 f
U114/CO (ADDF_X1)	0.15	0.92 f
U113/CO (ADDF_X1)	0.15	1.07 f
U112/CO (ADDF_X1)	0.15	1.23 f
U118/CO (ADDF_X1)	0.15	1.38 f
U119/CO (ADDF_X1)	0.14	1.52 f
U120/Y (MUXI2_X1)	0.07	1.59 f
U121/Y (MUXI2_X1)	0.07	1.66 f
U43/Y (INV_X1)	0.04	1.70 r
U44/Y (INV_X1)	0.02	1.73 f
out_reg_9_/D (DFF_RSTB_X1)	0.00	1.73 f
data arrival time		1.73
max_delay	3.00	3.00
clock network delay (propagated)	0.02	3.02
clock uncertainty	-0.20	2.82
library setup time	-0.05	2.77
data required time		2.77
-----		
data required time		2.77
data arrival time		-1.73
-----		
slack (MET)		1.04

# Equivalence Checking by Formality

- ▶ After running synthesis, you can quickly check the equivalence by running Formality
- ▶ Let's first take a look at the script for Formality
  - ▶ Please type the below commands at the terminal

```
$ cd ~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn/tcl/
```

```
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn/tcl> vi fm_digital_lf.tcl
```

# Automation script for Formality

► \$~/IDEC\_CBDF/xmodel\_dp11/dp11/tb/tb\_syn/tcl> vi fm\_digital\_lf.tcl

```
source -echo ./tcl/digital_lf.tcl  
set SVF_FILE "${OUTDIR}/${DESIGN}/${DESIGN}_mapped.svf"
```

Load SVF

```
set_svf -append ${SVF_FILE}
```

```
read_db ${LIBDIR}/cello.slow.db
```

Load digital cell library

```
read_sverilog -container r -libname WORK 12 ${PROJDIR}/${Submodule_6}
```

```
...
```

```
read_sverilog -container r -libname WORK 12 ${PROJDIR}/${Topmodule}
```

Load model & set top

```
set_top r:/WORK/${DESIGN}
```

```
read_sverilog -container i -libname WORK 12 ${PROJDIR}/${DESIGN}_mapped.sv
```

Load synthesis  
result & set top

```
set_top i:/WORK/${DESIGN}
```

```
match
```

Match & Verify

```
verif
```

# Running Formality

- ▶ Move to directory containing Makefile, and type “make fm”

```
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn/tcl> cd ../  
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn> make fm
```

# Formality Results

- GUI Console will pop up, and show the result

The screenshot shows the Formality (R) Console interface. The title bar reads "Formality (R) Console - Synopsys Inc.". The menu bar includes File, Edit, View, Designs, Run, ECO, Window, and Help. The toolbar contains various icons for file operations, design navigation, and analysis. A red circle highlights a green status bar in the top right corner that says "Verification Succeeded". Below the toolbar, the Reference and Implementation paths are both set to "r:/WORK/digital\_If". A progress bar at the bottom of the toolbar shows six steps: 0. Guidance, 1. Reference, 2. Implementation, 3. Setup, 4. Match, 5. Verify, and 6. Debug. The main table displays the results of the verification, with columns for Failing Points, Passing Points, Aborted Points, Unverified Points, Probe Points, Analyses, and Loops. The table lists 17 items, all of which are passing points. The bottom status bar shows "Number of Passing Points: 24" and "Display names: Original Mapped". There are buttons for "Analyze", "Analyze Selected Points", and "Get Loop Data".

Failing Points	Passing Points	Aborted Points	Unverified Points	Probe Points	Analyses	Loops
1	Port	clk_lf_dco				
2	Port	dsm_out[0]				
3	Port	dsm_out[1]				
4	Port	dsm_out[2]				
5	Port	out[0]				
6	Port	out[1]				
7	Port	out[2]				
8	Port	out[3]				
9	Port	out[4]				
10	Port	out[5]				
11	Port	out[6]				
12	Port	out[7]				
13	Port	out[8]				
14	Port	out[9]				
15	DFF	out_reg_0				
16	DFF	out_reg_1				
17	DFF	out_reg_2				



# Formality Results (2)

- Unverified points should be checked

The screenshot shows the Formality (R) Console interface. At the top, a green banner indicates "Verification Succeeded". Below this, the Reference and Implementation paths are both set to "r:/WORK/digital\_if". A progress bar shows six steps: 0. Guidance, 1. Reference, 2. Implementation, 3. Setup, 4. Match, 5. Verify, and 6. Debug. Steps 0, 1, and 2 are marked with green checkmarks. The main table displays verification results with columns for Failing Points, Passing Points, Aborted Points, Unverified Points, Probe Points, Analyses, and Loops. The Unverified Points column is currently empty. At the bottom left, a red circle highlights the text "Number of Unverified Points: 0". At the bottom right, there are buttons for "Analyze" and "Analyze Selected Points", and a "Display names" section with radio buttons for "Original" and "Mapped".

Formality (R) Console - Synopsys Inc.

File Edit View Designs Run ECO Window Help

Verification Succeeded

Reference: r:/WORK/digital\_if

Implementation: i:/WORK/digital\_if

0. Guidance 1. Reference 2. Implementation 3. Setup 4. Match 5. Verify 6. Debug

Failing Points	Passing Points	Aborted Points	Unverified Points	Probe Points	Analyses	Loops
Type	Reference	Size	Implementation	Size	+/-	Cause

Number of Unverified Points: 0

Display names: ☐ Original ☒ Mapped

Analyze Analyze Selected Points

# Static Timing Analysis by PrimeTime

- ▶ After running synthesis, you can quickly check the timing by running PrimeTime
- ▶ Let's first take a look at the script for Formality
  - ▶ Please type the below commands at the terminal

```
$~/IDEC_CBDF/xmodel_dpII/dpII/tb/tb_syn> cd tcl/
```

```
$~/IDEC_CBDF/xmodel_dpII/dpII/tb/tb_syn/tcl> vi pt_digital_lf.tcl
```

# Automation script for PrimeTime

```
source -echo ./tcl/digital_lf.tcl
```

```
...
```

```
set_app_var search_path      "${LIBDIR}"
```

```
set_app_var target_library   "${LIBDIR}/cello.slow.db"
```

Link Library of digital cells

```
set_app_var link_library     "${LIBDIR}/cello.slow.db"
```

```
read_verilog ${PROJDIR}/${DESIGN}_mapped.sv
```

