

BASICS for ASIC's ISTEAM TRAINING MODULE

Presented by: Gayatri M Srinivasan, 07.30.2012

Accelerate.



Agenda

- Module 1: Review of Digital Design Fundamentals
 - Semiconductor Physics
 - Timing Paths
- Module 2: Overview of Technology and Libraries
- Module 3a: ASIC Flow
- Module 3b: STA Basics
- Module 4: Constraints & its Analysis
- Module 5: Using Primetime for STA

Review of Digital Design Fundamentals

Part1 : Semiconductor Device Physics



Accelerate.

Review of Digital Design Fundamentals

- Digital Design Definition
- Components of a Design
- Electrical Behavior of CMOS Circuits
- Crosstalk

Digital Designs/Systems

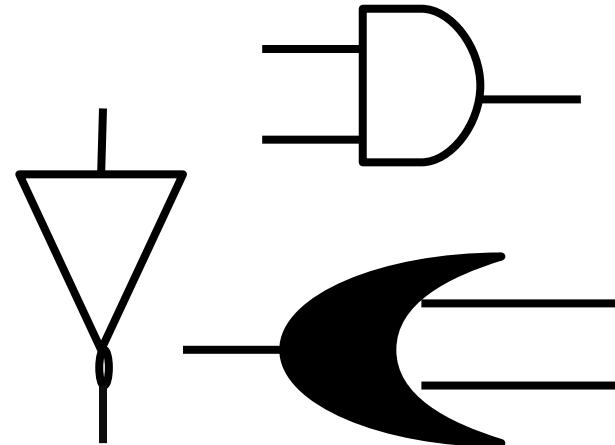
- Analog Devices process time varying signals across a continuous range of voltage, current etc.
- Digital Circuits do the same but we can pretend they don't!
 - Modeled as taking on, at anytime only one of two discrete values
 - HIGH [LOGIC 1]
 - LOW [LOGIC 0]
 - Noise Margin is more in digital



Why are we moving to Digital Systems?

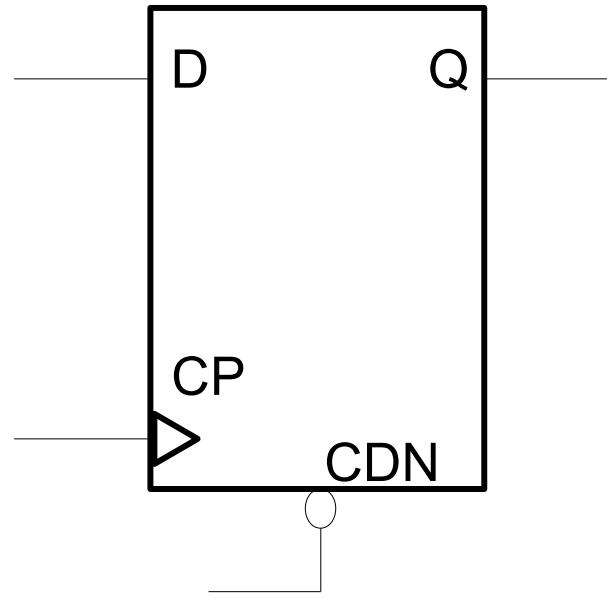
Components of A Digital Circuit - Gates

- Gates
 - Got their name from their function of allowing the flow of digital information
 - Called a combinational circuit because its outputs depend only on current input combination
 - What are the 3 basic gates?
 - AND
 - OR
 - NOT



Components of a Digital Circuit – Flip-Flops

- Flip-Flops
 - A device that stores 1 or 0
 - State of a flip-flop is the value it stores
 - Can be changed only at certain times determined by a “clock” input
 - Can be built by a collection of gates
 - A design with flops is called Sequential circuit.
Why?
 - o/p at any time depends not only on current input but also the past sequence of inputs applied [memory]



What are the components of a single Silicon Chip?

Other elements in Digital Systems

- Memories
 - RAM's
 - ROM's
- Latches
- Analog circuits
 - SerDes
 - PLL's

Electrical Behavior of CMOS Circuits

- Static Behaviors : Input and Output Signals are not changing
- Dynamic Behaviors: Inputs and Outputs are changing
- Static & Dynamic :
 - Logic voltage Level
 - DC Noise Margins
 - Fan-Out
 - Speed
 - Power Consumption
 - Noise
 - Electrostatic Discharge
 - Open Drain outputs
 - 3 state outputs

A Typical Data Sheet

- Electrical Characteristics of CMOS circuits will help you understand salient points of any datasheet in the LSI library.
- Help understand customer requirements and create robust ASIC's.

DC ELECTRICAL CHARACTERISTICS OVER OPERATING RANGE							
The following conditions apply unless otherwise specified: Commercial: $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$, $V_{CC} = 5.0\text{V}\pm5\%$; Military: $T_A = -55^\circ\text{C}$ to $+125^\circ\text{C}$, $V_{CC} = 5.0\text{V}\pm10\%$							
Sym.	Parameter	Test Conditions ⁽¹⁾		Min.	Typ. ⁽²⁾	Max.	Unit
V_{IH}	Input HIGH level	Guaranteed logic HIGH level		3.15	—	—	V
V_{IL}	Input LOW level	Guaranteed logic LOW level		—	—	1.35	V
I_{IH}	Input HIGH current	$V_{CC} = \text{Max.}$, $V_I = V_{CC}$		—	—	1	μA
I_{IL}	Input LOW current	$V_{CC} = \text{Max.}$, $V_I = 0\text{ V}$		—	—	-1	μA
V_{IK}	Clamp diode voltage	$V_{CC} = \text{Min.}$, $I_N = -18\text{ mA}$		—	-0.7	-1.2	V
I_{QOS}	Short-circuit current	$V_{CC} = \text{Max.}$, ⁽³⁾ $V_O = \text{GND}$		—	—	-35	mA
V_{OH}	Output HIGH voltage	$V_{CC} = \text{Min.}$,	$I_{OH} = -20\text{ }\mu\text{A}$	4.4	4.499	—	V
		$V_{IN} = V_{IL}$	$I_{OH} = -4\text{ mA}$	3.84	4.3	—	V
V_{OL}	Output LOW voltage	$V_{CC} = \text{Min.}$	$I_{OL} = 20\text{ }\mu\text{A}$	—	.001	0.1	V
		$V_{IN} = V_{IH}$	$I_{OL} = 4\text{ mA}$	—	0.17	0.33	
I_{CC}	Quiescent power supply current	$V_{CC} = \text{Max.}$ $V_{IN} = \text{GND or } V_{CC}$, $I_O = 0$		—	2	10	μA

SWITCHING CHARACTERISTICS OVER OPERATING RANGE, $C_L = 50\text{ pF}$

Sym.	Parameter ⁽⁴⁾	Test Conditions	Min.	Typ.	Max.	Unit
t_{PD}	Propagation delay	A or B to Y	—	9	19	ns
C_I	Input capacitance	$V_{IN} = 0\text{ V}$	—	3	10	pF
C_{pd}	Power dissipation capacitance per gate	No load	—	22	—	pF

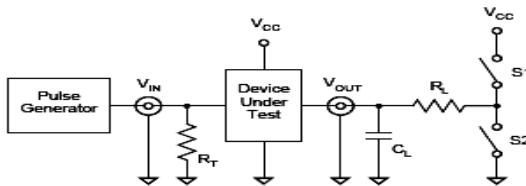
NOTES:

- For conditions shown as Max. or Min., use appropriate value specified under Electrical Characteristics.
- Typical values are at $V_{CC} = 5.0\text{ V}$, $+25^\circ\text{C}$ ambient.

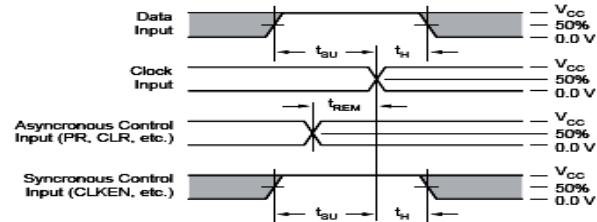
Courtesy : Digital Design Principles and Practices by J. Wakerly

Datasheet (Continued)

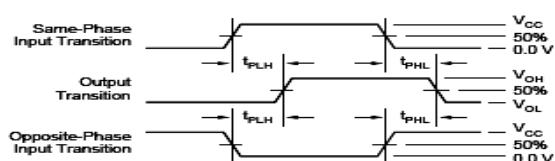
TEST CIRCUIT FOR ALL OUTPUTS



SETUP, HOLD, AND RELEASE TIMES



PROPAGATION DELAY



LOADING

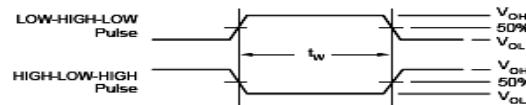
Parameter	R_L	C_L	S_1	S_2
t_{en}	1 k Ω	50 pF or 150 pF	Open	Closed
			Closed	Open
t_{dis}	1 k Ω		Open	Closed
			Closed	Open
t_{pd}	—	50 pF or 150 pF	Open	Open

DEFINITIONS:

C_L = Load capacitance, includes jig and probe capacitance.

R_T = Termination resistance, should equal Z_{out} of the Pulse Generator.

PULSE WIDTH



THREE-STATE ENABLE AND DISABLE TIMES

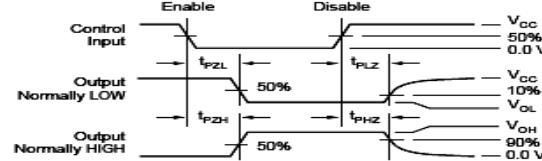


Figure 3-24 Test circuits and waveforms for HC-series logic.

Courtesy : Digital Design Principles and Practices by J. Wakerly

Datasheet @LSI

- For Standard cell Libraries & Documentation of the Flow
 - Type in ASKK in your browser
<http://askk>
 - http://marketing.teams.lsi.com/sites/vertmktg/t_docs.html
- For IO libraries :
<http://web.lsi.com/io/>

Engineering Technical Documentation Website

This website contains technical documentation for CSG products including library databooks, tool documentation, application notes and other documents for cell design.

The left-hand navigation frame contains a number of folders. Click on each folder to reveal more detail on its contents.

Choose the correct folder based on the type of documentation you wish to view:

- Customer Released Documentation
- Engineering Documents
- Internal Customer Documentation

Last updated on: 2012-07-21 00:00:00

Search is NOT specific to this website.

Search

CSG Technical Documentation Website

This website contains technical documentation for CSG products including library databooks, tool documentation, application notes and other documents for cell design.

The left-hand navigation frame contains a number of folders. Click on each folder to reveal more detail on its contents.

Choose the correct folder based on the type of documentation you wish to view:

- Customer Released Documentation
- Engineering Documents
- Internal Customer Documentation

This folder contains the current version of cell-based ASIC and RapidChip Platform ASIC documentation that has been released to LSI customers. The A

The RapidChip Online Help System contains RapidChip documentation.

This folder contains documents written by members of CSG engineering organizations. The documentation in this folder is not productionized and is provided to area to which the document pertains.

This folder contains in-progress versions of external customer documentation and documents for use only by LSI employees and design partners. The versions are in-progress snapshots and have not been through a complete editorial or production process. The other documents in this folder may be in-progress design partners only. Do not distribute any documents in this folder to customers.

CMOS Static Electrical Behavior – Logic Levels and Noise Margins

- Complete Input/Output Characteristic of an INVERTER

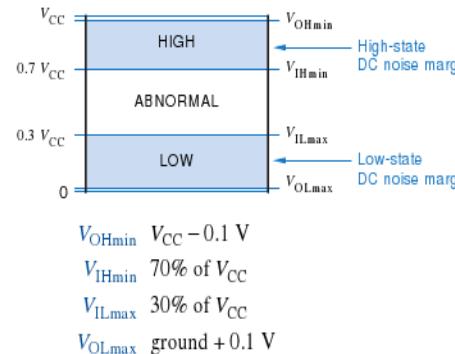
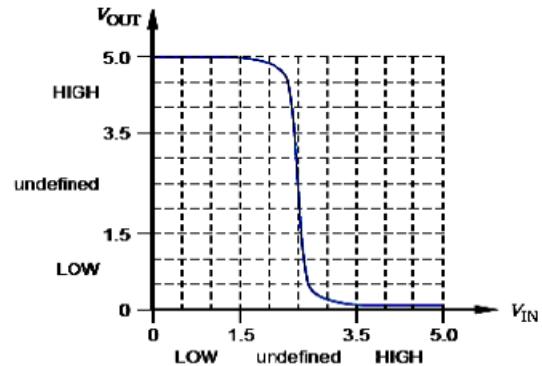


Figure 3-26
Logic levels and
noise margins
for the HC-series
CMOS logic family.

Figure 3-25
Typical input-output
transfer characteristic
of a CMOS inverter.



Courtesy : Digital Design Principles and Practices by J. Wakerly

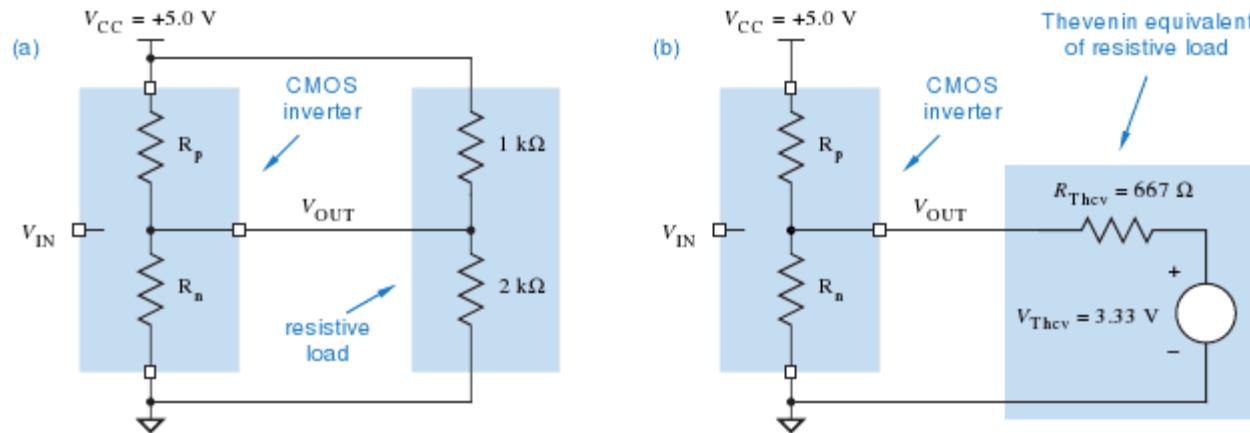
CMOS Static Electrical Behavior – DC Noise Margins

- A measure of how much noise it takes to corrupt a worst-case output voltage into a value that may not be recognized properly by an input
- Regardless of the voltage applied to the input, the input consumes very little current (regardless L or H)
 - This is just the leakage of the gates the inputs are tied to.
 - – I_{IIH} max. current flows into the input at H state
 - – I_{IIL}: max current flows into the input at L state

Very little power to maintain a CMOS input in one state or other

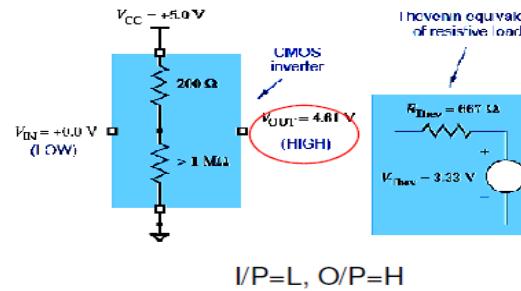
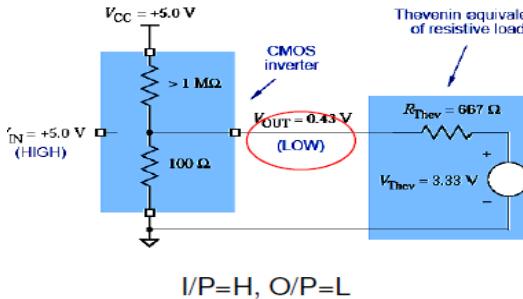
CMOS Static Electrical Behavior – With resistive Loads

- Devices like Xmission Lines LED Loads, TTL Loads.
 - Behavior is Different than normal
 - LOW State is higher than 0.1V
 - HIGH State is lower than 4.4V



Courtesy : Digital Design Principles and Practices by J. Wakerly

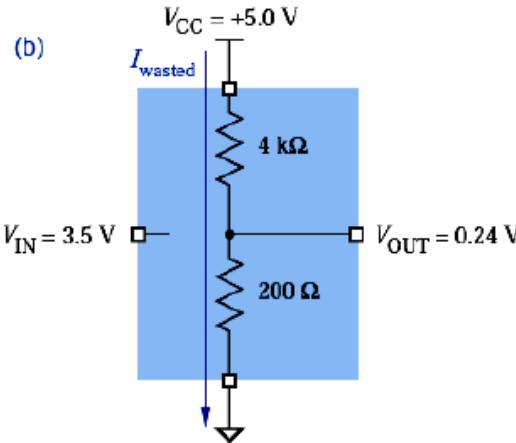
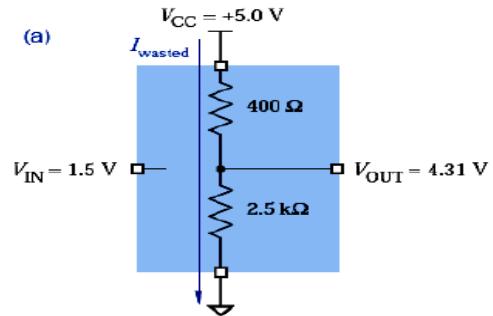
CMOS Static Electrical Behavior – With resistive Loads



- When INV o/p = 0,
 - N channel Resistor is on.
 - $V_{out} = 3.3V \left(\frac{100}{100+667} \right) = 0.41V$
 - In Reality, you don't need to calculate this.
 - For each State:
 - A maximum Load for the output will be specified
 - Worst case voltage will be guaranteed
- When INV o/p = 1,
 - P channel Resistor is on.
 - $V_{out} = 3.3 + (5-3.3) \left(\frac{667}{(200+667)} \right) = 4.61V$

CMOS Static Electrical Behavior – Non Ideal Inputs

- When the input voltage is not equal to Vdd or Vss
 - We will be wasting Power
 - Logic may be undefined
 - Especially when resistive loads



Courtesy : Digital Design Principles and Practices by J. Wakerly

CMOS Static Electrical Behavior – Fanout

- Fanout of a logic gate is the number of inputs that gate can drive without exceeding its worst case load specifications
 - Depends on the characteristics of
 - The Gate Output
 - Input it is driving
 - Effect of Loading
 - Propagation Delay to the output will increase beyond specification
 - Output Rise and Fall times may increase beyond specification
 - Operating Temperature of the device may increase and thus reducing the reliability of the device.

Click icon to add picture

CMOS Static Electrical Behavior – Unused Inputs

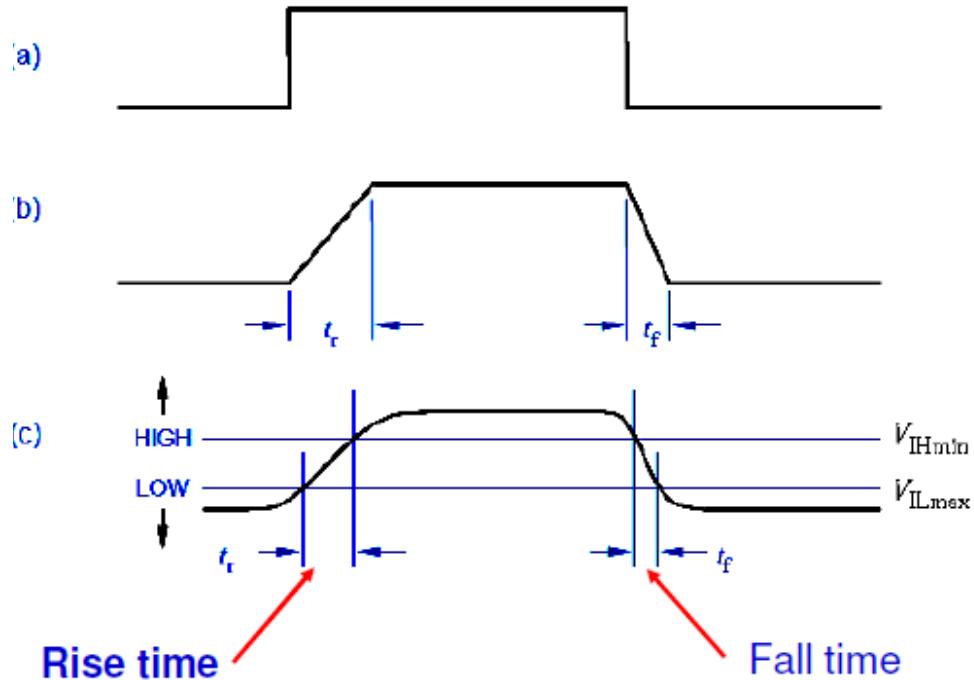
- Unused Inputs need to be tied to a constant value.
- Floating Inputs cause erratic behavior depending on noise and other conditions on the chip

CMOS Dynamic Electrical Behavior

- Speed and Power Usage depend on the device AC characteristics and its load
 - What happens when the data transitions from one state to another
- Speed
 - Transition Time
 - Propagation Delay
- Power Consumption
 - Static (quiescent)
 - Dynamic

Transition Time

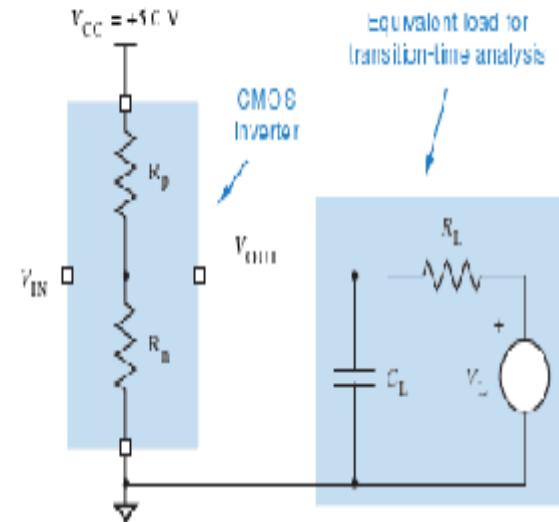
- Amount of time that the output takes to change from 1 state to another is called transition time
- Rise and Fall times depends on Resistance and Capacitance
 - Fig a: ideal transition
 - Fig b: A realistic approximation
 - Fig c: Actual Timing showing rise and fall times



Transition Time – Equivalent Circuit

- 3 components to the load
- $R(I)$, $V(I)$: Represent the DC Load and does not contribute to transition time
 - Determines the voltage and current when the output settles.
- $C(I)$: Represents the AC load and determines
 - the current and voltage when the data transitions
 - How long it takes to transition

Figure 3-37
Equivalent circuit for analyzing transition times of a CMOS output.



Transition Time –Example (High to Low)

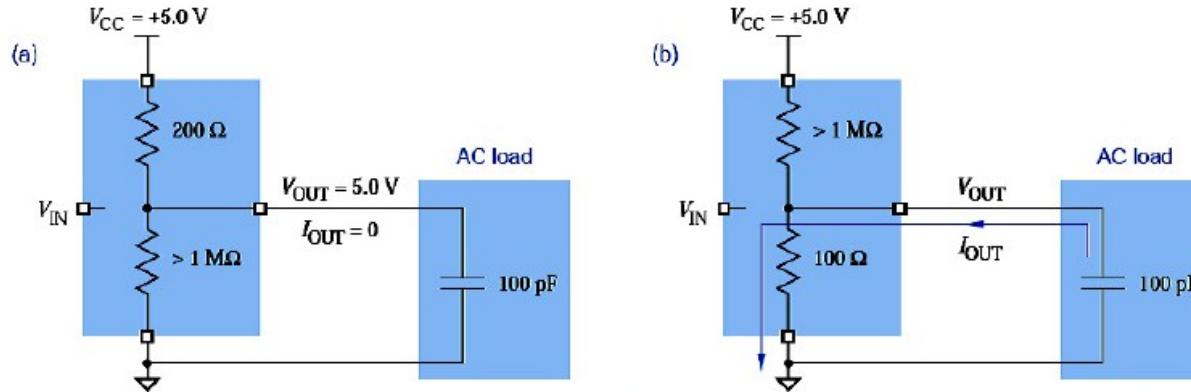
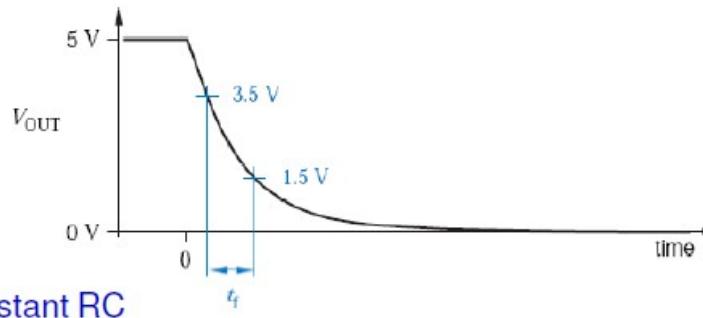


Figure 3-38 Model of a CMOS HIGH-to-LOW transition: (a) in the HIGH state; (b) after *p*-channel transistor turns off and *n*-channel transistor turns on.

$$\begin{aligned}V_{OUT} &= V_{DD} \cdot e^{-t/R_n C_L} \\&= 5.0 \cdot e^{-t(100 \cdot 100 \cdot 10^{-12})} \\&= 5.0 \cdot e^{-t/(10 \cdot 10^{-9})} \text{ V}\end{aligned}$$

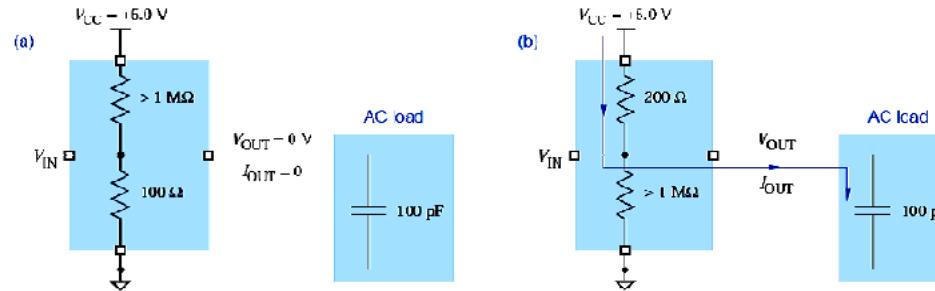
Time constant RC



Courtesy : Digital Design Principles and Practices by J. Wakerly

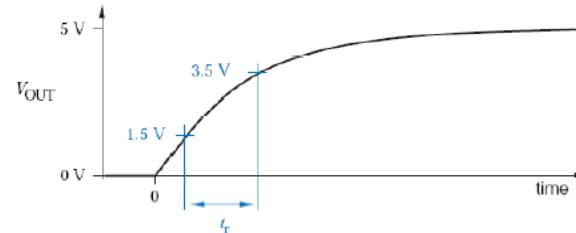
Transition Time – Example Low to High

Figure 3-40 Model of a CMOS LOW-to-HIGH transition: (a) in the LOW state; (b) after *n*-channel transistor turns off and *p*-channel transistor turns on.



$$\begin{aligned}V_{OUT} &= V_{DD} \cdot (1 - e^{-t/R_p C_L}) \\&= 5.0 \cdot (1 - e^{-t/(200 \cdot 100 \cdot 10^{-12})}) \\&= 5.0 \cdot (1 - e^{-t/(20 \cdot 10^{-9})}) \text{ V}\end{aligned}$$

Time constant RC



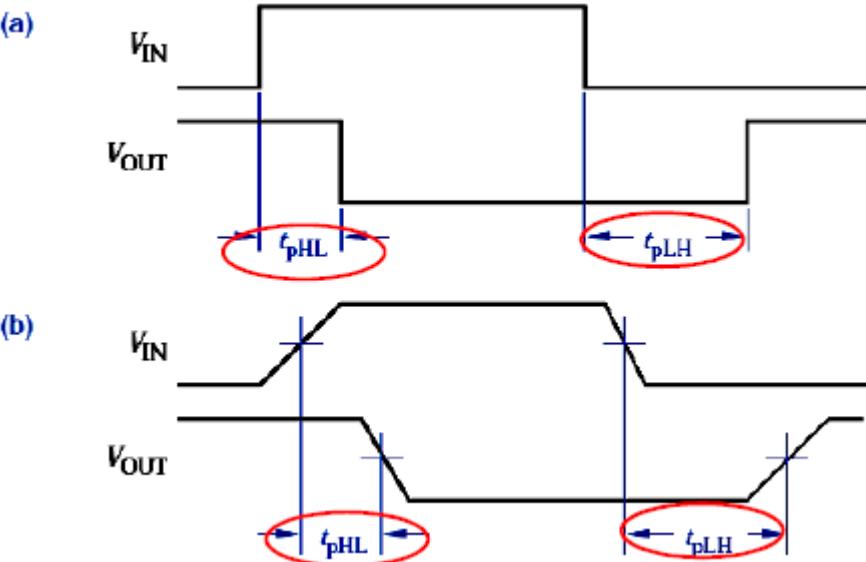
Transition Time -Summary

- Increase in load capacitance causes an increase in RC time constant and a corresponding increase in transition time of the output
 - So for high speed circuits you need to reduce the number of inputs an output drives
 - Can be done either by replicating the signal
 - Or careful layout of the design
- In real design, you don't calculate transition time manually. Tools do that
 - Rule of thumb RC constant equals transition time.
 - In each technology, how much fanout and distance a buffer can drive needs to be understood.

Click icon to add picture

Propagation Delay

- A signal path is the electrical path from a particular input signal to a particular output signal of a logic element
- The propagation Delay of a signal path is the amount of time it takes for a change in input signal to produce a change in output signal



Propagation Delay

- The rate at which transistors change is influenced by
 - Semiconductor Physics of the device
 - What type of gate is being used
 - Does it need several transistors to change state or just one
 - With rise and fall times it takes a while to come to a steady state [learnt earlier]
 - By the circuit environment
 - Input Transition Rate
 - Input Capacitance
 - Output Loading

Power Dissipation[Consumption]

- Static power
 - Power dissipated when output is not changing is called static power or quiescent power.
 - In 40 nm and below this power [also called leakage] is a concern
 - In applications like laptops and smart phones manufacturers try to bring it to zero by using various techniques
- Dynamic Power
 - Power dissipated during transitions is dynamic power. It is a function of frequency and the capacitance which dissipates power

$$P_d = P_t + P_l$$

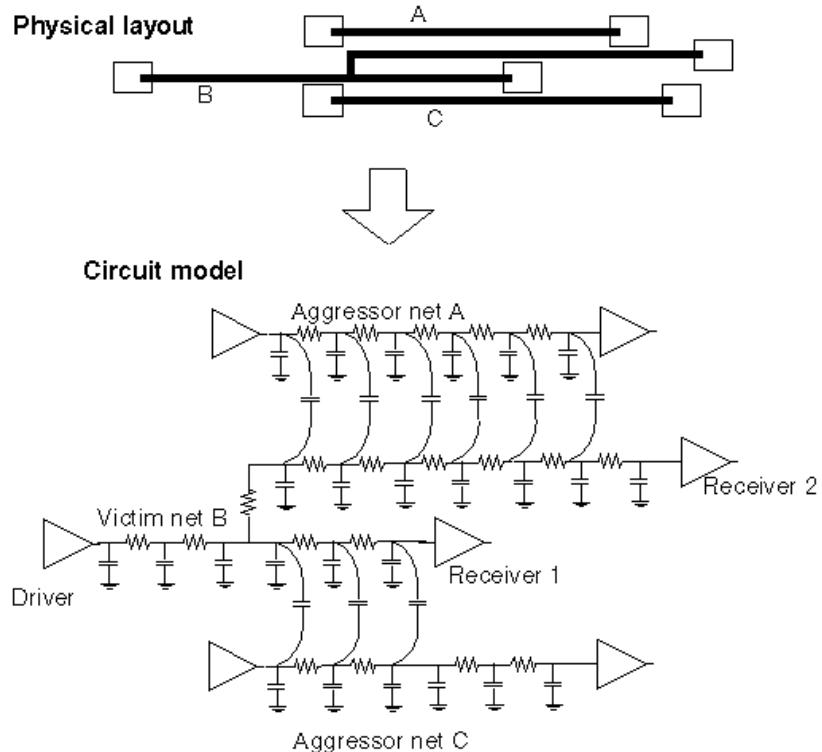
$$\begin{aligned} &= C_{pd}.V_{cc}^2.f + C_l.V_{cc}^2.f \\ &= (C_{pd} + C_l).V_{cc}^2.f \end{aligned}$$

C_l = LoadCapaci tan ce

C_{pd} =DissipationCapaci tan ce

Crosstalk Effects

- In electronics, **crosstalk** is any phenomenon by which a signal transmitted on one circuit or channel of a transmission system creates an undesired effect in another circuit or channel
 - Caused by undesired capacitive, inductive, or conductive coupling from one circuit, part of a circuit, or channel, to another
- IC design, crosstalk normally refers to a signal affecting another nearby signal
 - Coupling is capacitive, and to the nearest neighbor
 - Other forms of coupling and effects on signal further away are sometimes important
 - Substrate coupling conveyed through the integrated circuit substrate

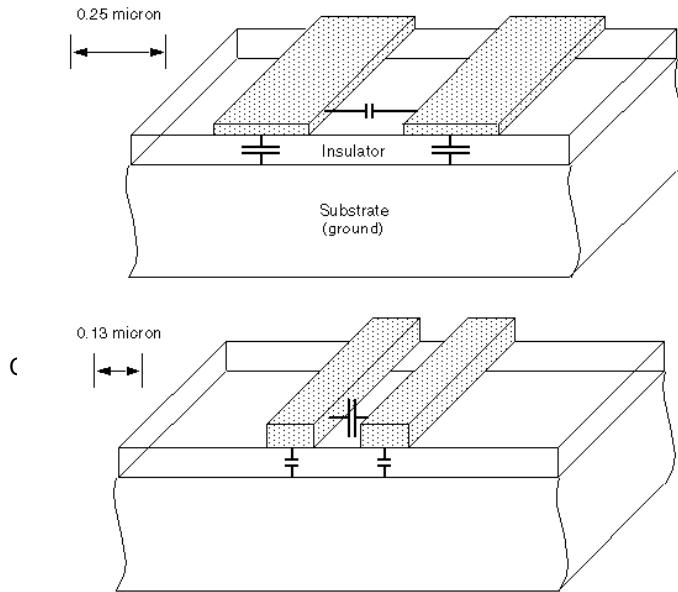


Signal Integrity & Crosstalk Effects

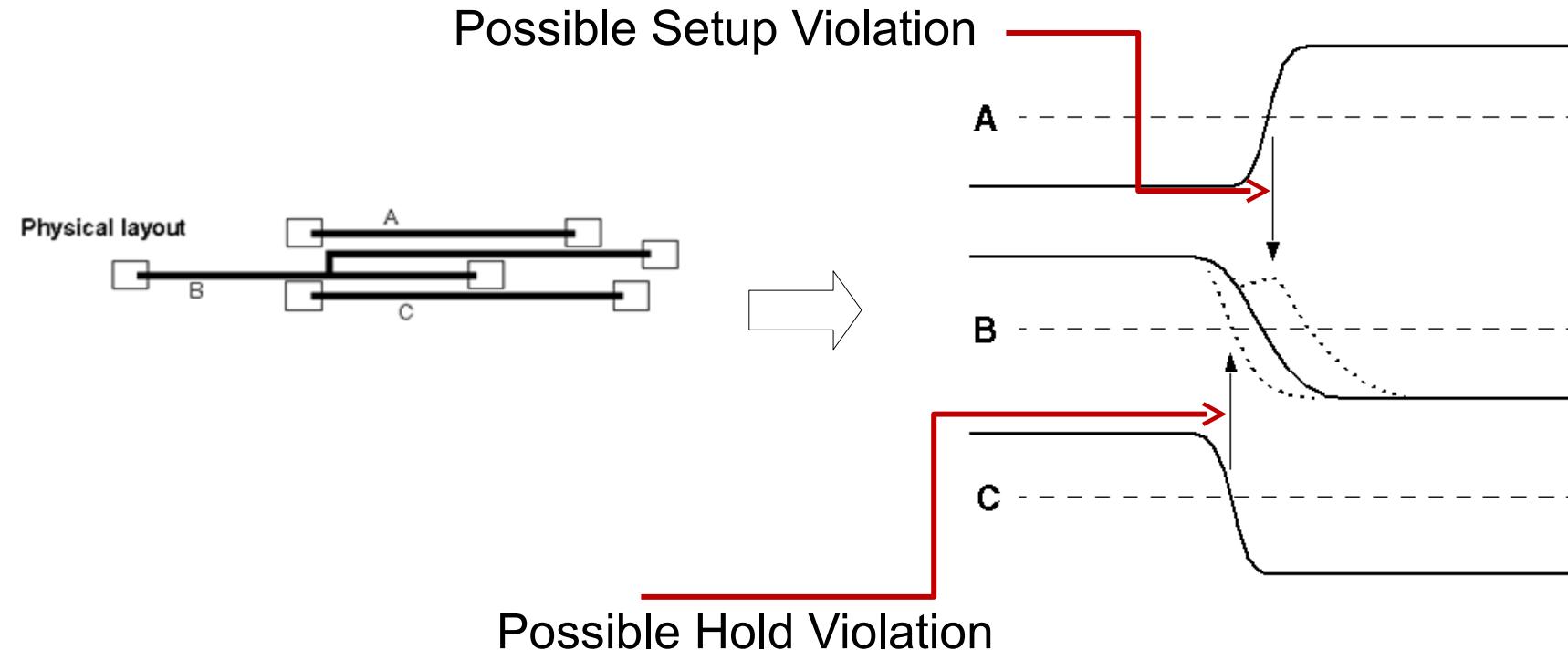
- Signal Integrity is the ability of an electrical signal to carry information reliably and resist the effects of high-frequency electromagnetic interference from nearby signal crosstalk
- Signal integrity can be degraded by the electrical interaction between two or more physically adjacent nets due to capacitive cross-coupling
- Degradation of Signal Integrity
 - Due to capacitive coupling
 - More pronounced in smaller geometries
 - Crosstalk effects become increasingly important compared to cell delays and net delays

Reasons for poor Signal Integrity

- As circuit geometries become smaller
 - wire interconnections become closer together and taller
 - This increases the cross-coupling capacitance between nets
 - At the same time, parasitic capacitance to the substrate becomes less
 - interconnections become narrower
 - cell delays are reduced as transistors become smaller

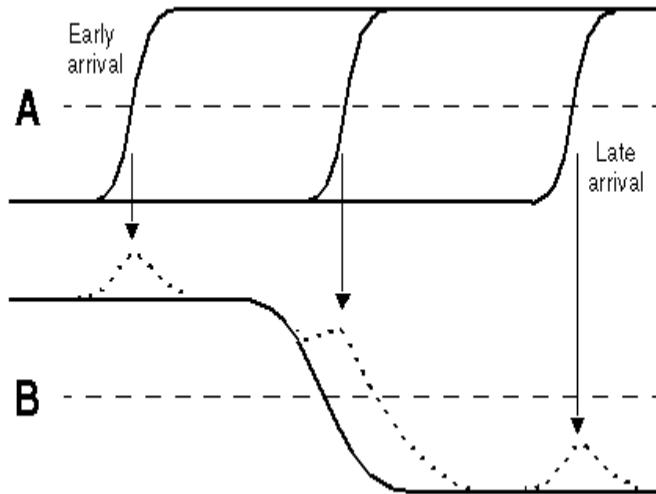


Effects of Crosstalk : *Transition Slowdown or Speedup*



Crosstalk Effects: *Different Arrival Times*

- If the transition on A occurs at about the same time as the transition on B
 - it could cause the transition on B to occur later as shown in the figure, possibly contributing to a setup violation.
- Crosstalk timing effects can occur only when the victim and aggressor timing windows overlap.



