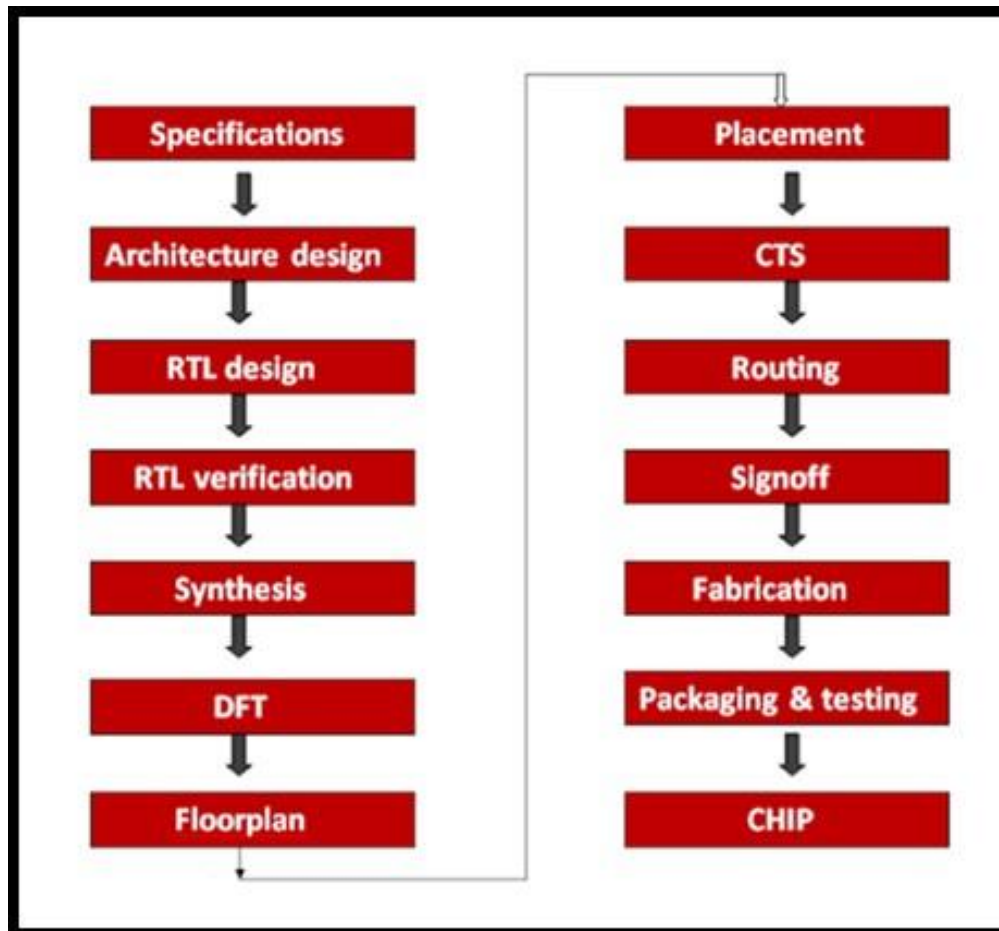


## CONTENTS

SI NO	CHAPTER	PAGE NO
1	ASIC DESIGN FLOW	1
2	SYNTHESIS BASICS	4
3	PD INPUT FILES	7
4	FLOOR-PLAN	12
5	PLACEMENT	20
6	STATIC TIMING ANALYSIS	26
7	SIGNAL INTEGRITY ISSUES	45
8	CLOCK TREE SYNTHESIS	57
9	ROUTING	70
10	PHYSICAL VERIFICATION	75

## ASIC Design Flow



### Specifications: -

In this stage the features information which is expecting from the customer is collected by some marketing people.

### Architecture design: -

The architecture team will design an architecture based on the specifications. The architecture is like a block diagram we can find the all the details which are using in the design (like Processors, memories) and how they are connected. This architecture team will estimate the block area, how much power is required and cost for the design.

### RTL design: -

Register transfer level(RTL) constructing a digital design using combinational and sequential circuit in hardware description language like verilog or VHDL. The above architecture is converted into verilog or VHDL code. This code describes how data is transformed as it is passed from register to register.

**RTL Verification:-**

It is a functional verification of RTL design. After the RTL design by applying test cases we verify the design in verification stage. If any mistakes are found, then the design is re send to the RTL designing department.

**Synthesis:-**

It is a process of converting the RTL code into gate level netlist. Up to RTL verification the design is technology independent. In synthesis process the design is converted into technology dependent gate level netlist.

**DFT:-**

Design for testability(DFT) is a technique which facilitates a design to become testable after production. In this stage we put extra logic like Scan path insertion along with the design logic during implementation process which helps post production process. The DFT will make the testing easy at post production process. At this stage an ATPG (Automatic Test Pattern Generator) file will generated.

**Floorplan:-**

The floorplan is the process of determining die area/core area . It determines the size of the die and creates wire tracks for placement of standard cells. It creates power straps and specifies power ground connection. In this step port/pad Placement and Macro placement is also done.

**Placement:-**

Placement is the process of automatically assigning correct position to standard cells inside core area with no overlapping. By global placement, standard cells will placed be inside core area roughly. By the detailed placement the standard cells will place in site rows (legalize placement). after placement stage we check for congestion and Timing we also reduce it.

**CTS (clock tree synthesis):-**

In this stage we build the clock tree by using inverters and buffers. In the chip clock signal is essential to the flip flops. To supply the clock signal from clock source we built the clock tree. It is the process of balancing the clock skew and minimizing insertion delay in order to meet timing and power.

**Routing: -**

Before the routing stage the connection between the macros, standard cells, clock, i/o port are logical connections. In this stage we connect all the cells physically with the metal straps. Routing is divided as two parts 1) Global routing 2) detailed routing. The global routing will tell for which signal which metal layer is used. In detailed routing the physical connections are done.

**Timing Signoff: SPEF extraction and STA is done after this stage****Signoff :**

After the routing the physical layout of chip is completed. In signoff stage all the tests are done (DRC, LVS and LEC) to check the quality and performance of the layout before tape out. After this the design is converted into GDS II file.

**Fabrication: -**

By the GDS II file information we fabricate the chip. The total design is converted into chip by the manufacturing process in foundry.

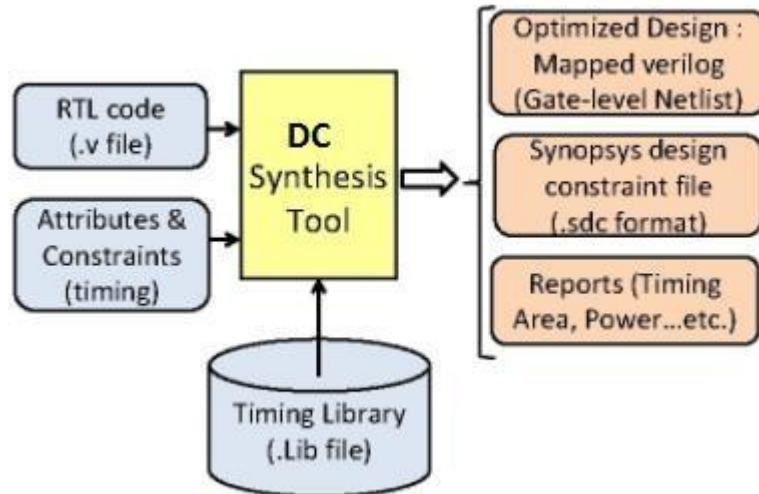
Assignment: Know all the tools for each and every step by different vendors.

Si No	Design Phase	Synopsis	Cadence	Mentor Graphics / Siemens
1	Functional Verification			
2	Synthesis			
3.	Placement and Routing			
4.	SPEF extraction			
5.	STA			
6	Physical verification			
7	Power Sign-off			

## Synthesis

Synthesis transforms the simple RTL design into a gate-level netlist with all the constraints as specified by the designer. In simple language, Synthesis is a process that converts the abstract form of design to a properly implemented chip in terms of logic gates.

### Input Files for Synthesis:

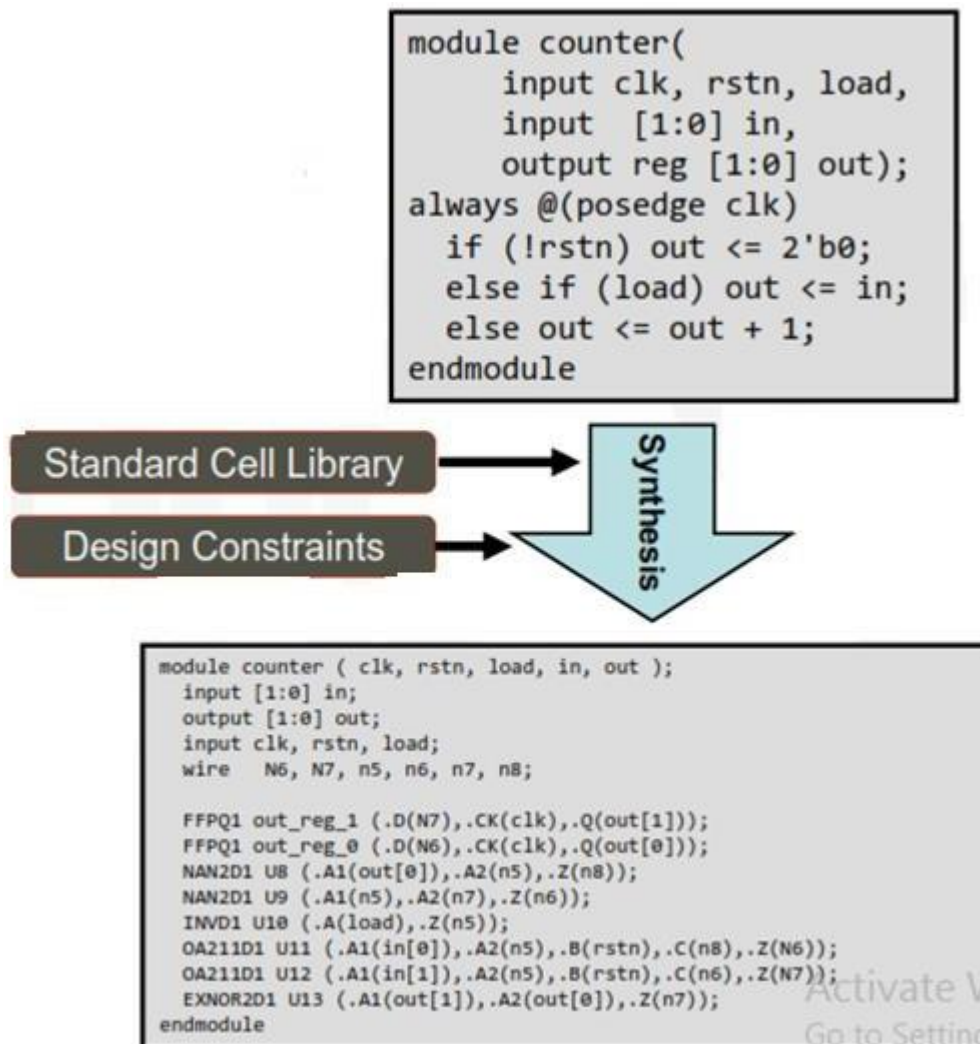


- 1) RTL Code (.v) : Design Written in Verilog/VHDL in behavioral style modelling.
- 2) Logical Library file (.lib) : The logical library is also called a timing library or functional library or power library as it contains the functionality, time and power information of standard cells of particular technology.
- 3) Constraint file. (.sdc) : It provides design Timing, area and power constraints to synthesis tool ( Mainly clock information ).

### Synthesis steps

Synthesis takes place in multiple steps:

- 1) Elaboration: Elaboration is the **process of expanding your HDL description to represent all instances of** all modules(Verilog) or entities(VHDL) into unique objects
- 2) Converting RTL into simple logic gates (Technology independent Schematic) .
- 3) Mapping those gates to actual technology-dependent logic gates available in the technology libraries.
- 4) Optimizing the mapped netlist keeping the constraints set by the designer intact.



### Output from Synthesis tool:

- 1) Verilog Netlist (.v)
- 2) Constraint file (.sdc)

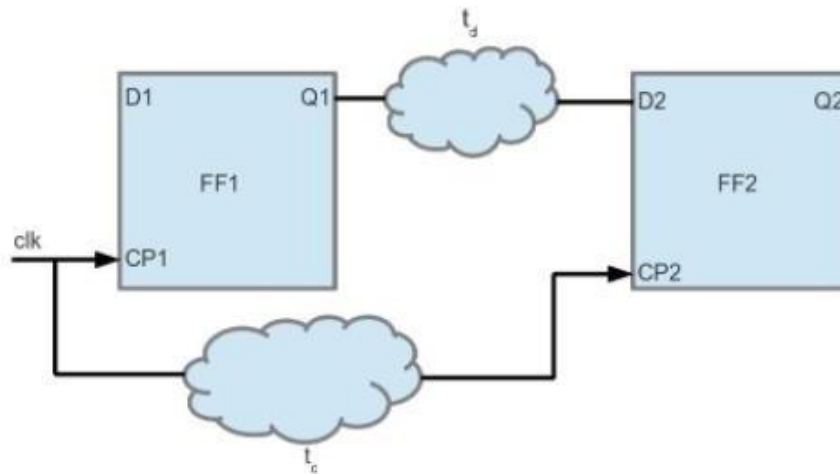
### Checks performed During synthesis

- 1) Static Timing Analysis: Setup and hold check
- 2) Power Analysis
- 3) Area Analysis.

**Assignment :**

For timing Analysis : Following information is required to do STA . Kindly mention where you get following information

1	Flip-flop setup time and hold time	
2	Flip-flop delay	
3	Combinational circuit delay.	
4	Wire delay	
5	Clock frequency/clock information	



## **INPUT FILES FOR PD**

Following input files are required to start PD:

SI no	File Name	Extension	Lender
1	Verilog Netlist	.v	Synthesis team
2	Constraints	.sdc	Synthesis team
3	Logical libraries (Std cells And macros )	.lib (.db)	Vendor
4	Physical libraries (Std cell and macros )	.lef (.Fram_views)	Vendor
5	Technology File	.tf	Foundry
6	TLU + file	. TLUP	Foundry

### **NETLIST**

Netlist: Format is .V

1. It contains Logical connectivity Of all Cells (Std cells, Macros).
2. It contains List of nets Connecting std cells and Macros
3. Each cell has its own instance/cell name and library/ref name

### **SDC**

SDC :Format is .SDC :

These Constraints are timing Constraints .

These Constraints are mainly used to meet timing requirements of design .

Constraints are

1. CLOCK DEFINITIONS: To create different types of clocks
2. Clock uncertainty.
3. Setting Input Delay
4. Setting Output Delay
5. Driving Cell
6. Setting load on output ports

.....>Exceptions<.....

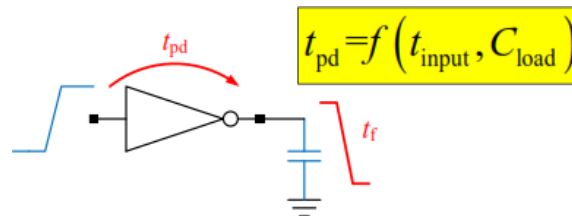
7. Multi cycle path
8. False path
9. Half cycle path
10. Disable timing arcs
11. Case Analysis



## LOGICAL LIBRARIES

Logical libraries: format is .lib

1. Timing information of Standard cells, macros.
2. Functionality information of Standard cells.
3. Timing DRV like max transition, max capacitance, max fan-out.
4. In timing information look-up table is used for output transition, Cell delays, Setup, hold time.
5. Cell delay is Function of input transition and output load. Cell delay is calculated based on lookup tables.
6. It also has wire load model to calculate resistance and capacitance of wires
7. Functionality is used for Optimization Purpose.
8. It also Contain Power information of Std cells.



### Look-Up Table

```
lu_table_template(delay_template_5x5) {
  variable_1 : input_net_transition;
  variable_2 : total_output_net_capacitance;
  index_1 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0");
  index_2 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0");
}
```

```
cell (INVX1) {
  pin(Y) {
    timing() {
      cell_rise(delay_template_5x5) {
        values ( \
          "0.147955, 0.218038, 0.359898, 0.922746, 1.76604", \
          "0.224384, 0.292903, 0.430394, 0.991288, 1.83116", \
          "0.365378, 0.448722, 0.584275, 1.13597, 1.97017", \
          "0.462096, 0.551586, 0.70164, 1.24437, 2.08131", \
          "0.756459, 0.874246, 1.05713, 1.62898, 2.44989"); }
    }
  }
}
```

Above is the table for calculating rise cell delay, Index-1 is input transition values and index-2 is output load capacitance values.

**Wire load models:**

```

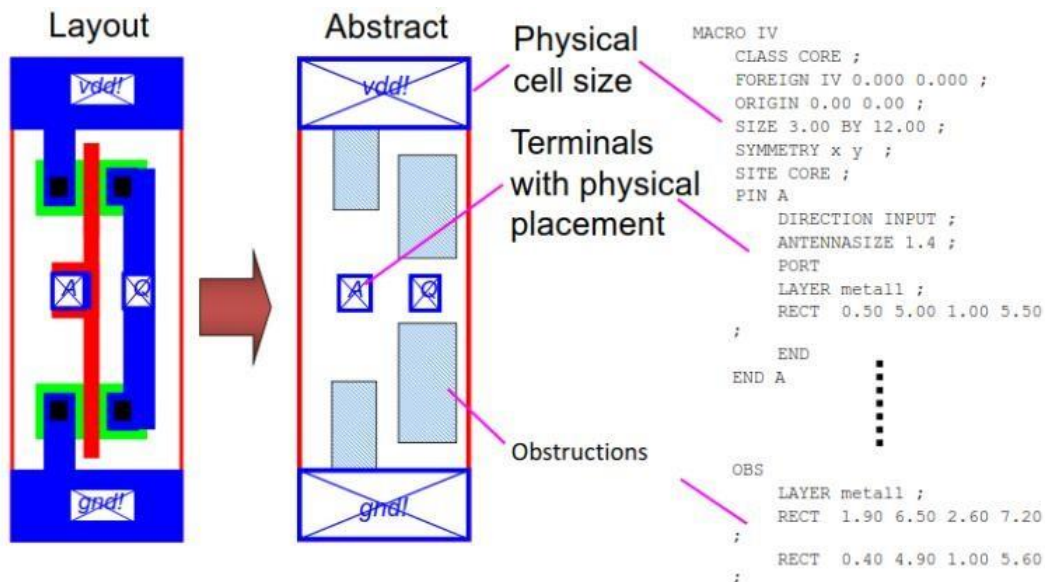
library (myLib) {
  wire_load("WLM1")
    resistance: 0.0006 ; // R per unit length
    capacitance: 0.0001 ; // C per unit length
    area : 0.1 ; // Area per unit length
    slope : 1.5 ; // Used for linear extrapolation
    fanout_length(1, 0.002) ; // for fo=1, Lwire=0.002
    fanout_length(2, 0.006) ; // for fo=2, Lwire=0.006
    fanout_length(3, 0.009) ; // for fo=3, Lwire=0.009
    fanout_length(4, 0.015) ; // for fo=4, Lwire=0.015
    fanout_length(5, 0.020) ; // for fo=5, Lwire=0.020
    fanout_length(6, 0.028) ; // for fo=6, Lwire=0.028
  }
} /* end of library */

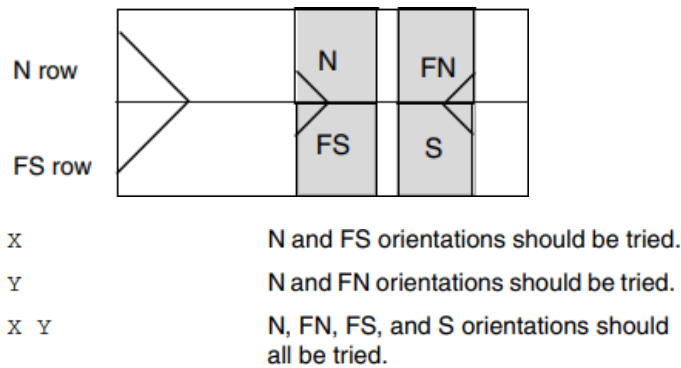
```

**PHYSICAL LIBRARIES**

Physical libraries: format is .lef (. Fram views for synopsis)

1. It contains physical information of standard cells, macros, pads.
2. Contain the name of the pin, pin location (Co-ordinates) , pin layers, direction of pin (in, out, inout), uses of pin (Signal, Power, Ground) , height and width of the pin and cell.
3. Size of the cell (Height and width)
4. Symmetry of cell.





### **TECHNOLOGY FILE:**

1. It contains manufacturing grid definition and site/unit tile definition
2. It contains Name, Number conventions of layer and via
3. It contains Physical, electrical characteristics of layer and via
4. In Physical characteristics Min width, Min Spacing, Min Height are present.
5. In Electrical characteristics Max Current Density is present.
6. Colors and pattern of layer and via .
7. Physical Design rules of layer and via
8. Tech file used by the Cadence tool is .techlef format and .tf format by Synopsys tool.

```

LAYER MET1
  TYPE ROUTING ;
  PITCH 3.5 ;
  WIDTH 1.2 ;
  SPACING 1.4 ;
  DIRECTION HORIZONTAL ;
  RESISTANCE RPERSQ .7E-01 ;
  CAPACITANCE CPERSQDIST .46E-04 ;
END MET1

LAYER VIA
  TYPE CUT ;
END VIA

```

### **TLU PLUS**

TLU+ files: format is .TLUP:

1. R,C parasitics of metal per unit length.
2. These (R,C parasitics) are used for calculating Net Delays.
3. If TLU+ files are not given then these are getting from. ITF file.
4. For Loading TLU+ files we have to load two files .
5. Those are Max TLU+, Min TLU+

**MAP file.**

1. MAP file maps the layer and via names of TLU+ file and .tf file .

**UPF- File (Unified Power Format )**

1. UPF is designed to reflect the power intent of a design at a relatively high level.
2. UPF scripts describe which power rails should be routed to individual blocks, when blocks are expected to be powered up or shut down.
3. It describes how voltage levels should be shifted as signals cross from one power domain to another and whether measures should be taken to retain register and memory-cell contents if the primary power supply to a domain is removed.

Note : To Store Library and design Information ICC2 uses NDM format .  
NDM has all input files in compiled format.

**Assignment**

1. Open Netlist (.v) and Understand Hierarchical design
2. Open .lib and for cell AND2X4\_HVT note following information
  - a) Leakage power
  - b) Internal Power table
  - c) Dynamic power table
  - d) Rise/fall output transition table
  - e) Rise/fall cell delay table
  - f) On output pin Note max\_capacitance and max\_transition
3. Open. lef file for AND2X4\_HVT cell and note
  - a) Its dimension
  - b) Allowed Orientation
  - c) Its pins and Pin layers
4. Open .tf and Note
  - a) Site row information
  - b) Manufacturing grid
  - c) For all routing metal layers note
    - i) Min spacing
    - ii) Min width
    - iii) Pitch
    - iv) Routing direction

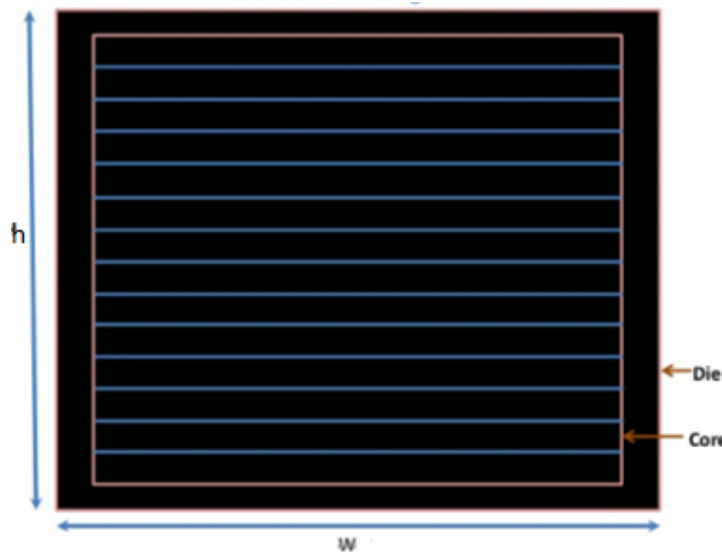
## Floor Planning

Floor-plan design is an important step in physical design of VLSI circuits to plan the positions of a set of circuit modules on a chip in order to optimize the circuit performance. *Floor planning* is the process of creating an area for macros and standard cells to be placed

### Floor Planning Steps :

1. Decide core width and height for die size estimation.
2. Placement of IO pads/Ports .
3. Creating Voltage area.
4. Placement of macros.
5. Adding physical only cells
6. Power planning (pre routing)

#### 1) Decide core width and height for die size estimation.



Core area depends upon :

1. **Aspect ratio:** Aspect ratio will decide the size and shape of the chip. ratio of height and width of core.

$$\text{Aspect ratio} = \text{width/height}$$

2. **Core utilization:** - Utilization will define the area occupied by the standard cells, macros, and other cells. If core utilization is 0.8 (80%) that means 80% of the core area is used for placing the standard cells, macros, and other cells, and the remaining 20% is used for routing purposes.

$$\text{core utilization} = (\text{macros area} + \text{std cell area}) / \text{total core area}$$

## 2) I/O Pad placement:

In ASIC design three types of IO Pads. Generally, pad placement and pin placement is done by Top-Level people.

## 3) Voltage area Creation :

Multi voltage design is used to have trade-off between power consumption and performance

In multi-voltage design different blocks operate with different voltages. We use level-shifters when signals crossing one power domain to another.

