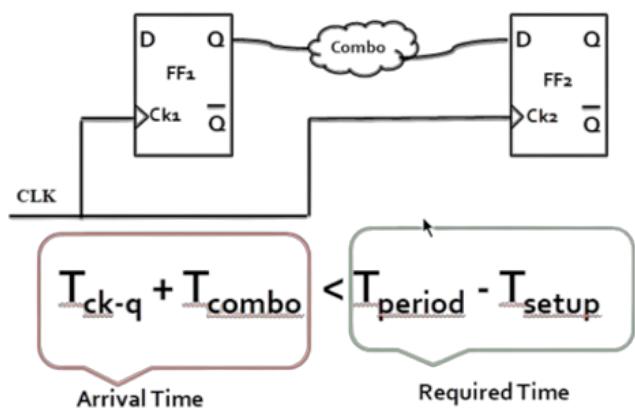
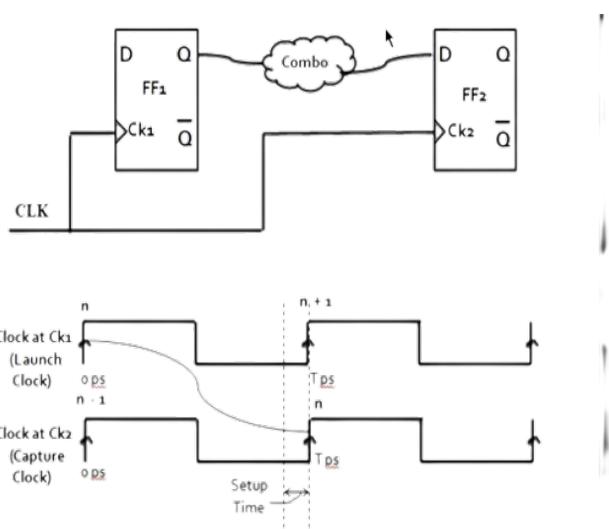
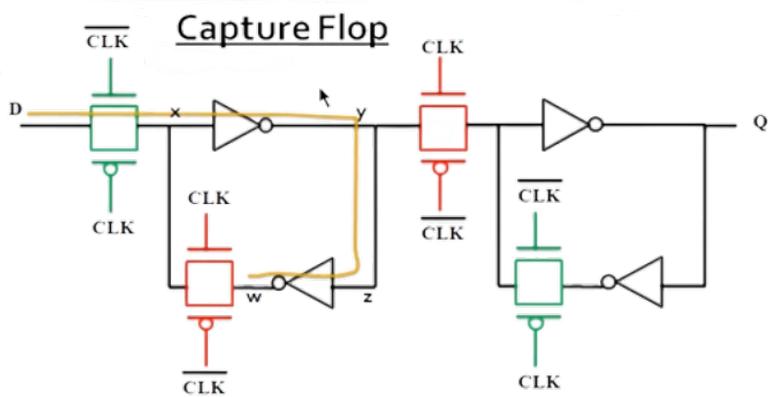


STATIC TIMING ANALYSIS →

Setup time →

It is the minimum amount of time for which data should be stable at the input before the active edge of the clock.

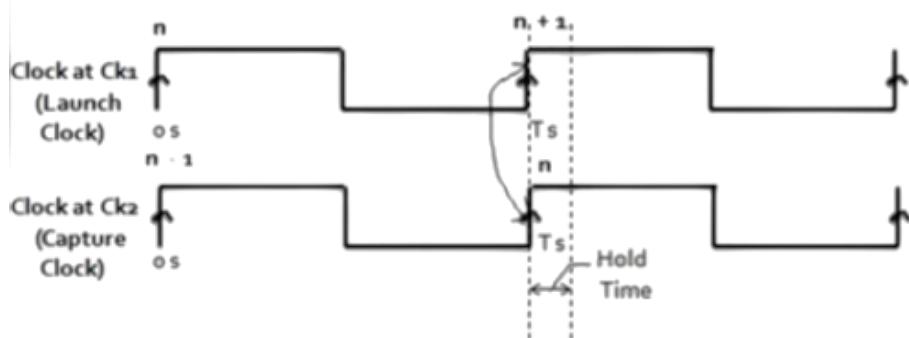
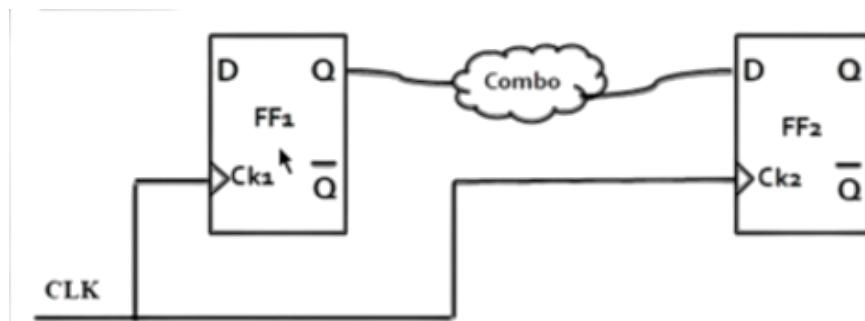
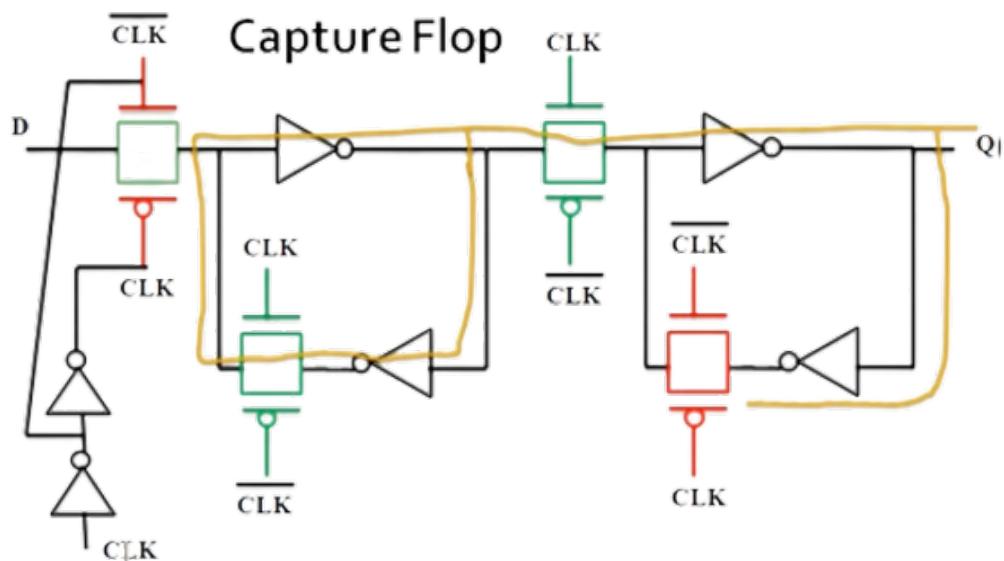
Setup Time of a Flip Flop



Required Time > Arrival Time; Slack = Required Time – Arrival Time
Slack = +ve => No Set Up Violation ✓
Slack = -ve => Set Up Violation ✗

Hold Time →

Hold time is the minimum time for which the data should be stable at the input at the active edge of the clock has arrived.



Latency →

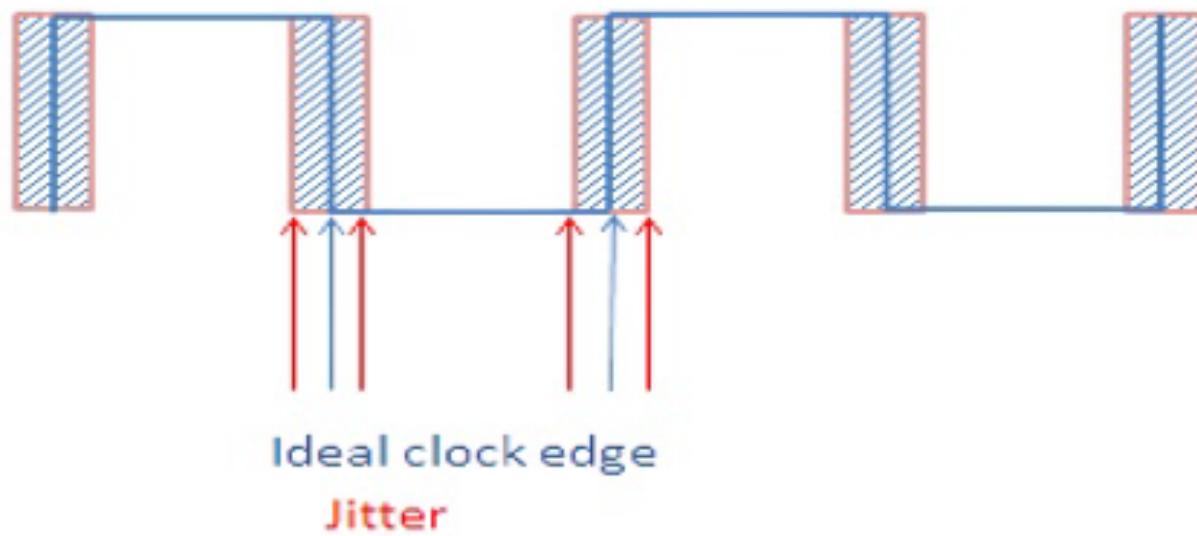
latency is defined as the amount of time taken by the clock signal in travelling from its source to the sinks.

Insertion Delay →

Time taken by the clock to reach the farthest flop is known as max Insertion Delay.