Read & Link design

```
link_design ${DESIGN}
```

```
read_sdc ${PROJDIR}/${DESIGN}_mapped.sdc -version 1.9
```

Read SDC & SDF

```
read_sdf -analysis_type on_chip_variation ${OUTDIR}/${DESIGN}/${DESIGN}_mapped.sdf
```

```
check_timing -verbose >> "${LOG_PATH}/${RPT_FILE}"
```

Check & Report

```
...
```

# PrimeTime Result

- Open the \*.rpt file

```
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_syn/pt_log> vi pt_digital_1f.rpt
```

- The report file contains the result of static timing analysis

U113/CO (ADDF_X1)	0.15 *	1.07 f
U112/CO (ADDF_X1)	0.15 *	1.22 f
U118/CO (ADDF_X1)	0.15 *	1.38 f
U119/CO (ADDF_X1)	0.14 *	1.52 f
U120/Y (MUXI2_X1)	0.07 *	1.59 f
U121/Y (MUXI2_X1)	0.07 *	1.66 f
U43/Y (INV_X1)	0.04 *	1.70 r
U44/Y (INV_X1)	0.02 *	1.73 f
out_reg_9_/D (DFF_RSTB_X1)	0.00 *	1.73 f
data arrival time		1.73
max_delay	3.00	3.00
clock network delay (propagated)	0.02 *	3.02
clock uncertainty	-0.20	2.82
library setup time	-0.05 *	2.77
data required time		2.77
-----		
data required time		2.77
data arrival time		-1.73
-----		
slack (MET)		1.04

# PrimeTime Result (2)

- ▶ The report file also illustrates detail information about the test set

Type of Check	Total	Met	Violated	Untested
setup	10	10 (100%)	0 ( 0%)	0 ( 0%)
hold	10	10 (100%)	0 ( 0%)	0 ( 0%)
recovery	10	0 ( 0%)	0 ( 0%)	10 (100%)
removal	10	0 ( 0%)	0 ( 0%)	10 (100%)
min_pulse_width	30	20 ( 67%)	0 ( 0%)	10 ( 33%)
All Checks	70	40 ( 57%)	0 ( 0%)	30 ( 43%)

# PrimeTime Result (3)

- ▶ Untested points should be check, whether it is truly unnecessary to perform STA

Constrained Pin	Related Pin	Check Type	Slack	Reason
out_reg_5_/RSTB(low)		min_pulse_width		
			untested	no_clock
out_reg_5_/RSTB(rise)	CK(rise)	recovery	untested	no_startpoint_clock
out_reg_5_/RSTB(rise)	CK(rise)	removal	untested	no_startpoint_clock
out_reg_2_/RSTB(low)		min_pulse_width		
			untested	no_clock
out_reg_2_/RSTB(rise)	CK(rise)	recovery	untested	no_startpoint_clock
out_reg_2_/RSTB(rise)	CK(rise)	removal	untested	no_startpoint_clock
out_reg_6_/RSTB(low)		min_pulse_width		
			untested	no_clock
out_reg_6_/RSTB(rise)	CK(rise)	recovery	untested	no_startpoint_clock
out_reg_6_/RSTB(rise)	CK(rise)	removal	untested	no_startpoint_clock
out_reg_3_/RSTB(low)		min_pulse_width		
			untested	no_clock
out_reg_3_/RSTB(rise)	CK(rise)	recovery	untested	no_startpoint_clock
out_reg_3_/RSTB(rise)	CK(rise)	removal	untested	no_startpoint_clock
out_reg_7_/RSTB(low)		min_pulse_width		

# PrimeTime Result (4)

- Report file contains timing margin of each points

out_reg_9_/D	CK(rise)	hold	0.00
out_reg_2_/D	CK(rise)	hold	0.00
out_reg_3_/D	CK(rise)	hold	0.00
out_reg_4_/D	CK(rise)	hold	0.00
out_reg_5_/D	CK(rise)	hold	0.02
out_reg_0_/D	CK(rise)	hold	0.02
out_reg_1_/D	CK(rise)	hold	0.02
out_reg_6_/D	CK(rise)	hold	0.15
out_reg_7_/D	CK(rise)	hold	0.16
out_reg_8_/D	CK(rise)	hold	0.16
out_reg_9_/D	CK(rise)	setup	1.04
out_reg_8_/D	CK(rise)	setup	1.19
out_reg_7_/D	CK(rise)	setup	1.34
out_reg_6_/D	CK(rise)	setup	1.49
out_reg_5_/D	CK(rise)	setup	1.65
out_reg_4_/D	CK(rise)	setup	1.70
out_reg_3_/D	CK(rise)	setup	1.85
out_reg_2_/D	CK(rise)	setup	2.00
out_reg_1_/D	CK(rise)	setup	2.09
out_reg_0_/D	CK(rise)	setup	2.14

# Simulating Gate-level Code with Annotating SDF

- ▶ SDF (Standard Delay Format) : contains the delay information in interconnects and cells

```
DELAYFILE
(SDFVERSION "OVI 2.1")
(DESIGN "digital_lf")
(VENDOR "cello.slow")
(PROGRAM "Synopsys Design Compiler cmos")
(VERSION "I-2013.12-SP3")
(DIVIDER /)
(VOLTAGE 1.08:1.08:1.08)
(PROCESS "slow")
(TEMPERATURE 110.00:110.00:110.00)
(TIMESCALE 1ns)
(CELL
  (CELLTYPE "digital_lf")
  (INSTANCE)
  (DELAY
    (ABSOLUTE
      (INTERCONNECT U87/Y U90/A (0.000:0.000:0.000))
      (INTERCONNECT U88/Y U89/A (0.000:0.000:0.000))
      (INTERCONNECT U90/Y U88/A (0.000:0.000:0.000))
      (INTERCONNECT U116/SUM U87/A (0.000:0.000:0.000))
      (INTERCONNECT U83/Y U84/A (0.000:0.000:0.000))
      (INTERCONNECT U117/SUM U83/A (0.000:0.000:0.000))
      (INTERCONNECT U81/Y U82/A (0.000:0.000:0.000))
      (INTERCONNECT U84/Y U81/A (0.000:0.000:0.000))
      (INTERCONNECT U75/Y U76/A (0.000:0.000:0.000))
      (INTERCONNECT U82/Y U75/A (0.000:0.000:0.000))
      (INTERCONNECT U69/Y U70/A (0.000:0.000:0.000))
      (INTERCONNECT U76/Y U69/A (0.000:0.000:0.000))
      (INTERCONNECT U67/Y U68/A (0.000:0.000:0.000))
      (INTERCONNECT U58/Y U67/A (0.000:0.000:0.000))
      (INTERCONNECT U65/Y U66/A (0.000:0.000:0.000))
      (INTERCONNECT U60/Y U65/A (0.000:0.000:0.000))
    )
  )
)
```