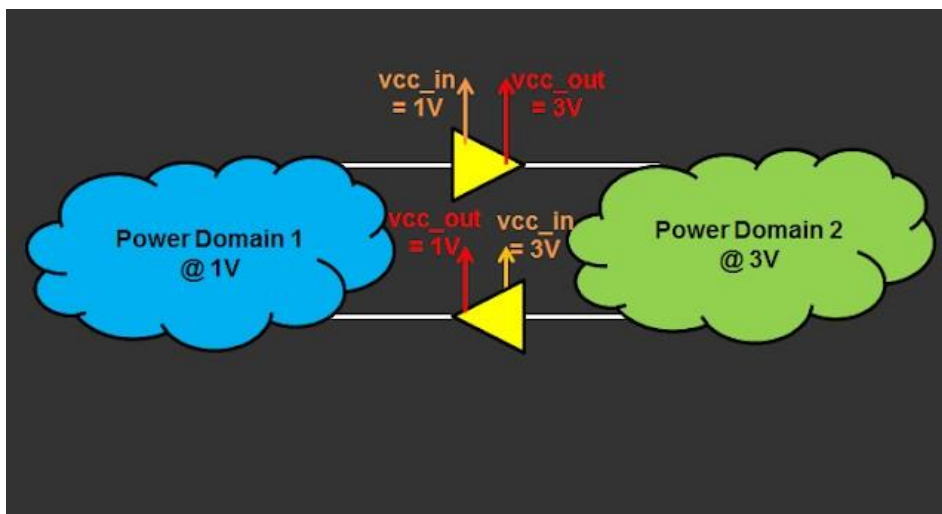


Fig : Different power domain with level shifters

## 4) Macro Placement:

Macros may be memories, analog blocks. Proper placement of macros has a great impact on the quality and performance of the ASIC design. Macro placement can be manual or automatic. Generally manual macro placement is Preferred

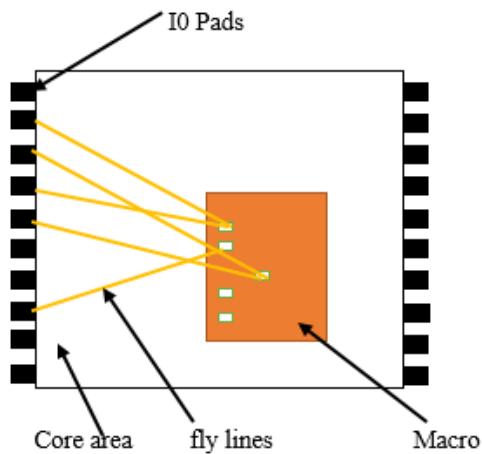
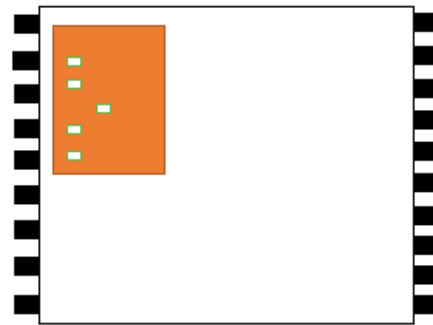
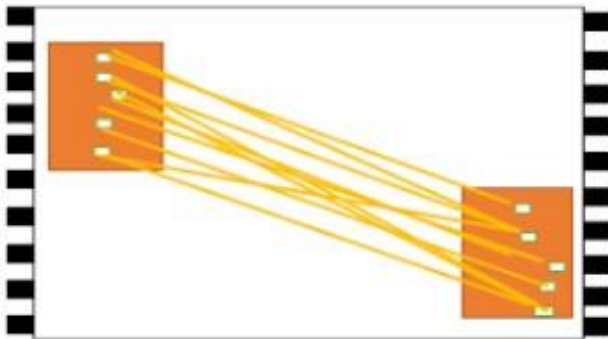
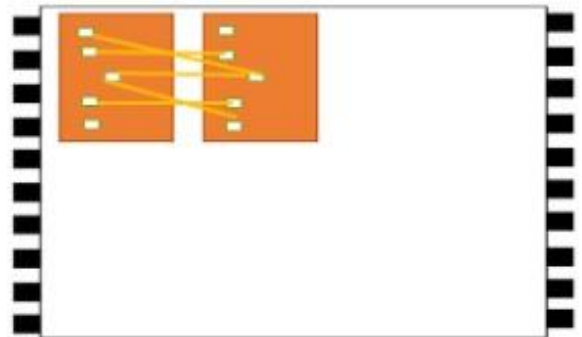
### **Note :**

#### **Types of macros:**

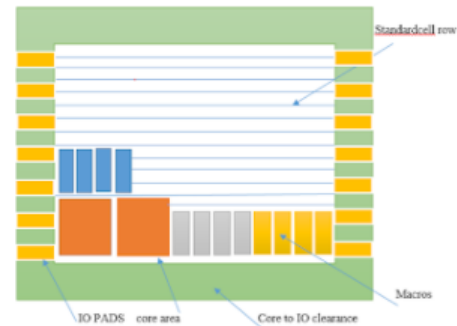
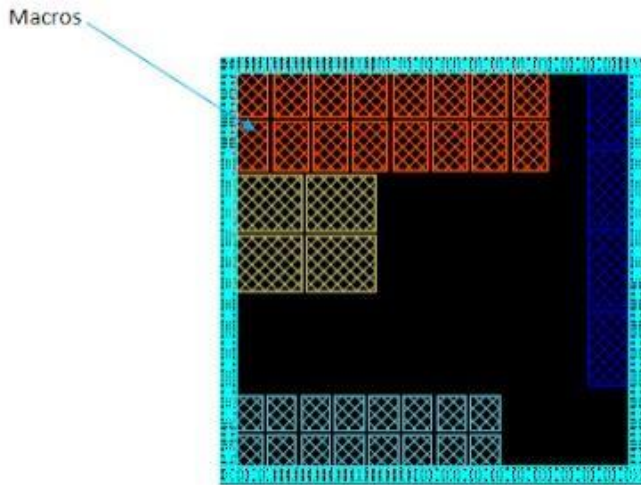
- **Hard macros:** The circuit is fixed. We can't see the functionality information about macros. Only we know the timing information.
- **Soft macros:** The circuit is not fixed and we can see the functionality and which type of gates are using inside it. Also we know the timing information.

**Guidelines to place macros:**

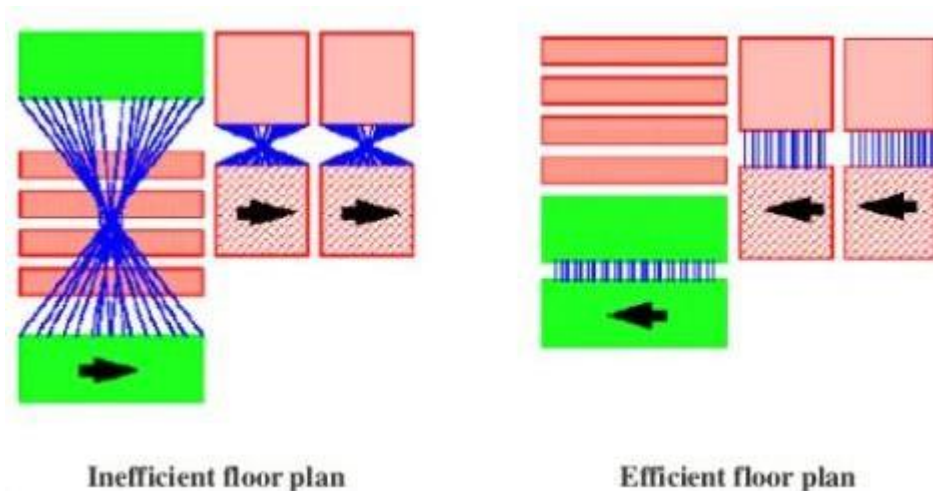
- 1) Placement of macros are the based on the fly-lines (Fly-lines shows the connectivity b/w macro to macro and macro to pins) so we can minimize the interconnect length between IO pins and other cells.

**macros to IO pin:****Fig a: macro to IO ports fly lines****Fig b: macro placed at the core boundary****macros to macros fly lines:****Fig a: macro to macro fly lines****fig b: macros placed near to each other**

- 2) Place the macros around to the boundary of the core, leaving some space between macro to core edge so that during optimization this space will be used for buffer/inverter insertion
- 3) Place the macros of same hierarchy together.

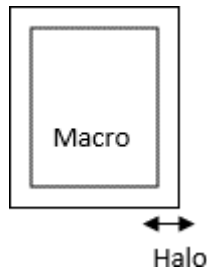


- 4) Keep sufficient channel between macros  
**channel width = (number of pins\*pitch )/ number of layers either horizontal or vertical**  
**Eg.** Let's assume If there are two macros having 50 pins and the pitch values are 0.6 and the total number of horizontal and vertical layers are 12. Means M0 M2 M4 M6 M8 M10 are horizontal layers and M1 M3 M5 M7 M9 M11 are vertical layers.  
 Channel width =  $((50+50)*0.6)/6 = 10$
- 5) Avoid crisscross connection of macro placement.



- 6) Apply keep-out margin/Halo around the four sides of macros so no standard cells will not sit near to Macro pins. This technique avoids the congestion.





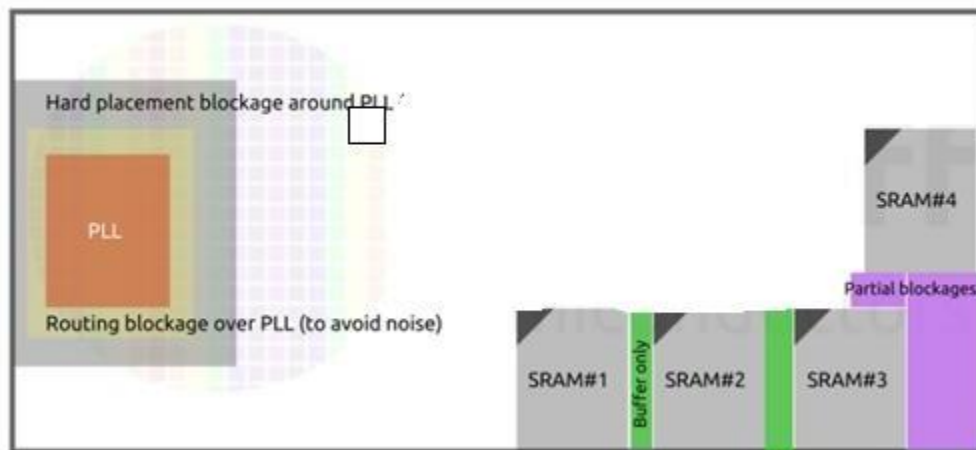
ICC command for keepout margin:

`create_keepout_margin -outer {10 10 10 10} macro_name -type soft/hard`

## 7) Use Placement Blockages near macros to avoid congestion

**Note : Blockages:** Blockages are the specific location where the placing of cells is blocked. If the macros moved from one place to another place, blockages **will not move**.

Placement & routing blockages



**Blockages are of three types. a) Soft b) Hard c) Partial**

### a) Soft blockages:

- prevents from the placement of std cell and hard macro within the specified area during coarse placement but allows placement of buffer/inv during optimization, legalization and clock tree synthesis.

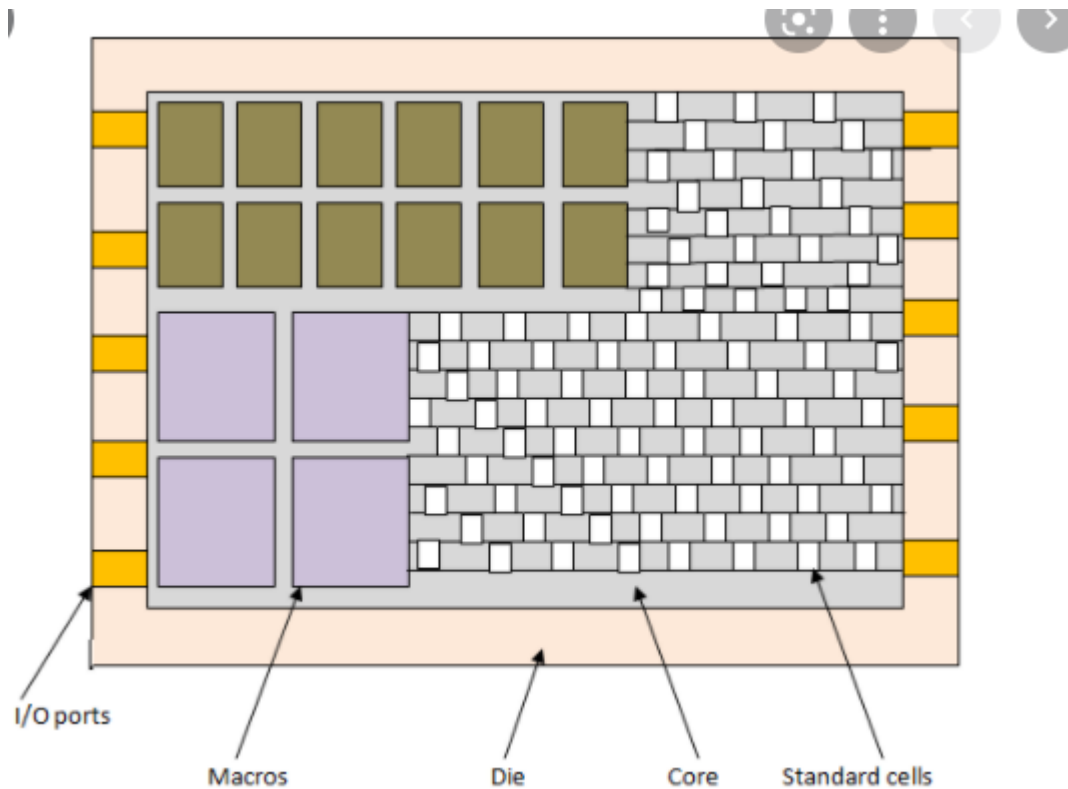
### b) Hard blockages:

- No standard cells, macros and buffer/inv can be placed within the specified area during coarse placement, optimization, and legalization.
- Used to avoid routing congestion at macros corners.

### c) Partial blockages:

- Partial blockages limit the cell density in the specified area.
- Ex: If partial blockage is 40 % , Then in that area the cell density is 60% (blocked % is 40)
- To allow unlimited usage of a partial blockage area specify a blockage percentage to zero.

## 5) The standard cell rows created for standard cell placement.



- The area allotted for the standard cells on the core is divided into rows where standard cells are placed.
- The height of the row is equal to the height of the standard cell and width varies. The height varies according to multiple standard cell row height. there may be double-height cells, triple-height cells, etc.
- The standard cells will sit in the row with proper orientation.
- The rows at the macro placement should be removed.

## 6) Physical-Only Cells :

### a) Tap cells:

- A tap cell is a special nonlogic cell with a well tie, substrate tie, or both to avoid latch-up Problem.
- Tap cells are placed in regular intervals in standard cell row and distance between two tap cells given in the design rule manual.
- Generally, the design rules specify the maximum distance allowed between every transistor in a standard cell and a well or substrate tap.
- Before global placement (during the floorplanning stage), you can insert tap cells in the block to form a two-dimensional array structure to ensure that all standard

cells placed subsequently comply with the maximum diffusion-to-tap distance limit.

**b) Tie –Cells**

- These are special-purpose cells whose output is constant high or low.
- The input needs to be connected to the gate of the transistor and there are only two types of input logic 1 and logic 0, but we do not connect them directly to gate of the transistor as with supply glitches can damage the transistor so we used tie high and tie low cells and output of these cell are connected to the gate of the transistor.
- Tie Cells are inserted during placement stage

**c) Filler Cells**

- To have n-well and Substrate continuity we Use filler cells
- If there is continuity b/w nwell and implant layer it is easier for foundry people to generate them and the creation of a mask is a very costly process so it is better to use only a single mask.
- If nwell is discontinuous the DRC will be flagged to place cells further apart i.e maintain the minimum spacing because there is a **well proximity effect**.
- After routing and after timing sign-off we add Filler cells.

**d) Decap Cells**

- If standard cells and macros are not getting sufficient power due to IR-drop, They may go to metastable state.
- Decap filler cells are small capacitors which are placed between VDD and GND all over the layout when the logic circuit draw a high amount of current, this capacitor provides extra charge to that circuit. when logic circuit not drawing any current, the de-cap is charged up to maximum capacitance.

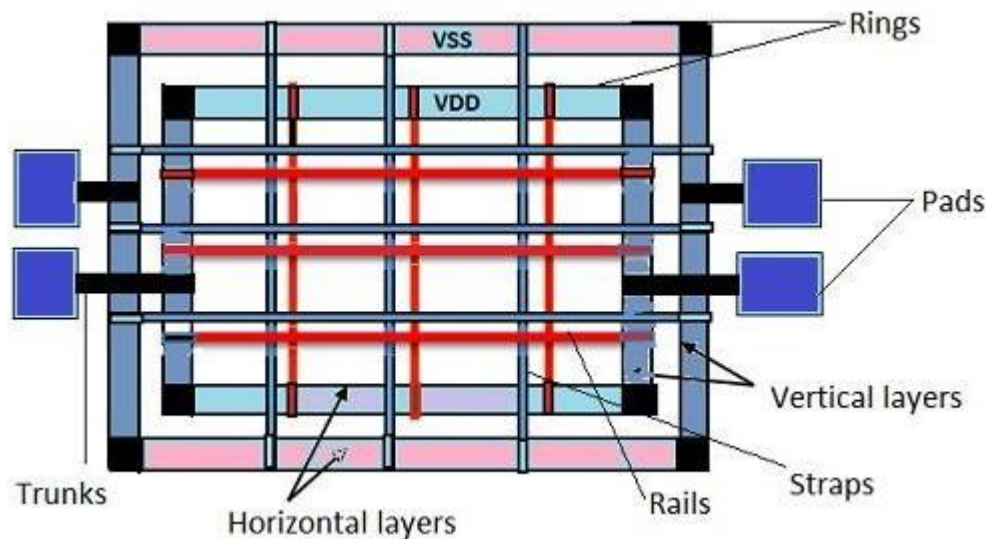
**e) Endcap Cells**

- Before placing the standard cells, we can add boundary cells to the block. Boundary cells consist of end-cap cells, which are added to the ends of the cell rows and around the boundaries of objects such as the core area, hard macros, blockages, and voltage areas, and corner cells
- End-cap Cells are used to protect the gate of a standard cell placed near the boundary from damage during manufacturing and to avoid the base layer DRC (Nwell and Implant layer) at the boundary.

## 7) Power planning (pre routing):

### What is power planning?

- In power planning power grid network is created to distribute power to each part of the design equally.
- It is used to provide power to macros and standard cells within the IR-drop limit.
- Power planning can be done manually as well as automatically through the tool.



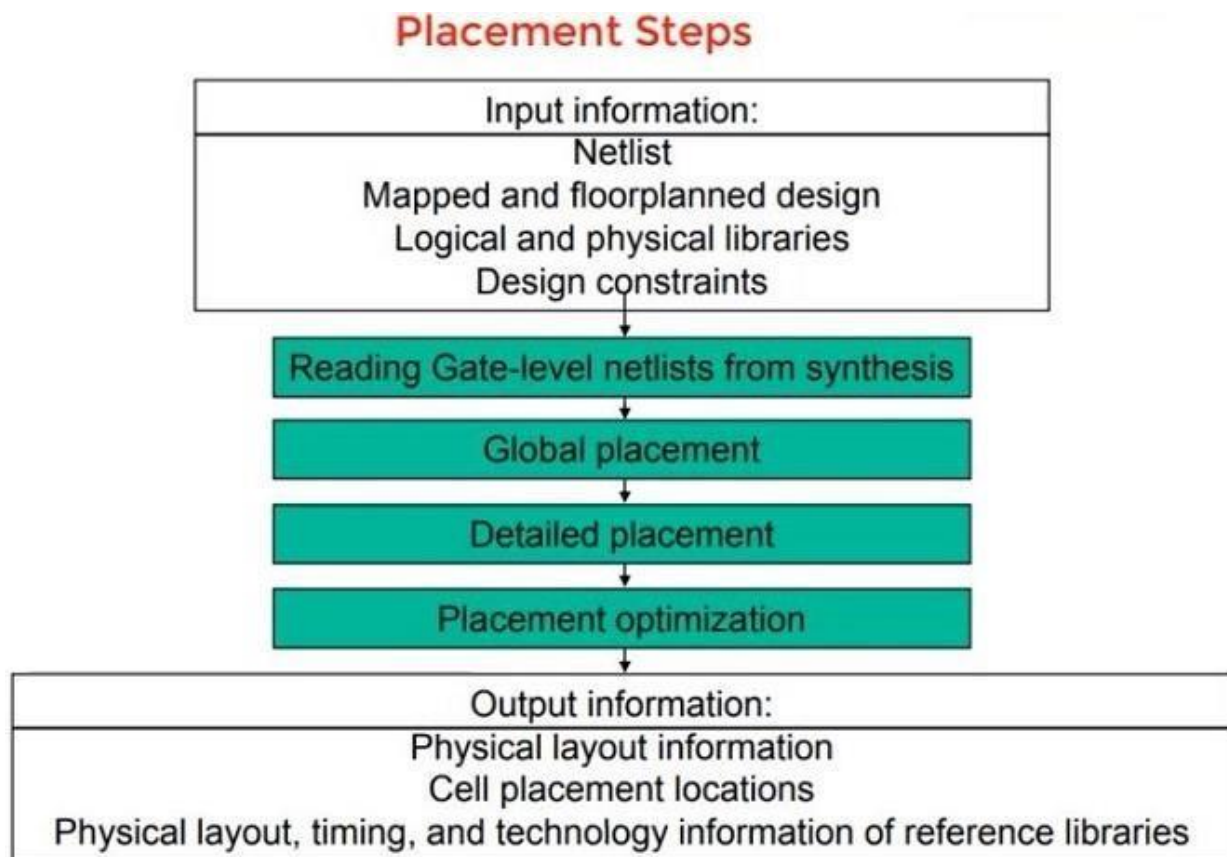
- **Rings:** It Carries VDD and VSS around the chip
- **Stripes:** It Carries VDD and VSS from Rings across the chip
- **Rails:** It connects VDD and VSS to the standard cell VDD and VSS.
- **Trunk:** The connection between Pad and Ring
- **Pad:** Interface from IC to the outside world.

## Placement

Placement is the process of placing the standard cells inside the core boundary in an optimal location. The tool tries to place the standard cell in such a way that the design should have minimal congestions and the best timing. Every PnR tool provides various commands/switches so that users can optimize the design in a better way in terms of timing, congestion, area, and power as per their requirements.

### Steps in Placement :

- i) **Pre-placement Stage**
- ii) **Initial Placement / Global Placement / Coarse Placement**
- iii) **Legalization**
- iv) **Tie Cell insertion**
- v) **Scan-Chain Reorder**
- vi) **HFNS (High Fanout Net Synthesis)**
- vii) **Iteration for Congestion, Timing, DRV, and Power Optimization**



- i) **Pre-placement Stage:**
  - Perform checks on port placement
  - Perform checks on end-cap cells and tap-cells placement
  - Perform Power planning check
  - Perform checks on macro-placement and use blockages at required places

- See that all macros are fixed and all macros have sufficient keep-out margin
- Verify whether all required input files are sourced
- Set local density limit (G-cell density)
- Make clock ideal and use path-grouping if required

**ii) Initial Placement / Global Placement / Course Placement**

- During the coarse placement, the tool determines an approximate location for each cell according to the timing, congestion and multi-voltage constraints.
- The placed cells don't fall on the placement grid and might overlap each other. Large cells like RAM and IP blocks act as placement blockages for standard cells.
- Coarse placement is fast and sufficiently accurate for initial timing and congestion analysis.

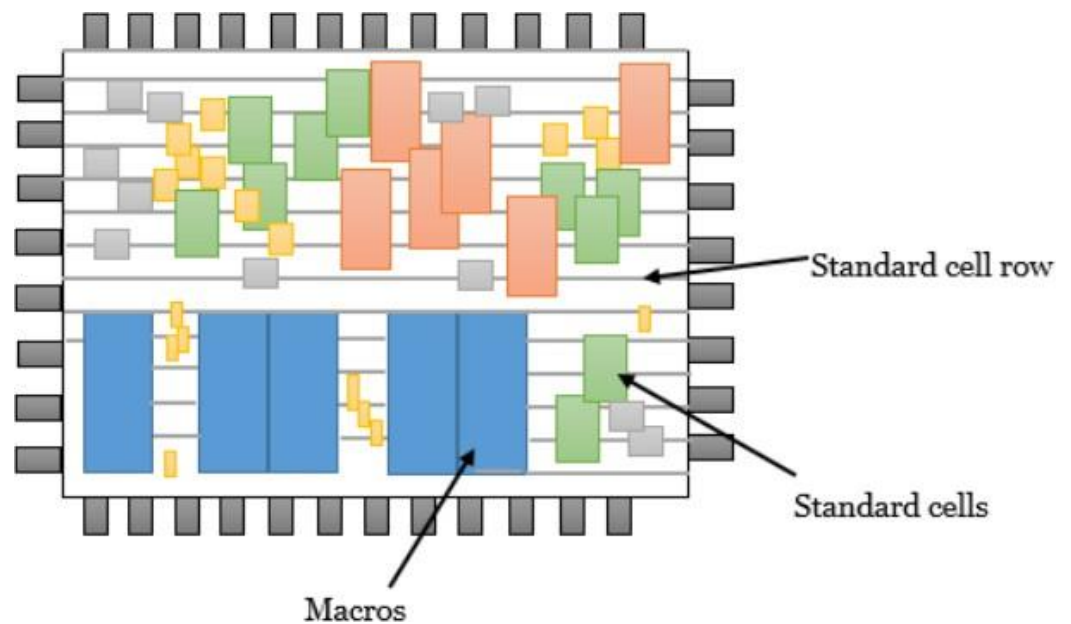


Fig : Coarse placement of standard cells

**iii) Legalization:**

- During legalization, the tool moves the cells to legal locations on the placement grid and eliminate any overlap between cells.
- These small changes to cell location cause the lengths of the wire connections to change, possibly causing new timing violations.
- Such violations can often be fixed by incremental optimization, for example: by resizing the driving cells.

