PRIME VLSI

If their is AND gate and i want it to define as exception, which one you will select false path or disable timing arc?

False path	Disable timing arc
Point to point timing exception	Pin to pin timing exception
Declaring as FP removes all timing constraints from the path but prime time still calculates path delay but not reported even though that path delay is more or less.	Disable timing analysis for a specific pin/cell/port and don’t calculate path delay
Removing timing constraints on that particular path	Using this we can remove affected/unwanted cell/pin/port from timing analysis. related arc itself is removed timing analysis

EP= o/p port (or) register data in
SP= i/p port (or) register clock pin
	Disable a particular arc within a cell sometimes its also defined with SP and EP disables timing arcs from a start port to an end port of a cell. This constraints is from Pin to Pin of a cell.

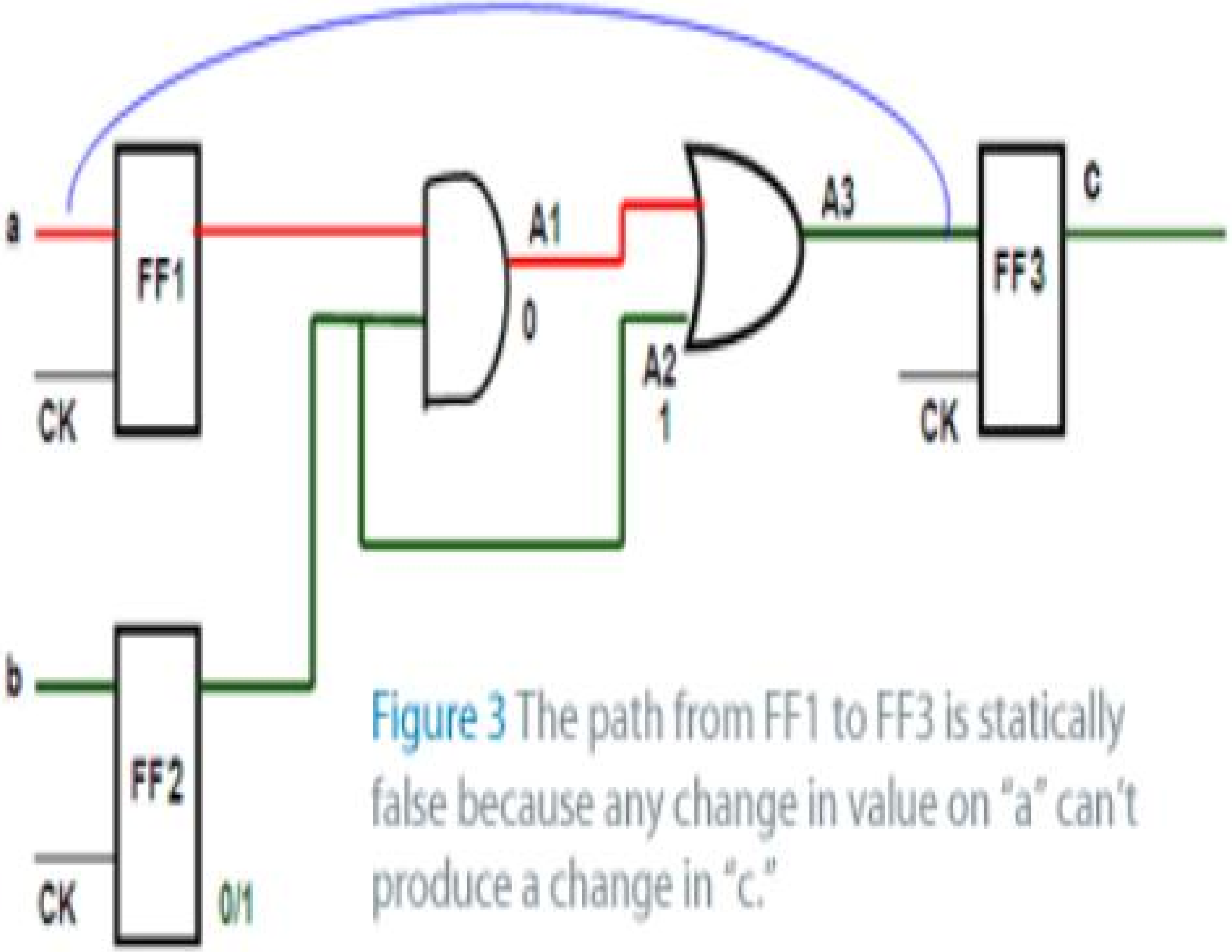
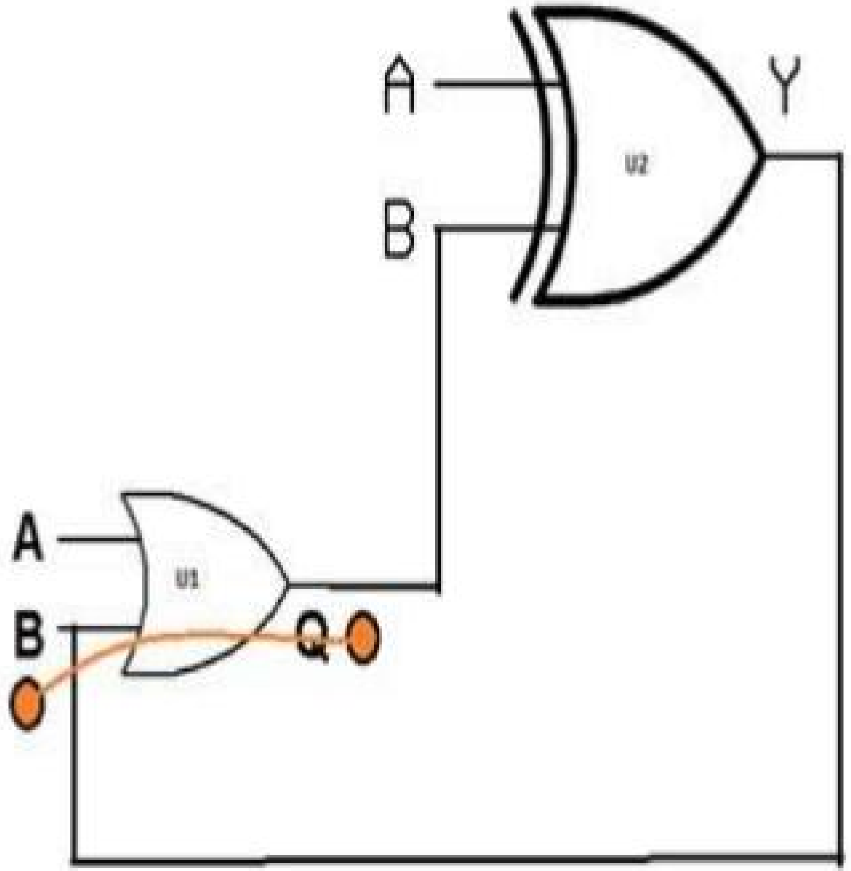


Figure 3 The path from FF1 to FF3 is statically false because any change in value on "a" can't produce a change in "c."



CROSS CORNERS

In fabrication, a process corner represents a three or six sigma variation from nominal doping concentrations and other parameters in transistors on a wafer. This variation can cause significant changes in the duty cycle and slew rate of digital signals, and can sometimes result in total failure of the entire system.
Variation may occur for many reasons, such as minor changes in the humidity or temperature changes or due to the position of the die relative to the center of the wafer.

Types of corners

1. **FEOL**

These corners will affect the performance of devices,
The **front-end-of-line** (**FEOL**) is the first portion of IC fabrication where the individual devices (transistors, capacitors, resistors, etc.) are patterned in the semiconductor.
FEOL generally covers everything but not including the deposition of metal interconnect layers.

2. **BEOL**

back end of line is the second portion of IC fabrication where the individual devices (transistors, capacitors, resistors, etc.) get interconnected with wiring on the wafer, the metallization layer.

Three corners exist: typical, fast and slow.
Fast and slow corners For example, a corner designated as FS denotes fast NFETs and slow PFETs.

There are five possible corners

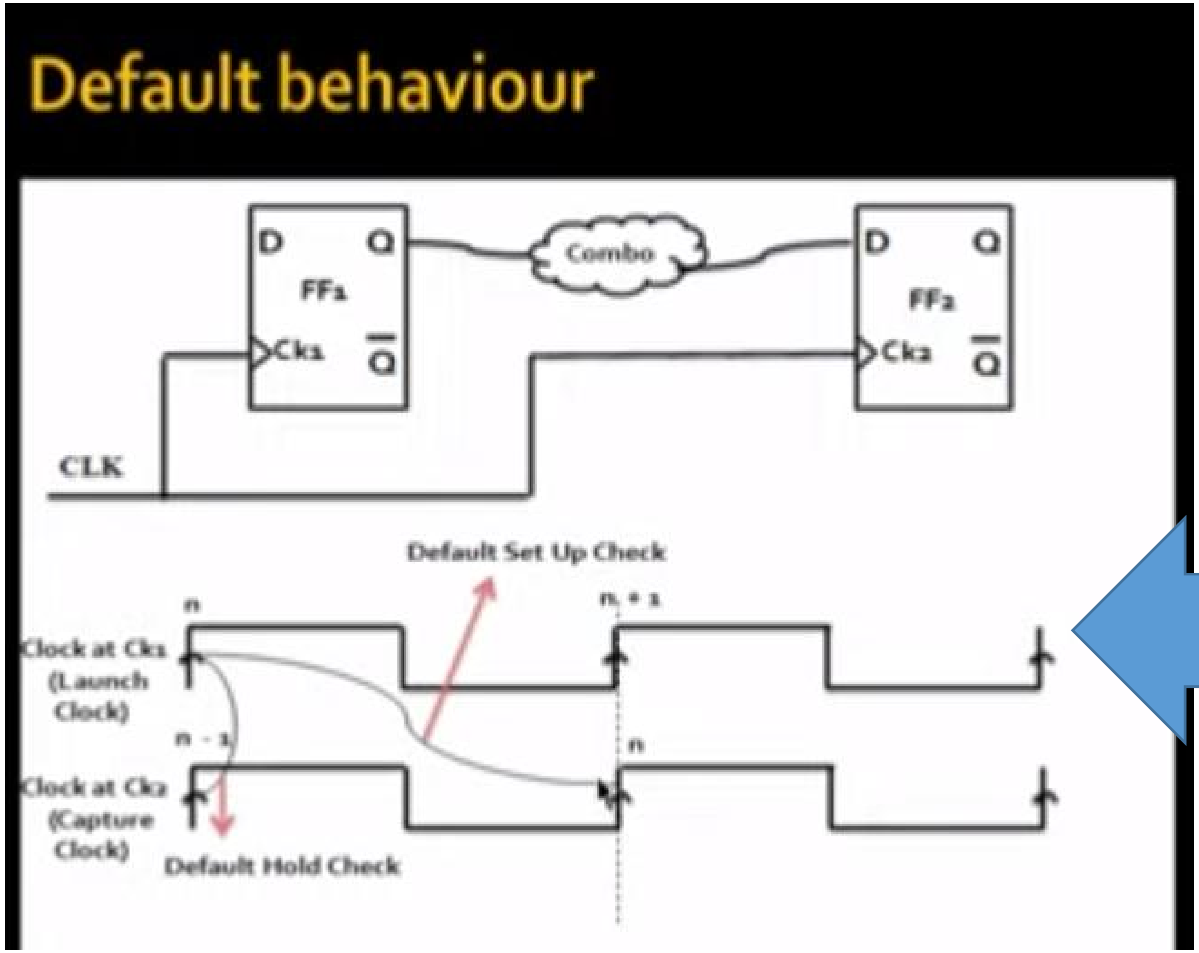
1. typical-typical (TT)
2. fast-fast (FF)
3. slow-slow (SS)
4. fast-slow (FS)
5. slow-fast (SF)

The first three corners (TT, FF, SS) are called even corners, because both types of devices are affected evenly, and generally do not adversely affect the logical correctness of the circuit. The resulting devices can function at slower or faster clock frequencies.

The last two corners (FS, SF) are called "skewed" corners, and are cause for concern. This is because one type of FET will switch much faster than the other, and this form of imbalanced switching can cause one edge of the output to have much less slew than the other edge.

Cross corner means if we run even and odd corners at a time.
Cross corner Is considered due to temperature inversion in lower technologies.

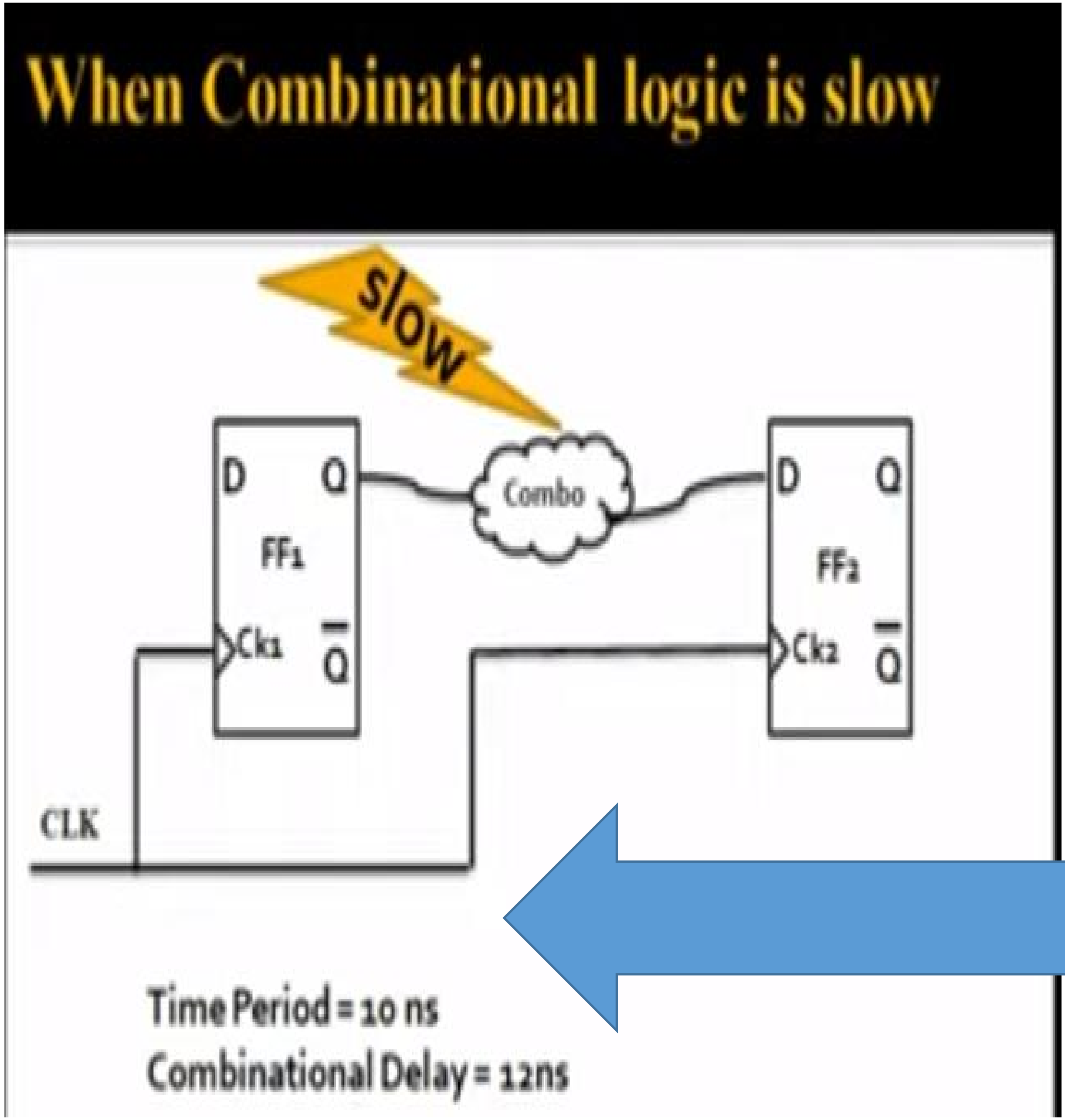
MULTI CYCLE PATH



Default behaviour

By default in timing checks what happen setup is checked at next clock edge of same cycle and hold is checked for zeroth cycle i.e. same clock edge .

Data has to propagate in one time period cycle.



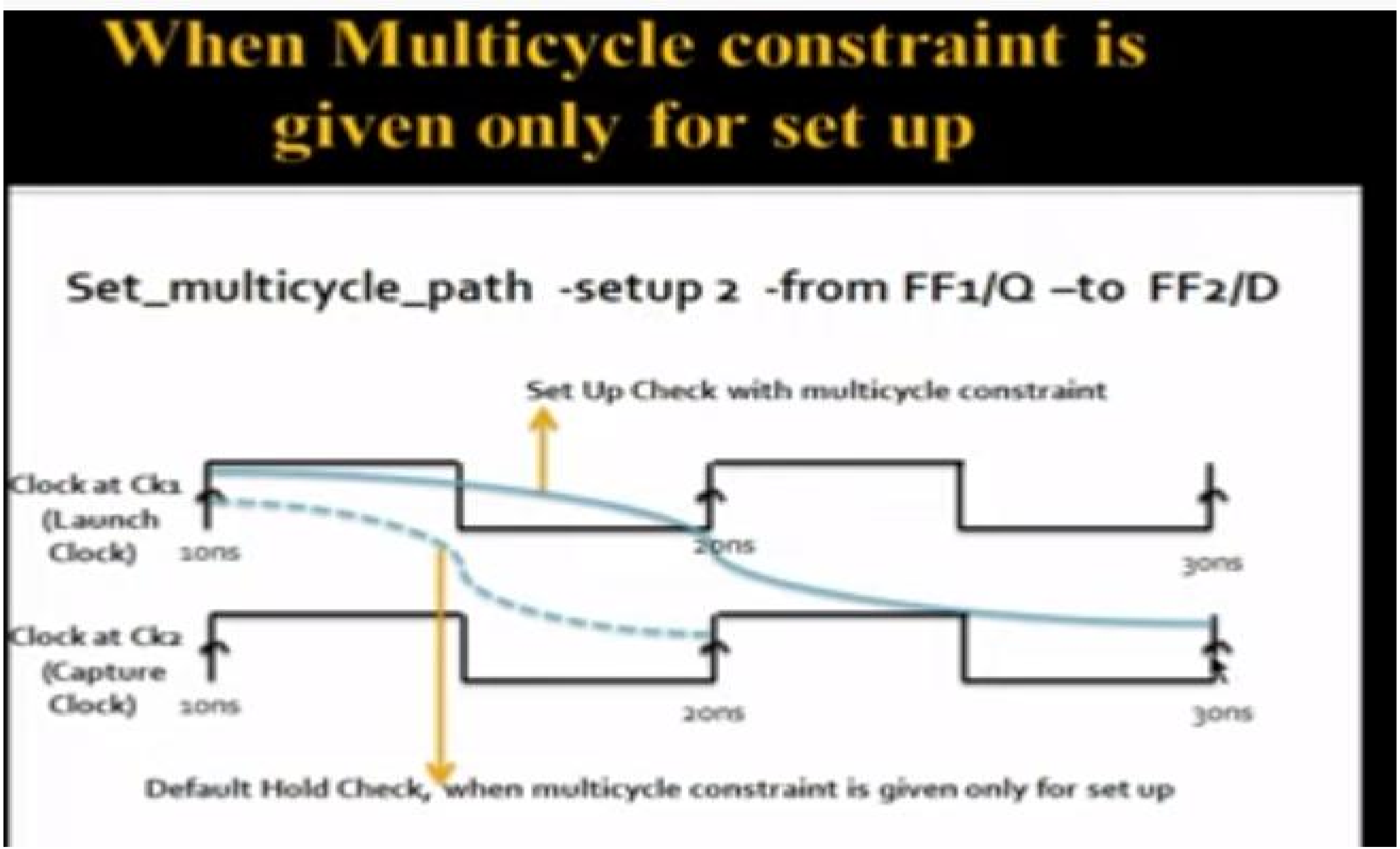
When Combinational logic is slow

Time Period = 10 ns
Combinational Delay = 12ns

Here time period is 10ns if we do setup check after one time cycle by default i.e. 10ns its incorrect because for complete analysis of tcomb is 12 ns which is greater than one clock cycle (10 ns)

In this case we go for MCP here we need to define where to do setup and hold check.

If we don't define set/hold it will do default check as in previous slide.