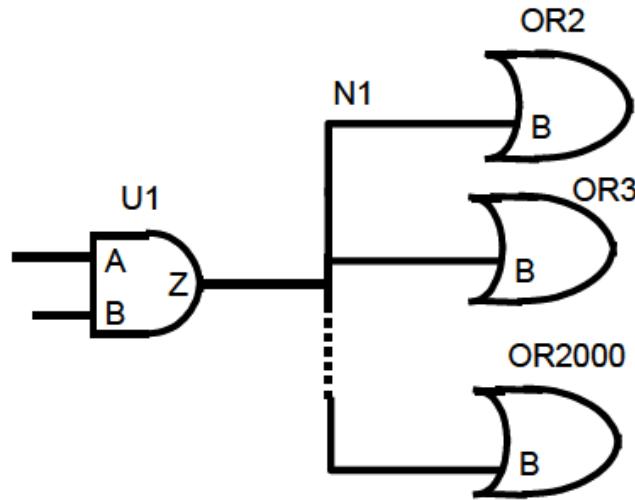
The range of switching times from earliest to latest is called a **TIMING WINDOW**

Review : QUIZ1

- You learnt that Unused Inputs are not allowed. You see a 2 Input NAND Gate with it's A input not connected. How will you tie the input to make sure the functionality is the same

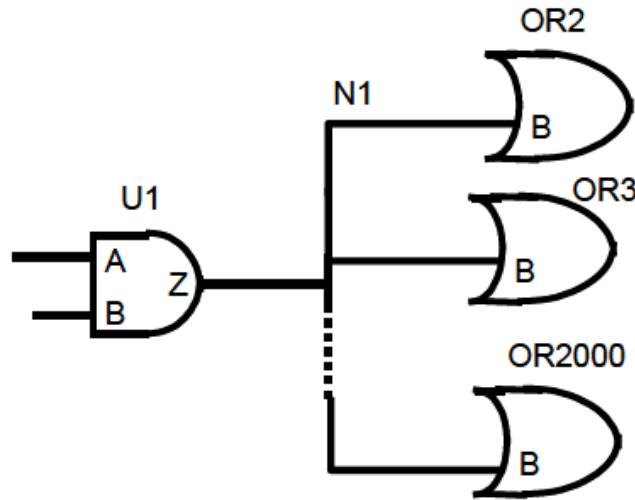
Review: QUIZ2

- What does Cell Delay U1 (prop delay) depend on?



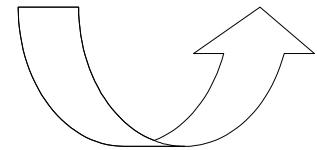
Review : QUIZ3

- What can you do to reduce the delay on U1?





Storage. Networking. **Accelerated.**TM



Review of Digital Design Fundamentals

Part 2 : Timing Paths and Various Definitions related to Design

Accelerate.



Agenda

- Synchronous and Asynchronous Clocks
- Timing Paths
 - Different types of Paths
- Various Definitions
 - Setup Time
 - Hold Time
 - Recovery & Removal
 - Metastability

Synchronous Dictionary Meaning

- occurring or existing at the same time or having the same period or phase;
- "recovery was synchronous with therapy"; "a synchronous set of clocks"; "the synchronous action of a bird's wings in flight";
- pertaining to a transmission technique that requires a common clock signal (a timing reference) between the communicating devices in order to coordinate their transmissions
- opposed to asynchronous

Asynchronous Dictionary Meaning

- not occurring or existing at the same time or having the same period or phase
- pertaining to a transmission technique that does not require a common clock between the communicating devices
- Not simultaneous; not concurrent in time;
- opposed to synchronous.

What synchronous/asynchronous mean for clocks in a design?

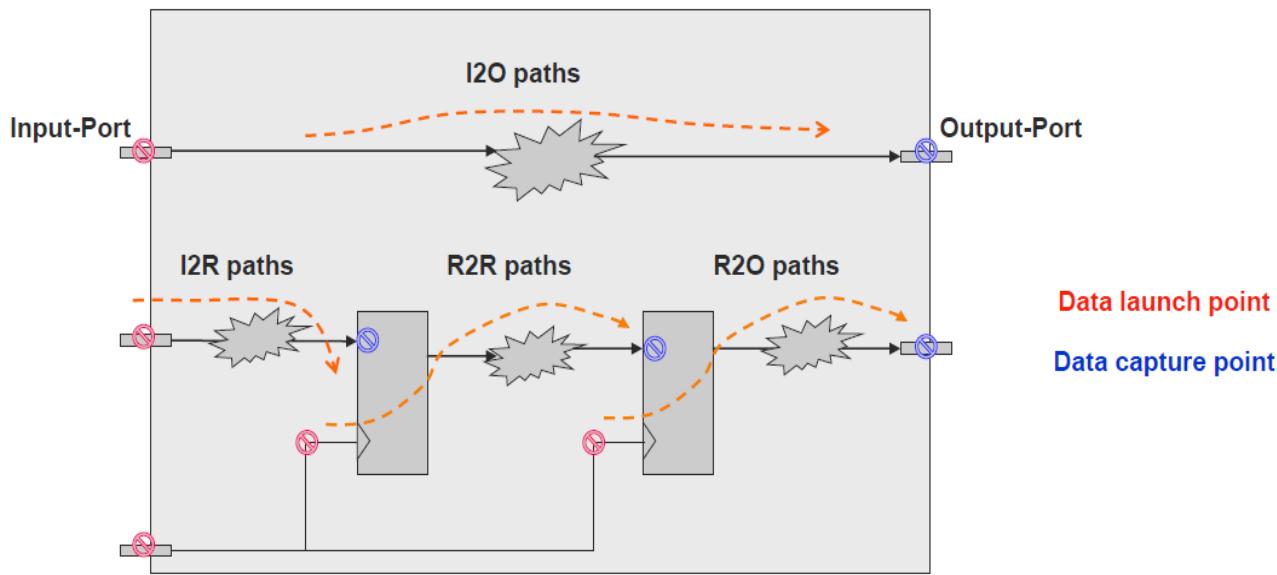
- All devices
- Synchronous systems are predictable for correct data and timing.
- All clocks are synchronous; well defined waveform that is continuous in time.

Synchronous vs. Asynchronous Designs

- Two clocks are considered '**SYNCHRONOUS**' to each other when:
 - The frequency of each clock is DIVISIBLE to a common multiple.
 - The absolute START TIME for either clock is the same or DEFINED relative to each other.
 - Basically when both clocks are GENERATED from the SAME MAIN SOURCES with DIVISIBLE frequencies to each other.
 - ONLY then both clocks are SYNCHRONOUS to each other as there exists predictability for timing relative to each other.
- Two clocks are considered '**ASYNCHRONOUS**' to each other when:
 - The frequency of each clock is NOT DIVISIBLE to a common multiple.
 - The absolute START TIME for either clock is TOTALLY INDEPENDENT of the other clock.
 - Basically when both clocks are generated from TOTALLY INDEPENDENT MAIN SOURCES or the frequencies are NOT DIVISIBLE to a common multiple
 - Both clocks are considered INDEPENDENT and therefore ASYNCHRONOUS to each other as there exists NO predictability for timing relative to each other.

Timing Paths – Block Level

- Data Path – A timing Path that
 - Starts at a data launch point
 - Ends at a data Capture Point



Data launch point

Data capture point

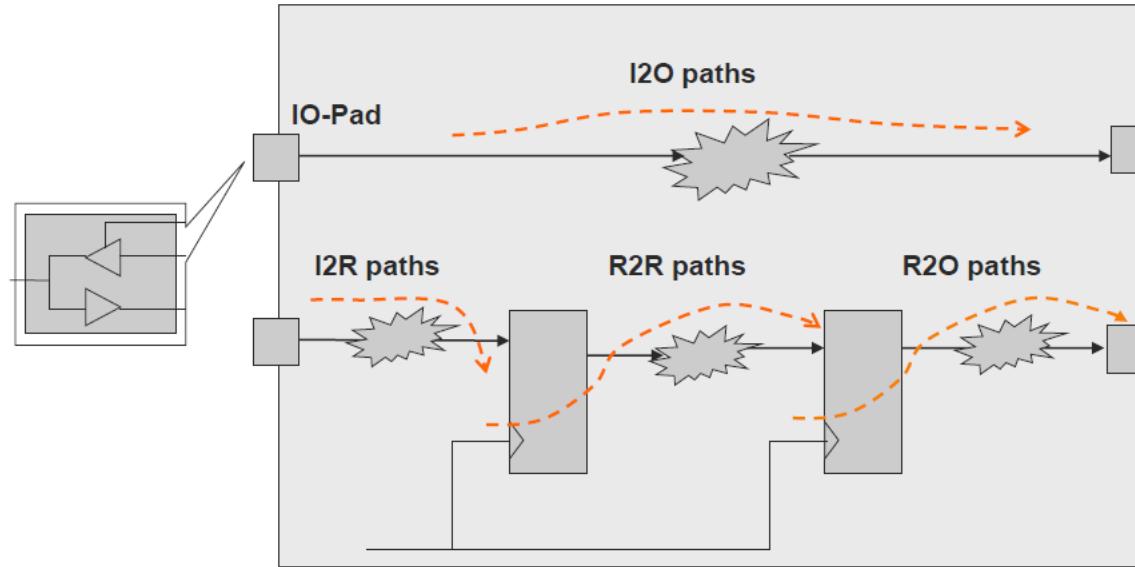
I2O – Input to Output

I2R – Input to Register

R2R – Register to Register

R2O – Register to Output

Data Path – SOC Level



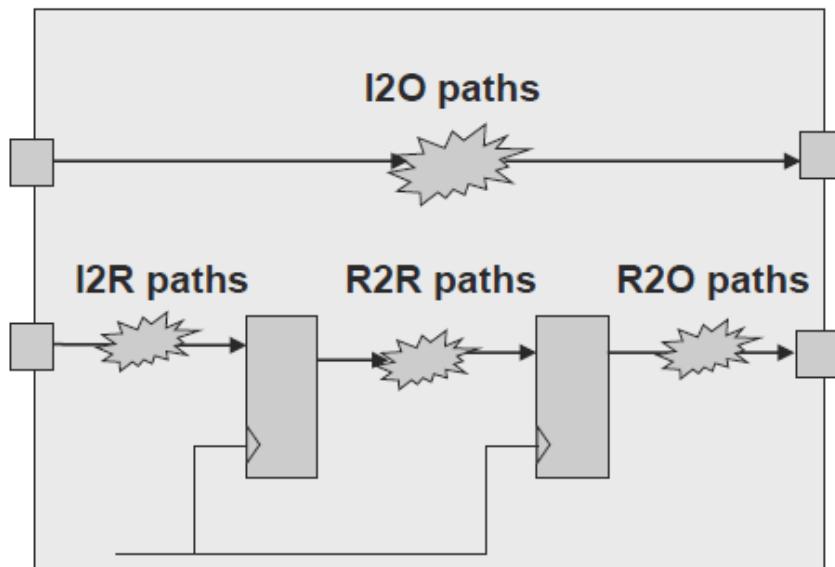
Sync Points

- Clock inputs
- ASIC outputs
- ASIC inputs
- Internal storage elements such as flip flops, latches, RAMs and ROMs, Hardmac ports
- PLL's
- Clock Gating logic

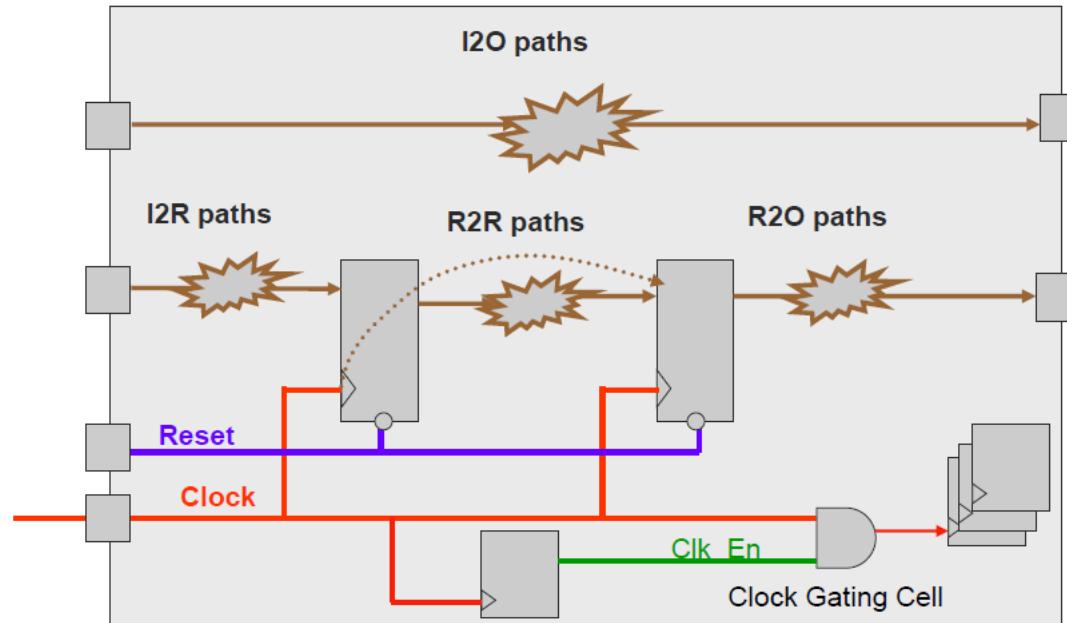
All paths start and end at Sync Points for Timing Paths

What is a Timing Path?

- A timing path is a point –point sequence through a design, that
 - Starts At
 - A register clock pin
 - An Input Port
 - Passes through
 - Combinational logic elements
 - End At
 - A register data input pin
 - An output port

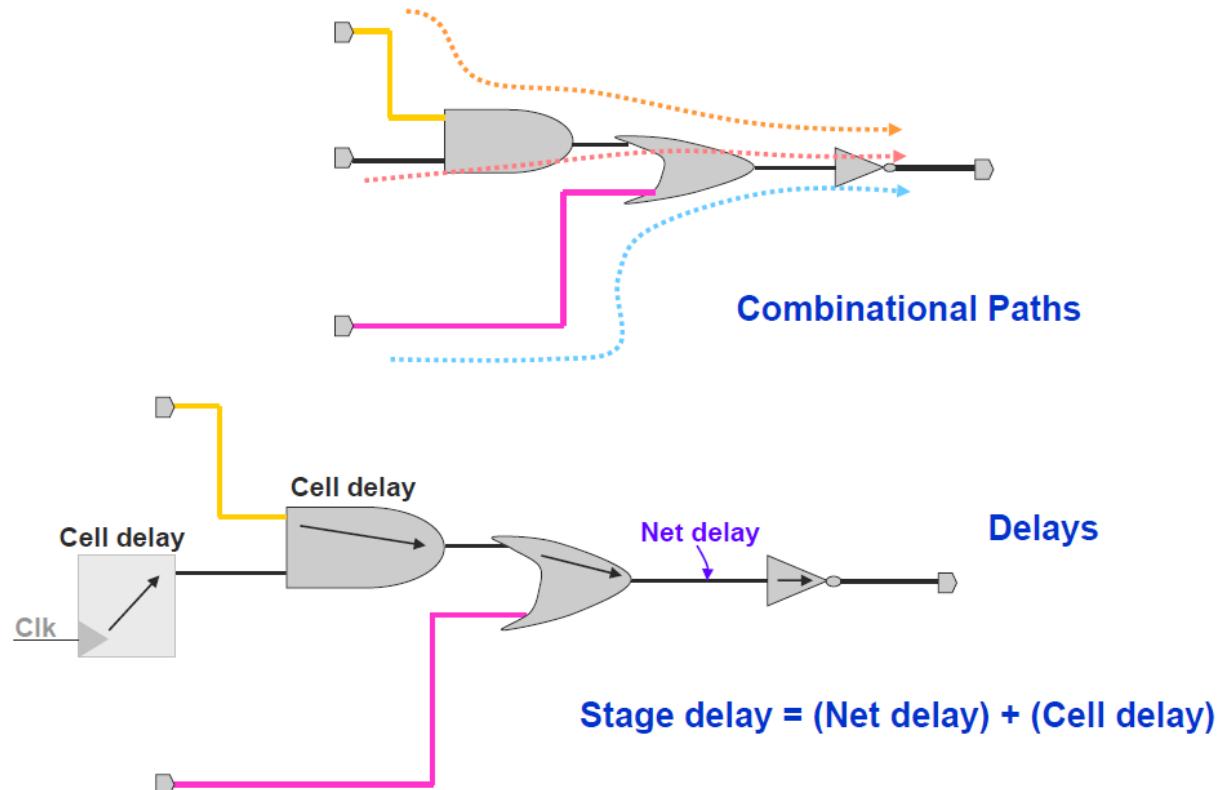


Various Timing Paths

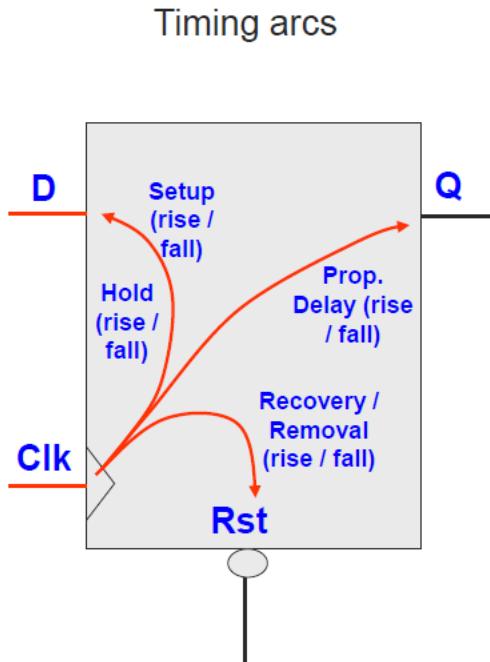


- Combinational, Data Paths
- Asynchronous Paths: Paths to Reset pin
- Clock Paths: Paths to Clock pin
- Clock-gating Paths

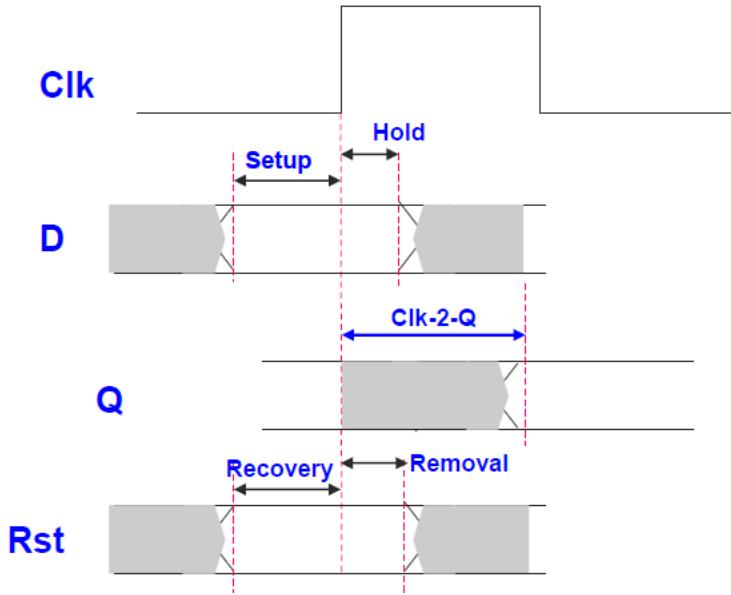
Combinational Paths & Delays



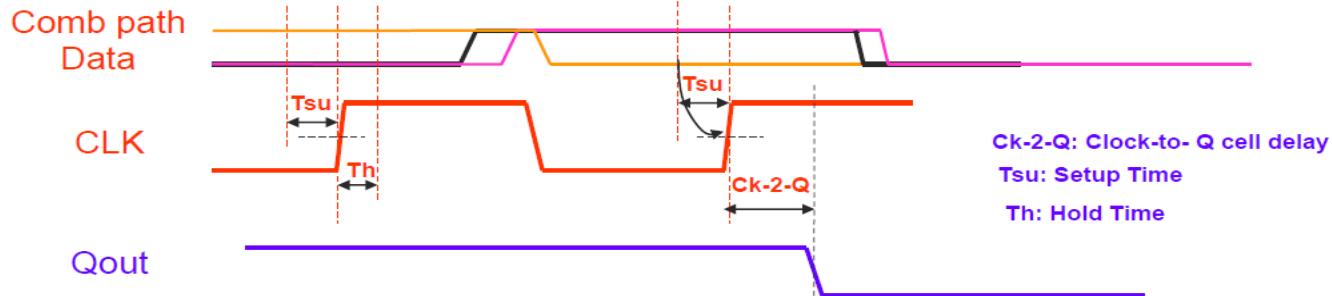
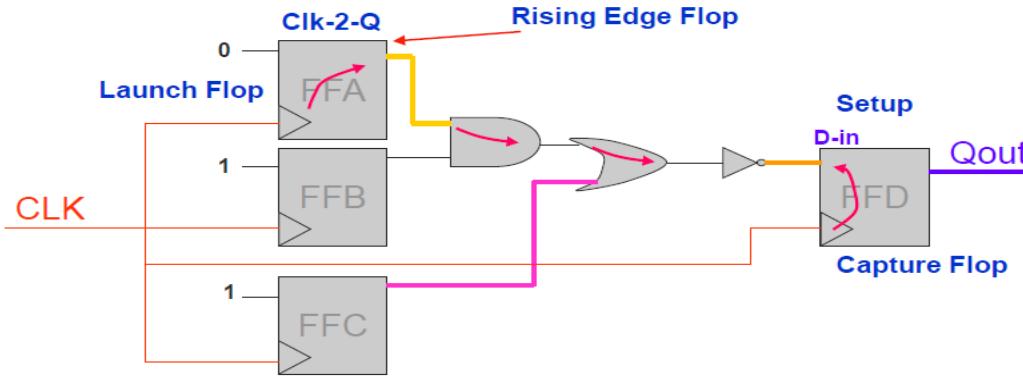
Flip Flop Timing Arcs



For Synchronous Inputs: Setup, Hold
For Async inputs: Recovery, Removal
For Synchronous outputs: Propagation Delay

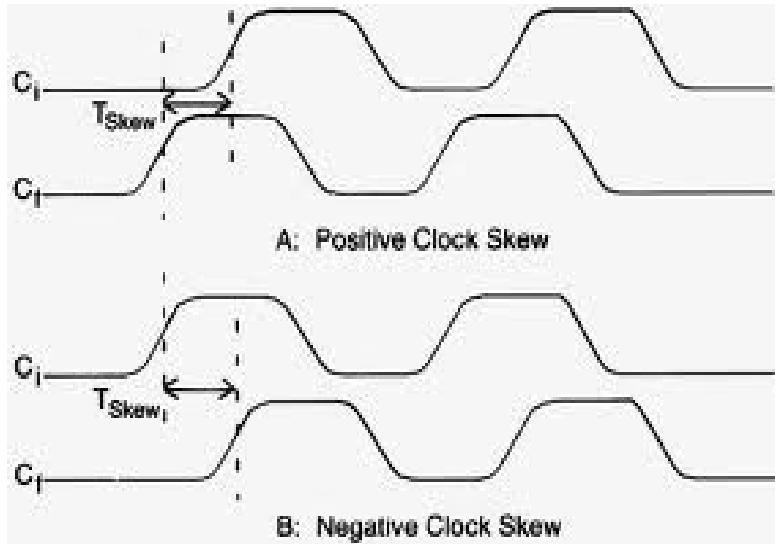


Flop to Flop Timing



Clock Skew

- Clock Signal arrives at different times for different components
- Clock Skew (T_{skew}) can be defined as the difference in the arrival time between two sequentially adjacent registers

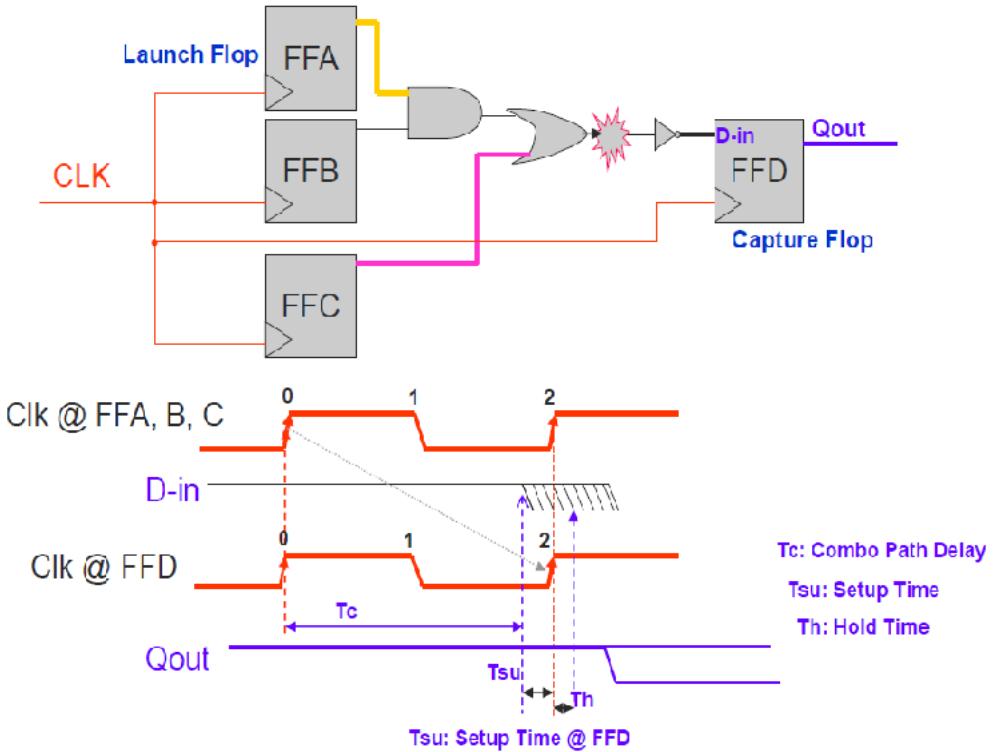


As the clock period increases, timing becomes more critical and less variation can be tolerated if the circuit is to function properly

Setup Timing

Setup time is the minimum amount of time the data signal should be held steady **before** the clock event so that the data are reliably sampled by the clock

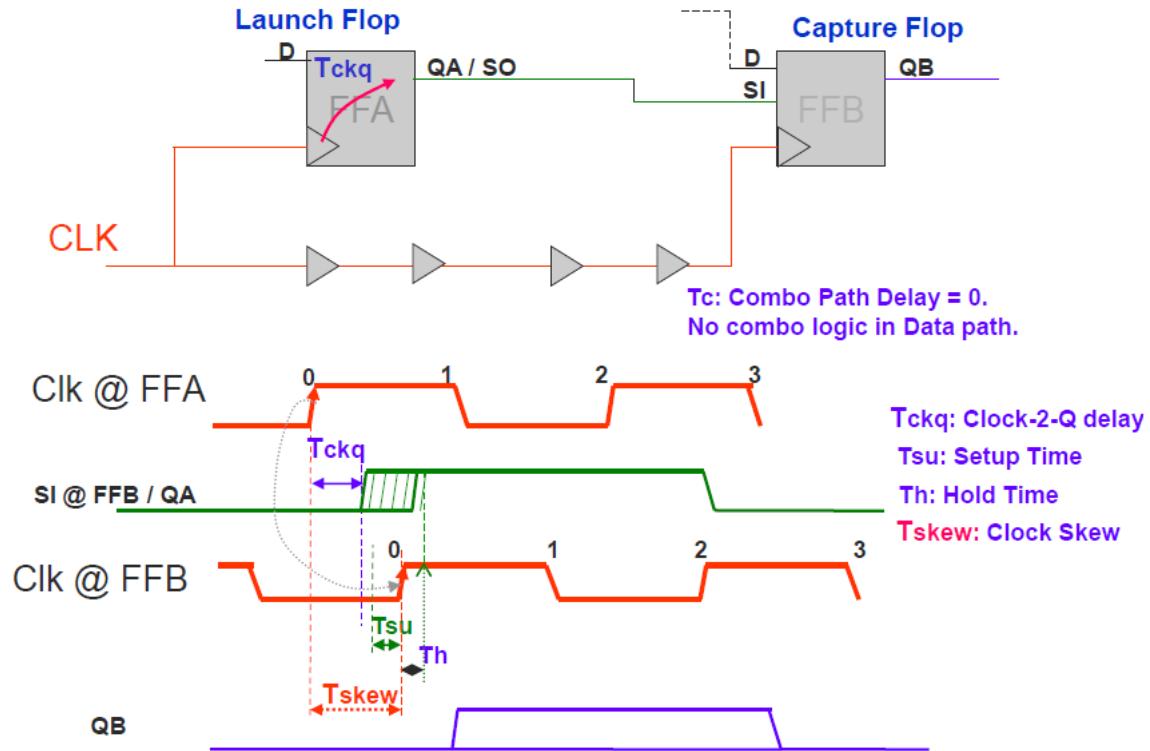
- This applies to synchronous input signals to the flip-flop



Hold Timing

Hold time is the minimum amount of time the data signal should be held steady after the clock event so that the data are reliably sampled

- This applies to synchronous input signals to the flip-flop



Setup Time & Hold Time Calculation

· Setup Time

$$T_{C1} + T_{CK2Q1} + T_{combo1} < T_{clk} + T_{C2} - T_{setup2}$$

$$T_{setup2} < T_{C2} - T_{C1} - T_{CK2Q1} - T_{combo1} + T_{clk}$$

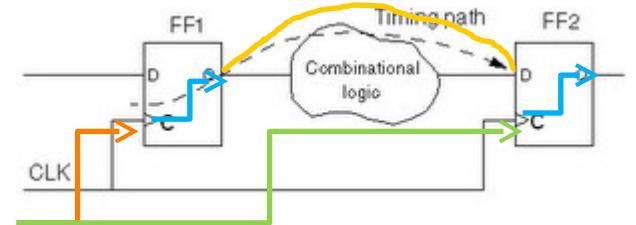
· Hold Time

$$T_{C1} + T_{CK2Q1} + T_{combo1} > T_{C2} + T_{hold2}$$

$$T_{C1} - T_{C2} + T_{CK2Q1} + T_{combo1} > T_{hold2}$$

· Clock Skew

$$T_{C1} - T_{C2}$$



T_{clk} = Clock Period

T_{C1} = Clock Delay to FF1

T_{C2} Click icon to add picture
= Clock Delay to FF2

T_{combo1} = Combinational Logic Prop Delay

T_{CK2Q1} = Clock to Q delay of FF1

T_{setup1} = Setup Time of FF1

T_{setup2} = Setup Time of FF2

T_{hold1} = Hold Time of FF1

T_{hold2} = Hold Time of FF2

T_{CK2Q2} = Clock to Q Delay of FF2

Library Requirements

Does setup or Hold depend on clock frequency?

Setup Time & Hold Time Calculation contd

• Setup Time

$$T_{C1\max} + T_{CK2Q1\max} + T_{combo1\max} < T_{clk} + T_{C2\min} - T_{setup2}$$

$$T_{setup2} < T_{C2\min} - T_{C1\max} - T_{CK2Q1\max} - T_{combo1\max} + T_{clk}$$

T_{clk} = Clock Period

• Hold Time

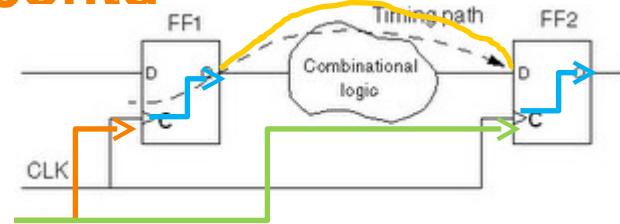
$$T_{C1\min} + T_{CK2Q1\min} + T_{combo1\min} > T_{C2\max} + T_{hold2\min}$$

$$T_{C1\min} - T_{C2\max} + T_{CK2Q1\min} + T_{combo1\min} > T_{hold1\min}$$

• Clock Skew

$$T_{C1\max} - T_{C2\min} = \text{Clock Skew for Setup}$$

$$T_{C1\min} - T_{C2\max} = \text{Clock Skew for Hold}$$



T_{C1max} = Clock Delay to FF1

T_{C1min} = Clock Delay to FF1

T_{C2max} = Clock Delay to FF2

T_{C2min} = Clock Delay to FF2

T_{combo1max} = Combinational Logic Prop Delay

T_{combo1min} = Combinational Logic Prop Delay

T_{CK2Q1max} = Clock to Q delay of FF1

T_{CK2Q1min} = Clock to Q delay of FF1

T_{setup1} = Setup Time of FF1

T_{setup2} = Setup Time of FF2

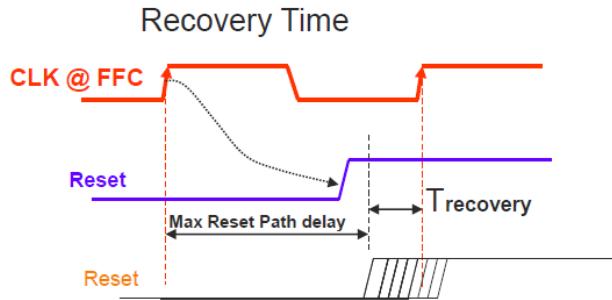
T_{hold1} = Hold Time of FF1

T_{hold2} = Hold Time of FF2

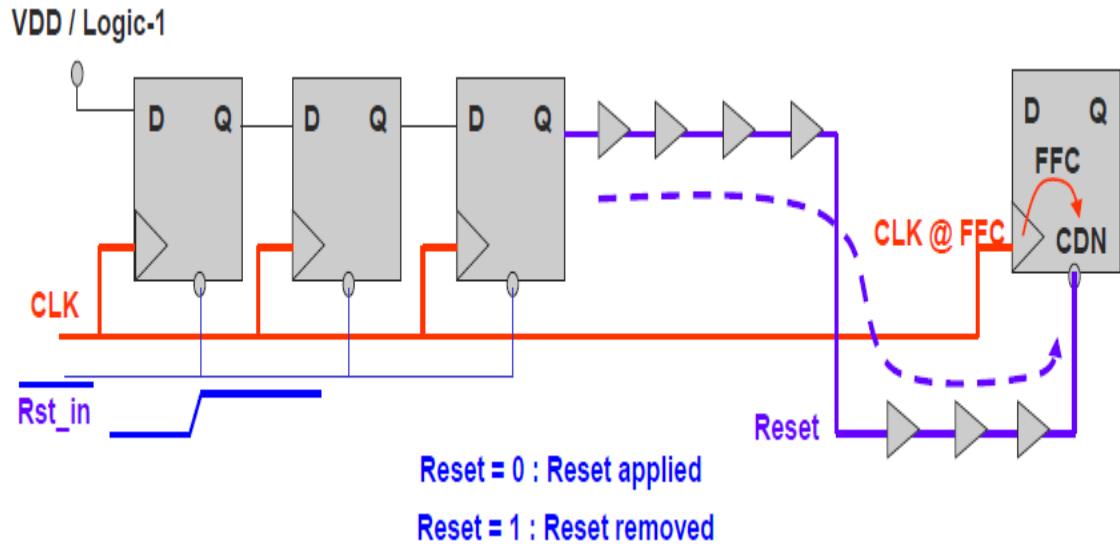
T_{CK2Q2max} = Max Clock to Q Delay of FF2

T_{CK2Q2min} = Min Clock to Q Delay of FF2

Recovery Timing

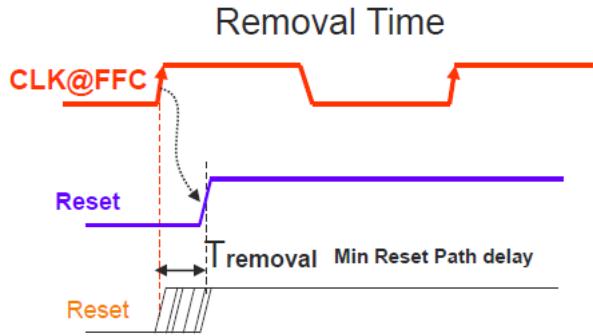


T_{recovery}: Minimum time before active clock edge for which reset should not change after getting removed / de-asserted

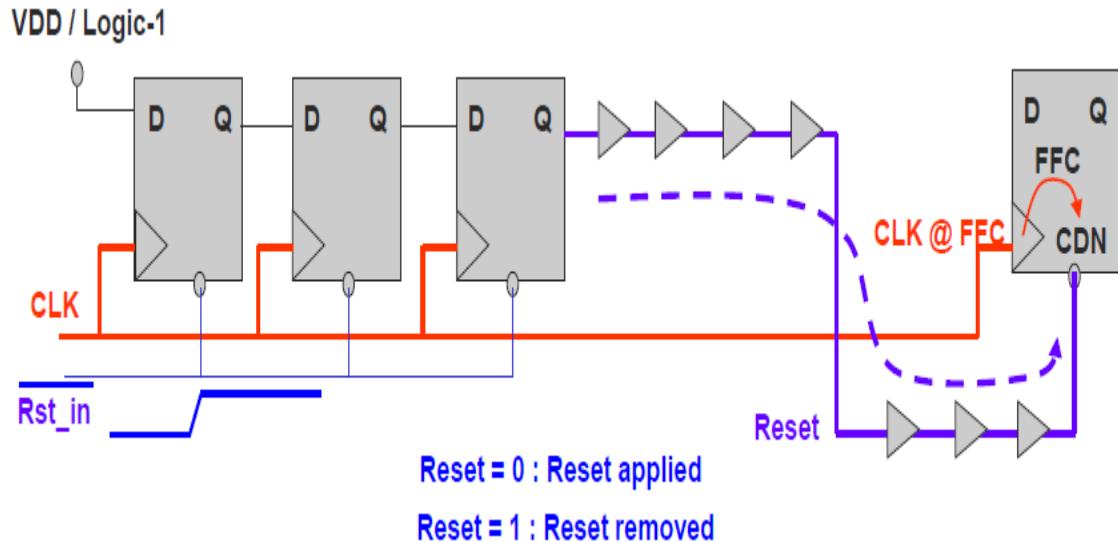


- **Recovery time** is like setup time for asynchronous ports (set, reset). It is the time available between the asynchronous signals going inactive and the active clock edge.