# Simulate DPLL with Synthesized DLF

- ▶ You can simulate DPLL with SDF file
  - ▶ Check if DPLL with digital timing information operates correctly
- ▶ To simulate DPLL, all you need is ...
  - ▶ DPLL netlist file : **dp11.sv**
  - ▶ Digital loop filter mapped file : **digital\_lf\_mapped.sv**
  - ▶ SDF file of the loop filter : **digital\_lf\_mapped.sdf**
  - ▶ Digital cell library : **~/IDEC\_CBDF/ACL/verilog/**

# Exercise: DPLL w/ SDF Simulation

- ▶ Testbench: 'tb\_locking' ← equivalent to GLISTER testbench
- ▶ First, check if there's digital\_if\_mapped.sv is located at the correct directory

```
$ cd ~/IDEC_CBDF/xmodel_dp11/dp11  
$~/IDEC_CBDF/xmodel_dp11/dp11> ls
```

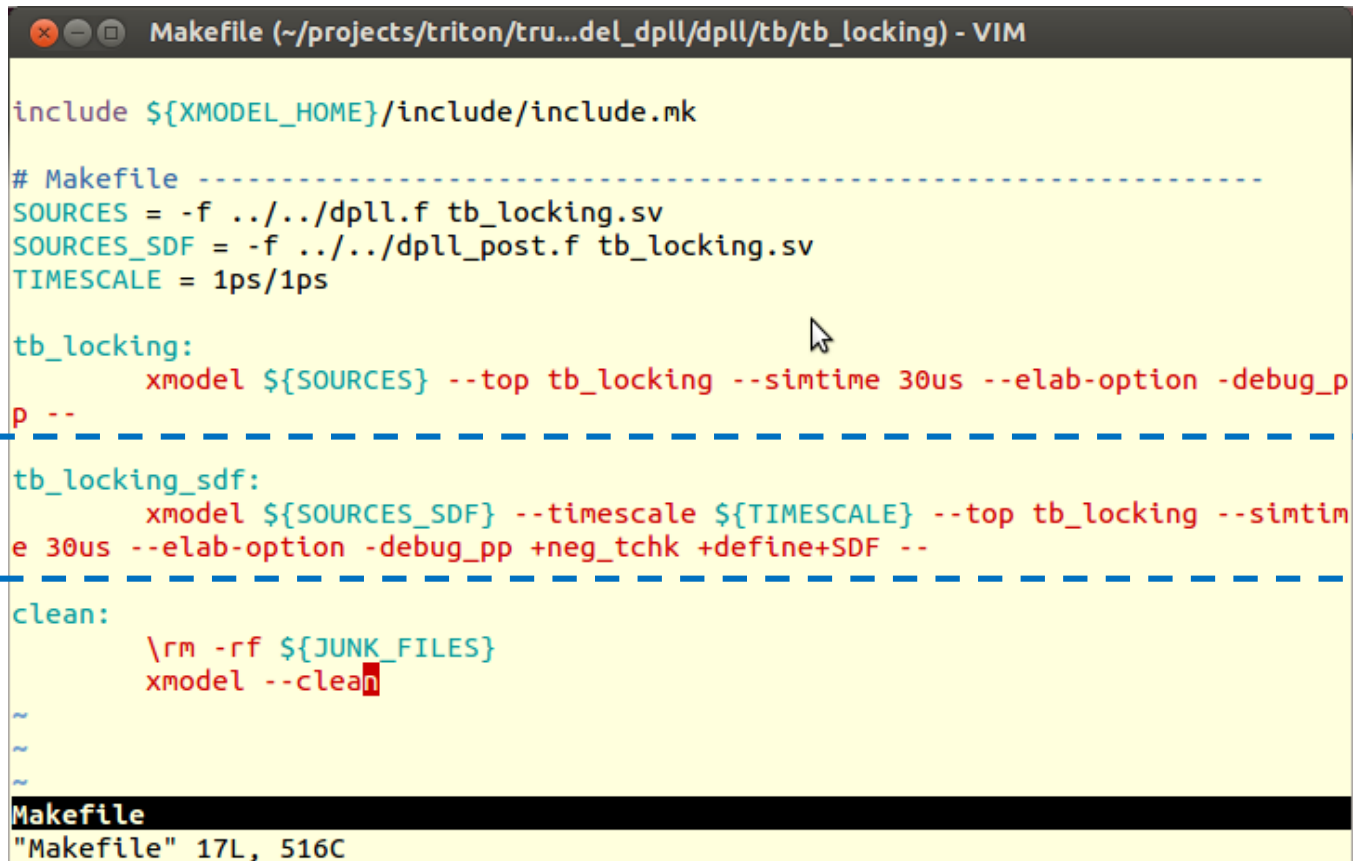
- ▶ Change the directory to 'tb\_locking'

```
$~/IDEC_CBDF/xmodel_dp11/dp11> cd tb/tb_locking
```

# Exercise: DPLL w/ SDF Simulation (2)

## ► Open the Makefile

```
$~/IDEC_CBDF/xmodel_dp11/dp11/tb/tb_locking> vi Makefile
```



```
Makefile (~/projects/triton/tru...del_dp11/dp11/tb/tb_locking) - VIM

include ${XMODEL_HOME}/include/include.mk

# Makefile -----
SOURCES = -f ../../dp11.f tb_locking.sv
SOURCES_SDF = -f ../../dp11_post.f tb_locking.sv
TIMESCALE = 1ps/1ps

tb_locking:
    xmodel ${SOURCES} --top tb_locking --simtime 30us --elab-option -debug_pp --

tb_locking_sdf:
    xmodel ${SOURCES_SDF} --timescale ${TIMESCALE} --top tb_locking --simtime 30us --elab-option -debug_pp +neg_tchk +define+SDF --

clean:
    \rm -rf ${JUNK_FILES}
    xmodel --clean

~
~
~

Makefile
"Makefile" 17L, 516C
```

# Exercise: DPLL w/ SDF Simulation (3)

- ▶ Open the tb\_locking.sv

```
$~/IDEC_CBDF/xmodel_dpll/dpll/tb/tb_locking> vi tb_locking.sv
```