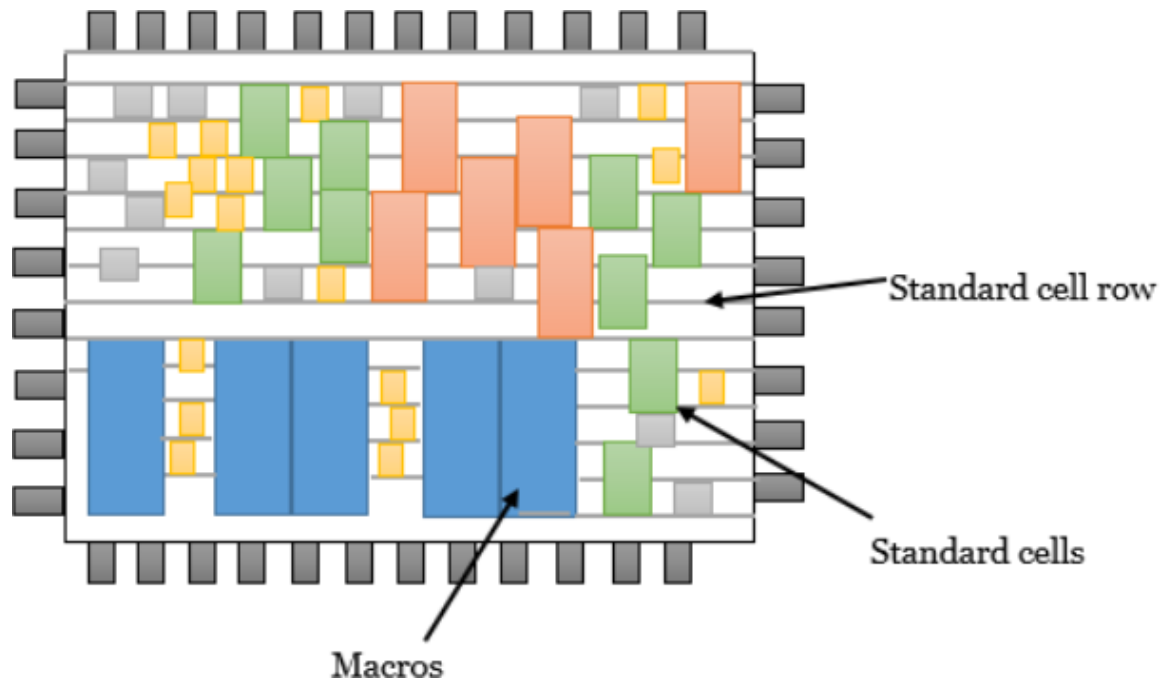


Fig : Legalization of standard cells

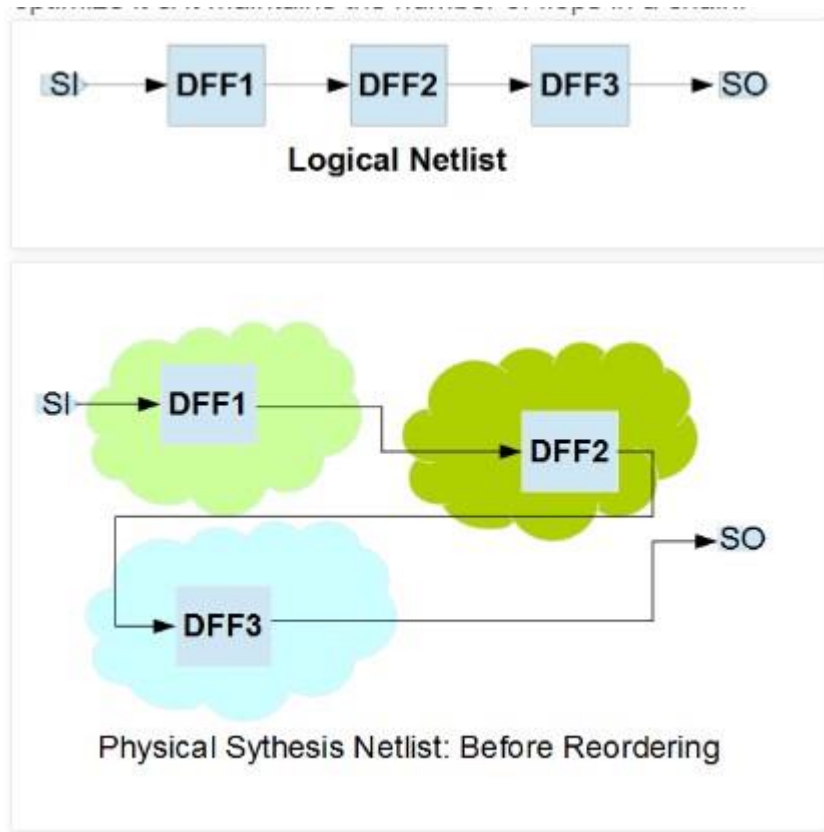
**iv) Tie Cell insertion**

Sometimes in the netlist some unused inputs are tied to VDD/VSS (logic1/logic0). It is not recommended to connect a gate directly to the power network, so we connect gate to TIEHI or TIELO cells.

Tie cells are single pin cells that effectively ties the pin it connects high or low voltage. Placement tool also does Tie-cell optimization, which places Tie cell near to parent cell.

**viii) Scan-Chain Reorder**

- If block contains scan chains by default Placement and CTS tools perform DFT optimization.
- During initial placement, the tool focuses on the QOR for the function nets by ignoring the scan chains. After initial placement, the tool further improves the QOR by repartitioning and reordering the scan chains based on the initial placement.
- Scan chains reordering reduces wire length so timing will improve.
- Scan chains reordering minimize congestions and improves routability



v) **High fanout Net Synthesis (HFNS)**

- The process of buffering the high fan-out to reduce the fanout load is called as High fanout net Synthesis. because if design has too many loads then it affects delay and transition time.
- High fanout nets are mainly reset, preset, scan enable etc. these nets are not synthesized in the synthesis stage, also make sure you set an appropriate fan-out limit for your library

vi) **Placement Optimization stage:**

- In this stage tool tries to optimize placement to reduce congestion , improve timing and to fix timing DRVs
- Tool optimizes timing DRVs and setup violation by different methods like
  - Cell sizing
  - Vt swapping
  - Buffering
  - Cloning
  - Pin-swapping
  - Logical restructuring

**Constraints for placement :**

Placement constraints provide guidance during placement and placement optimization and legalization so that congestion and timing violations will be reduced.



- i) Placement blockages
  - It is the area where the cells are restricted during placement, optimization and legalization.
  - It can be hard soft and partial blockage.
- ii) Placement bounds
  - It is a constraint that controls the placement of groups of leaf cells and hierarchical cells.
  - It allows you to group cells to minimize wire length and place the cells at most appropriate locations.
  - When our timing is critical during placement then we create bounds in that area where two communicating cells are sitting far from another. It is a fixed region in which we place a set of cells.

Types of bounds:

- Soft move bound
- Hard move bound
- Exclusive move bound

**Soft move bound:**

In this tool tries to place the cells in the move bound within a specified region, however, there is no guarantee that the cells are placed inside the bounds.

**Hard move bound:**

In this tool must place the cells in the move bound within a specified region.

**Exclusive move bound:**

In this tool tries to place the cells in the group bound within a floating region, however, there is no guarantee that the cells are placed inside the bounds

**iii) Density controls:**

It means how the density of cells can be packed. We can control the overall placement density for the block or the cell density for specific regions. To control the cell density for specific regions we can also use partial placement blockages.

**iv) Constraints for Max\_fanout , max\_transition and max\_capacitance :**

We can specify the Max fanout , max\_transistion and max\_capacitance constraints to placement tools , so that tool performs placement to meet given constraints

**Checks after Placement:**

i) Analyze and fix congestion

Congestion: It is a difference between available routing track and required routing track

To fix congestion

- a) Apply soft blockage between macros
- b) Apply keep-out margin to macros and for cells with more pin count

- c) Apply partial blockage
- ii) Analyze max capacitance and max transition violation  
Fix by using
  - a) Load splitting: Split fanout by using buffers
  - b) Cloning : Split fanout by cloning driver cell
  - c) Increase drive strength of driver
  - d) Split net length by inserting buffer
- iii) Analyze and Fix Setup violation .
  - a) Upsize cells in combinational path
  - b) Vt swapping : Swap cell with lower Vt
  - c) Path grouping method: Assign weightage for most violating path

# **STATIC TIMING ANALYSIS**

## **Difference between DTA & STA**

<b>Dynamic timing analysis [ DTA ]</b>	<b>Static timing analysis [ STA ]</b>
Verifies functionality of the design by applying input vectors and checking for correct output vectors	Checks static delay requirements of the circuit without any input or output vectors, so analysis times are relatively short and STA does not check for logical correctness of the design
Quality increases with the increase of input test vectors	Clock related all information has to be fed to the design in the form of constraints and the correctness of the constraints decides the quality
Increased test vectors increase simulation time	Timing can be analysed for worst and best cases simultaneously and also all timing paths are considered
Can be used for synchronous as well as asynchronous designs	Not suitable for asynchronous designs
Also best suitable for designs having clocks crossing multiple domains	Not suitable for designs having clocks crossing multiple domains
Computational complexity involved in finding the input patterns/ vectors that produces maximum delay at the output	Has more pessimism and thus gives maximum delay of the design and STA and it works with timing models

## **Static Timing Analysis**

- Effective methodology for verifying the timing characteristics of a design without the use of test vectors
- STA can be done only for register-transfer-logic (RTL) designs
- Functionality of the design is subjected to STA
- STA approach typically takes a fraction of the time it takes to run logic simulation
- STA basically method of adding the net delays and cell delays to obtain path delays. Then STA tool analyses all paths from each and every start point to each and every end point and compares it against the constraints (timing specification) that exists for that path.

### **Purpose of STA**

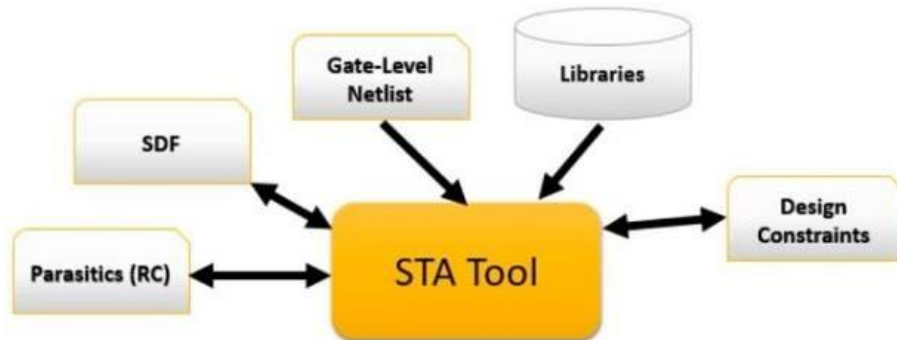
- First, STA calculates the path delays for optimization tools. Then based on the path delays, the optimization tool chooses cells from the timing library to create a circuit that meets your timing requirements.

- Second, STA analyzes the timing of a circuit to verify that the circuit works at the specified frequency.

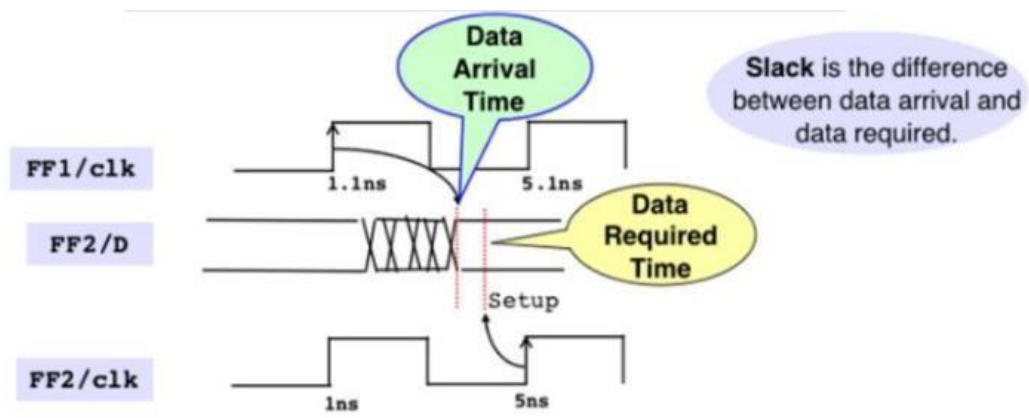
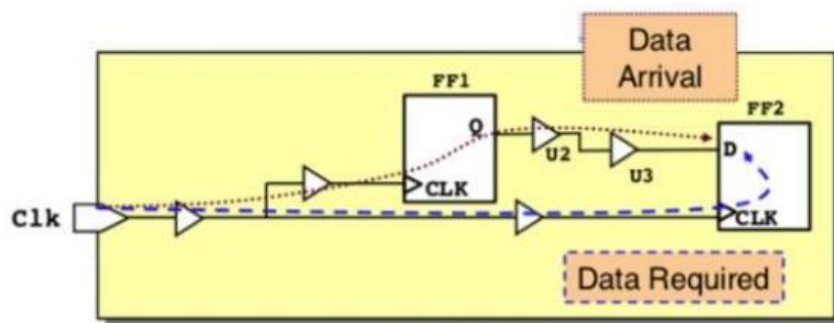
### Steps in STA

- Break the design into sets of timing paths
- Calculate the delay of each path
- Check all path delays to see if the given timing constraints are met

### STA (input & output)



### Timing report



header	Startpoint: modem/qr_tmp (rising edge-triggered flip-flop clocked by Sysclk altpll clk[2])		
	Endpoint: modem/qr (rising edge-triggered flip-flop clocked by Sysclk altpll clk[2])		
	Path Group: Sysclk altpll clk[2]		
	Path Type: max		
Data arrival	Point	Incr	Path
	-----	-----	-----
	clock Sysclk altpll clk[2] (rise edge)	0.000	0.000
	clock network delay (propagated)	0.204	0.204
	modem/qr_tmp/CLK (DFF_D1_CLK1_NCLR1_CKEN1_RSCN1_SCIN1)	0.000	0.204 r
	modem/qr_tmp/Q (DFF_D1_CLK1_NCLR1_CKEN1_RSCN1_SCIN1)	0.146 &	0.350 f
	modem/qr_tmp_ASTfhinst7779/OUT (BUF_D3)	0.073 &	0.423 f
	lcell_comb6052/OUT (BUF_D6)	0.166 &	0.589 f
Data required	modem/qr/D (DFF_D1_CLK1_NCLR1_RSCN1_SCIN1)	0.026 &	0.614 f
	data arrival time		0.614
	-----	-----	-----
	clock Sysclk altpll clk[2] (rise edge)	6.510	6.510
	clock network delay (propagated)	0.321	6.831
	clock reconvergence pessimism	0.003	6.835
	inter-clock uncertainty	-0.160	6.675
	modem/qr/CLK (DFF_D1_CLK1_NCLR1_RSCN1_SCIN1)		6.675 r
slack	library setup time	-0.340	6.334
	data required time		6.334
	-----	-----	-----
	data required time		6.334
slack	data arrival time		-0.614
	-----	-----	-----
	slack (MET)		5.720

### Header

- It consists of start point (FF1) and end point (FF2)
- Path group which tells for which timing path group it belongs
- Path type: here it is max which states setup and if it was min then it is hold

### Data arrival section

- Reports the total time taken to arrive at D pin of flip flop2.

### Data required section

- Reports the total time taken to arrive the clock pin of FF1 – setup time of FF2

### Slack

- Timing difference between required and arrival time i.e., RT-AT

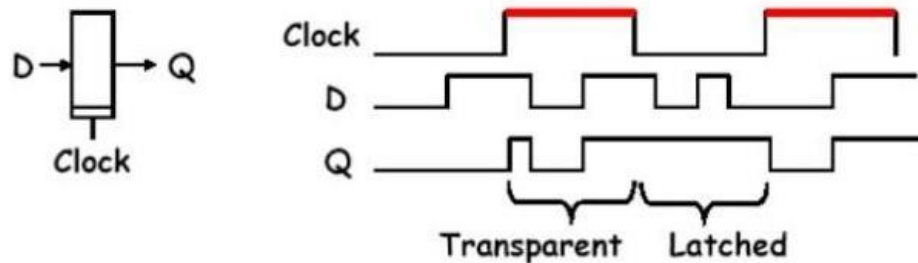
Typical symbols which can be seen in prime time report:

- “&” after an incremental delay numbers shows that the delay number is calculated with resistor-capacitor (RC) network-back-annotation.
- “\*” for standard delay format (SDF) back-annotation
- “+” for lumped RC
- “H” for hybrid annotation
- “r” in the path column for the rising edge of the signal
- “f” in the path column for the falling edge of the signal
- Most timing reports use ns for the time unit.

## **Clocked storage elements**

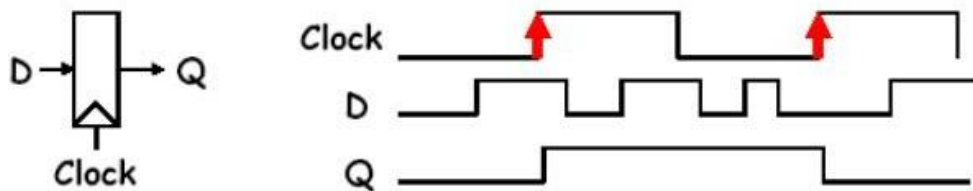
### **Transparent latch, Level sensitive**

- Data passes through latch when clock high, latched when clock is low



### **D-type register of flip-flop, edge triggered**

- Data captured on rising edge of clock, held for rest of the cycle



## **Delays**

- Time taken by a signal to propagate through a cell or net
- Actual path delay is sum of net and cell delays along the timing path
- Cell delay is a function of input transition time (slew rate), total output load (net cap + sum of attached pin caps) and process parameters (temperature, power level)

### **Intrinsic delay**

- Internal to the cell from input pin to output pin caused by internal capacitance

### **Propagation delay**

- Delay by a cell for a change of input signal to result a change at output signal as a function of input slew and output load
- Propagation delay can be low to high (tPLH) and high to low (tPHL)
- Maximum propagation delay (clock to Q) is considered for setup check

### **Contamination delay**

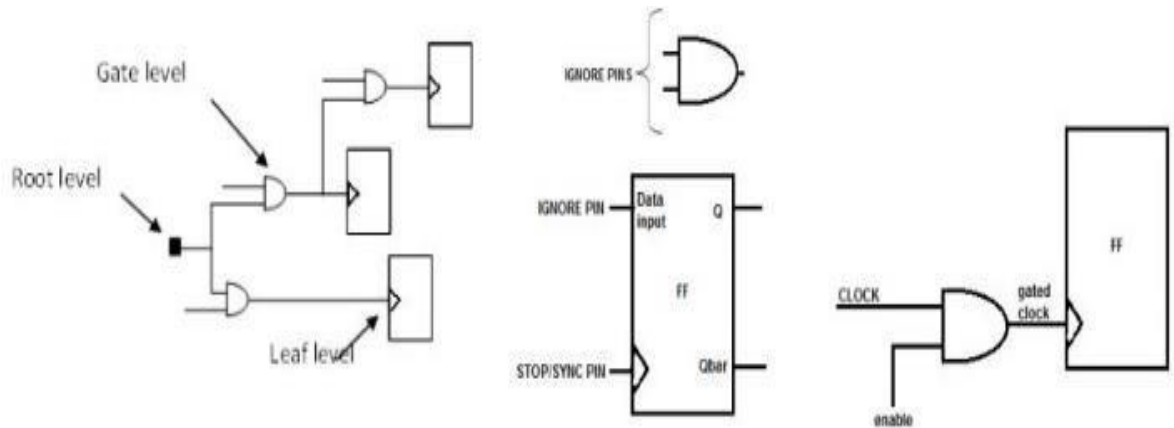
- Best case delay from valid input to output

- Minimum propagation delay ( clock to Q) which is called contamination delay is considered for hold check

### Net delay

- Total time for charging/ discharging all the parasitic present in the given net

### Pins related to clock



### Start/ source/ root pins -

- Source pin of a clock

### Stop/ sink/leaf pins –

- All clock pins of flip flops
- Clock won't propagate after this pin

### Through pin –

- To make a clock pin of a flop not a CTS leaf pin

### Preserved pin –

- If we need to preserve a pin with respect to location etc.

### Exclude / implicit pins –

- All non-clock pins (D pin of flip flops or combo logic inputs)
- Not considered entry pin of the hard macro

### Float pins (implicit stop / macro model) –

- Same as stop/sink pin but internal clock latency of it is considered for clock tree
- Its actually entry pin of the hard macro

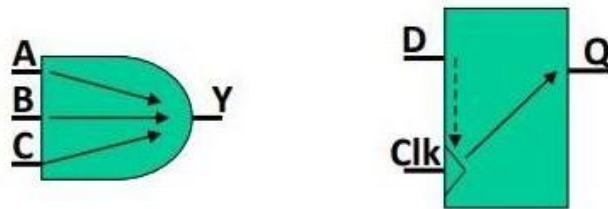
### Explicit sync (stop) pin –

- Input of combinational logic while considering clock tree
- Important while considering clock gating

### Explicit exclude (ignore) sync pin –

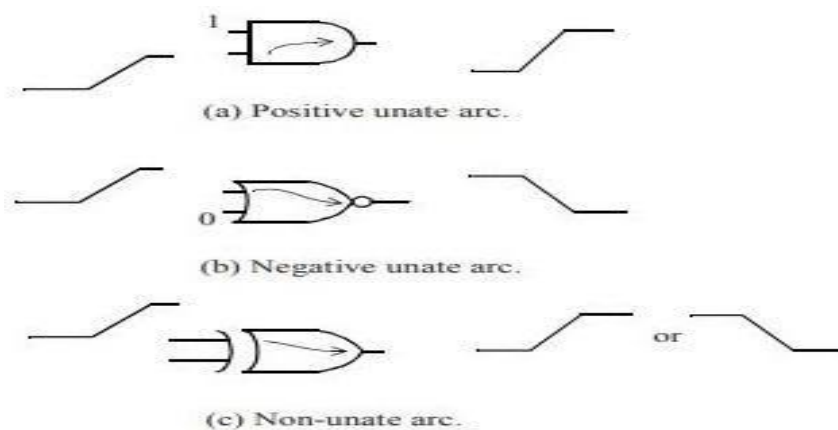
- Clock pin of flop is not considering as sync/ stop pin
- This pin is due to clock gating concept
- In clock gating the signal will be given to AND gate

## 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/ constraints arc)
- Latches have 2 timing arcs :
  - Clock pin to output Q pin, when D is stable
  - Data D pin to Q output Q pin when D changes

## Timing Unate





**Positive unate :**

If a rising transition on an input causes the output to rise (or not to change) and a falling transition on an input causes the output to fall (or not to change). For example: the timing arcs for AND and OR type cells are positive unate.

**Negative unate :**

Timing arc is one where a rising transition on an input causes the output to have a falling transition and a falling transition on an input causes the output to have a rising transition. For example : the timing arcs for NAND and NOR type cells are negative unate.

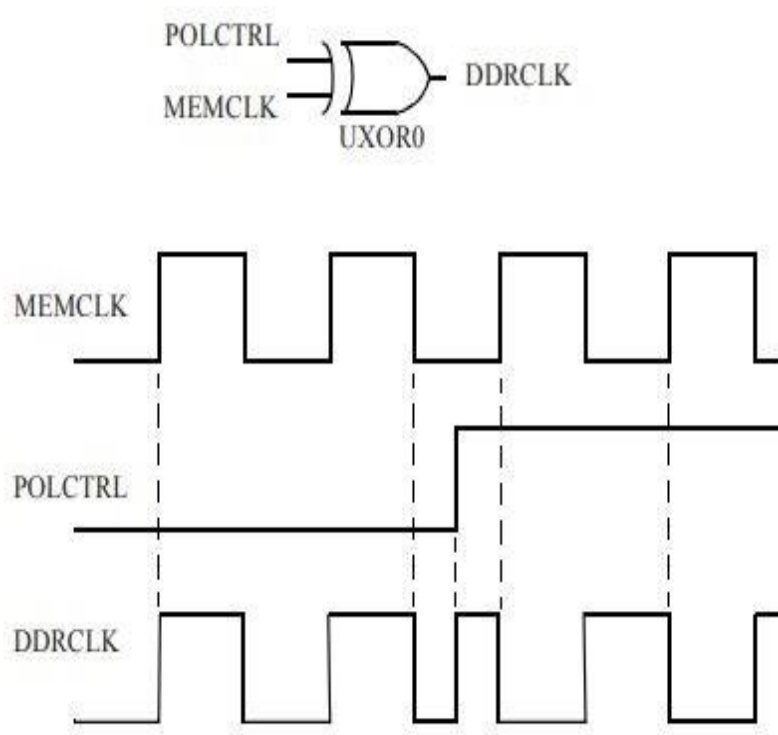
**Non unate :**

In a non-unate timing arcs, the output transition cannot be determined solely from the direction of change of an input but also depends upon the state of the other inputs. For example : the timing arc in an XOR cell are non-unate.

Unateness is important for timing as it specifies how the edges can propagate through a cell and how they appear at the output of the cell.

One can take advantage of the non-unateness property of a timing arc, such as when an XOR cell is used, to invert the polarity of a clock.

For example : if input POLCTRL is a logic-0, the clock DDRCLK on output of the cell UXOR0 has the same polarity as the input clock MEMCLK. If POLCTRL is a logic-1, the clock on the output of the cell UXOR0 has the opposite polarity as the input clock MEMCLK.