Removal Timing

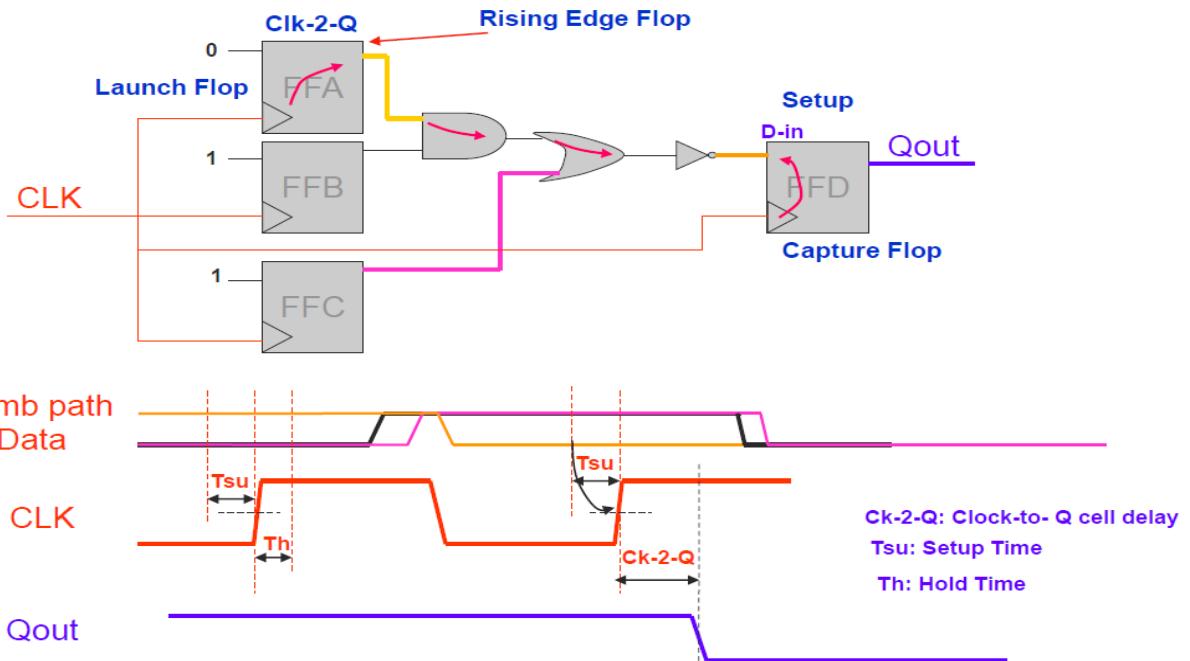


T_{removal}: Minimum time after active clock edge for which reset should not change before getting removed / de-asserted



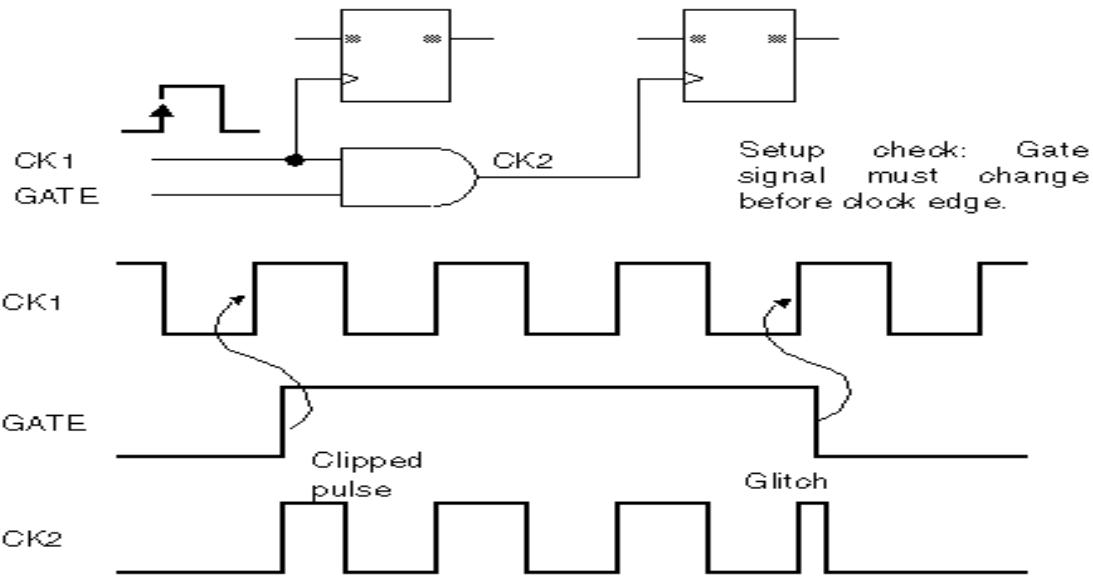
- **Removal time** is like hold time for asynchronous ports (set, reset). It is the time between active clock edge and asynchronous signal going inactive

Meta Stability



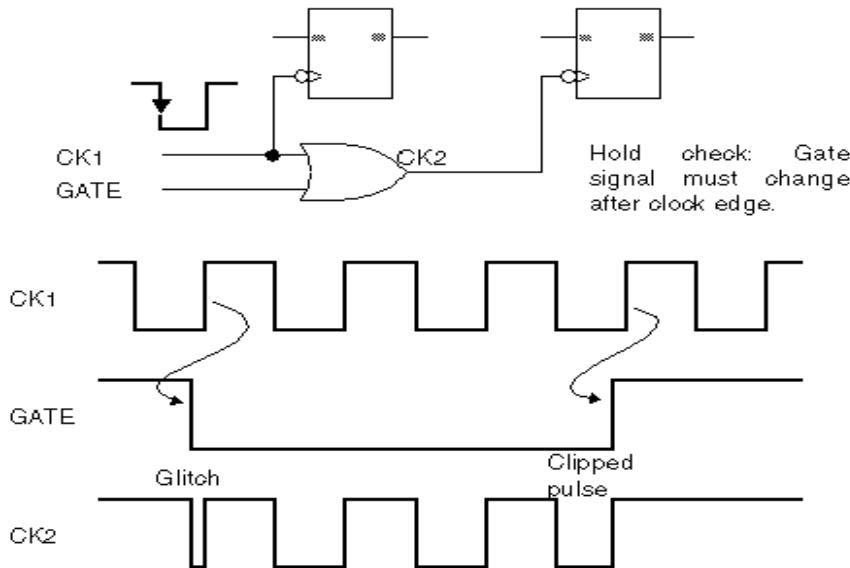
- When data changes in the region $T_{su} + T_h$, Metastability occurs
- One cannot predict the value of Q_{out} when data changes here

Clock Gating Paths : Setup Checks



The clock-gating setup check ensures that the controlling data signal enables the gate before the clock becomes active

Clock Gating Paths: Hold Checks



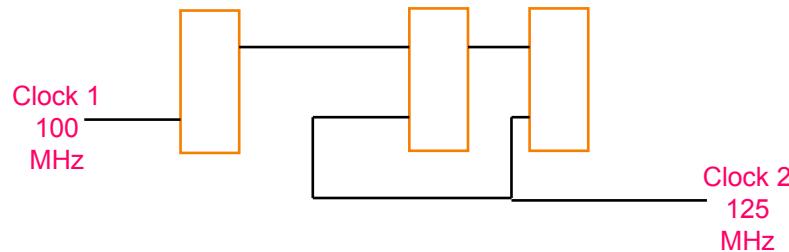
The clock-gating hold check ensures that the controlling data signal remains stable while the clock is active

Review : QUIZ1 – Synchronous Paths

- Can registers clocked by clocks synchronous to each other able transmit data to/from each other?
- What does this mean for setup/hold requirements?
- What will be the cycle time?

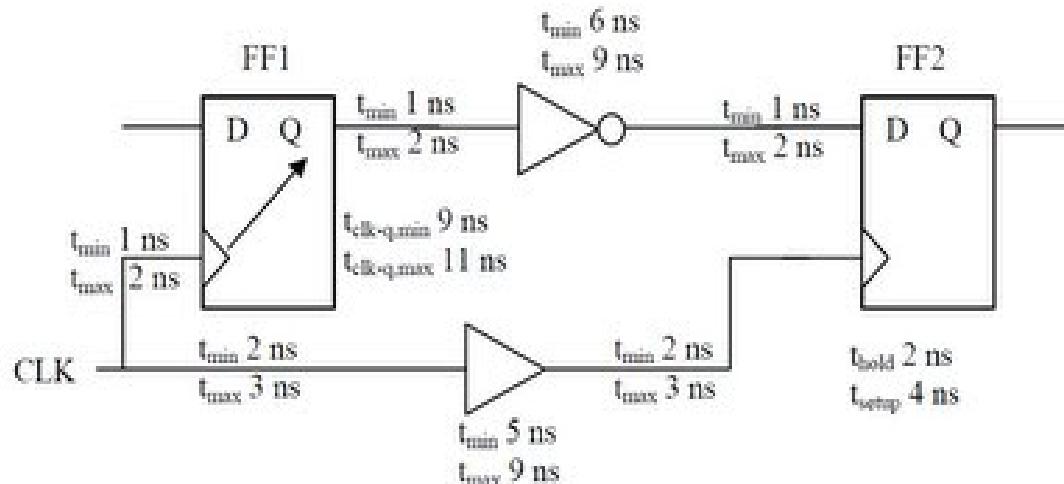
Review: QUIZ2 -Asynchronous Paths

- Can registers clocked by clocks asynchronous to each other able transmit data to/from each other?
- What is synchronizing logic?
- How are asynchronous clock crossing paths defined in terms of STA?
- What do we do then for these types of paths?



Review QUIZ3 – Setup and Hold

- Find out if there is setup or hold violation in this circuit



Review: QUIZ4 Sequential and Combinational Circuits

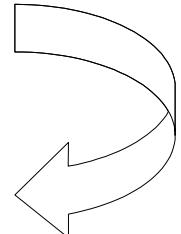
- How do we know whether a given circuit is a Combinational Circuit or a Sequential Circuit?
- Why do we have to identify the type of circuit? Does it really matter?

Review: QUIZ5 Circuit

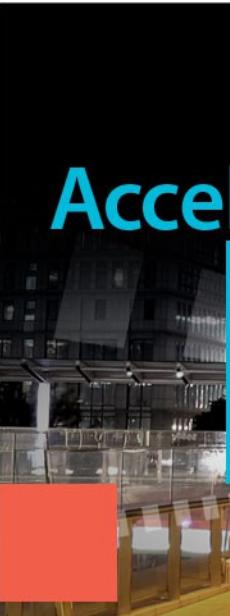
- What timing parameters must be provided/calculated for a synchronous sequential device?



Storage. Networking. **Accelerated.**TM



Overview of Technology and Libraries



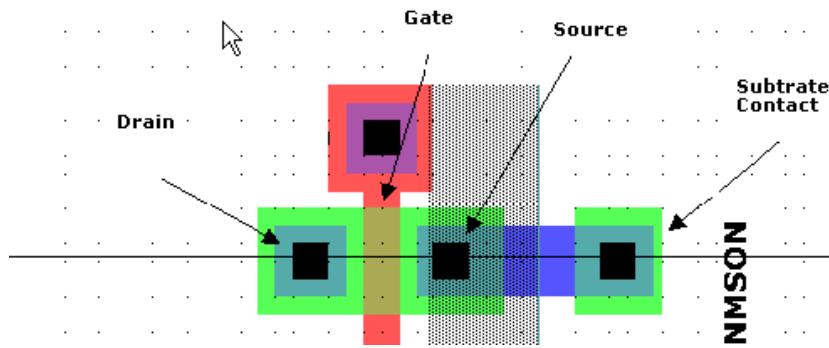
Accelerate.



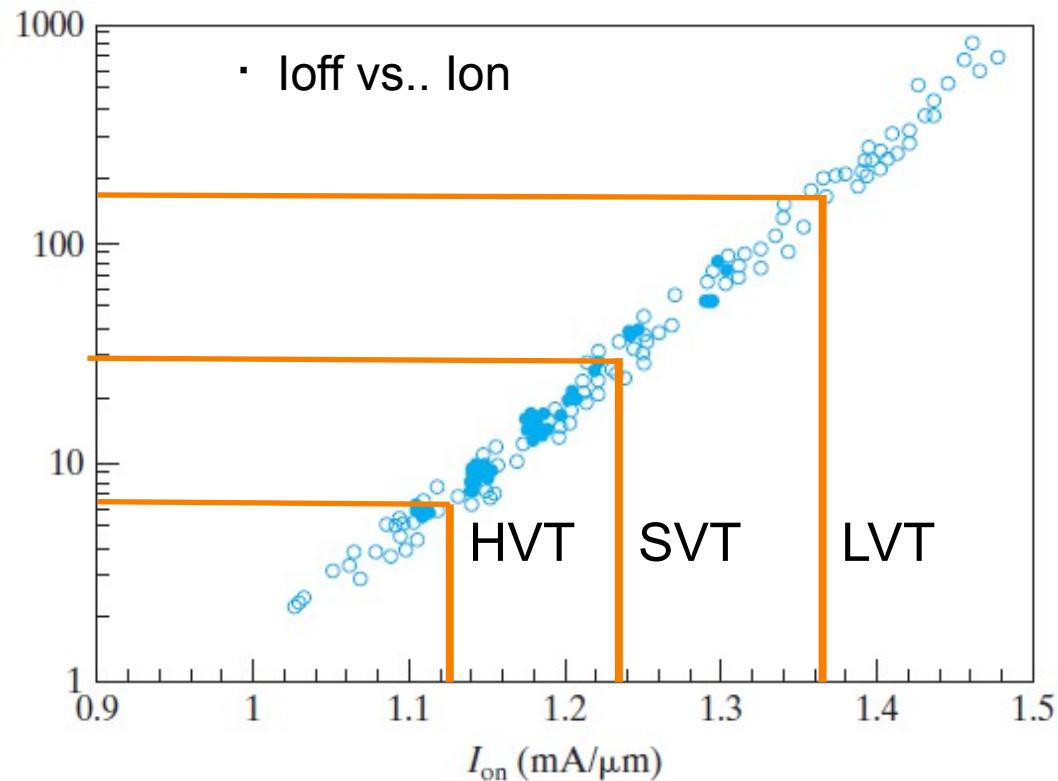
What do you mean by Process Technology?

- Process Technology refers to a particular method used to make Si Chips
 - Typically denoted by the size of the minimum length of the transistor supported in that process.
 - A 40 nm process technology refers to features 40 nm or 0.040 μm in length.
 - The smaller the length of transistors
 - the more transistors in the same area,
 - the faster they switch

Layout of a transistor



Multiple Voltage Thresholds (Multi- V_t) : An overview



@ IEEE 2003

- V_t (Threshold Voltage): Broadly, voltage @ the Gate when transistor switches on.
- Low V_t :
 - V_t ↓ Switching Speed ↑
 - Penalty I_{off} is larger so Standby Power is larger
- High V_t  Click icon to add chart
 - V_t ↑ Switching Speed ↓
- Strategy:
 - Low V_t for critical delay paths
 - High V_t in non-critical to save power.

Design Components of an ASIC

- Standard Library Cells
- Memories
- IO buffers
- IP

LSI Technology Offerings

- LSI offers the following technologies – Mentioning the current 4 technologies
 - 65nm
 - 40nm
 - 28nm
 - 20nm (work in progress)

28nm Technology – General Specifications

Symbol	Parameter	Min	Nom/Typ	Max	Unit
V _{DD}	Core supply voltage ^a	0.765	0.85	0.935	V
V _{DDIO}	1.5V supply (V _{DDIO15})	1.4	1.5	1.6	V
	1.8V supply (V _{DDIO18})	1.62	1.8	1.98	V
	2.5V supply (V _{DDIO25})	2.25	2.5	2.75	V
	3.3V supply (V _{DDIO33})	2.97	3.3	3.465	V
T _J	Operating junction temperature	-40	25	125	°C
T _{STG}	Storage temperature (ceramic)	-65	25	150	°C
	Storage temperature (plastic)	-40	25	125	°C

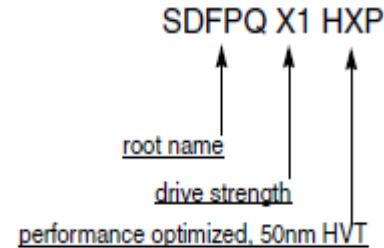
40nm Specifications

Symbol	Parameter	Min	Nom/Typ	Max	Unit
V _{DD}	Core supply voltage ¹	0.81	0.9	0.99	V
V _{DDIO}	1.5V supply (V _{DDIO15})	1.4	1.5	1.6	V
	1.8V supply (V _{DDIO18})	1.62	1.8	1.98	V
	2.5V supply (V _{DDIO25})	2.25	2.5	2.75	V
	3.3V supply (V _{DDIO33})	2.97	3.3	3.465	V
T _J	Operating junction temperature	-40	25	125	°C
T _{STG}	Storage temperature (ceramic)	-65	25	150	°C
	Storage temperature (plastic)	-40	25	125	°C

40nm Technology – Libraries & Naming Convention

Threshold Implant	Cell Architecture				
	Performance		Density		
	Channel Length		Channel Length		
	50 nm	40 nm	50 nm	45 nm	40 nm
	LVT (Low)	LXP	LMP	LXD	LID
SVT (Standard)	SXP	SMP	SXD	SID	SMD
HVT (High)	HXP	-	HXD	-	-

Click icon to



28 nm Technology – Libraries

20 Total Libraries

	35nm P	35nm D
HVT	35	35
SVT	31/35/38	31/35/38
LVT	31/35/38	31/35/38
ULVT	31/35/38	31/35/38
Total	10	10

HVT devices have default +3nm bias

950+ cells in library

- Rich functional set of cells
- Fractional drive strengths
- Power management kit
- P/N ratio tuned by cell

Custom cells to support IP

- DDR/processor cells
- Cell flow for IP teams

Manufacturing focus

- Highly regular layout style
- Optimized to DFM rules

Collaborative emphasis on optimizing library for place & route
· Iterated 50 cell comparison libraries to find best architecture

28nm Naming Convention

<base_name>X<drive><type>\<version><implant><family><length><library>

Where:

<base_name> is the ‘root’ name of the cell (nand2, inv, etc)

<drive> is the drive strength; for fractional drives, replace the decimal point with a “P”

<type> indicates the design methodology:

M : optimized to minimize the maximum delay

B : optimized to balance rise/fall and delay

F : skewed to favor falling edge delay

R : skewed to favor rising edge delay

<version> is the version number; in 28nm all cells will have version numbers, beginning with “V0”



28nm Naming Convention contd.

<implant> is the threshold voltage:

U : ultra-low Vt

L : low Vt

S : standard Vt

H : high Vt

<family> is the channel length “class”:

M : minimum channel length (30nm nominal)

I : Intermediate channel length (35nm nominal)

X : extended channel length (40nm nominal)

<length> is the channel length (in nm)

<arch> is the cell architecture (currently either P or D)

Complete cell list with naming convention can be found :

<http://ssg.teams.Lsi.com/sites/cs/dept/fips/ckts/Shared%20Documents/FIP%20Libraries/28HP%20PRD%20V1.0.pdf>



Supplemental Libraries

Library suffix	Description
_bf	BackFill cells. These are cells for metal only ECO programming based on ACELLs. In the 40nm libraries, these were built into the primary synthesis libraries with dont_use properties. However to reduce risk of accidentally removing the property and synthesizing using these cells by mistake, they were moved into a standalone library.
_ls	Level Shifters (SVt only initially) LSBUFH - level-shifter buffer with isolation high LSBUFL - level-shifter buffer with isolation low LSBUFX - level-shifter buffer with no isolation
_phys	Physical only cells. DCAP, FILL and TAP cell types

Where are these Libraries

- Flexstream6.0/40nm IP
 - \$LSI_RELEASE: lsi_fs_6.0
 - \$LSI_IP: lsi40nm_1.0
 - \$LSI_CW: lsi_cw_6.0
- Flexstream7.0/28nm IP
 - \$LSI_RELEASE: lsi_fs_7.0
 - \$LSI_IP: lsi28nm_1.0
 - \$LSI_CW: lsi_cw_7.0
- General 28nm Library Twiki:
<http://ftctwiki.co.lsil.com/twiki/bin/view/Fo>

The library structures are very similar for both 40 nm and 28nm

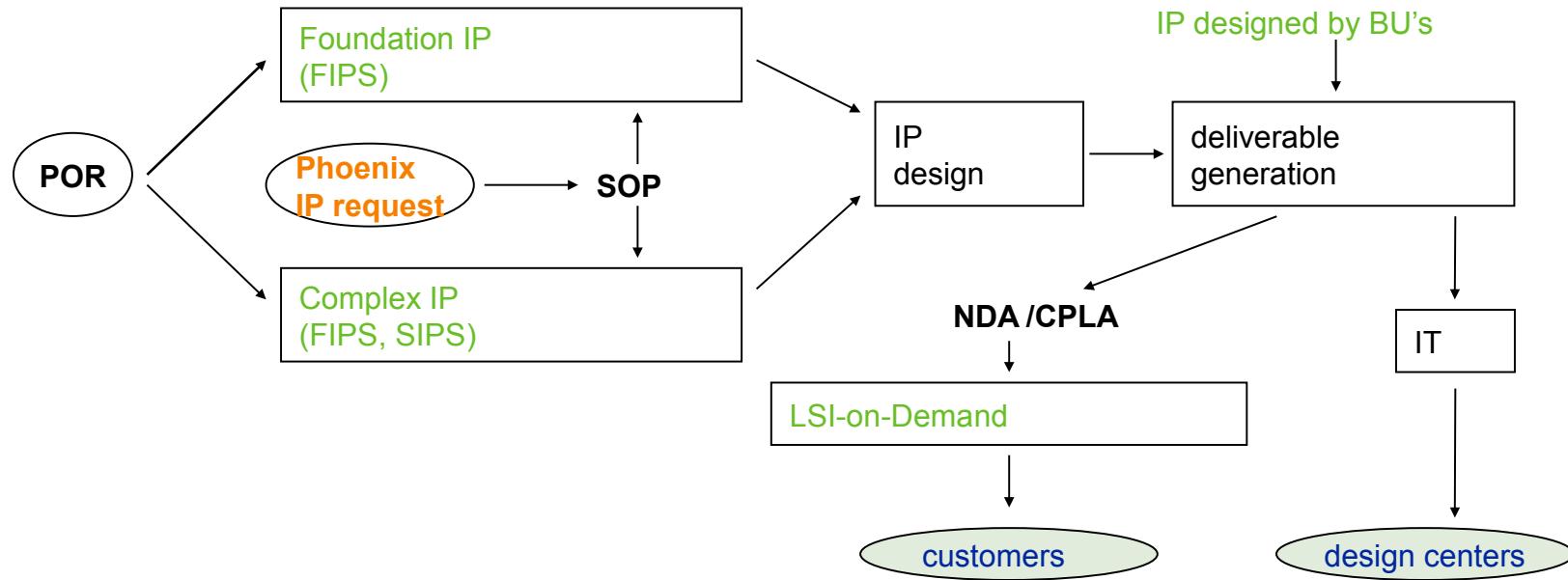
Memories : 40nm

Type	Source	Memory	Description
SRAM	Virage	111HD	1-Port High Density Compiler (Low Voltage capable)
	LSI/TSMC JDA	111UHD	1-Port Ultra-High Density Compiler
	LSI	111HS	1-Port High Speed Compiler
	Virage	111RF	1-Port Register File Compiler (Low Voltage capable)
	LSI/TSMC JDA	111RF	1-Port Register File Compiler (Low V capable, High Speed)
	Virage	211RF	2-Port Register File Compiler (Low Voltage capable)
	LSI	211MRF	Micro 2-Port Latch Based Register File Compiler
	Virage	222HD	Dual-Port High Density Compiler (Low Voltage capable)
	LSI	N-Port	Multi-Port (3-5) SRAM Compiler (Optimized for 422, wrap 2 n-ports for larger configs)
eDRAM	TSMC	eDRAM-HS	eDRAM Random Access Macro (412-500Mhz)
	TSMC	eDRAM-HD	eDRAM Interleaved Macro (500 Mhz-1Ghz)
CAM	LSI	BCAM	Binary CAM Compiler
	LSI	XY-TCAM	XY Ternary CAM Compiler
Other	LSI/TSMC JDA	vROM	Via Programmable ROM Compiler (Low Voltage capable)
	TSMC	OTP	128x8 eFuse OTP
	LSI	Custom	Custom Memories

Memories : 28nm

Type	Memory	Source	Bitcell	Operating Voltage	Flexible Perf vs. Power						
						Active	Deep Sleep	Shut Down	Dual Rail	Low Leak	TSB
SRAM	111HD	Virage	0.155 HC	0.85v +/- 10%		LS					
	111UHD	LSI	0.127 HD	0.85v +/- 10%	CB/Vt	LS, TSB					
	111HS	LSI	0.155 HC	0.85v +/- 10%	CB/Vt	LS, TSB					
	111UHS	LSI	0.155 HC	0.85v +/- 10%	CB/Vt	LS					
	111RFS	LSI	0.155 HC	0.85v +/- 10%	CB/Vt	LS					
	111RF	Virage	0.155 HC	0.85v +/- 10%		LS					
	211HD	LSI	0.240 HC	0.85v +/- 10%	CB/Vt	LS, TSB					
	211UHD	LSI	0.155 HC	0.85v +/- 10%	CB/Vt	LS					
	211RFS	LSI	0.315 HC	0.85v +/- 10%	CB/Vt						
	211ARS	LSI	LSI	0.85v +/- 20%	CB/Vt						
	222HS	Virage	0.315 HC	0.85v +/- 10%		LS					
	222HD	Virage	0.315 HC	0.85v +/- 10%		LS					
	211RF	Virage	0.315 HC	0.85v +/- 10%		LS					
	211MRF	LSI	LSI	0.85v +10/-20%	CB/Vt						
eDram	RA HS	TSMC	TSMC	0.9v +/- 5%							
	RA HD	TSMC	TSMC	0.9v +/- 5%							
CAM	BTCAM	LSI	LSI	0.85v +10/-20%	CB/Vt	Nand					
	HDXY-TCAM	LSI	LSI	0.85v +/- 10%	CB/Vt	PreSrch					
Other	vROM	Virage	Virage	0.85v +/- 10%		LS					
	OTP - eFUSE	TSMC	TSMC	0.85v +/- 10%							
	111HSD	LSI	0.155 HC	0.85v +/- 10%	CB/Vt	TBD					
	111RFC	LSI	0.155HC	0.85v +/- 10%	CB/Vt						

IP Work Flow

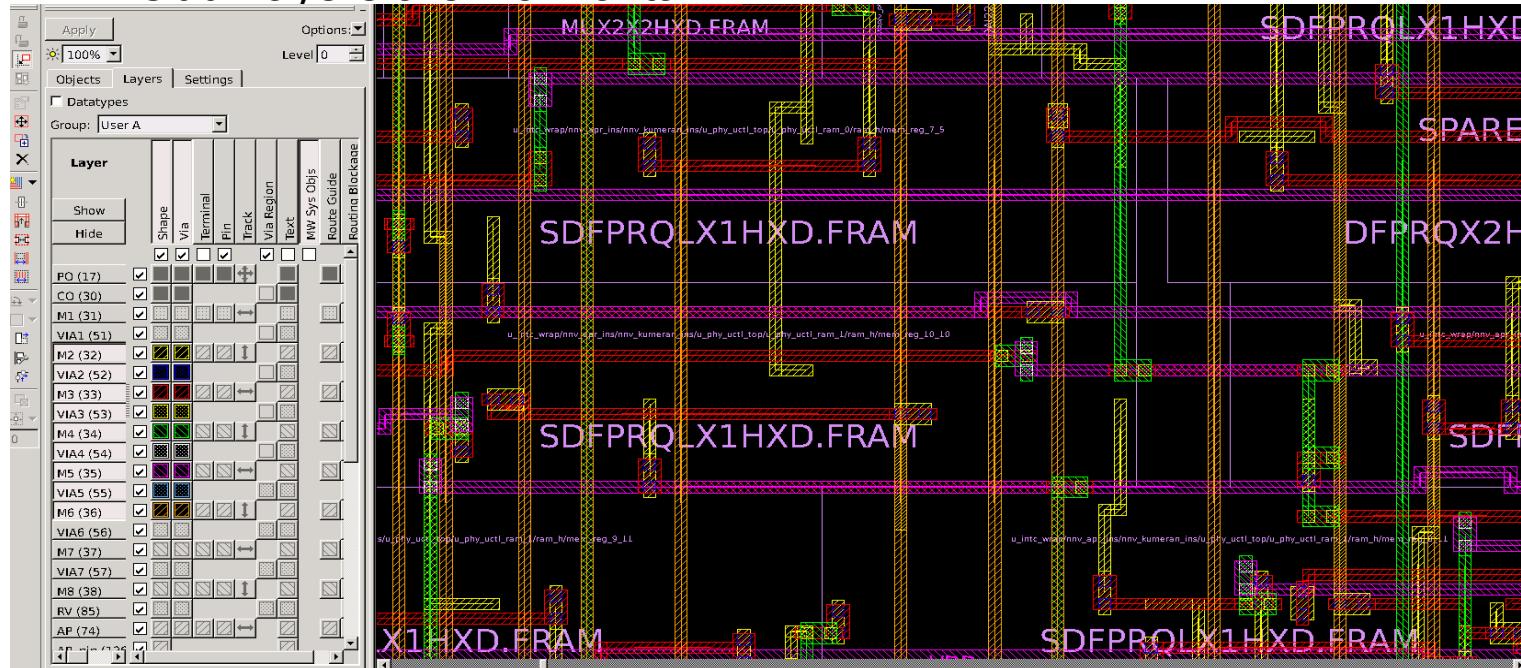


Layout of the Building Blocks

- What is a Layer Stack?
 - The metal layers used in a design to lay it out
 - Depends on the technology and also size of ASIC
- Examples of Layer Stack
 - 6+2F : M1-M6 thin layers and M7 and M8 are thick
 - 8+2F :??

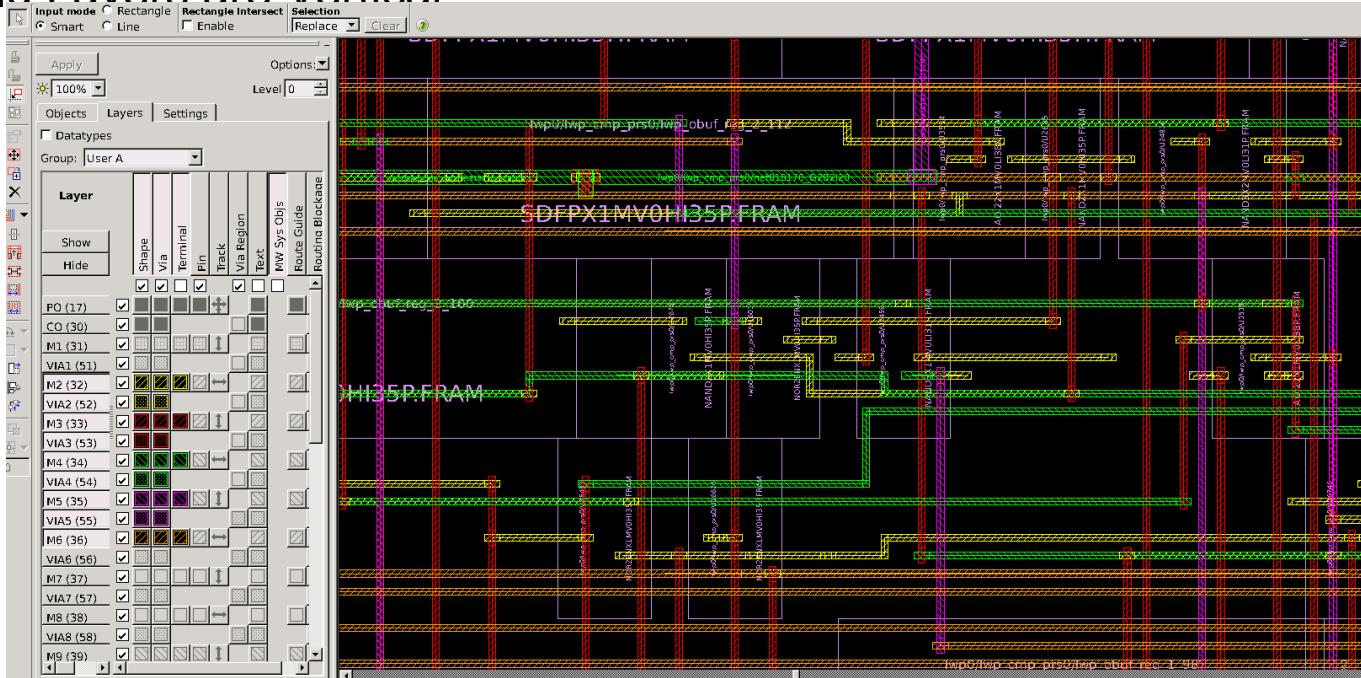
Layer Stack : 40 nm

- 40 nm
 - Odd Layers are Horizontal



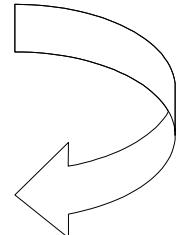
Layer Stack : 28nm

- 28 nm
 - Odd Layers are Vertical





Storage. Networking. **Accelerated.**TM

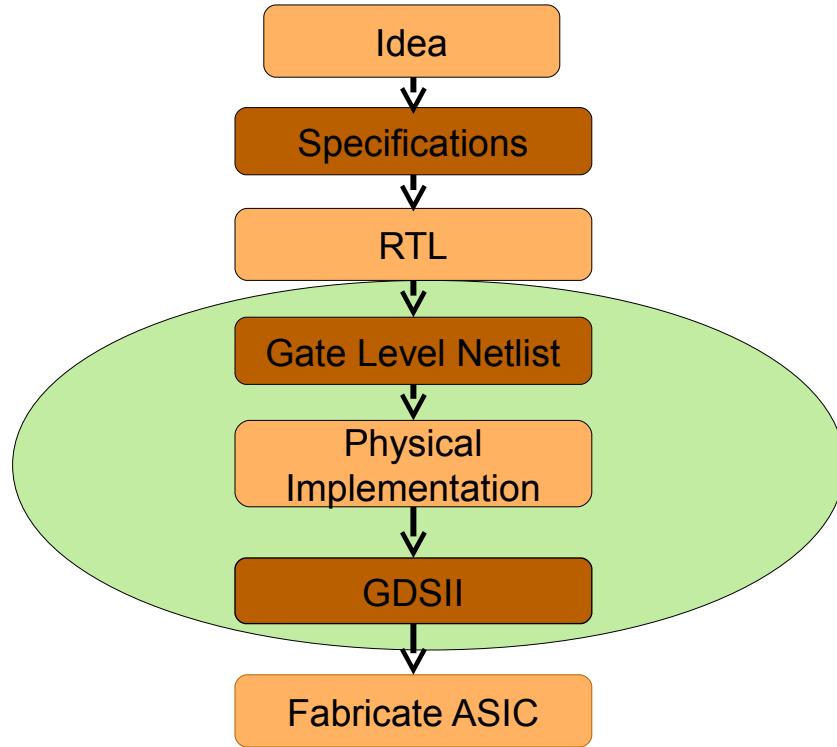


ASIC FLOW

Accelerate.



Simple ASIC Flow



Design Implementation Spec (DIS)

- Is your Bible/Gita/Kuran for implementing your design
- Designer needs to check if every aspect covered in the DIS has been met.
- Need to cross- verify requirements with the DIS to make sure it is consistent at all phases of the design

Table of Contents

1	Introduction
1.1	Purpose of this Document
1.2	Contact Information
1.3	Glossary
1.4	Related Documents
1.5	General Design Considerations
2	Overview
2.1	Features List
2.2	Top Level Block Diagram
2.3	IP, Memories, and Design Re-Use
2.4	Testability Requirements
3	Signal IO
3.1	IO Buffer Description
3.2	Pin Descriptions
4	Operating Parameters and Reliability
4.1	Device Operating Parameters
4.2	Reliability Requirements
5	Package and Pinout
5.1	Package Description
5.2	Bonding Requirements
6	Timing Diagrams and Requirements
6.1	Clocks
6.2	Phase Locked Loops
6.3	Zero Cycle Paths
6.4	Timing Modes
6.5	False Paths
6.6	Multicycle Paths
6.7	Timing Diagrams
6.8	External Skew Requirements
6.9	Resets
7	Physical Design Considerations

Engagement Models

- Netlist handover (NHO)
 - Customer delivers a netlist and constraints to LSI
 - LSI does test insertion and design closure
- RTL handover (RHO)
 - Customer delivers RTL and timing requirements (Design Spec) to LSI
 - LSI takes on all implementation work from synthesis to test insertion to design closure
 - Minor changes to handoff milestones
- Turnkey
 - Customer delivers requirements spec to LSI
 - LSI handles all development work from spec to proto delivery
 - Minimum interaction between customer and LSI needed during implementation phase
 - Two Verification milestones added

Design Phases for an ASIC

Milestone Acronym Directory

RTL Handover Milestones	
DKR	Design Kickoff Review
INITIAL PHASE	
DSR	Design Implementation Spec. Review
IDH	Initial Design Handoff
IDR	Initial Design Review
TRIAL PHASE	
CRR	Chip-Level RTL Review
TLR	Trial Layout Review
FINAL PHASE	
RFR	RTL Freeze Review
FDR	Final Design Review
PSR	Proto Ship Release
PAR	Proto Approval Release

Netlist Handover Milestones	
DKR	Design Kickoff Review
INITIAL PHASE	
DSR	Design Implementation Spec. Review
IDH	Initial Design Handoff
IDR	Initial Design Review
TRIAL PHASE	
TDH	Trial Deliverables Handoff
TLR	Trial Layout Review
FINAL PHASE	
FDH	Final Deliverables Handoff
FDR	Final Design Review
PSR	Proto Ship Release
PAR	Proto Approval Release

Purpose of Milestones and Design Flow Checklists

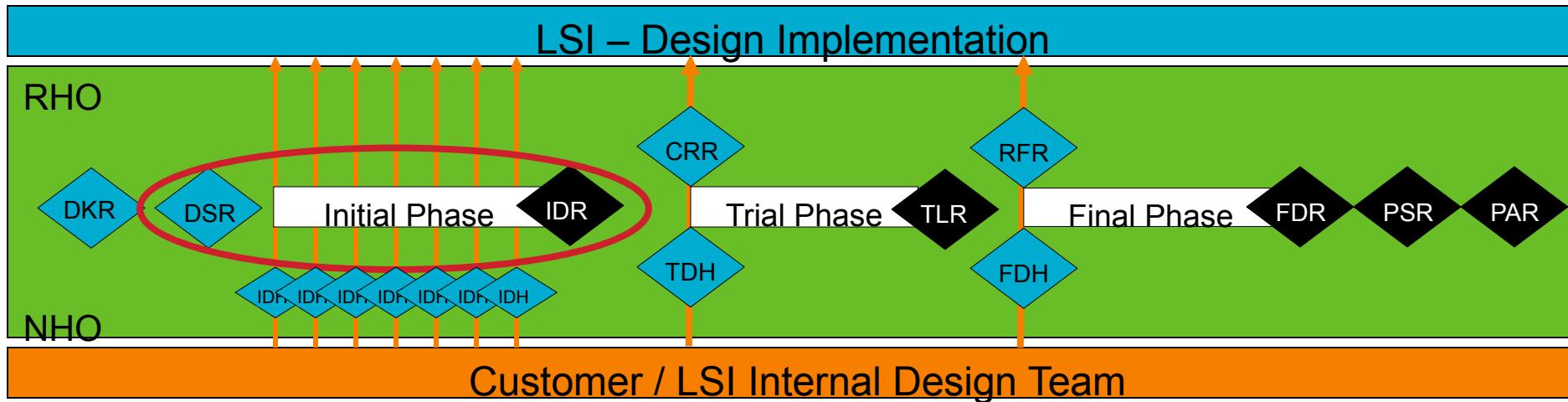
- Provide a structured, converging framework for silicon development
 - Same basic flow applicable to all engagement models
- Ensure that required tasks are completed by a certain point in the Flow
 - Early identification and resolution of technical issues
 - Enable smooth execution of implementation of final handoff
- Ensure that the Customer provided deliverables of sufficient quality to achieve these tasks at the time they are needed
- Both a technical framework and a program management framework

To improve quality, improve efficiency/TAT, incorporate lessons learned, improve ability to forecast/track progress of project

Phases of the Design Flow

- Design Flow is a Framework of Three Phases: Initial, Trial, and Final
 - Separate deliverable requirements from goals to accomplish with the deliverables
- Entry criteria for proceeding into a phase
 - Requirements to ensure handoff is of sufficient quality to enter the phase
 - Phase does not begin until the entry criteria are achieved
- Exit criteria for completion of the phase
 - Set of tasks which must be completed to finish the phase.
 - If one or more exit criteria is not met, either
 - Milestone review delayed until the task is completed; program schedule adjusted, or
 - If decision is made to still proceed to the next phase, a risk assessment should be done regarding the incomplete tasks to highlight risk to the project
 - Completion of the exit criteria for a phase is a necessary but not sufficient condition to enter the next phase
- Within each phase, flexibility to decide how to best approach each of the tasks achieve the exit criteria.
 - Must still follow approved methodologies and use approved design tools
 - Prioritize the investigation of design-specific risk items or technical issues

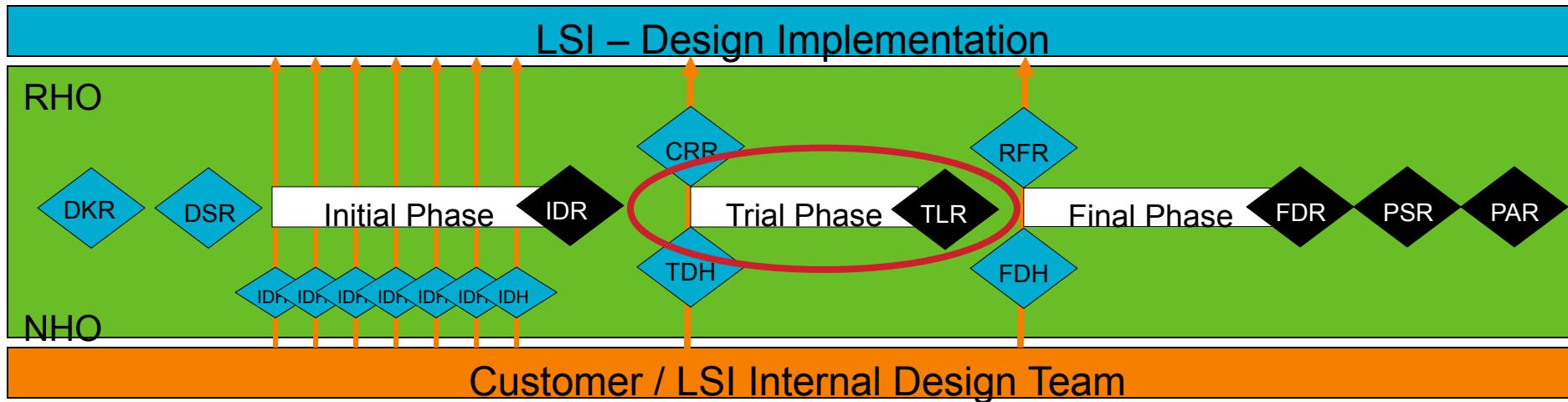
Phases: INITIAL



Phases: INITIAL

- Focuses on learning about the design, investigating and resolving technical issues, design closure issues, and addressing design specific risk items
- Iterative phase that converges on the implementation solution
 - Multiple handoffs are received and processed
 - Feedback from the DI Team is provided to the Customer and the impact of the changes is checked by evaluating a subsequent handoff
- Phase is task-oriented
 - e.g. IO Planning, RTL Analysis, Synthesis, design-for-test, Constraint Development, Floorplanning, Design Closure
 - Multiple handoffs also allows for flexible start times for tasks at appropriate times as netlist quality/completeness improves
- It is left to the discretion of Project Lead / Technical Lead as to whether to process a new handoff or continue to learn from current handoff