- ▶ **Important**

- ▶ You need to fix sdf path to the right “absolute” path of the target .sdf file (ex. digital\_lf\_mapped.sdf)

```
initial begin
-- `ifdef SDF
    $sdf_annotate("/afs/mics.snu.ac.kr/user/eunseo/IDEC_CBDF/Synthesis/output/
:/digital_lf/digital_lf_mapped.sdf", DUT.digital_lf);
-- $vcdplusfile("tb_locking.sdf.vpd");
`else
    $vcdplusfile("tb_locking.vpd");
`endif
    $vcdpluson;
end
```

# Run the Simulation

- ▶ Run the simulation by typing 'make <mode>'

```
$ make tb_locking_sdf
```

- ▶ (If you type 'make tb\_locking', the simulation result will be the same with what you did in the GLISTER environment)
- ▶ The simulation is now running

# SDF Annotation Checking

- ▶ You need to check the following at your terminal (while in simulating)

```
Thanks for flying Vim
```

```
50 2016  
Doing SDF annotation ..... Done
```

```
 _ |/_/_/_/_/_/_/_/_/_/_/_/_/_/_ (TM)  
_ |/_/_/_/_/_/_/_/_/_/_/_/_/_/_  
/_/_/_/_/_/_/_/_/_/_/_/_/_/_ ANALOG/MIXED-SIGNAL SIMULATOR
```

```
XMODEL Release 2016.0517 (x86_64)  
Copyright (c) 2012-2015 Scientific Analog, Inc.  
All rights reserved and patents pending.
```

```
#LMX: LM-X feature 'XMODEL' has been successfully checked out.  
#LMX: Version: 2.1 (academic)  
#LMX: Expiration: 2016-12-31 23:59  
#LMX: License type: network  
#LMX: License server: arctic
```

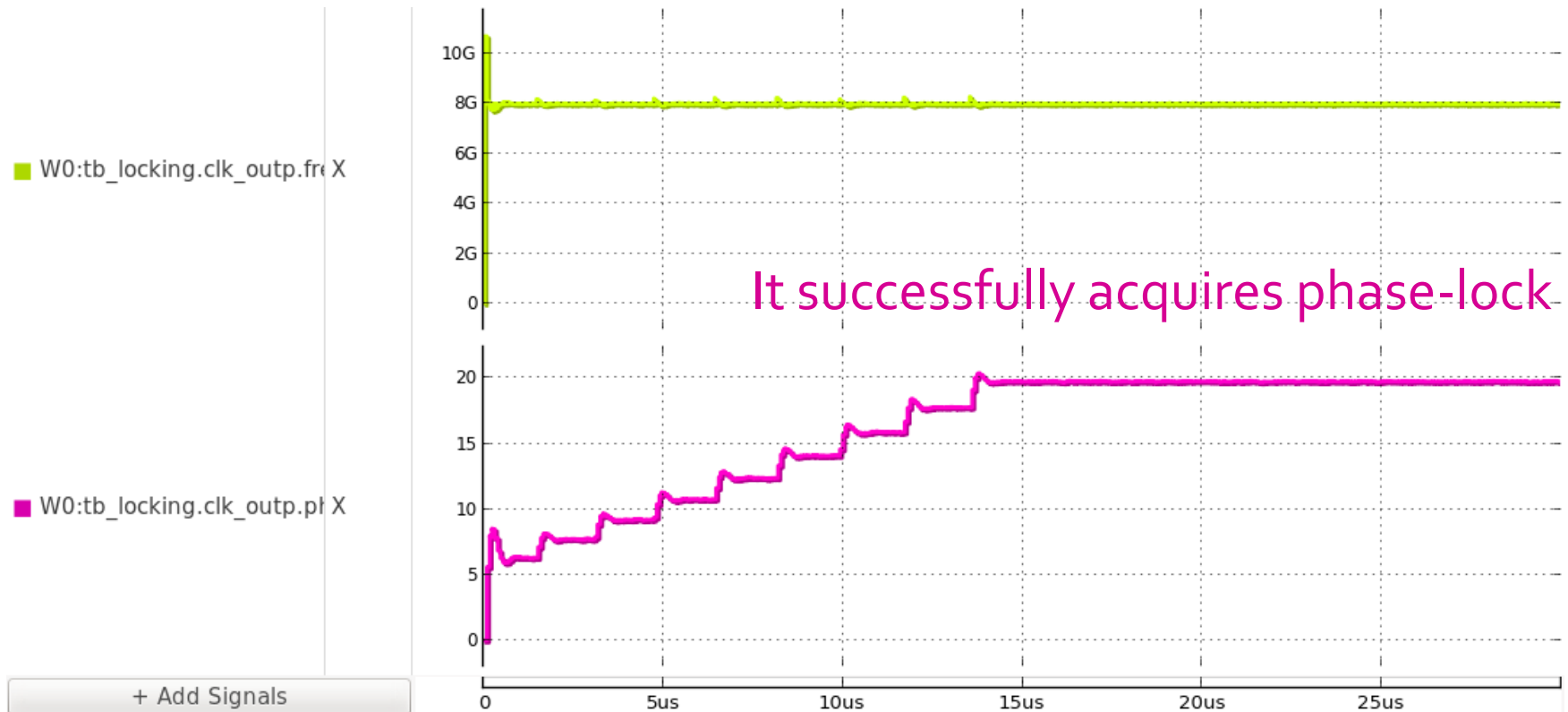
```
NOTICE: this software is licensed for research and educational  
purposes only and not permitted for commercial, income-producing  
activities.
```

```
VCD+ Writer H-2013.06_Full64 Copyright (c) 1991-2013 by Synopsys Inc.
```