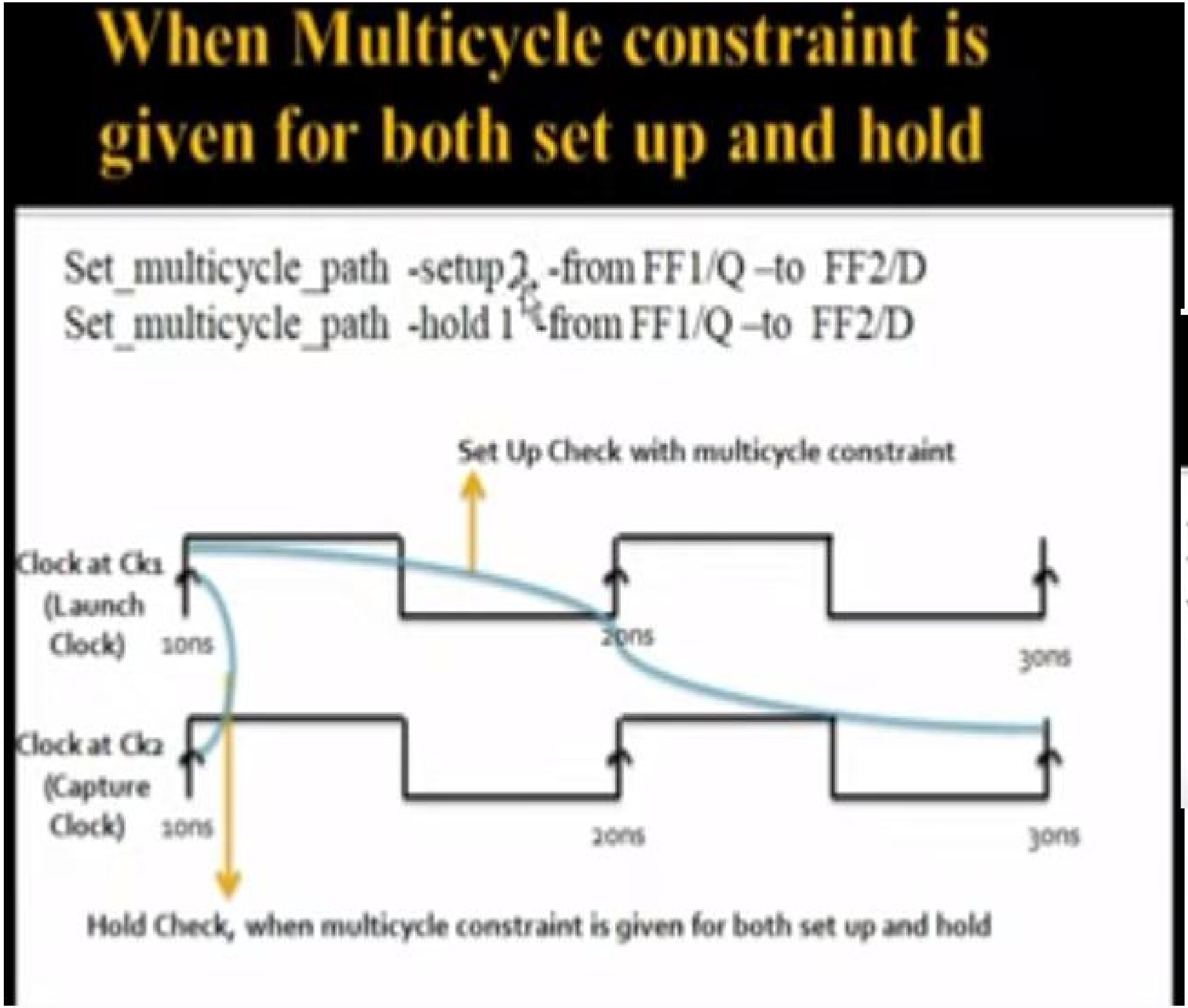


When Multicycle constraint is given only for set up

Set_multicycle_path -setup 2 -from FF1/Q –to FF2/D

Set Up Check with multicycle constraint

Default Hold Check, when multicycle constraint is given only for set up

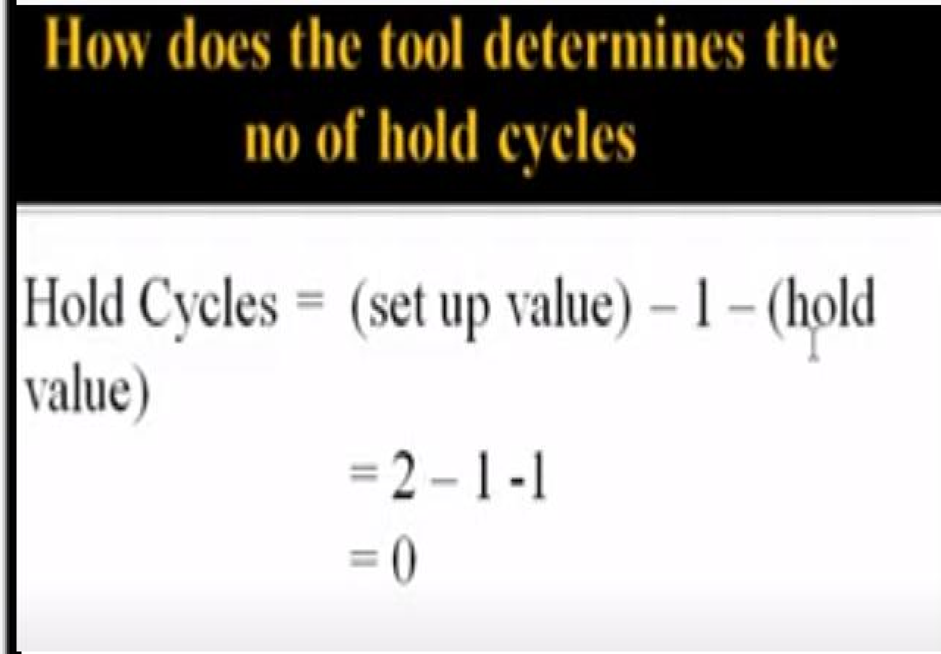


When Multicycle constraint is given for both set up and hold

Set_multicycle_path -setup 2 -from FF1/Q –to FF2/D
Set_multicycle_path -hold 1 -from FF1/Q –to FF2/D

Set Up Check with multicycle constraint

Hold Check, when multicycle constraint is given for both set up and hold

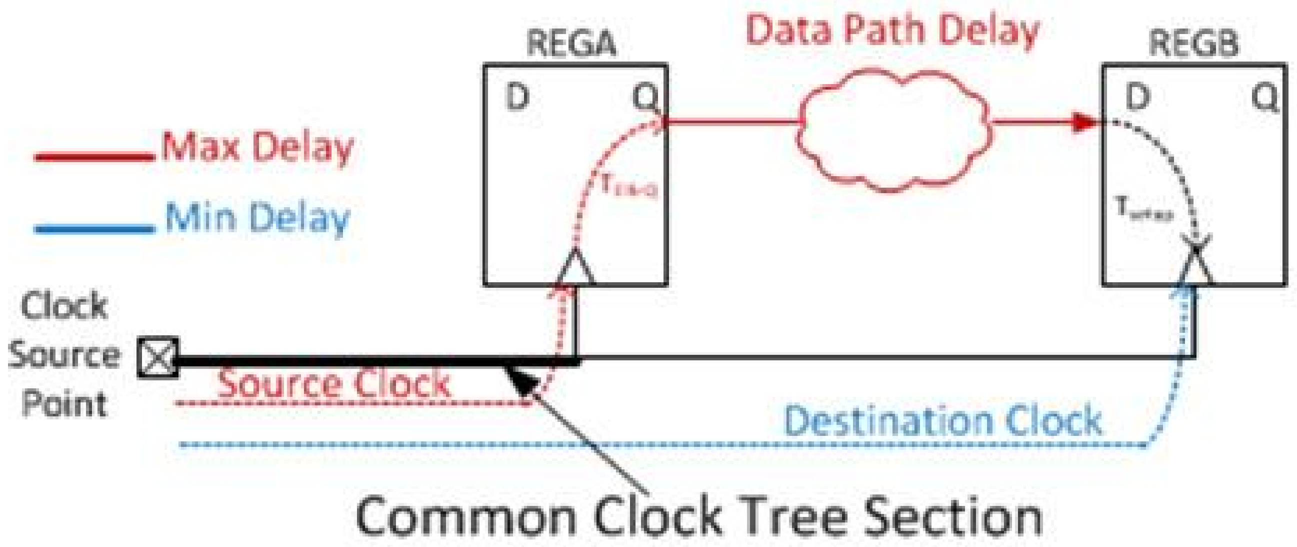


How does the tool determines the no of hold cycles

Hold Cycles = (set up value) – 1 – (hold value)

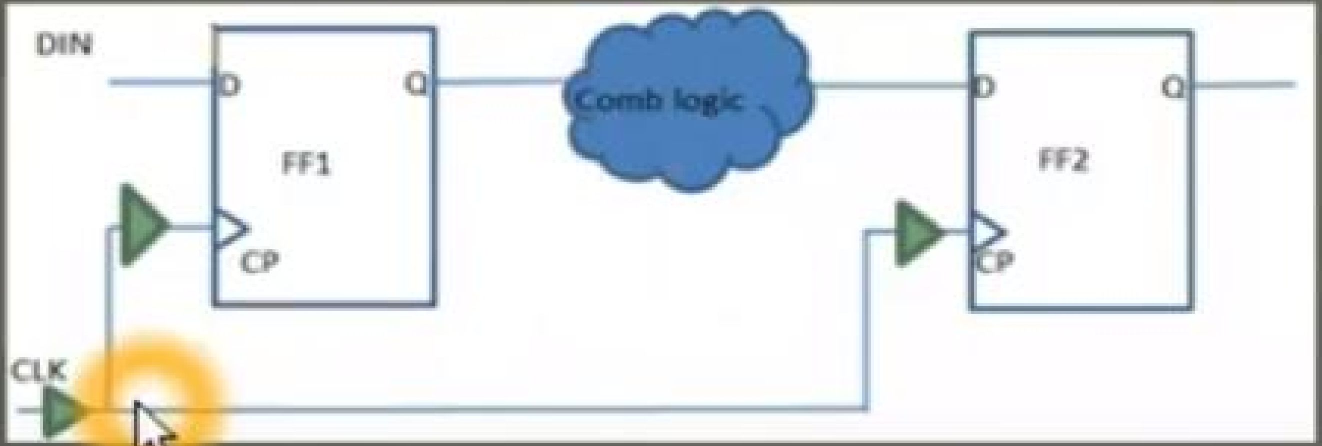
= 2 – 1 -1
= 0

The clock pessimism **shows the absolute amount of extra clock skew introduced due to source and destination clocks are reported with different types of delay even on their common circuitry**. In reality, it is not possible for common circuitry to be analyzed with 2 different delays (occurred due to OCV) at the same time.

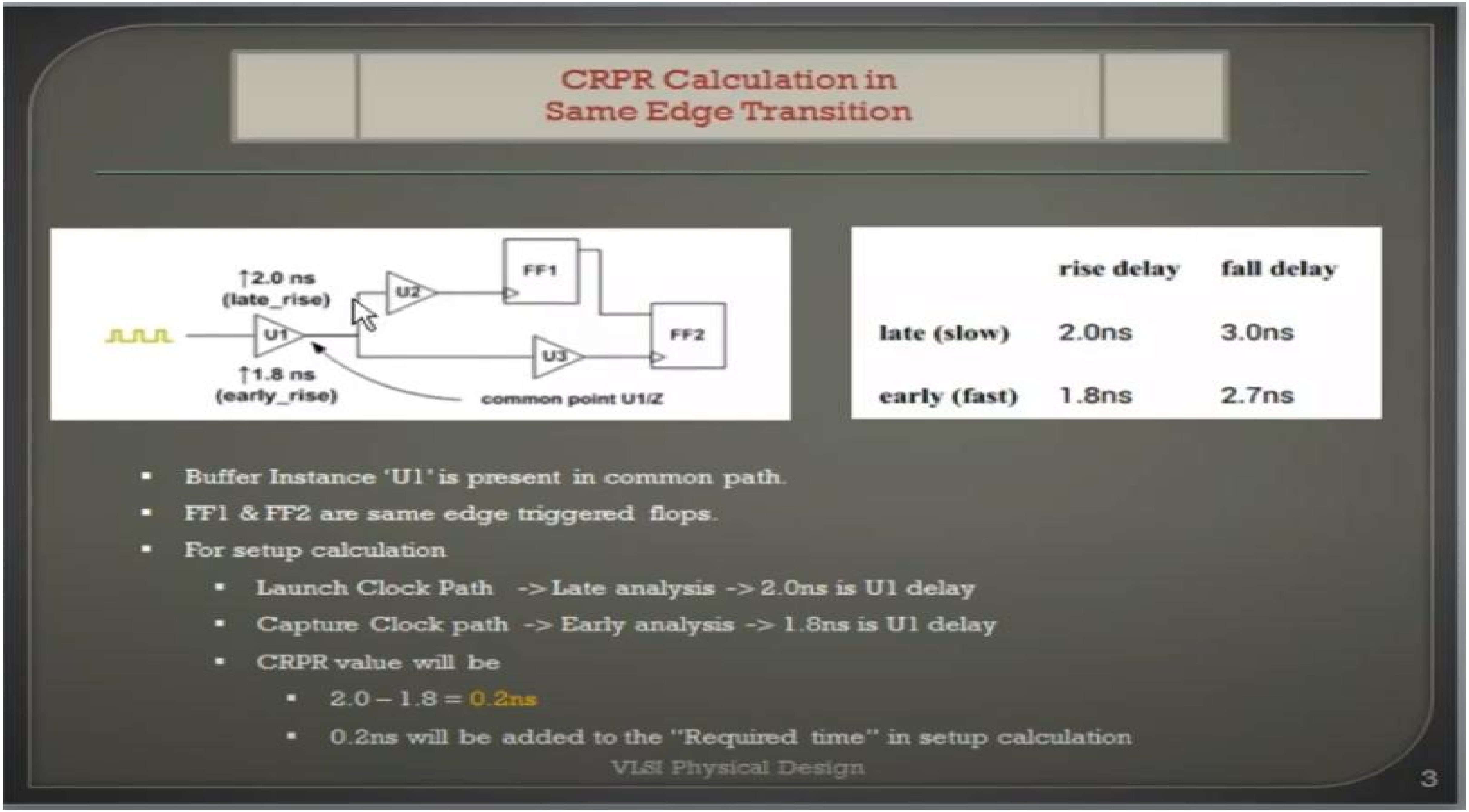


CRPR Introduction

- Clock Reconvergence Pessimism Removal
- Clock cells can not have different delays at same time.
- Controlled by "**timing_clock_reconvergence_pessimism**"variable.
  - If you set this variable to "normal" (the default), opposite edges are considered.
  - If you set this variable to"same_transition", then no CRPR is applied to opposite polarity edges.



Analysis Mode	Timing Check	Launch Clock Path	Data Path	Capture Clock Path
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

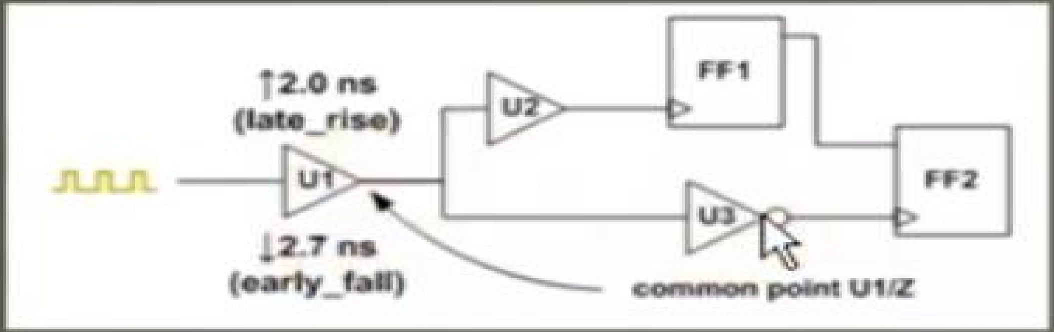


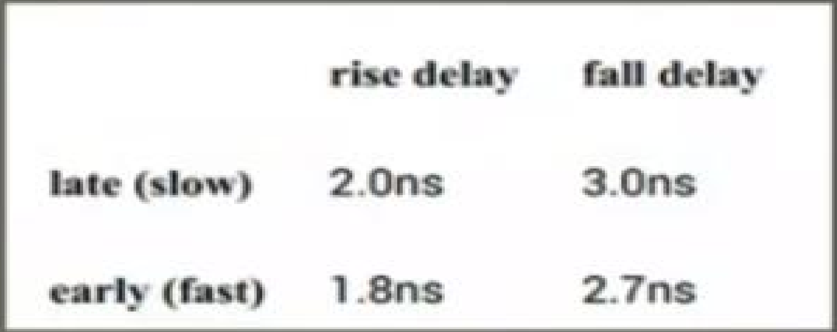
CRPR Calculation in Same Edge Transition

	rise delay	fall delay
late (slow)	2.0ns	3.0ns
early (fast)	1.8ns	2.7ns

- Buffer Instance 'U1' is present in common path.
- FF1 & FF2 are same edge triggered flops.
- For setup calculation
  - Launch Clock Path -> Late analysis -> 2.0ns is U1 delay
  - Capture Clock path -> Early analysis -> 1.8ns is U1 delay
  - CRPR value will be
    - 2.0 – 1.8 = 0.2ns
    - 0.2ns will be added to the "Required time" in setup calculation

CRPR Calculation in Opposite Edge Transition





	rise delay	fall delay
late (slow)	2.0ns	3.0ns
early (fast)	1.8ns	2.7ns

- Buffer Instance 'U1' is present in common path. Inverter 'U3'is present.
- FF1 is a rising edge triggered flop and FF2 is a falling edge triggered flop.
- For setup calculation
  - CRP calculation

    rise_CRP = max_rise_arrival - min_rise_arrival = 2.0ns - 1.8ns = 0.2ns

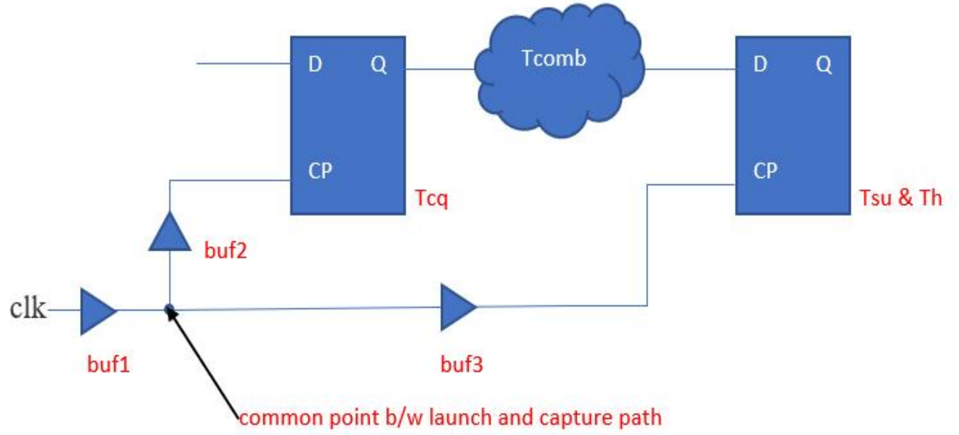
    fall_CRP = max_fall_arrival - min_fall_arrival = 3.0ns - 2.7ns = 0.3ns
  - CRPR value = minimum(rise_CRP,fall_CRP)
    - CRPR value = minimum(0.2ns,0.3ns) = 0.2ns
    - 0.2ns will be added to the "Required time" in setup calculation

- CPPR is primarily due to OCV variations while CRPR is an architectural artifact.

Clock reconvergence pessimism (CRP) is a difference in delay along the common part of the launching and capturing clock paths.

the most common causes of CRP are reconvergent paths in clock network, and different min and max delay of cells in the clock network.

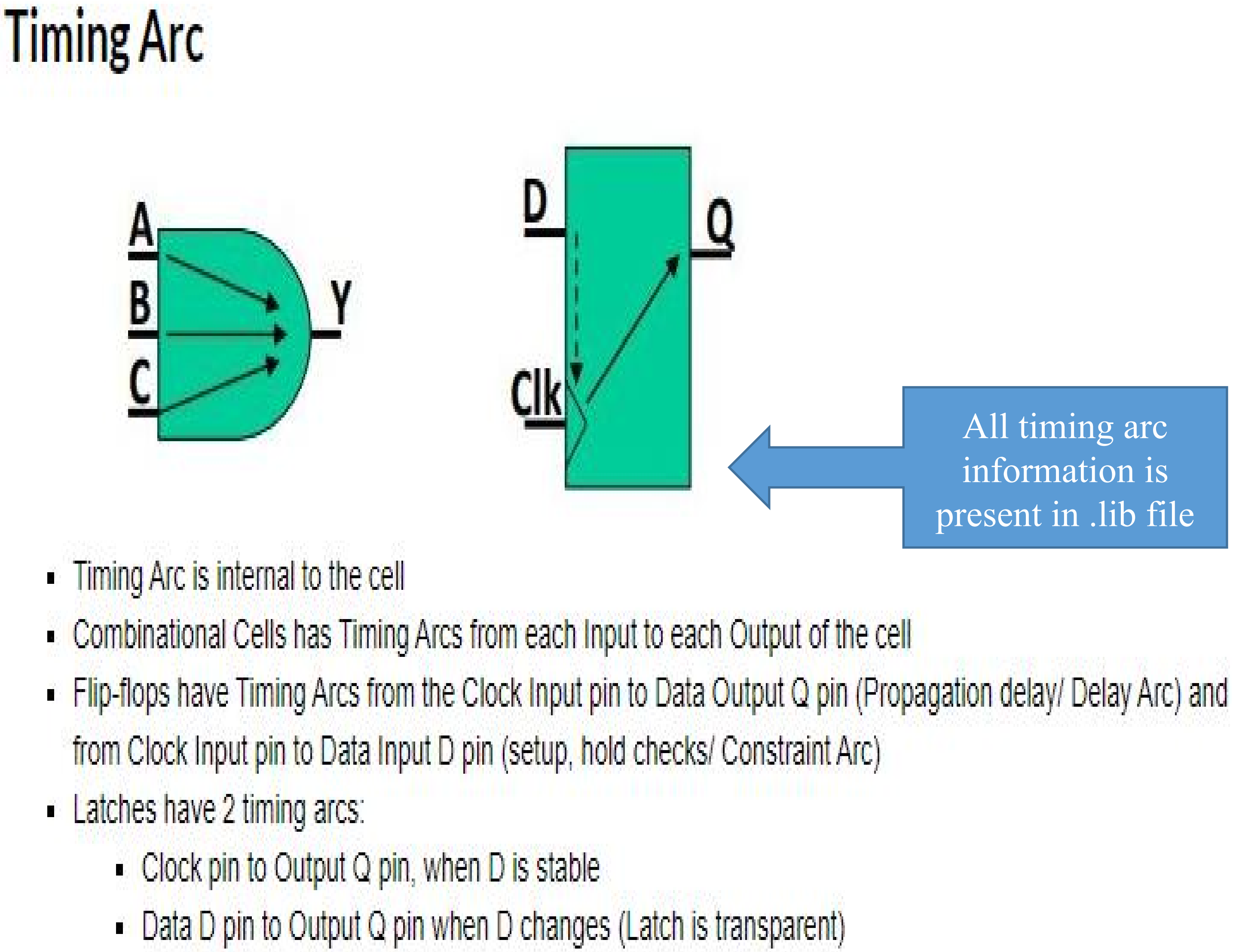
CRP is an undesired effect.