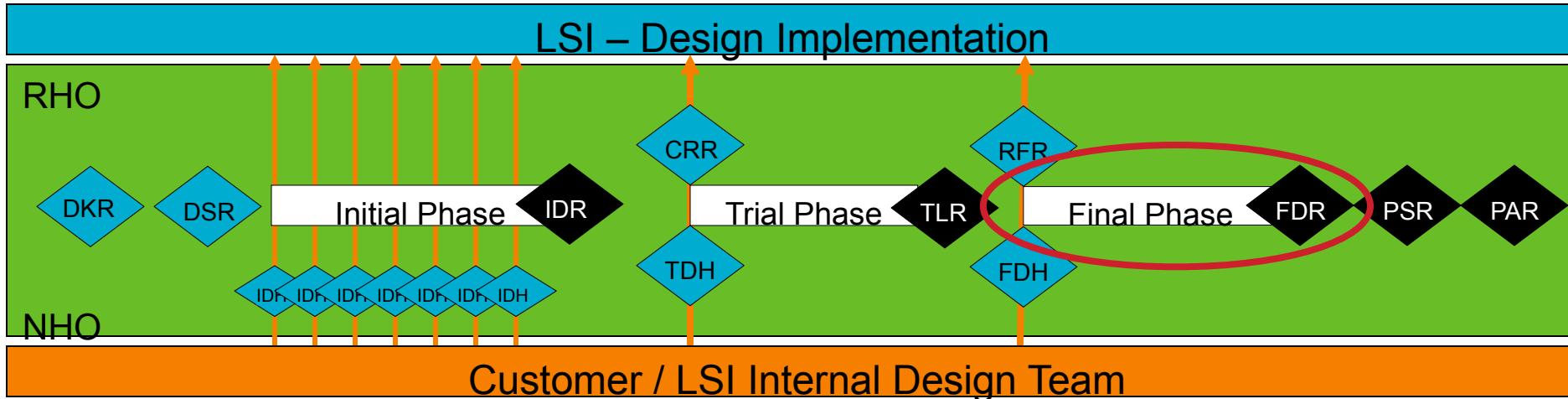
Phases: TRIAL



Phases: TRIAL

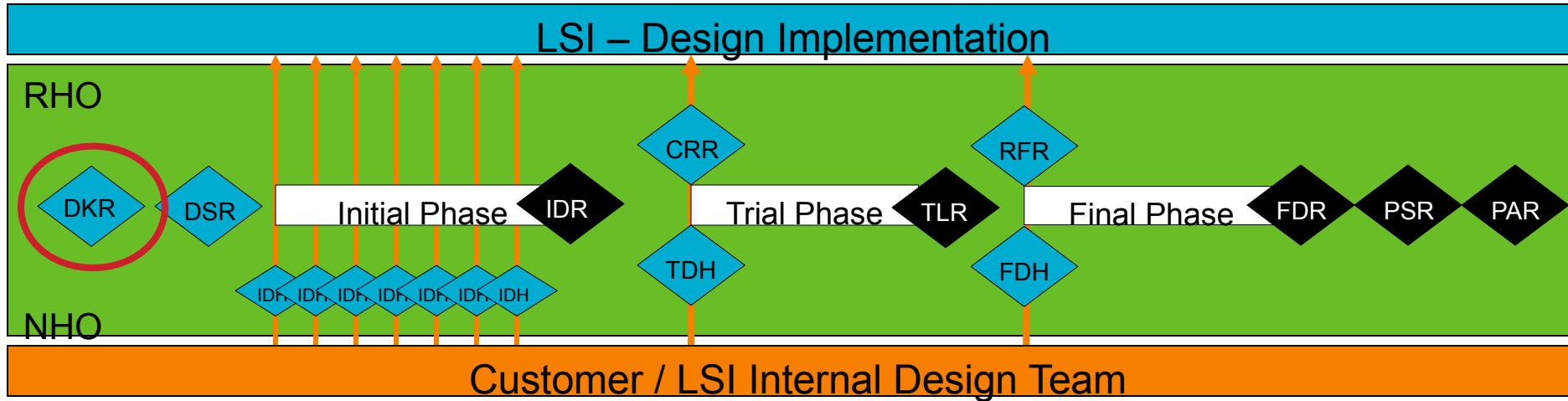
- Focuses on bringing a high quality handoff from the Customer through a trial implementation including a full detail route
 - Move even further down the implementation flow, and identify and resolve any new issues
- All of the feedback from the Initial Phase has been incorporated
- Limited customer deliveries are expected in this phase
 - Ideally there would be no iterations.
 - Trial layout is highest priority and should not be disrupted
 - Changes based on Customer verification and RTL debug are expected to be of a more limited scope such that these would not invalidate the trial
 - The possibility does exist that the Customer may find a bug which requires design changes that are significant enough in scope to justify restarting the trial
 - It is also possible that LSI may uncover design closure issues that must be addressed by the Customer in the RTL
 - It is left to the discretion of Project Lead / Technical Lead as to whether to process a new handoff or continue to learn from current handoff

Phases: FINAL



- Full Implementation and Design Closure to tape-out

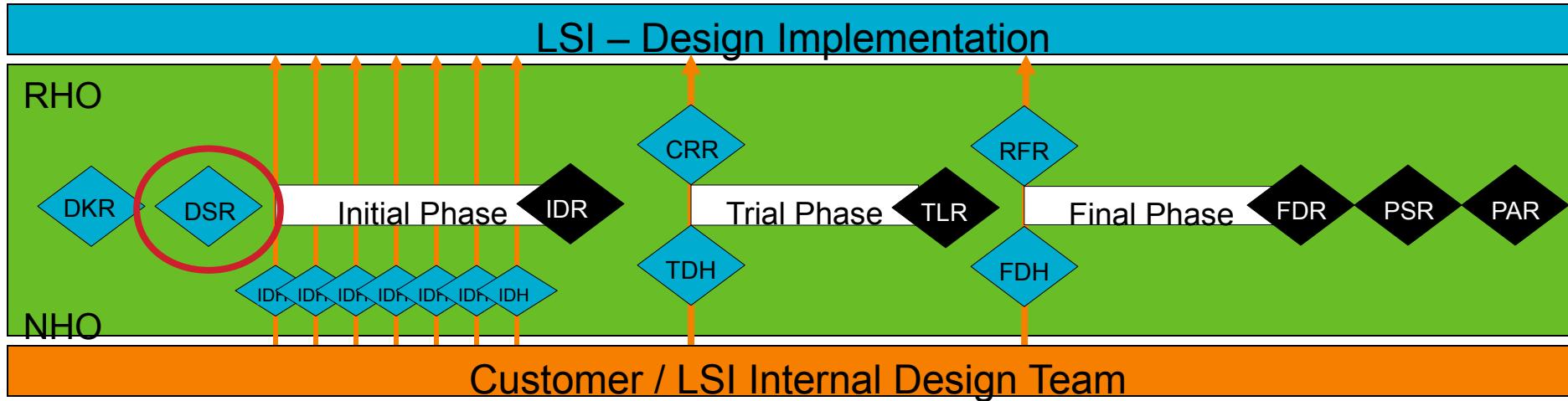
Milestones: Design Kickoff Review (DKR)



Milestones: Design Kickoff Review (DKR)

- Introduction meeting between LSI and the Customer to kick off the project
- Topics to be discussed at this meeting include such items as
 - Project Summary and Objectives
 - Design Flow and Checklist Overview
 - Statement of Work (SOW)
 - Roles and Responsibilities of the Team members
 - Schedule
 - Presentation of the Design Implementation Spec. Template which the Customer must complete
 - Change Request Procedures

Milestones: Design Specification Review (DSR)



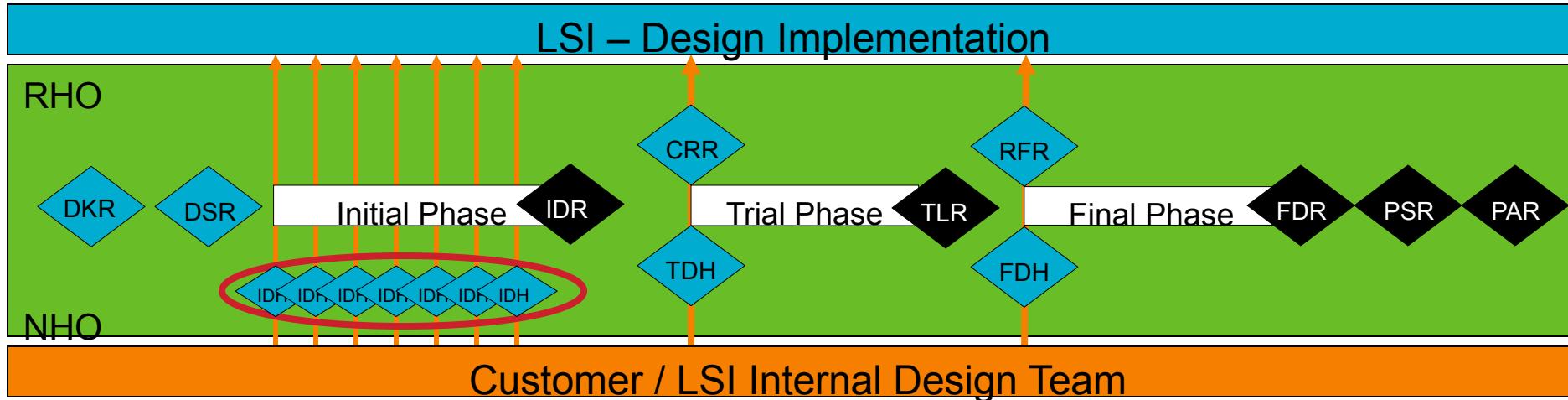
Milestones: Design Specification Review (DSR)

- Review between LSI and the Customer to freeze the contents of the Design Implementation Spec.
 - The customer is the owner of the Spec.
 - Between DKR and DSR, LSI Project Lead and/or Technical Lead supports the Customer to complete the Spec.
- DSR must be completed to start the activities of the Initial Phase
- After DSR has been accomplished, changes to the Design Implementation Spec. must be made using a Change Request

Table of Contents

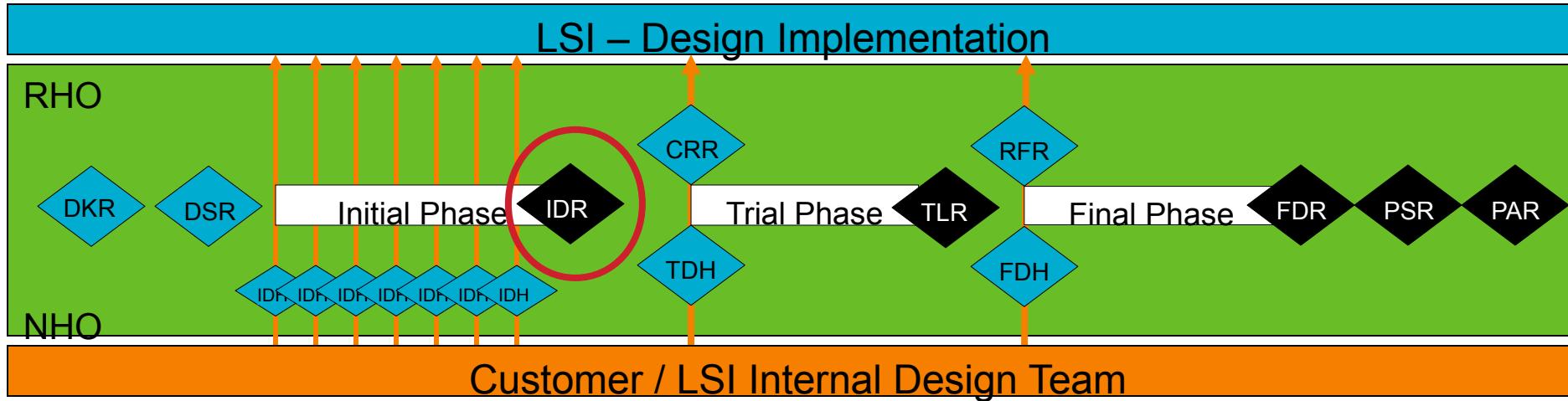
1	Introduction
1.1	Purpose of this Document
1.2	Contact Information
1.3	Glossary
1.4	Related Documents
1.5	General Design Considerations
2	Overview
2.1	Features List
2.2	Top Level Block Diagram
2.3	IP, Memories, and Design Re-Use
2.4	Testability Requirements
3	Signal IO
3.1	IO Buffer Description
3.2	Pin Descriptions
4	Operating Parameters and Reliability
4.1	Device Operating Parameters
4.2	Reliability Requirements
5	Package and Pinout
5.1	Package Description
5.2	Bonding Requirements
6	Timing Diagrams and Requirements
6.1	Clocks
6.2	Phase Locked Loops
6.3	Zero Cycle Paths
6.4	Timing Modes
6.5	False Paths
6.6	Multicycle Paths
6.7	Timing Diagrams
6.8	External Skew Requirements
6.9	Resets
7	Physical Design Considerations

Handoffs Made in the INITIAL Phase (IDH)



- Multiple releases from the Customer during the initial phase
- Entry Criteria are task oriented
 - Establish the purpose for each handoff and check against appropriate entry criteria for that purpose
 - Initial Design Handoff (IDH) Checklist

Milestones: Initial Design Review (IDR)

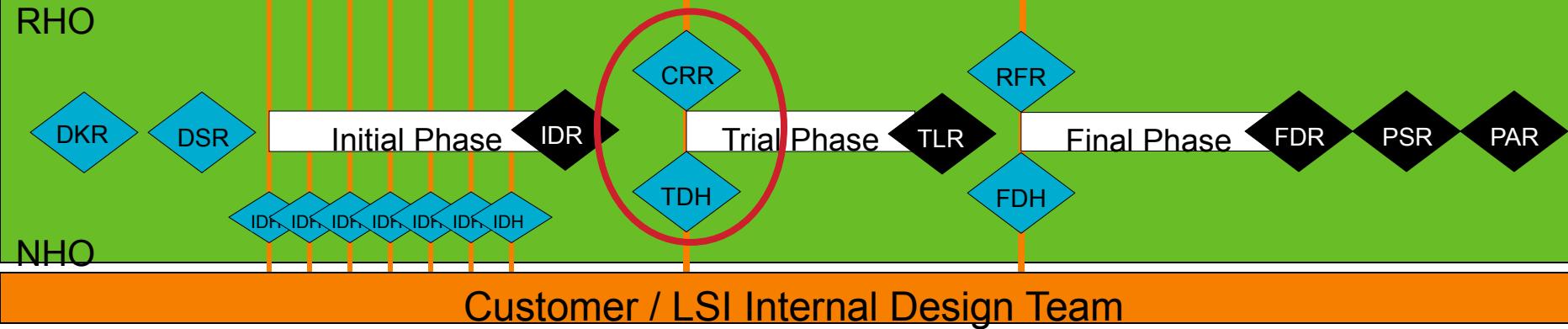


Milestones: Initial Design Review (IDR)

- End of the iterative Initial Phase
- Review that the Exit Criteria of the Initial phase have been met.
 - Multiple handoffs from the Customer have been processed
 - Final summary review of the results of the activities of each of the tasks of the Initial Phase
- Examples of major tasks that should have been completed to exit this phase include the following:
 - Test insertion
 - Test vector generation and simulation of subset of vectors
 - Package selection finalized
 - Place and Route completed enough to validate floorplan, die size and layout timing constraints
 - This does not include Detailed routing or manual routing
 - Formal verification (gate-to-gate)

Milestones: Chip-Level RTL Review (CRR) / Trial Deliverables Handoff (TDH)

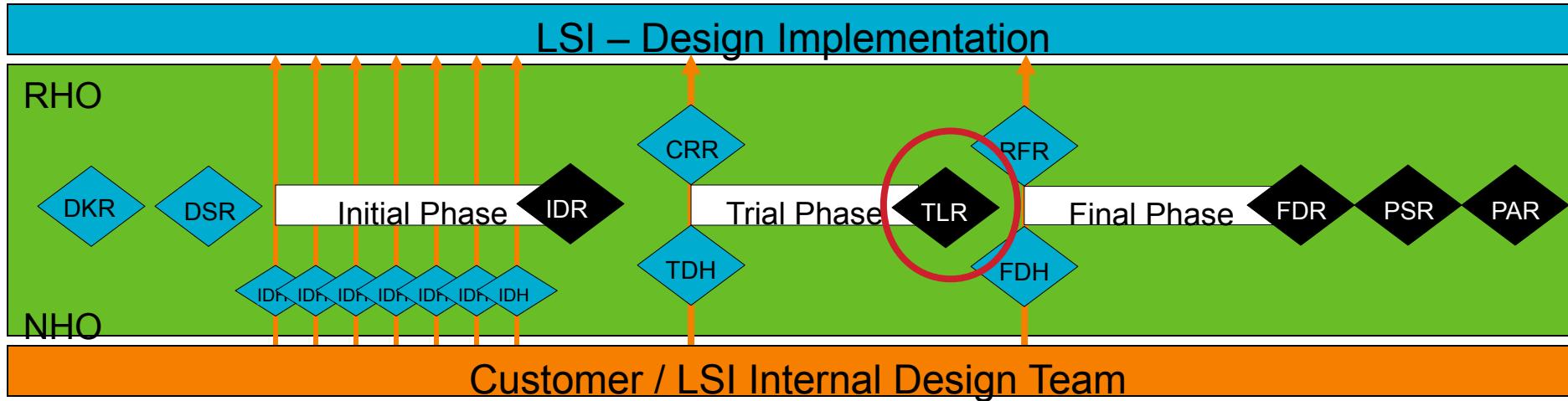
LSI – Design Implementation



Milestones: Chip-Level RTL Review (CRR) / Trial Deliverables Handoff (TDH)

- CRR marks the start of the Trial Phase for the RTL Handover Engagement
 - From a design development standpoint, it represents the transition from the functional design phase to the RTL debug phase:
 - Every function in the chip has been designed, integrated, and compiled
 - Sufficient verification has been done to prove that the design behaves correctly in its key operations (e.g. register access, data-path flows, standard functions)
 - No further feature design is intended and future RTL changes should be limited to fixing functional errors and structural design closure problems
 - Timing constraints, exceptions, and conditions should be fully specified
 - Completion of the IDR milestone is required before CRR can be accomplished.
- TDH marks the start of the Trial Phase for the Netlist Handover Engagement
 - Entry Criteria are the same as for CRR except that the deliverables are based on a netlist handoff rather than RTL

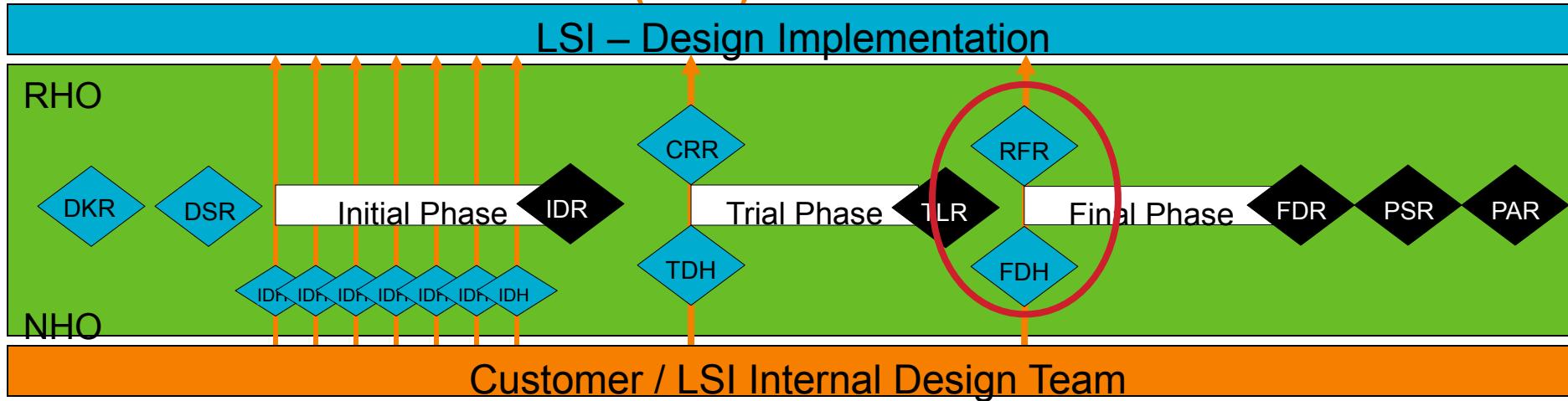
Milestones: Trial Layout Review (TLR)



Milestones: Trial Layout Review (TLR)

- End of the Trial Phase
- Review that the Trial phase exit criteria have been met.
 - Final summary review of the results of the activities of each of the tasks of the Trial Phase
- Examples of major tasks that should have been completed to exit this phase include the following
 - Test insertion of Trial netlist. All coverage targets met.
 - Test vector generation and simulation of all vectors (using unit delay)
 - Pinout Frozen
 - Place and Route completed enough to validate floorplan, die size and layout timing constraints.
 - This does include detail routing
 - STA constraint and script development
 - Formal verification (gate-to-gate, RTL-to-gate)
 - Power estimate confirmed
 - SPICE analysis of critical interfaces

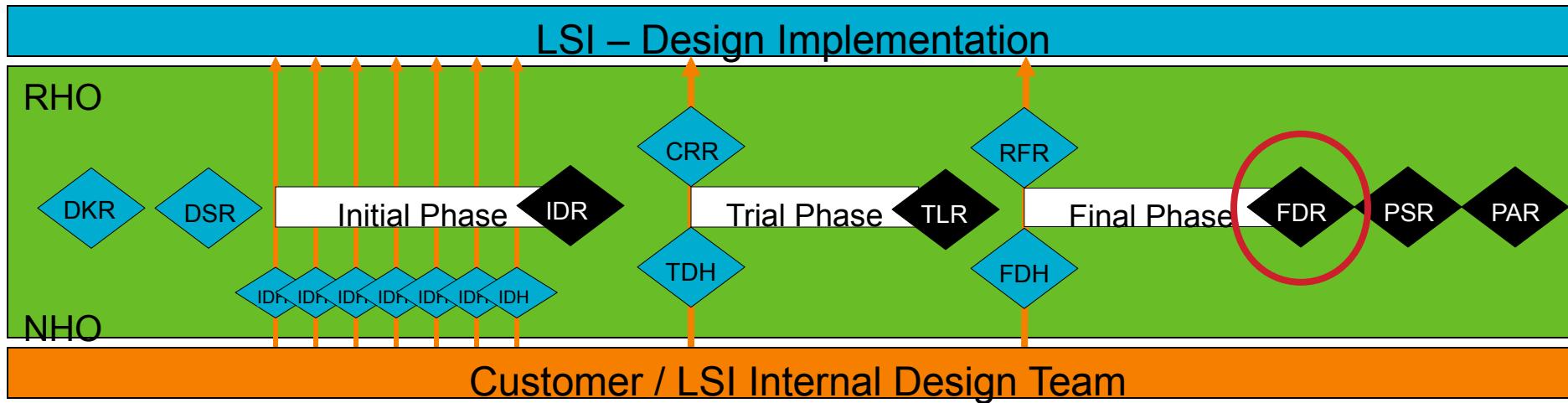
Milestones: RTL Freeze Review (RFR) / Final Deliverables Handoff (FDH)



Milestones: RTL Freeze Review (RFR) / Final Deliverables Handoff (FDH)

- RFR marks the start of the Final Phase for the RTL Handover Engagement
 - Performed when the design has reached a point where it is stable enough to start the final pass of physical implementation
 - No more changes to the design are anticipated
 - Verification may not be complete, but all “must have” tests are complete and passing.
 - Verification of “should have” / “nice to have” tests may continue after RFR.
 - Completion of the TLR milestone is required before RFR can be accomplished
 - Any changes to the RTL after this point require a formal Change Request.
- FDH marks the start of the Final Phase for the Netlist Handover Engagement
 - Entry Criteria are the same as for RFR except that the deliverables are based on a netlist handoff rather than RTL

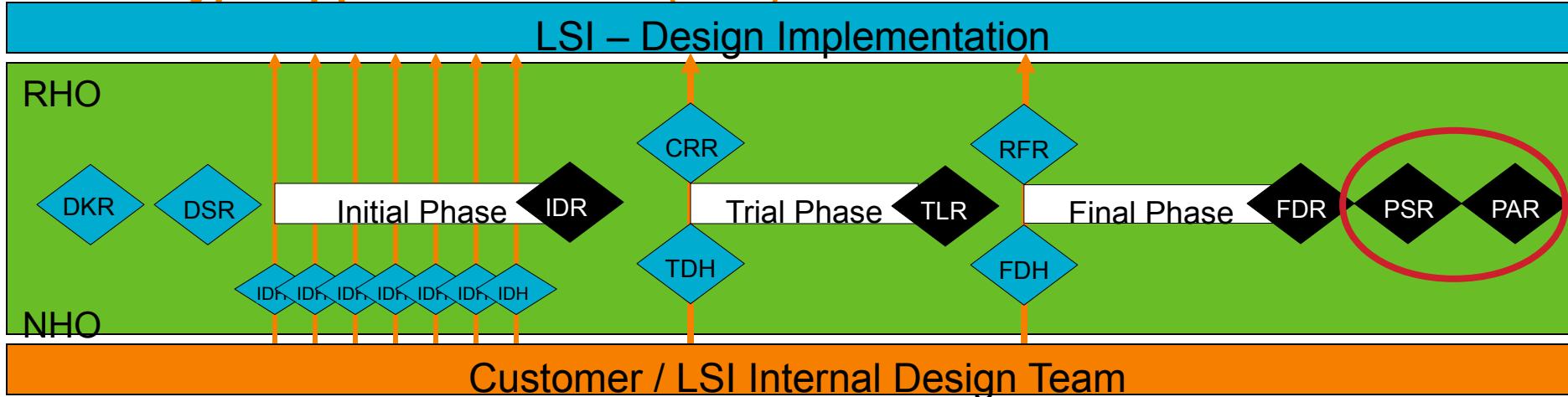
Milestones: Final Design Review (FDR)



Milestones: Final Design Review (FDR)

- The purpose of the FDR is to review the design prior to release to prototype manufacturing
- All aspects of the design must be complete
- Examples of major tasks that should have been completed to exit this phase include the following:
 - Test insertion of Final netlist. All coverage targets met.
 - Test vector generation and simulation of all vectors (using backannotated delays)
 - Final Place and Route
 - Timing closed in all functional and test STA modes
 - Formal verification (gate-to-gate, RTL-to-gate)
 - Power estimate confirmed
 - SPICE analysis of critical interfaces
 - Customer functional verification (including gate level simulations) complete

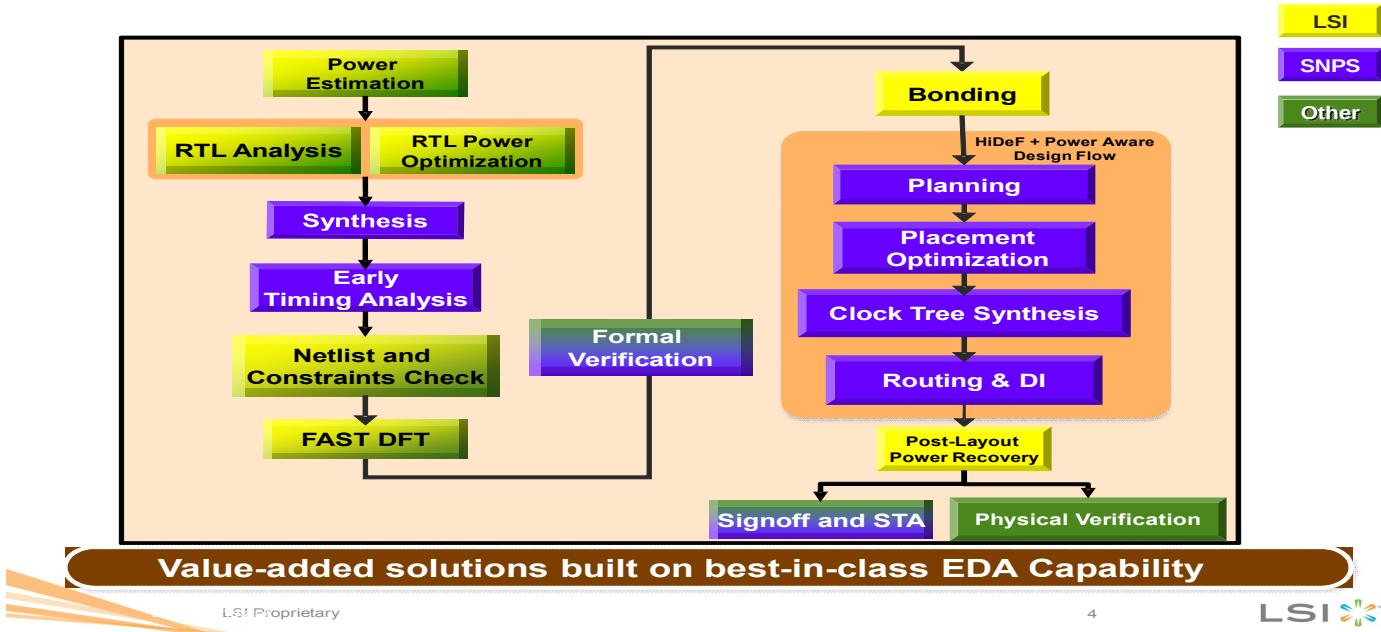
Milestones: Prototype Ship Release (PSR) / Prototype Approval Release (PAR)



- PSR accomplished by Test Engineering when protos have shipped
 - There is no review meeting or associated documentation
- PAR accomplished when the customer approves the prototypes for production
 - Document owned by the Customer Service organization for this purpose.

ASIC FLOW at LSI

LSI 40nm and 28nm Design Flow



Netlist Audits

- Anytime a netlist is delivered by customer or created Audits needs to be run
- What is an audit?
 - Checks to confirm everything is in place and is good to go to the next step
- Netlist Audits help identify the problems in design
 - Efficiency Checks hampering Designer and Tool Efficiencies
 - DRC that risks the quality of the implemented design
- Tool used at LSI is called Spotlight
 - Uses Atrenta Tool called Spyglass for its infrastructure

Netlist Audits - Spotlight

- Has 5 categories of audit to check problems
 - Hierarchical Design Flow
 - Enforces LSI's Flow Guidelines
 - FAST Flow
 - Checks for LSI's design-for-test (DFT)/FAST Flow
 - IP
 - Verifies the connectivity to LSI Digital and Analog IP
 - Topology
 - Assists with Design Setup
 - Layout and DRC
 - Ensures Silicon Quality

When will you run audits in your Design?

DFT (Design for Test)

- What do you expect when you buy a new gadget?
 - Work correctly right out of the box.
- What is Test?
 - It is impossible to guarantee every unit produced is perfect
 - Most manufacturers prefer to find the problems before it ships
- What is DFT?
 - Basic Test , a go/no-go test, yields just one bit of Information
 - Working or not. If not working throw it out....
 - Instead of throwing it out, a more diagnostic test to find why the subsystem is failing needs to be done.

A term applied to design methods which lead to a more thorough and less costly testing

DFT - Benefits

- The outcome of a go/no-go test is more believable.
 - If fewer systems with hidden faults are shipped, fewer customers get upset, which yields obvious economic as well as psychological benefits
- Diagnostic tests run faster and produce more accurate results.
 - This reduces the cost of salvaging a subsystem that fails the go/no-go test, making it possible to manufacture more systems at lower cost.
- Both go/no-go and diagnostic tests require less test-engineering time to develop.
 - Although the savings in test-engineering time may be offset by added design-engineering effort to include DFT, any increase in overall product development cost usually can be offset by decreased manufacturing cost

DFT – Basic Tests & Flow

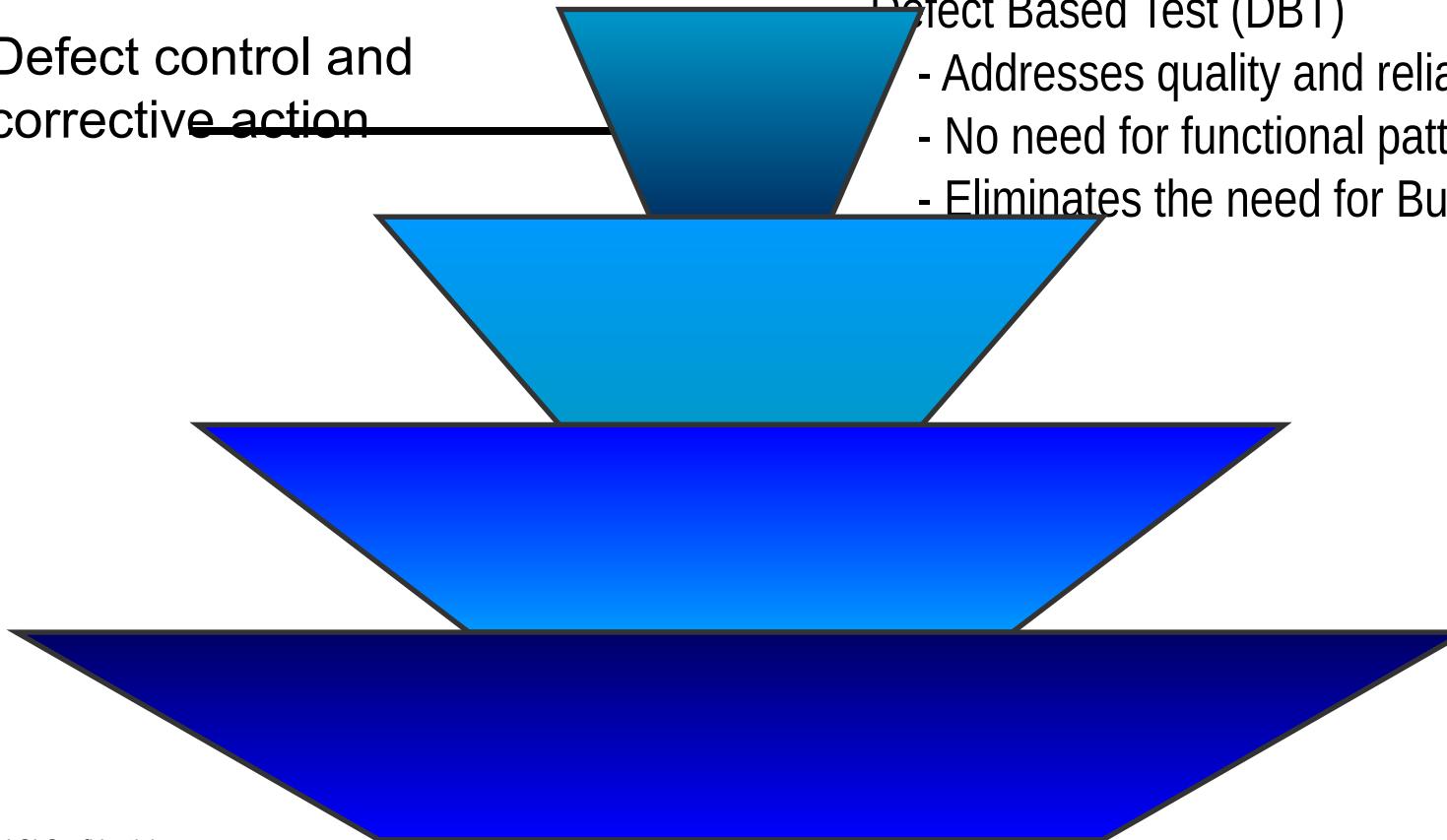
- Various Tests done in an ASIC
 - Scan Tests
 - To check the Standard Cell Logic present in the design
 - Boundary Scan Tests
 - To check the IO 's in the Design
 - Memory BIST & BISR
 - To check the memories in the design
 - IP Tests
 - To check the different IP's instantiated in the design
- LSI flow to generate all the tests and verification is called FAST

LSI's Test Strategy

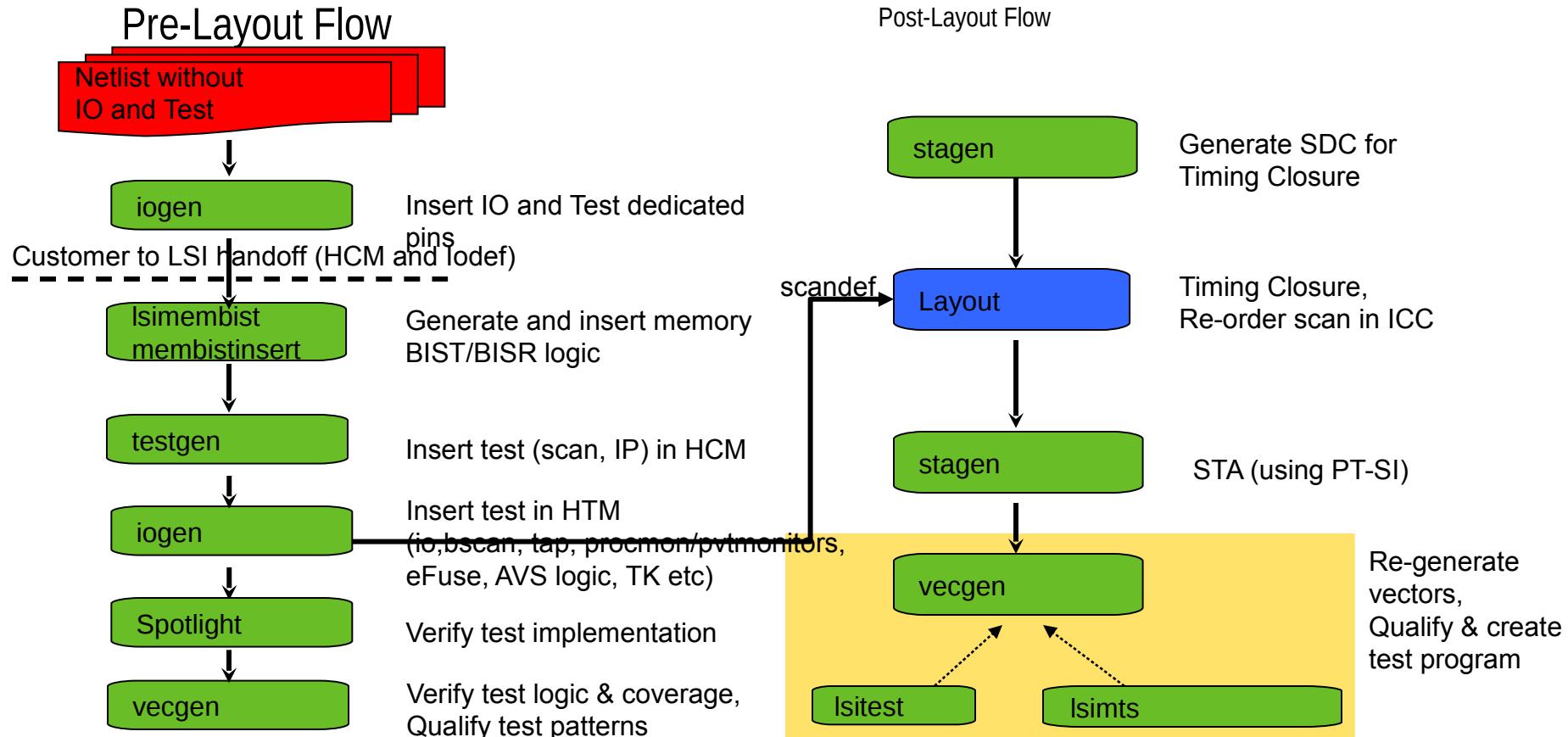
Defect control and corrective action

Defect Based Test (DBT)

- Addresses quality and reliability
- No need for functional patterns
- Eliminates the need for Burn-In

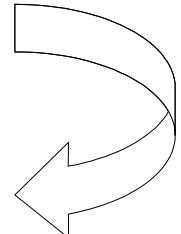


FAST – Flow for Automated Standardized Test Flow





Storage. Networking. **Accelerated.**TM



Static Timing Basics



Accelerate.



Objectives

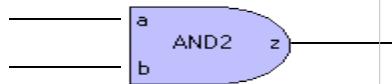
- Dynamic Verification
- Static Verification
- Static vs. Dynamic Analysis
- Delay Calculation Techniques

Why STA?