## **Clock definitions in STA**

### **Synchronous clocks :**

- 2 clocks are synchronous with respect to each other
- Timing paths launched by one clock and captured by another

### **Asynchronous clocks :**

- 2 clocks are asynchronous with respect to each other
- If no timing relation, STA can't be applied, so the tool won't check the timing

### **Master clocks :**

- It is a source clock defined at input clock port of design

### **Generated clocks :**

- Clock generated from a master clock as a multiple of the master clock frequency
- The generated clock frequency can be a multiple or can be a divided by master clock frequency

### **Virtual clocks :**

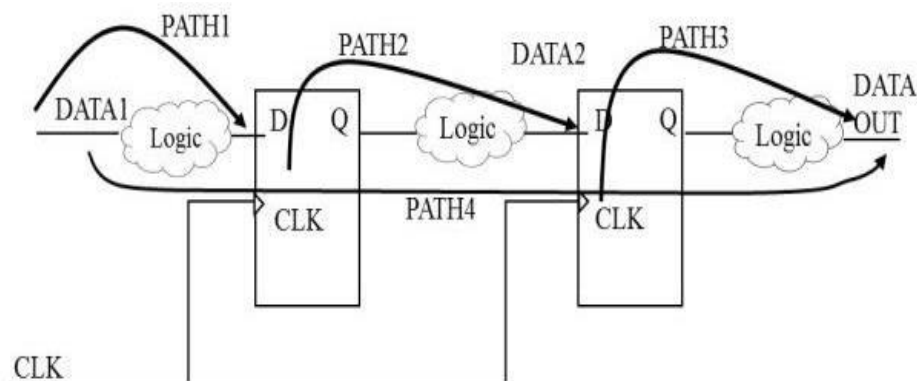
- This Clock is not associated with any pin or port of the design
- Used as a reference in STA to specify input delays and output delays relative to a clock

## **Different terms in STA**

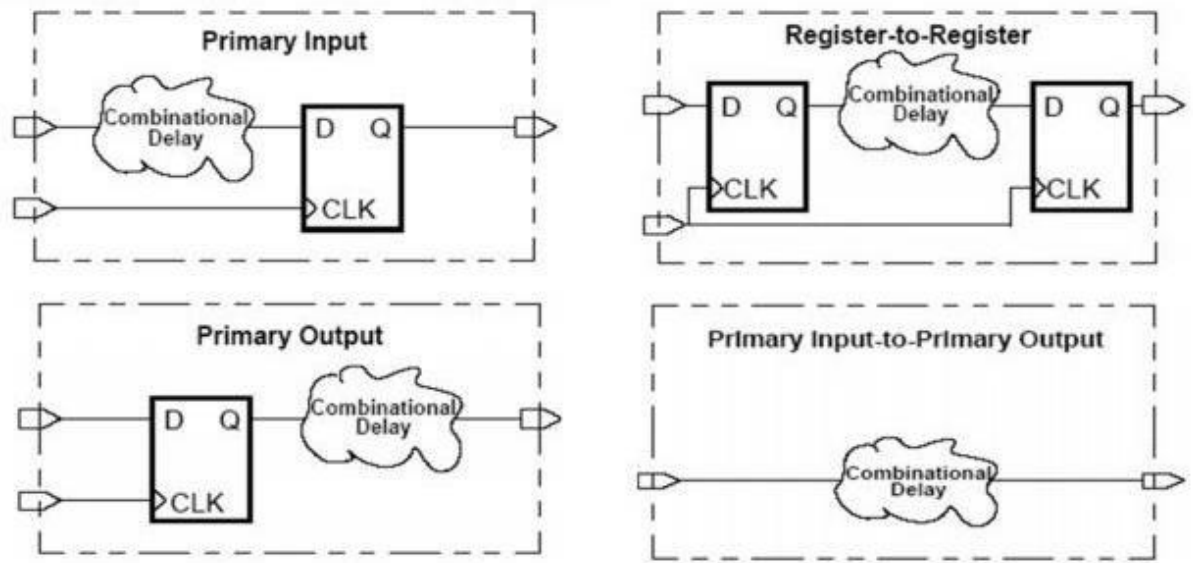
### **Timing paths**

A timing path is a point-to-point path in a design which can propagate data from one flip-flop to another

- Each path has a start point and an end point
- Start point : input ports or clock pins of flip-flops
- End point : output ports or data input pins of flip-flops



## Timing path groups



Timing paths are grouped into path groups by the clock controlling their endpoints

### **Input pin/ port to register**

Delays off chip + combinational logic delays up to the first sequential device

### **Register to register**

Start at a sequential device

CLK-to-Q transition delay + the combinational logic delay + external delay requirements

### **Register to output pin/port**

Delay and timing constraint ( setup and hold) times between sequential devices for synchronous clocks + source and destination clock propagation times

### **Input pin/port to output pin/port**

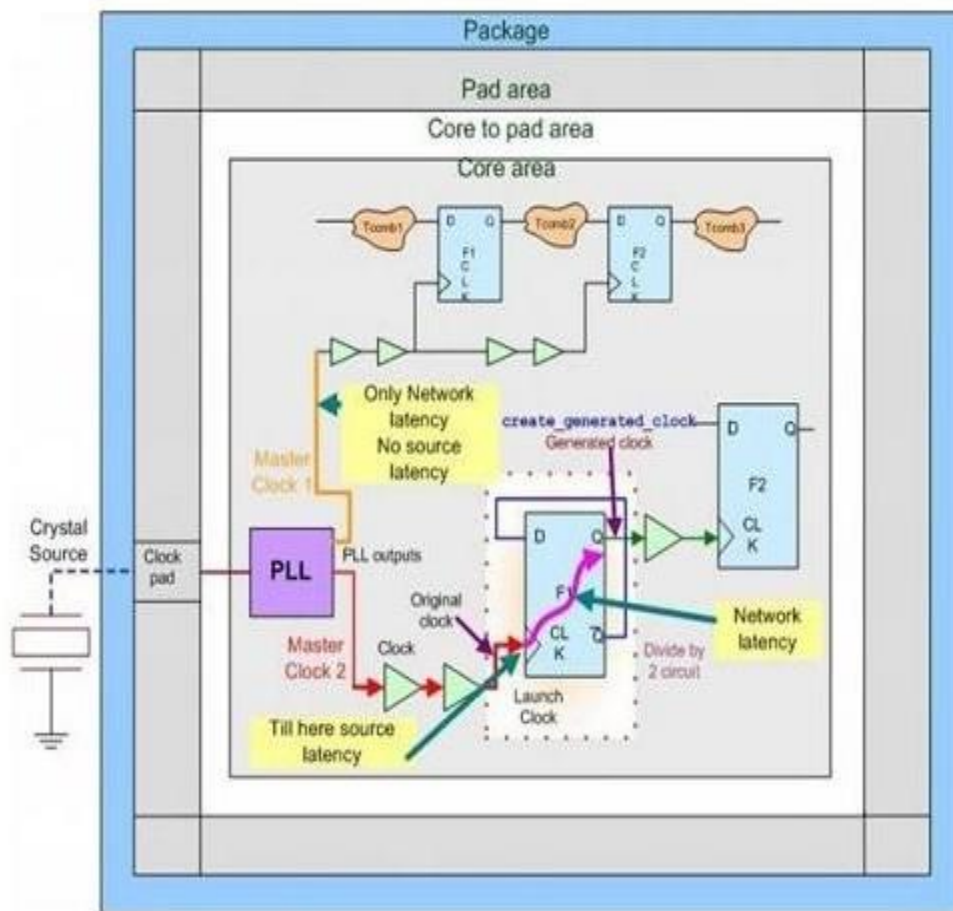
Delays off chip + combinational logic delay +external delay requirements

## Clock latency

- Total time taken by the clock signal to reach the clock input of the register
- Source latency is the time between clock sources to clock definition ports
- Network latency is the time between clock definition ports to clock leaf cells in the design

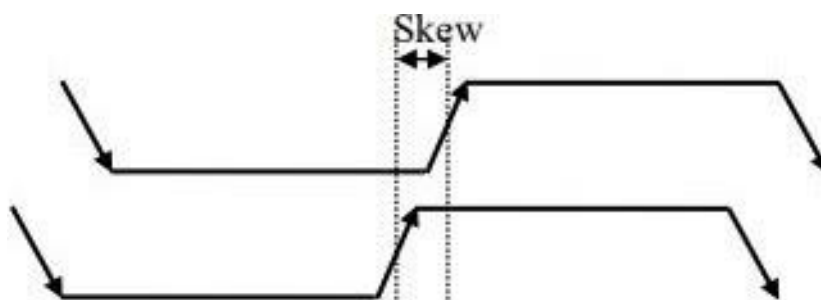
## Insertion delay (ID)

- ID is the clock latency, but after clock tree is synthesized
- ID is the physical delay and clock latency is the virtual delay
- Latency is a target given to the tool through SDC file or clock tree attribute file and insertion delay is the achieved delay after CTS



### Clock uncertainty

- Clock uncertainty is the time difference between the arrivals of clock signals at registers in one clock domain or between domains
- Uncertainties include clock skew, clock jitter and clock margin



### **Clock skew**

Clock skew refers to the absolute time difference in clock signal arrival between two points in the clock network

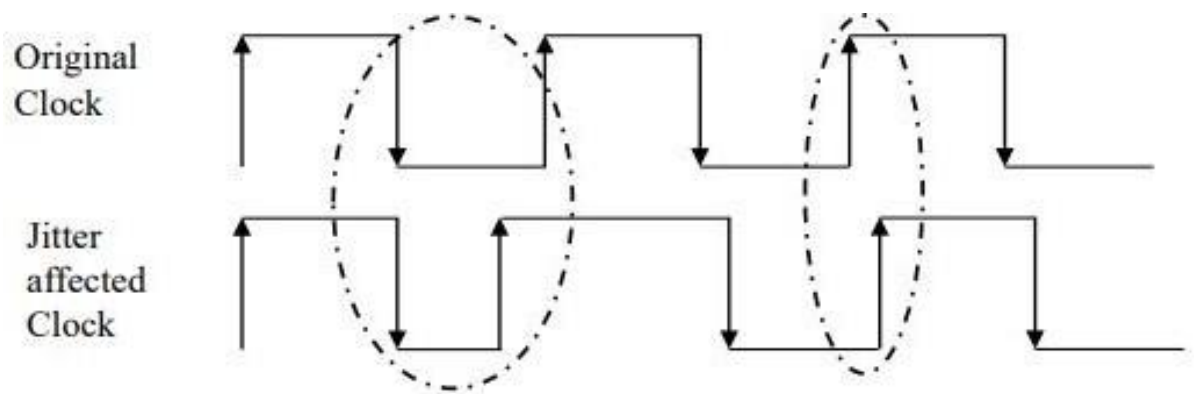
$$T_{skew} = T_{launch\_clock} - T_{capture\_clock}$$

- **Positive clock** : Occurs when the capture clock is late with respect to launch clock

- **Negative clock :** Occurs when the capture clock is early with respect to launch clock
- **Local skew :** is the skew between the clock delays of two flip-flops which are the source and target flop of a path ( source and destination flop)
- **Global skew :** is the difference between the longest and shortest branch of a clock tree (maximum insertion delay – minimum insertion delay)

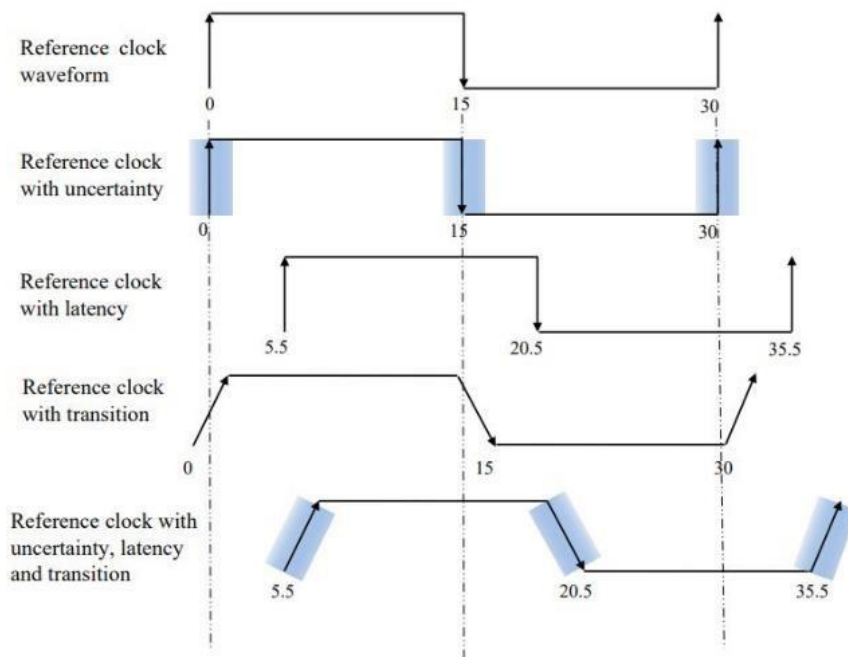
### Clock jitter

- Jitter is the short-term variations of a signal with respect to its ideal position in time
- The two major components of jitter are random jitter and deterministic jitter
- Factors causing jitter includes imperfections in clock oscillator, supply voltage variations, temperature variations, crosstalk



### Glitch

- Unexpected switching of any waveform
- Due to late arrival time of gate and it is for a short period of time
- Cause extra delay and also it can cause extra power from false transitions



### **Pulse width**

- Pulse width is the time between the active and inactive states of the same signal
- Minimum high pulse width is the amount of time after the rising edge of a clock, that the clock signal of a clocked device must remain stable
- Minimum low pulse width is the amount of time after the falling edge of a clock, that the clock signal of a clocked device must remain stable

### **Duty cycle**

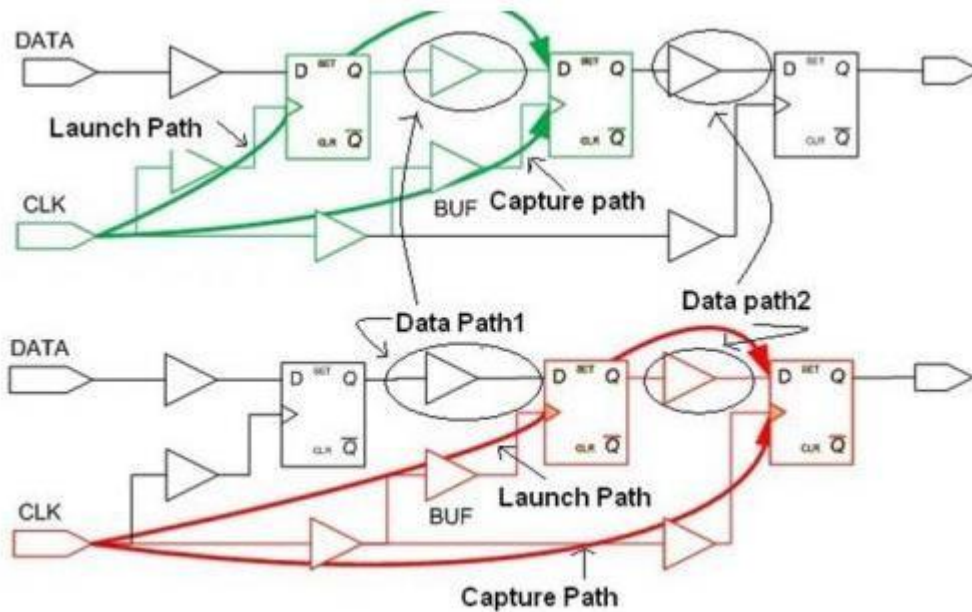
- Percentage of clock period having high pulse
- Typically clock waveforms are of 50% duty cycle

### **Transition/ Skew**

- Time taken by a signal to change the state . (high to low or low to high)
- Rise slew ( $T_r$ ) is called rise time and fall slew ( $T_f$ ) is called fall time
- Minimum / maximum transition is the minimum/maximum slope allowed at leaf pins
- Transition affects power dissipation, latency and pulse width

### **Common Path Pessimism (CPP/CPPR)**

- Same clock path may be a launch path for one data path and can be a capture path for another data path
- While doing OCV derating, same path may get both min/max delay
- But a path can have either as a maximum delay or a minimum delay (or anything in between) but never both delays at the same time
- STA tools will have techniques to remove artificially introduced pessimism between the launch clock path and the capture clock path



### Setup and hold time

**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

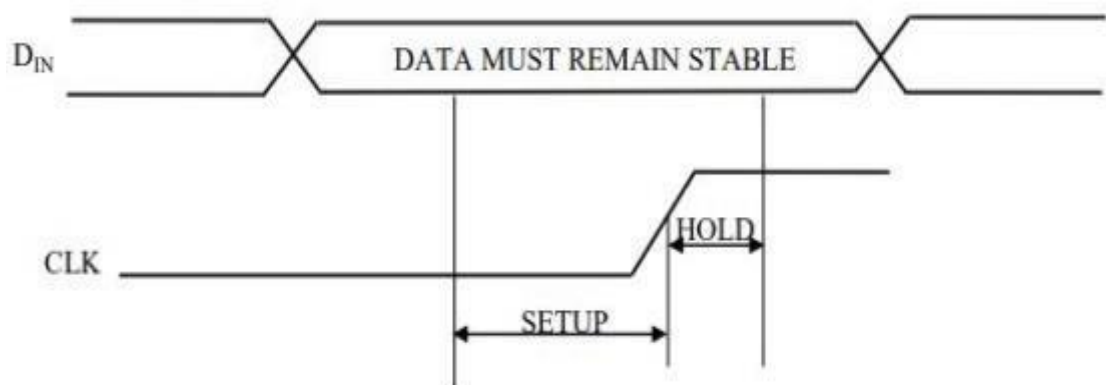
$$T_{launch\_clk} + T_{clkq\_max} + T_{comb\_max} \leq T + T_{capture\_clk} - T_{setup} - T_{un}$$

$$Slack = (T + T_{capture\_clk} - T_{setup} - T_{un}) - (T_{launch\_clk} + T_{clkq\_max} + T_{comb\_max})$$

**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 by the clock

$$T_{launch\_clk} + T_{clkq\_max} + T_{comb\_min} \geq T_{capture\_clk} + T_{hold} + T_{un}$$

$$Slack = (T_{launch\_clk} + T_{clkq\_max} + T_{comb\_min}) - (T_{capture\_clk} + T_{hold} + T_{un})$$



*If Slack is positive timing is met*

*If Slack is Negative Timing is not met*

## **Recovery time and Removal time**

### **Recovery time**

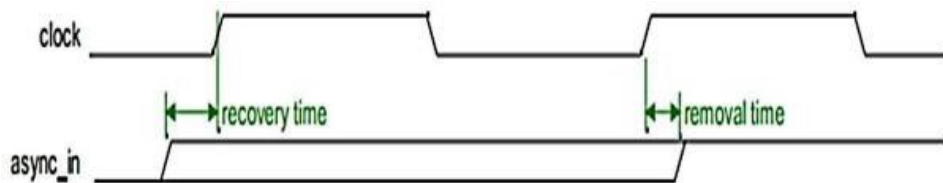
- Recovery time is the minimum time that an asynchronous control input pin must be stable after being de-asserted and before the next clock transition (active edge)

### **Removal time**

- Removal time is the minimum time that an asynchronous control input pin must be stable before being de-asserted and before the previous clock transition (active edge)

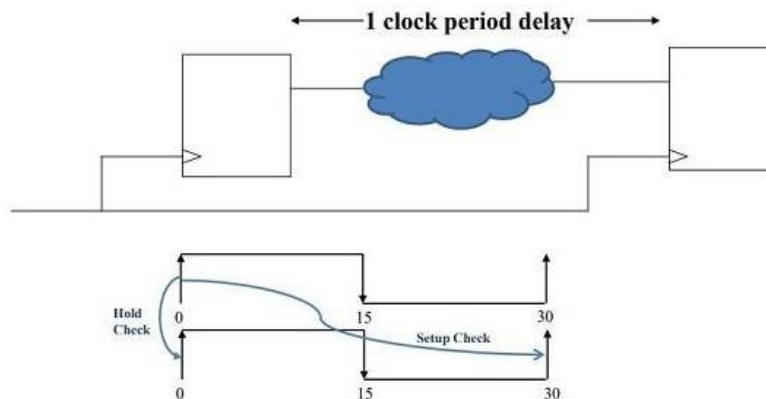
### **Recovery time and Removal time violations**

- This check is to ensure that the asynchronously signal rise/fall edge is not occurring at the clock edge, it should be some time before or after the clock edge
- If that violates, then recovery time and removal time violations
- Although a flip-flop is asynchronously SET or CLEAR, the negation from its RESET state synchronous



## **Single cycle path**

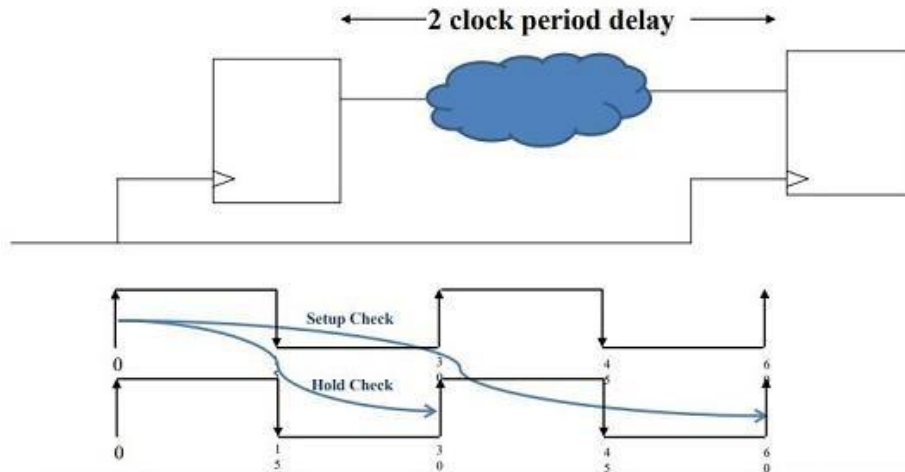
- Timing path that is designed to take only one clock cycle for the data to propagate from the start point to the end point
- Start point and end point are flops clocked by the same clock
- By default tool will consider all timing paths as single cycle paths





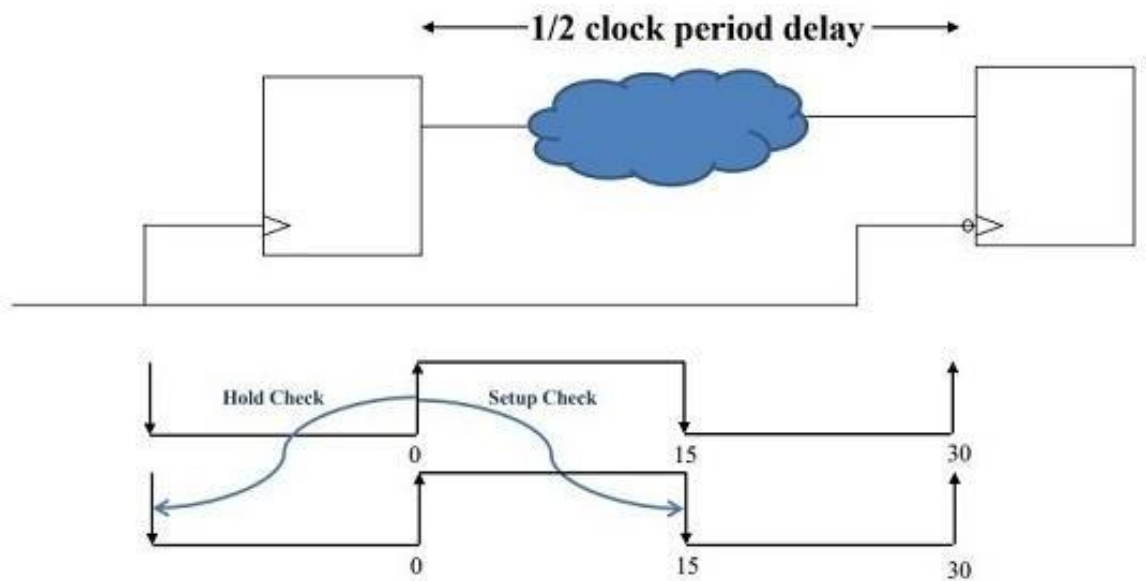
## Multi cycle path

- Timing path that is designed to take more than one clock cycle for the data to propagate from the start point to the end point
- Start point and end point are flops clocked by the same clock
- Need to specify the launch edge and capturing edge in SDC



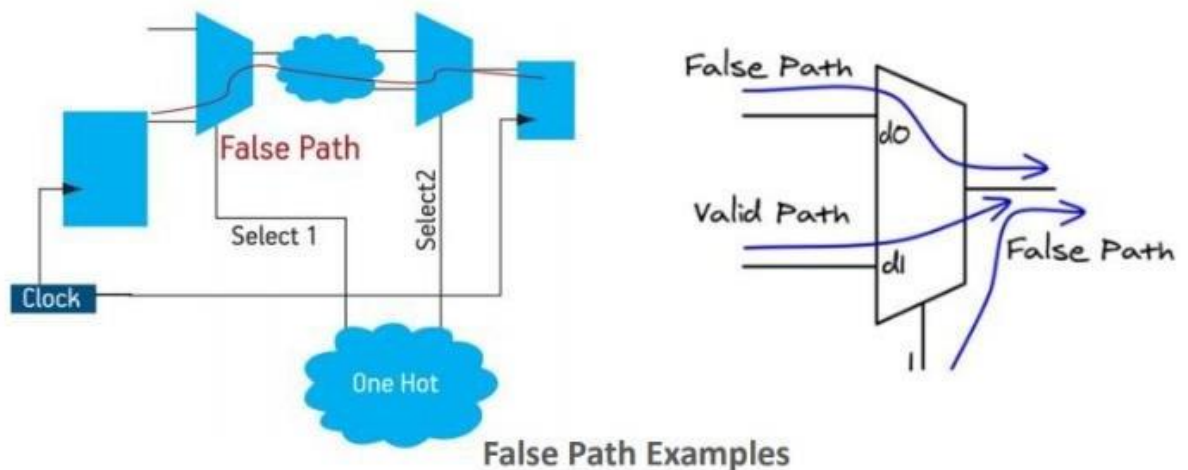
## Half cycle path

- 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
- Start point and end point are flops clocked by the same clock
- No need to specify the launch edge and capturing edge in SDC, since the tool can identify it from the netlist



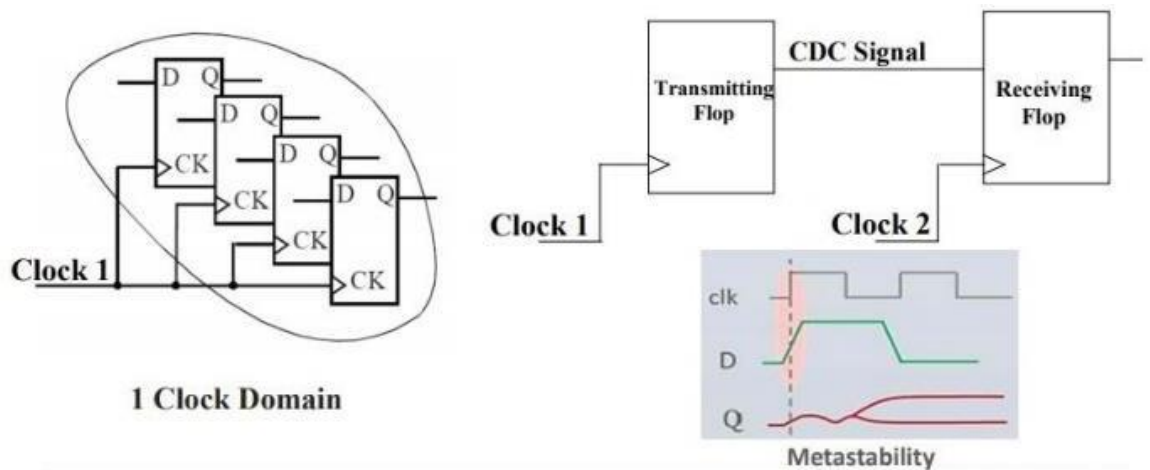
## False path

- Physically exist in the design but are logically/ functionally inactive/ incorrect path
- Means no data is transferred from start point to end point
- The goal in STA is to do timing analysis on all “true” timing paths, so these paths are excluded from timing analysis
- Similarly timing can be disabled for a pin or port or cell where the delay will be computed but won't report it



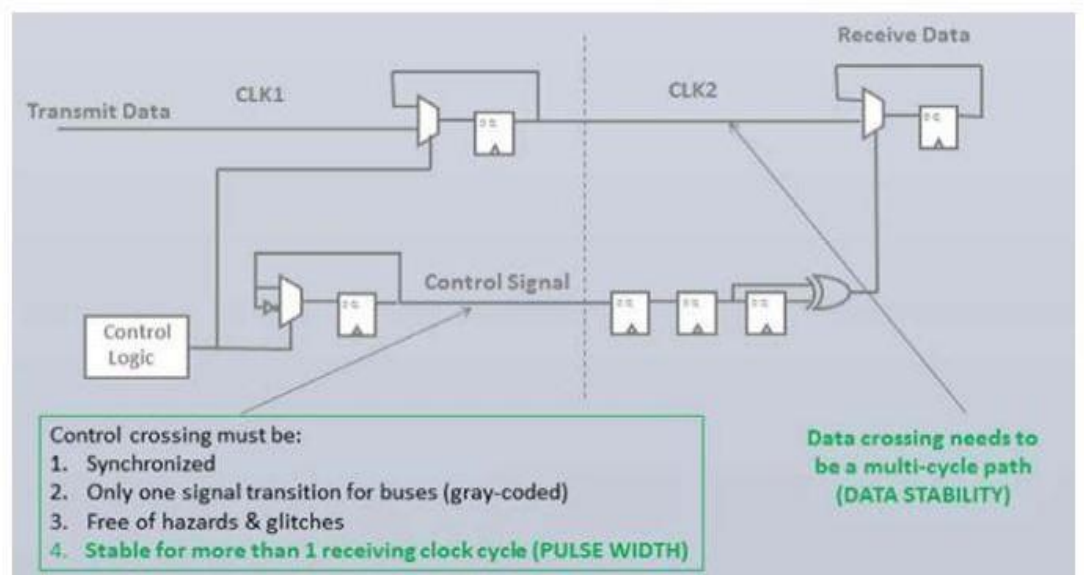
## Clock domain crossing (CDC)