Difference between max and min delay is CRPR

Cell name	Max delay	Min delay
buf1	0.70	0.60
buf2	0.65	0.55
buf3	0.75	0.45
Tcomb	3.6	2.5
Tcq	0.60	0.48

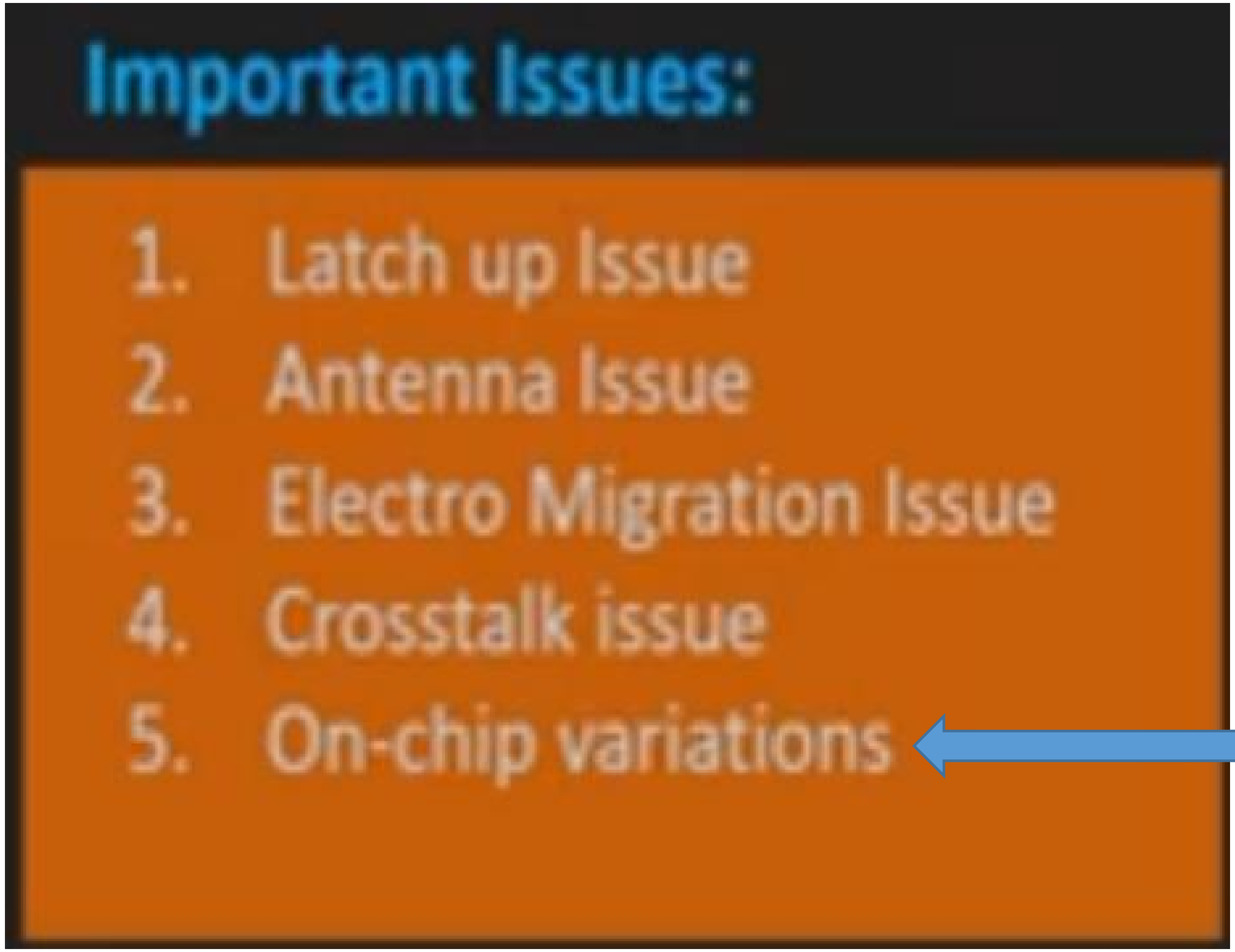
3D TIMING ARC



Timing Arc

- Timing Arc is internal to the cell
- Combinational Cells has Timing Arcs from each Input to each Output of the cell
- Flip-flops have Timing Arcs from the Clock Input pin to Data Output Q pin (Propagation delay/ Delay Arc) and from Clock Input pin to Data Input D pin (setup, hold checks/ Constraint Arc)
- Latches have 2 timing arcs:
  - Clock pin to Output Q pin, when D is stable
  - Data D pin to Output Q pin when D changes (Latch is transparent)

All timing arc information is present in .lib file



Important Issues:

1. Latch up Issue
2. Antenna Issue
3. Electro Migration Issue
4. Crosstalk issue
5. On-chip variations

ON CHIP VARIATION

- Fabrication issue
- After fabrication all IC inside the same wafer and all transistor inside the same IC may or may not have same electrical behaviour. We can see slight variations.
- These variations is due to PVT conditions.
- Here we can see two types of variations

[1] Local Variation

Inter chip variation i.e. die to die

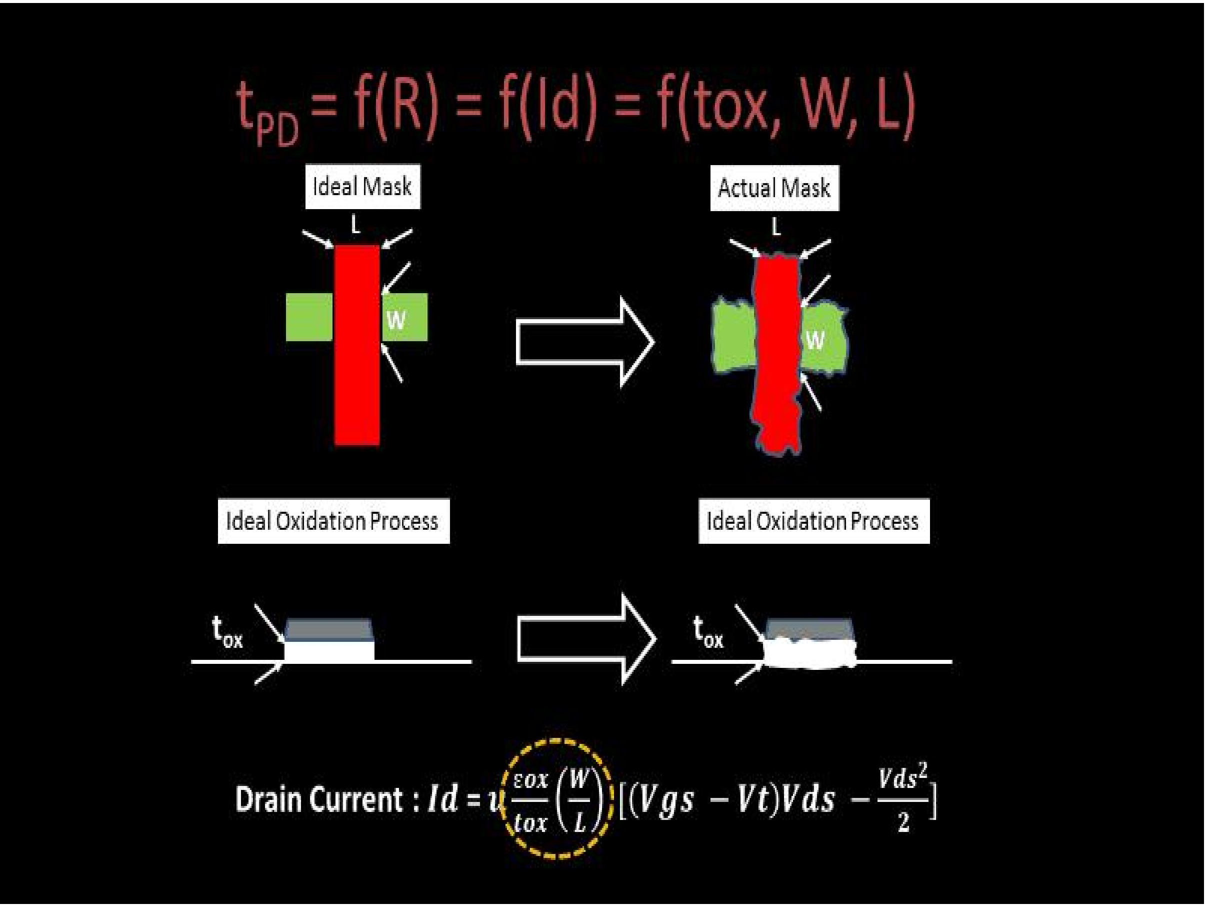
[2] Global Variation

Intra chip variation

Sources Of OCV.

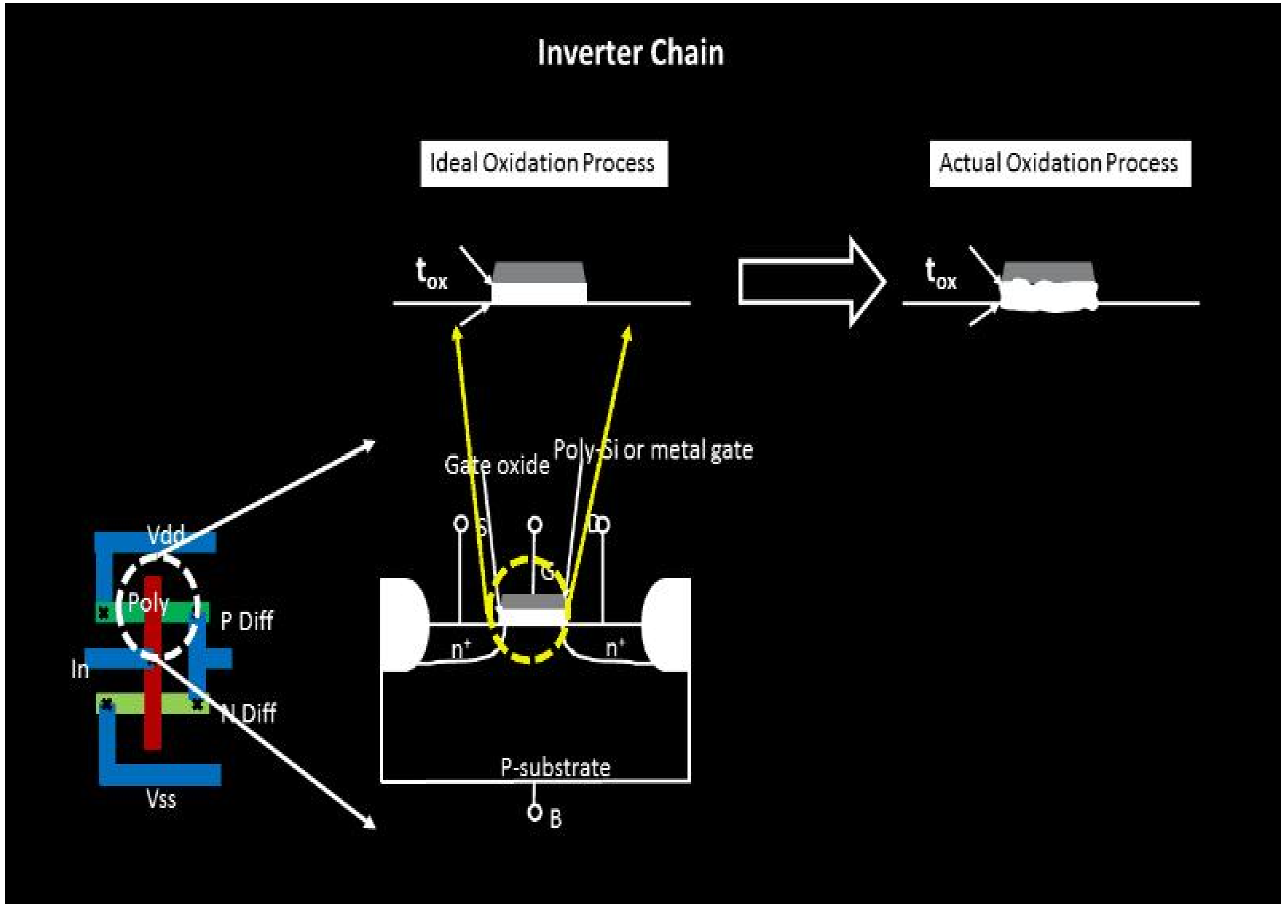
[1] Etching

[2] Oxide Thickness



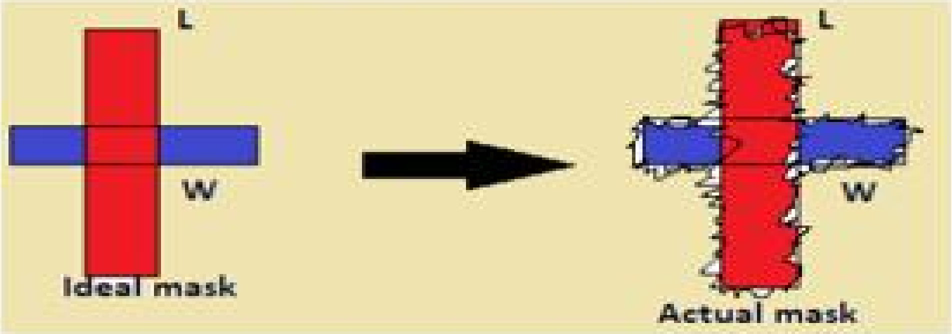
$$t_{PD} = f(R) = f(Id) = f(tox, W, L)$$

Drain Current : $Id = \upsilon \frac{\varepsilon ox}{tox}\left(\frac{W}{L}\right)\left[(Vgs - Vt)Vds - \frac{Vds^2}{2}\right]$

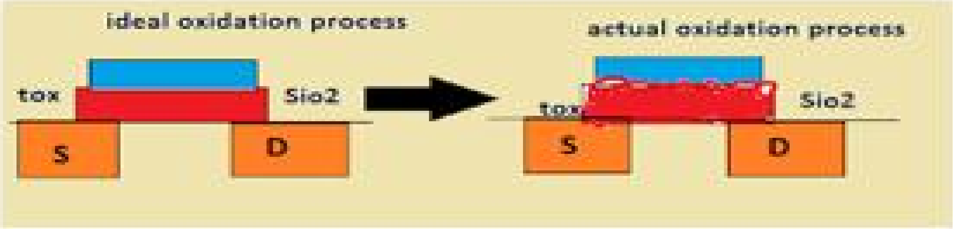


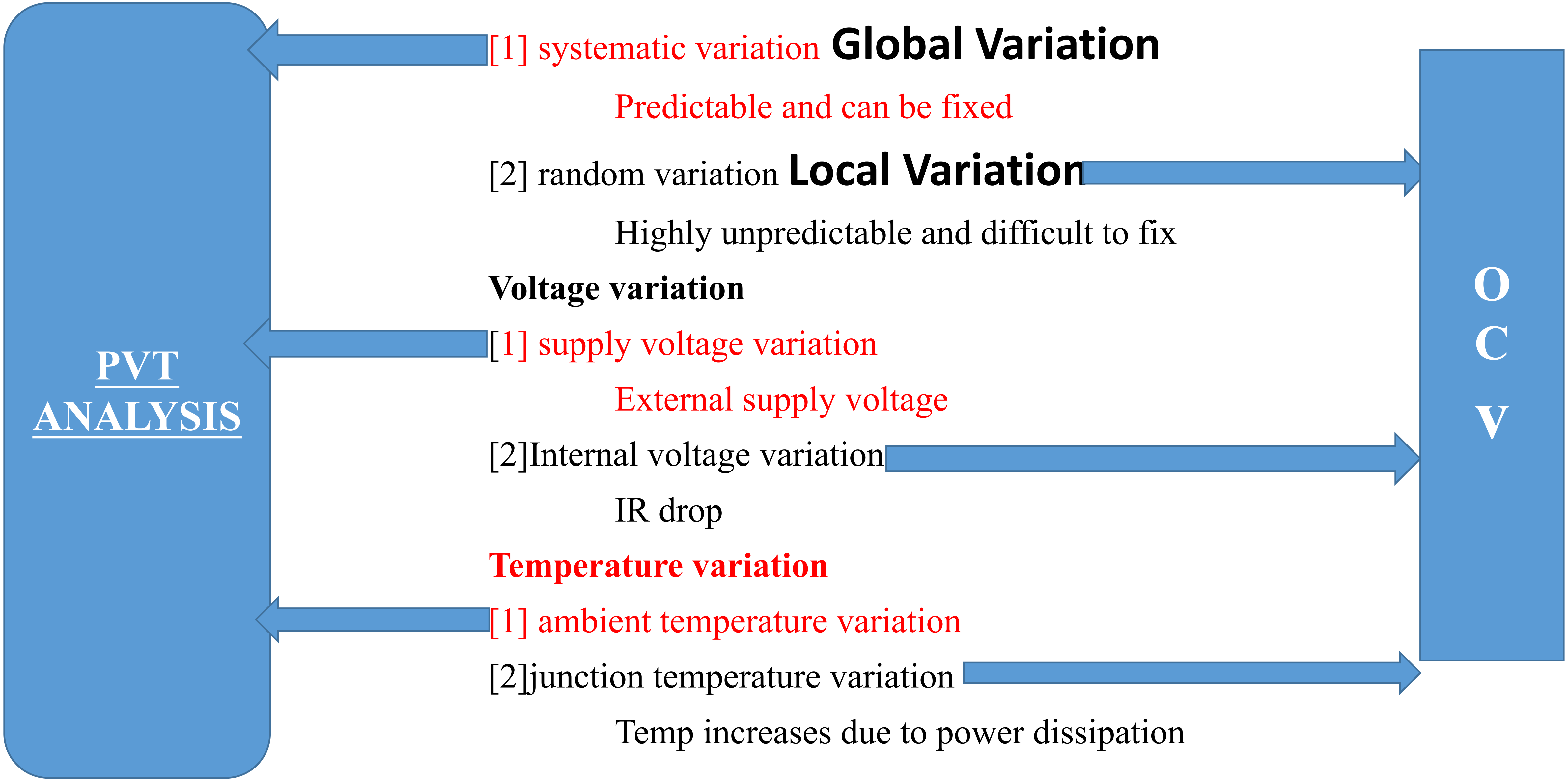
Inverter Chain

Etching: Etching is one of the sources of variation, which is used the define the structure of a transistor. In practical case the width and length of transistor is different from the expected W and L ratio.The small change for one transistor will result in big impact on chain of billion of transistors.



Oxide thickness: Oxide thickness is another source of variation,In practical case the oxide thickness is not uniform along the channel.





PVT ANALYSIS

Process variation

[1] systematic variation **Global Variation**

Predictable and can be fixed

[2] random variation **Local Variation**

Highly unpredictable and difficult to fix

Voltage variation

[1] supply voltage variation

External supply voltage

[2]Internal voltage variation

IR drop

Temperature variation

[1] ambient temperature variation

[2]junction temperature variation

Temp increases due to power dissipation

OCV

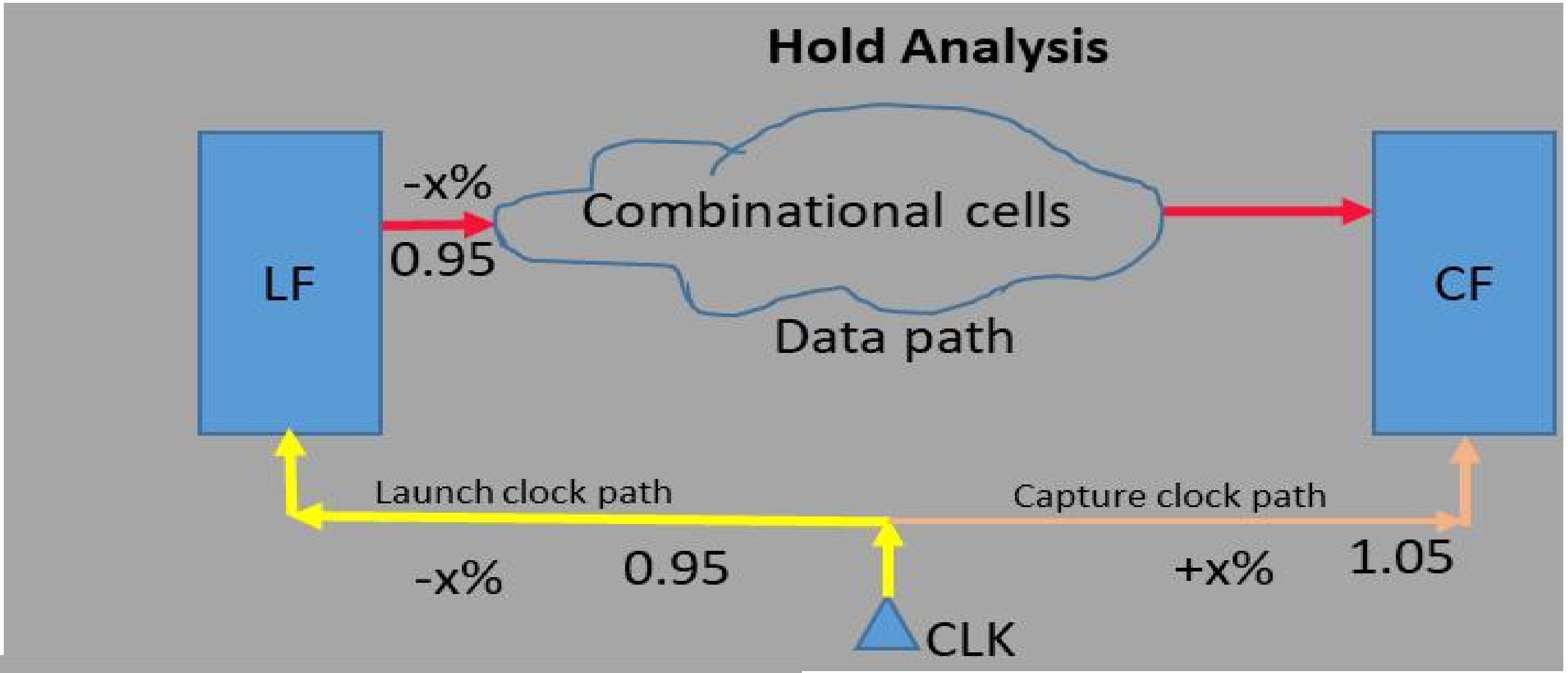
REMEDY FOR THIS VARIATION

On Chip Variation could lead to post-silicon failure i.e. chip failue if it is not taken care while designing.

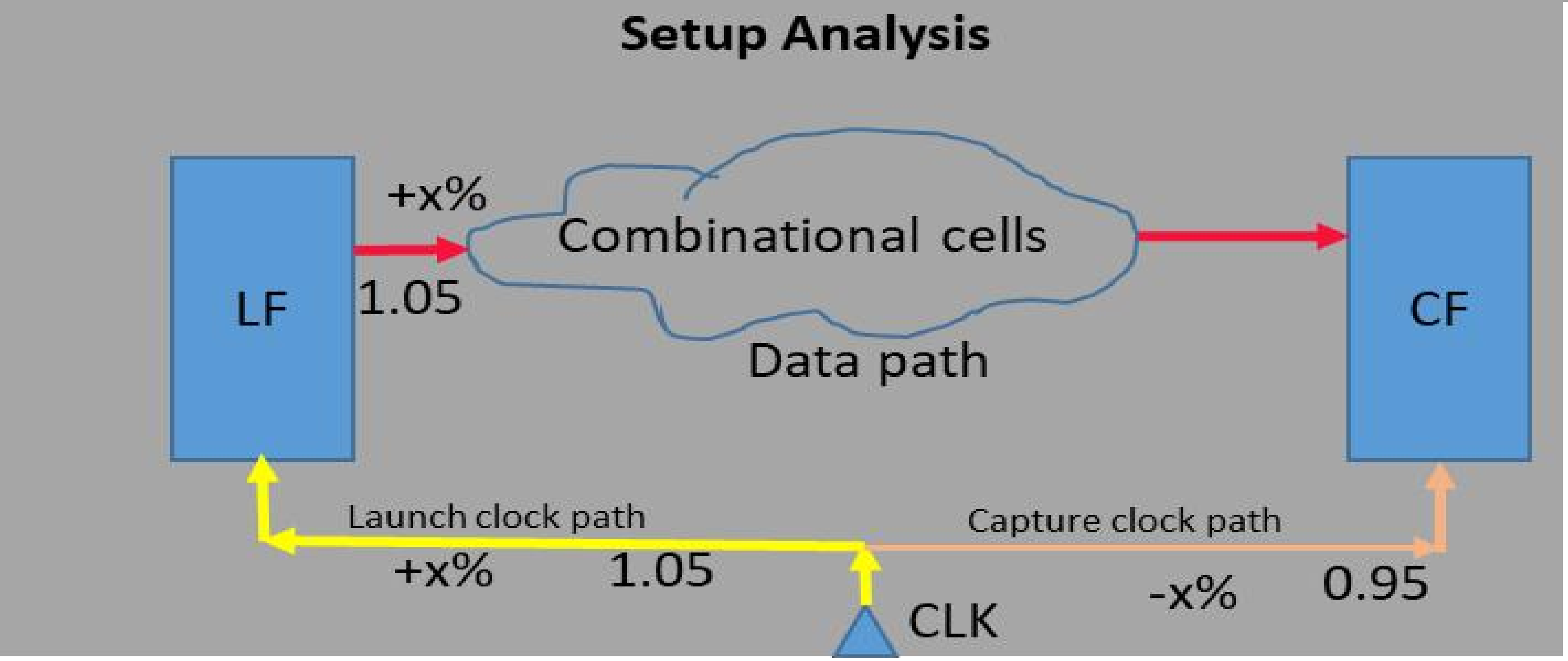
Fabrication process variations could either increase or decrease the delay of a cell. So we need to set early and late values while setting the derate factor.