Dynamic Verification

- Checks that user generated *input patterns* cause an **expected** output response.
- All paths may not be tested.
- Simulation may not catch all faults.
- No easy way to determine margin or to identify timing problems.

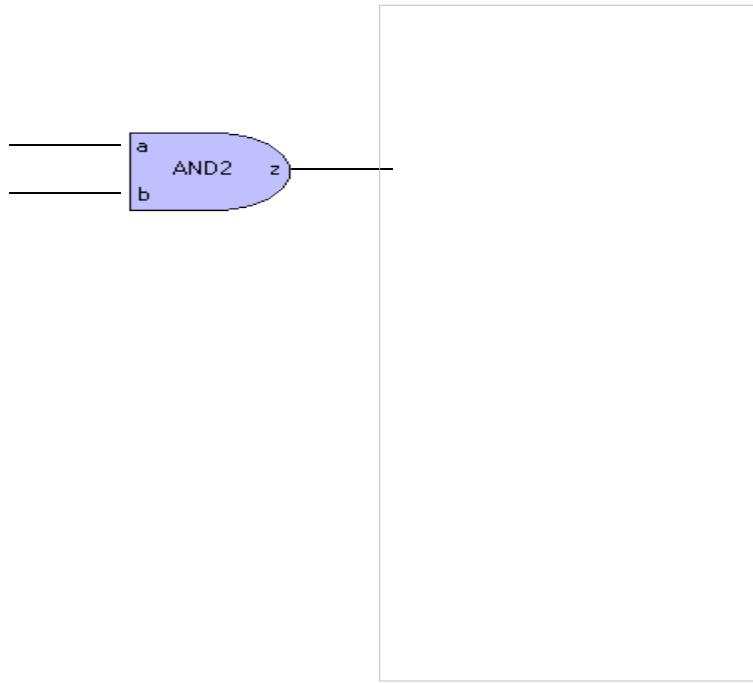
Dynamic Verification



A	B	Z	Vector #	Description
0	0	0	1	Initialization
0	1	0	2	No Transition
1	0	0	3	No Transition
1	1	1	4	Test I->h Transition on B -Z

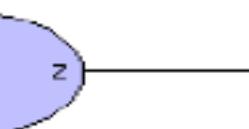
NOT DONE!

Dynamic Verification



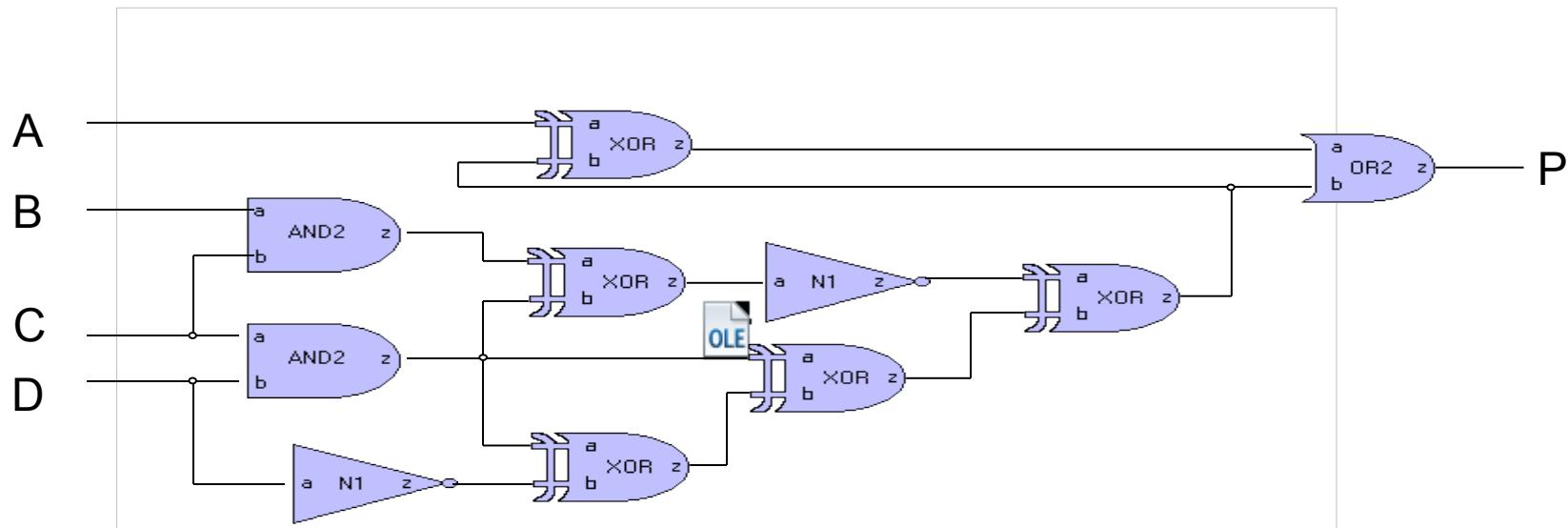
A	B	Z	Description	Vector #
0	0	0	Initialization	1
0	1	0	No Transition	2
1	0	0	No Transition	3
1	1	1	Test I->h Transition on B-Z	4
1	0	0	Test h->I Transition on B-Z	5
0	1	0	No Transition (or glitch)	6
0	0	0	No Transition	7
0	1	0	No Transition	8

NOT DONE!



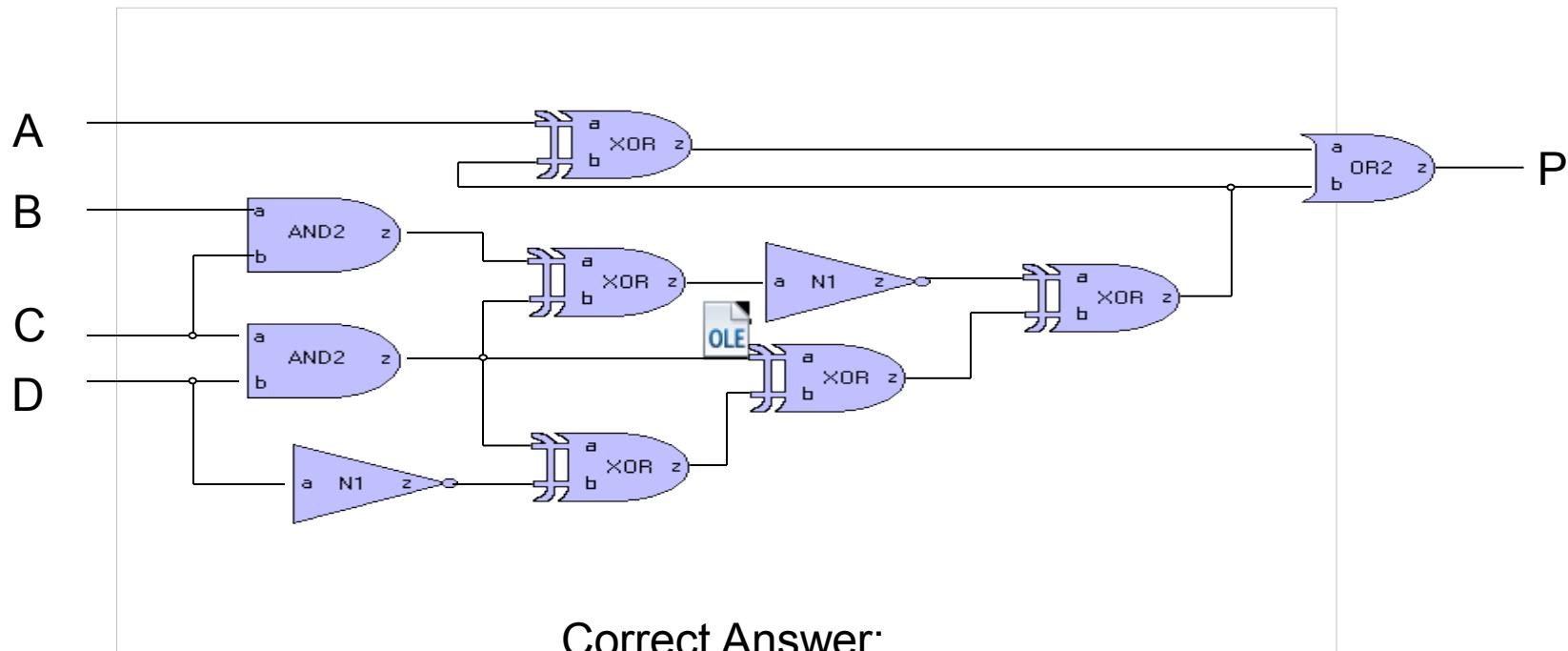
0	0	0	Initialization
0	1	0	No Transition
1	0	0	No Transition
1	1	1	Test I->h Transition on B-Z
1	0	0	Test h->I Transition on B-Z
0	1	0	No Transition (or glitch)
0	0	0	No Transition
0	1	0	No Transition
1	1	1	Test I->h Transition on a-z

Dynamic Verification



Exercise 1. Write functional patterns to check all transitions for the 4 bit parity generator shown.

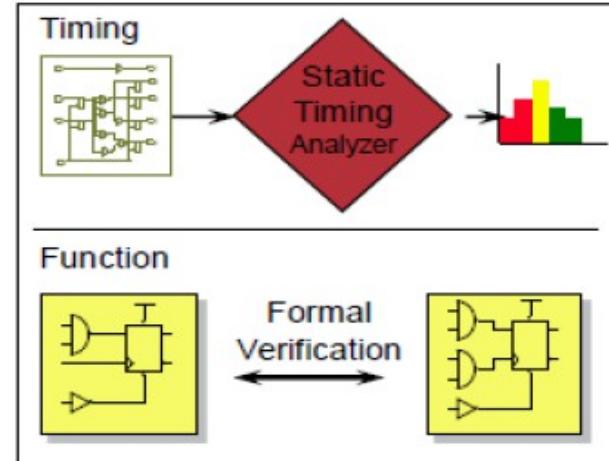
Dynamic Verification



Correct Answer:
This is a waste of my time!

Static Verification

- Method of computing the expected behavior of a digital circuit without requiring simulation
 - Examples : Static Timing Analysis & Formal Verification
 - Faster than Simulation
 - Analyzes the design without vectors



(Source Synopsys)

Verification Techniques & its Benefits

- Any design implemented needs to be verified
 - Static Verification
 - Orders of magnitude faster than Gate Level Simulation
 - Analyzes Design exhaustively without vectors
 - Capacity for millions of gates
 - Focus is on the Design and not on the vectors
 - Dynamic Verification/Gate Level Simulation
 - Coverage Dependant on Vector Quality
 - Limited Performance
 - Testbench Development
 - Not acceptable for full ASIC timing.
 - Used for filling in gaps that static timing analysis cannot handle.
 - Good to sanity check assumptions made in Static Timing Analysis
 - Coverage Dependant on Vector Quality

Introduction to Static Timing Analysis

- Static alludes to the fact Timing Analysis is carried out in an input –independent manner
 - All paths checked, even paths that cannot be exercised in reality
- Purports to find Worse Case Delay of circuit over all possible input combinations
- Plays a vital role in facilitating the fast and reasonably accurate measurement of circuit timing
- Little knowledge of design needed

Static Timing Analysis Capabilities

DOES

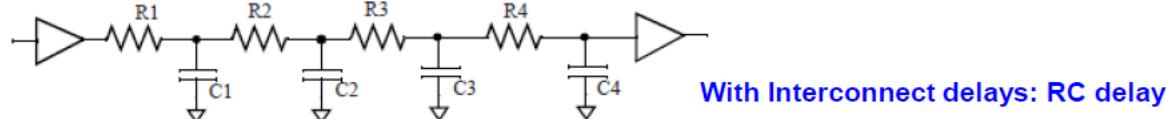
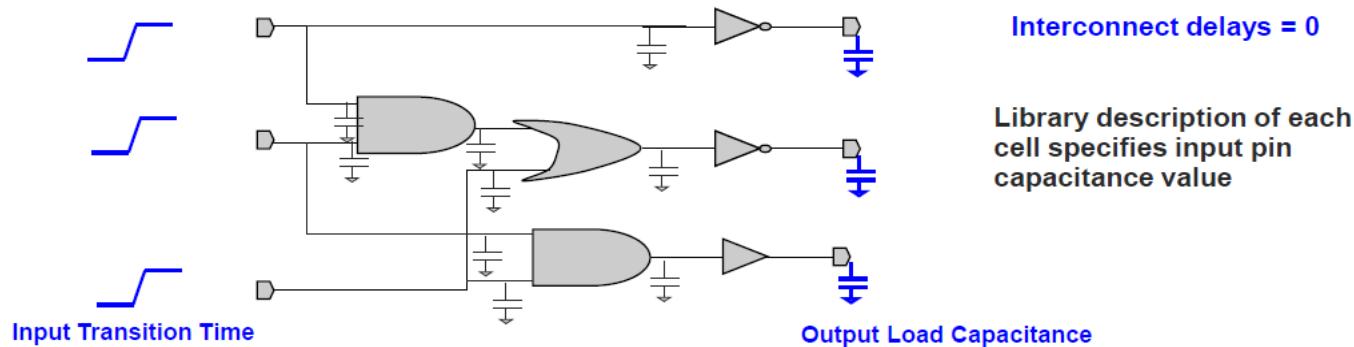
- Compute expected timing of a Digital Circuit without using vectors
- Computes timing in a short time
- Simpler than Simulation
- Analyzes Cross Talk and Noise
- Calculates effects of chip Fabrication Process variation

DOES NOT VERIFY

- Reset Sequence
- X- Handling
- PLL Settings
- Asynchronous clock domains
- IO interface – Read/Write Interface
- Interface between Analog & Digital
- Functionally False Paths
- Functional Behavior across clock cycles

Delay Calculation Basics

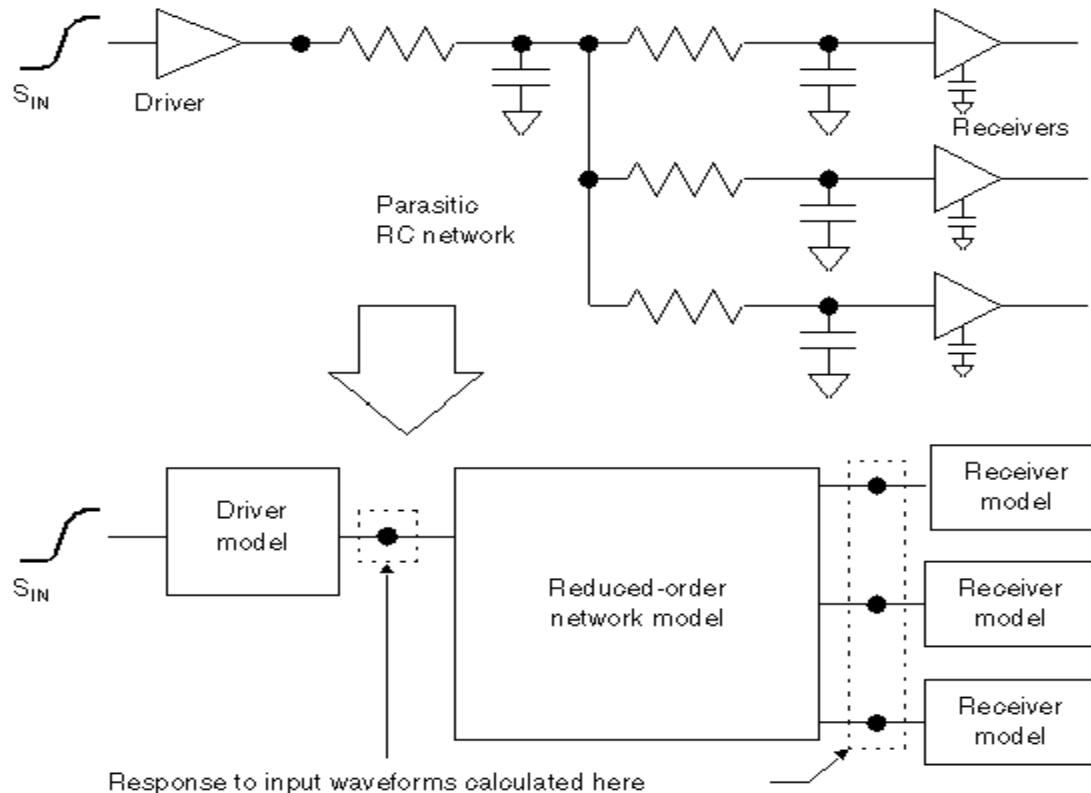
- Delay calculation of a timing Path
 - Calculating the time to propagate a transition across a net or cell



Delay Calculation

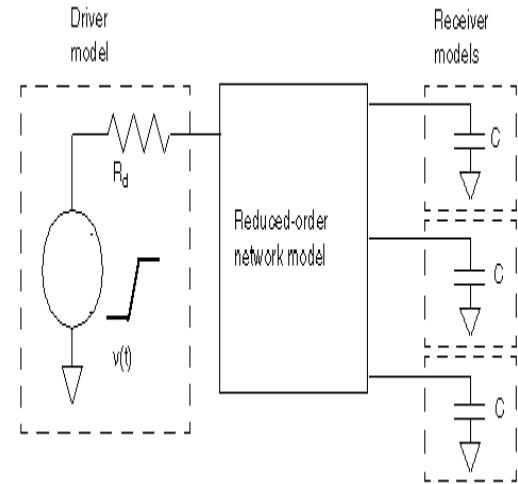
- A tool must accurately calculate the delay and slew (transition time) at each stage of each timing path.
- A stage consists of
 - a driving cell,
 - the annotated RC network at the output of the cell,
 - the capacitive load of the network load pins.
- The goal is to compute the response at the driver output and at the network load pins, given the input slew or waveform at the driver input, using the least amount of runtime necessary to get accurate results.

Models used to Calculate Stage Delays and Slew



Delay Calculation : Non Linear Delay Models (NLDM)

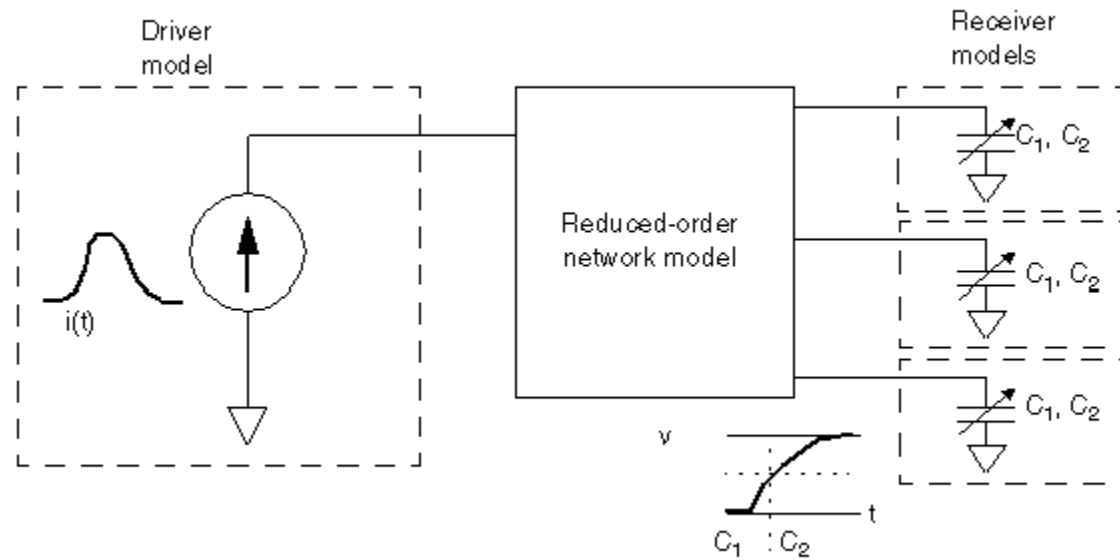
- Driver Model
 - Drive Resistance R_d
 - Ramp Start Time T_z
 - Ramp duration t
- Issues
 - Receiver Model does not account for Miller Effect
 - Driver Model
 - When the drive resistor is much less than the impedance of the network to ground, the smoothing effect is reduced, potentially reducing the accuracy of RC delay calculation
- Receiver Model
 - A capacitor that represents the load capacitance of the receiver input
 - Different caps for different conditions
 - Does not account for Miller Effect



Delay Calculation – CCS Model

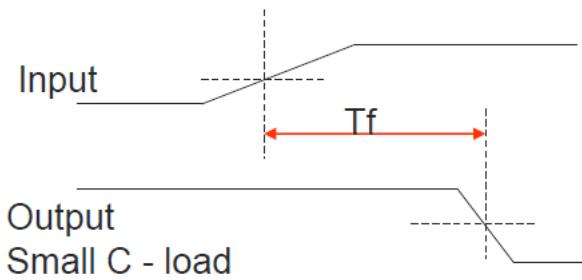
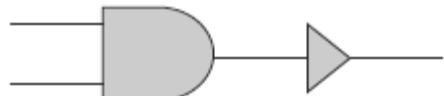
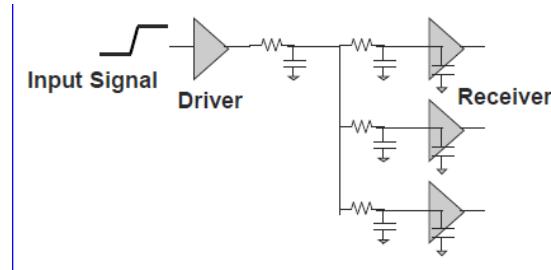
- With the advent of smaller nanometer technologies, the CCS timing approach of modeling cell behavior has been developed to address the effects of deep submicron processes
- The driver model uses a time-varying current source,
 - The advantage of this driver model is its ability to handle high-impedance nets and other nonmonotonic behavior accurately.
- The CCS timing receiver model uses two different capacitor values rather than a single lumped capacitance
 - The first capacitance is used as the load up to the input delay threshold
 - When the input waveform reaches this threshold, the load is dynamically adjusted to the second capacitance value
 - This model provides a much better approximation of loading effects in the presence of the Miller Effect.

Delay Calculation – CCS Models



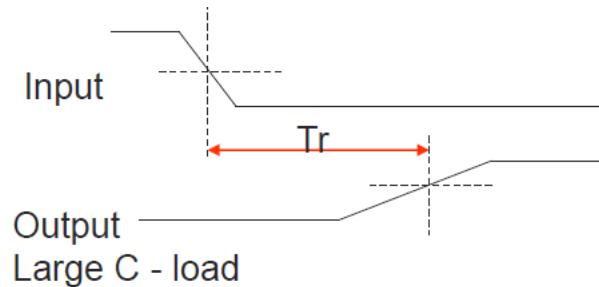
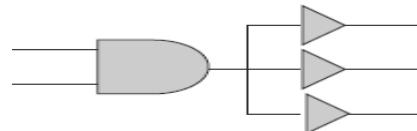
Overview of Delay Calculation and Linear Model

- Delay of a cell depends on
 - Input Transition
 - Output Load



Output
Small C - load

Output transition improvement

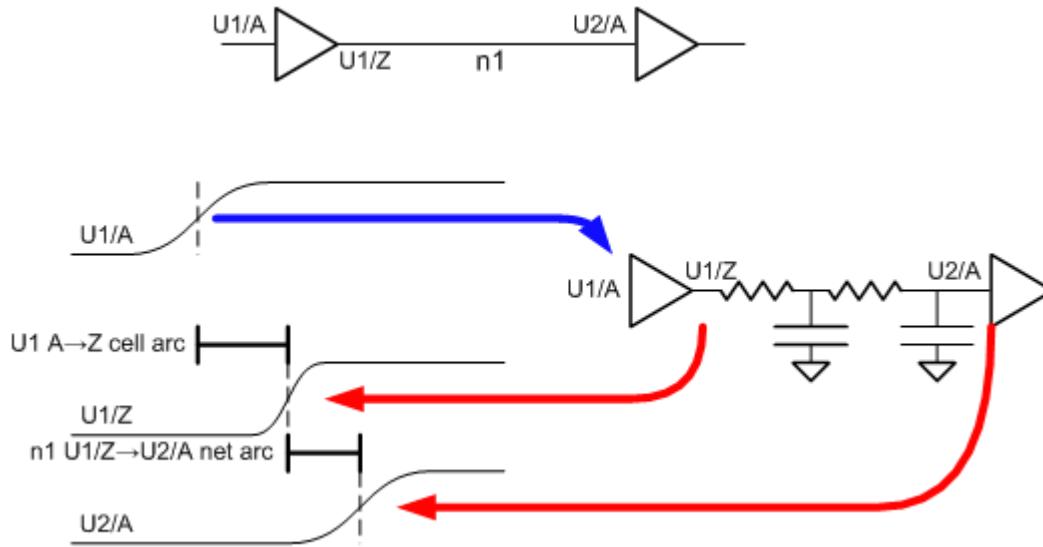


Output
Large C - load

Output transition degradation

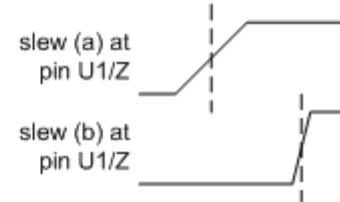
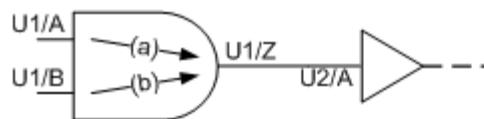
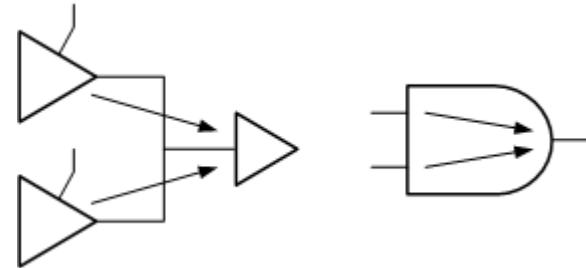
Slew Propagation – Simple Model (buffers)

- Timing paths consist of a series of cells and nets connected together. The delays of the cells and nets represent the amount of time it takes for a signal transition (or edge) to propagate across those cells or nets



Slew Propagation – Multiple Arcs

- What happens when Multiple Slews arrive at a common point?
 - Must Choose one of the slews to propagate forward
 - These points are called Slew Merge Points



- Arc a, arrives faster but with a slow transition (worst slew)
- Arc b, arrives slower but with a fast transition (worst arrival)

Worst Slew Propagation

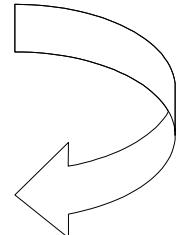
- In this method worst slew is chosen and propagated forward.
 - This is the slowest (numerically largest) slew for max delays, and
 - the fastest (numerically smallest) slew for min delays.
 - For our example circuit, we would propagate the slower slew (a) forward into U2/A for a max-delay calculation.
- This is the default mode for PrimeTime and PrimeTime SI.
 - The resulting timing correctly bounds (that is, it will never be optimistic) the analysis for sub-critical paths, although the critical path is more accurate in worst arrival mode
 - The memory and runtime requirements are less than worst arrival
 - The transition times selected for signal integrity effects correctly bound the analysis

Worst Arrival Propagation

- the slew with the worst arrival time is chosen and propagated forwards
 - This is the latest-arriving slew for max delay
 - the earliest-arriving slew for min delays.
 - the faster slew (b) arrives last at U1/Z, and would be propagated forward into U2/A for a max-delay calculation.
- enables us to track slews on a per-clock domain basis
 - since arrival times are measured against a reference launching event (our clock edge)
- The memory and runtime requirements of `worst_arrival` are somewhat increased over `worst_slew`.



Storage. Networking. **Accelerated.**TM



Constraints

Accelerate.



Agenda

- Constraints Rudiments
- Constraint Definitions
 - Clock
 - Input and output delay
 - Constants
 - MultiCycle & False Paths
 - Disable Timing
- 13 Most Important Golden Rules for Constraints

Constraints Rudiments

- A Design constraint refers to some limitations on the conditions under which an ASIC is developed
 - Clocks
 - Input and Output Delays
 - Multicycle & False Paths
 - Load and Transition
- Usually available in the Design Implementation Spec
- Needs to be cross-verified with the constraints delivered with the design netlist
 - TCL format or SDC format

Constraint Definitions

- Clock Definition
- I/O Timing
- Constants (`set_case_analysis`)
- False Paths
- Multicycle Paths (MCPs)
- `Disable_timing`

Types of Clocks

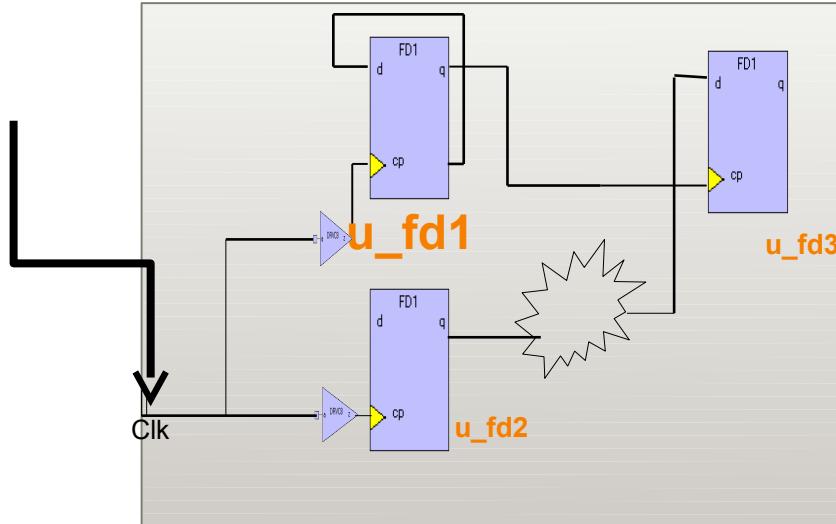
STA is dictated largely by the Design Clocks

Faster Easier Debugging of timing violations with familiarity of the design clocks!

3 Types of Clocks can be found in the Design
Synchronous Clocks
Asynchronous Clocks
Exclusive Clocks

Basic Facts of Clocks

- Master Clocks are usually defined at input ports
 - Or o/p of black-boxes
- Clocks are propagated through Combinational Logic
- Generated Clocks – Internally derived Clocks
 - STA tools cannot identify the internally derived Clocks – Why?
 - User needs to define the clock as a constraint



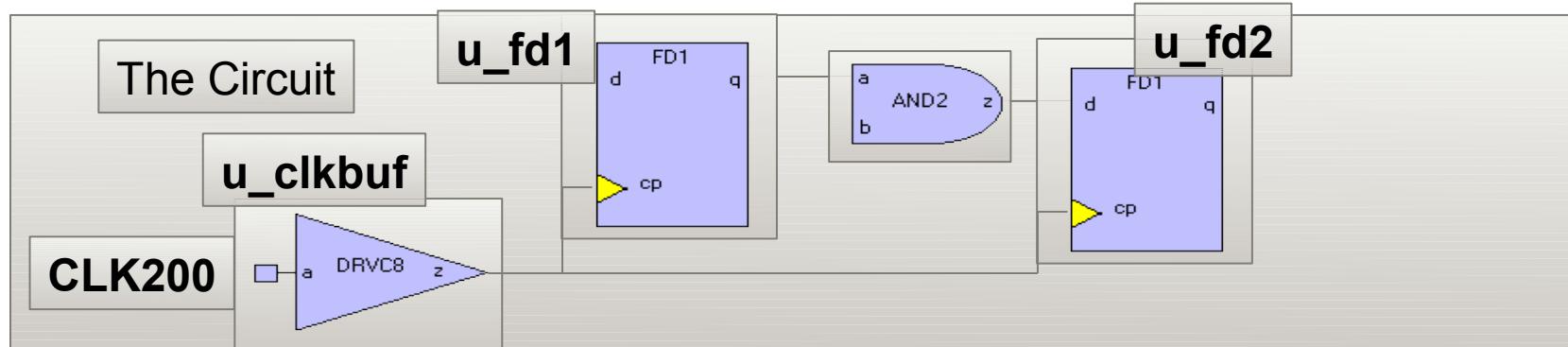
Are there any internal pins which can be clocks here?

Clock Definitions – Generated Clocks

- Only limited **REAL** sources exists on any design
 - Input pins
 - PLL outputs / analog devices
 - Coreware (maybe)
- All other clocks are *generated* clocks which should be derived from one of the **ULTIMATE** sources
 - Most *generated* clocks will be defined at the output of a register Q pin
 - Although dividing logic (odd division) may require the output of a cell instance
 - REAL source clocks CANNOT propagate through register cells, therefore, creating generated clocks is required
 - PT and ICC will correctly calculate the insertion delay (latency) properly for all generated clocks

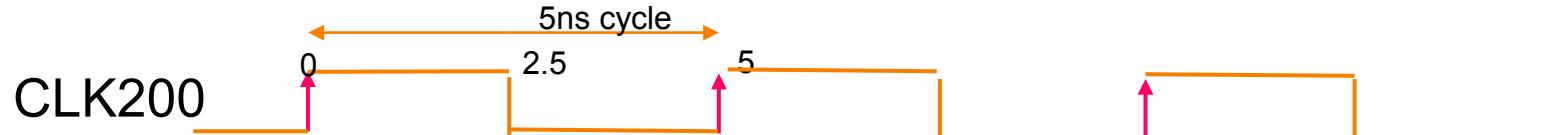
VERY IMPORTANT to know all the clocks in the Design!

Clock Definitions

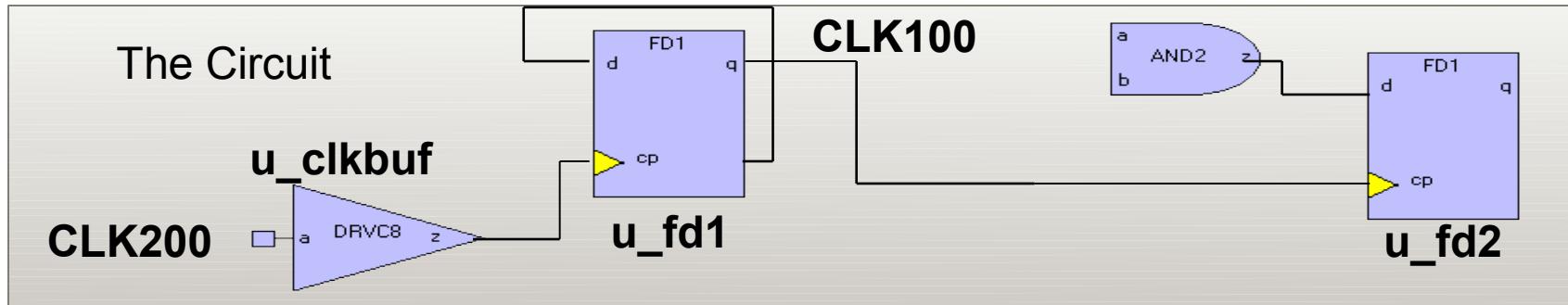


```
create_clock -name clock_name -period cycle_time -waveform  
                rise_time fall_time   source_point
```

```
create_clock -name CLK200 -period 5.0 -waveform { 0 2.5 } u_clkbuf/A
```



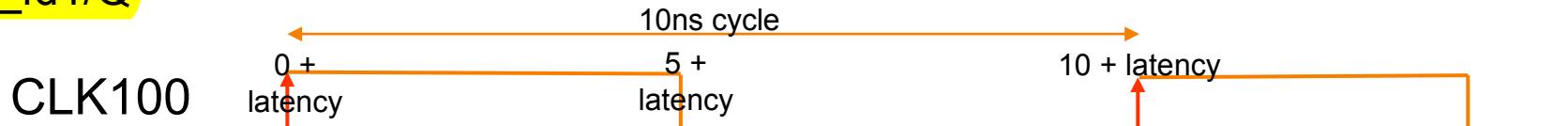
Generated Clock Definitions



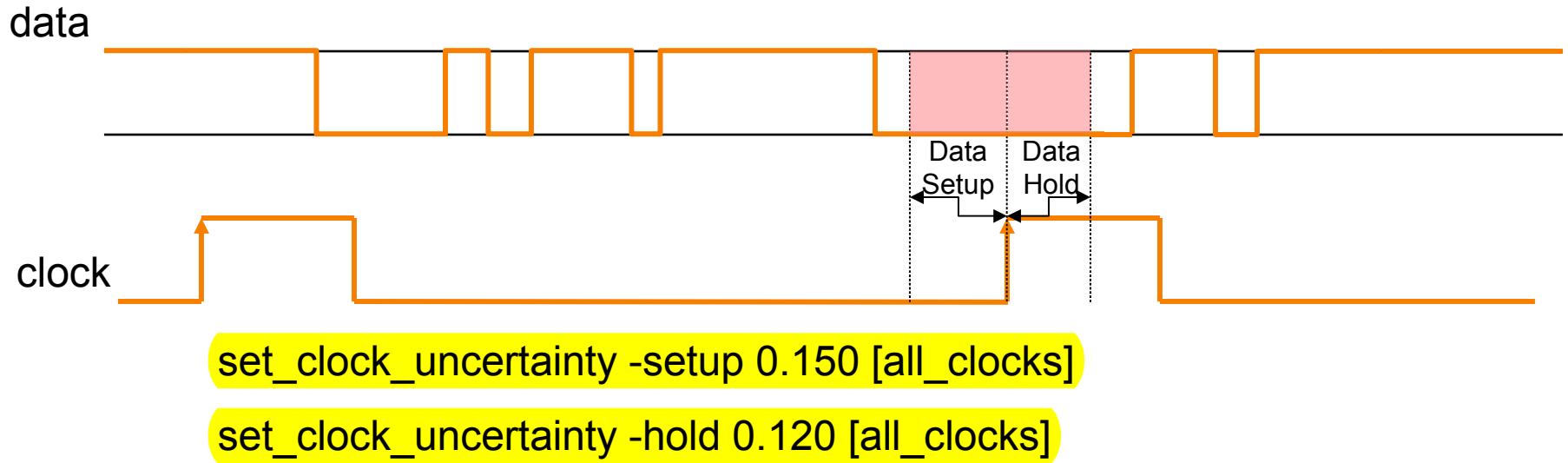
```
create_generated_clock -name clock_name -divide_by divide_by  
-source real_source_point    generated_clock_source_point
```

EXAMPLE:

```
create_generated_clock -name CLK100 -source u_clkbuf/A -divide_by 2  
u_fd1/Q
```



Clock Uncertainty Definitions

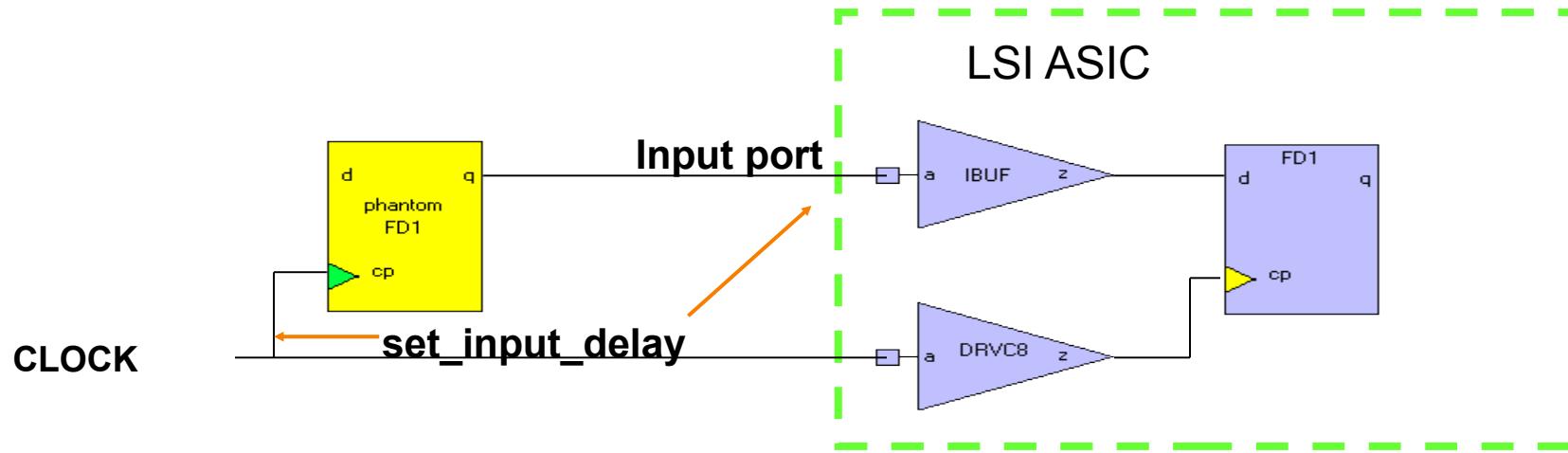


- Usually defined for jitter effects on clocks.
- Translates into stricter setup/hold margin requirements