Jitter →

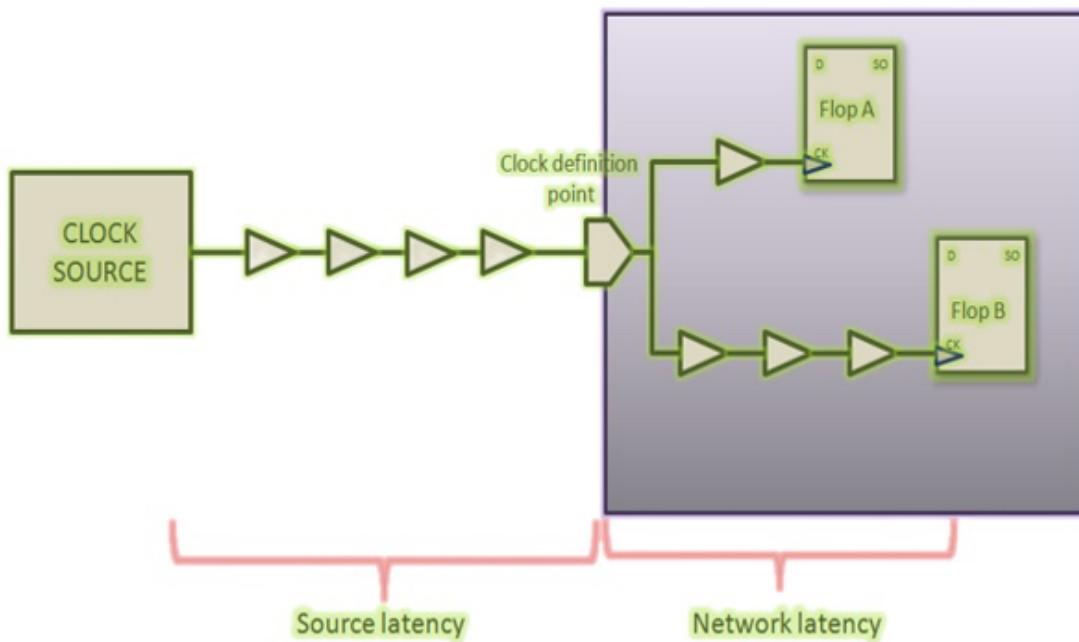
Clock jitter is the deviation of a clock edge from its ideal position in time



Types of latency →

Source latency → Source latency is defined as the time taken by the clock signal in transversing from clock source.

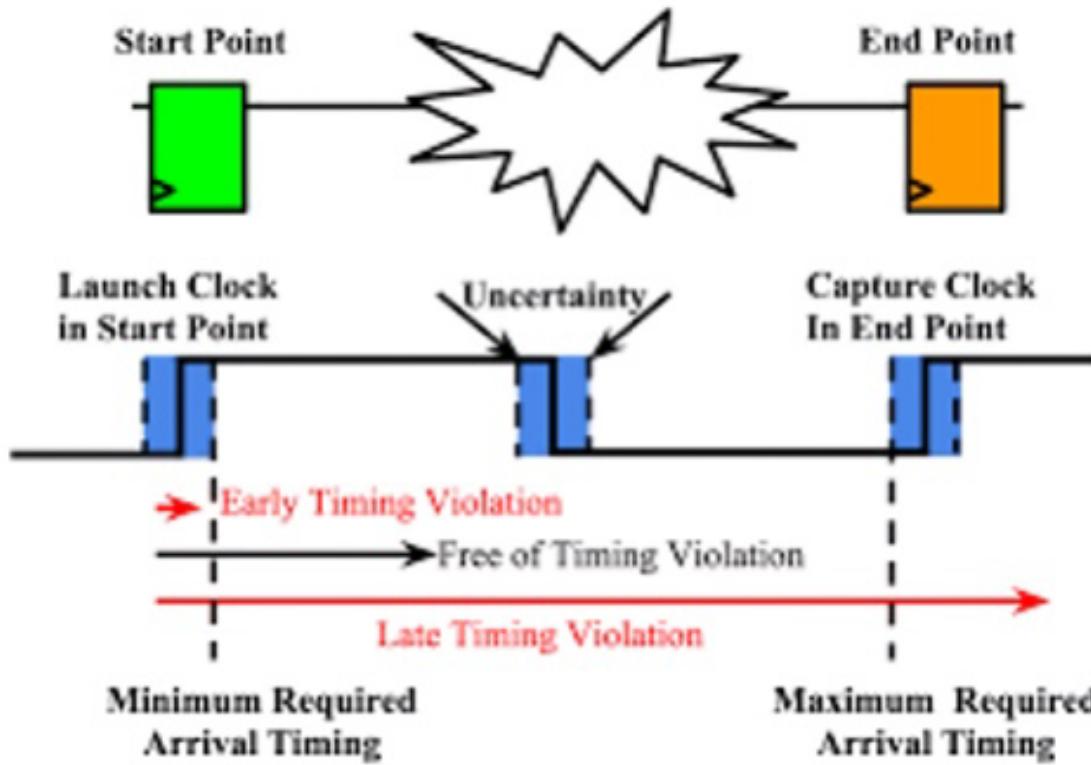
Network latency → Network latency is defined as the time taken by the clock signal in travelling from clock definition point to the sink of the clock.



Skew →

positive skew → If the capture clock comes late than the launch clock then it is called +ve skew.

Negative Skew → If the capture clock comes early than launch clock it is called -ve skew.



Uncertainty →

Clock Uncertainty is the time difference between the arrivals of clock signals at the registers in one clock domain or between domains.

$$T_{\text{Uncertainty}} = \text{Skew} + \text{Jitter} + \text{clock Margin (Noise + OCV)}$$

Types of timing paths →

- i) Input pin/port to Register (In-Reg).
- ii) Register to Register (Reg-Reg).
- iii) Register to Output port (Reg-out)
- iv) Input pin/port to Output pin/port (In-out)
- v) Reg-CG.S.

Setting Input-Output Delay Constraints →

Source latency →

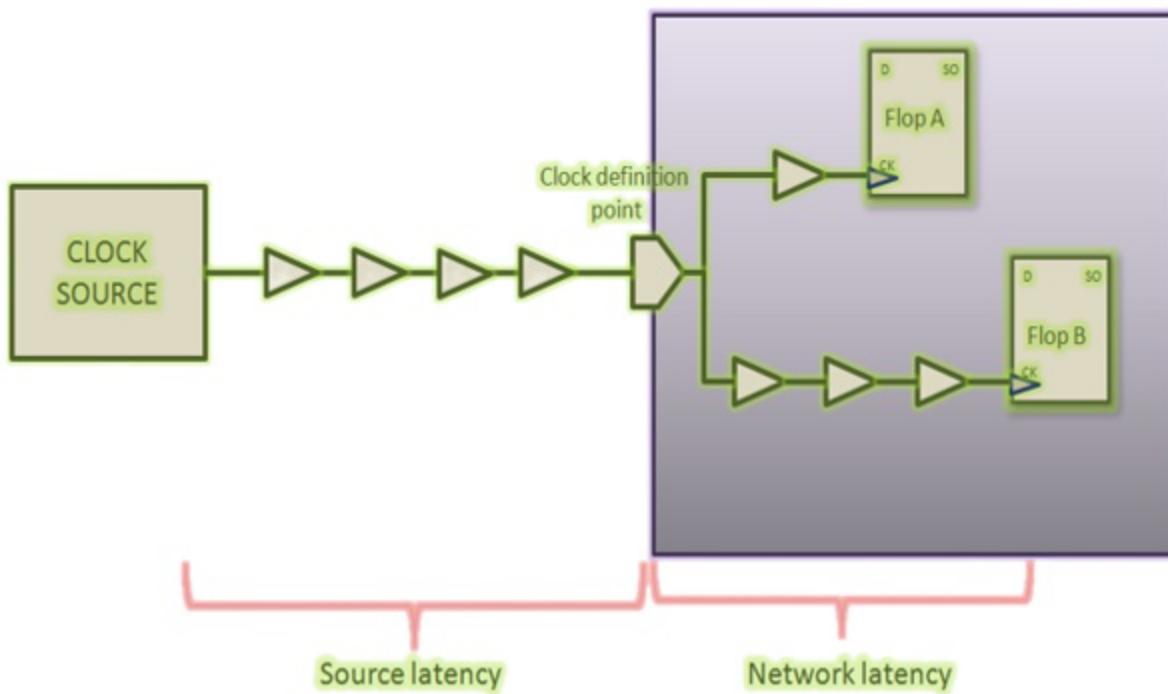
create_clock -period 10 [get_ports A]

set_clock_latency -source -max 3 [get_ports_clk]

Network latency →

set_clock_latency -source -max 2 [get_ports.Clk] # pre CTS

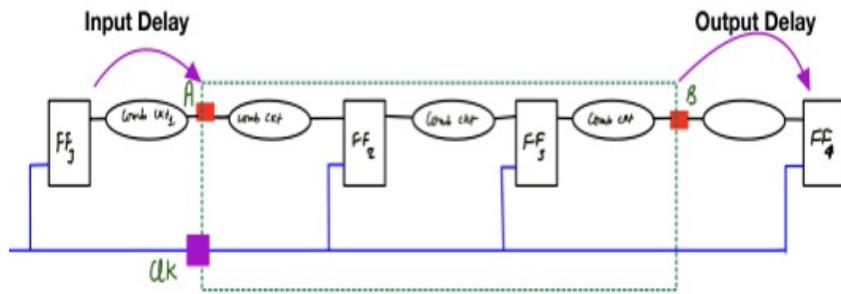
set_propagated_clock [get_ports A] # post CTS.



Setting Input Delay →

set_input_delay -max 0.6 -clock Clk [get_ports A]

set_output_delay -max 0.8 -clock Clk [get_ports B]



Timing Fixes :

Setup →

- (i) Upsizing
- (ii) V_t swapping (HVT → LVT/ULVT)
- (iii) Buffer addition (Breaking net)
- (iv) Cloning
- (v) Pin swapping
- (vi) logic restructuring

Hold →

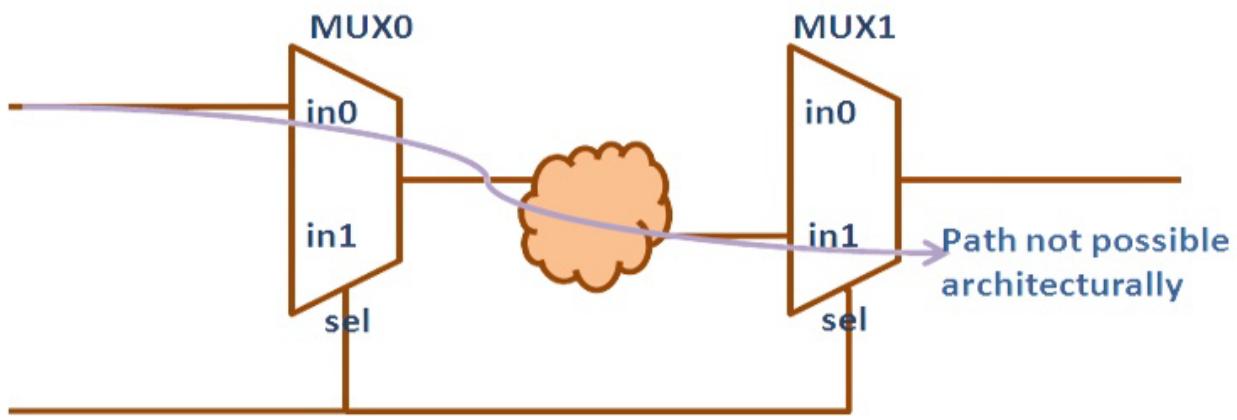
- (i) Downsizing
- (ii) V_t swapping (LVT/HVT - Higher V_t)
- (iii) Buffer Addition (Near to Capture flop)

Timing Exception →

(i) False path:

False path refers to a timing path in timing analysis is not done on that particular ^{path} as it will never get captured in a limited time frame when excited.