To take care of OCV we need to add some pessimism in the timing of standard cells. We basically apply ±x% of additional delay to all the standard cells. Which is called OCV derate.

Hold Analysis



Setup Analysis



ISSUES IN OCV

- Same shirt does not fit all like that fixed derate factor does not fit to all cells because not all the cell in the path is simultaneously become varied it may vary from one path to another
- Its highly pessimistic
- Its inaccurate for lower technologies (<20nm)

AOCV

- ➢Its introduced for below 20nm technology node.
- ➢Instead of applying fixed derate for all ,the derate factor is calculated based on

[1]path depth (cell delay)

Derate is Inversely proportional to depth.

[2]distance(net delay)

As distance increases system variation increases.

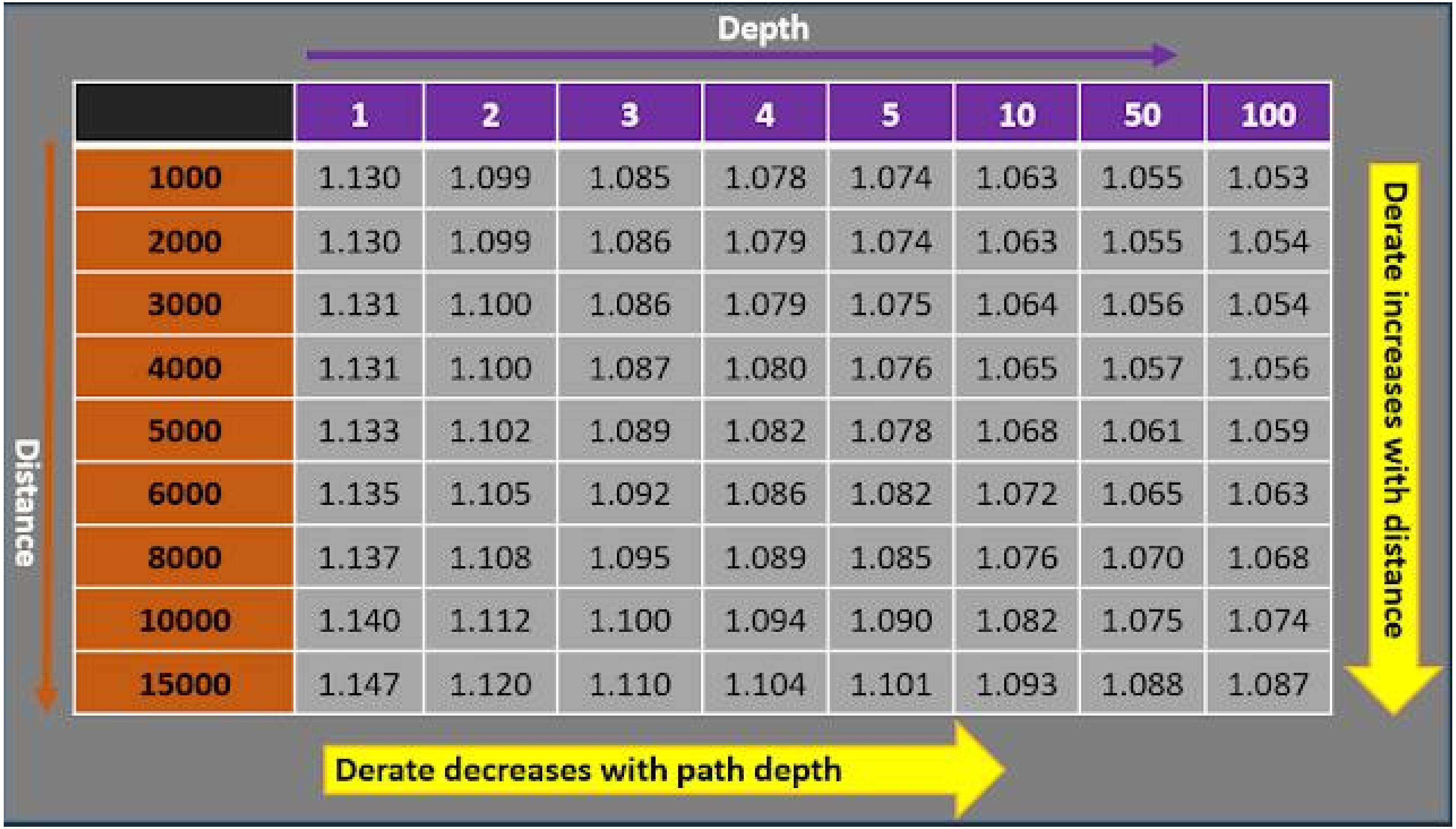
[3]cell type [optional]

[4]drive strength [optional]

Table is of two types

1D (either depth or distance is considered)

2D(both distance and depth are considered)



Depth

	1	2	3	4	5	10	50	100
1000	1.130	1.099	1.085	1.078	1.074	1.063	1.055	1.053
2000	1.130	1.099	1.086	1.079	1.074	1.063	1.055	1.054
3000	1.131	1.100	1.086	1.079	1.075	1.064	1.056	1.054
4000	1.131	1.100	1.087	1.080	1.076	1.065	1.057	1.056
5000	1.133	1.102	1.089	1.082	1.078	1.068	1.061	1.059
6000	1.135	1.105	1.092	1.086	1.082	1.072	1.065	1.063
8000	1.137	1.108	1.095	1.089	1.085	1.076	1.070	1.068
10000	1.140	1.112	1.100	1.094	1.090	1.082	1.075	1.074
15000	1.147	1.120	1.110	1.104	1.101	1.093	1.088	1.087

Distance

Derate increases with distance

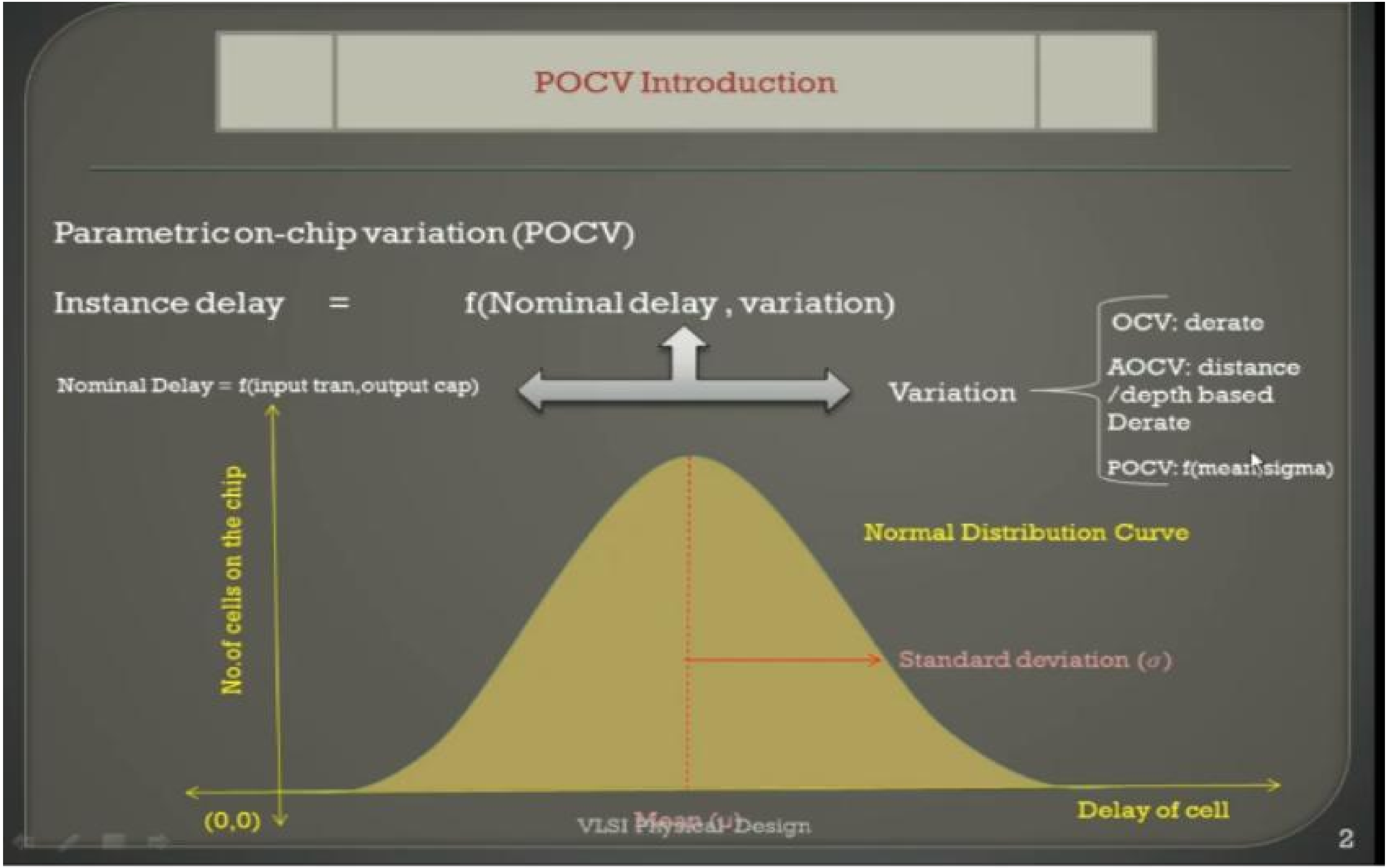
Derate decreases with path depth

Issues in AOCV

- Inaccurate for below 10nm
- Not reduces the pessimism

POCV

- POCV are more realistic than AOCV and OCV
- In POCV derate is calculated based on delay standard variation (σ) of the cell.
- It uses delay sigma to calculate derate factor.
- The information of POCV variation is directly provided in the library itself in LVF format. In LVF format there are two indexes used one for input transition and other for output load. 3sigma = cell delay + net delay+ switching.



POCV Introduction

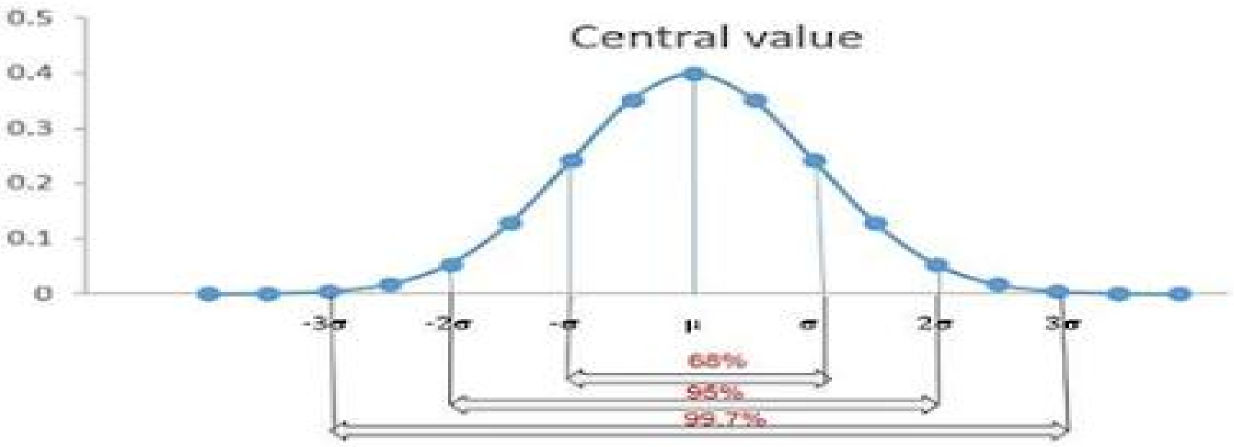
Parametric on-chip variation (POCV)

Instance delay = f(Nominal delay , variation)

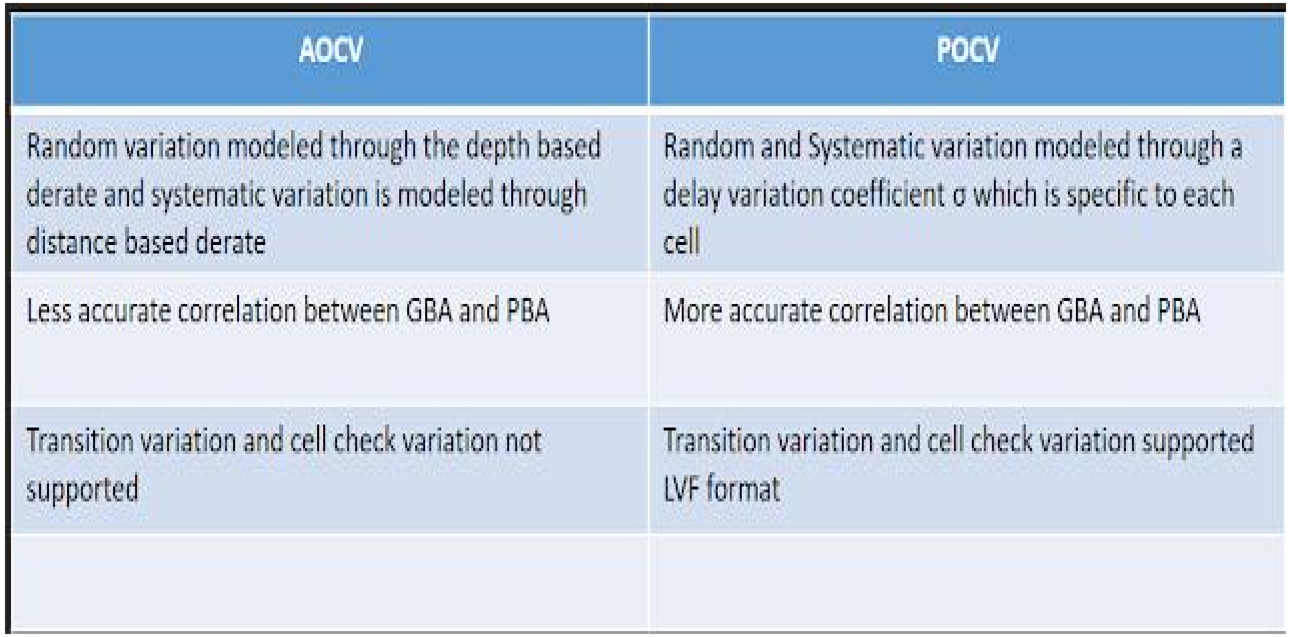
Nominal Delay = f(input tran,output cap)

Variation

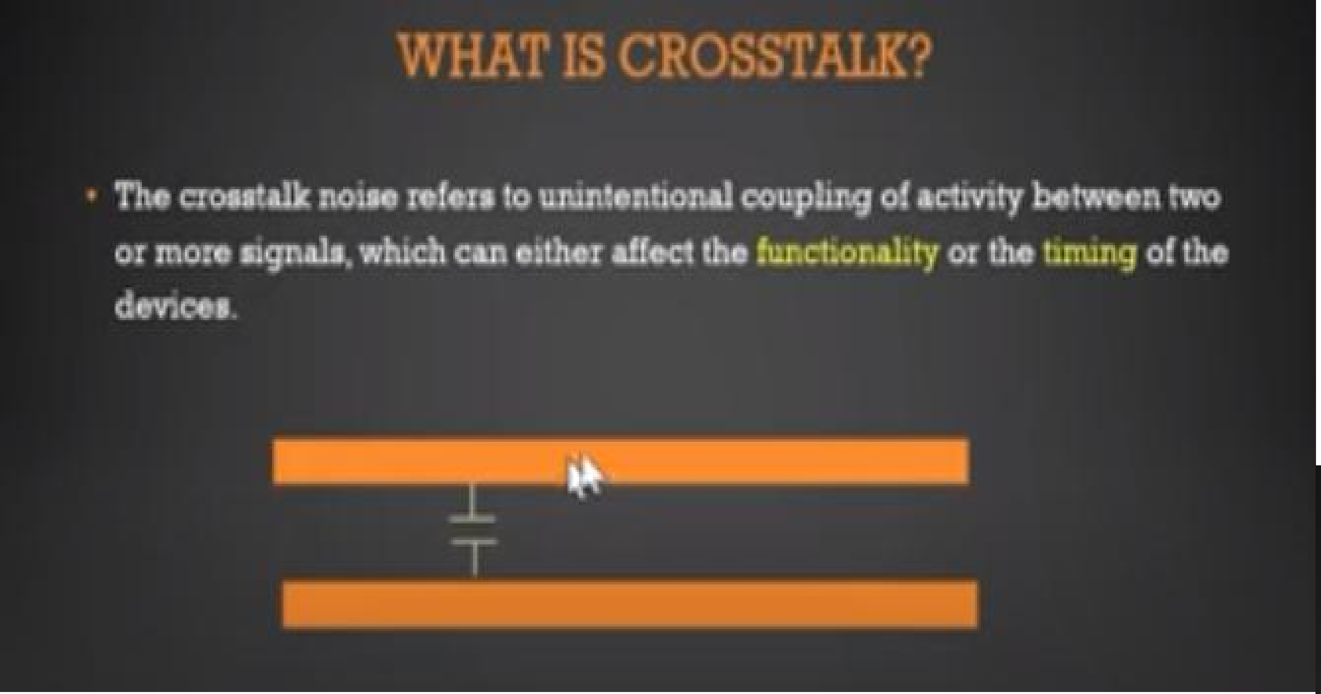
- OCV: derate
- AOCV: distance /depth based Derate
- POCV: f(mean,sigma)



In normal distribution 68% of data falls within the 1σ range, 95% data falls within 2σ and 99.7% data fall within the range of 3σ.

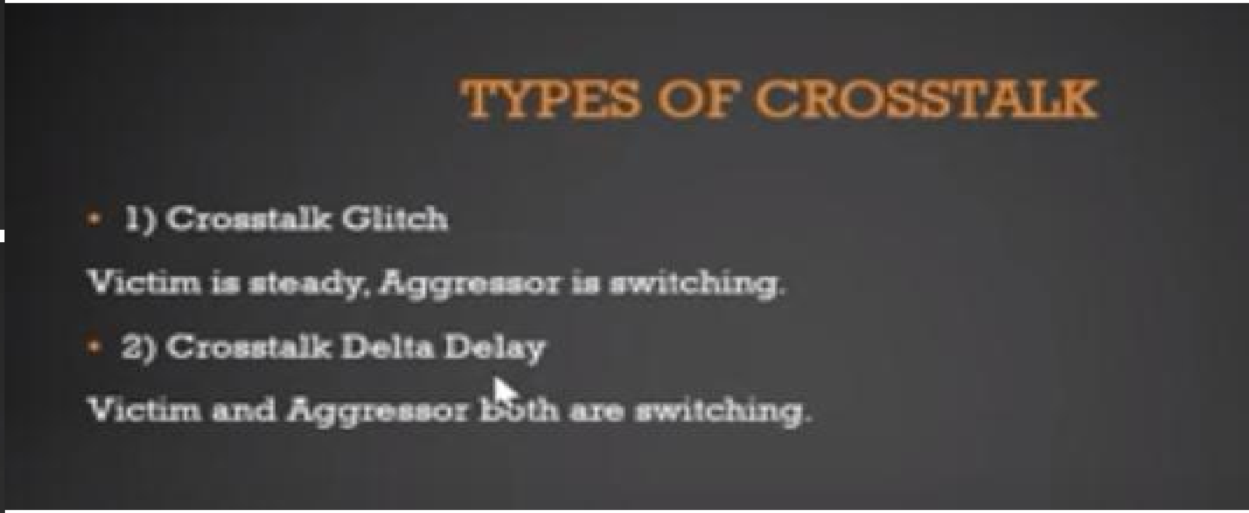


AOCV	POCV
Random variation modeled through the depth based derate and systematic variation is modeled through distance based derate	Random and Systematic variation modeled through a delay variation coefficient σ which is specific to each cell
Less accurate correlation between GBA and PBA	More accurate correlation between GBA and PBA
Transition variation and cell check variation not supported	Transition variation and cell check variation supported LVF format



WHAT IS CROSSTALK?

- The crosstalk noise refers to unintentional coupling of activity between two or more signals, which can either affect the functionality or the timing of the devices.



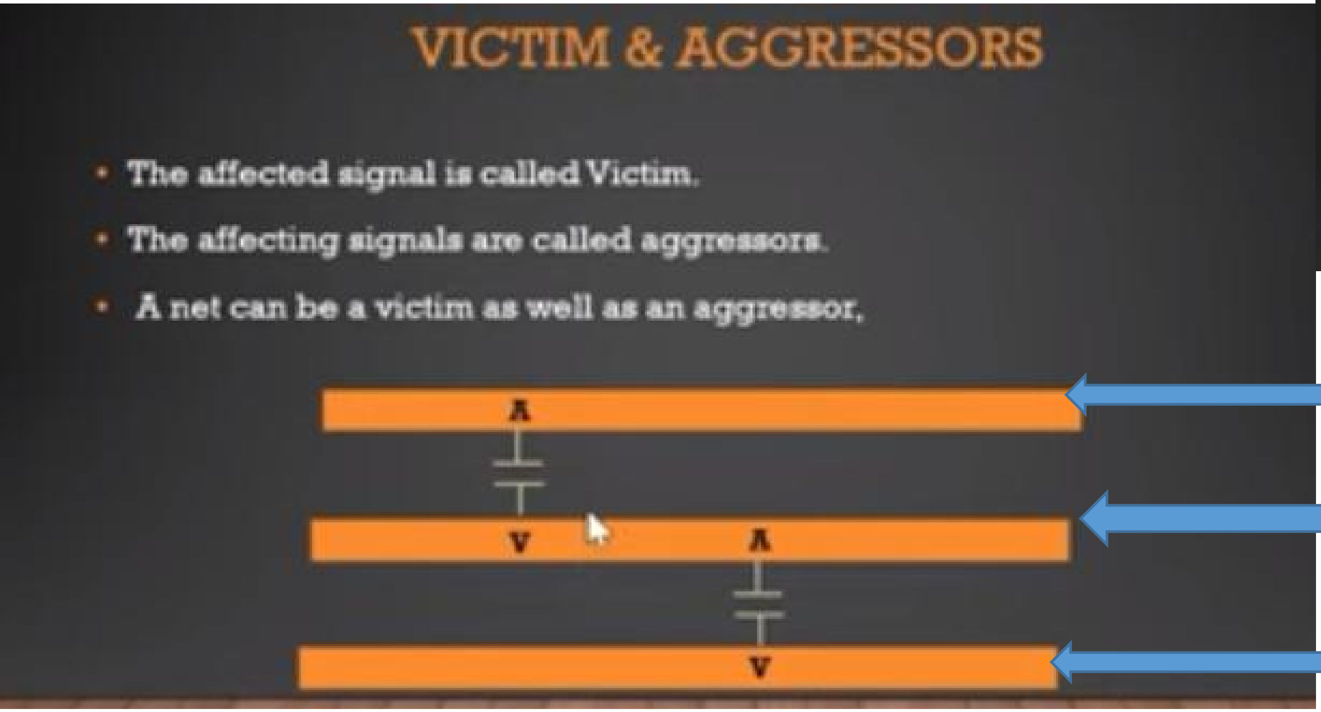
TYPES OF CROSSTALK

- 1) Crosstalk Glitch

Victim is steady, Aggressor is switching.