I/O Constraints

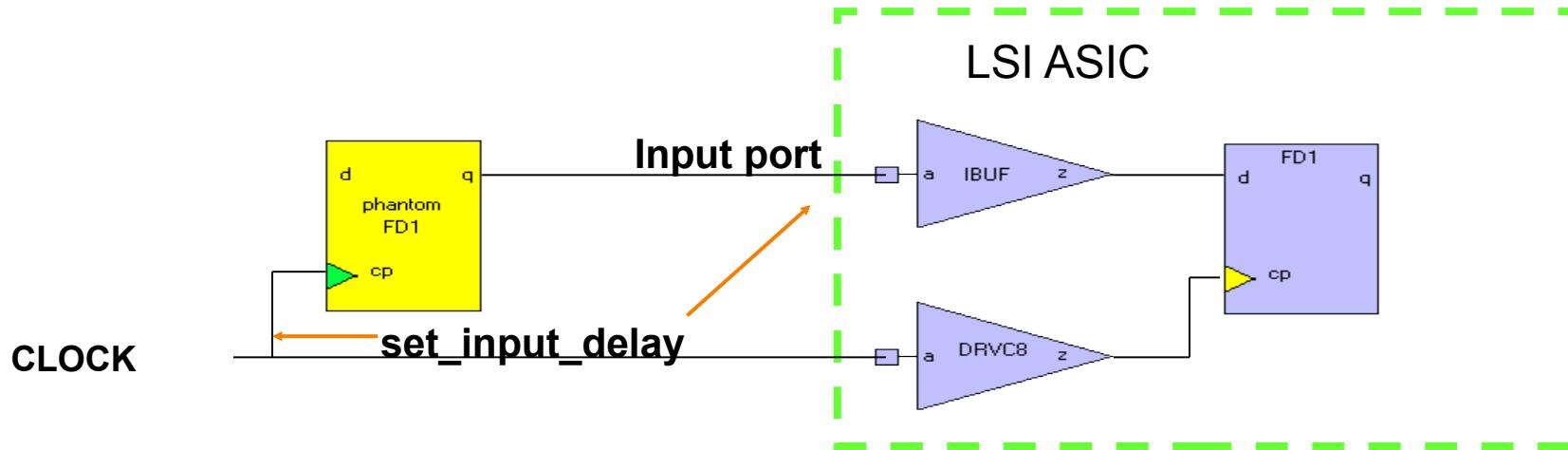
- ASIC inputs are constrained with the *set_input_delay* command in PrimeTime
- ASIC outputs are constrained with the *set_output_delay* command in PrimeTime
- Bidirect ports use both *set_input_delay* and *set_output_delay*

set_input_delay



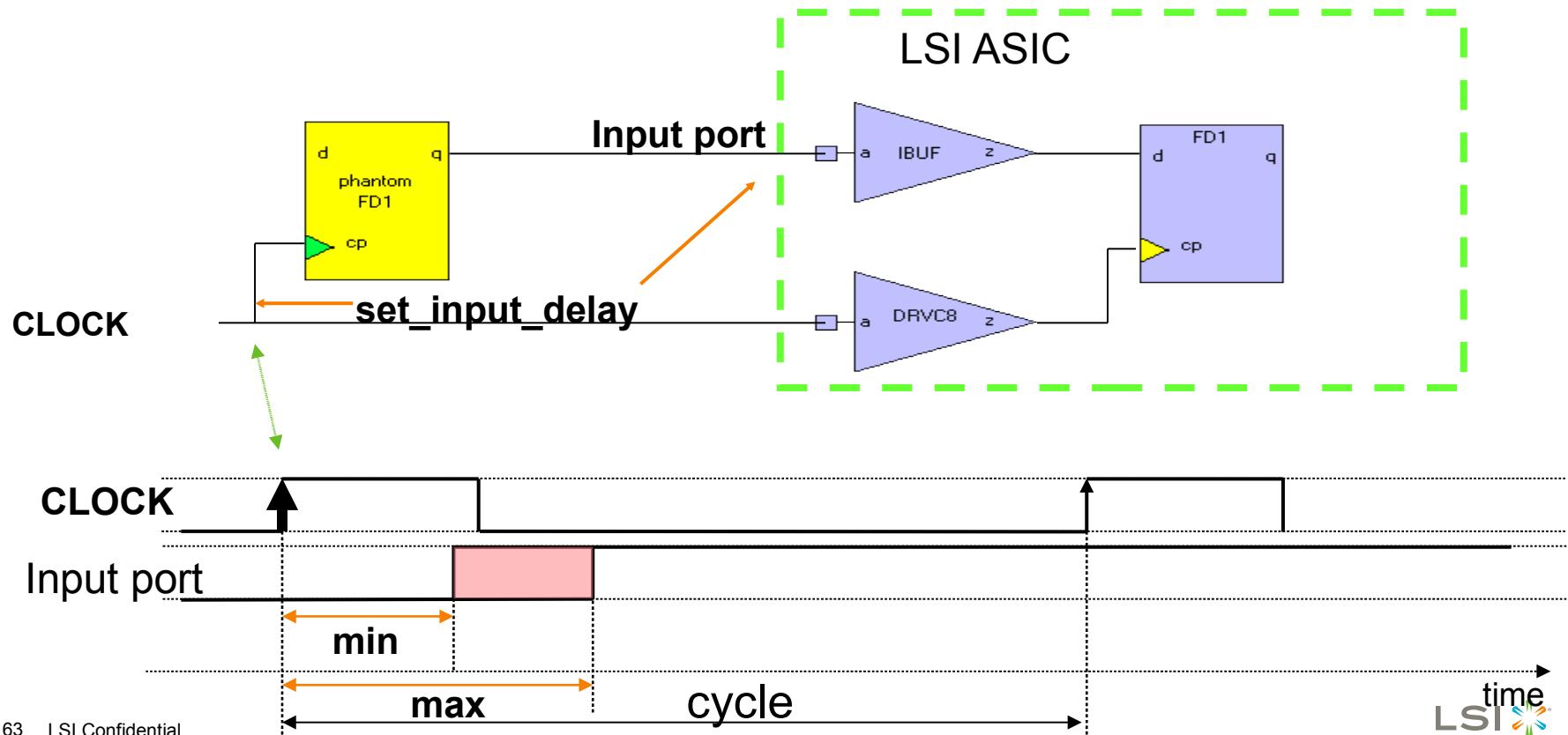
set_input_delay can be considered the delay from the rising edge of CLOCK at the “phantom” FD1 through its “q” pin and to the Input Port of the ASIC.

set_input_delay

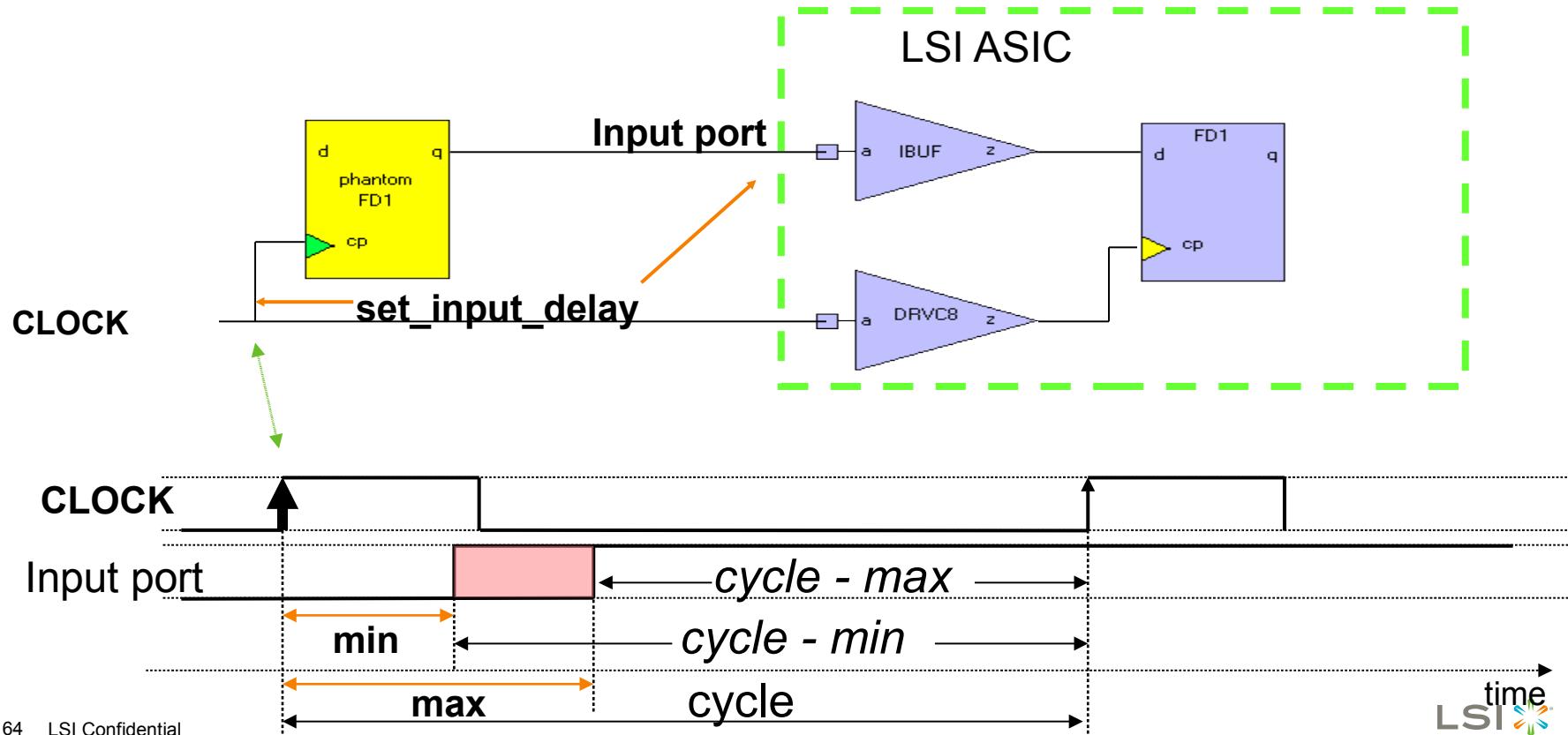


- `set_input_delay -min` specifies the fastest path (or earliest arrival) of the signal at the input port.
- `set_input_delay -max` specifies the slowest path (or latest arrival) of the signal at the input port.

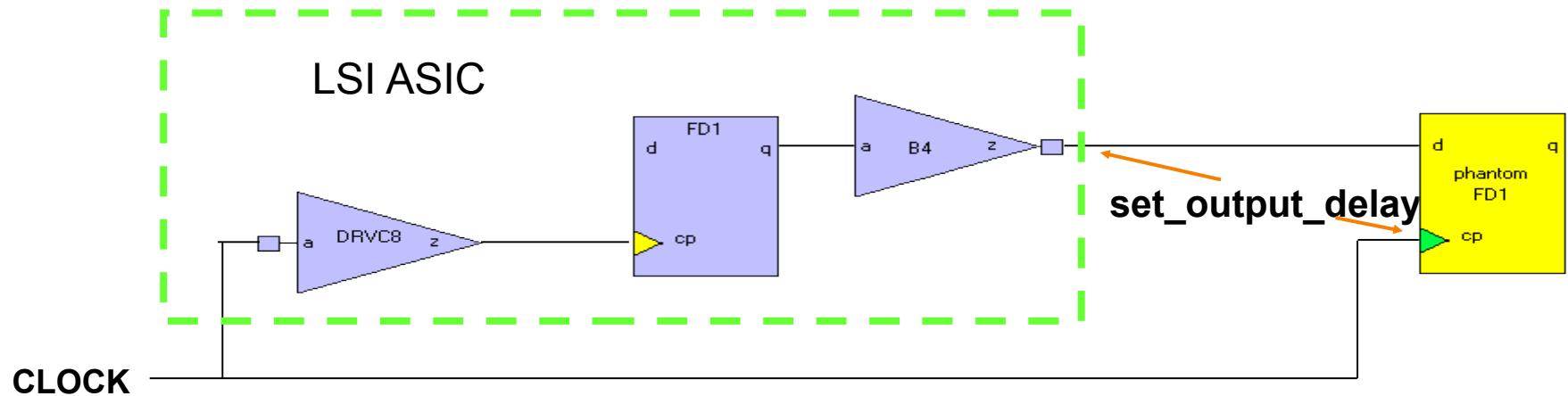
set_input_delay



set_input_delay

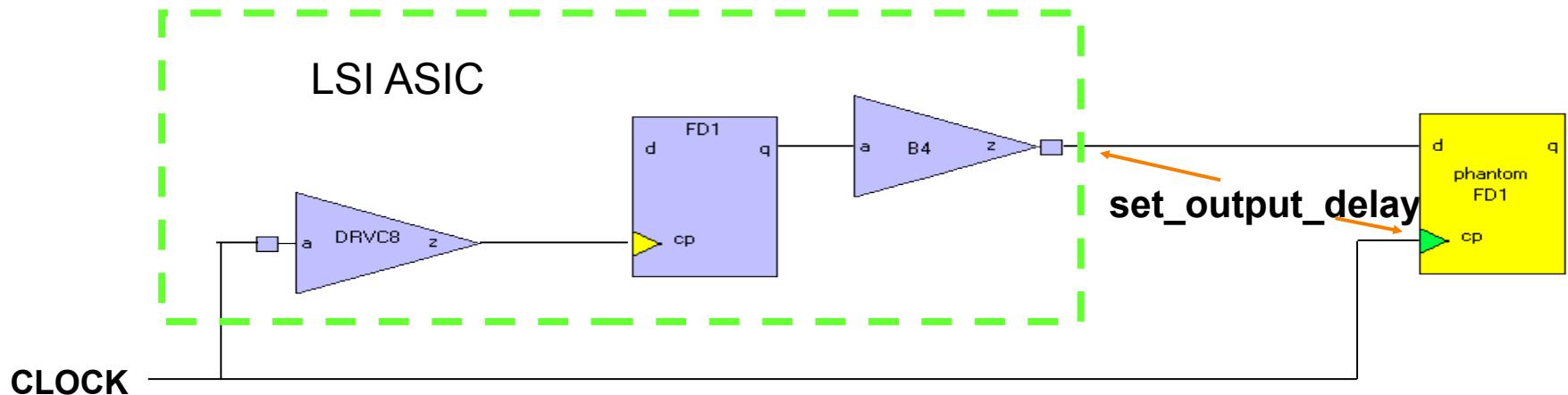


set_output_delay



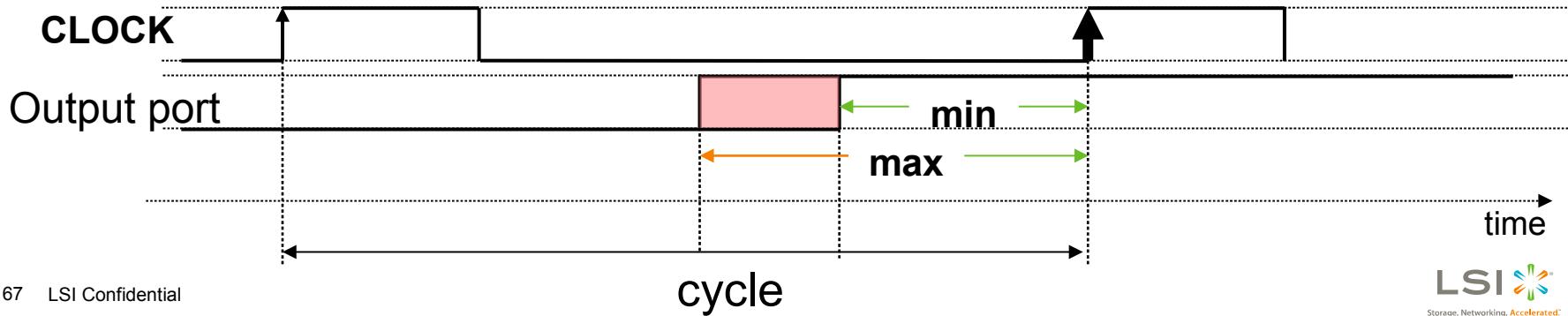
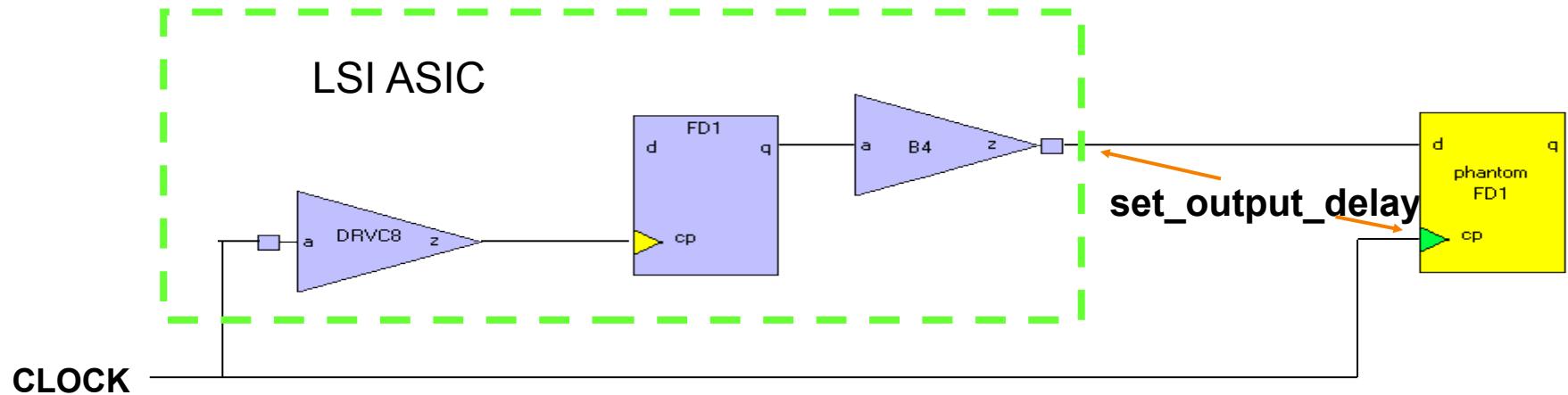
`set_output_delay` is defined as the delay at which the signal will be available at the output port. This delay is relative to the next rising edge of the clock.

set_output_delay



- `set_output_delay -min` is the least delay between the output port and the rising edge of clock. (this is the slowest path from the LSI ASIC “CLOCK” to the Output Port).
- `set_output_delay -max` is the biggest delay between the output port and the rising edge of clock. (this is the fastest path from the LSI ASIC “CLOCK” to the Output Port).

set_output_delay



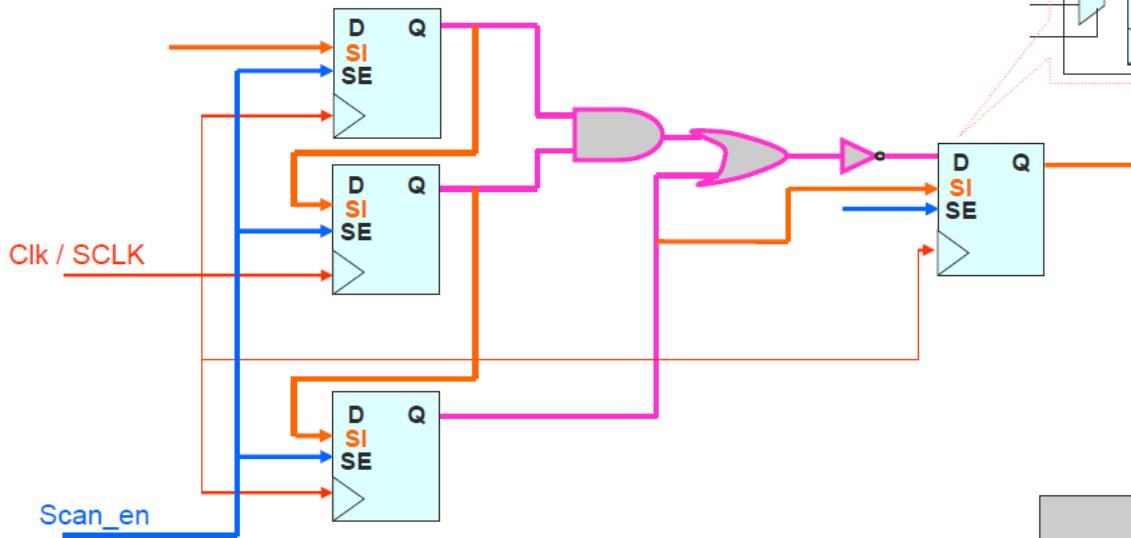
set_case_analysis

- set_case_analysis is used to define a constant state.
- Usually used for controlling mux selects or static signals.
- Only "0" and "1" are allowed.
- set_case_analysis statements will disable check for setup/hold on that signal, but ramp times will be fixed in ICC to the default technology ramp time requirement.

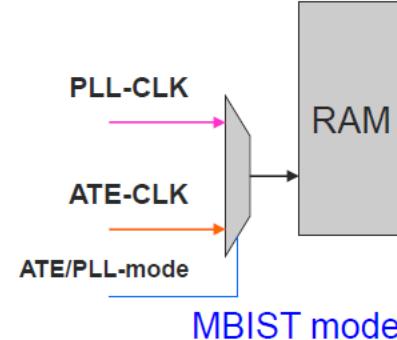
Example:

```
set_case_analysis 0 TEST_MODE  
set_case_analysis 0 adder_top/test_mode
```

Applying Case Analysis



- Modes to be realized
 - Functional mode
 - Scan Shift Mode
 - MBIST Mode

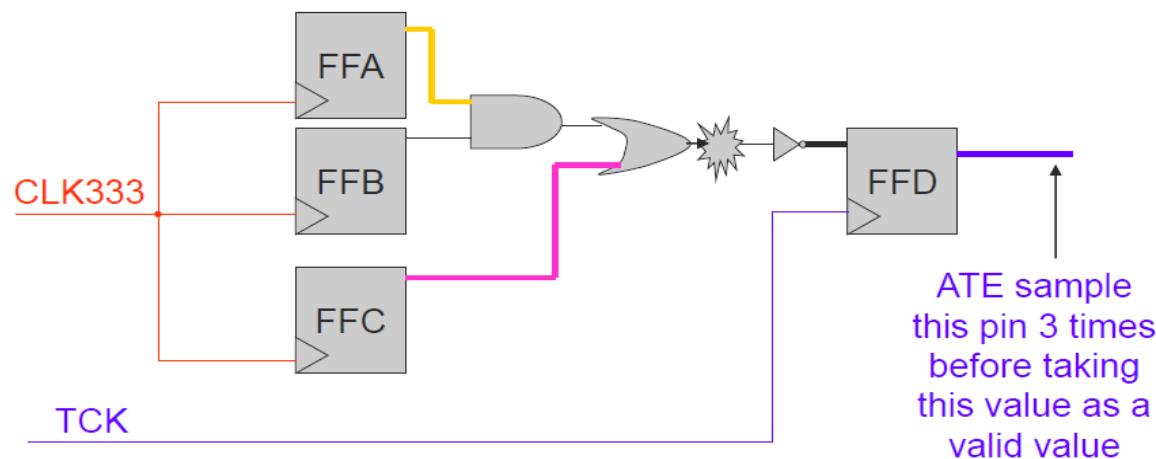


False paths

- A false path is defined as a path that will NOT occur in the real use of the design.
- All false paths will NOT be timed or timing optimized. ICC WILL fix ramp times for all false paths.
- A path may be false in only certain situations. Each situation may need to be analyzed with a different PrimeTime run. (different timing modes for a design)
- Common examples of false paths:
 - false paths for test logic
 - control logic
 - asynchronous cross-clock domain to/from paths
- Can specify –setup or –hold timing as false independently

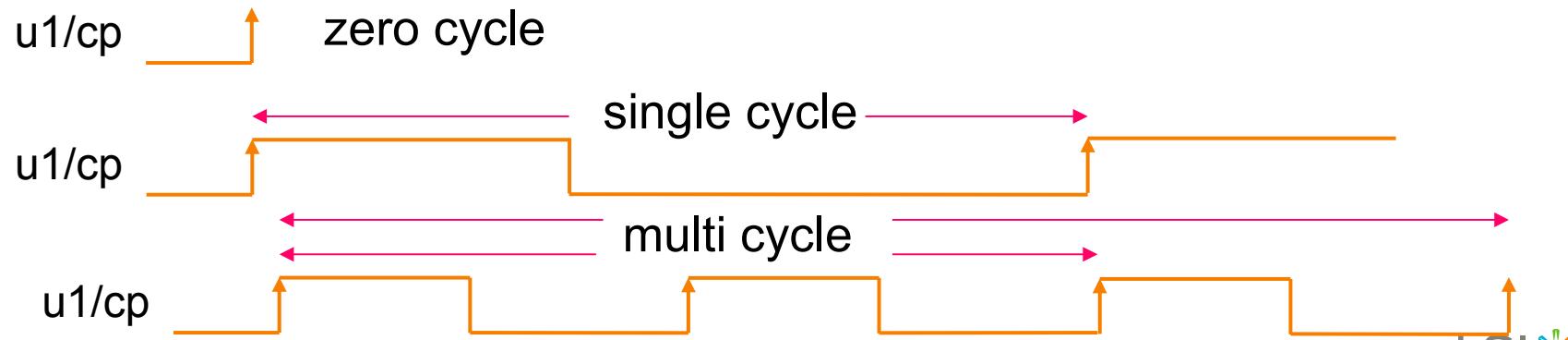
```
set_false_path -from u1/cp -to u3/d  
set_false_path -setup -through uprocmon/Z  
set_false_path -from clk1 -to clk2
```

False Path Example



Multicycle paths

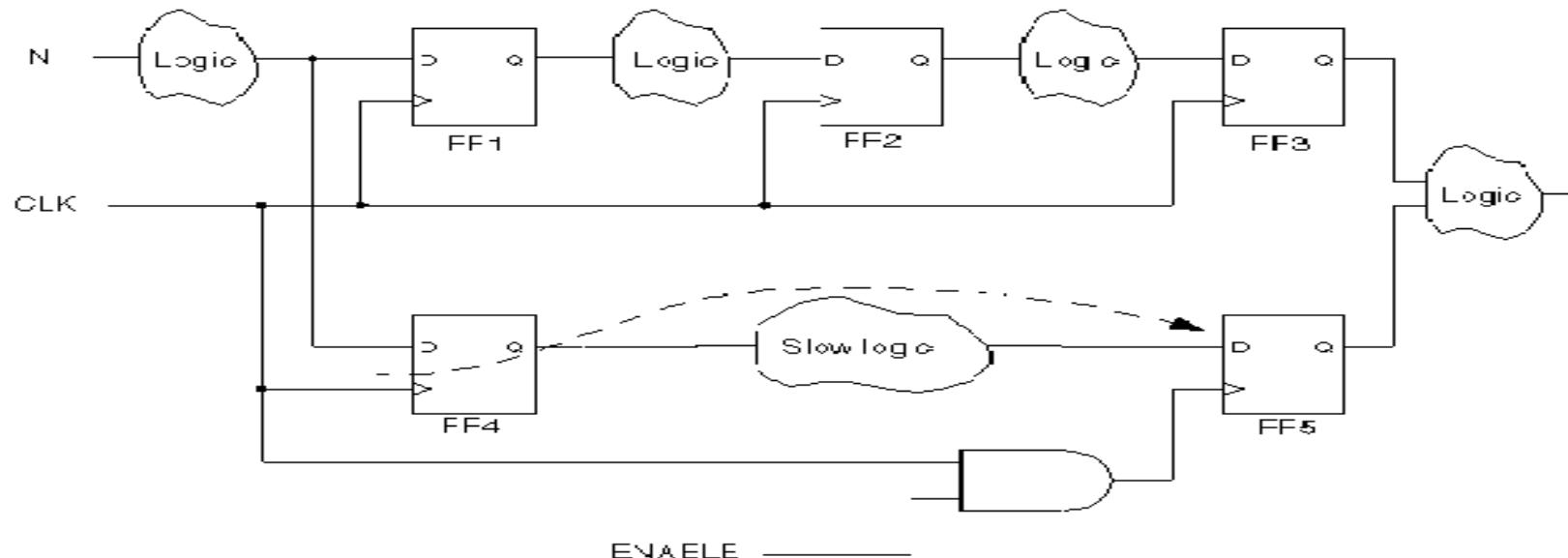
- A single cycle path is one full clock cycle with two rising edges of a clock.
- A zero cycle path is a path that starts and ends with the same edge of a clock.
- Therefore, a multicycle path is a path greater than a single cycle.



Multicycle Paths

Example:

set_multicycle_path -setup 4 -from [get_cells FF4] -to [get_cells FF5]



Disable Timing paths

- `set_disable_timing` will remove timing arcs from a cell to be timed through.
- Usually set on library cells or specific instance cells.
- Examples:

```
set_disable_timing -from A -to Z [get_cells {top/u_buffer}]
```

- One particular instance timing arc is disabled
- NO timing path THROUGH A->Z will exist.

```
set_disable_timing -from D -to QN [get_lib_cells {PLD2T2QND}]
```

- D -> QN arc for EVERY library cell type will be disabled

```
set_disable_timing -from CLKA -to CLKKB [get_cells {u_top/2port_RRinst}]
```

- 2-port memory WRITE-THRU arc disabled for particular instance

More Constraints

- Primetime supports a much larger set of tcl constraints than SDC
 - List of tcl constraints vs. supported sdc:
https://solvnet.synopsys.com/dow_retrieve/Z-2006.12/sdc/sdc_a.html
- May require using source command instead of read_sdc. E.g..
 - get_cells command supported in SDC but –filter expression option is not
 - set_false_path command supported in SDC but –rise_from option is not

Constraint Audits

- Can be done in Spotlight
- Primetime has a lot of commands to audit constraints
 - `check_timing`
 - Checks unconstrained endpoints
 - Checks missing clocks
 - Checks for loops
 - Checks for missing input delays
 - `report_analysis_coverage`
 - Reports percentage of constrained paths
 - `transform_exceptions -dry_run`
 - Provides information on constraints that will not be applied due to:
 - Overlapping constraints
 - Invalid specifier
 - no path



Completeness and Correctness of Constraints is Very Important

Constraint Guidelines : Generic

- TO constraints always END at sync point pins
- FROM constraint always START at sync point pins
- THROUGH any cell pins
- If constrained at non-register instance (cell instance)
 - instance hierarchical name identified as *don't touch* to placement optimization tools so that it does NOT get altered during timing optimization
- If constrained at synthesized cell instance (U* cell instance name)
 - constraints will not match for each netlist received
- DO NOT declare constraints at soft module ports
 - Layout tools do not like this

All Constraints should be @Primary IO, Registers & Hardmac Boundaries

Constraint Guidelines: More specific

- Avoid Over-Constraining
 - is counter-productive to use constraints that cannot be met
- Complex Exceptions
 - Speed up analysis
 - Large numbers of multicycle paths improve the run time of the optimization engines
- DFT Ports
 - scan and DFT related ports need to be constrained
 - constraint preparation needs to be revisited after test-insertion
- Readability of Constraints
 - Easier to read, debug and maintain multiple tcl files.

Rule #1 : Clocks

- RULE : All state elements must be driven by a clock
- EXCEPTIONS:
 - Spare Cells
 - Other flops which are tied off
- CHECKING METHODS:
 - Almost any constraint audit tool
 - Synopsys Tools – *check_timing* report

Rule #2 : Clock Period Relationship

- RULE : Clock periods between communicating domains must have a simple least-common-multiple
- EXCEPTIONS:
 - Asynchronous clock transfers which have false path or Asynch Clock Groups set
- CHECKING METHODS:
 - *check_timing* report
 - information on unexpanded clocks.

Design exploration (& margin adding) is greatly aided if the periods are parameterized.

Rule #3 : Asynchronous Clocks

- RULE : Clocks which are asynchronous must be declared using `set_clock_groups -asynchronous` command - even if they do not communicate directly
- EXCEPTIONS:
 - If `infinite_fanin_windows` are set on all nets
 - Pessimistic Analysis is expected
- CHECKING METHODS:
 - Manual
- Reasoning:
 - Avoid optimistic results for asynchronous Clock Groups

Rule #4 : Generated Clocks

- RULE : Generated clocks must be defined such that they are physically realizable
- EXCEPTIONS:
 - None
- CHECKING METHODS:
 - Report_clock in Primetime

<https://solvnet.synopsys.com/retrieve/020373.html?otSearchResultSrc=advSearch&otS>

Rule #5 : Clocks Transition

- RULE : Clocks should be given a transition value
- EXCEPTIONS:
 - None
- CHECKING METHODS:
 - Report_clock in Primetime
- Reasoning:
 - provides better PT/ICC correlation before clock propagation and more realistic synthesis timing

http://twiki.lsi.com/twiki/pub/DIfocus/Clocks/generate_freq_based_sdc.tcl

Rule #6 : Inputs

- RULE : All inputs must have a set_input_delay with respect to at least one clock in any balanced group of clocks that drive any of the state-elements on its path destinations.
- EXCEPTIONS:
 - None
- CHECKING METHODS:
 - Check_timing Report
- Reasoning:
 - Default is to start an input path at time zero
 - Different timing will be seen in ideal mode vs. propagated clocks
 - Timing path starting at zero will not be correct
 - user will have to add a constraint such as a multicycle or false path if the input should not be timed

Rule #7 : Input Transitions

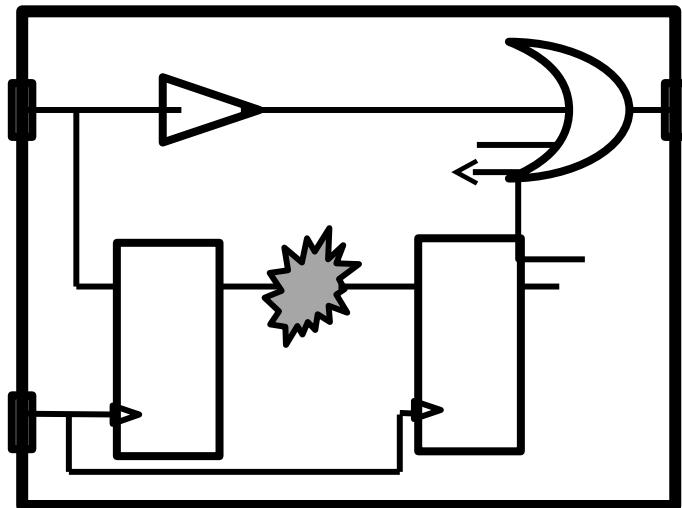
- RULE : All inputs (including clocks) require a `set_driving_cell`
- EXCEPTIONS:
 - None
- CHECKING METHODS:
- Reasoning:
 - provide realistic input slopes and load-dependent delay
 - In Primetime and ICC this provides
 - load-dependent transition for the input signals going into the block
 - load-dependent delay

Rule #8 & 9 : Outputs

- RULE : All outputs must have a `set_output_delay` with respect to at least one clock in any balanced group of clocks that drive any of the state-elements on its path sources
- RULE: All outputs must have `set_load`
- EXCEPTIONS:
 - None
- CHECKING METHODS:
 - Synopsys Tools – `check_timing` report for Rule 8
- Reasoning:
 - Same as Rule 6 & 7

Rule #10 : Direct Input to Output Paths(Feed Through)

- RULE : Direct input-to-output paths need special attention to constrain them within this paradigm.
 - Set_input_delay will take care of inp-reg
 - Set_output_delay will take care of reg-out
 - For the through path additional input and output delay must be added
 - With –add option
 - Ensures 2 external delays for the Input and Output Port
 - Should be wrt to a virtual clock



Rule #11 : Constraints to Avoid whenever possible

- RULE :
 - Set_max_delay
 - constraints specified on hierarchical boundaries
 - Set_false_paths
- EXCEPTIONS:
 - Depends on the Design
 - Likely you will see this in every design

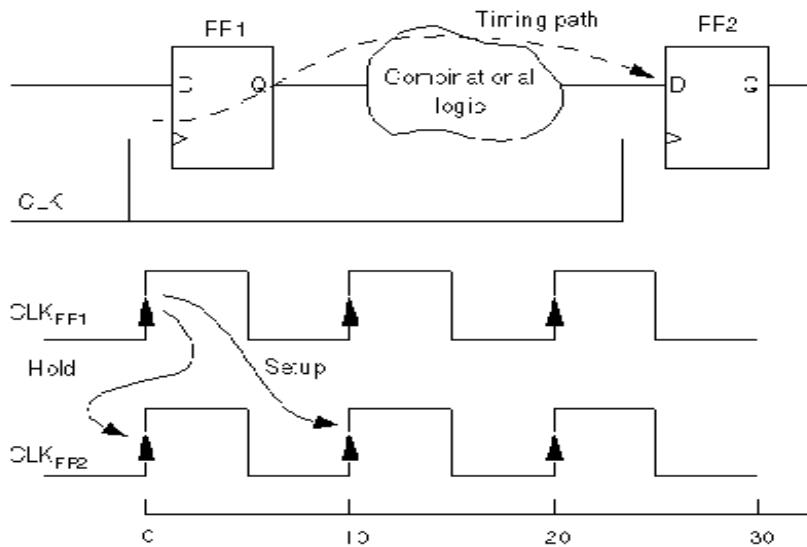
Rule #12 : MultiCycle Paths

- RULE : *set_multicycle_path -setup* must always be matched with a corresponding *set_multicycle_path -hold*
 - Always we consider hold to be zero cycle.
 - If hold constraint not mentioned, hold checks will not be zero cycle

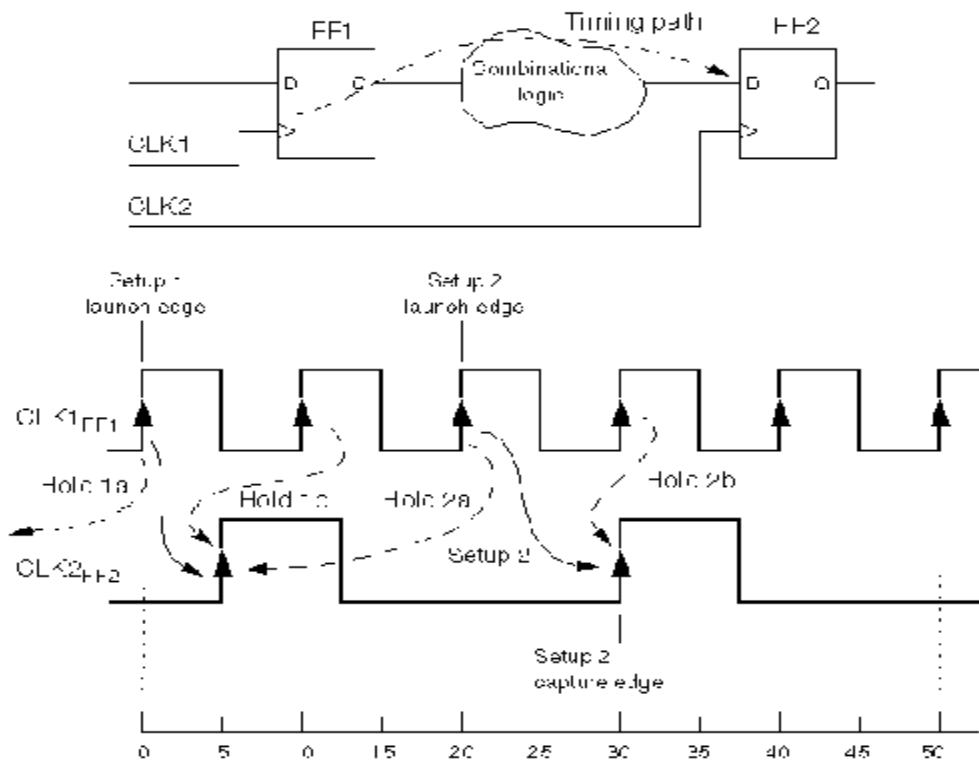
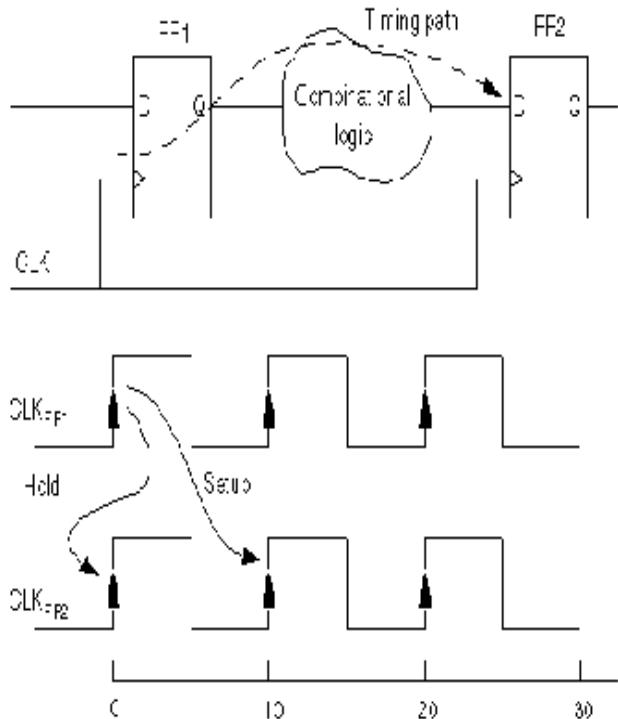
Rule #13 : IO constraints