Set-false-path -from (Start point) -to (End point)



(ii) Multicycle path:

A multicycle path is flop to flop path , where the combinational logic delay in between the flops is permissible to take more than one clock cycle.

- set_multicycle_path 2 -setup **-end** -from CLK1 -to CLK2
- set_multicycle_path 1 -hold **-end** -from CLK1 -to CLK2

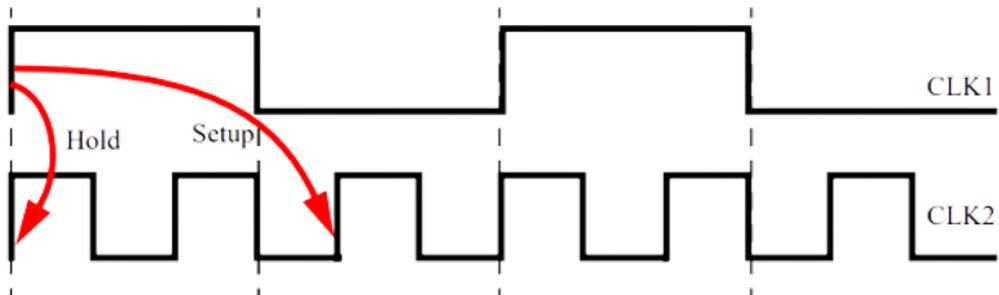


Figure 0-8. Setup 2 (**-end**) , Hold 1 (**-end**)

- set_multicycle_path 2 -setup **-start** -from CLK1 -to CLK2
- set_multicycle_path 1 -hold **-start** -from CLK1 -to CLK2

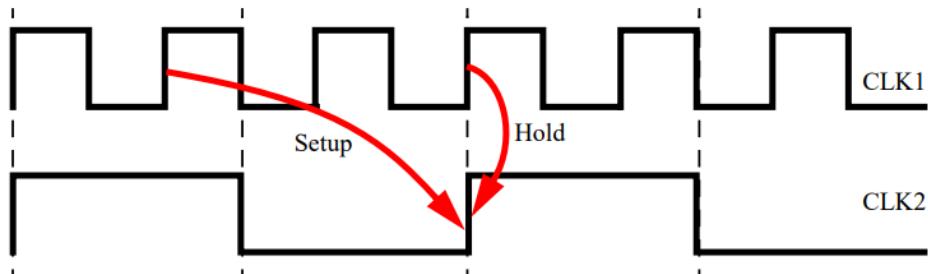
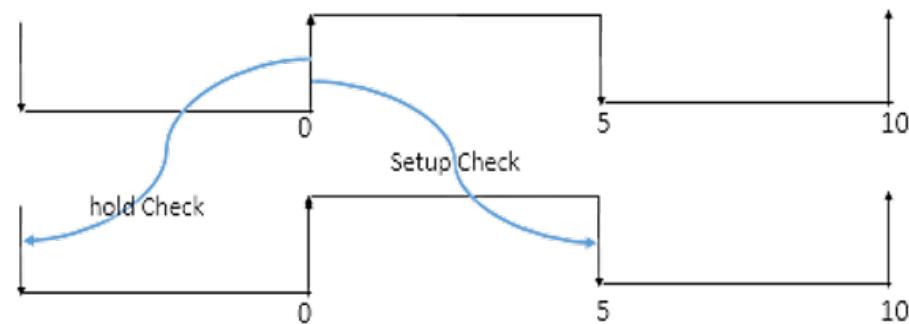
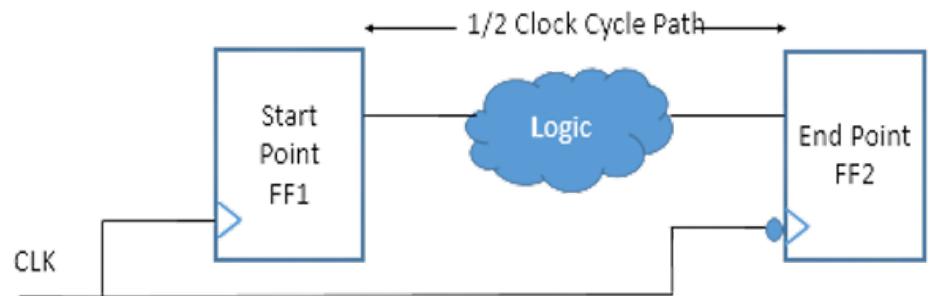


Figure 0-12. Setup 2 (**-start**) , Hold 1 (**-start**)

(iii) Half cycle path \rightarrow

Timing path that is designed to take half clock cycle (both of the clock edges) for the data to propagate from the start point to the end point.



Recovery and removal Checks \rightarrow

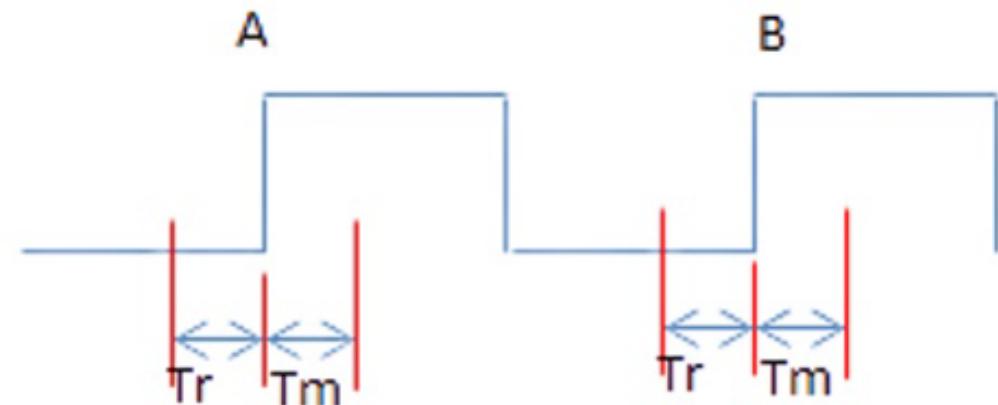
(i) Recovery and removal analysis are done on asynchronous signals like resets.

(i) Recovery time \rightarrow

It is the minimum required time to the next active edge after the reset

(ii) Removal time \rightarrow

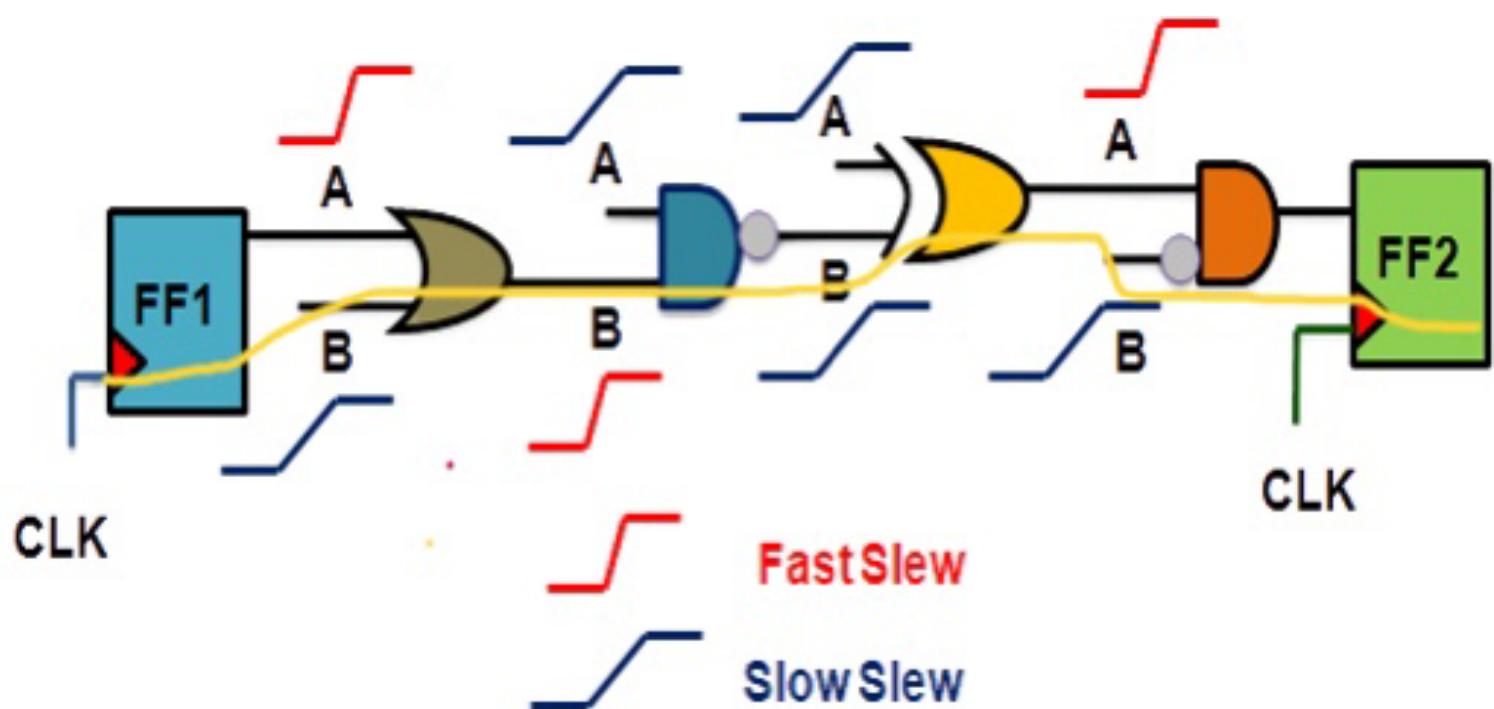
It is the minimum required time after which reset can be released.