- 2) Crosstalk Delta Delay

Victim and Aggressor both are switching.



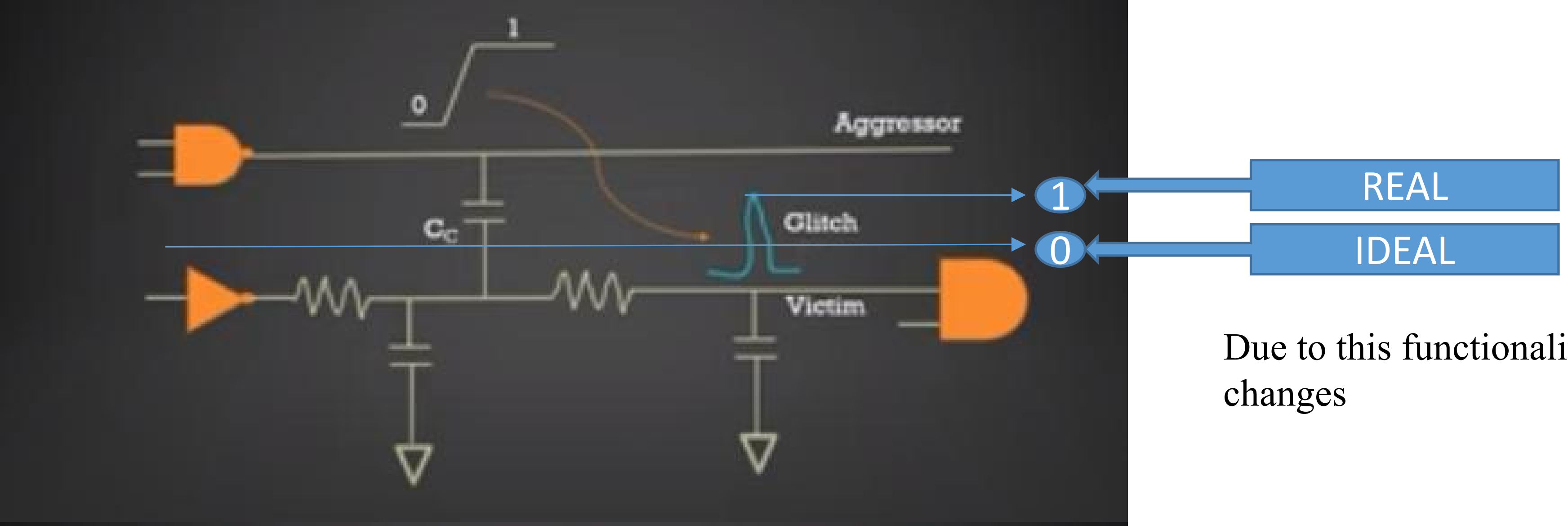
VICTIM & AGGRESSORS

- The affected signal is called Victim.
- The affecting signals are called aggressors.
- A net can be a victim as well as an aggressor,

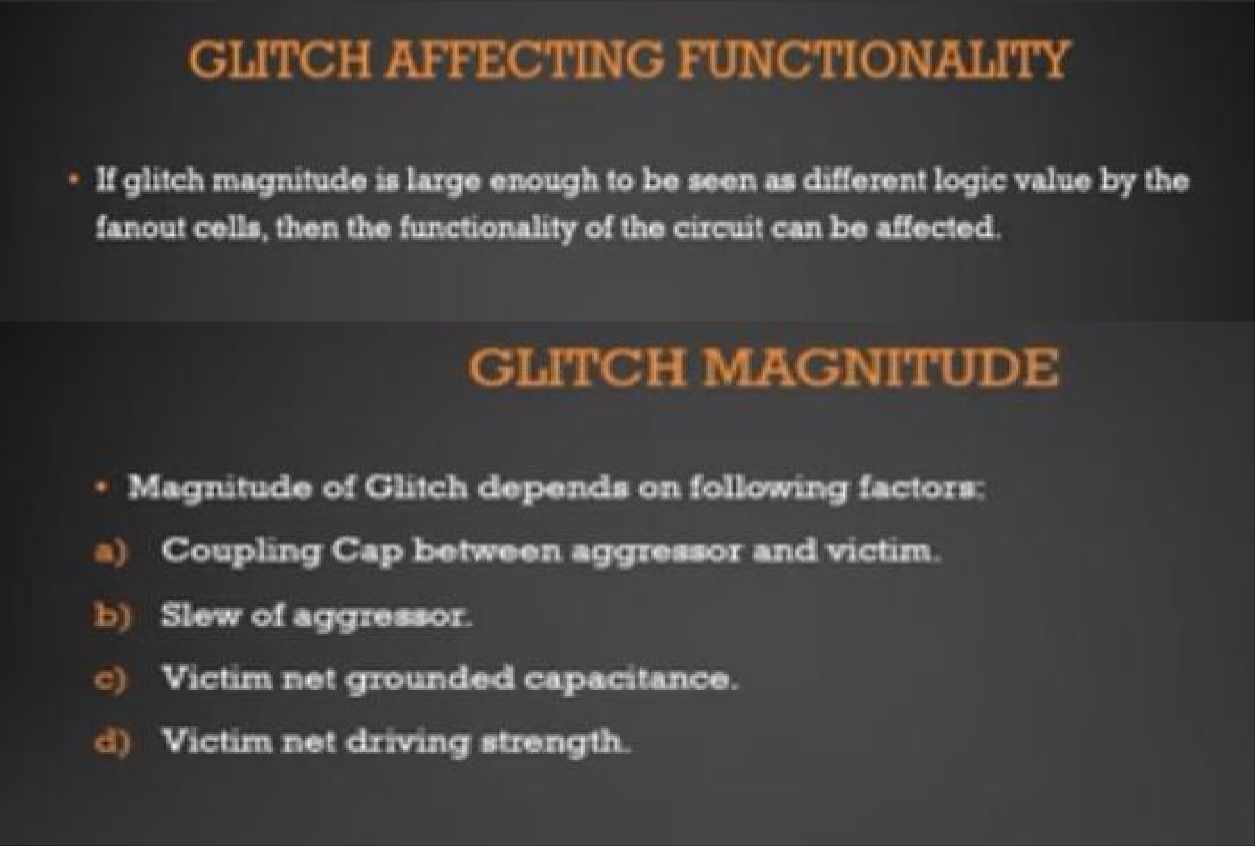
Switching strength of N1 is greater than N2 & N3

Switching strength of N2 is greater than N3 so its VN for N1 and AN for N3

Switching strength of N3 is smaller than N2 and N3



Due to this functionality of AND gate changes



GLITCH AFFECTING FUNCTIONALITY

- If glitch magnitude is large enough to be seen as different logic value by the fanout cells, then the functionality of the circuit can be affected.

GLITCH MAGNITUDE

- Magnitude of Glitch depends on following factors:

a) Coupling Cap between aggressor and victim.

b) Slew of aggressor.

c) Victim net grounded capacitance.

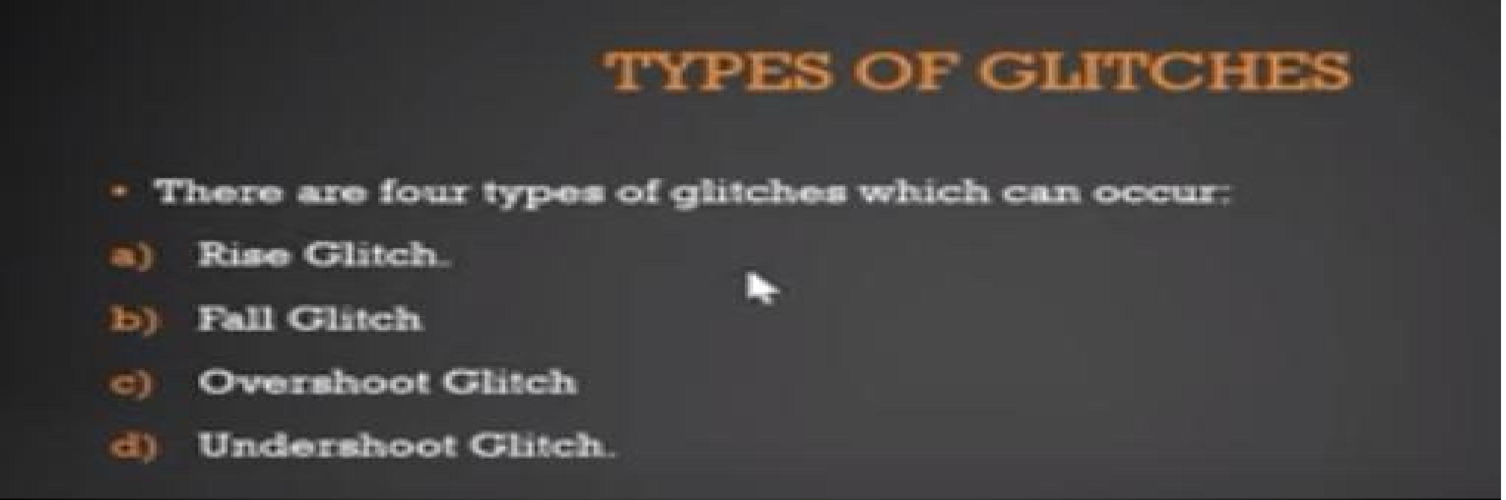
d) Victim net driving strength.

➢ glitch directly proportional to coupling capacitance.

➢ Glitch is indirectly proportional to slew(transition)

➢ Glitch is indirectly proportional to victim net ground capacitance(store 0 strongly)

➢ Glitch is indirectly proportional to driving strength



TYPES OF GLITCHES

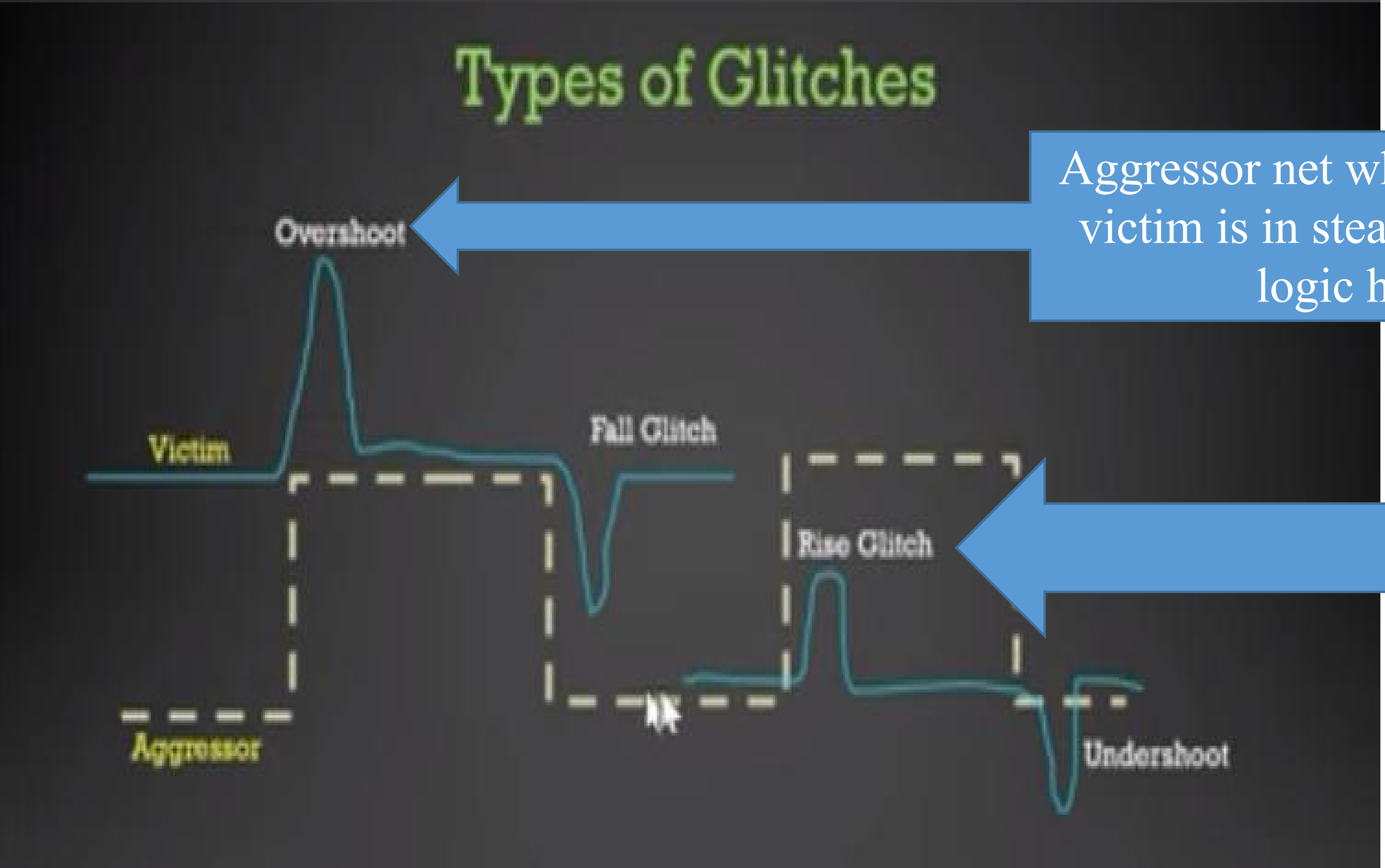
- There are four types of glitches which can occur:

a) Rise Glitch.

b) Fall Glitch

c) Overshoot Glitch

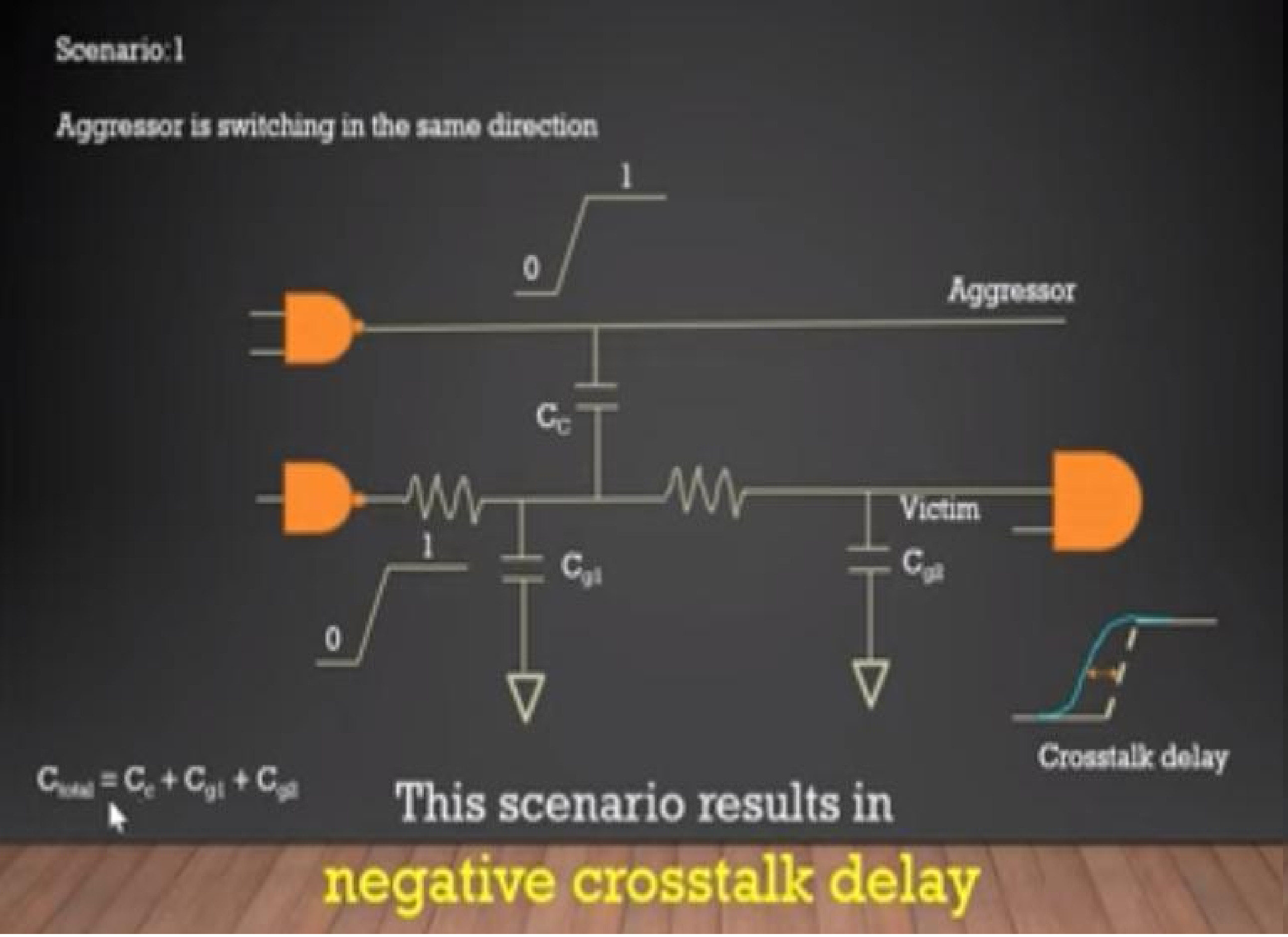
d) Undershoot Glitch.



Types of Glitches

Aggressor net which is switching from low to high and victim is in steady high, extra pulling on victim net to logic high in **<u>SAME DIRECTION</u>**

Aggressor net is switching from low to high and victim is at steady low but pulling victim net to logic high in **<u>OPPOSITE DIRECTION</u>** from steady low to high when victim is in steady low level vice versa