- For design with asynchronous clock domains, the CDC signal violates the setup/ hold window of the receiving clock, resulting in metastability
- Metastability results in unpredictable values and unpredictable delays
- Those clocks has to be balanced together else, due to difference in the latency that may lead to timing violations
- Max delay constraints is used to make CDC paths to get synchronized



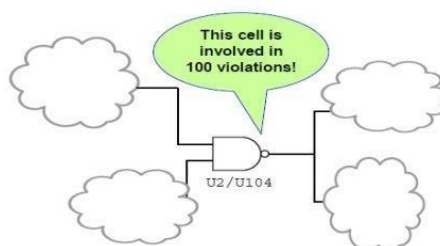
## Clock domain synchronization scheme

- **Pulse width check**
  - The control signals is stable for longer than one receive clock period
  - Ensures that data will not be lost due to inadequate width of the control signal
- **Data stability check**
  - The data updated by the transmit domain cannot be required by the immediately following receive clock edge
  - Ensures that the captured data will not be metastability in the receive domain



## Bottleneck analysis

- Lists the cells causing the timing violations on multiple paths
- By identifying and fixing the violation caused by a bottleneck cell improved timing can be achieved



### **Multi-VT cells (HVT,SVT/RVT,LVT)**

- Different threshold voltages are achieved by implanting dopants in different concentration
- Need multi-VT library
- Sub-threshold leakage varies exponentially with VT compared to the weather dependency of delay over VT
- If the optimization target is power performance, first use the HVT cells library and then try LVT cells
- If the optimization target is to meet timing then first use the LVT cells library and then HVT cells
- If you swap capture flop from SVT to LVT or HVT, there will be very minimal setup/hold impact in most flops, it is of zero impact for hold
- If you swap launch flop from SVT to LVT or HVT, setup will be improve and hold will be impacted correspondingly

#### **High Voltage Threshold (HVT)**

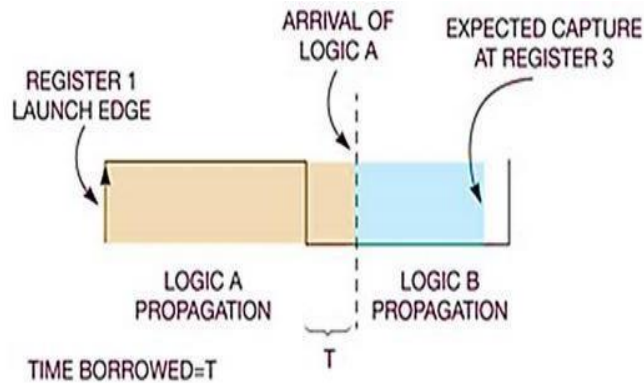
- Use in non-timing critical paths
- Use in power critical paths
- Has low leakage and low speed

#### **Low Voltage Threshold (LVT)**

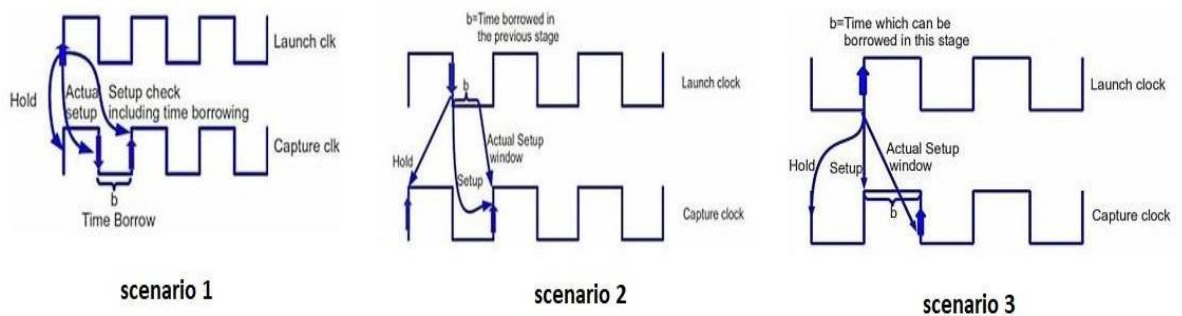
- Use in timing critical paths
- Use in non-power critical paths
- Has high leakage and high speed

### **Time borrowing /stealing**

- Edge-triggered flip-flops change states at the clock edges, whereas latches change states as long as the clock pin is enabled
- In latch based design longer combinational path can be compensated by shorter path delays in the subsequent logic stages
- The technique of borrowing time from the shortest paths of the subsequent logic stages to the longer path is called time borrowing or cycle stealing
- Time borrowing typically only affects setup slack calculation since time borrowing slows data arrival times
- When the clocks of the launching and capturing latches are out of phase, time borrowing is not to happen
- Timing borrowing can be multistage
- Maximum borrow time : clock pulse width minus the library setup time of the latch
- Negative borrow time : arrival time minus the clock edge is negative number, the amount of time borrowing is negative (no borrowing)



### Time borrowing : scenarios



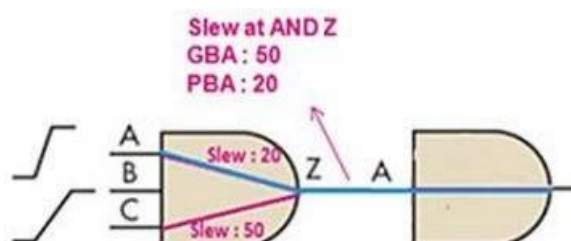
- **Scenario 1 :** when data is launching from a positive edge triggered flip flop and capture is to a negative level sensitive latch
- **Scenario 2 :** when launch is from a negative level sensitive latch and capture is to a positive edge triggered flip flop
- **Scenario 3 :** when launch and capture are from positive level sensitive latches

### Types of STA (PBA, GBA)

**Path based STA (PBA)** – PBA chooses the propagation of the specific path

**Graph based STA (GBA)** – GBA chooses the worst case propagation delay from all inputs through a gate.

In PBA mode run time is more



## **SIGNAL INTEGRITY ISSUES**

### **CROSS-TALK :**

Effects of Crosstalk is due to coupling capacitance between two nets .

There are two effects of cross-talk

1. Crosstalk glitch or crosstalk noise
2. Crosstalk delta delay or crosstalk delay

#### **Crosstalk glitch**

In order to explain the crosstalk glitch, we will consider the following two cases. There might be many more similar cases.

**Case-1: Aggressor net is switching low to high and victim net is at a constant low**

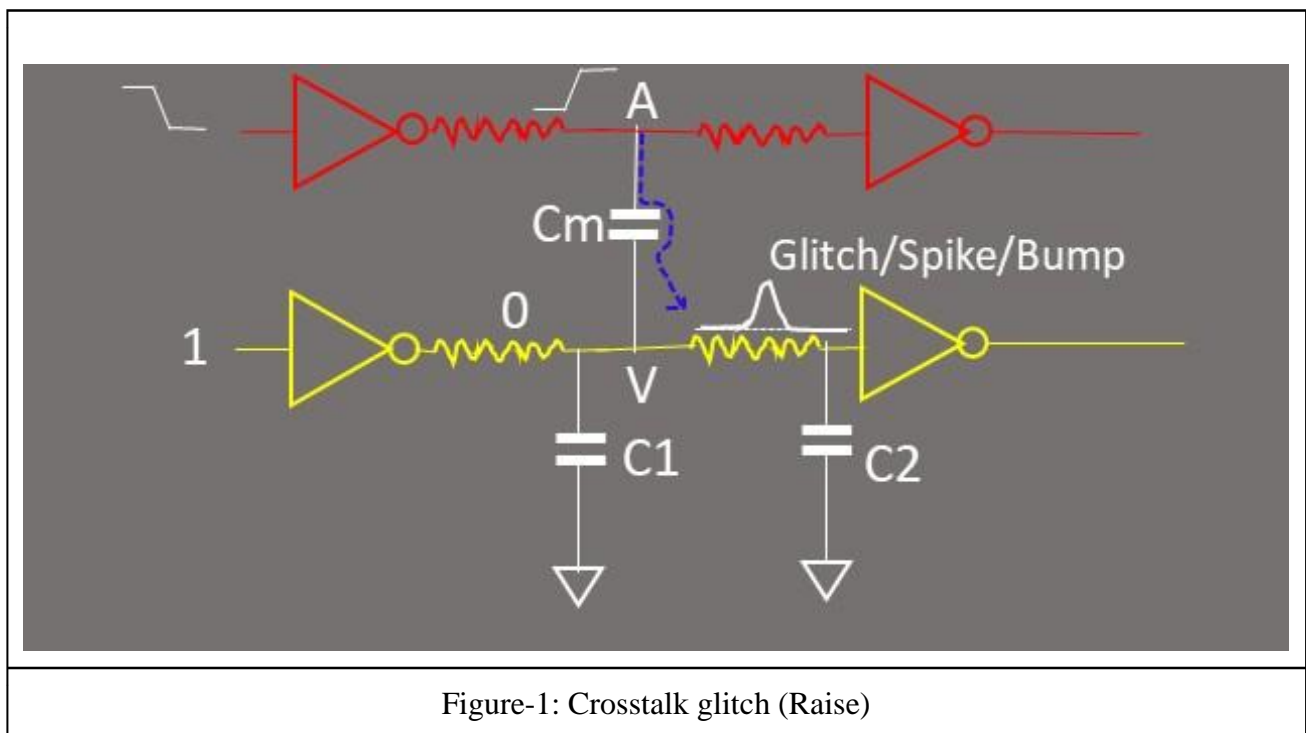


Figure-1: Crosstalk glitch (Raise)

In this case, the aggressor net switches from logic 0 to logic 1 and the victim net is at constant zero as shown in the figure-1. Now consider the node A, node V, Mutual capacitance  $C_m$  and the path from A to V. As node A start switching from low to high, a potential difference across the mutual capacitance gets developed and the mutual capacitor  $C_m$  starts charging. During this event, there is some leakage current which starts flowing from node A to node V through the mutual capacitance  $C_m$  due to the leaky nature of mutual capacitance. This leakage current will raise the potential of node V, which creates a raising spike or raising

glitch on the victim net as shown in figure-1. The magnitude of this voltage or height of the glitch will depend on the various factors which will be discussed later.

So, whenever one net switches from low to high and other neighboring net is supposed to remain constantly low, will get affected by the switching net and have a glitch on it. Now let's discuss case-2 which is similar to case-1.

**Case-2: Aggressor net is switching high to low and victim net is at a constant high**

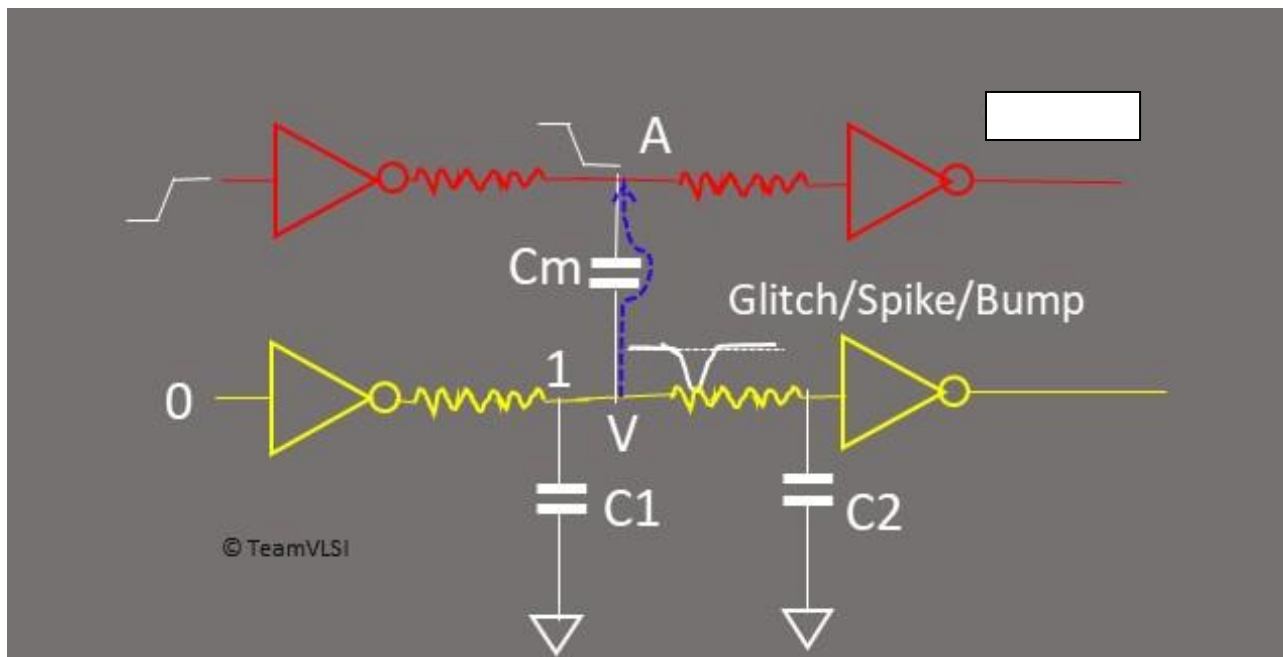


Figure-2: Crosstalk glitch (Fall)

In this case, the aggressor net switches from logic 1 to logic 0 and the victim net is at constant high logic as shown in the figure-2. Now consider the node A, node V, Mutual capacitance  $C_m$  and the path from V to A. As node A start switching from high to low, a potential difference across the mutual capacitance gets developed and the mutual capacitor  $C_m$  starts charging through node V to node A. During this event, there is a leakage current which starts flowing from node V to node A through the mutual capacitance  $C_m$  due to the leaky nature of mutual capacitance. This leakage current will drop the potential of node V, which creates a falling spike or falling glitch on the victim net as shown in figure-2.

So, whenever one net switches from high to low and other neighbouring net is supposed to remain constantly high, will get affected by the switching net due to the mutual capacitance and have a falling glitch on it.

In case-1 and case-2 we have seen that if one net is switching and another neighbouring net is at constant logic and if they have mutual capacitance between them, the other net may get affected and that net may have a sudden raising or falling bump or spike. such a spike on the

victim net is called a crosstalk glitch or crosstalk noise. Figure-3 shows the situations when there is a raise glitch or fall glitch.

### **Effects of crosstalk glitch**

Does every glitch unsafe? The answer is it depends on the height of the glitch and the logical connection of the victim net. If the height of the glitch is within the noise margin low (NML), Such a glitch is considered a safe glitch. If the glitch height is above the noise margin high (NMH), such a glitch is considered a potentially unsafe glitch. In the case of a glitch, height is in between NMH and NML, this is an unpredictable case

### **Crosstalk glitch height**

Crosstalk glitch will be safe or unsafe depending on the height of the crosstalk glitch and the logic pin from which the victim net is connected. So let's investigate the factors on which the crosstalk glitch height depends. Crosstalk glitch height depends basically on three factors:

1. *Coupling capacitance*
2. *Aggressor's drive strength*
3. *Victim's drive strength*

Closer the nets will have greater coupling capacitance. More the capacitance will have a larger glitch height. Drive strength of the aggressor and victim driver will also affect the glitch height. The high drive strength of the aggressor net will impact more the victim net. If the drive strength of the victim net is high, then it will not be easy to change its value, which means lesser will be the effect of crosstalk.

It was all about the crosstalk glitch or crosstalk noise, Now let's move to the second effect which is crosstalk delta delay or crosstalk delay.

### **Crosstalk Delay**

Crosstalk delay occurs when both aggressor and victim nets switch together. It has effects on the setup and hold timing of the design. Crosstalk delay may cause setup and hold timing violation. So it is important to do a crosstalk delay analysis and fix the timing considering the effect of crosstalk. Crosstalk could either increase or decrease the delay of a cell depending upon the switching direction of aggressor and victim nets. We will take two cases one when both nets switch in the same direction (high to low or low to high) and the other both the nets switch in opposite directions and will analyze the effect of crosstalk delay.**Case-3: Aggressor and victim net switch in opposite directions**

Let's consider aggressor net switches from low to high logic and victim net switches from high to low (opposite). as shown in figure-6.



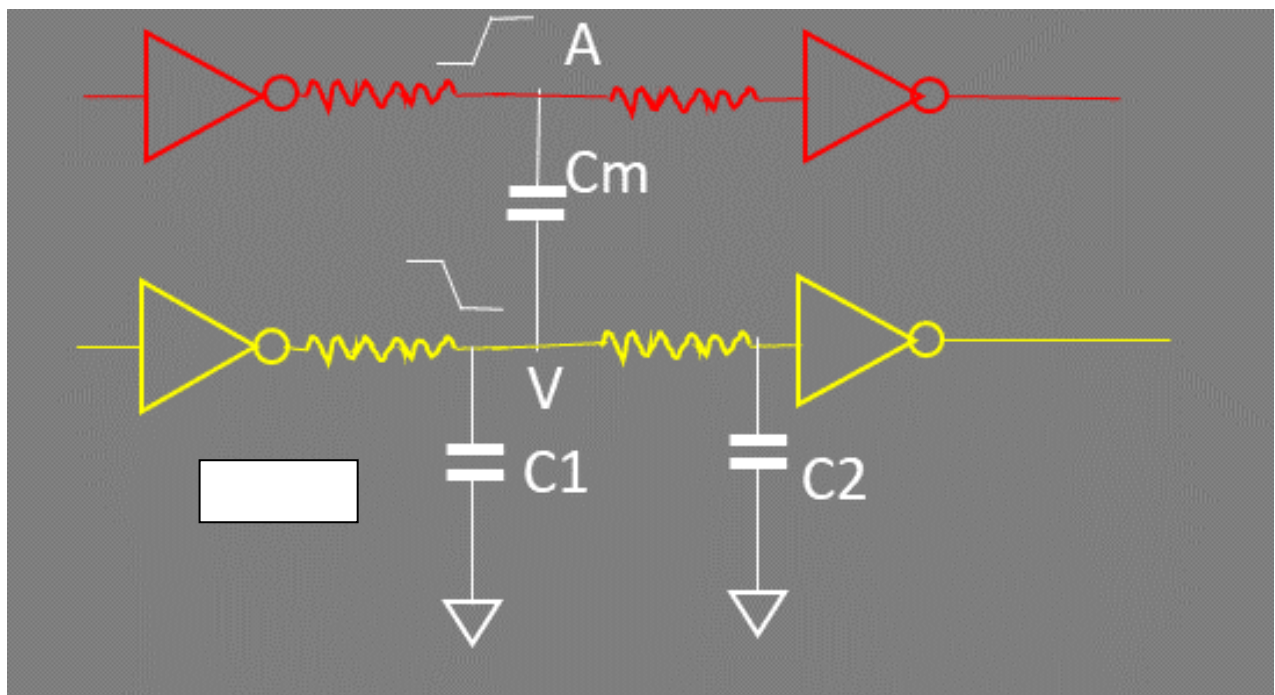


Figure-6: Crosstalk delay due to opposite direction switching

As node A starts to transition from low to high at the same time, node V starts switching from high to low. There will be a potential difference from node A to V as half of the transition happened. There is a coupling capacitance between A and V so the aggressor node will try to pull up the victim node. This will affect the smooth transition of the victim node from high to low and will have a bump after half of the transition and this will result in an increase in the transition time of the victim net. Figure-7 shows the transition of nets. After crosstalk, the delay of the cell will be increased by  $\Delta$  and the new delay will be  $D + \Delta$ .

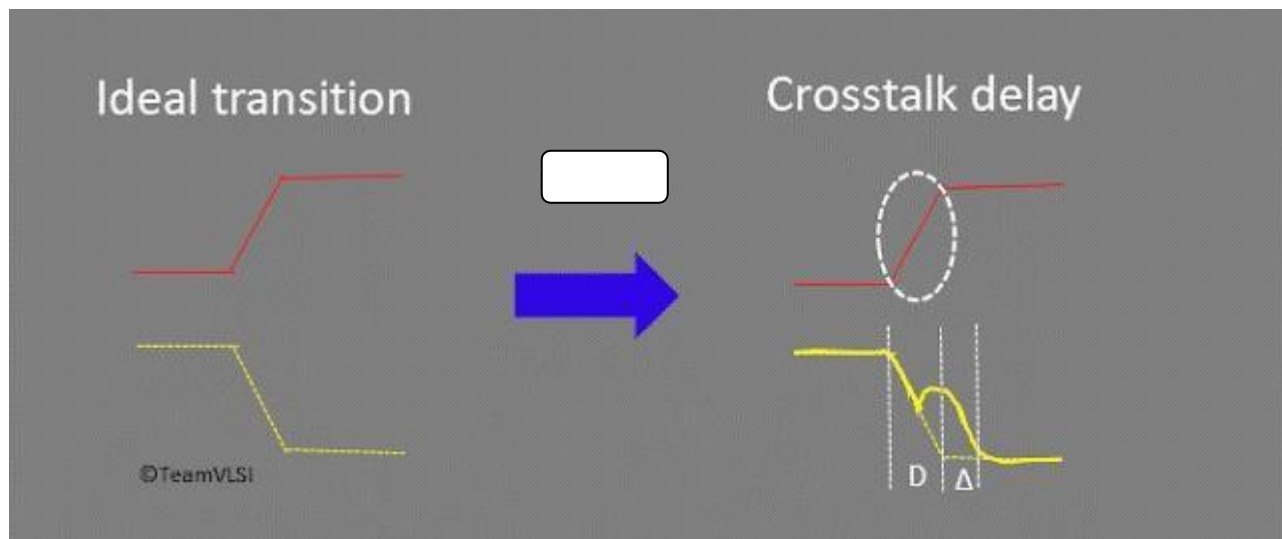


Figure-7: Crosstalk delay (increase)

#### Case-4: Aggressor and victim nets switch in the same direction

Let's consider the aggressor net switches from low to high logic and the victim net also switches from low to high (same direction). as shown in the figure-8.

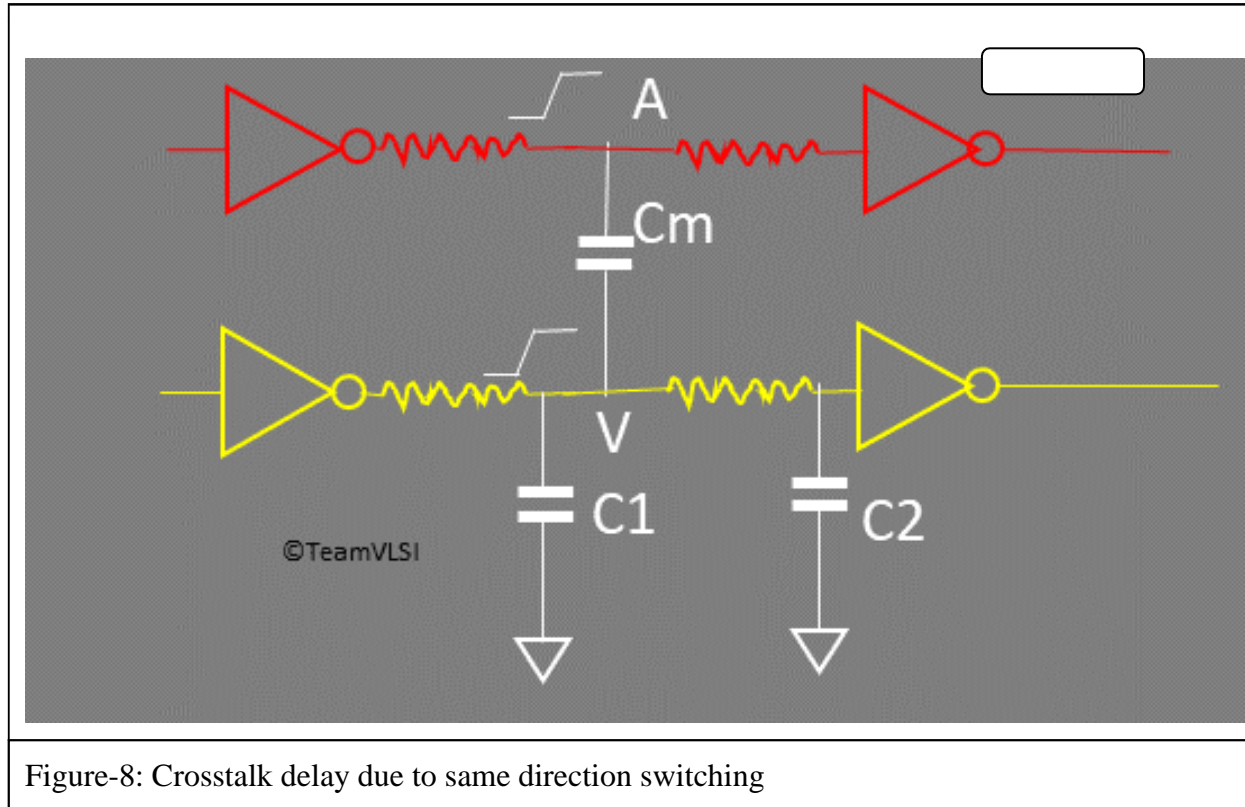


Figure-8: Crosstalk delay due to same direction switching

As node A starts to transition from low to high at the same time, node V also starts switching from low to high. Suppose the aggressor net has high drive strength and so fast transition, a potential difference from node A to V will be developed after half of the transition happened. There is a coupling capacitance between A and V so the aggressor node will try to fast pull up the victim node. This will affect the smooth transition of the victim node from low to high and will have a bump after half of the transition and this will result in a decrease in the transition time of the victim net. Figure-9 shows the transition of nets. After crosstalk, the delay of the cell will be decreased by  $\Delta$  and the new delay will be  $(D - \Delta)$ .