GIBA and PBA \rightarrow

GIBA \rightarrow

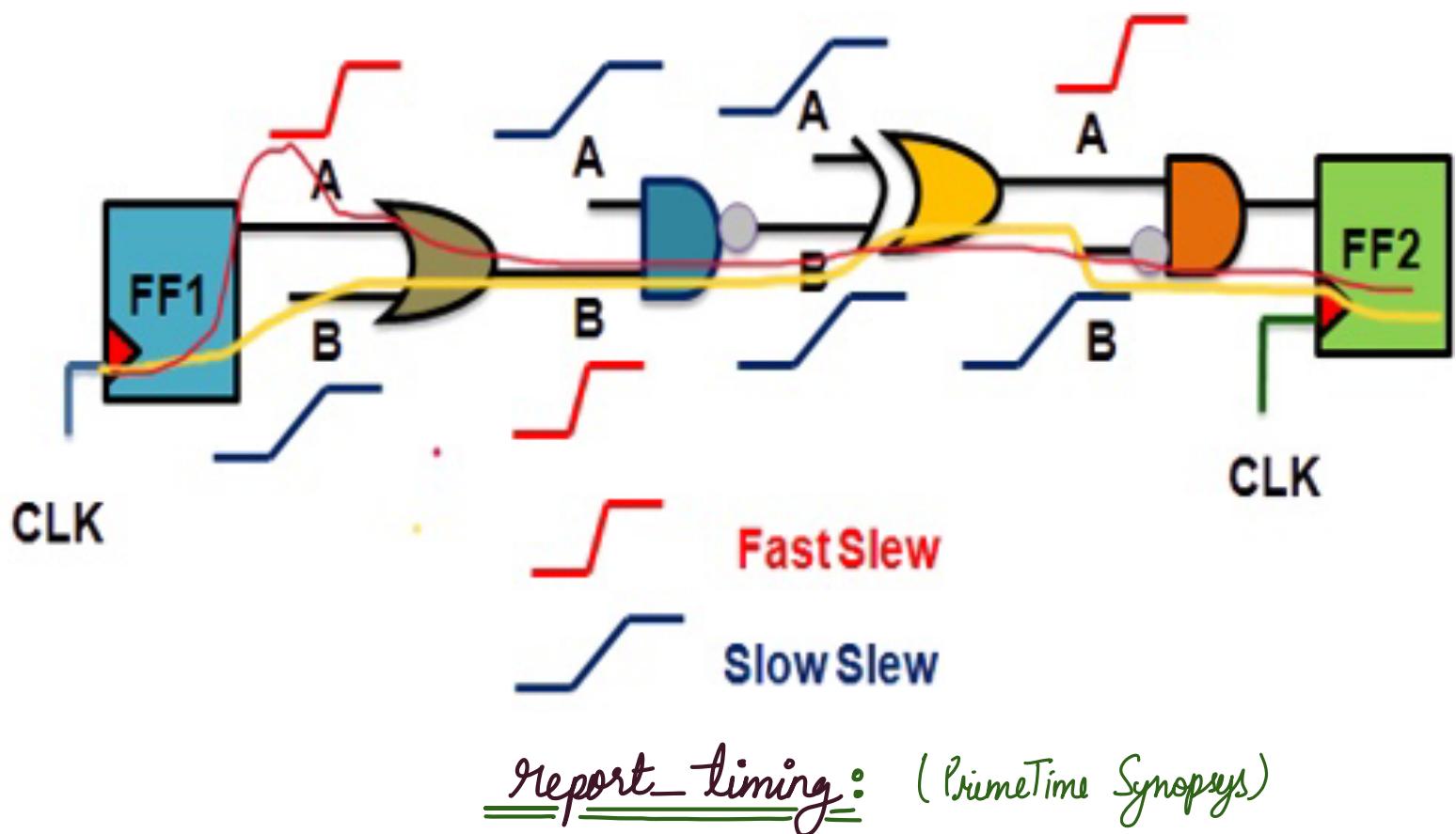
- * In GIBA mode the tool computes the path delay based worst case timing arcs of all the instances.
- * GIBA take less runtime as compared to PBA.



PBA →

In path based timing analysis, the tool considers each path in isolation from other paths, which eliminates impossible combination of worst slew and worst arrivals, and similar combination of effects such as crosstalk and CRP.

As a result, path based timing analysis reduces pessimism and increases accuracy at the cost of more runtime.



- The report_timing command reports the timing paths in the current design that have worst slack. These are the paths that violate the timing constraints by the large amounts, or paths with positive slack that comes closest to causing timing violation.

- Each path has a start point and an endpoint. Data is launched by a clock edge at the path startpoint, propagated through combinational logic in the path, and then captured at the path endpoint by another clock edge. The startpoint can be a register clock pin or an input port. The endpoint can be a register data input pin or an output port.
- By default, the report-timing command without options reports the single paths in the design with the worst max delay (setup constraint) violation. To consider constraints other than max delay, use the -delay-type option.
- To control the number of paths reported, use the -nworst option, which specifies the maximum number of worst paths reported per endpoint.
- The -max-paths option, which specifies overall maximum number of paths reported by the command.

TIMING REPORTS :

To invoke path based analysis, use the -pba-mode option. In path-based timing analysis, the tool considers each path in isolation from other paths.

The default timing report shows the point-to-point sequence of cell output pins in the data path, as i

```
Startpoint: I_ORCA_TOP/I_PCI_CORE/pad_en_reg
(rising edge-triggered flip-flop clocked by PCI_CLK)
Endpoint: pad[1] (output port clocked by PCI_CLK)
Path Group: PCI_CLK
Path Type: max
```

Point	Incr	Path
-------	------	------

			Symbol	Annotation
clock PCI_CLK (rise edge)	0.000	0.000		
clock network delay (propagated)	1.105	1.105		Hybrid annotation
I_ORCA_TOP/I_PCI_CORE/pad_en_reg/CP (sdcrq1)	0.000	1.105 r	H	Ideal network latency annotation
I_ORCA_TOP/I_PCI_CORE/pad_en_reg/Q (sdcrq1)	0.404 &	1.509 r	^	SDF back-annotation
U7/ZN (inv0d1)	0.066 &	1.575 f	*	RC network back-annotation
U62/ZN (inv0d1)	0.042 &	1.617 r	&	RC pi back-annotation
U63/Z (or02d7)	0.269 &	1.887 r	\$	Lumped RC
U63ASTipoInst495/Z (bufbdk)	0.306 &	2.192 r	+	HyperScale annotation
U63ASTipoInst494/Z (bufbda)	3.798 &	5.991 r		Wire-load model or none
pad_iopad_1/PAD (pc3b03)	6.118 H	12.109 r		
pad[1] (inout)	0.048	12.157 r		
data arrival time		12.157		
clock PCI_CLK (rise edge)	15.000	15.000	@	
clock network delay (propagated)	0.000	15.000		
output external delay	-4.000	11.000		
data required time		11.000		
data required time		11.000		
data arrival time		-12.157		
slack (VIOLATED)		-1.157		

```
pt_shell> report_timing -nets -input_pins -capacitance -transition_time
```

```
...
```

```
Startpoint: I_ORCA_TOP/I_PCI_CORE/pad_en_reg
(rising edge-triggered flip-flop clocked by PCI_CLK)
Endpoint: pad[1] (output port clocked by PCI_CLK)
Path Group: PCI_CLK
Path Type: max
Min Clock Paths Derating Factor : 0.900
```

Point	Fanout	Cap	Trans	Incr	Path
clock PCI_CLK (rise edge)			0.000	0.000	
clock network delay (propagated)			1.105	1.105	
I_ORCA_TOP/I_PCI_CORE/pad_en_reg/CP (sdcrq1)			0.363	0.000	1.105 r
I_ORCA_TOP/I_PCI_CORE/pad_en_reg/Q (sdcrq1)			0.140	0.404 &	1.509 r
I_ORCA_TOP/I_PCI_CORE/pad_en (net)	2	0.008			
U7/I (inv0d1)			0.142	0.008 &	1.517 r
U7/ZN (inv0d1)			0.089	0.058 &	1.575 f
n43 (net)	1	0.004			
U62/I (inv0d1)			0.089	0.004 &	1.579 f
U62/ZN (inv0d1)			0.071	0.038 &	1.617 r
n8 (net)	1	0.004			
U63/A2 (or02d7)			0.072	0.004 &	1.621 r
U63/Z (or02d7)			0.207	0.265 &	1.887 r
n134 (net)	8	0.060			
U63ASTipoInst495/I (bufbdk)			0.276	0.054 &	1.941 r
U63ASTipoInst495/Z (bufbdk)			0.136	0.252 &	2.192 r
n134ASTipoNet251 (net)	3	3.458			
U63ASTipoInst494/I (bufbda)			8.904	2.664 &	4.857 r
U63ASTipoInst494/Z (bufbda)			0.524	1.134 &	5.991 r
n134ASTipoNet250 (net)	1	1.725			
pad_iopad_1/OEN (pc3b03)			4.964	1.444 &	7.434 r
pad_iopad_1/PAD (pc3b03)			1.272	4.675	12.109 r
pad[1] (net)	1	8.586			
pad[1] (inout)			1.272	0.048	12.157 r
data arrival time					12.157
...					