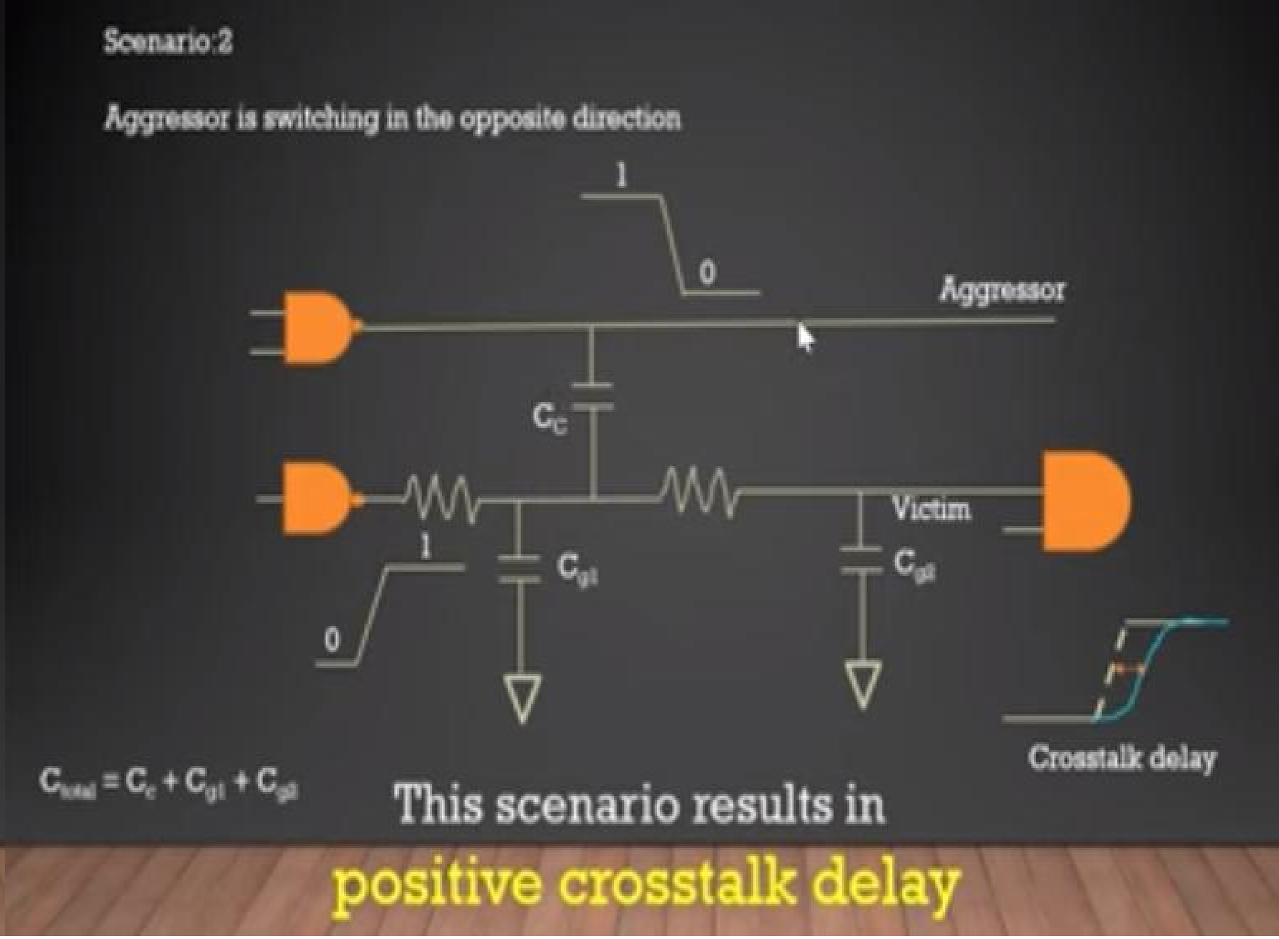
Scenario:1

Aggressor is switching in the same direction

$C_{total} = C_c + C_{g1} + C_{g2}$

This scenario results in

negative crosstalk delay



Scenario:2

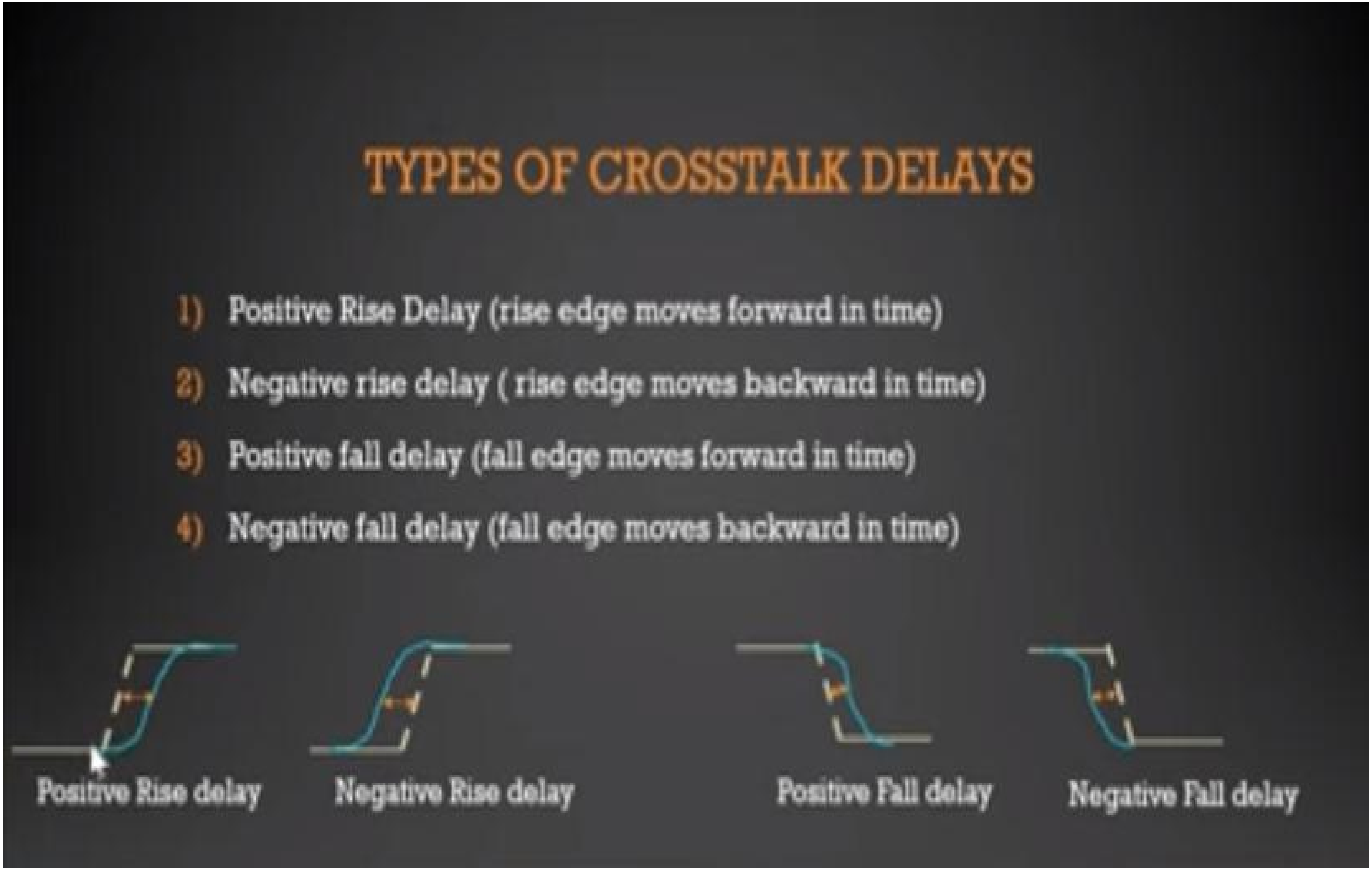
Aggressor is switching in the opposite direction

$C_{total} = C_c + C_{g1} + C_{g2}$

This scenario results in

positive crosstalk delay

Aggressor net (RIVER)
Victim net (BOAT)
When boat and river flow in same direction then it reaches fast
i.e. delay is less vice versa.



TYPES OF CROSSTALK DELAYS

1) Positive Rise Delay (rise edge moves forward in time)
2) Negative rise delay (rise edge moves backward in time)
3) Positive fall delay (fall edge moves forward in time)
4) Negative fall delay (fall edge moves backward in time)

Positive Rise delay

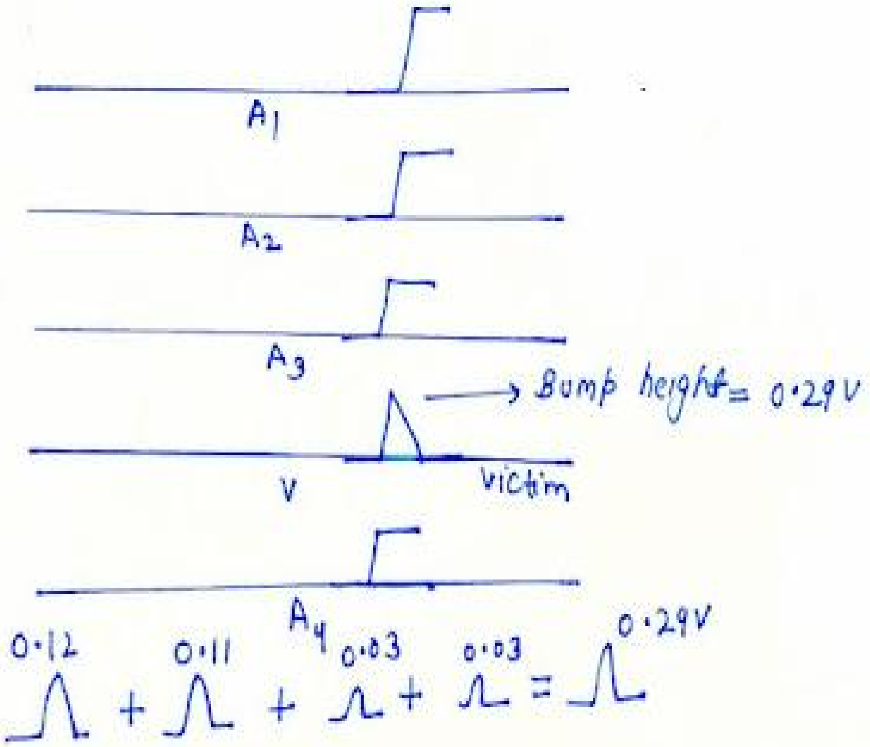
Negative Rise delay

Positive Fall delay

Negative Fall delay

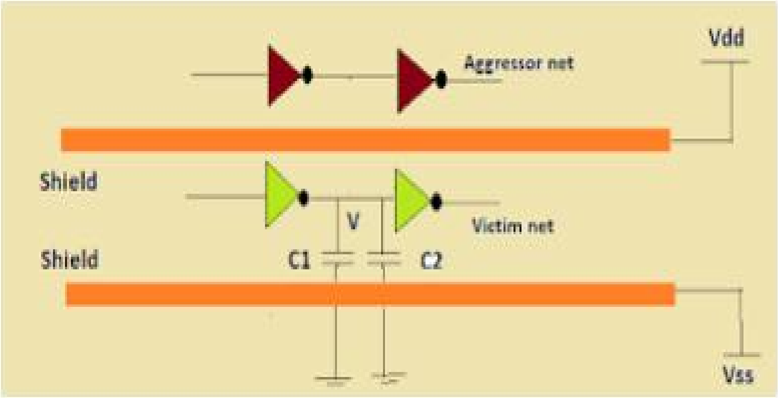
What happens when there are multiple aggressors

When multiple nets switch concurrently the crosstalk coupling noise effect on the victim is added due to multiple aggressors.



Noise Protection Techniques

1.Shielding: This method requires that shield wires are placed on either side of the critical signals. The shields are connected to power or ground rails.Shielding done only for critical nets.



Shielding

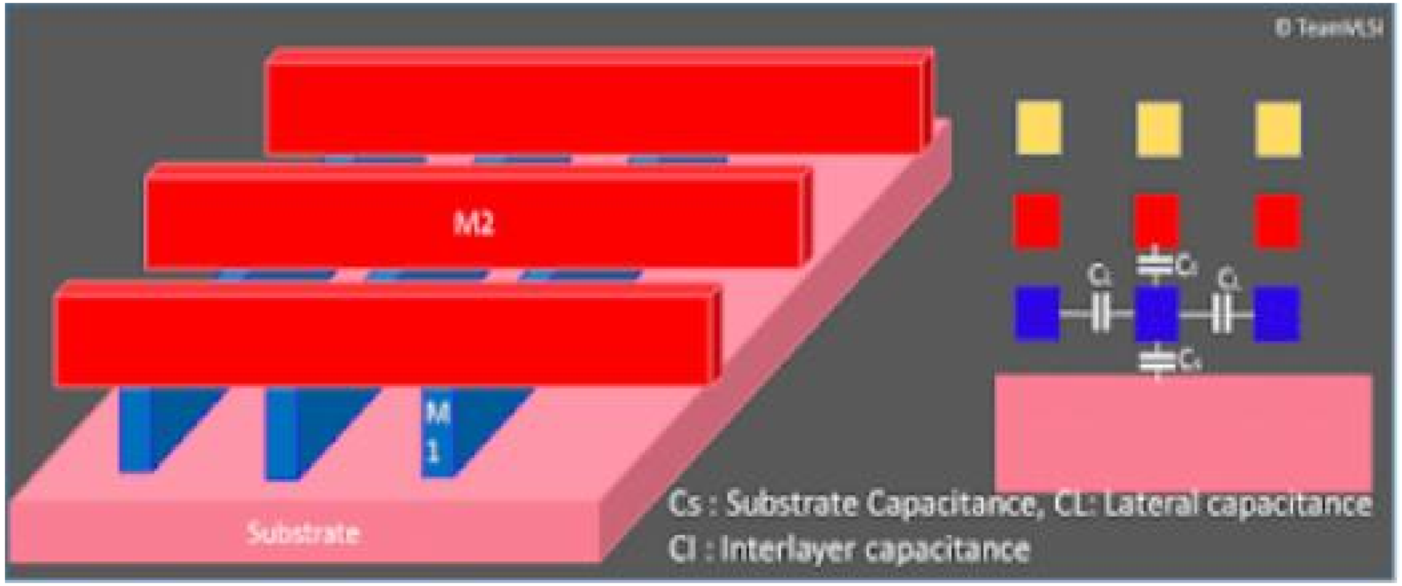
2.Wire spacing: This reduces the coupling to the neighboring nets.

3.Guard ring: A guard ring in the substrate helps in shielding the critical analog circuitry from digital noise.

4.Fast slew rate: A fast slew rate on the net implies that it is less susceptible to crosstalk and is inherently immune to crosstalk effects.

5.Increased the drive strength of victim net.

6.Decrease the drive strength of aggressor net.


© TeamVLSI
M2
M1
Substrate
Cl
Cl
CL
Cs
Cs : Substrate Capacitance, CL: Lateral capacitance
Cl : Interlayer capacitance

How To fix Crosstalk Issue

[1] Double spacing=>more spacing=>less capacitance=>less cross talk.

[2] Multiple vias=>less resistance=>less RC delay

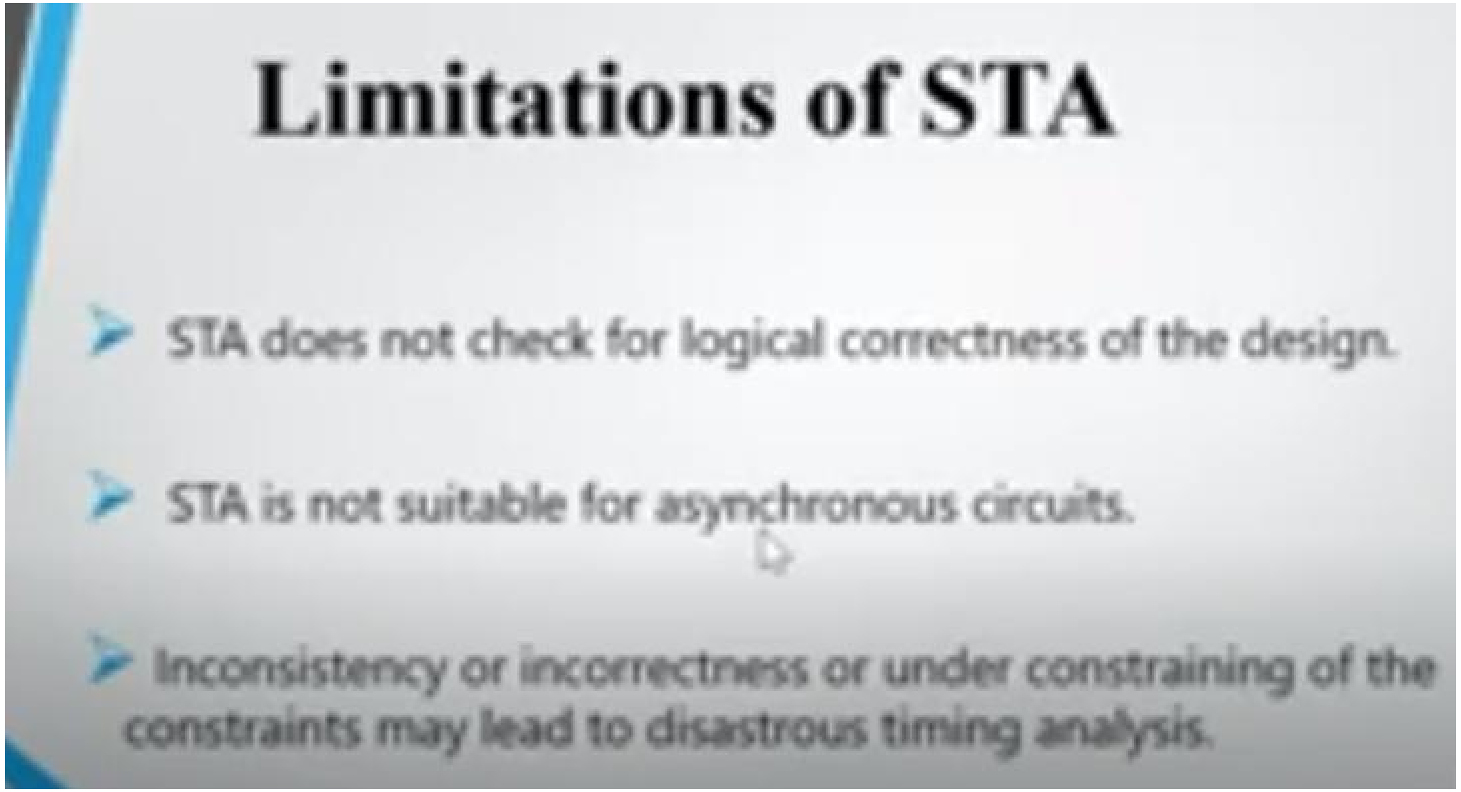
[3] Shielding=> constant cross coupling capacitance =>known value of crosstalk

[4] Buffer insertion=>boost the victim strength.

[5] upsizing, swaping (lvt to hvt viceversa).

[6] re routing.

[7] Maintain proper NDA rules.



Limitations of STA

- STA does not check for logical correctness of the design.
- STA is not suitable for asynchronous circuits.
- Inconsistency or incorrectness or under constraining of the constraints may lead to disastrous timing analysis.

CHECK_TIMING

Shows possible timing problems for design.

-verbose Shows detailed information about potential problems.

clock_crossing

Checks clock interactions when there are multiple clock domains. If a clock launches one or more paths, which are captured by other clocks, it will have an entry in clock crossing report. If all paths between two clocks are false paths or they are exclusive/asynchronous clocks, the path is marked by *. If only part of paths are set as false paths or exclusive/asynchronous clocks, the path is marked by #.

generated_clocks

Checks generated clock network. The master must be driven by a clock source. There cannot be loops of generated clocks. For example, the source of generated clock CLK1 cannot be used to generate clock CLK2 if CLK2 also is used to generate CLK1.

generic

Warns about generic (unmapped) cells in the design. The timing of paths through generic cells is inaccurate because generic cells have zero delay.

ideal_clocks

Shows the clocks that are not defined as propagated using the /set_propagated_clock. Generally, all clocks should be propagated so that the clock network timing is accurately calculated. Especially, in presence of crosstalk, the delay changes induced by other nets on the clock network are not reflected in the calculated slacks in the design.

latch_fanout

Checks fanout of level-sensitive latches. A warning is issued if a level-sensitive latch fans out to itself. An information message also appears for a latch that fans out to a latch of the same clock .

latency_override

Warns of clock latency specification conflicts. If clock source latency is defined for both a clock and its port (source pin), the source latency for clock object is ignored. If input_delay is set on clock port, which also has source latency specified, the input_delay is ignored as a source latency. Also warns if more than one clock latency fan out to any latch clock pin.

ms_separation

Checks minimum separation of master and slave clock pulses on master/slave latches.

multiple_clock

Warns if multiple clocks reach a register clock pin. If more than one clock signal reaches a register clock pin, and timing_enable_multiple_clocks_per_reg is set to FALSE, then it is undefined which one is used for analysis. In this case, use set_case_analysis so only one clock can propagate from its sources to the register clock pin. Using this check and the no_clock check run significantly faster than other checks. Hence, to save time, user may want to issue these checks separately from other checks.

no_clock

Warns if no clock reaches a register clock pin. In this case, no setup or hold checks are performed on data pins related to that clock pin, and the path starting at the clock pin is not relative to a clock. For performance of this check, see the description in the multiple_clock check.

no_driving_cell

Warns if a port does not have any driving cell. This warning is issued only when the net connected to the port has parasitics. In such case, the accuracy of delay calculation could be impacted, as a default strong driver is assumed in absence of driving cell definition. Especially, in presence of crosstalk, a port with no driving cell could act as a strong aggressor which could lead to significant amount of pessimism in the analysis. Also, a port with no driving cell could act as a string victim, which could underestimate the crosstalk effect.

no_input_delay

Warns if no clock related delay specified on an input port, where it propagates to a clocked latch or output port. With -verbose, the port name will be listed. Note that with timing_input_port_default_clock set to 'true', a default clock will be assumed for the input port. Otherwise it will not be clocked, and the paths are unconstrained. In this case, if there is no input delay specified, check_timing will not generate warnings.

partial_input_delay

Warns if any port has partially defined input delay. This happens when set_input_delay -min is applied on a port to set the min input delay with respect to a clock, however no set_input_delay -max is applied to that port to specify the max delay, or vice versa. As a result, some paths starting from the port with partially defined input delay may become unconstrained and some potential violations could be missed.

pll_configuration

Warns if a problem is detected in the configuration of any phase locked loop (PLL) cells. For a PLL to be correctly configured, the PLL output clock should reach the PLL feedback pin. If a PLL is not correctly configured, it will not show any phase adjustment on the PLL output clock.

signal_level

Checks that the driver signal level matches the load signal level. The signal levels are determined from the cell specific operating conditions and rail voltages (or UPF), and from the following library attributes: input_signal_level, output_signal_level, input_voltage, output_voltage. The check is performed on all input pins that have input_voltage defined in the timing library. If the driver pin does not have output_voltage defined in the library then the voltage of the rail powering the driver pin is used as the output signal level.

supply_net_voltage

Checks that each segment of UPF supply nets has voltage assigned to it by set_voltage command.

unconnected_pins

Checks that each power and ground pin is connected to a UPF supply net. The connection can be implicit (e.g., power domain) or explicit (for example, connect_supply_net).

unconstrained_endpoints

Warns about unconstrained timing endpoints. This warning identifies timing endpoints (output ports and register data pins) that are not constrained for maximum delay (setup) checks. If the endpoint is a register data pin, it can be constrained by using create_clock for the appropriate clock source. You can constrain output ports using the set_output_delay or set_max_delay commands.

unexpandable_clocks

Warns if there are sets of clocks for which periods are not expandable with respect to each other. The checking is only done for the related clock domains, such as ones where there is at least one path from one clock domain to the other. This could be because of an incorrectly defined clock period for one or more of the clocks. Another possibility is when asynchronous clocks with unexpandable periods are interacting where they should have been defined in different clock domains.

pulse_clock_non_pulse_clock_merge

Warns if pulse clock and normal clock propagate to the same network.

pulse_clock_no_pulse_generator

Warns if the pulse clock constraints cannot be honored. This could be because pulse clock constraints have been set on clocks that do not drive pulse generators or the design may not have any pulse generators.

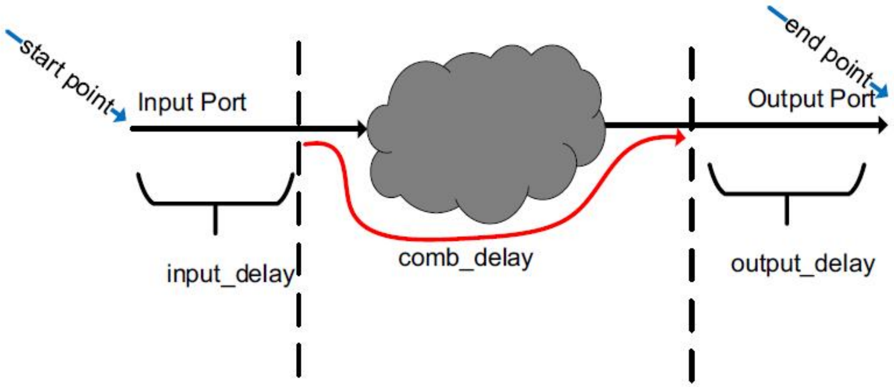
INPUT AND OUTPUT DELAYS

Between input and output we have to set margins as input 60% and output 40% with respect to clk frequency because the design is hierarchy(different blocks)

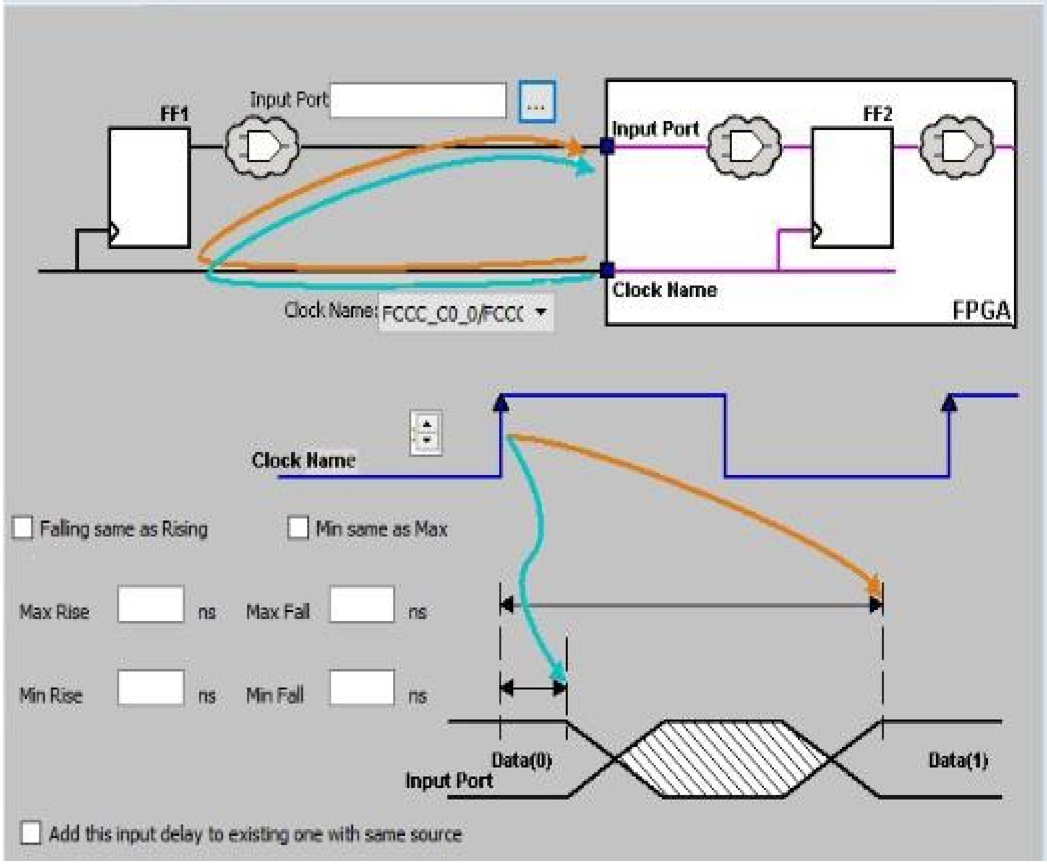
Input delay-The time at which the data arrives at the input pin of the block from external ckt with respect to reference ckt.