# Result - Locking

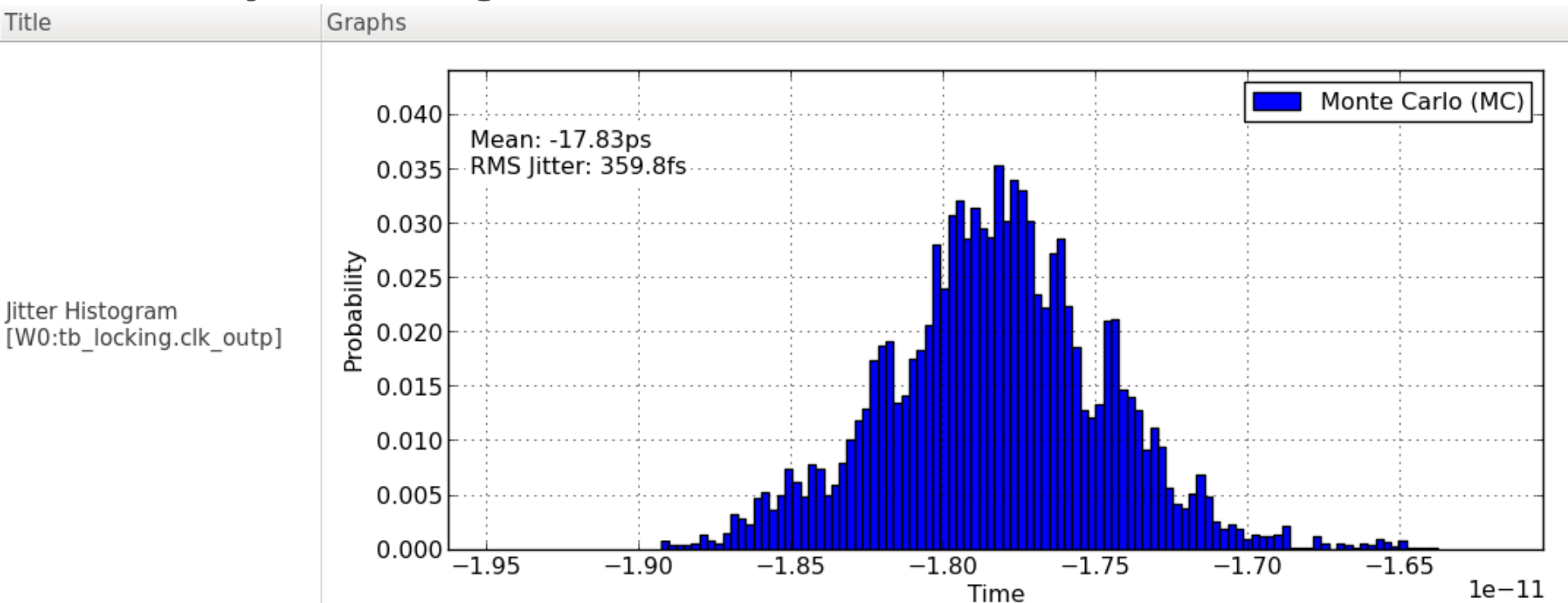
- Open XWAVE waveform viewer

```
$ xwave xmodel.jez
```



# Result – Jitter Histogram

- ▶ You can also plot jitter histogram as you did in the previous DPLL simulation
  - ▶ (analysis page → add analysis → select signal ('clk\_outp') → jitter histogram)





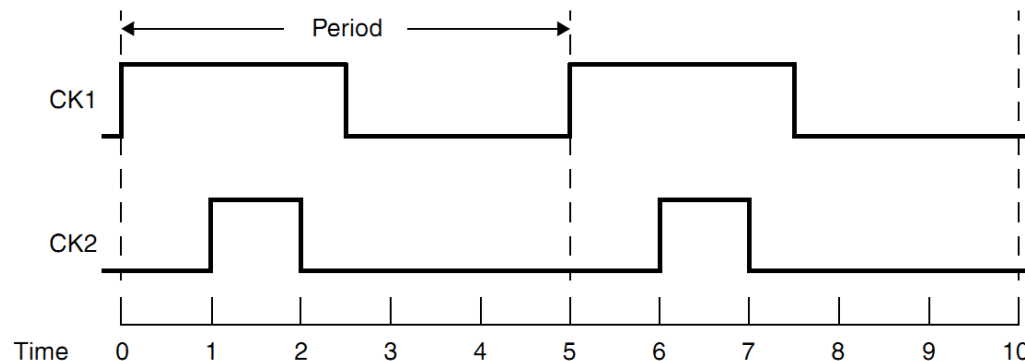
# Reference

- ▶ [solvnnet.synopsys.com](http://solvnnet.synopsys.com) - Documentation
  - ▶ Design Compiler – User guide, Tool commands
  - ▶ Formality – User guide, Tool commands
  - ▶ PrimeTime – User guide, Tool commands

# Appendix

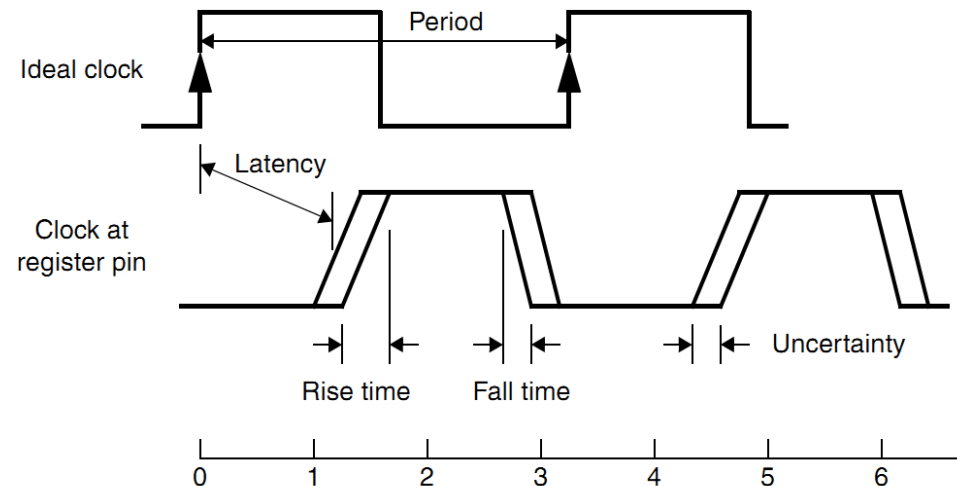
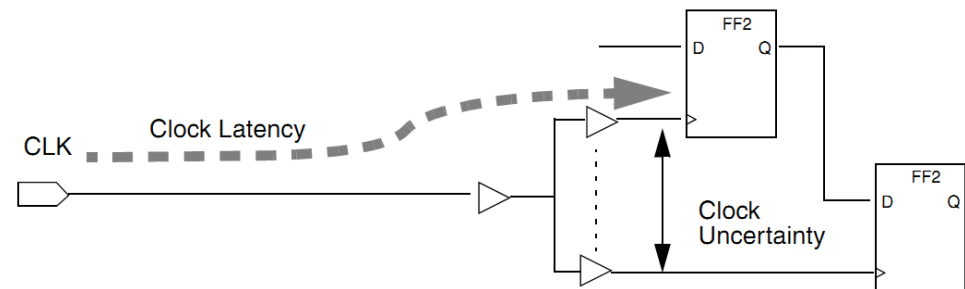
# Commands with Clocks