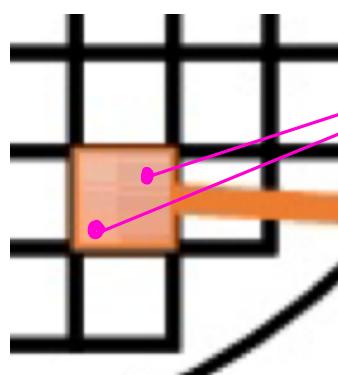
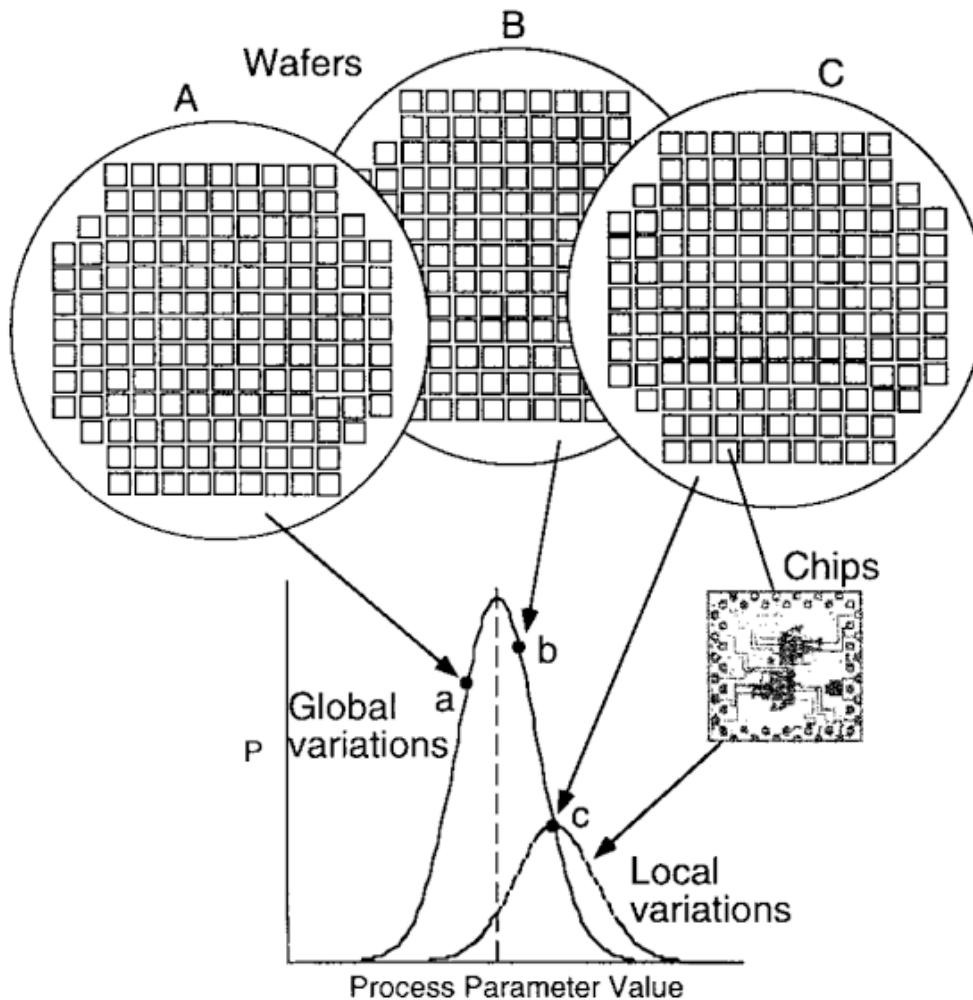
MINIMUM PULSE WIDTH CHECK:

- Minimum pulse width checks are done to ensure that width of the clock signal is wide enough for the cell's internal operation to complete. i.e to get a stable output you need to ensure that the clock signal at the clock pin of the flop is at least of certain 'minimum' width.
- Minimum pulse width is the interval between the rising edge of the signal crossing 50% of V_{DD} and the falling edge of the signal crossing 50% of V_{DD} .

Command to report Minimum Pulse width violation \Rightarrow
report-timing -check-type pulse-width

SEQUENTIAL CLOCK PULSE WIDTH			
Pin	Required Pulse Width	Actual Pulse Width	Slack
clk_ctrl_reg/CP (high)	0.3202	2.9731	2.6529

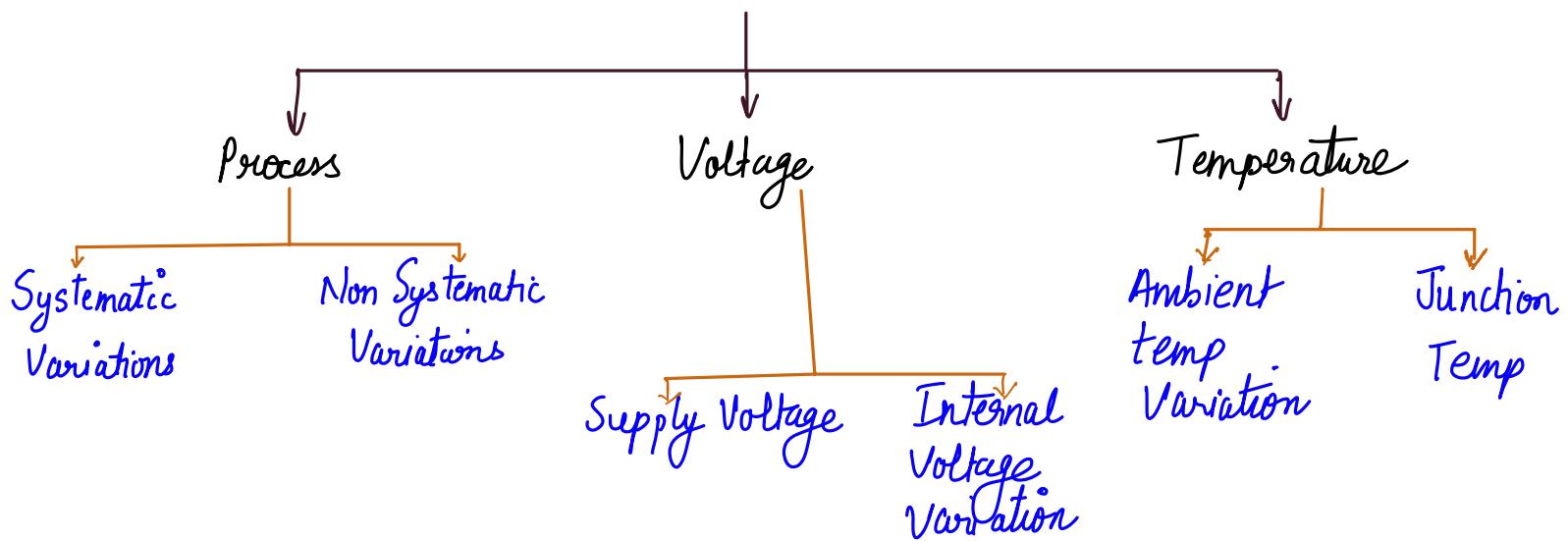
ON CHIP VARIATION :



Variation within a die
is termed as local variation

So inside a wafer, there is a variation in each die and also there is variation in characteristics of transistors even inside even inside a single IC along with the die.

Sources of Variation →



Process Variation →

In process variation there are two types of variation one is systematic variation and other is non systematic variation

Systematic Variation come due to optical proximity correction (OPC) or chemical Mechanical Policing which are predictable in nature and can be modeled in PVT Variations.

Non Systematic Variations come from the Random dopant Fluctuation (RDF), Line Edge Roughness (LER) or due to Oxide thickness variation (OTV) which are highly unpredictable and cannot be modeled easily.

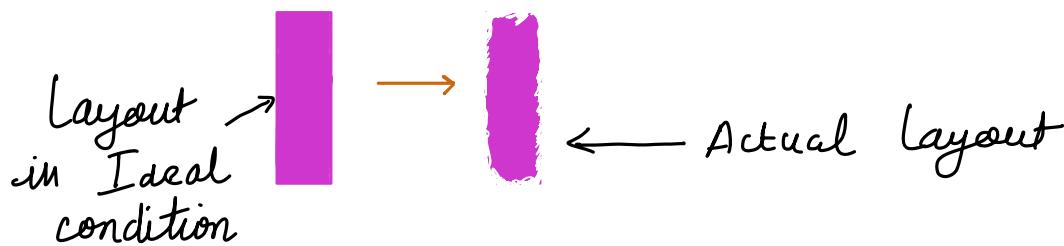
$$I_D = \frac{\mu_n \epsilon_{ox} (W)}{L} \left[(V_{gs} - V_{th}) V_{ds} - \frac{V_{ds}^2}{2} \right]$$

So drain current depends on

- $\mu_n \rightarrow$ mobility of electrons
- $\epsilon_{ox} \rightarrow$ Permittivity of silicon oxide
- $t_{ox} \rightarrow$ oxide thickness
- $W \rightarrow$ width of transistor
- $L \rightarrow$ gate length

So if any of these factors mentioned above varies during fabrication process. It will affect the drain current. The delay of a standard cell is going to vary.

A photolithography process is a non ideal process and it is very hard to print the exact layout on the silicon wafer.



So in conclusion, there are many factors and high chances of variation while fabrication of a chip and these can lead to vary the delay of the standard cells.

Voltage Variation -

One is due to the variation in external supply voltage and other is internal voltage variation inside the chip.

→ 2-5% variations in supply voltage.
Variations due to IR drop.

The external voltage variations is taken care in the PVT but, there could occur IR drop in your power delivery network which may lead to variation in available voltage to operate a cell.

Distance between the power pads and standard cells could not be the same for all standard cells. So there will be variation of available V_{DD} for the standard cells depending on the design. Delay of a cell is dependent on the available V_{DD} , If V_{DD} is less delay will be more.

$$V_{DD} \downarrow \rightarrow \text{Delay} \uparrow$$

Temperature Variations →

There is ambient temperature on which the chip is operating and another temperature is junction temperature of the transistors, junction Temperature is the sum of ambient temperature plus the temperature raised due to power dissipation of Chip.

Junction Temperature is always much greater than the ambient temperature and the characteristics of any transistors majorly depend on the junction temperature. Ambient temperature can be taken care in PVT but for the junction Temperature variations, we need to take care in OCV →

Sometimes there is also the formation of local hotspots based on the placement density and power requirements of cells which affects the temperature of the junction and ultimately lead to the variation in current and delay of cells

On Chip Variation →

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

OCV Derate factor →

A fixed derate factor is applied on throughout the design. So in that case any variation occurs will not cause failure of the chip. But it added too much of timing pessimism

which leads to difficulties in the timing closure, especially in the lower nodes.

% Set-timing-degrade 0.9 early
% Set-timing-degrade 1.1 late

Issues in OCV:

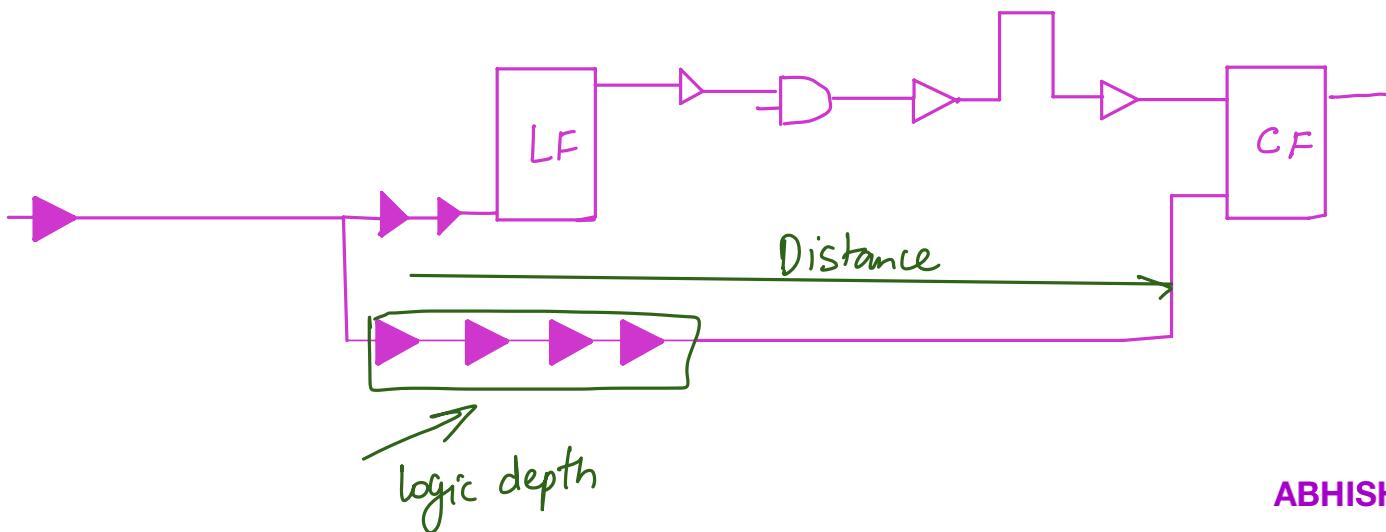
Fixed timing derate is used for all the cells in the OCV is over pessimistic. In reality, there is a cancellation of Random Variation effect.