Output delay-The minimum time required to obtain a valid output at an output pin.

```
set_input_delay  -clock name min/max [get_ports x]
set_output_delay  -clock name min/max [get_ports x]
```



Input and output delays



Input Port: ...

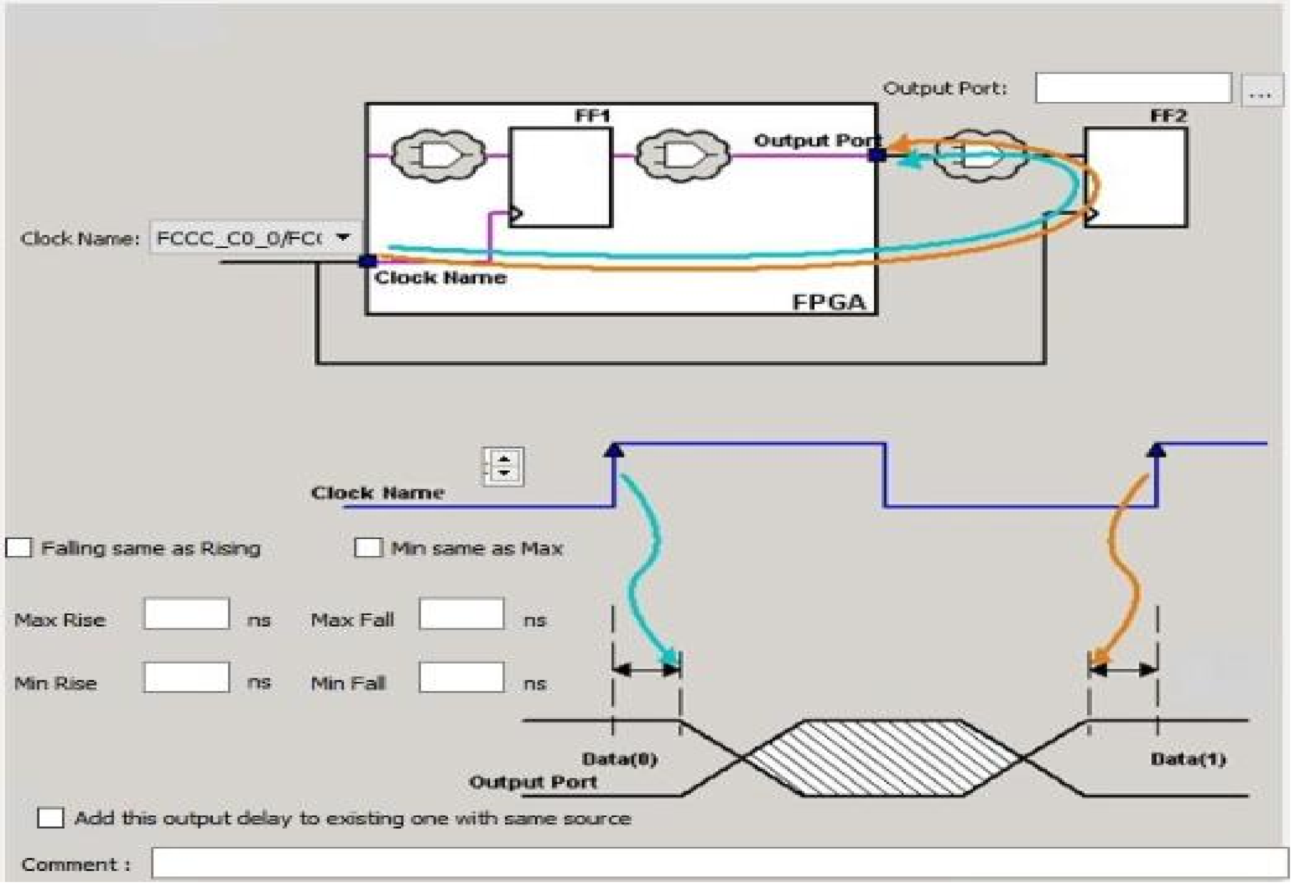
Clock Name: FCCC_C0_0/FCCC

□ Falling same as Rising □ Min same as Max

Max Rise ns Max Fall ns

Min Rise ns Min Fall ns

□ Add this input delay to existing one with same source



Output Port: ...

Clock Name: FCCC_C0_0/FCC

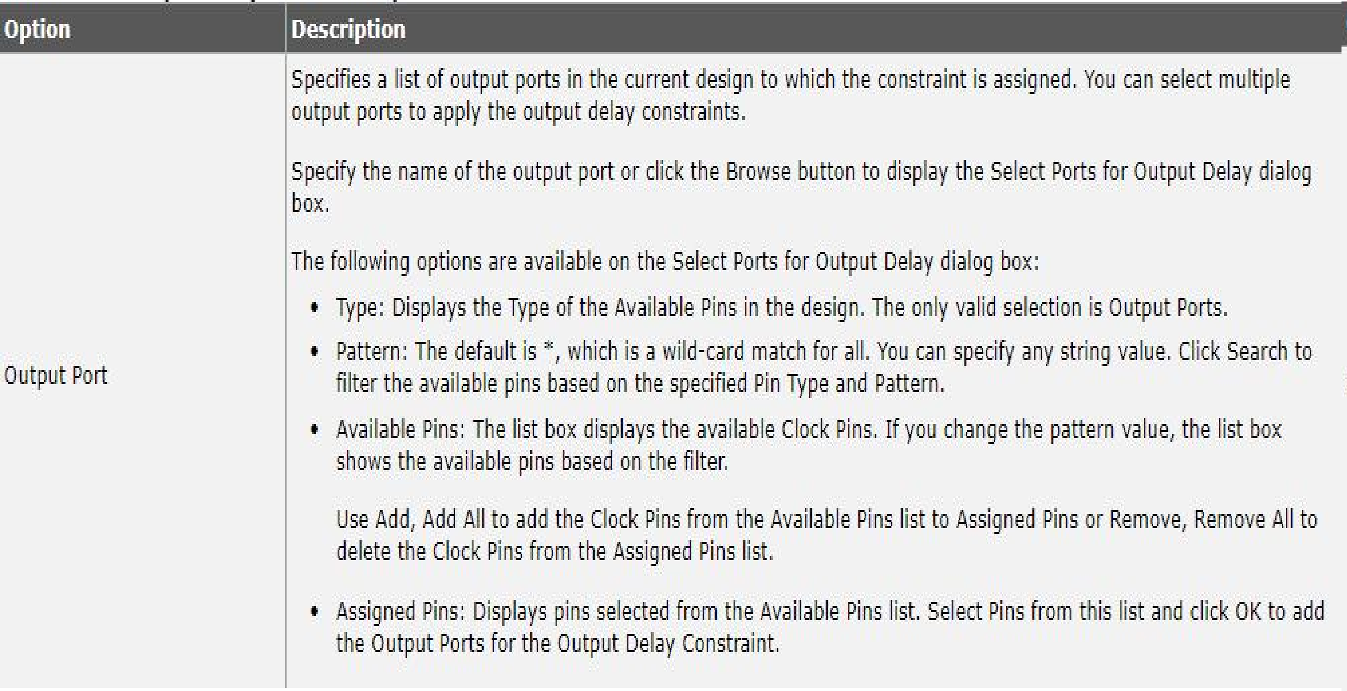
□ Falling same as Rising □ Min same as Max

Max Rise ns Max Fall ns

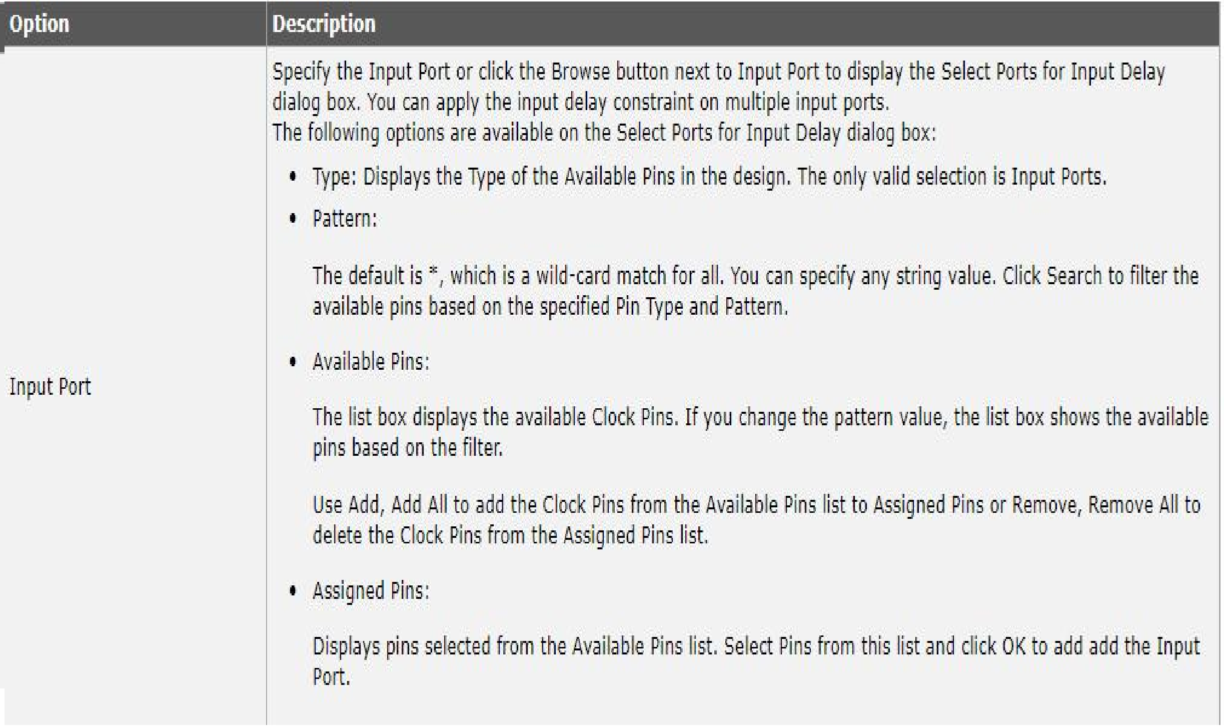
Min Rise ns Min Fall ns

□ Add this output delay to existing one with same source

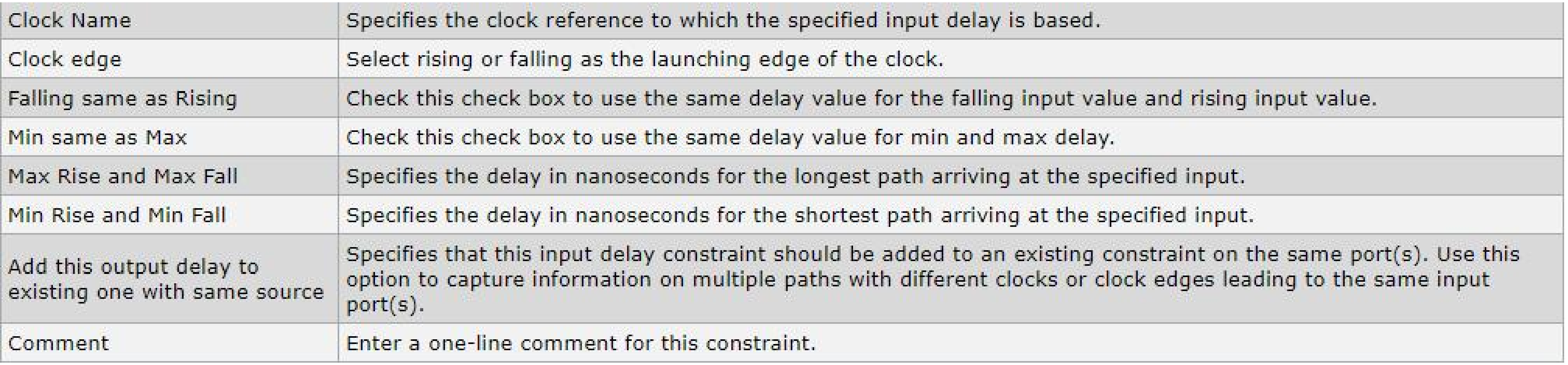
Comment :



Option	Description
Output Port	Specifies a list of output ports in the current design to which the constraint is assigned. You can select multiple output ports to apply the output delay constraints. Specify the name of the output port or click the Browse button to display the Select Ports for Output Delay dialog box. The following options are available on the Select Ports for Output Delay dialog box: • Type: Displays the Type of the Available Pins in the design. The only valid selection is Output Ports. • Pattern: The default is *, which is a wild-card match for all. You can specify any string value. Click Search to filter the available pins based on the specified Pin Type and Pattern. • Available Pins: The list box displays the available Clock Pins. If you change the pattern value, the list box shows the available pins based on the filter. Use Add, Add All to add the Clock Pins from the Available Pins list to Assigned Pins or Remove, Remove All to delete the Clock Pins from the Assigned Pins list. • Assigned Pins: Displays pins selected from the Available Pins list. Select Pins from this list and click OK to add the Output Ports for the Output Delay Constraint.



Option	Description
Input Port	Specify the Input Port or click the Browse button next to Input Port to display the Select Ports for Input Delay dialog box. You can apply the input delay constraint on multiple input ports. The following options are available on the Select Ports for Input Delay dialog box: • Type: Displays the Type of the Available Pins in the design. The only valid selection is Input Ports. • Pattern: The default is *, which is a wild-card match for all. You can specify any string value. Click Search to filter the available pins based on the specified Pin Type and Pattern. • Available Pins: The list box displays the available Clock Pins. If you change the pattern value, the list box shows the available pins based on the filter. Use Add, Add All to add the Clock Pins from the Available Pins list to Assigned Pins or Remove, Remove All to delete the Clock Pins from the Assigned Pins list. • Assigned Pins: Displays pins selected from the Available Pins list. Select Pins from this list and click OK to add add the Input Port.



Clock Name	Specifies the clock reference to which the specified input delay is based.
Clock edge	Select rising or falling as the launching edge of the clock.
Falling same as Rising	Check this check box to use the same delay value for the falling input value and rising input value.
Min same as Max	Check this check box to use the same delay value for min and max delay.
Max Rise and Max Fall	Specifies the delay in nanoseconds for the longest path arriving at the specified input.
Min Rise and Min Fall	Specifies the delay in nanoseconds for the shortest path arriving at the specified input.
Add this output delay to existing one with same source	Specifies that this input delay constraint should be added to an existing constraint on the same port(s). Use this option to capture information on multiple paths with different clocks or clock edges leading to the same input port(s).
Comment	Enter a one-line comment for this constraint.

- Consider your chip is going to be placed in a board and input comes from pre block(assume a chip) and your output goes to other chip.
- Then if u operate all this three chips as same clock. Then from the previous chip it takes time to reach your chip considering delay of i/o delays of previous block.
- If you don’t give input delay then at rising clock edge your chip then due to this external delay data will arrive late this leads to fault logic.

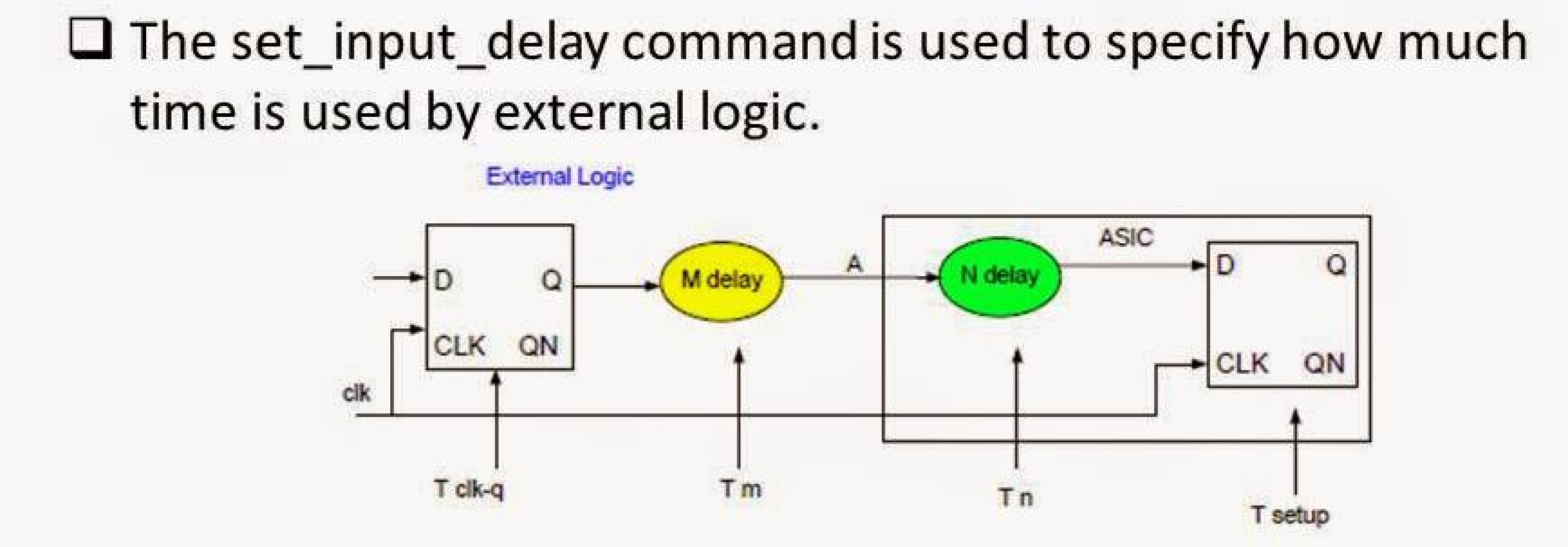
Example=If you don’t set the timing for class to attend before then everyone will attend the class as per their interest and its useless

- If u give delay then you chip makes some delay within it such that the data reaches the input register(not input pin) at next rising edge and you logic works.

- Similarly to output pin also so that next module prepare themselves..

Input delay --> Sets input delay on pins or input ports relative to a clock signal.
means time given to outer world to reach our own design.
Example =time we need to fix before starting the session so that everyone (outsiders from different location will attend on that fix time without miss)

Output delay --> Sets output delay on pins or output ports relative to a clock signal means **time taken by my design**.
Example=(fixing the duration of the class to be taken considering every students requirement and issues)

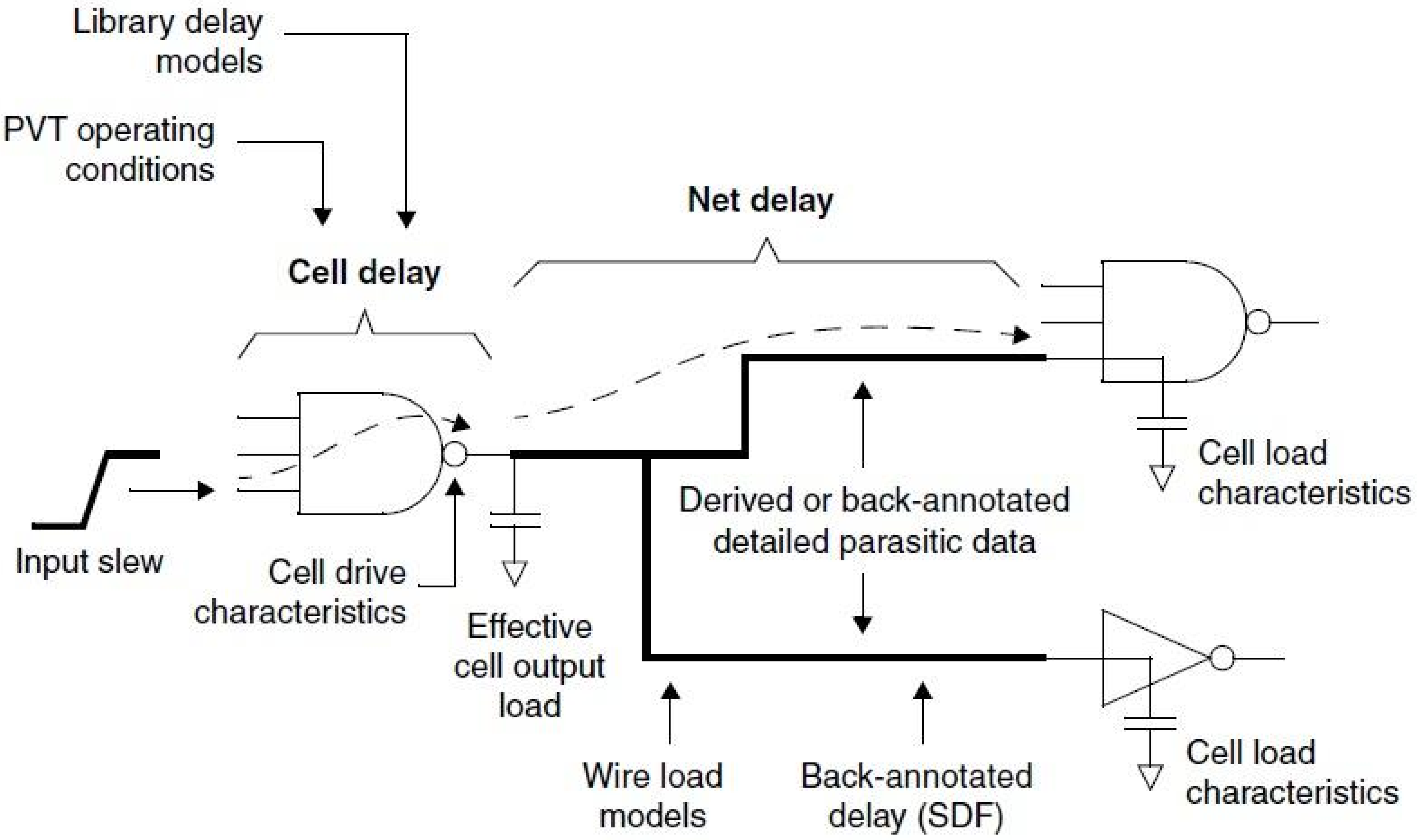


❑ The set_input_delay command is used to specify how much time is used by external logic.