- ▶ To implement digital circuits in target frequency and to perform timing analysis, you must specify **clock period** used in the design
- ▶ **create\_clock**
  - ▶ *create\_clock "clock\_port" -name {clk\_name} -period {clock\_period} -waveform {edge\_list}*
  - ▶ *create\_clock clk\_in -name CK2 -period 5.0 -waveform {1.0 2.0}*



# Commands with Clocks

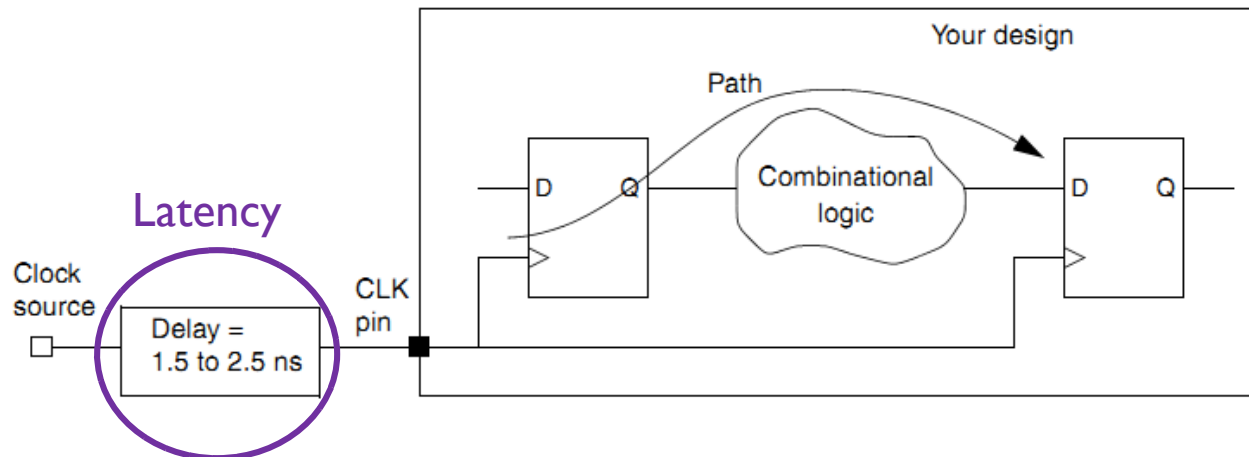
- ▶ For accurate timing analysis, you must describe the characteristics such as **latency**, **uncertainty** and **transition time of clock**, since the clock signal is not ideal!
- ▶ **set\_propagated\_clock**
  - ▶ To propagate clock network delays and automatically determine latency at each register clock pin
  - ▶ If this command is not used, ideal clock(no delay) is assumed
  - ▶ *set\_propagated\_clock [get\_clocks clk\_in]*



# Commands with Clocks

## ► set\_clock\_latency

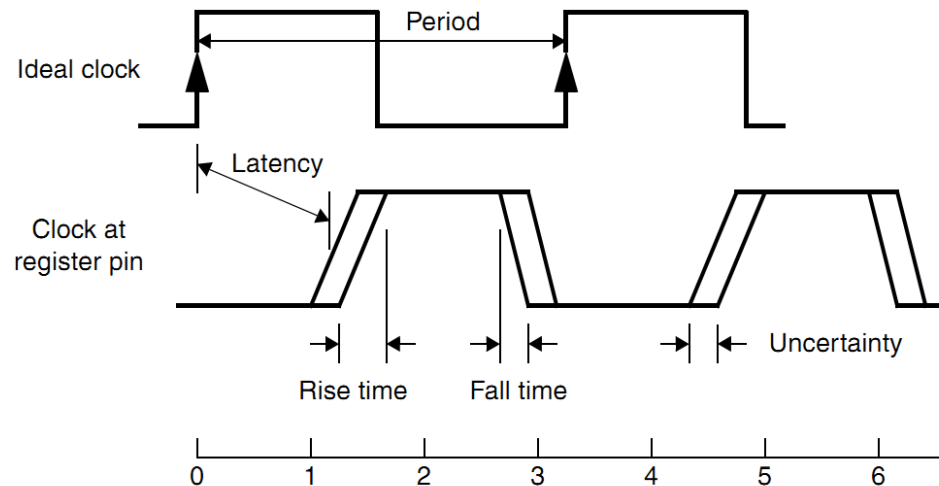
- the amount of time it takes for the clock signal to be propagated from the original clock source
- *set\_clock\_latency 1.5 -source -early [get\_clocks clk\_in]*
- *set\_clock\_latency 2.5 -source -late [get\_clocks clk\_in]*



# Commands with Clocks

## ► set\_clock\_transition

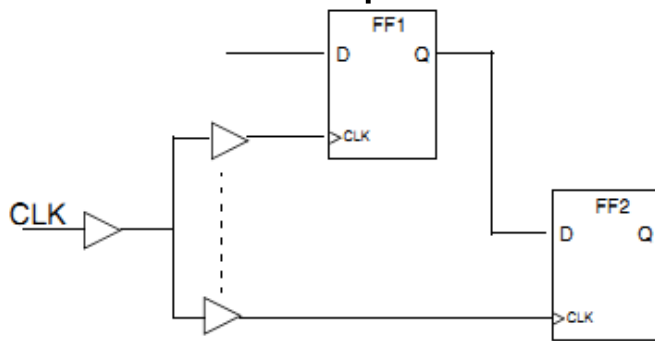
- set the transition time of clock
- *set\_clock\_transition 0.75 -max [get\_clocks clk\_in]*
- *set\_clock\_transition 0.64 -rise [get\_clocks clk\_in]*
- *set\_clock\_transition 0.52 -fall [get\_clocks clk\_in]*



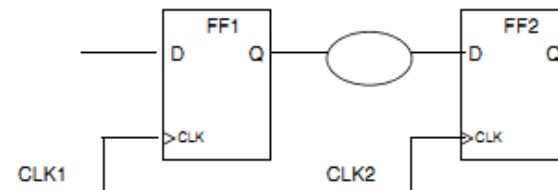
# Commands with Clocks

## ► set\_clock\_uncertainty

- In performing a setup or hold time violation check, the tool adjusts the timing check according to the **worst possible difference in clock edge times**. You can use *set\_clock\_uncertainty* to model clock network skew
- *set\_clock\_uncertainty -setup 0.03 [get\_clocks clk\_in]*
- *set\_clock\_uncertainty -hold 0.10 [get\_clocks clk\_in]*
- Include both simple clock skew and inter-clock skew