So for the lower technology mode we want to resolve this issue. And so the concept of Advance on chip variation (AOCV) has evolved which does not use the fixed derates.

AOCV →

In AOCV derate is applied on each cell based on path depth and distance of the cell in the timing path and it also varies with cell type and drive strength of the cell.

Distance is defined by a bounding box for nets and cells.



Distance → If the distance increases, systematic variation would increase and to mitigate the variation, we need to use higher derate value. So along with the distance, derate values increases.

Path depth →

In the case of distance is fixed and path depth increases, systematic variation would be constant but the random variation would tend to cancel each other. Therefore the path depth increases the derate factor would decreases.

Cell type →

The derate is based on the cell type as an "AND" gate and "OR" gate does not exhibit the same variation. Derate value also varies with drive strength of the cell, like AND₂X₂ and AND₂X₆ will have drive derate values.

AOCV derate values depend upon 1D & 2D look up tables →

	Depth →							
	1	2	3	4	5	10	50	100
1000	1.130	1.099	1.055	1.058
2000	1.130	1.099	1.055	1.054
3000	1.131	1.100	1.056	1.055
4000	1.132	1.102	1.057	1.055
5000	1.137	1.105	1.061	1.059
6000	1.140	1.107	1.063	1.061
7000	1.141	1.109	1.071	1.062

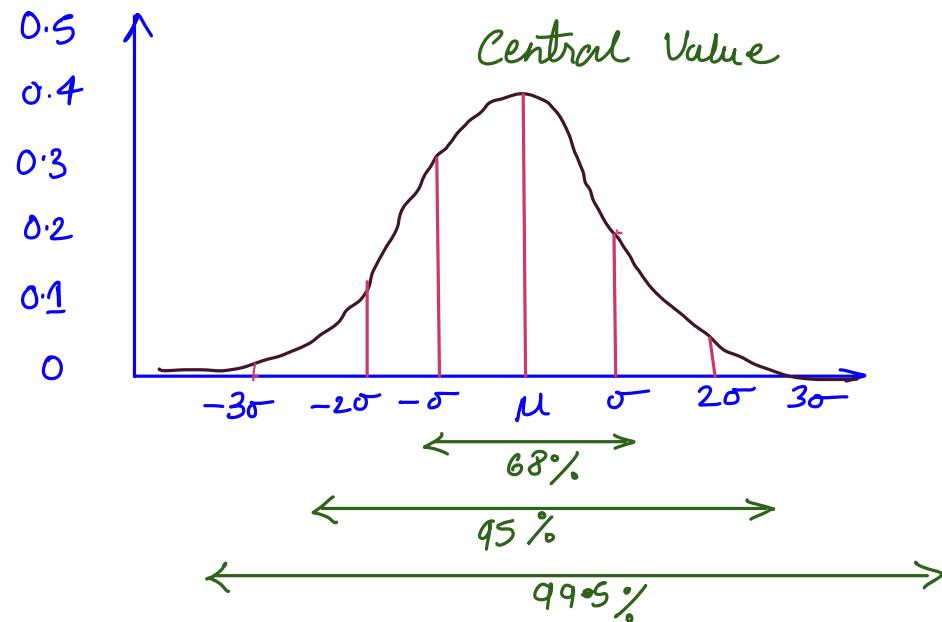
Issues in AOCV →

* AOCV does not perform very well below 40nm technology node and to improve that we need to improve the timing pessimism further. Distance and Depth based cludge factor used in AOCV is good for technology nodes above 40nm but for the below node we need it to improve it further. To address these issues Parametric on Chip Variation (POCV).

POCV →

In POCV instead of applying the specific cludge factor to a cell, cell delay is calculated based on delay variation (σ) of the cell.

In POCV it is assumed that the normal delay value of a cell follows the normal distributed curve.



POCV Analysis:

* POCV uses nominal delay values (μ) instead of using the min or max value of delay to model the random variations.

- * Timing Analysis is done using the nominal delay value(μ) and delay variation(σ) in the following way →
 - (i) Tool takes the value of σ from the timing library or an external file containing the POCV coefficient value C .
 - (ii) Each arc time is then calculated statically as the total of nominal delay and the variation.
 - (iii) The tool then calculates the delay of the path by statically combining these arc delay and perform setup and hold timing analysis

POCV Input data :

- (i) Using single POCV Coefficient (C) →
 - * An external file containing the delay coefficient values C for each library cell, hierarchical cell or design.
 - * There is only one value of C for each timing arc of the cell irrespective of the input transition or output load. The cell delay variation σ is calculated based on C as follows →

$$\text{The delay variation } (\sigma) = C * \text{Nominal delay}$$

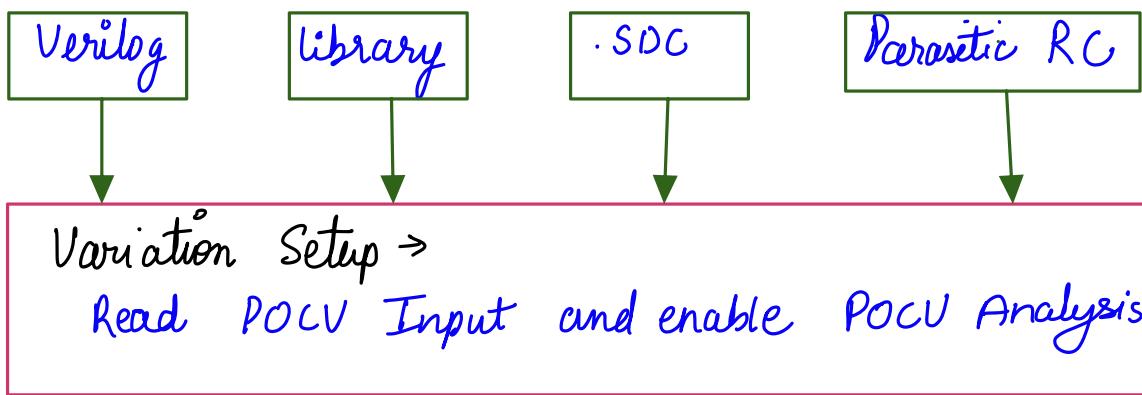
POCV Calculation \Rightarrow

$$\text{Delay of a cell} = \text{Nominal Delay} (\mu) \pm (C * \text{POCV coefficient})^{\frac{1}{N}}$$

where C = POCV Coefficient

N = Number of standard deviation

Prime time POCV Analysis flow \Rightarrow



Timing Analysis \Rightarrow
Apply statistical Addition / Subtraction & min/
max calculation.

Generate N-Sigma corner timing reports :
Path report Slack report Stage delay report

Comparison between POCV and AOCV \rightarrow

AOCV

(i) Random Variation modeled through the depth based derate and systematic variation is modeled through distance-based derate.

(ii) less accurate correlation between GBA and PBA.

(iii) Transition variation and cell check variation not supported.

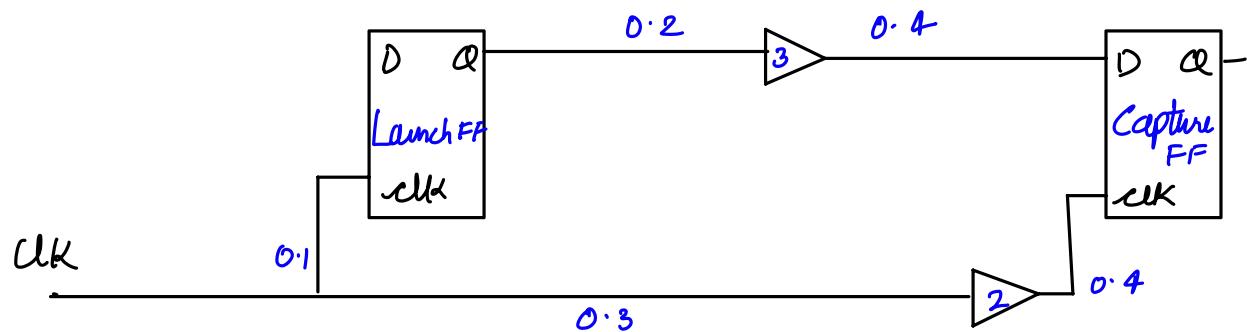
POCV

Random and systematic variation modeled through a delay Variation Coefficient σ which is specific to each cell.

More accurate correlation between GBA and PBA.

Transition Variation and cell check Variation is supported in LVF format.

Numericals \rightarrow



$$T_{clk} = 2 \text{ ns}$$

$$T_{clk-q} = 0.2 \text{ ns}$$

$$T_{setup} = 0.2 \text{ ns}$$

$$T_{hold} = 0.1 \text{ ns}$$

$$\text{Late derate} = 1.1 \text{ ns}$$

$$\text{Early derate} = 0.9 \text{ ns}$$

Setup Analysis \Rightarrow

while calculating the setup analysis we need to take late derate along launch path and early derate along capture path.

Arrival time \Rightarrow

Arrival time includes addition of cells wire delays along the launch path.