CELL DELAY AND NET DELAY

All cells have their own delay information in .lib file.
Delay depends on three factors

1. input transition of cell (how signal is rising and falling)
2. Output capacitance (wire/net capacity + input capacitance)
3. Operating conditions(PVT)


Library delay models
PVT operating conditions
Net delay
Cell delay
Input slew
Cell drive characteristics
Effective cell output load
Derived or back-annotated detailed parasitic data
Cell load characteristics
Wire load models
Back-annotated delay (SDF)
Cell load characteristics

- i/p transition, o/p capacitance are directly proportional to cell delay.

PVT condition

- If p=slow delay is more
- If P = fast delay is less.
- Voltage is indirectly proportional to delay
- Temperature is directly proportional to delay (higher technology)
- Final delay = 6ns+ (P + V + T values all defined in .lib)

	1	2	3	4	5	6	7
0.1							
0.2			6ns				
0.3							
0.4							
0.5							

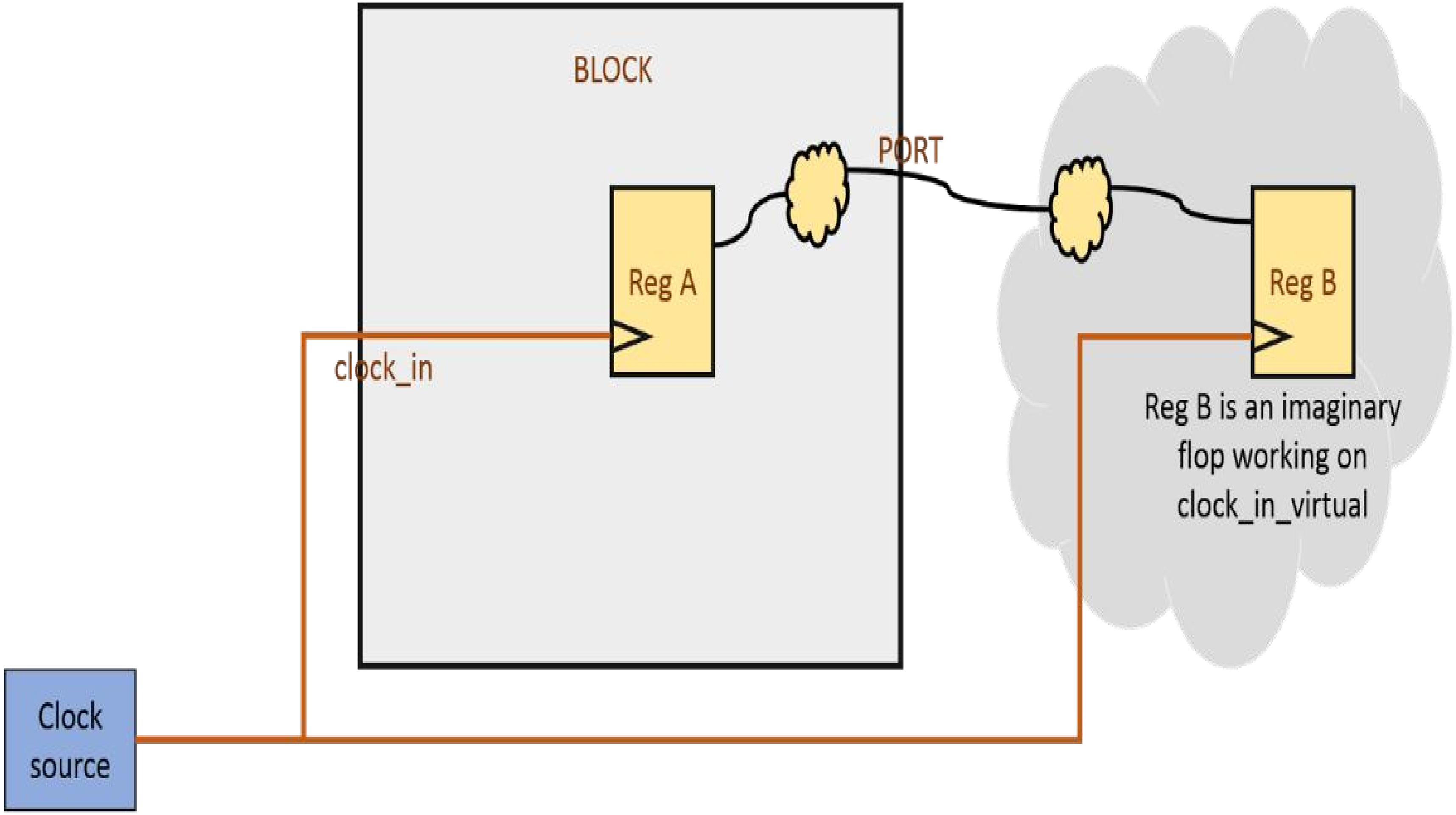
Input transition

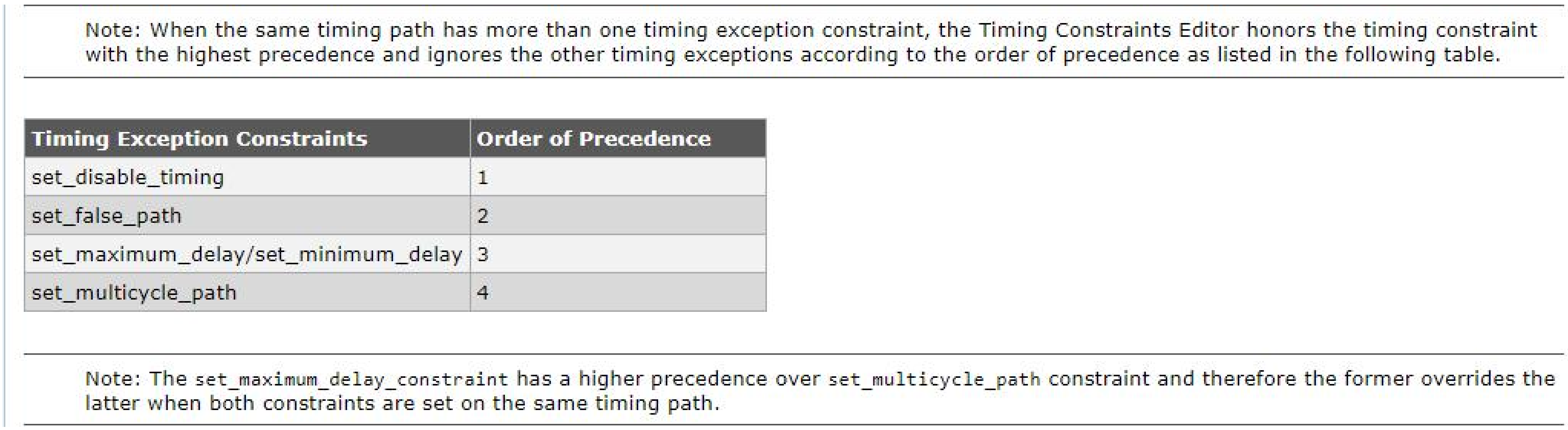
O/p capacitance

out of cell delay and net delay which is more effective as we move from higher technology to lower technology?

Virtual clock

Suppose we have no clocks in in the current design which is purely combinational logic at this time we use a dummy clock to set input and output delay/slave clock.

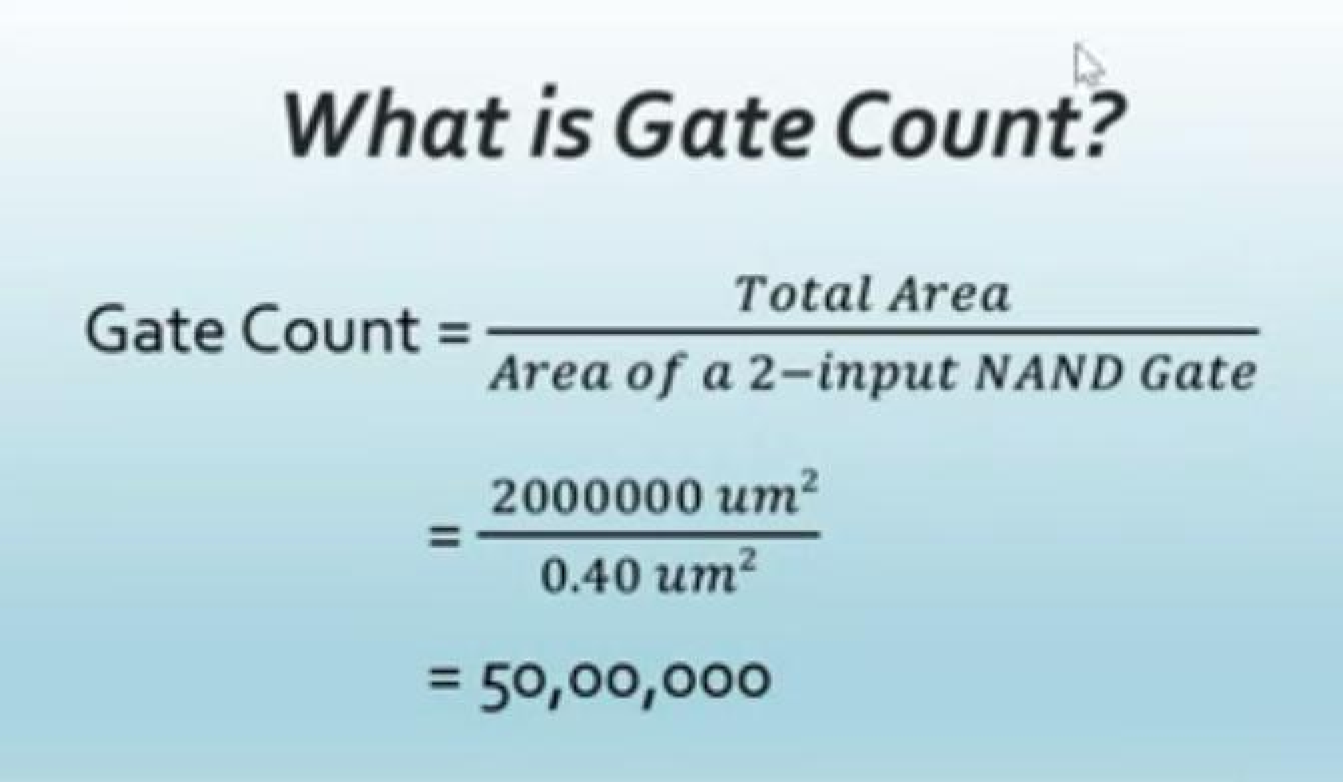




Note: When the same timing path has more than one timing exception constraint, the Timing Constraints Editor honors the timing constraint with the highest precedence and ignores the other timing exceptions according to the order of precedence as listed in the following table.

Timing Exception Constraints	Order of Precedence
set_disable_timing	1
set_false_path	2
set_maximum_delay/set_minimum_delay	3
set_multicycle_path	4

Note: The `set_maximum_delay_constraint` has a higher precedence over `set_multicycle_path` constraint and therefore the former overrides the latter when both constraints are set on the same timing path.

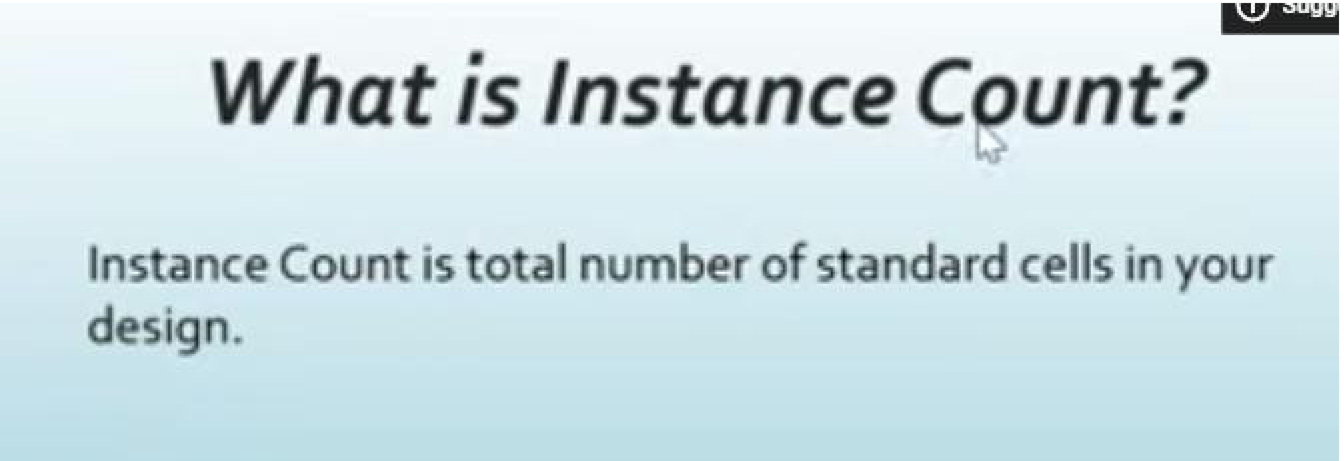


What is Gate Count?

$$\text{Gate Count} = \frac{\mathit{Total\ Area}}{\mathit{Area\ of\ a\ 2{-}input\ NAND\ Gate}}$$

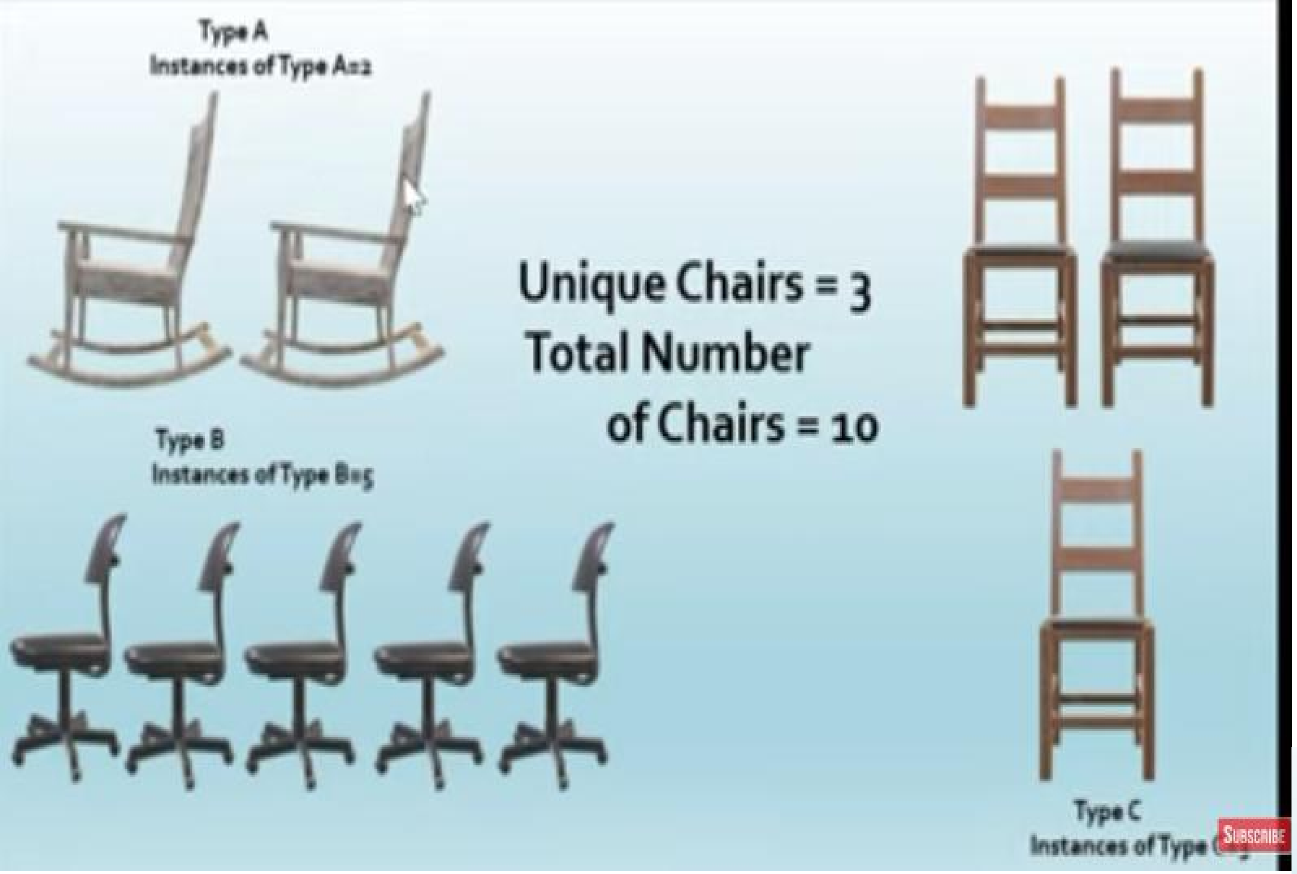
$$= \frac{2000000\ um^2}{0.40\ um^2}$$

$$= 50,00,000$$



What is Instance Count?

Instance Count is total number of standard cells in your design.

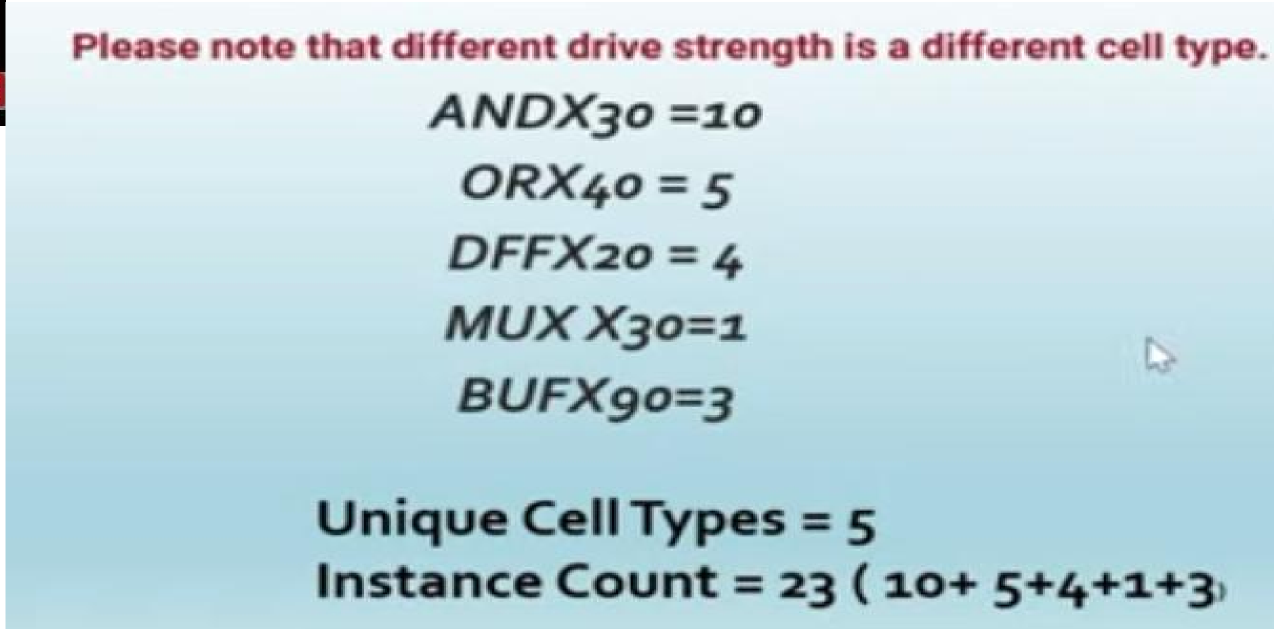


Type A
Instances of Type A=2

Unique Chairs = 3
Total Number
of Chairs = 10

Type B
Instances of Type B=5

Type C
Instances of Type (



Please note that different drive strength is a different cell type.

ANDX30 =10

ORX40 = 5

DFFX20 = 4

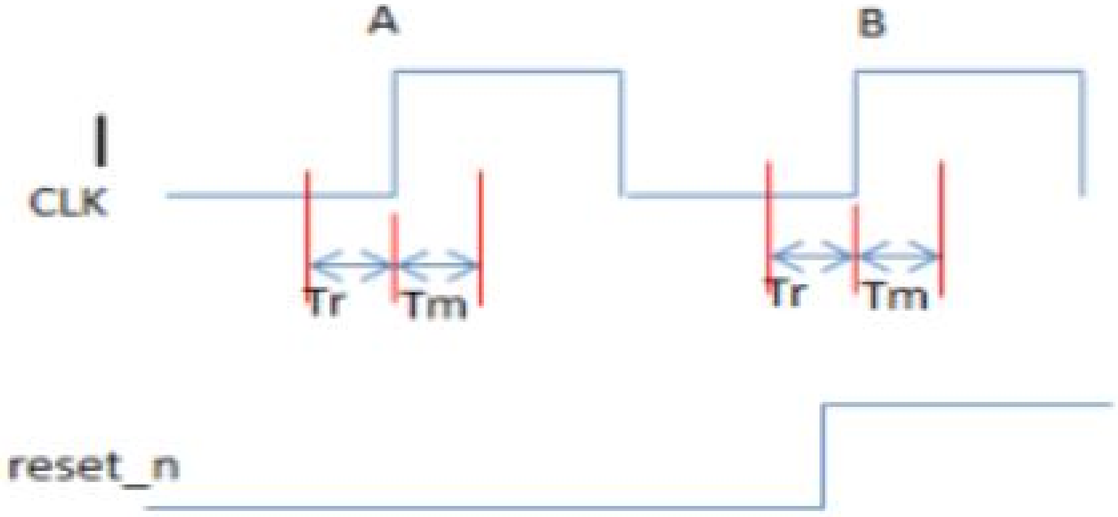
MUX X30=1

BUFX90=3

Unique Cell Types = 5
Instance Count = 23 (10+ 5+4+1+3)

Recovery and Removal Checks

- Recovery and removal analysis are done on asynchronous signals like resets.
- These specify the requirement of these signals with respect to clock.
- *Recovery Time* is the minimum required time to the next active clock edge the after the reset is released.
- *Removal Time* is the minimum required time after the clock edge after which reset can be released.
- If we don’t use Reset for asynchronous it creates functional/metastability(quasi state) issues.



Difference between 14nm and 7nm

As we move from higher to lower nodes due to shrinking technology different issues evolves and existing issues get stronger.

- Crosstalk increases
- More DRV
- More IR drop
- Congestions issue because of reducing area
- More power dissipation (waste of heat in form of heat)

References

- Recorded session of Anji sir

YouTube channels.

- Back to basics
- Digital shri
- VLSI team

websites

- VLSI physical design
- Backend VLSI adventure
- Team VLSI
- Micro-ip.com