- RULE : IO should be constrained such that timing does not change after clock is switched from ideal mode to propagated
- METHODS:
 - Virtual Clock
 - `set_propagated_clock [all_fanout -flat -clock_tree]`

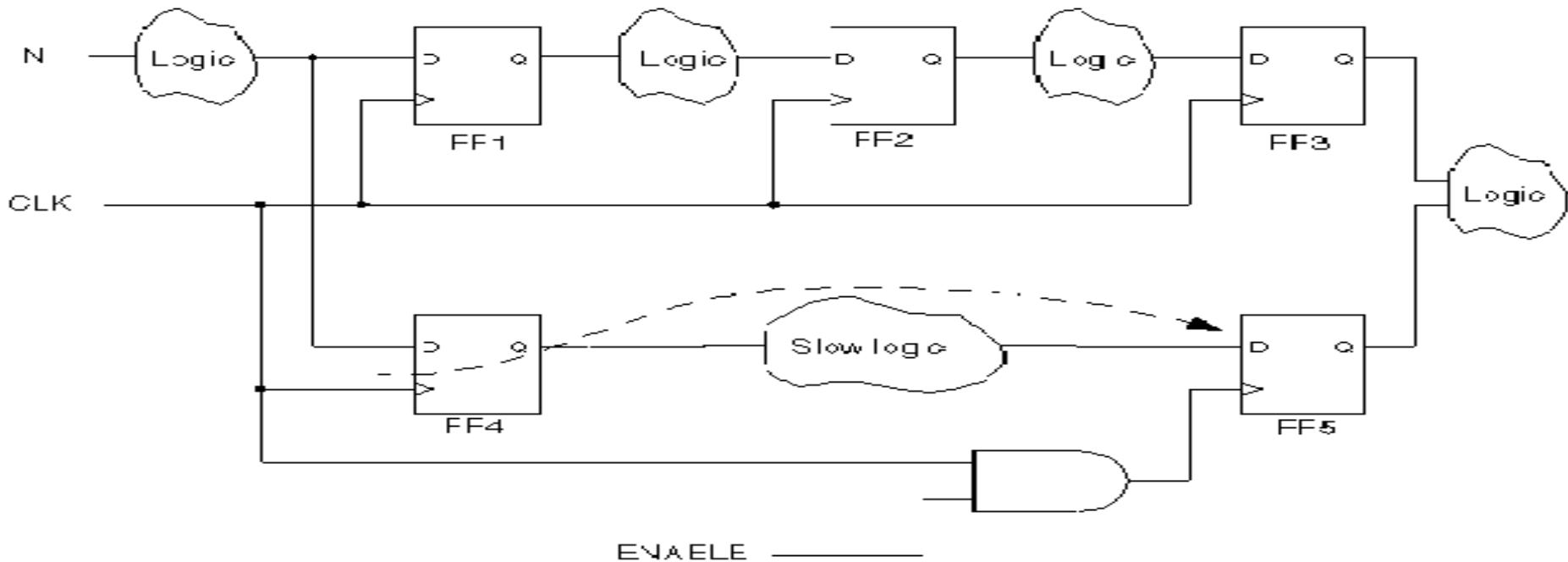
Advanced : Review Setup and Hold



Advanced : Setup and Hold for 2 clocks

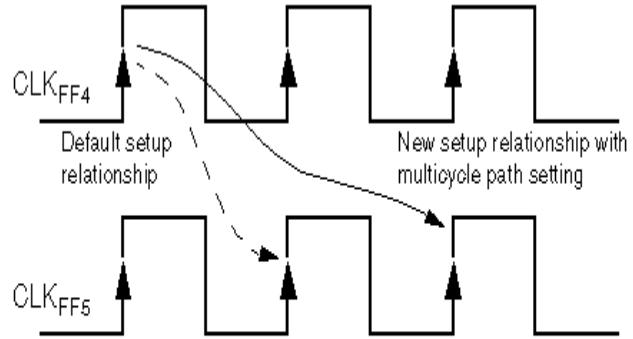


Advanced: Setup & Hold for MultiCycle

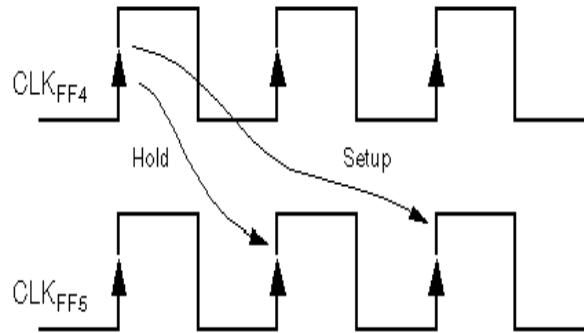


- **set_multicycle_path -setup 2 \ -from [get_cells FF4] -to [get_cells FF5]**

Advanced: Setup & Hold for MultiCycle



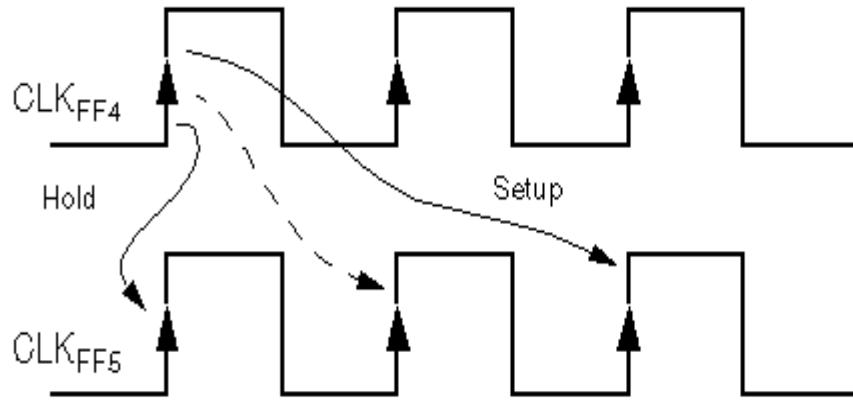
```
pt_shell> set_multicycle_path -setup 2 \
           -from [get_cells FF4] -to [get_cells FF5]
```



```
pt_shell> set_multicycle_path -setup 2 \
           -from [get_cells FF4] -to [get_cells FF5]
```

Is the Hold Check Right? Why or Why not?

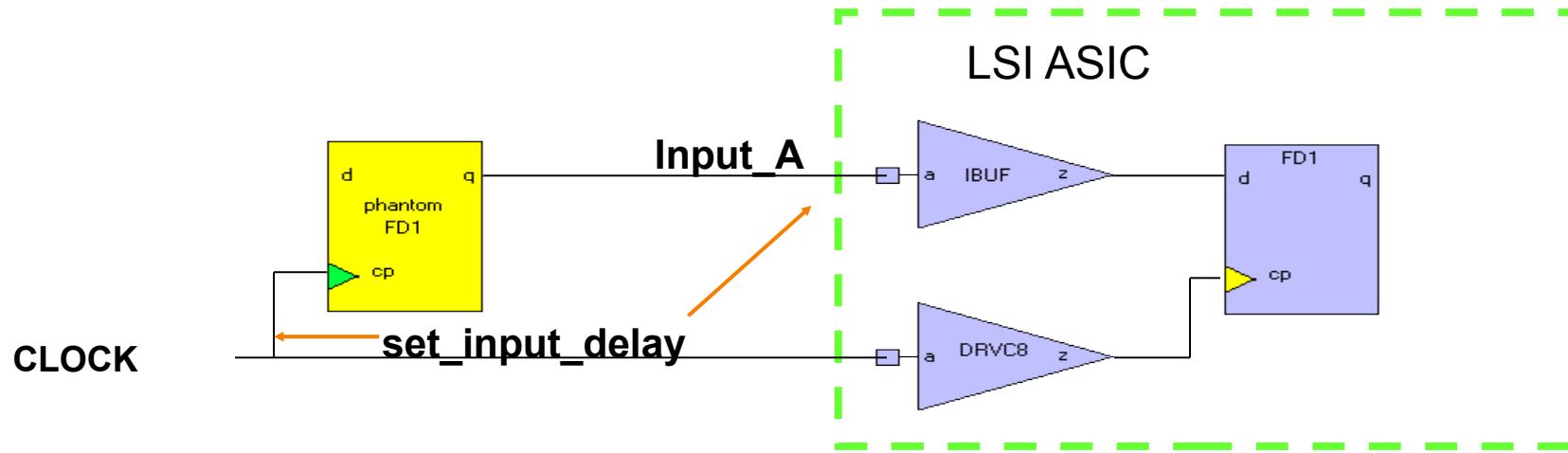
Advanced: Setup & Hold for MultiCycle



```
pt_shell> set_multicycle_path -setup 2 \
           -from [get_cells FF4] -to [get_cells FF5]
```

```
pt_shell> set_multicycle_path -hold 1 \
           -from [get_cells FF4] -to [get_cells FF5]
```

Review: QUIZ1 set_input_delay



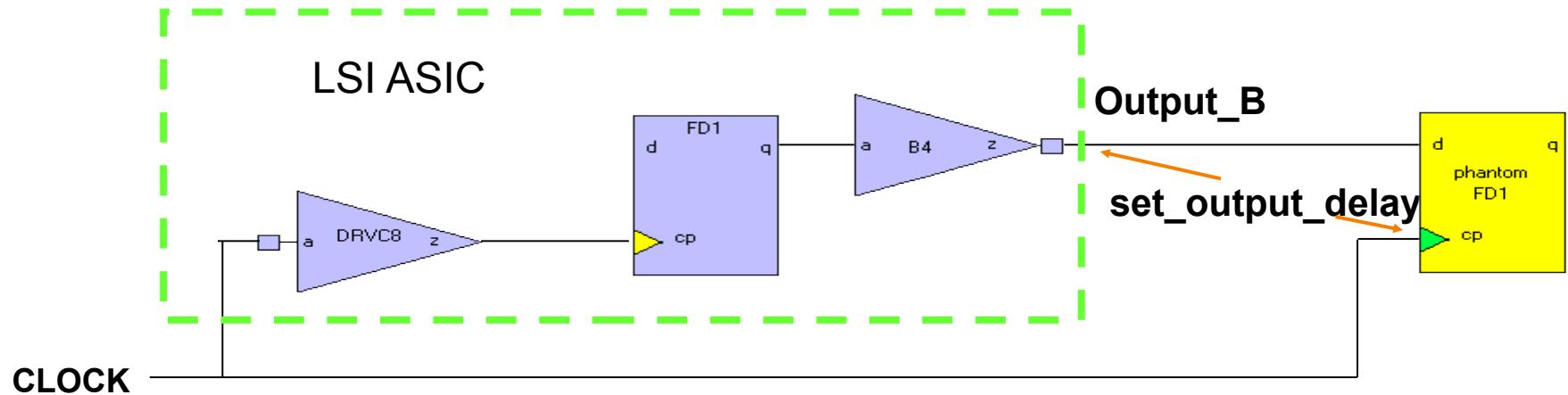
Example:

Cycle = 8ns

```
set_input_delay -clock CLOCK -max 2.0 [Input_A]  
set_input_delay -clock CLOCK -min 1.0 [Input_A]
```

What does this mean?

REVIEW:QUIZ2



Example:

Cycle = 8ns

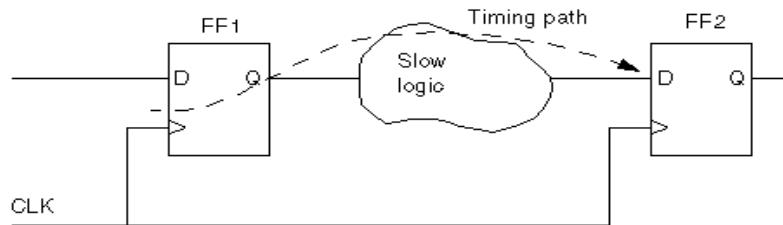
```
set_output_delay -clock CLOCK -max 6.0 [Output_B]
```

```
set_output_delay -clock CLOCK -min -1.0 [Output_B]
```

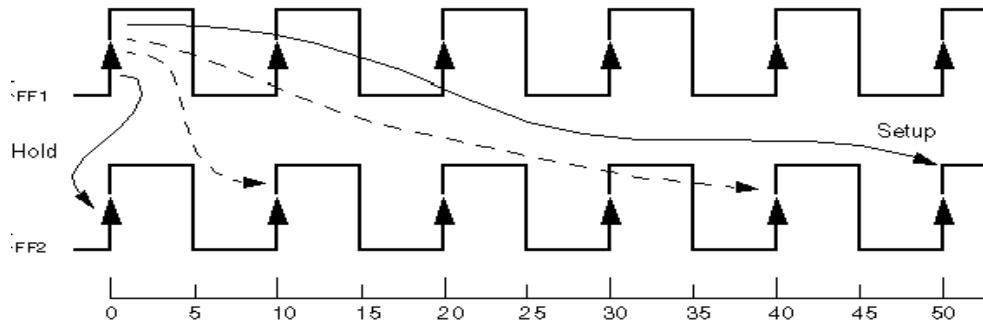
What does this mean?

Review: QUIZ3

- What should the constraints be for this waveform

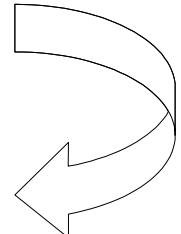


```
pt_shell> create_clock -period 10 -waveform {0 5} CLK1
```





Storage. Networking. **Accelerated.**TM



Primetime

Accelerate.



Objectives

- When do we run PTSI?
- Tool access and Setup
- Inputs to Primetime
- Understanding Outputs of Primetime

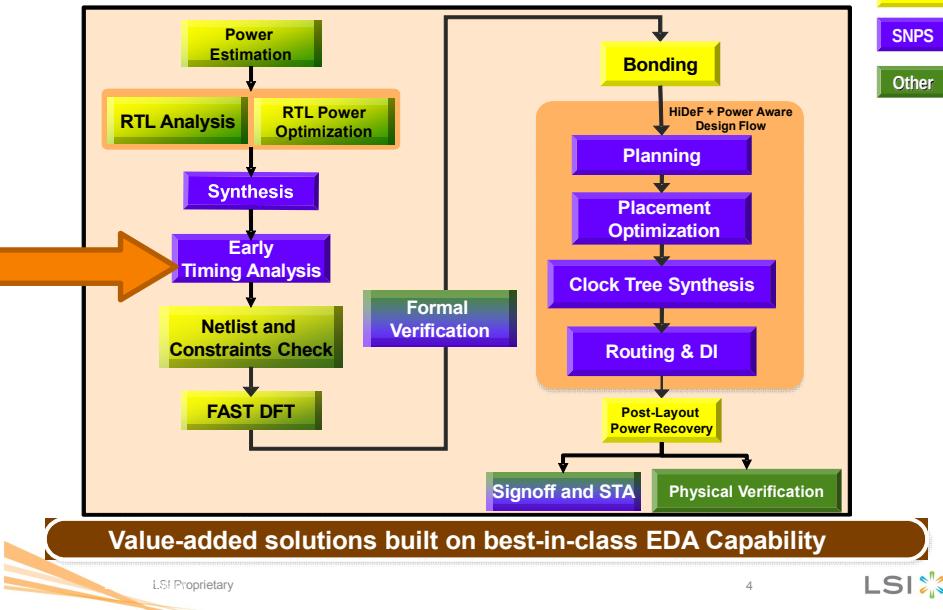
Get some knowledge to run and analyze Static Timing using Primetime

Running PTSI – Validation of Customer Delivery

- Pre Test : Customer Netlist & Constraints delivery

- Focus :
 - ZWLM is clean
 - Check Timing is clean
 - Analysis of clocks, resets wrt DIS

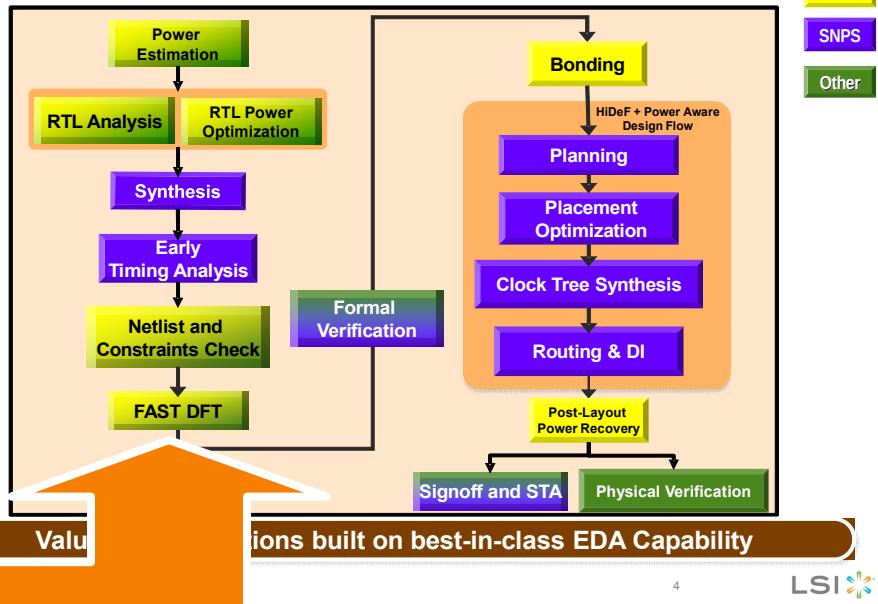
LSI 40nm and 28nm Design Flow



Running PTSI : Post DFT

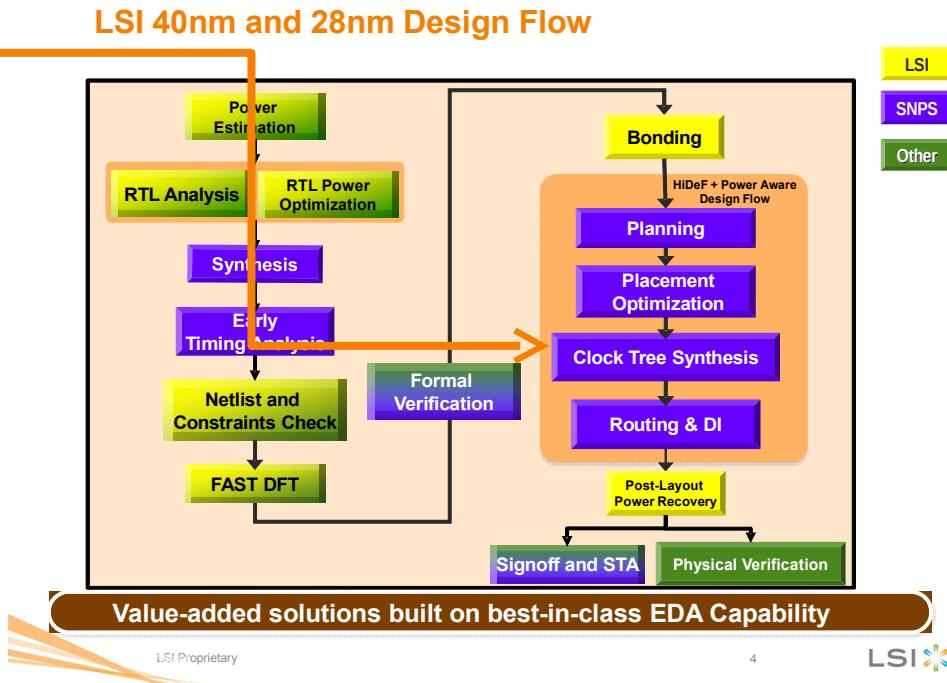
- Post Test : DFT inserted Netlist & Constraints Delivery
 - Focus:
 - ZWLM is still clean
 - Check Timing is clean
 - Analyze the new constraints to make sure that the timing is still as clean as when customer delivered it
 - Merged Mode constraints will have more end points to verify
 - MultiMode constraints need to verify all modes separately.

LSI 40nm and 28nm Design Flow



Running PTSI – For CTS correlation

- Clock Tree insertion in ICC [post DFT netlist]
 - Merged constraints with all clocks
 - Make sure you understand all MUXes inserted during test
 - Make sure all case analysis is correct
 - Just define only clocks and generated clocks in the constraints file.
 - Using Multimode even during CTS
 - Not aware of it being done at present in LSI



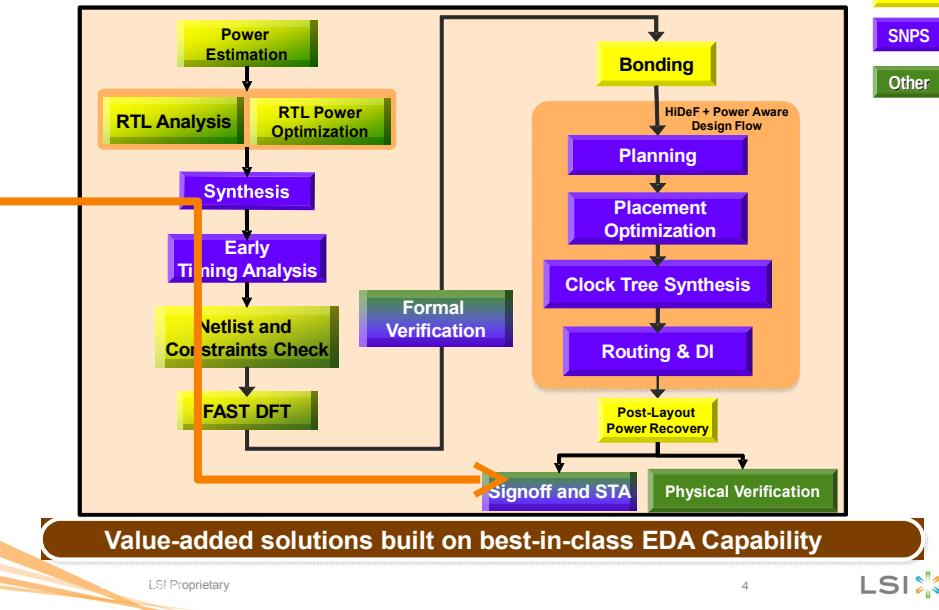
Running PTSI for Signoff

- Post Layout & Signoff

- Focus:

- Check Timing still has the same number of endpoints – Make sure ICC did not break anything
 - Violators Reports are all clean

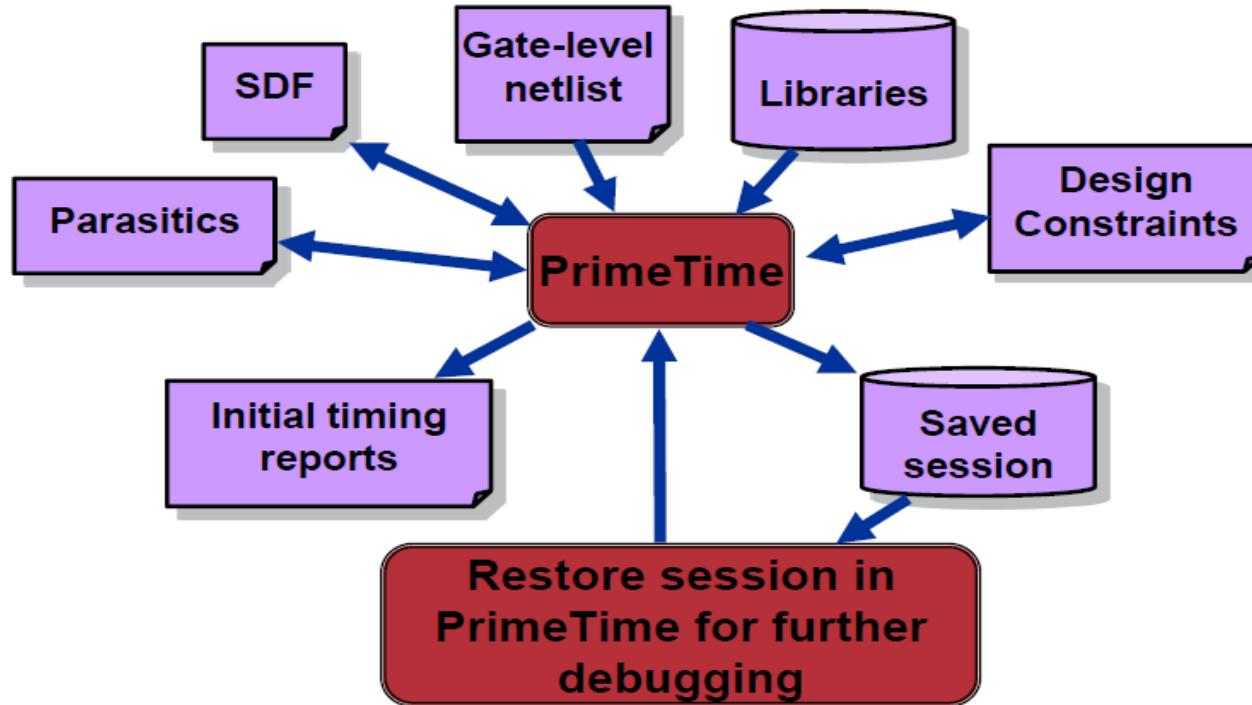
LSI 40nm and 28nm Design Flow



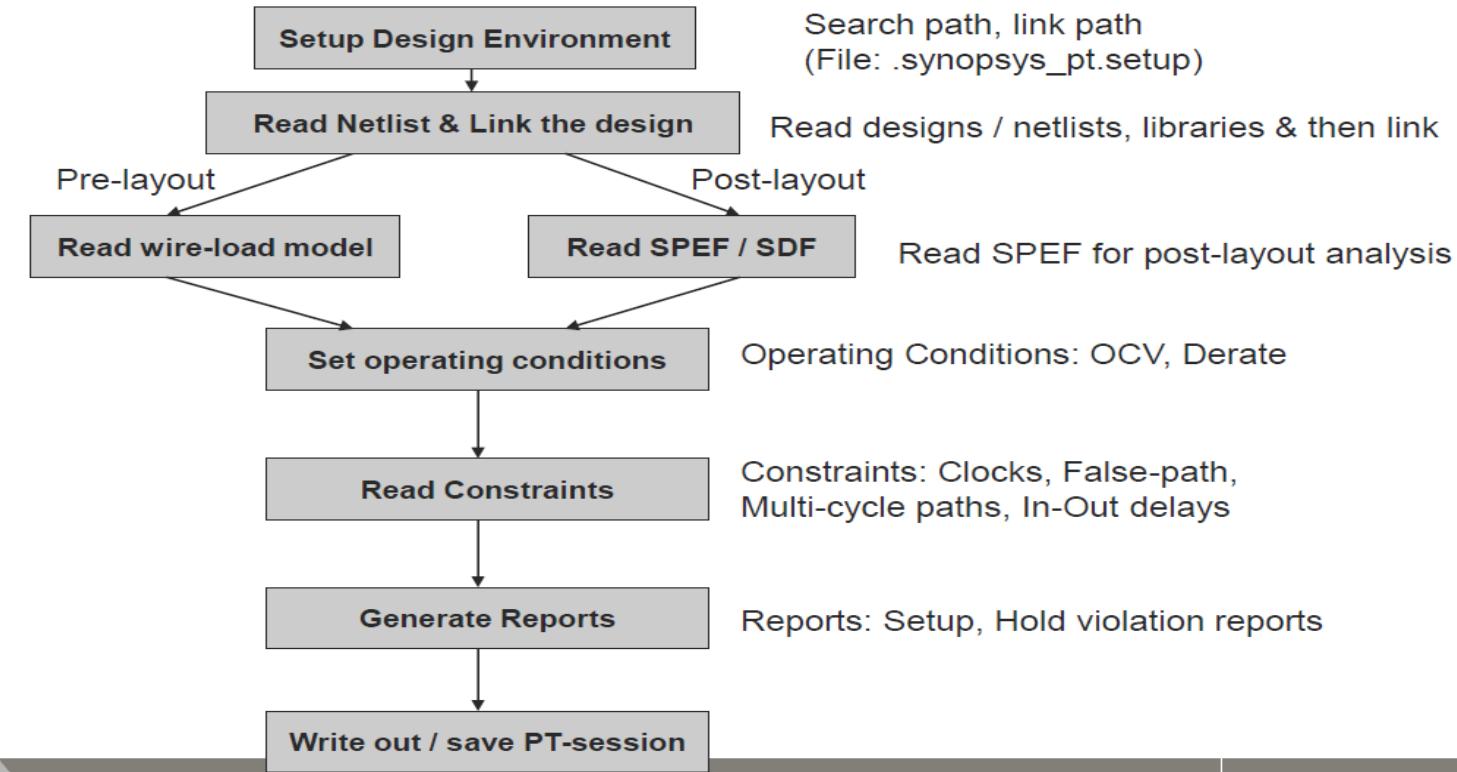
Access to PrimetimeSI and Startup

- ICDS sets the version of PTSI you are using
 - addpg@primetime@2011.06
- Invoke Primetime
 - Batch Mode : pt_shell /pt_shell -64bit
 - GUI Mode: primetime / primetime -64bit
 - GUI can be invoked from pt_shell using start_gui

Primetime Inputs and Outputs



Basic Primetime STA Flow



Primetime Inputs : Setup

- What does Setup mean
 - Where to search for Libraries
 - What Libraries to use to Analyze
 - Default settings for the Tool
- Primetime automatically reads in a *.synopsys_pt.setup* file
 - From synopsys release area
 - From \$HOME directory – store typical variables
 - From \$CWD
- Example:
 - set *search_path* [list . \
 \$LIB_PATH/sclxd/synopsys/db \
 \$LIB_PATH/sclid/synopsys/db]
 - set *link_library* [list * \
 tsmc_cln40g_sclxd_\${process}_\${temp}_\${v_core}.db \
 tsmc_cln40g_sclmd_\${process}_\${temp}_\${v_core}.db]

Primetime Inputs: Setup

- Can use a common setup file available
 - 3 sections
 - Sources of data
 - Links to libraries
 - Tool specific setup
 - Can be Found in:

<http://ssg.teams.lsi.com/sites/cs/dept>

The screenshot shows the LSI Core Platform Flow homepage. At the top, there's a navigation bar with links like 'SSG Home', 'Custom Solutions Division', 'Departments', 'Welcome Gayatri_Mahadevan Srinivasan', 'My Site', 'My Links', and a search bar. Below the header is the LSI logo and the text 'Core Platform Flow'. A left sidebar contains a 'Documents' section with links to 'ICC Sample Scripts', 'Script Documentation', 'Miscellaneous ICC', 'Synthesis', 'Formal verification', 'ICC Log Files for Debug', 'Vendor Documentation', and 'Encounter Sample Scripts & Documentation'. It also has 'Sites' and 'People and Groups' sections. The main content area displays the message 'This is the homepage for the Core Platform Flow and ICC Sample scripts'. Below this, under 'Current links are:', there are several blue hyperlinks: 'Vendor Documentation', 'Synthesis & Formal Verification', 'ICC Sample Scripts', 'ICC Script Documentation', and 'Miscellaneous ICC'.

Primetime Inputs : Setup (libraries)

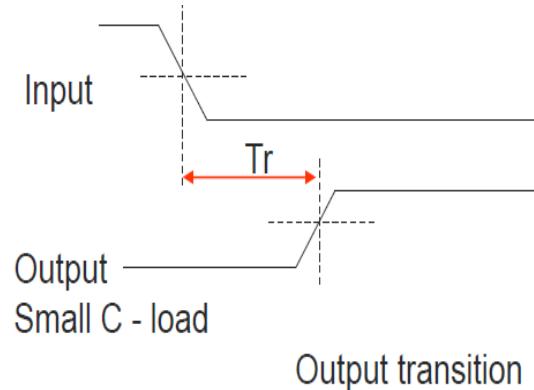
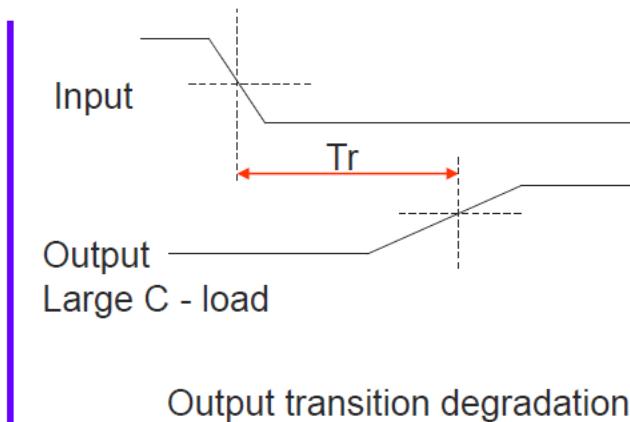
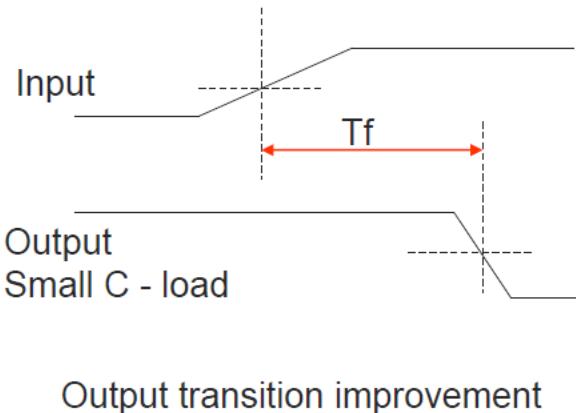
- Technology Libraries
 - LDM (Linear delay Model)
 - NLDM (Non Linear Delay Model)
 - CCS (Composite current source)



Most Accurate

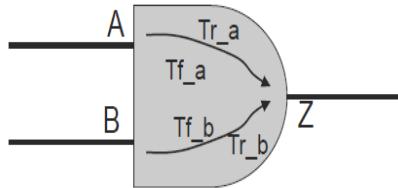
Linear Delay Model

- Delay of a cell depends on
 - Input Transition
 - Output Load



Non Linear Delay Model

- Table models cell delay and output transition times
 - Capture various combination of
 - Input Transition
 - Output Load
 - Non-Linear variation of delay wrt to input transition and output load



Timing Arcs:

- $A \rightarrow Z$: output rise
- $A \rightarrow Z$: output fall
- $B \rightarrow Z$: output rise
- $B \rightarrow Z$: output fall

Library has 4 tables
-2 for cell delays
-2 for transition time

Composite Current Source Libraries

- Effective interconnect resistance starts showing effects for smaller technologies and NLDM becomes inaccurate
- Output Driver is modeled by an equivalent current source
- Output Current Information depends on Input Transitions and Output Load

Library and Corners

- STA needs to be performed for different conditions of
 - Process
 - Voltage
 - Temperature
- Delay values vary wrt to PVT corners
- Each Technology has various Library Corners
 - E.g. : 40nm
 - SLOW_125c_0p81v
 - FAST_M40c_0p99v
 - SLOW_M40c_0p81v
 - FAST_125c_0p99v

PrimeTime Inputs : Netlist

- Netlist is typically read in with the “read_verilog” command
 - Netlist must be structural
- Link command resolves all references in the design
- Example commands to read in netlist and link design

```
source setup.slow  
read_verilog block1.v  
read_verilog block2.v  
read_verilog top.v  
current_design top  
link
```

Read_verilog <file.v> ; current_design; link

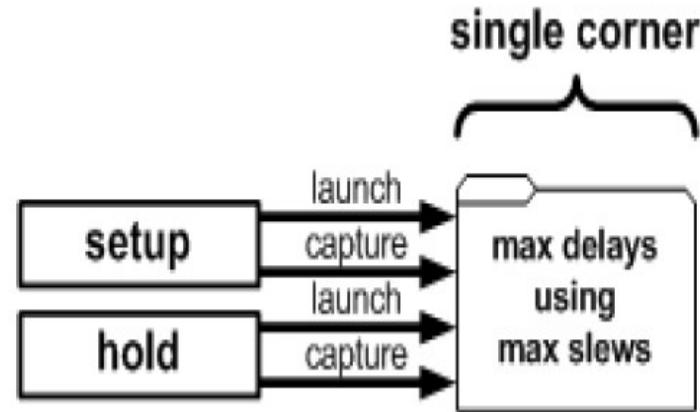
Primetime Inputs : Operating Conditions

- Semiconductor device parameters can vary with conditions
 - fabrication process
 - operating temperature
 - power supply voltage
- 3 types (analysis)
 - Single Operating condition
 - On chip Variation Mode
 - Advanced on Chip variation Mode

```
set_operating_condition -analysis_type <...>
```

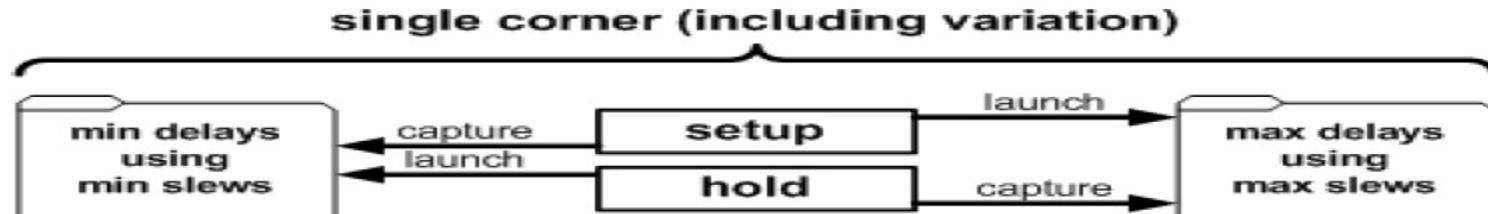
Primetime Inputs : Operating Conditions - Single

- Single Operating condition Mode
 - Uses a single set of delay parameters for the entire design
 - based on one
 - set of process
 - Temperature
 - voltage conditions
 - All delays represent worst case timing at a given corner



Primetime Inputs : Operating Conditions - On Chip Variation

- Performs a conservative analysis that allows both minimum and maximum delays to apply to different paths at the same time
- For a setup check
 - maximum delays for the launch clock path and data path,
 - minimum delays for the capture clock path
- For a hold check
 - minimum delays for the launch clock path and data path
 - maximum delays for the capture clock path.
- Can Perform a constant derate to the delays



On Chip Variation

- Why OCV?
 - Delays vary across die due to process variations
 - Variation needs to be modeled by scaling coefficients of library
 - Variation is applied as a Derate Factor
- Works on a single operating corner
 - Timing analysis uses variation in timing arc values within a corner
- Sources for OCV
 - Timing Derate Values(early/late)
 - Input Drive Strength/Input Transition (min/max)
 - Output Load (min/max)
 - Multiple Input Gates and their corresponding slew Values at Slew Merge Point
 - Net Delays (min/max)

Primetime Inputs : Operating Conditions - Advanced On Chip Variation

- Determines derating factors
 - on metrics of path logic depth
 - the physical distance traversed by a particular path
- For more information refer to additional training or Solvenet

Concept of Derating

- Is done to add more margin to the paths
- Adjusts minimum delays and maximum delays by specified factors
 - models the effects of operating conditions
- Uses `set_timing_derate` command
- Calculated using this formula
 - $delay_new = old_delay + ((derating_factor - 1.0) * abs(old_delay))$
 - Early option value < 1.0
 - Will reduce value of `delay_new`
 - Late Option Value >1.0
 - Will increase the value of `delay_new`

Derate - syntax

Usage:

```
set_timing_derate      # Capture delay derating factor
[-early]                (Specify early derate factor)
[-late]                 (Specify late derate factor)
[-rise]                 (Specify rise derate factor)
[-fall]                 (Specify fall derate factor)
[-data]                  (Specify derate factor for data paths only)
[-clock]                 (Specify derate factor is for clock paths only)
[-cell_delay]            (Specify derate factor for cell delays only)
[-cell_check]            (Specify derate factor for cell timing checks only)
[-net_delay]              (Specify derate factor for net delays only)
[-static]                 (Specify derate factor for non-delta delays only)
[-dynamic]                (Specify derate factor for delta delays only)
[-scalar]                 (Specify derate factor for deterministic delays only)
[-variation]              (Specify derate factor for statistical delays only)
derate_value             (Derate factor)
[object_list]             (List of cells, library cells or nets)
```

- More specific derates override general ones

OCV example report

Point	Derate	Incr	Path
clock fir_clk (rise edge)	0.000	0.000	
clock source latency	0.000	0.000	
fir_clk (in)	0.017 &	0.017 r	
CLKINV200XP_G1B1I1/A (CLKINV200XP)	1.000	0.004 &	0.021 r
CLKINV200XP_G1B1I1/Z (CLKINV200XP)	1.000	0.070 &	0.091 f
CLKBUF150XP_Id/A (CLKBUF150XP)	1.000	0.005 &	0.095 f
CLKBUF150XP_Id/Z (CLKBUF150XP)	1.000	0.054 &	0.150 f
CLKINV100XP_G1B2I27/A (CLKINV100XP)	1.000	0.005 &	0.155 f
CLKINV100XP_G1B2I27/Z (CLKINV100XP)	1.000	0.083 &	0.238 r
...			
fb/U142/Z (AOI211M05XP)	1.000	0.193 &	5.307 r
fb/sum_out_reg_54_/D (FD1SQM1XP)	1.000	0.068 &	5.375 r
data arrival time			5.375
clock fir_clk (rise edge)	6.000	6.000	
clock source latency	0.000	6.000	
fir_clk (in)	0.016 &	6.016 r	
CLKINV150XP_G1B1I2/A (CLKINV100XP)	0.920	0.006 &	6.021 r
CLKINV150XP_G1B1I2/Z (CLKINV100XP)	0.920	0.109 &	6.130 f
CLKINV100XP_G1B2I20/A (CLKINV100XP)	0.920	0.001 &	6.132 f
CLKINV100XP_G1B2I20/Z (CLKINV100XP)	0.920	0.096 &	6.227 r
fb/sum_out_reg_54_/CP (FD1SQM1XP)	0.920	0.001 &	6.228 r
clock reconvergence pessimism	0.001		6.230
clock uncertainty	-0.200		6.030
library setup time	1.000	-0.101	5.929
data required time			5.929

data required time			5.929
data arrival time			-5.375

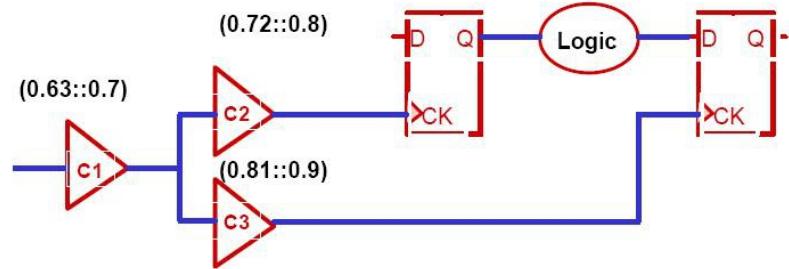
slack (MET)			0.554

Review

Analysis mode	Timing check	Launch clock path	Data path	Capture clock path
Single operating condition	Setup	Late clock, maximum delay in clock path, single operating condition (no derating).	Maximum delay, single operating condition (no derating)	Early clock, minimum delay in clock path, single operating condition (no derating).
	Hold	Early clock, minimum delay in clock path, single operating condition (no derating).	Minimum delay, single operating condition (no derating)	Late clock, maximum delay in clock path, single operating condition (no derating).
OCV mode	Setup	Late clock, maximum delay in clock path, late derating, worst-case operating condition.	Maximum delay, late derating, worst-case operating condition	Early clock, minimum delay in clock path, early derating, best-case operating condition.
	Hold	Early clock, minimum delay in clock path, early derating, best-case operating condition.	Minimum delay, early derating, best-case operating condition	Late clock, maximum delay in clock path, late derating, worst-case operating condition.