Simple clock uncertainty



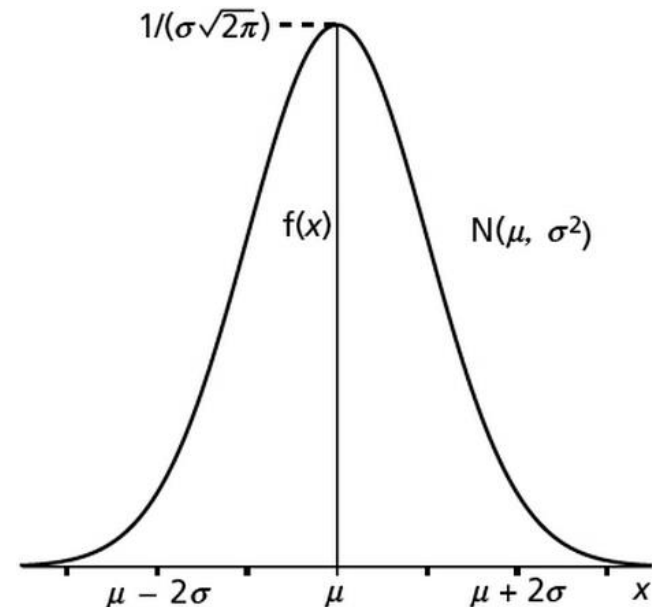
Inter-clock uncertainty

# Setting Reasonable Clock Uncertainty

- Generally, clock jitter form is **Gaussian distribution**

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where  $X$  is a normal random variable,  $\mu$  is the mean,  $\sigma$  is the standard deviation,





# Setting Reasonable Clock Uncertainty

## ► Criteria of clock uncertainty

- Requirement : Bit-error-rate(BER)  $< 10^{-12}$
- $\text{BER} < 10^{-12} \rightarrow (x > -7\sigma) \ \& \ (7\sigma < x)$
- Criteria : **14 $\sigma$**  uncertainty to achieve  $10^{-12}$  Bit-error-rate

## ► Setting step

- First, find **rms jitter(UI or %)** of clock source
- Second, find **rms jitter time** from the result of first step
- Third, multiply **rms jitter time** by **14**

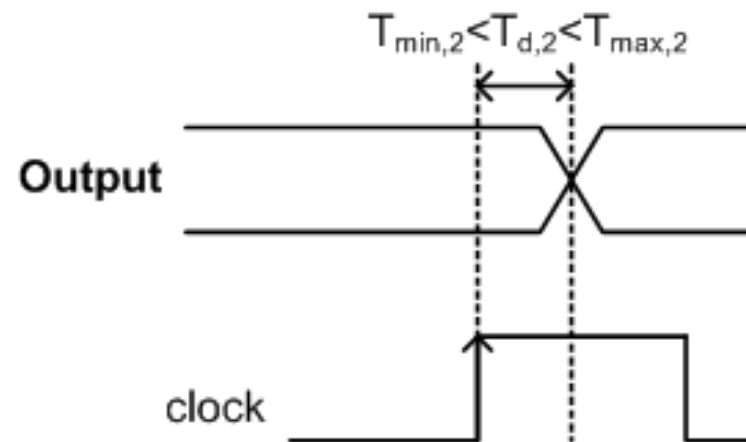
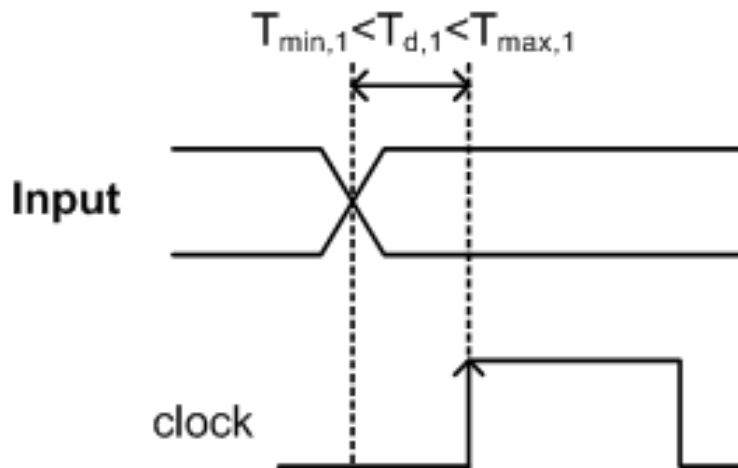
# Min/Max Delay Constraints in I/O

## ► Input

- *set\_min\_delay 1.0 -from INPUT -to [get\_clocks \*]*  
*set\_max\_delay 2.5 -from INPUT -to [get\_clocks \*]*