Arrival time (A.T) =

$$(\text{Wire 1 delay} \times \text{late derate}) + (\text{clk to q} \times \text{late derate}) + (\text{wire 2 delay} \times \text{late derate}) + (\text{inverter delay} \times \text{late derate}) + (\text{wire 3 delay} \times \text{late derate})$$

$$\text{Arrival time (A.T)} = (0.1 \times 1.1) + (0.2 \times 1.1) + (0.2 \times 1.1) + (3 \times 1.1) + (0.4 \times 1.1)$$

$$\text{Arrival Time (A.T)} = 4.29 \text{ nsec}$$

Required time \Rightarrow

$$\begin{aligned} \text{Required time (R.T)} &= T_{\text{clk}} - T_{\text{setup}} + [(\text{Wire delays} \times \text{early derate}) \\ &\quad + (\text{Tcell delays} \times \text{early derate})] \end{aligned}$$

$$R.T = 2 - 0.2 + [(0.3 \times 0.9) + (2 \times 0.9) + (0.4 \times 0.9)] = 4.23 \text{ ns}$$

$$\begin{aligned} \text{Setup Slack} &= R.T - A.T \\ &= 4.23 - 4.29 \\ &= -0.06 \text{ (-ve slack)} \end{aligned}$$

Hold Analysis →

while calculating the hold analysis we need to take early derate along the launch path and late derate along the capture path.

Arrival time →

(wire 1 delay * early derate) + (T_{clk-q} * early derate) + (wire 2 delay * early derate) + (inverter delay * early derate) + (wire 3 delay * early derate).

$$\begin{aligned} \text{Arrival time} &= (0.1 \times 0.9) + (0.2 \times 0.9) + (0.2 \times 0.9) + (3 \times 0.9) \\ &\quad + (0.4 \times 0.9) = 3.51 \text{ nseconds} \end{aligned}$$

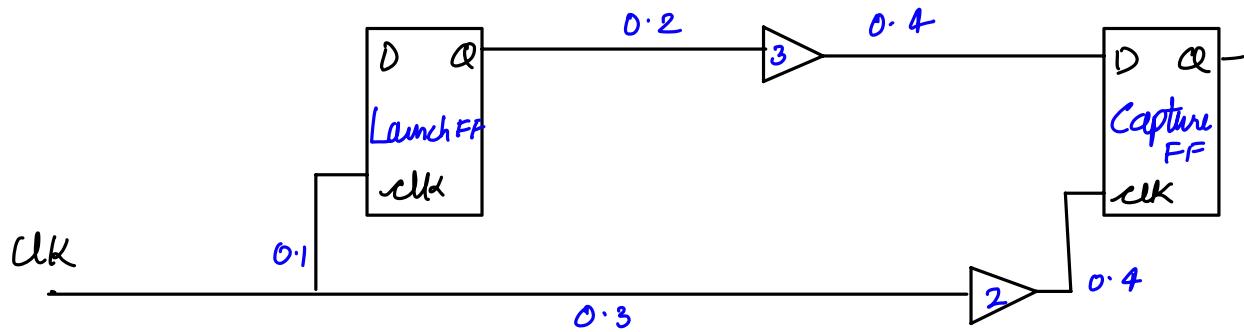
Required time →

Required time (R.T) = $T_{hold} + (\text{Twire delays} \times \text{late derate}) + (T_{cell} \times \text{late derate})$

$$\begin{aligned} \text{Required Time (R.T)} &= 0.1 + (0.3 \times 1.1) + (2 \times 1.1) + (0.4 \times 1.1) \\ &= 3.07 \text{ nseconds} \end{aligned}$$

$$\begin{aligned} \text{Hold Slack} &= A.T - R.T = 3.51 - 3.07 \\ &= 0.44 \text{ nseconds (positive slack)} \end{aligned}$$

Case - 2 Without OCV



$$T_{clk} = 2 \text{ ns}$$

$$T_{clk-q} = 0.2 \text{ ns}$$

$$T_{setup} = 0.2 \text{ ns}$$

$$T_{hold} = 0.1 \text{ ns}$$

$$\text{Late derate} = 1.1 \text{ ns}$$

$$\text{Early derate} = 0.9 \text{ ns}$$

Setup Analysis \Rightarrow

$$(A \cdot T) = \text{wire 1} + \text{clk to } q + \text{inv delay} + \text{wire 3}$$

$$(A \cdot T) = 0.1 + 0.2 + 0.2 + 3 + 0.4 = 3.9 \text{ nanoseconds}$$

$$(R \cdot T) = T_{clk-q} - T_{setup} + T_{wire delays} + T_{cell delays}$$

$$(R \cdot T) = 2 - 0.2 + 0.3 + 2 + 0.4 = 4.5 \text{ nanoseconds}$$

$$\begin{aligned} \text{Setup Slack} &= R \cdot T - A \cdot T \\ &= (4.5 - 3.9) = 0.6 \text{ nanoseconds} \end{aligned}$$

Hold Analysis \Rightarrow

$$A \cdot T = \text{wire 1} + \text{clk- } q + \text{inv delay} + \text{wire 3}$$

$$A \cdot T = 0.1 + 0.2 + 0.2 + 3 + 0.4 = 3.9 \text{ ns}$$

$$R \cdot T = T_{hold} + T_{wire delays} + T_{cell delays}$$

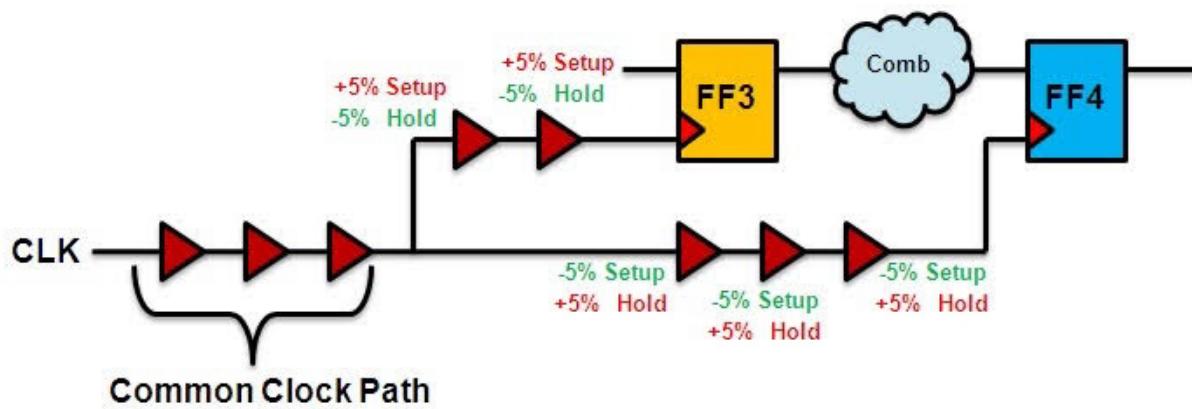
$$R \cdot T = 0.1 + 0.3 + 2 + 0.4 = 2.8 \text{ ns}$$

$$\begin{aligned} \text{Hold Slack} &= (A \cdot T - R \cdot T) \\ &\Rightarrow 1.1 \text{ nanoseconds} \end{aligned}$$

CRPR/GPPR :

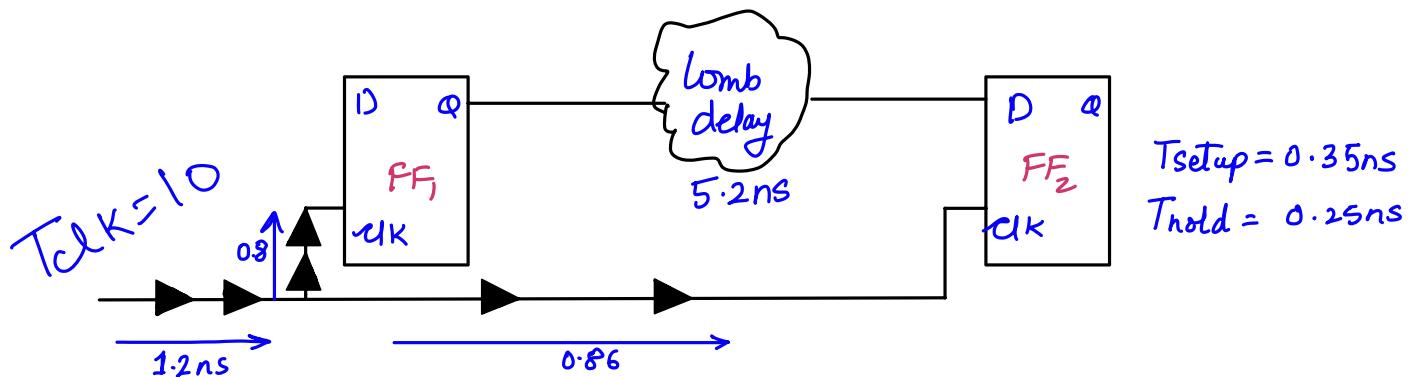
CRPR \rightarrow Clock Reconvergence Pessimism removal.

C PPR \rightarrow Clock Path pessimism removal.



- * During Timing Analysis a condition arise where you have to use max delay for one timing path and min delay for another timing path.
- * Due to the common path we cannot take max delay/min delay at the same time (For example Common to Data and Clockpath).
- * Removing common clock buffer delay between launch path & capture path is C PPR and CRPR.