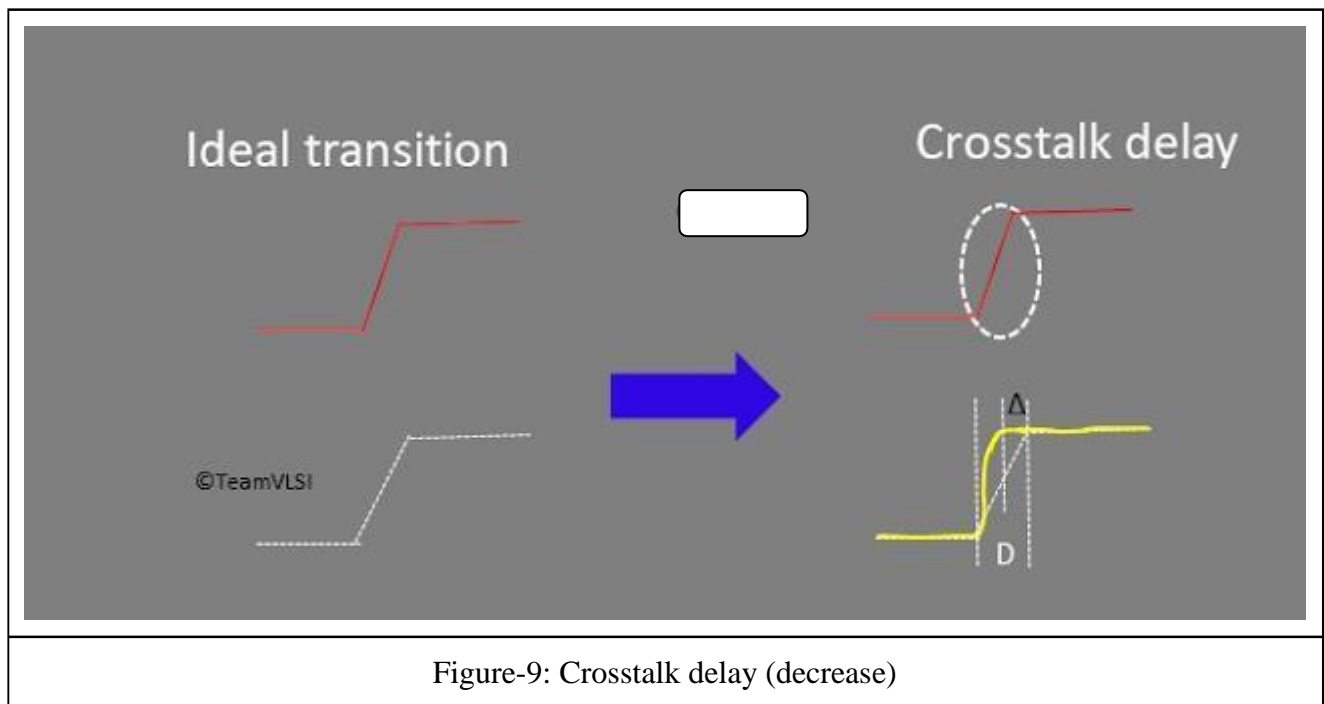


Figure-9: Crosstalk delay (decrease)

#### Effects of crosstalk delay

There are various effects of crosstalk delay on the timing of design. It could make unbalance a balanced clock tree, could violate the setup and hold timing. In this section, we will discuss some of them.

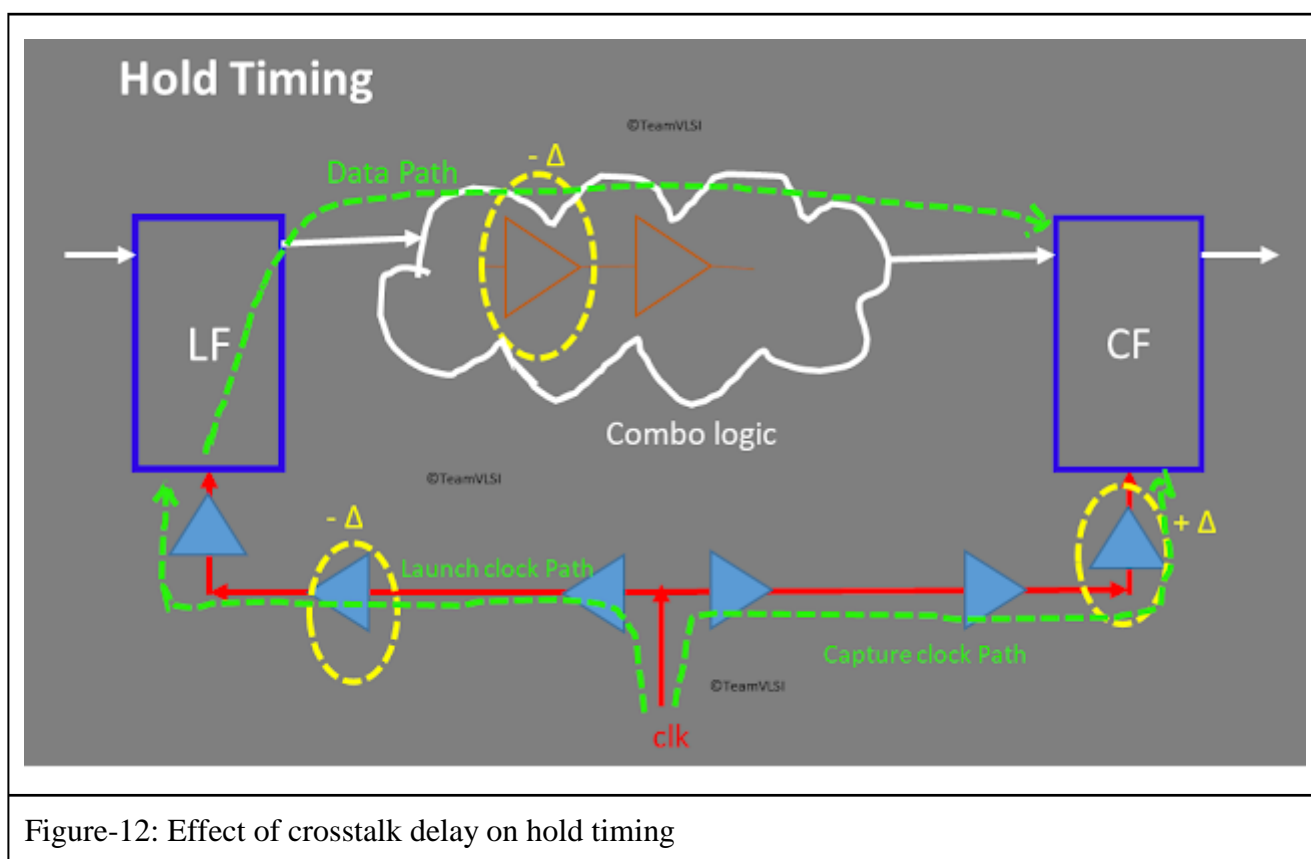
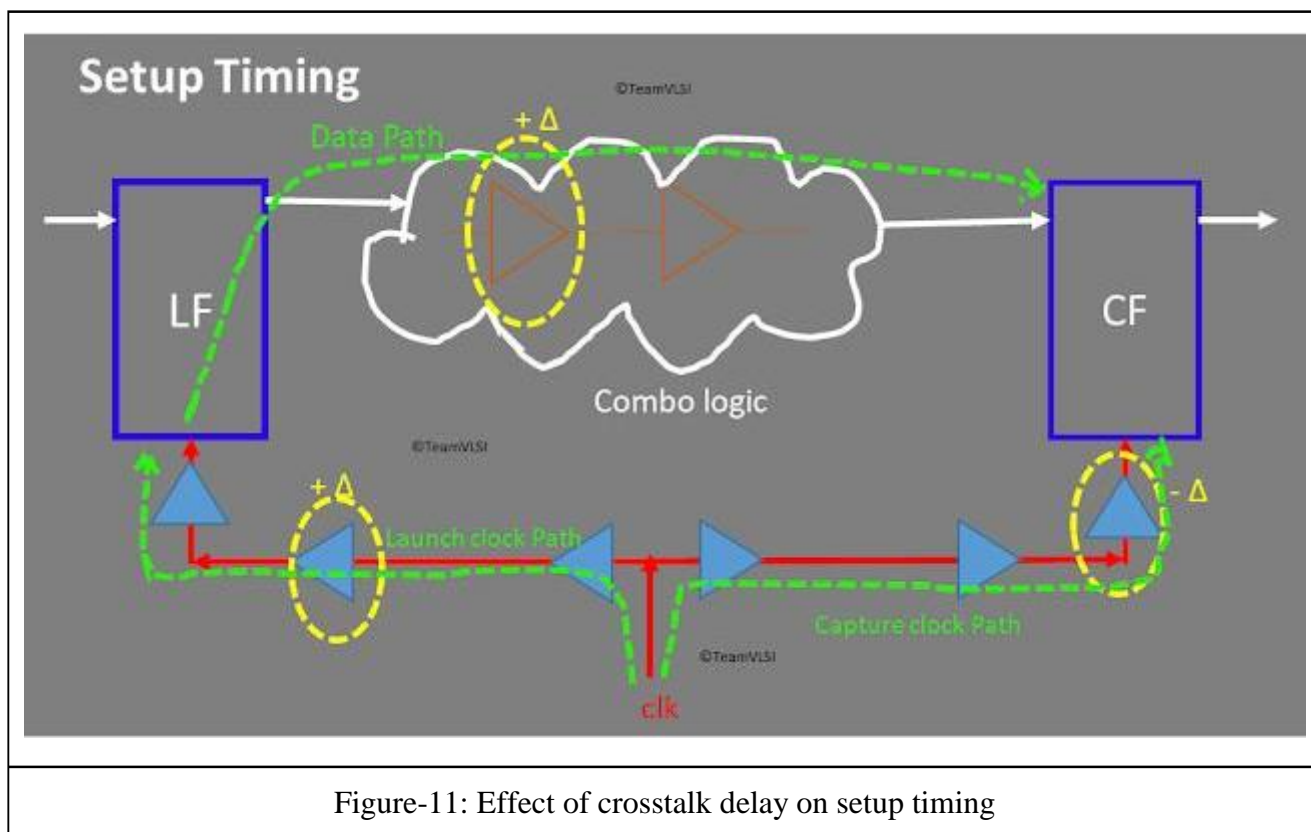
#### **Effect on setup and hold timing:**

Crosstalk delay can violate the setup timing. Figure-11, shows the data path, launch clock path and capture clock path.

For setup timing, data should reach the capture flop before the required time of capture flop. So if there is an increase of delay in the data path or launch clock path it may cause a setup violation. Setup violation may also happen if there is a decrease in delay on the capture clock path. These effects of crosstalk delay must be considered and fixed the timing.

Hold timing may be violated due to crosstalk delay. Figure-12, explains the situations where the hold time could violate due to crosstalk delay.

If there is a decrease in the delay of any cells in the data path and launch clock or there is an increase of delay of cells in the capture clock path due to crosstalk delay, It may result in the hold timing violation. Such cases must be considered and fix the timing.



## Crosstalk prevention techniques

There are various ways to prevent crosstalk, some of the well-known techniques are as follow.

### 1. Increase the spacing between aggressor and victim net:

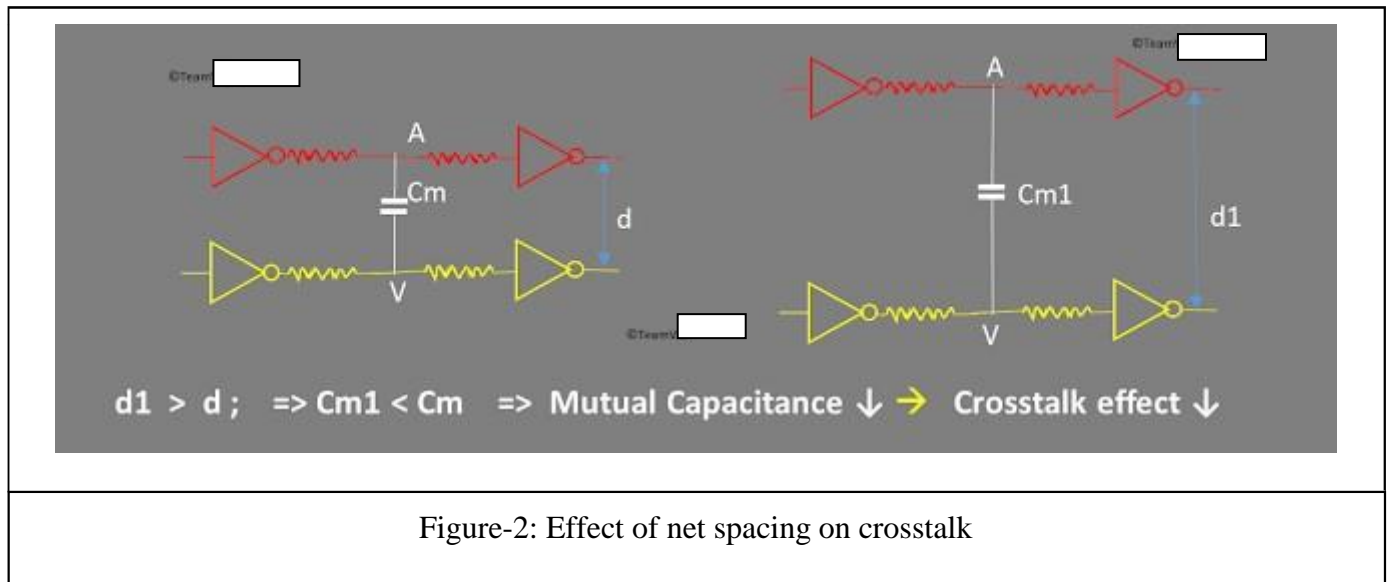
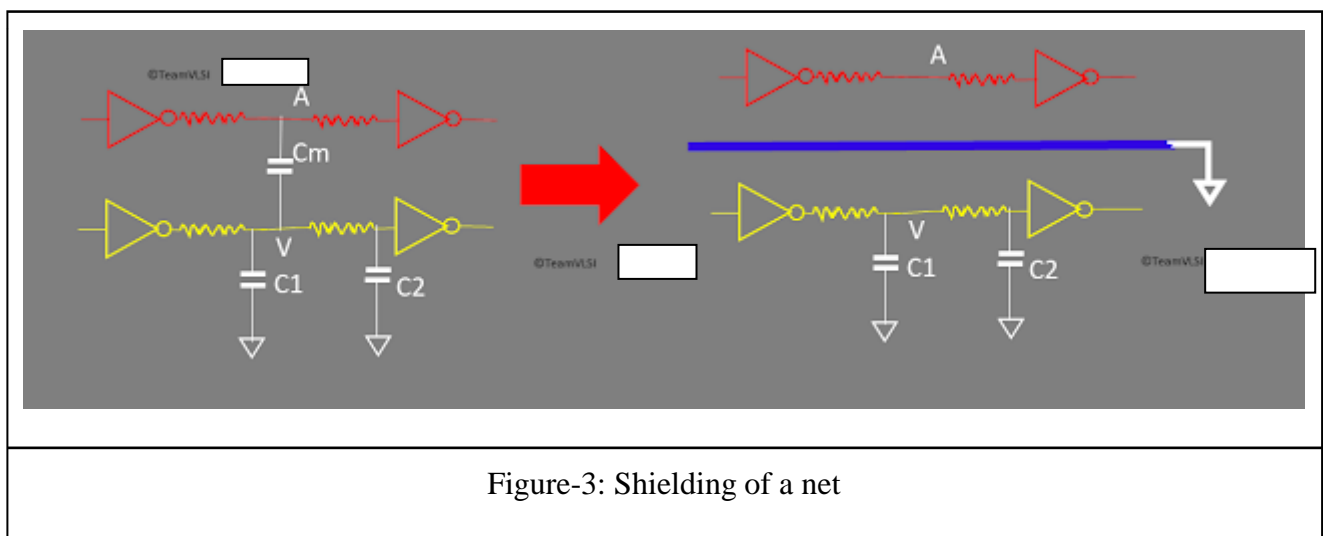


Figure-2 shows that by increasing the spacing between aggressor and victim net we are ultimately reducing the coupling capacitance between them as the capacitance is inversely proportional to the distance between them. So by increasing the spacing crosstalk will decrease.

### 2. Shielding of nets:

Figure-3 shows the shielding technique used to prevent crosstalk. Generally, we insert a shielding net between the victim and the aggressor net. The shielding net is connected to strong VDD or VSS.



By shielding a net the two things will happen, one is the direct coupling capacitance between the aggressor and victim net will vanish and secondly the shielding net will remain at a constant logic so there are no chances of crosstalk.

The above two techniques will prevent the crosstalk but it has an impact on the area. Both techniques will require more area to route them.

### **3. Upsizing the victim cell:**

If we increase the drive strength of the victim cell it will not be easy to affect by the aggressor net.

### **4. Downsize the aggressor cell:**

Higher the drive strength of aggressor cell, higher is the impact of crosstalk on the victim. So by reducing the drive strength we can reduce the crosstalk effect.

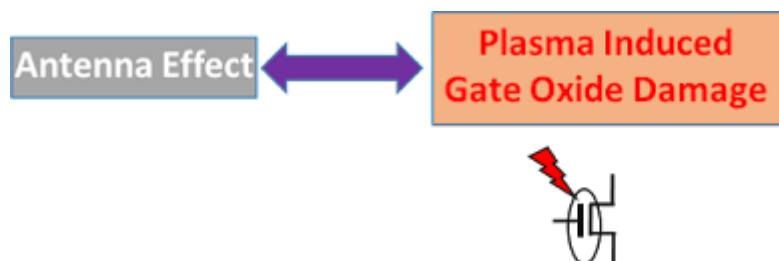
Crosstalk timing window analysis is based on the concept that we need to consider a timing window on which the aggressor has an effect on victim net.

## **Antenna Violation :**

Gate Oxide of a MOS transistor is the most sensitive part of a MOS device. Special care needs to be taken during fabrication of ASIC to protect it from any damages during fabrication steps and ASIC operation too. The antenna effect is a phenomenon that may cause damage to the gate oxide of MOS during the fabrication process especially due to the plasma etching process. In this article, we will investigate the antenna effect phenomena in detail and the reasons which are responsible for this effect.

### **What is the Antenna Effect?**

The term Antenna Effect might not give you the right intuition about the actual effect it may lead you to think about electromagnetic radiation or transmitter-receiver concepts but here the case is different. So It has another popular name which is called “Plasma Induced Gate Oxide Damage” which provides the right intuition about the effect. As this name itself indicates that this is an effect caused by the Gate Oxide Damage due to the Plasma Etching process during the fabrication process of VLSI chips.



Although the antenna effect occurs during the fabrication stage of the chip especially at the time of plasma etching but the prevention mechanism should be set in the physical design stage. The fabrication laboratory provides the antenna rule file which must be checked and designed should be cleaned as per the antenna rule during the physical signoff stage.

In fabrication flow first FEOL (Front End Of Line) is fabricated which involves the fabrication of all MOS transistors. Once the FEOL fabrication is done BEOL (Back End Of Line) fabrication starts which involves the fabrication of metal interconnects. Antenna effect comes into the picture while BEOL fabrication. In IC manufacturing process Plasma etching process is used to fabricate the metal interconnects. Plasma etching is a dry and anisotropic etching process, used for selective etching. Plasma contains high energetic ions and radicals which get collected by the metal interconnects while the etching process of metals. Figure-1 shows the structure of MOS and collection of plasma by the interconnect.

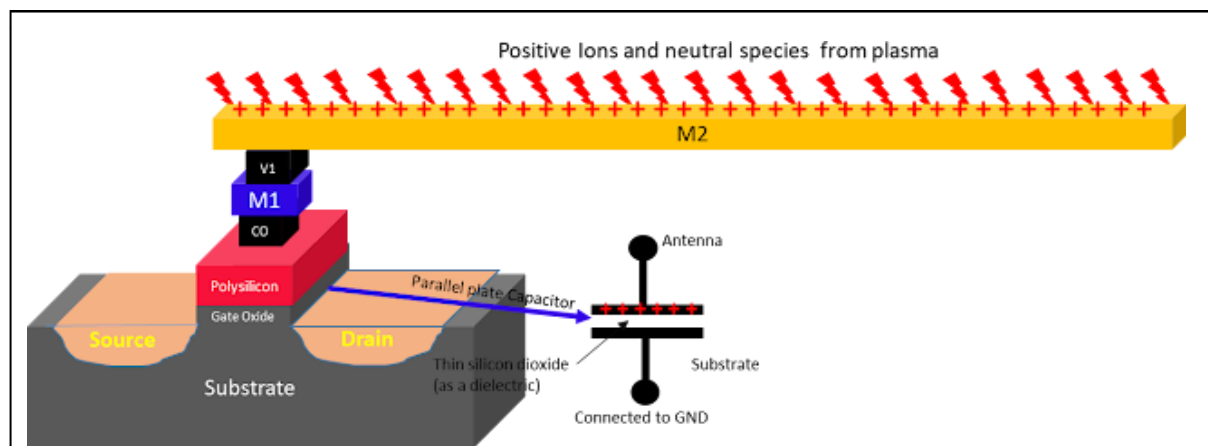


Figure-1: MOS structure and plasma etching

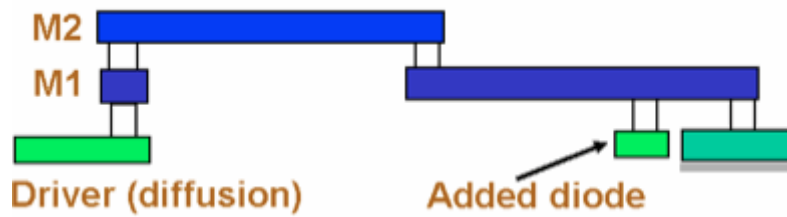
The amount of charge accumulation depends on the surface area of interconnect. These collected ions increase the potential of the interconnect and if the interconnect is connected to the poly gate, ultimately the potential of the gate will increase. Due to this increased potential of the gate, a drainage path may be formed through the gate oxide to substrate to balance this extra accumulated charge on the gate. If the amount of charge accumulation is high, this drainage path through the gate oxide may either breakdown the gate oxide which leads to permanent damage of MOSFET or may create charge trapping in the gate oxide which further leads to many side effects like early gate oxide breakdown, mobility degradation and threshold voltage shift.

Gate oxide damage occurs basically due to plasma etching of interconnects connected to the gate, that's why this effect is also called "Plasma Induced Gate Oxide Damage" or "Antenna Effect". The metal interconnect which collects the plasma (ions) and is connected to the gate is basically termed as the antenna.

### Ways to remove antenna violations

1. **Antenna diode** – To avoid this deposition of charge at the gate of a transistor, a diode is generally used in reverse biased mode which can drain out the charge without affecting the transistor circuitry.

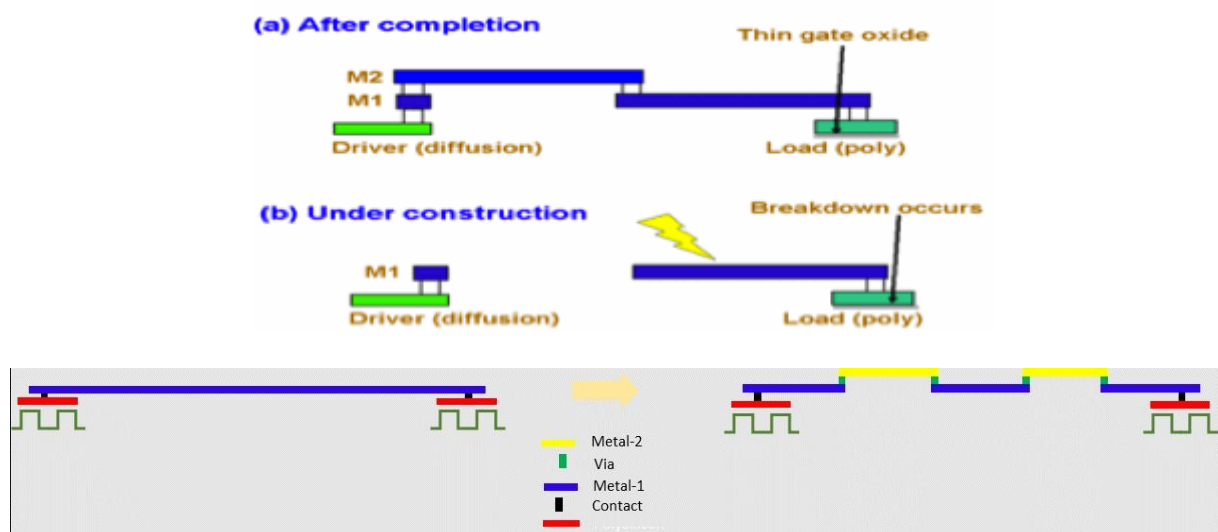




**Figure 1** An antenna diode is used to remove antenna violation near the receiver.

2. **Metal hopping** – When the metal connected to a gate is long and there is space for a higher metal hop, it is always advisable to do so instead of using an antenna diode in order to avoid antenna violations.

Suppose we jog a metal 2 net with metal 3. Then while etching metal 2, the part of the net which is drawn in metal 2 only comes into picture because metal 3 has not been manufactured yet. So the effective charge reduces. This is the reason for using a higher metal as a jumper.



**Figure 2** Use metal hopping to avoid antenna violations.

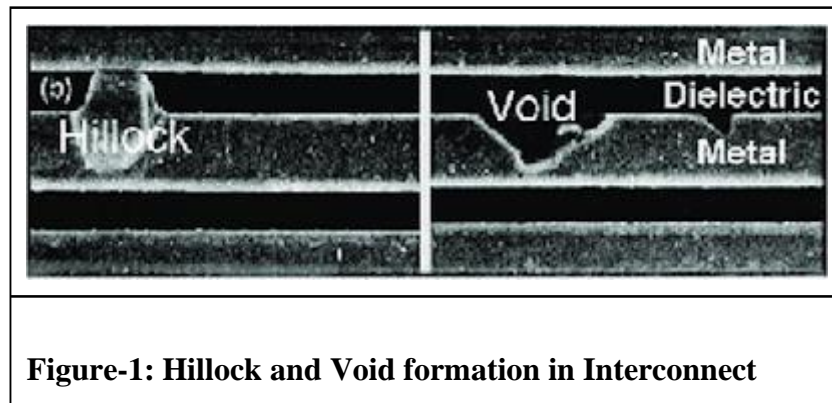
## **Electromigration:**

When a high current density passes through a metal interconnect, the momentum of current-carrying electrons may get transferred to the metal ions during the collision between them. Due to the momentum transfer, the metal ions may get drifted in the direction of motion of electrons. Such drift of metal ions from its original position is called the electromigration effect

### **Effects of EM:**

Once the metal ions get started to shifting from its original position, these will create problems in the interconnect. It could result in an excess of ions accumulation in a particular location or deficient of ions. So either Hillocks or Void could occur in the metal interconnect.