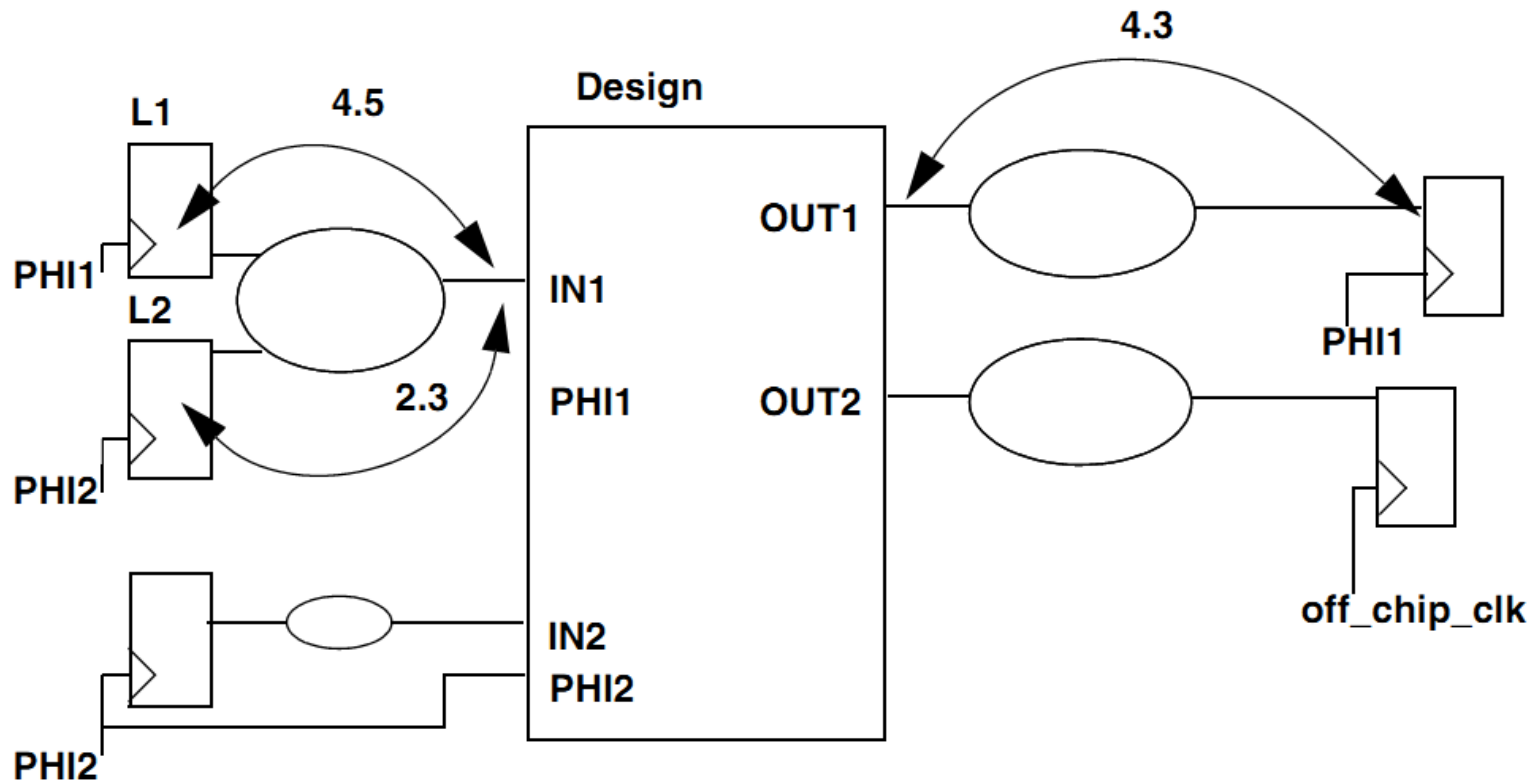
## ► Output

- *set\_min\_delay 1.0 -from [get\_clocks \*] -to OUTPUT*  
*set\_max\_delay 2.5 -from [get\_clocks \*] -to OUTPUT*



# Modeling delay due to external paths

## ► Example



# Modeling delay due to external paths

## ► **set\_input\_delay**

- Specifies the timing of external paths leading to an input port
- *set\_input\_delay 4.5 -clock PHI1 {IN1}*
- *set\_input\_delay 2.3 -clock PHI2 -add\_delay {IN1}*
  - if paths from multiple clocks or edges reach the same port, specify each one using the *-add\_delay* option

## ► **set\_output\_delay**

- Specifies output delays that represent an external timing paths from an output port to register
- *set\_output\_delay 4.3 -clock PHI1 {OUT1}*
- *create\_clock -period 10 -waveform {3 8} -name off\_chip\_clk*  
*set\_output\_delay 2.5 -clock off\_chip\_clk {OUT2}*

# Modeling specific external driver of I/O

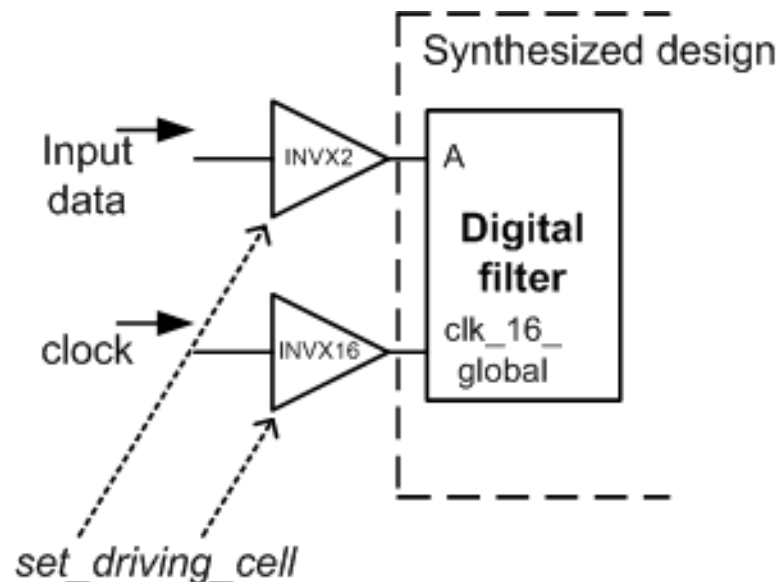
## ► **set\_driving\_cell**

- “Models” the external driver at an input port of the current design as the output of a library cell
- This command takes into account the design rule constraints associated with the specified driving cell, in addition to the drive information used to compute the transition time of the input net
- When you specify a driving cell, the design rule constraints are annotated on the input port of the block being synthesized

# Modeling specific external driver of I/O

## ► Example

- *set\_driving\_cell -lib\_cell INV\_X16 -library cello clk\_16\_global*  
*set\_driving\_cell -lib\_cell INV\_X2 -library cello [all\_inputs]*



# Modeling capacitance load

## ► **set\_load**

- To accurately perform timing analysis, the tool needs information about the external load capacitance of nets connected to ports, including pin capacitance and wire capacitance
- *set\_load 0.100 [all\_outputs]*
- *set\_load -pin\_load 3.5 {IN1 OUT1 OUT2}*  
*set\_load -wire\_load 5.0 {OUT3}*

# Fix Hold Time Violation

## ► `set_fix_hold`

- This command sets the hold time fix attribute on specified clocks, which directs the tool to check for and fix hold violation during the compile operation
- If this command does not exist, Design Compiler **do not fix hold time violation!**
- Delayed clock is generated by this command
- `set_clock_uncertainty -hold 0.50`  
`set_fix_hold clk_global`



# Some Other Commands

## ▶ **set\_max\_capacitance**

- ▶ This command sets the maximum value of capacitance attribute to a specified value on the specified clocks, ports or designs.
- ▶ *set\_max\_capacitance 2.0 [current\_design]*

## ▶ **set\_max\_transition**

- ▶ This command sets the maximum transition time, a design rule constraint, on specified clocks, ports, or designs.
- ▶ *set\_max\_transition 0.25 [current\_design]*