Numericals \rightarrow



$$T_{setup} = 0.35 \text{ ns}$$

$$T_{hold} = 0.25 \text{ ns}$$

Set_timing_derate	- early	0.9
set_timing_derate	- late	1.2

Solution →

$$\text{Setup Slack} = (R \cdot T) - (A \cdot T)$$

$$A \cdot T = 1.2 \text{ ns} + 0.8 + 5.2$$

$$R \cdot T = 10 + 1.2 + 0.86 - 0.35 \quad (T_{\text{Setup}})$$

$$\begin{aligned}\text{Setup Slack} &= 11.71 - 7.2 \\ &= 4.51\end{aligned}$$

$$\text{Hold Slack} = (A \cdot T) - (R \cdot T)$$

$$A \cdot T = 1.2 \text{ ns} + 0.8 \text{ ns} + 5.2 \text{ ns}$$

$$R \cdot T = 1.2 \text{ ns} + 0.86 + 0.25 \quad (T_{\text{Hold}})$$

$$\begin{aligned}\text{Hold Slack} &= 7.2 - 2.31 \\ &= 4.89 \text{ nsec}\end{aligned}$$

Using CPPR and derates → Derate Value = 0.9/1.1

$$\text{Setup Slack} = R \cdot T - A \cdot T$$

$$R \cdot T = [10 + (0.86 \times 0.9) - (0.35 \times 1.1) + 1.2(1.1 - 0.9)] \quad (\text{CPPR})$$

$$A \cdot T = [(1.2 \times 1.1) + (0.8 \times 1.1) + (5.2 \times 1.1)]$$

$$\text{Hold Slack} = A \cdot T - R \cdot T$$

$$R \cdot T = [(0.8 \times 1.1) + (0.25 \times 1.1) - 1.2(1.2 - 0.9)] \quad (\text{CPPR})$$

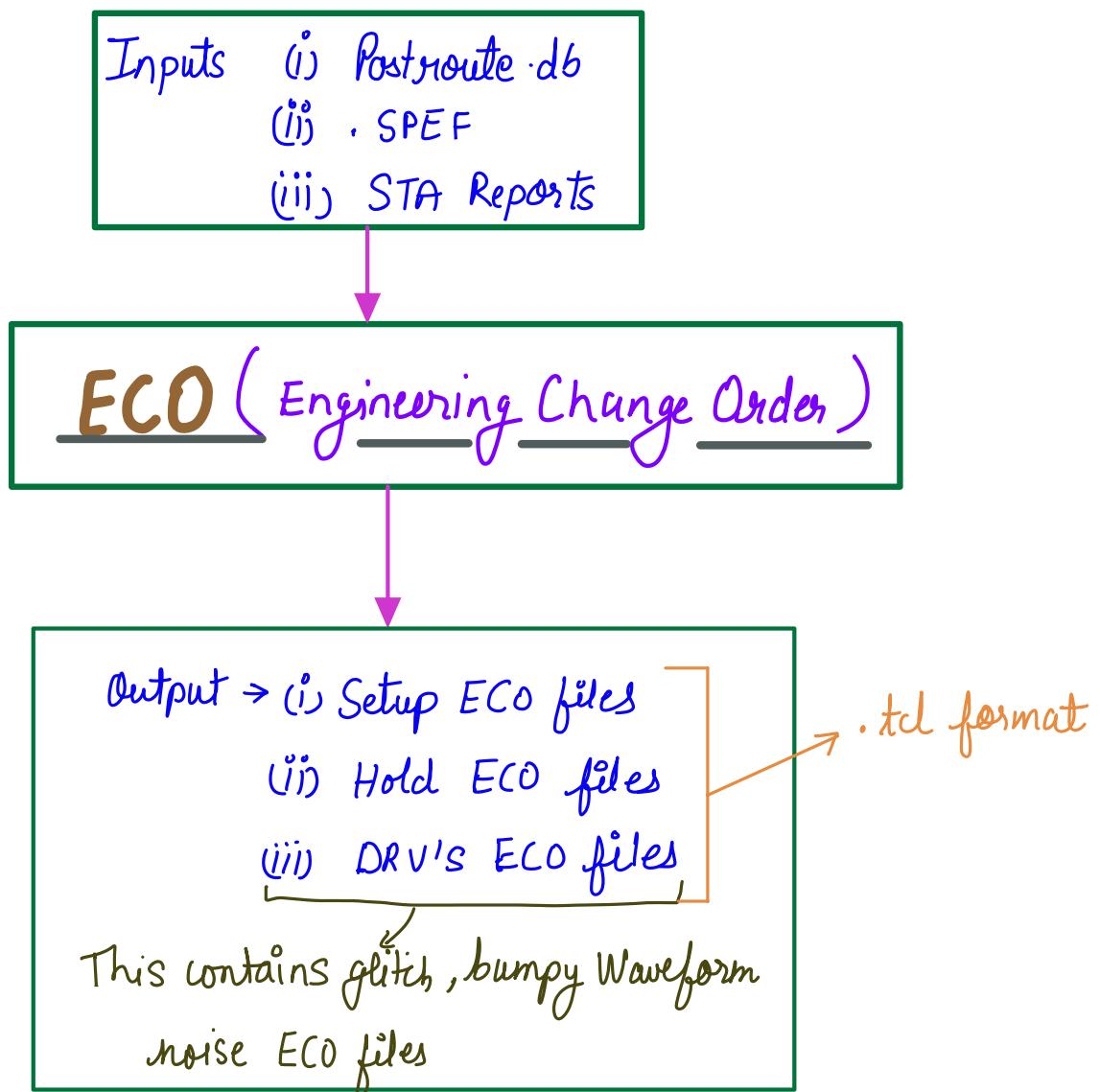
$$A \cdot T = [(1.2 \times 0.9) + (0.8 \times 0.9) + (5.2 \times 0.9)]$$

Question Why do we need to go with STA if the timing is already clean in Post route stage?

Answer → After .SPEF generation we get accurate values of R and C which our PnR tool was not able to determine and may be due to this RC delays there may be degradation in the timing path. Hence we go for Static timing Analysis.

Sign off tools can generate tool based ECO's for the current existing violations.

ECO (Engineering Change Order)



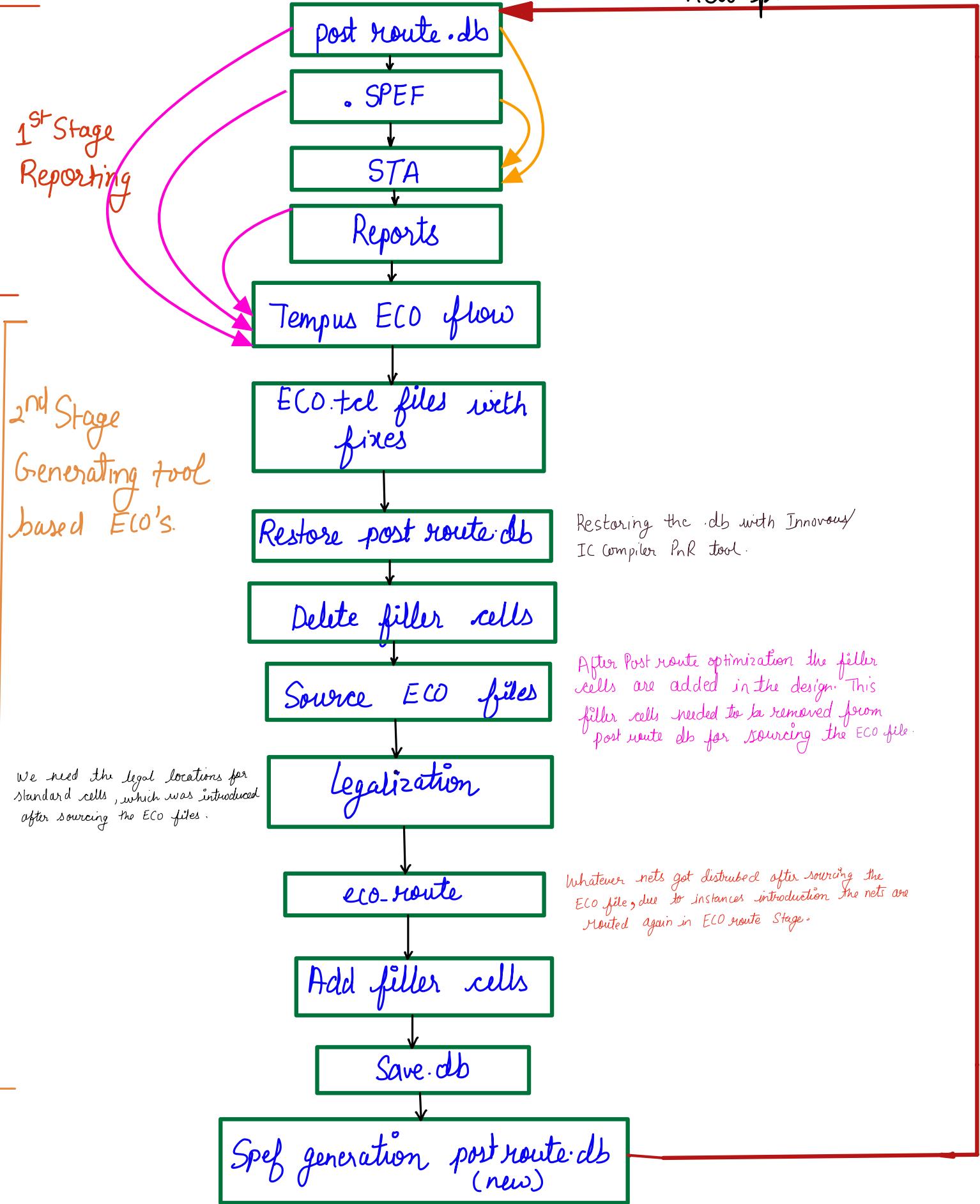
ECO Flow →

1st Stage
Reporting

new postroute.db

2nd Stage
Generating tool
based ECO's.

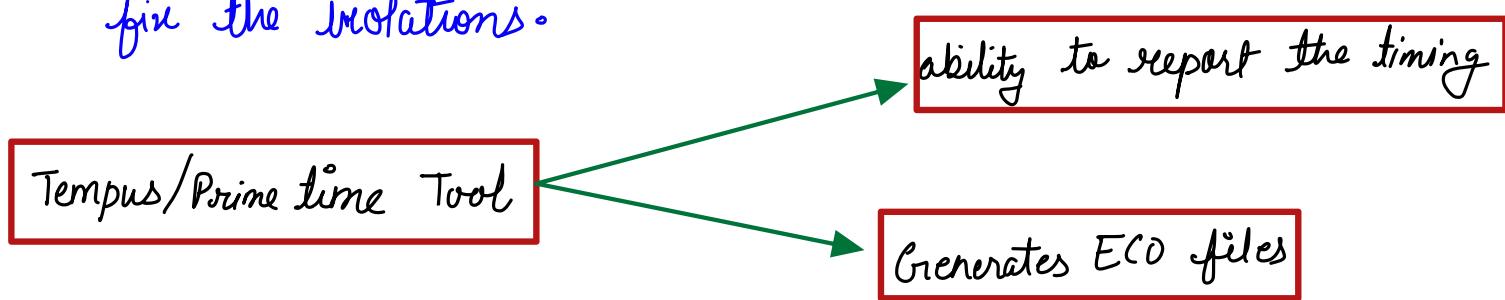
We need the legal locations for standard cells, which was introduced after sourcing the ECO files.



* If the tool generated ECO is not able to fix the timing then we have to go with manual ECO's. Generally 1-2 iteration is done with tool ECO still if degradation in timing is observed we go with manual ECO's.

Manual ECO →

Manual ECO means user have to write the .tcl script to fix the violations.



(i) `Size_cell -cellname Fusion Compiler`

(ii) `eco_update_cell -instance name -celltype Innovous`

a) you can either source the .tcl file by restoring the.db or

b) you can do it in the PnR stage.