Review

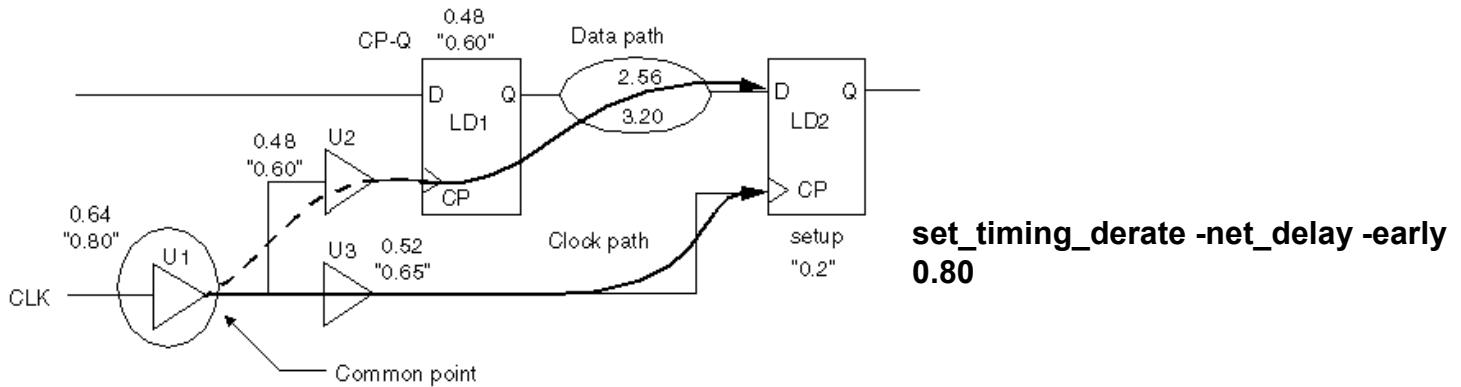
- What delays will be taken for
 - Single Mode
 - OCV Mode
 - With a derate of 4%



CRPR (Clock Reconvergence Pessimism Removal)

- Clock reconvergence pessimism is an accuracy limitation that occurs
 - when two different clock paths partially share a common physical path segment
 - the shared segment is assumed to have a minimum delay for one path and a maximum delay for the other path
 - This condition can occur any time that launch and capture clock paths use different delays most commonly with OCV analysis
 - Automated correction of this inaccuracy is called clock reconvergence pessimism removal (CRPR).

CRPR



- Clock Reconvergence Pessimism (CRP) occurs when the launch and capture clock paths use different delays through common gates/nets (U1 common gate)
- A gate can not have different process delays based on the timing path examined
- CRPR adds back the delta between early and late delays for the common path

CRPR Example

Timing report with & without CRPR

Startpoint: ecc_init_nrz_sync_flpsig_rrr_reg
 (rising edge-triggered flip-flop clocked by nrz_clk)
 Endpoint: I_ECC_INIT_NRZ_r_reg
 (rising edge-triggered flip-flop clocked by nrz_clk)
 Path Group: nrz_clk
 Path Type: max
 Max Data Paths Derating Factor : 1.040 **CRPR = false**
 Min Clock Paths Derating Factor : 0.960 **Pessimism is present**
 Max Clock Paths Derating Factor : 1.040

Point	Incr	Path
clock nrz_clk (rise edge)	0.000	0.000
clock network delay (propagated)	0.416	0.416
ecc_init_nrz_sync_flpsig_rrr_reg/CP (SDFPRQLK1HxD)	0.000	0.416 r
ecc_init_nrz_sync_flpsig_rrr_reg/Q (SDFPRLX1HxD)	0.310 &	0.726 r
U9303/Z (XNOR2X1LKD)	0.068 &	0.794 r
U9304/Z (DAI21X1LKD)	0.036 &	0.830 f
I_ECC_INIT_NRZ_r_reg/D (SDFPRQLX1HxD)	0.002 &	0.833 f
data arrival time		0.833
clock nrz_clk (rise edge)	2.900	2.900
clock network delay (propagated)	0.380	3.280
clock uncertainty	-0.200	3.080
I_ECC_INIT_NRZ_r_reg/CP (SDFPRQLX1HxD)		3.080 r
library setup time	-0.120	2.959
data required time		2.959
data required time		2.959
data arrival time		-0.833
slack (MET)		2.126

Startpoint: ecc_init_nrz_sync_flpsig_rrr_reg
 (rising edge-triggered flip-flop clocked by nrz_clk)
 Endpoint: I_ECC_INIT_NRZ_r_reg
 (rising edge-triggered flip-flop clocked by nrz_clk)
 Path Group: nrz_clk
 Path Type: max
 Max Data Paths Derating Factor : 1.040 **CRPR = true**
 Min Clock Paths Derating Factor : 0.960 **Pessimism is removed**
 Max Clock Paths Derating Factor : 1.040

Point	Incr	Path
clock nrz_clk (rise edge)	0.000	0.000
clock network delay (propagated)	0.416	0.416
ecc_init_nrz_sync_flpsig_rrr_reg/CP (SDFPRQLX1HxD)	0.000	0.416 r
ecc_init_nrz_sync_flpsig_rrr_reg/Q (SIFPRQLX1HxD)	0.310 &	0.726 r
U9303/Z (XNOR2X1LKD)	0.068 &	0.794 r
U9304/Z (DAI21X1LKD)	0.036 &	0.830 f
I_ECC_INIT_NRZ_r_reg/D (SDFPRQLX1HxD)	0.002 &	0.833 f
data arrival time		0.833
clock nrz_clk (rise edge)	2.900	2.900
clock network delay (propagated)	0.380	3.280
clock reconvergence pessimism	0.017	3.297
clock uncertainty	-0.200	3.097
I_ECC_INIT_NRZ_r_reg/CP (SDFPRQLX1HxD)		3.097 r
library setup time	-0.120	2.976
data required time		2.976
data required time		2.976
data arrival time		-0.833
slack (MET)		2.143

CRPR Example : Thresold is 20ps.

```

Startpoint: ecc_init_nrz_sync_flpsig_rrr_reg
            (rising edge-triggered flip-flop clocked by nrz_clk)
Endpoint: I_ECC_INIT_NRZ_r_reg
            (rising edge-triggered flip-flop clocked by nrz_clk)
Path Group: nrz_clk
Path Type: max
Max Data Paths Derating Factor : 1.040
Min Clock Paths Derating Factor : 0.960
Max Clock Paths Derating Factor : 1.040

```

Point	Incr	Path		CRPR Threshold = 20ps Run time & Memory usage
clock nrz_clk (rise edge)	0.000	0.000	clock nrz_clk (rise edge)	2.900 2.900
clock source latency	0.000	0.000	clock source latency	0.000 2.900
nrz_clk (in)	0.000 &	0.000 r	nrz_clk (in)	0.000 & 2.900 r
PORT_CLK_BUF_nrz_clk/Z (CLKBUFX8LXD)	0.039 &	0.039 r	PORT_CLK_BUF_nrz_clk/Z (CLKBUFX8LXD)	0.036 & 2.936 r
CLKINVX6LXD_G1B1I1/Z (CLKINVX8LXD)	0.041 &	0.080 f	CLKINVX6LXD_G1B1I1/Z (CLKINVX8LXD)	0.038 & 2.974 f
CLKINVX3LXD_G1B2I2/Z (CLKINVX3LXD)	0.064 &	0.144 r	CLKINVX3LXD_G1B2I2/Z (CLKINVX3LXD)	0.059 & 3.033 r
CLKINVX6LXD_G1B3I2_1/Z (CLKINVX5LXD)	0.079 &	0.223 f	CLKINVX6LXD_G1B3I2_1/Z (CLKINVX5LXD)	0.069 & 3.102 f
CLKINVX6LXD_G1B4I4/Z (CLKINVX6LXD)	0.057 &	0.280 r	CLKINVX6LXD_G1B4I4/Z (CLKINVX6LXD)	0.052 & 3.154 r
CLKINVX4LXD_G1B5I4/Z (CLKINVX3LXD)	0.068 &	0.348 f	CLKINVX4LXD_G1B5I4/Z (CLKINVX3LXD)	0.062 & 3.217 f
CLKINVX3LXD_G1B6I40/Z (CLKINVX2LXD)	0.068 &	0.415 r	CLKINVX3LXD_G1B6I40/Z (CLKINVX2LXD)	0.063 & 3.279 r
ecc_init_nrz_sync_flpsig_rrr_reg/CP (SDFPRQLX1HxD)	0.001 &	0.416 r	I_ECC_INIT_NRZ_r_reg/CP (SDFPRQLX1HxD)	0.000 & 3.280 r
ecc_init_nrz_sync_flpsig_rrr_reg/Q (SDFPRQLX1HxD)	0.310 &	0.726 r	clock reconvergence pessimism	0.017 3.297
U9303/Z (XNOR2X1LXD)	0.068 &	0.794 r	clock uncertainty	-0.200 3.097
U9304/Z (OAI21X1LXD)	0.036 &	0.830 f	library setup time	-0.120 2.976
I_ECC_INIT_NRZ_r_reg/D (SDFPRQLX1HxD)	0.002 &	0.833 f	data required time	2.976
data arrival time		0.833	data arrival time	-0.833
			slack (MET)	2.143

CRPR Example with Threshold removed

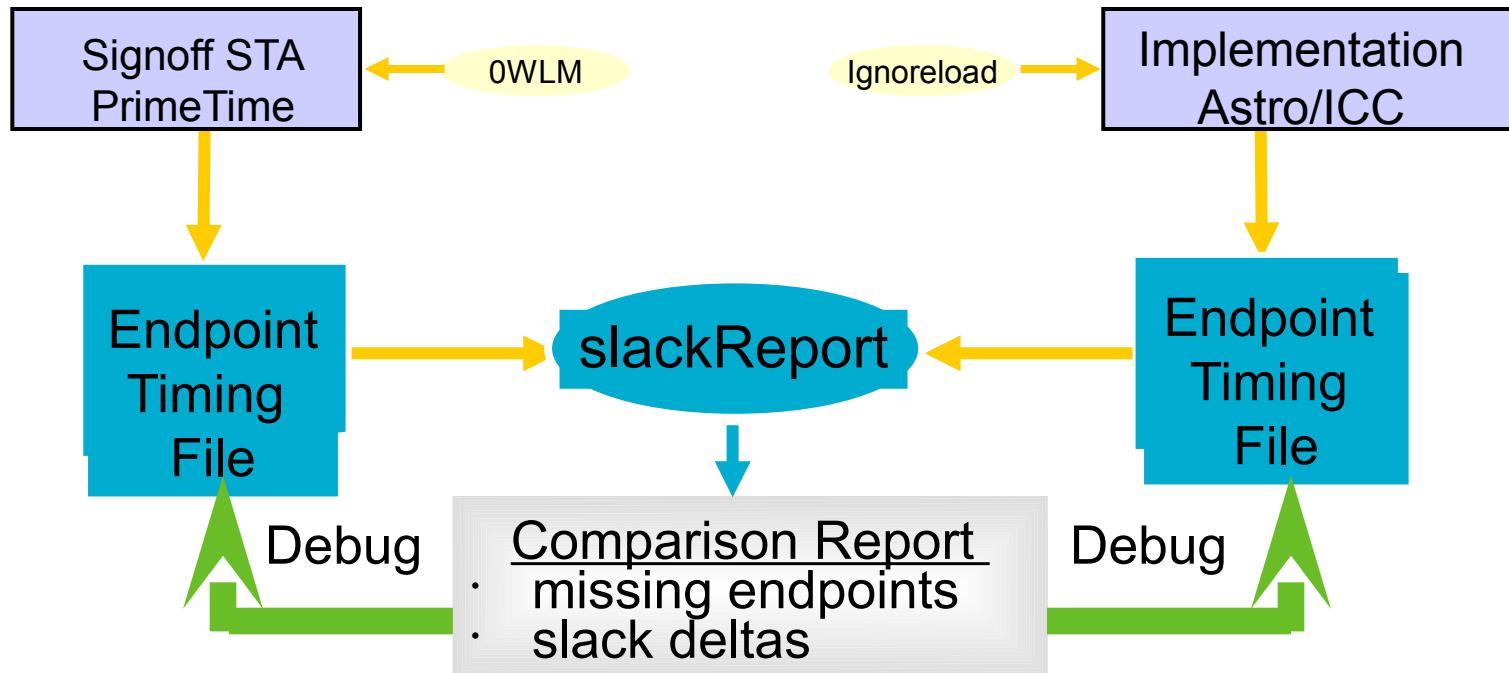
```
Startpoint: ecc_init_nrz_sync_flpsig_rrr_reg
            (rising edge-triggered flip-flop clocked by nrz_clk)
Endpoint: I_ECC_INIT_NRZ_r_reg
            (rising edge-triggered flip-flop clocked by nrz_clk)
Path Group: nrz_clk
Path Type: max
Max Data Paths Derating Factor : 1.040
Min Clock Paths Derating Factor : 0.960
Max Clock Paths Derating Factor : 1.040
```

Point	Incr	Path
clock nrz_clk (rise edge)	0.000	0.000
clock network delay (propagated)	0.416	0.416
ecc_init_nrz_sync_flpsig_rrr_reg/CP (SDFPRQLX1HxD)	0.000	0.416 r
ecc_init_nrz_sync_flpsig_rrr_reg/Q (SDFPRQLX1HxD)	0.310 &	0.726 r
U9303/Z (XNOR2X1LxD)	0.068 &	0.794 r
U9304/Z (OAI21X1LxD)	0.035 &	0.829 f
I_ECC_INIT_NRZ_r_reg/D (SDFPRQLX1HxD)	0.001 &	0.830 f
data arrival time		0.830
clock nrz_clk (rise edge)	2.900	2.900
clock network delay (propagated)	0.380	3.280
clock reconvergence pessimism	0.032	3.312
clock uncertainty	-0.200	3.112
I_ECC_INIT_NRZ_r_reg/CP (SDFPRQLX1HxD)		3.112 r
library setup time	-0.113	2.999
data required time		2.999
data required time	2.999	
data arrival time	-0.830	
slack (MET)	2.169	

Primetime Inputs : (Zero) Wire Load Model

- Wire Load Model is an estimation for nets before actual Physical Implementation
- Zero Wireload correlation is a process of validating constraints across multiple tools.
- Required to ensure the implementation tool is working on the correct timing requirements of the design.
- Zero wireload correlation is not fun!!!
 - It can be tedious to debug tool differences with respect to thousands of lines of expanded SDC.
- Zero wireload correlation is required to avoid issues late in design schedule.

Zero Wireload Correlation Flow



Zero wireload correlation - Common issues

- Consistant settings – recovery, multiple clocks, data to data checks
 - Primetime
 - timing_disable_recovery_removal_checks
 - timing_disable_clock_gating_checks
 - timing_enable_multiple_clocks_per_reg
 - ICC
 - enable_recovery_removal_arcs
 - timing_enable_multiple_clocks_per_reg
 - timing_disable_data_checks
- Complex cells in clock network – XOR/AOI/OAI
 - Primetime issues a warning

Information: A non-unate path in clock network detected.

Propagating both inverting and noninverting senses of clock (PTE-070)

- Different tools take different edges – get half cycle transfers vs.. single cycle.
· Need to set_case_analysis on these cells.

Zero wireload correlation - Common issues - cont

- Missing input delays or input delay with missing clock
 - Primetime & ICC time path starting at 0ns
 - timing_input_port_default_clock
- High fanout net overrides
 - Ensure consistent values
- Missing clock transitions
 - Do not assume defaults
 - Differences due to large transitions - high fanout overrides are not on clocks
- Constraints on hierarchical pins should be avoided
- Ordering of false paths and muticycle path
 - Can use transform_exceptions to remove ones that Primetime overrides
- Clock to data paths
 - Slope of clock in data path may need to be overridden
 - Example is check between CKA and CKB on memory
- Constraints to pins that are both endpoints and through points

Zero wlm – Troubleshooting timing mismatches

- Examine timing paths in each tool to see differences
 - Many times one report will be unconstrained or MCP due to timing exception in sdc
 - The difficult part is to figure out which constraint caused it
 - In the past this was done manually by recursively splitting the SDC in two.
 - In Primetime version 2006.06 report_timing has a new option “-exceptions all” that will display the SDC constraint affecting the timing report
 - If multiple constraints exist it will display the dominant one and the overridden one.
 - Compare the start and endpoint clocks
 - Many differences seen due to inversions in clock network

Zero wlm – CCS Issues

- Primetime and ICC use different pin capacitance with CCS models
 - Primetime derives cap from CCS receiver tables
 - Primetime (NLDM models): `pin_capacitance_range_rise/fall`
 - ICC (default): `pin_capacitance` attribute
 - ICC (enhanced): `pin_capacitance_range_rise/fall`
- For ZWLM correlation must read in non-CCS models in Primetime
- Use enhanced mode in ICC
 - `set timing_use_enhanced_capacitance_modeling true`

PrimeTime Inputs: Parasitic Extraction

- Parasitic Extraction is calculation of parasitic effects in devices and Interconnects of a circuit
- Interconnect Capacitance and Resistance is calculated
 - Layout of a design, in the form of layers
 - Mapping to a set of devices and pins
 - Cross sectional understanding of the layers
- Information is used to create a set of wires that have capacitancs and resistances
- Star-RCXT from Synopsys is a parasitic extractor tool we use to obtain the parasitics of a design

Parasitics

- Parasitics are read into Primetime with the `read_parasitics` command.
 - By default `report_annotated_parasitics` is run to show summary of annotations
 - `-quiet` option does not run `report_annotated_parasitics`
 - Used in hier flows when reading in multiple parasitic files
 - `-keep_capacitive_coupling` options used to keep coup caps (PTSI)
 - `-path` option used for block level parasitics to add instance name prefix
 - `-increment` option used to stitch parasitics together for hier nets.

- Example hier `read_parasitics` commands

```
read_parasitics -increment -quiet -path block_I1 block.spf.gz  
read_parasitics -increment top.spf.gz
```

- Example flat `read_parasitics` command

```
read_parasitics full.spf.gz
```

Parasitics - cont

- Example output of reading in parasitics

```
*****
Report : read_parasitics /home/bez/Projects/PT_CLASS/WALL/SPEF/021108/wall.spef.max.gz
  -increment
  -keep_capacitive_coupling
Design : wall
Version: Y-2006.06-SP3
Date   : Mon Feb 11 15:09:50 2008
*****

695 error(s)
Format is SPEF
Annotated nets          :      14538
Annotated capacitances  :     452056
Annotated resistances   :     437708
Annotated coupling capacitances  :    74171
Annotated PI models     :        0
Annotated Elmore delays :        0

Executing command 'report_annotated_parasitics':
*****
Report : annotated_parasitics
  -internal_nets
  -boundary_nets
Design : wall
Version: Y-2006.06-SP3
Date   : Mon Feb 11 15:09:50 2008
*****
```

Net Type	Total	Lumped	RC pi	RC network	Coupled network	Not Annotated
Internal nets						
- Pin to pin nets	21280	0	0	6552	14694	34
- Driverless nets	324	0	0	0	0	324
- Loadless nets	6	0	0	0	0	6
Boundary/port nets						
- Pin to pin nets	190	0	0	94	96	0
- Driverless nets	0	0	0	0	0	0
- Loadless nets	0	0	0	0	0	0
	21800	0	0	6646	14790	364

Parasitics cont

- report_annotated_parasitics command can be used to find info on annotated nets
 - *-list_not_annotated -max_nets num* displays not annotated nets
 - Other options such as
 - driverless_nets
 - loadless_nets
- Custom info on not_annotated nets obtained with scripts using attributes
 - has_detailed_parasitics
 - rc_network
 - rc_annotated_segment

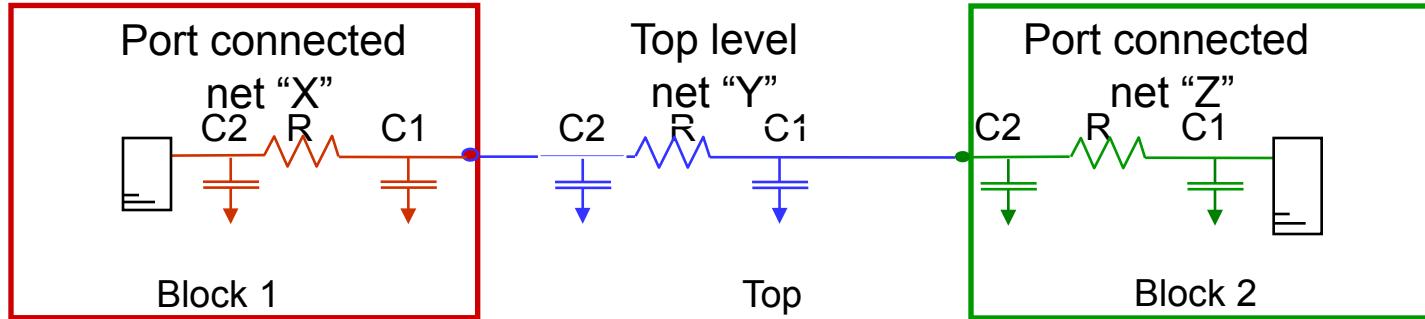
Parasitics Errors

- Most common parasitic error

Error: Cannot find net 'tile_I0/n57' in design 'wall' (DES-002)

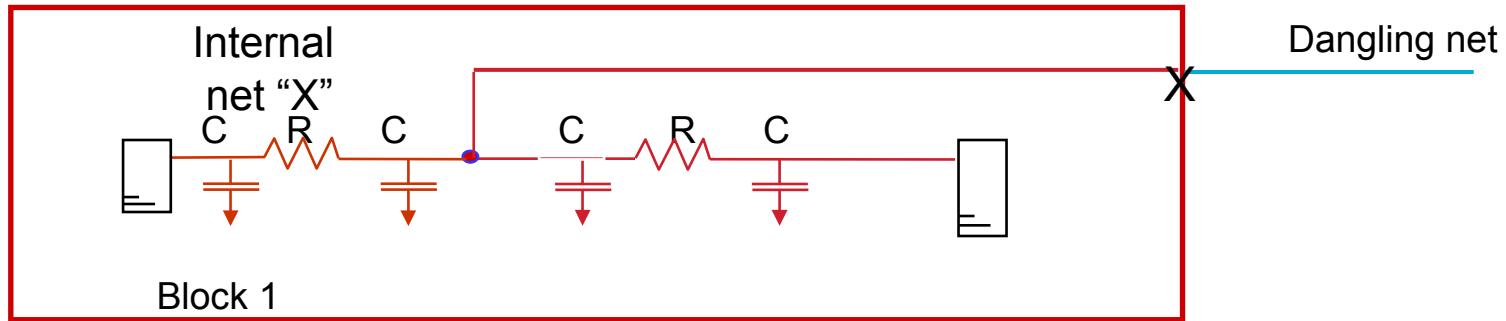
- Due to inconsistency between verilog netlist and spf file
 - Is netlist and spf from same layout database?
 - Sometimes change_names command run prior to verilog out
- report_net –ver –conn shows the net connectivity in Primetime
- Compare this with connectivity section in SPEF file
- For other errors – man acronym

Nets crossing hierarchical boundaries



- Primetime must find all segments of a hierarchical net in order to calculate an RC delay.
 - PARA-006 error issued if any net segments are missing.
 - Must resolve why net segment is missing

Dangling nets

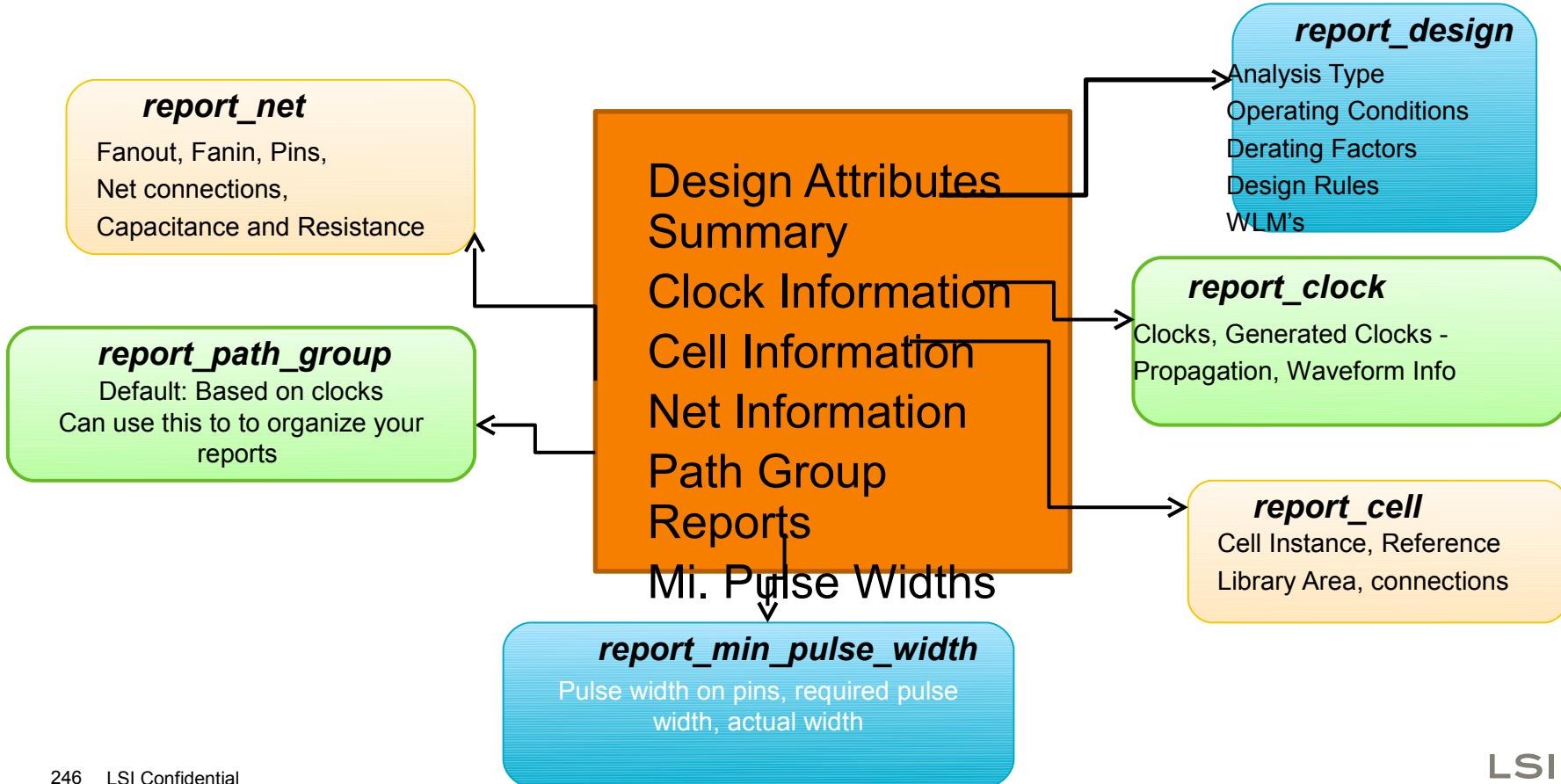


- The following has an internal net “x” that is connected to a port but does not connect to anything at the top level
 - Synopsys puts a dangling net called “SYNOPSYS_UNCONNECTED” at top level.
 - This net has multiple names through hierarchy
 - PARA-007 warning issued since parasitics are missing on net connected to hier pin.

Primetime Inputs : Constraints

- Constraints is the most important aspect of Design Closure
- Most important function for any DI person
 - Understanding the clocks in design
 - Analyzing the constraints
- Separate section in understanding constraints has been planned

Primetime Outputs: Reports – Design Information



Primetime Outputs : Reports : Design Constraints

- Always use the ***check_timing*** command to check any new design or any design with new constraints
 - Clock Definitions
 - I/O Delays
 - Timing exceptions
- Checks for
 - Constraint problems such as undefined clocking, undefined input arrival times, and undefined output constraints
 - Information about potential problems related to minimum clock separation
 - Master-slave Clocking
 - Ignored timing exceptions, combinational feedback loops, and latch fanout

MOST IMPORTANT REPORT AT ANY STAGE OF THE DESIGN

Primetime Outputs: Analysis Coverage

- Report showing information about timing checks in the current design/instance
- Generating information about timing checks is most critical for new designs
- Perform these checks right after you resolve errors found while using the check_timing
- Perform additional checks whenever significant changes are made to the design or the timing assertions.

GIVES A PERCENTAGE OF TESTED AND UNTESTED CONSTRAINTS

Analysis Coverage

- Setup
- Hold
- No-change
- Minimum period
- Recovery
- Removal
- Minimum pulse width
- Ensures that the analysis was complete with the constraints given
- Determine which checks are untested in different configurations and modes
 - E.g.: **report_analysis_coverage -status_details {untested} -check_type {setup}**
- Clock separation (master-slave)
- Clock-gating setup
- Clock-gating hold
- Output setup
- Output hold
- Maximum skew

Primetime Outputs : Constraint Violators (report_constraint)

- produces a summary of the constraint violations in the design
 - includes the amount by which a constraint is violated
 - Includes the design object that is the worst violator
- Types of Violations

Timing Constraints

- Maximum path delay and setup
- Minimum path delay and hold
- Recovery time
- Removal time
- Clock-gating setup and hold
- Minimum pulse width high or low
- Minimum period at a clock pin
- Maximum clock skew

Design Rules

- Total Net Capacitance
 - Min & Max
- Pin Transition Limits
 - Min and Max
- Sum of Fanout Loads
 - Min and Max

Primetime Output : Timing reports

- Help analyze the timing in the design
- The report_timing command provides options to control reporting of the following:
 - Number of paths
 - Types of paths
 - Amount of detail
 - Startpoints, endpoints, and intermediate points along the path

Understanding Timing Reports

- Primetime analyzes a Data Path in the following way
 - From constraints and /or spef values
 - Calculates the Data Arrival Time
 - Calculates the Data Required Time
 - If Data Required Time > Data Arrival Time -> Path PASSES
 - If Data Arrival Time > Data Required Time -> Path FAILS

Understanding Timing Reports

```
report_timing -from ecc_init_nrz_sync_flpsig_rrr_reg/CP -to I_ECC_INIT_NRZ_r_reg/D
```

Header		Startpoint: ecc_init_nrz_sync_flpsig_rrr_reg (rising edge-triggered flip-flop clocked by nrz_clk)	Endpoint: I_ECC_INIT_NRZ_r_reg (rising edge-triggered flip-flop clocked by nrz_clk)	
		Path Group: nrz_clk	Capture clock	
		Path Type: max	Report is for Setup	
Point		Incr	Path	
Data Arrival	clock_nrz_clk (rise edge)	0.000	0.000	Clock Start Edge / time
	clock_network_delay (ideal)	0.000	0.000	Ideal launch Clock
	ecc_init_nrz_sync_flpsig_rrr_reg/CP (SDFPRQLX1LXD)	0.000	0.000 r	
	ecc_init_nrz_sync_flpsig_rrr_reg/Q (SDFPRQLX1LXD)	0.138	0.138 r	
	U9666/Z (XNOR2X1LXD)	0.048	0.187 r	Stage Delay
	U9667/Z (XAI21X1LXD)	0.027	0.213 f	
Data Required	I_ECC_INIT_NRZ_r_reg/D (SDFPRQLX1LXD)	0.000	0.213 f	
	data_arrival_time	0.213		
	clock_nrz_clk (rise edge)	2.900	2.900	Clock Capture Edge / time
	clock_network_delay (ideal)	0.000	2.900	Ideal capture Clock
Slack	I_ECC_INIT_NRZ_r_reg/CP (SDFPRQLX1LXD)		2.900 r	
	library_setup_time	-0.075	2.825	
	data_required_time		2.825	
data arrival time		-0.213		
slack (MET)		2.612		

Understanding Timing Reports

```
report_timing -from HDC_RMEN -to lli_read_logic_lli_seed_fifo_u_mem/RMEN -nosplit -input_pins -nets -significant_digits 3
```

leader	Startpoint: HDC_RMEN (input port)		
	Endpoint: lli_read_logic_lli_seed_fifo_u_mem (rising edge-triggered flip-flop clocked by I_SYMCLK)		
	Path Group: I_SYMCLK		
	Path Type: max		
Point	Fanout	Incr	Path
input external delay		0.000	0.000 f
HDC_RMEN (in)		0.000 &	0.000 f
HDC_RMEN (net)	1		
PORT_BUF_619_0/A (BUFX8LXD)		0.000 &	0.000 f
PORT_BUF_619_0/Z (BUFX8LXD)		0.079 &	0.079 f
PORT_BUF_NET_619_0 (net)	3	0.006 &	0.085 f
icc_1237/A (BUFX8LXD)		0.058 &	0.143 f
icc_1237/Z (BUFX8LXD)			
n3029 (net)	3		
lli_read_logic_lli_seed_fifo_u_mem/RMEN (V111HDSPWV224M4X10B2L8HS)		0.035 &	0.179 f
data arrival time			0.179
Data arrival	max_delay	0.560	0.560
	clock reconvergence pessimism	0.000	0.560
	clock uncertainty	-0.200	0.360
	library setup time	-0.391	-0.031
	data required time		-0.031
Data required	data required time		-0.031
	data arrival time		-0.179
	slack (VIOLATED)		-0.210
Slack	LSI	SI *	

Understanding Timing Reports

```
report_timing -from ecc_init_nrz_sync_flpsig_rrr_reg/CP -to I_ECC_INIT_NRZ_r_reg/D
```

Header		Combo-path	
		Incr	Path
Startpoint:	ecc_init_nrz_sync_flpsig_rrr_reg (rising edge-triggered flip-flop clocked by nrz_clk)		
Endpoint:	I_ECC_INIT_NRZ_r_reg (rising edge-triggered flip-flop clocked by nrz_clk)		
Path Group:	nrz_clk		
Path Type:	max		
Point			
Data Arrival	clock nrz_clk (rise edge) clock network delay (ideal) ecc_init_nrz_sync_flpsig_rrr_reg/CP (SDFPRQLX1LXD) ecc_init_nrz_sync_flpsig_rrr_reg/U (SDFPRQLX1LXD) U9666/Z (XNOR2X1LXD) U9667/Z (OAI21X1LXD) I_ECC_INIT_NRZ_r_req/D (SDFPRQLX1LXD) data arrival time	Launch Flop – CP / Clk -pin 0.000 0.000 0.000 0.138 0.048 0.027 0.000 0.213	r r r r r f f f 0.213
		Capture Flop-D-pin	Incremental cell delays
Data required	clock nrz_clk (rise edge) clock network delay (ideal) I_ECC_INIT_NRZ_r_req/CP (SDFPRQLX1LXD) library setup time data required time	2.900 0.000 2.900 -0.075 2.825	2.900 2.900 2.900 r 2.825 2.825
Slack	data required time data arrival time slack (MET)	2.825 -0.213 2.612	

Interpreting Timing Reports – Find the Difference

Startpoint: u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_ctl/ir_reg_24
(rising edge-triggered flip-flop clocked by ref_clk_hrtbt_rcvr)

Endpoint: u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_regs/reg7_reg_31
(rising edge-triggered flip-flop clocked by ref_clk_hrtbt_rcvr)

Path Group: ref_clk_hrtbt_rcvr

Path Type: max

Max Data Paths Derating Factor : 1.04000

Min Clock Paths Derating Factor : 0.96000

Max Clock Paths Derating Factor : 1.04000

Point	Incr	Path
clock ref_clk_hrtbt_rcvr (rise edge)	0.00000	0.00000
clock network delay (ideal)	0.00000	0.00000
u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_ctl/ir_reg_24/CP (SDFPRQLX1HxD)	0.00000	0.00000 r
u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_ctl/ir_reg_24/Q (SDFPRQLX1HxD)	0.46497	0.46497 r

Startpoint: u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_regs/reg3_reg_0
(rising edge-triggered flip-flop clocked by ref_clk_hrtbt_rcvr)

Endpoint: u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_regs/reg7_reg_31
(rising edge-triggered flip-flop clocked by ref_clk_hrtbt_rcvr)

Path Group: ref_clk_hrtbt_rcvr

Path Type: max

Max Data Paths Derating Factor : 1.04000

Min Clock Paths Derating Factor : 0.96000

Max Clock Paths Derating Factor : 1.04000

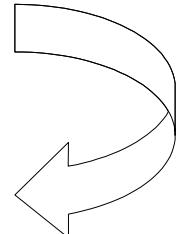
Point	Incr	Path
clock ref_clk_hrtbt_rcvr (rise edge)	0.00000	0.00000
clock network delay (propagated)	4.44363	4.44363
u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_regs/reg3_reg_0/CP (SDFPRQX4HxD)	0.00000	4.44363 r
u_intc_wrap/nvv_apr_ins/nvv_kumeran_ins/u_phy_uctl_top/u_phy_uctl_xN_core/u_phy_uctl_xN_core_regs/reg3_reg_0/Q (SDFPRQX4HxD)		

Next Steps for Primetime - Training

- Analyze timing reports in Detail
- How to run sessions in DMSA
- How to fix Violations
 - Fix_eco_drc
 - Fix_eco_timing
 - Dmsasizer
- How to fix leakage power : Running LIPO



Storage. Networking. **Accelerated.**TM





Storage. Networking. **Accelerated.**TM