**Void:** If the incoming ion flux is lesser than the outgoing ion flux, It will create a void in interconnect. A void can lead a discontinuity in the interconnect and result an open circuit.

**Hillocks:** If incoming ion flux is greater than the outgoing ion flux, It will cause the accumulation of ions and create a hillock in the interconnect. A hillock can increase the width of a metal interconnect and touch the neighbouring metal interconnect which may result in a short circuit.

### **Prevention techniques for EM:**

With the scaling of the technology node, the interconnect used is also changed. Initially, pure Aluminium was used as interconnect then the industry started using the Al-Cu alloy and later shifted to Copper interconnects. Copper interconnects can withstand approximately 5 times more current as compared to Aluminium interconnects while maintaining similar reliability requirements.

During the physical design, the following techniques could be used to prevent the EM issue

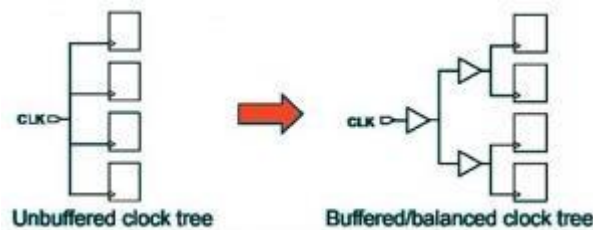
- Increase the metal width to reduce the current density
- Reduce the frequency
- Lower the supply voltage
- Keep the wire length sort
- Reduce the buffer size in clock lines

To prevent the EM issue, EM Checks is performed during the physical signoff stage with respect to the EM rules provided by the foundry.

# **CLOCK-TREE SYNTHESIS**

## **Clock tree synthesis**

- CTS is one of the most important stages in PnR. CTS QOR decides timing convergence and power. In most of the ICs clock consumes 30 to 40% of total power. So efficient clock architecture, clock gating and clock tree implementation helps to reduce power
- The process of distributing the clock and balancing the load is called CTS. Basically, delivering the clock to all sequential elements. CTS is the process of insertion of buffers and inverters along the clock paths of ASIC design in order to achieve zero or minimum skew or balanced skew. Before CTS, all clock pins are driven by a single clock source. CTS starting point is clock source and CTS ending point is clock pin of sequential cells.



## **Difference between HFNS and CTS**

- Buffers and clock inverter with equal rise and fall times are used. Whereas HFNS uses buffers and inverters with a relaxed rise and fall times.
- HFNS are used mostly for reset, scan enable and other static signals having fan-outs. There is not stringent requirements of balancing and power reduction.
- Clock tree power is given special attention as it is a constantly switching signal. HFNS are mostly performed for static signals and hence not much attention to power is needed.
- NDR rules are used for clock tree routing.

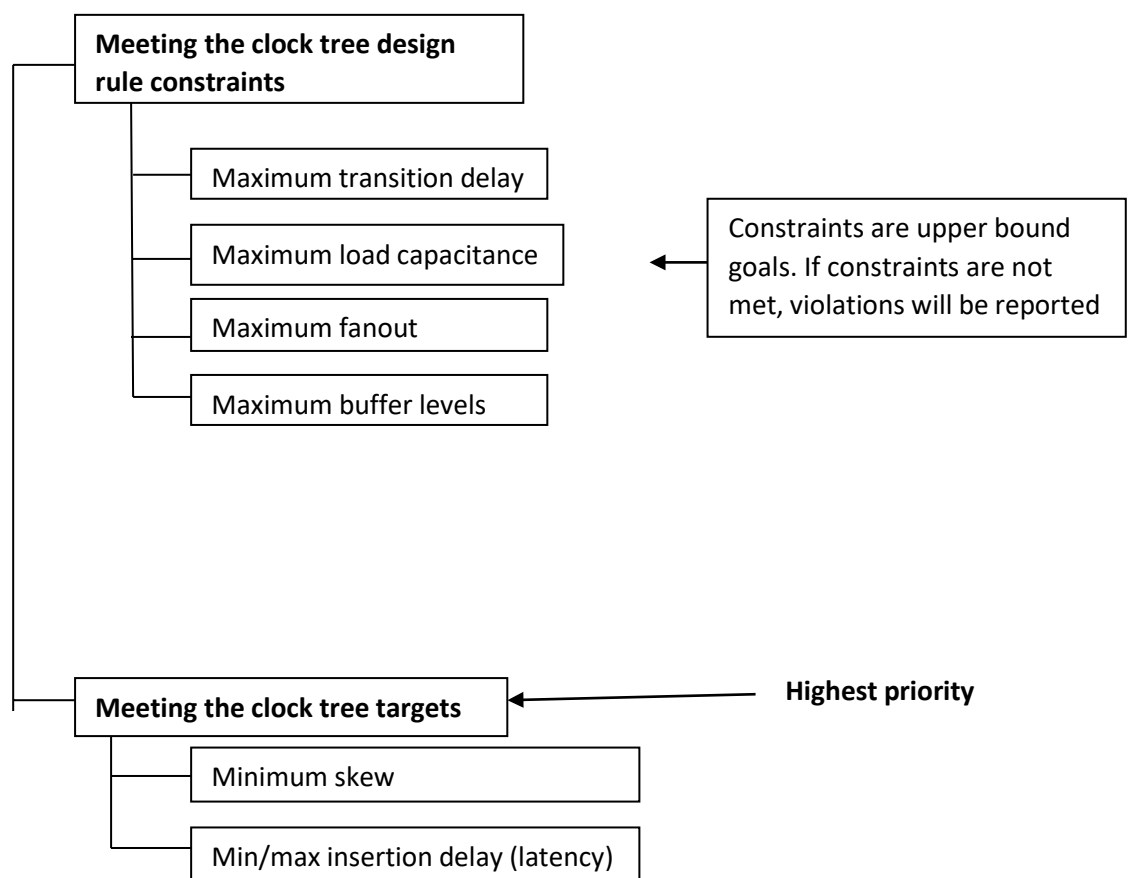
## **Difference between clock buffer and normal buffer**

- Clock buffer have equal rise time and fall time therefore pulse width violation is avoided. In clock buffer BETA ratio is adjusted such that rise and fall time are matched. This may increase size of clock buffer compared to normal buffer.
- Normal buffer may not have equal rise and fall time. The clock buffers are usually designed such that an input signal with 50% duty cycle produces an output with 50% duty cycle.

## Inputs of CTS

- Technology file(.tf)
- Net list
- SDC
- Library file(.lib, .lef) and TLU+ file
- Placement DEF .
- Clock specification file which content insertion delay skew ,clock transition, clock cells, NDR, CTS tree type, CTS exception, list of buffers or inverters etc.

## Goals of CTS

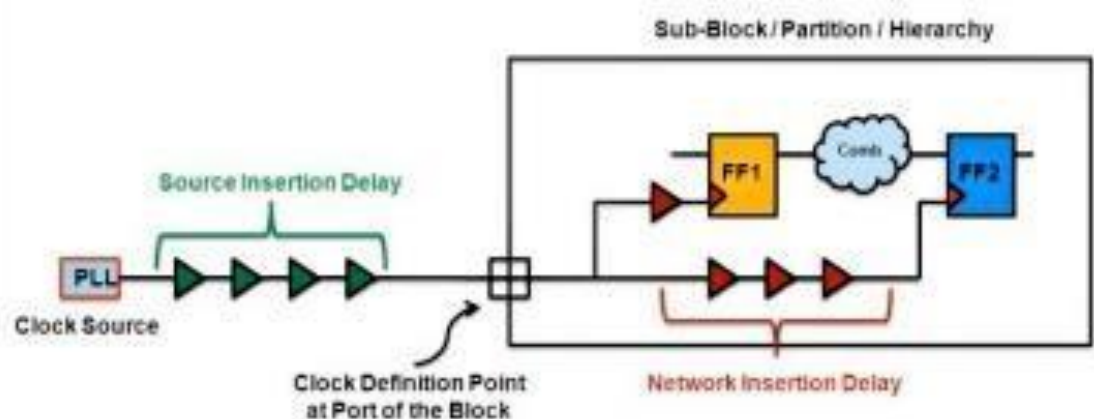


## **Sanity checks need to be done before CTS**

- Check legality
- Check power stripes, standard cell rails and also verify PG connections.
- Timing QOR (setup should be under control)
- Timing DRVs
- High fan out nets (Like scan enable/any static signal)
- Congestions {Running CTS on congestion design/ design with congestion hot spot can create more congestion and other issues(Noise/IR)}
- Remove don't \_use attribute on clock buffers and inverter
- Check whether all pre existing cells in clock path are balanced cells.
- Check and qualify don't \_touch, don't size attribute on clock components.

## **Different Clock Terms:**

### **a) Clock latency/insertion delay**



- Total time taken by the clock signal to reach the inputs of the register
- Source latency is the time between clock sources to clock definition ports
- Network latency is the time between clock definition ports to clock leaf cells in the design

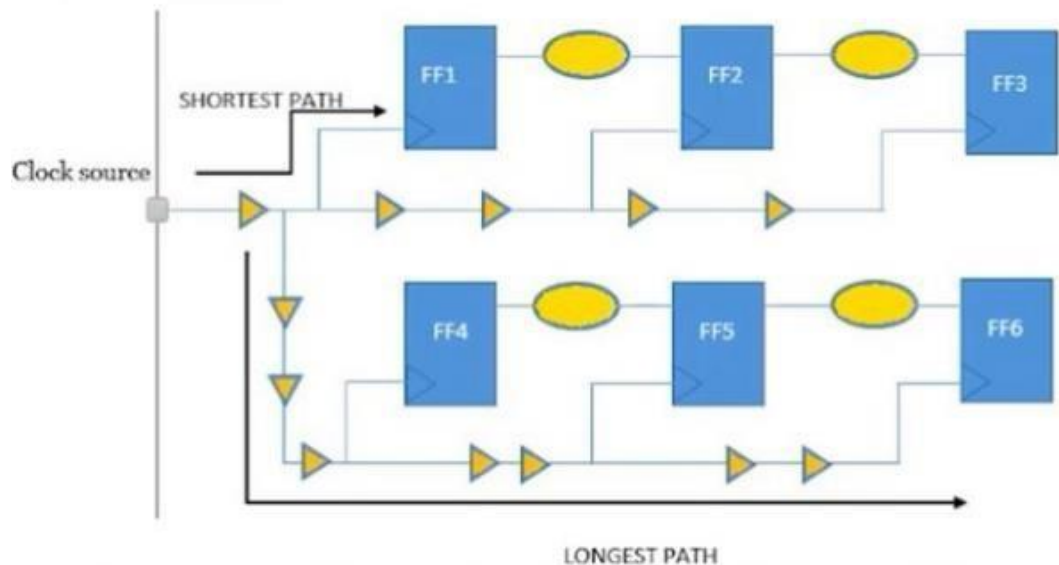
### **b) Insertion delay (ID)**

- ID is the clock latency, but it is after clock tree is synthesized.
- ID is the physical delay and clock latency is the virtual delay.
- Latency is a target/constraint given to the tool through SDC file or clock tree attribute file and insertion delay is the achieved delay value in clock path after CTS.

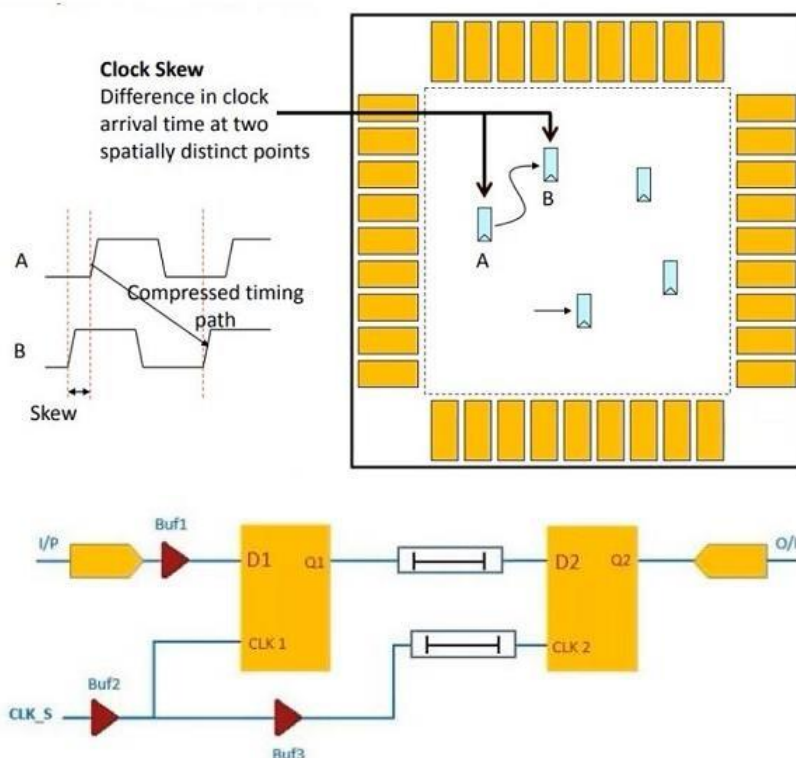
c) **Clock skew** : Clock skew between two flip-flops represents the difference in arrival times of clock signal at the respective clock pins

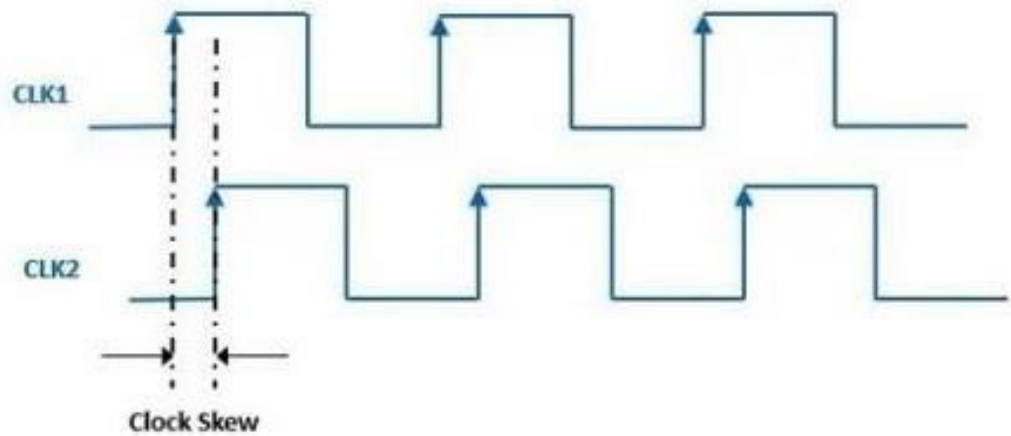
**Local skew**: It is the difference in arrival of clock at two related flip-flops .

**Global skew**: It is defined as the difference between max insertion delay and min insertion delay of any flops. It is also defined as the difference between shortest clock path delay and longest clock path delay reaching two sequential elements.

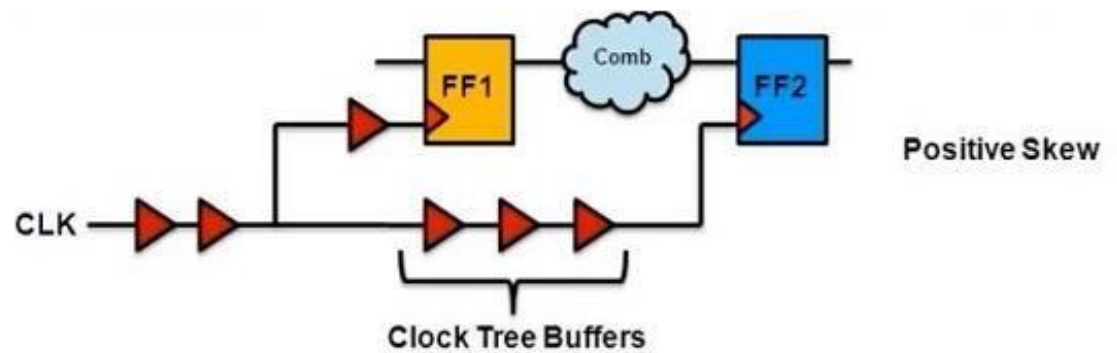


**Useful skew** : If clock is skewed intentionally to resolve violations. It is called useful skew.

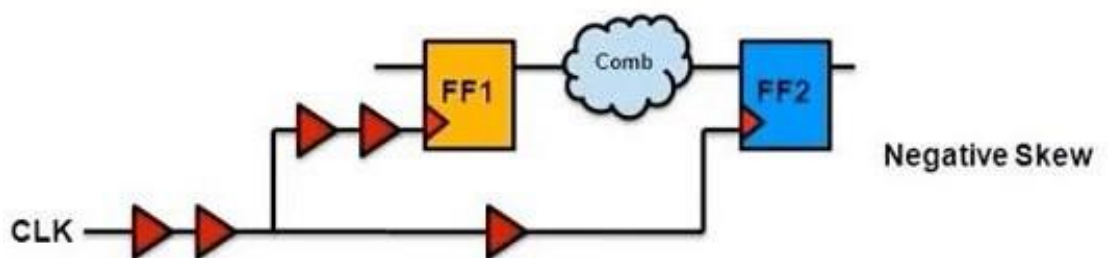




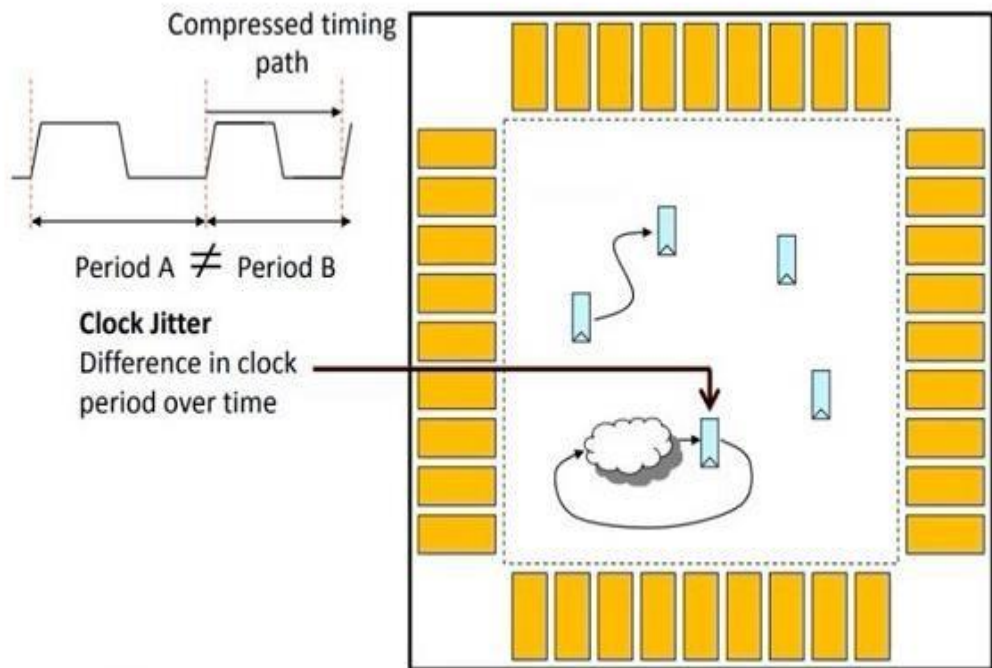
**Positive skew** : If capture clock comes late than launch clock then it is called positive clock.



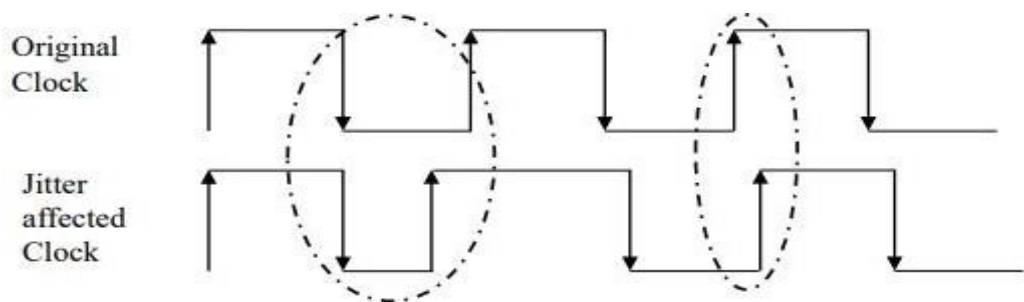
**Negative skew** : If capture clock comes early than launch clock then it is called negative clock.



#### d) Clock Jitter: Temporal Clock Variation



- Jitter is the short-term variations of clock edge from ideal position in time.
- The two major components of Jitter are random Jitter and deterministic Jitter.
- Factors causing Jitter includes imperfections in Clock oscillator, supply voltage variations, Temperature variations, Crosstalk.

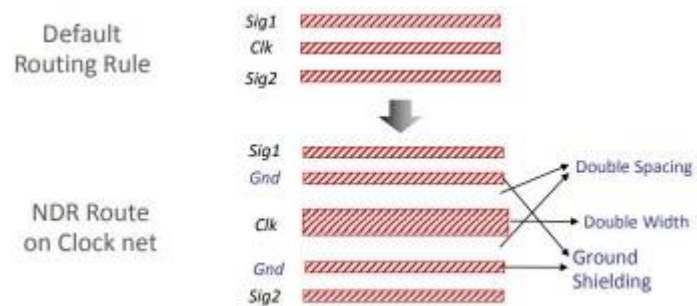
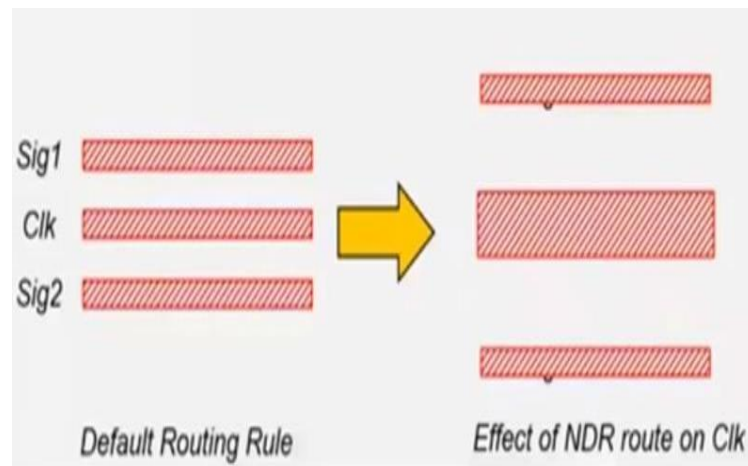


#### CONTENTS OF CTS SPECIFICATION :

- Cells to be used to balance clock trees
- CTS exceptions
- NDR rules (double width and double spacing)
- Max clock cap limit
- Clock inverters and buffers types
- Target skew (Different Skew groups )
- Max transistion
- Preferred routing layers

## **Non default rule**

- This is user defined routing rules apart from the default routing rule
- NDR s make the clock routes less sensitive to cross talk or EM effects
- Double /triple width and spacing is used to avoid EM and cross talk
- NDRs will improve insertion delay

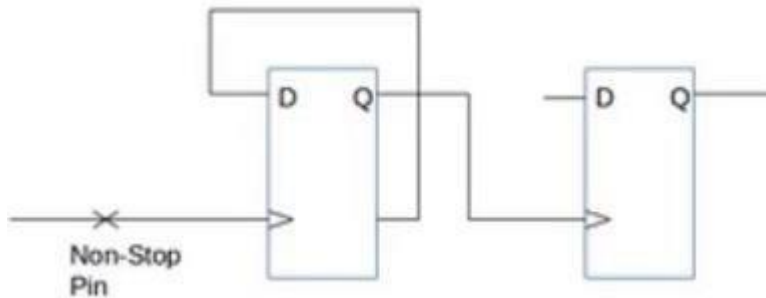


## **Clock tree exceptions:**

- Non-stop pin
- Exclude pin
- Float pin
- Stop pin
- Don't touch sub-tree
- Don't buffer net
- Don't size net



### Non stop pin :

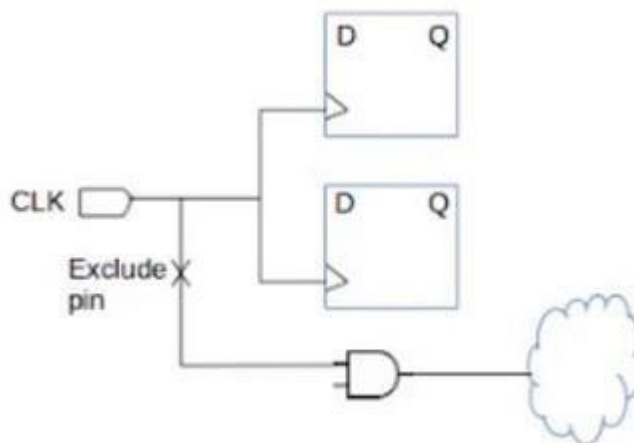


Non-stop pins trace through the endpoints that are normally considered as endpoints of the clock tree.

Example:

- The clock pin of sequential cells driving generated clock are implicit non stop pins
- Clock pin of ICG cells

### Exclude pin :



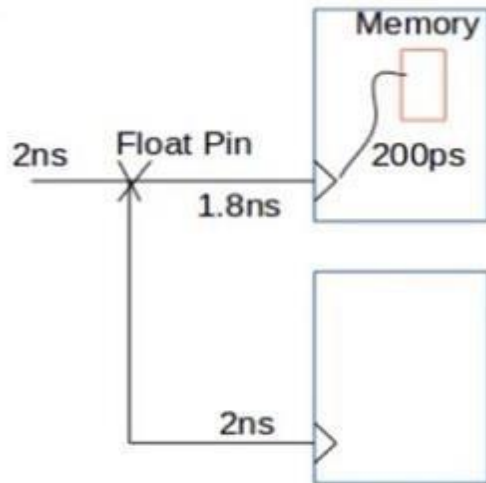
Exclude pins are non clock tree endpoints that are excluded from clock tree timing calculation and optimization. The tool considers exclude pins only in calculation and optimization of design rule constraints. During CTS the tool isolates exclude pins from the clock tree by inserting a guide buffer before the pin or these pins are need not be considered during the clock tree propagation.

Example :

- Non clock input pin of sequential cell
- Multiplexer select pin
- Three state enable pin
- Output port
- Incorrectly defined clock pin (if pin don't have trigger edge info)
- Cascade clock

In the above figure, beyond the exclude pin of the tool never perform skew or insertion delay optimization but does perform design rule fixing.

### Float pin :

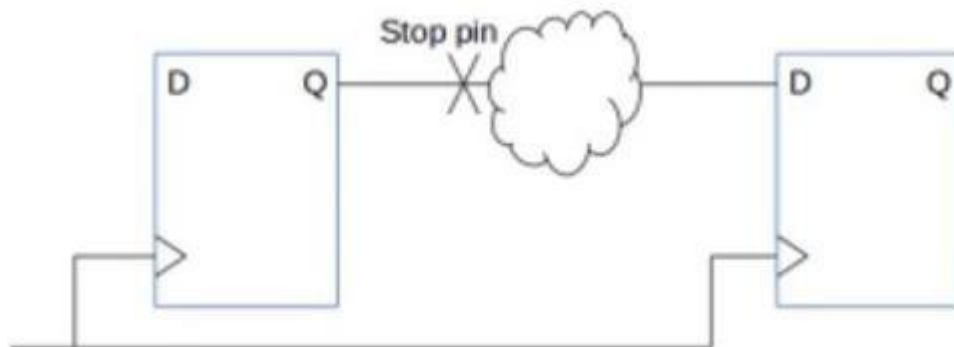


Float pins are pins that have special insertion delay requirements and balancing is done according to the delay. This is same as sync pin but internal clock latency of the pin is taken into consideration while building the clock tree. To adjust the clock arrival for specific endpoints with respect to all other endpoints.

Example :

- Clock entry pin of hard macros

### Stop pin:



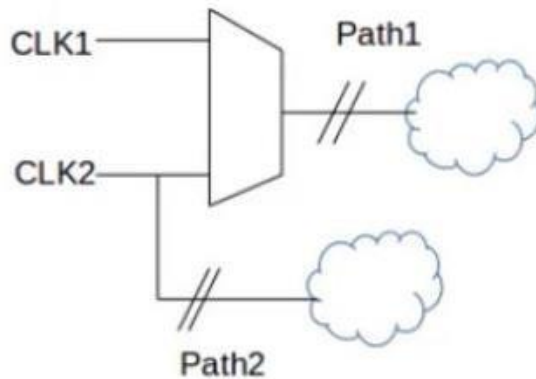
Stop pins are the endpoints of clock tree that are used for delay balancing. In CTS, the tool uses stop pins in calculation and optimization for both DRC and clock tree timing.

Example:

- Clock sink are implicit stop pins

The optimization is done only up to the stop pins as shown in above fig. The clock signal should not propagate after reaching the stop pin. This pin needs to be considered for building the clock tree.

### Don't touch sub-tree :



If we want to preserve a portion of an existing clock tree, we put don't touch exception on the sub-tree.

- CLK1 is the pre existing clock and path 1 is optimized with respect to CLK1.
- CLK2 is the new generated clock. Don't touch sub-tree attribute is set with respect to CLK1.

Example:

- If path1 is 300ps and path2 is 200ps, during balancing delay are added in path2.
- If path1 is 200ps and path2 is 300ps, during balancing delay cant be added on path 1 because on path1 don't touch attribute is set and we get violation.

### Don't buffer nets:

It is used in order to improve the results, by preventing the tool from buffering certain nets. Don't buffer nets have high priority than DRC. CTS do not add buffers on such nets.

Example:

- If the path is a false path, then no need of balancing the path. So set don't buffer net attribute.

### Don't size cell:

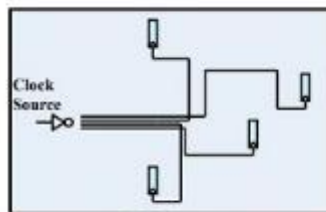
To prevent sizing of cells on the clock path during CTS and optimization, we must identify the cells as don't size cells.

### Specifying size only cells:

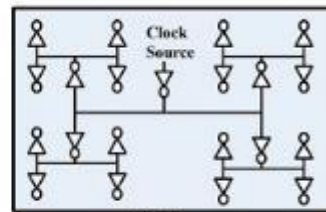
During CTS and optimization, size only cells only be sized not moved or split. After sizing, if the cells overlap with an adjacent cell after sizing , the size-only cell might be moved during the legalization step.

### CTS algorithms:

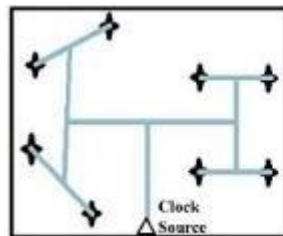
- RC tree based CTS.
- H tree based algorithm.
- X tree based algorithm.
- Method of mean and medium (MMM)
- Geometric matching algorithm (GMA)
- Pi configuration



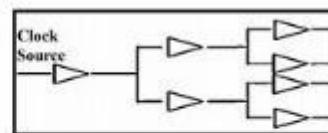
RC-Tree



H-Tree

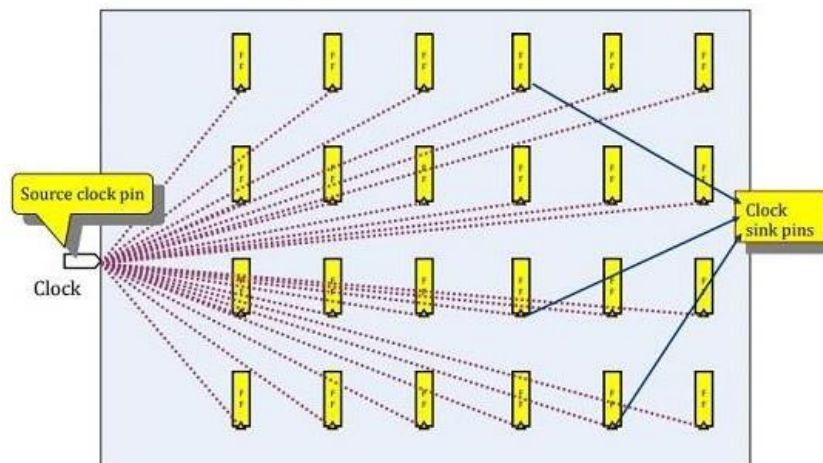


GMA

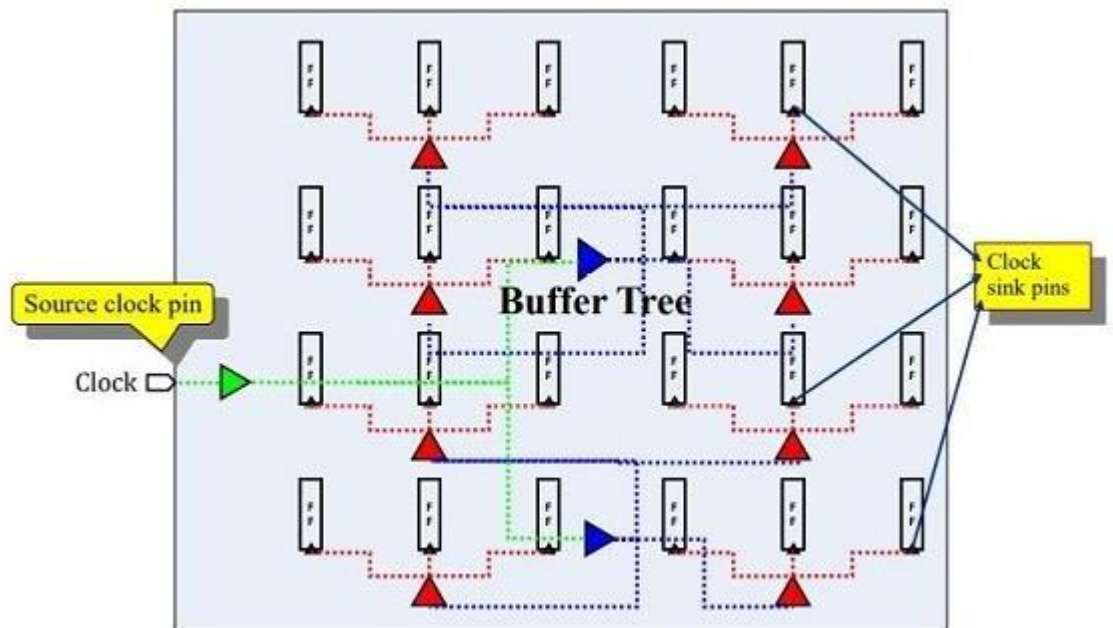


Pi Configuration

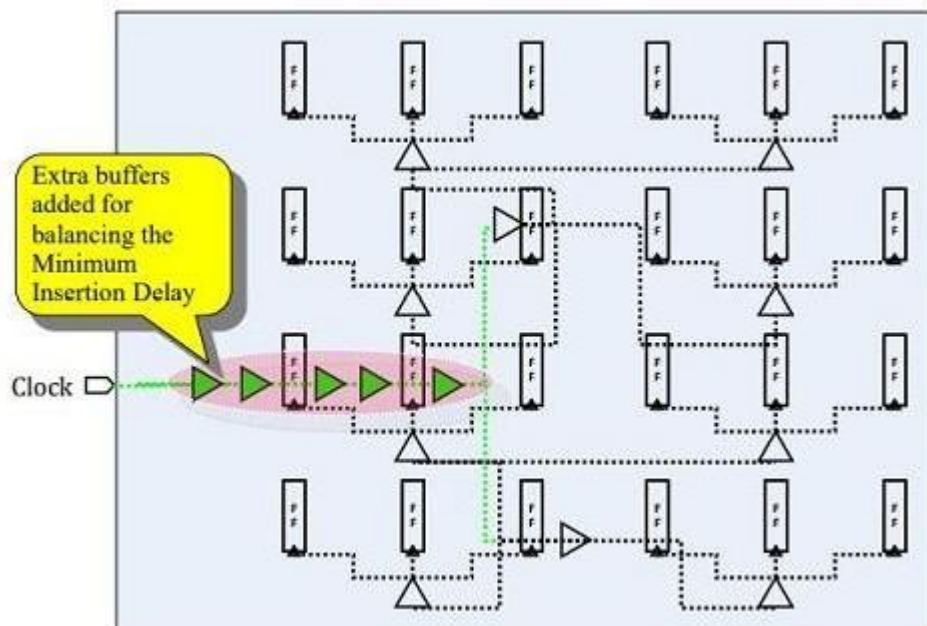
- Before CTS all clock pins are driven by a single clock source



- After CTS the buffer tree is built to balance the loads and minimize the skew



- After CTS a delay line is added to meet the minimum insertion delay (ID)



### **Analyze the clock tree:**

- Report timing (both setup and hold)
- If timing not met then check clocks to be grouped (balanced together)
- Report insertion delay and skew and verify that the targets are achieved
- Report DRV targets (fanout, capacitance and transition)
- Check the intended leaf cell (clock sinks) is reached
- Check the clock tree exceptions are not in the clock tree
- Report the pre-existing cells, such as clock gating cells

- Do quality of report (QoR)
- Check clock tree converges either with itself or with another clock tree
- Clock tree has timing relationship with other clock trees for inter clock skew balancing
- Check design rule constraints
- Report power and area

### **Post CTS optimization:**

- Optimization with useful skew to meet setup and hold time
- Optimization with total negative slack (TNS)
- Post CTS optimization techniques
  - Shielding
  - Sizing
  - Buffer re-location
  - Level adjustment
- Optimize the design for Hold time.
  - Hold violations should be fixed first in best corner and then in worst corner

### **CTS output**

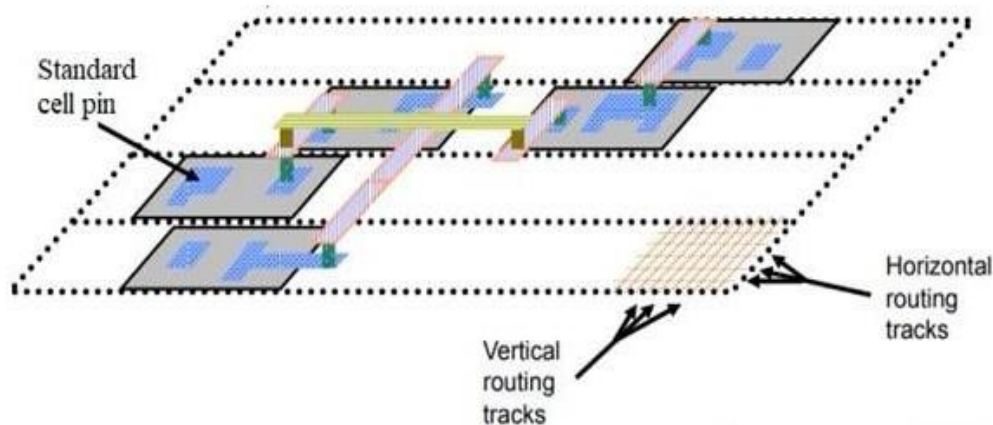
- Timing report
- Congestion report
- Skew report
- Insertion delay report
- CTS DEF file

### **Checks after CTS:**

- In latency report check is skew is minimum? And insertion delay is balance or not.
- In QoR report check is timing (hold) met, if not why?
- In utilization report check standard cell utilization is acceptable or not?
- Check global route congestion?
- Check placement legality of cells.
- Check whether the timing violations are related to the constrained paths or not like defining false paths, asynchronous paths, half cycle paths, multi-cycle paths in the design.

## **ROUTING**

- Making physical connections between signal pins using metal layers are called routing. Routing is the stage after CTS and optimization where exact paths for the interconnection of standard cells
- Electrical connections using metals and vias are created in the layout, defined by the logical connections present in the netlist ( i.e., logical connectivity is converted into physical connectivity)
- After CTS, we have information of all the placed cells, blockages, clock tree buffers / inverters and I/O pins. The tool relies on this information to electrically complete all connections defined in the netlist such that :
  - There are minimal DRC violations while routing
  - The design is 100% routed with minimal LVS violations
  - There are minimal SI (signal integrity ) related violations
  - There must be no or minimal congestion hot spots
  - The timing DRCs and QoR are met and good respectively



### **Routing inputs**

- Netlist
- All cells & ports should be legally placed with clock tree structure
- NDRs (Non Default Routing) rules
- Routing blockages
- Technology data ( metal layers (.lef, .tf etc.,) DRC rules, via creation rules, grid rules

### **Routing goals**

- Minimize the total interconnect / wire length
- Minimize the critical path delays
- Minimize the number of layer changes that the connections have to make ( minimizing the number of vias)
- Complete the connections without increasing the total area of the block

- Meeting the congestion hotspots
- SI driven : reduction in cross-talk noise and delta delays

## **Routing constraints**

- Set constraints to number of layer to be used during routing
- Setting limits on routing to specific regions
- Setting the maximum length for the routing wires
- Blocking routing in specific regions
- Set stringent guidelines for minimum width and minimum spacing
- Set preferred routing directions to specific metal layers during routing
- Constraining the routing density
- Constraining the pin connections

## **Routing flow**

The different tasks that are performed in the routing stage are as follows :



## **Global / Global Routing**

- Identifying routable path for the nets driving / driven pins in a shortest distance
- Does not consider DRC rules, which gives an overall view of routing and congested nets
- Assign layers to nets
- Identify and assign net segments over the specific routable window called Global Route Cell (GRC)
- Avoid congested areas and also long detours



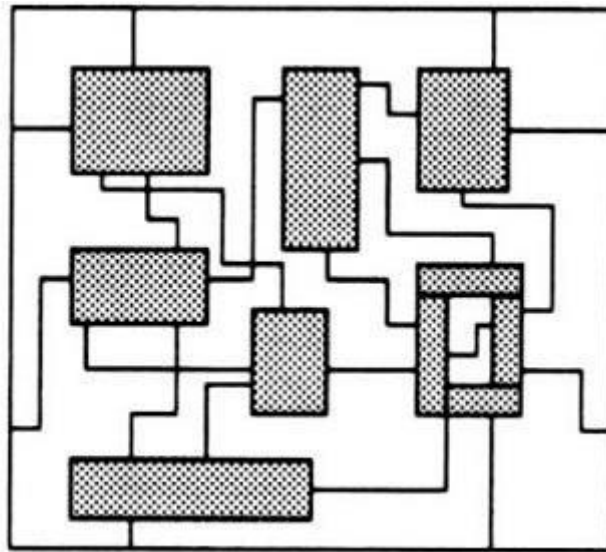
- Avoid routing over blockages
- Avoid routing for pre-route nets such as rings/stripes/rails
- Uses Steiner Tree and Maze algorithm

### **Track assignment**

- Takes the global routed layout and assigns each nets to the specific tracks and layer geometry
- It does not follow the physical DRC rules
- It will do the timing aware track assignment
- It helps in via minimization

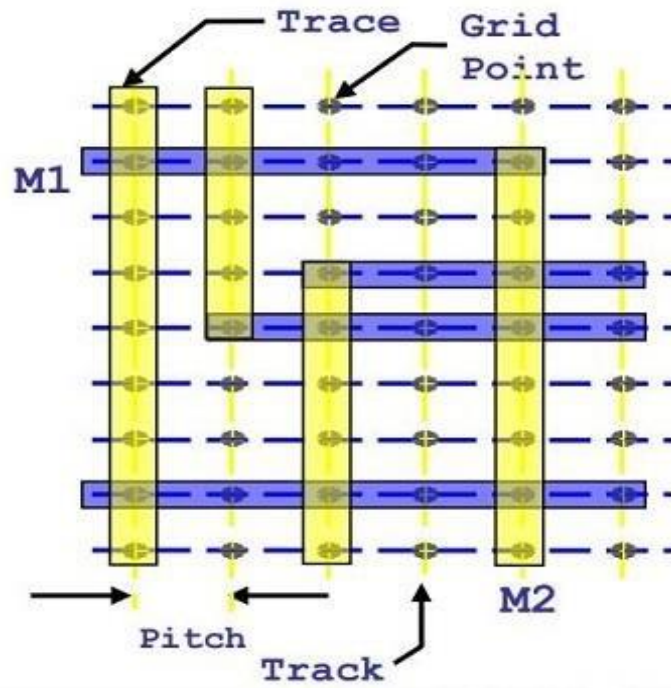
### **Detail / Nano Routing**

- Detailed routing follows up with the track routed net segments and performs the complete DRC aware and timing driven routing.
- It is the final routing for the design, built after the CTS and timing is freeze
- It routes Metal layers on tracks so that it maintains minimum spacing
- During routing it continuously checks for DRC violations and fixes it in iterative steps



### **Grid based Routing**

- Metal traces (routes) are built along and centred upon routing tracks on the grid points
- Various types of grids are manufacturing grid, routing grid (pitch) and placement grid
- Grid dimension should be multiple of manufacturing grid

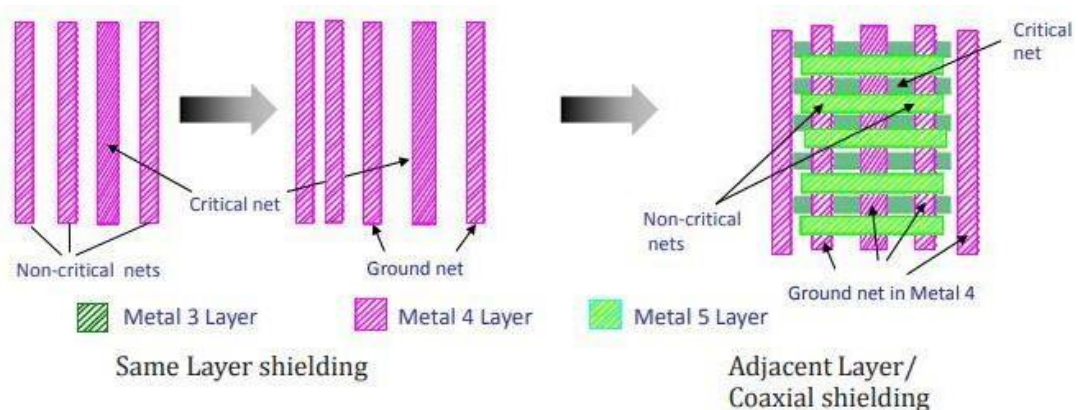


### Search and repair :

- The search and repair stage is performed during detailed routing after the first iteration. In search and repair, shorts and spacing violations are located and rerouting of affected areas to fix all possible violation is executed

### Post routing optimization

- Re-routing of time critical nets
- Signal integrity (SI) optimization by NDRs and shielding for the sensitive nets
- Types of shielding for sensitive nets
  - Same layers shielding
  - Adjacent layer / coaxial shielding



## **Filler cell insertion**

- Filler cells can be inserted before or after detailed routing
- If fillers contain metal routing other than pre-routing then fillers should be inserted before routing
- Width of the smallest filler cell is the placement grid width
- Once fillers are inserted then the placement is fixed and tool can't move cells for further optimization

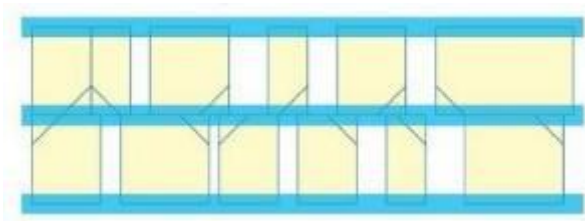


Fig. Before filler cell insertion

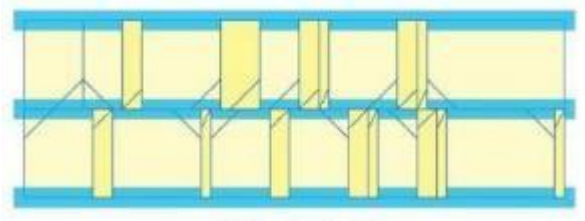


Fig. After filler cells are inserted

## **Metal fill**

- Filling up the empty metal tracks with metal shapes to met metal density rules
- Two types of metal fill :
  - Floating metal fill : doesn't completely shield the aggressor nets, so SI will be prominent
  - Grounded metal fill : completely shields the aggressor nets, so less SI impact. This is complex as compared to floating metal fill
- Metal density rule helps to avoid over etching / metal erosion

# **PHYSICAL DESIGN VERIFICATION**

## **Design Rule Check (DRC)**

- Design Rule Check (DRC) is the process of checking physical layout data against fabrication-specific rules specified by the foundry to ensure successful fabrication.
- Process specific design rules must be followed when drawing layouts to avoid any manufacturing defects during the fabrication of an IC.
- Violating a design rule might result in a non-functional circuit or low Yield.

There are many design rules at different technology nodes, a few of which are mentioned below.

### Types of DRC

- i) Base level DRC : Here DRC is checked for geometries inside transistors

Main Checks are

- Well spacing, Poly spacing and poly width check
- Tap cell requirement check
- Well continuity check

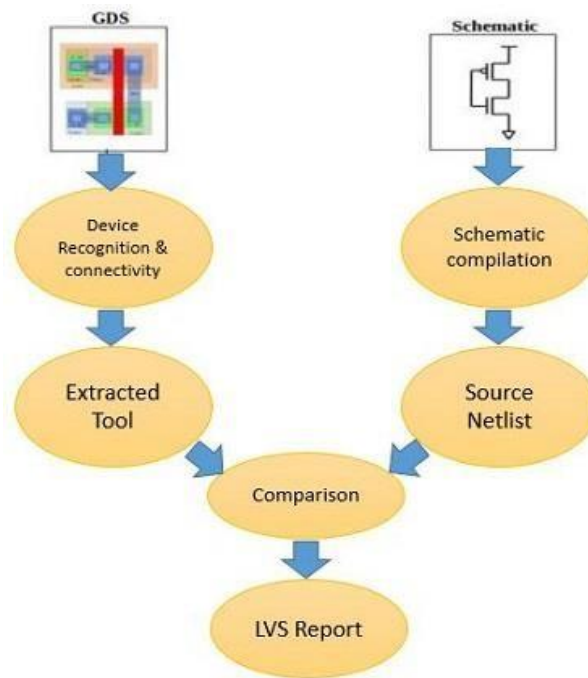
- ii) Metal level DRC : It is checked on all routing layers and vias

Types of DRCs:

- Minimum width and spacing for metal
- Minimum width and spacing for via
- Fat wire Via keep out Enclosure
- End of Line spacing
- Minimum area
- Different net spacing
- Shorts violation
- Antenna Violation

## **Layout Versus Schematic (LVS)**

- Layout Versus Schematic (LVS) verifies the connectivity of a Verilog Netlist and Layout Netlist (Extracted Netlist from GDS)
- Tool extracts circuit devices and interconnects from the layout and saved as Layout Netlist (SPICE format)
- As LVS performs comparison between 2 Netlist, it does not compare the functionalities of both the Netlist.
- Input Requirements
  - LVS Rule deck
  - Verilog Netlist
  - Physical layout database (GDS)
  - Spice Netlist (Extracted by the tool from GDS)



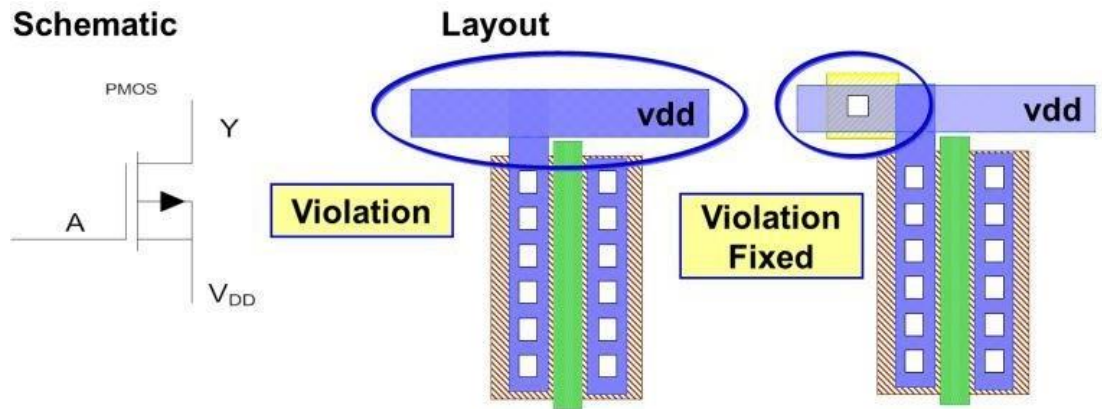
### LVS Flow

**LVS rule deck** is a set of code written in Standard Verification Rule Format (SVRF) or TCL Verification Format (TVF). It guides the tool to extract the devices and the connectivity of IC's. It contains the layer definition to identify the layers used in layout file and to match it with the location of layer in GDS. It also contains device structure definitions.

- LVS checks examples  
Short Net Error, Open Net Error, Extract errors, Compare errors

### Electrical Rule Check (ERC)

- Electrical Rule Check (ERC) is used to analyze or confirm the electrical connectivity of an IC design
- ERC checks are run to identify the following errors in layout
  - To locate devices connected directly between Power and Ground
  - To locate floating Devices, Substrates and Wells
  - To locate devices which are shorted
  - To locate devices with missing connections
- Well Tap connection error: The Well Taps should bias the Wells as specified in the schematics



- Well Tap Density Error: If there is not enough Taps for a given area then this error is flagged
- Taps need to be placed regularly which biases the Well to prevent Latch-up  
e.g., In typical 28nm process the Well Tap Density Rule requires Well-taps to be placed every 50 microns

