PHYSICAL DESIGN

CLOCK TREE SYNTHESIS

CTS : CTS is the process of connecting the clocks to all clock pin of sequential circuits by using inverters/buffers in order to balance the skew and to minimize the insertion delay.

Clock is not propagated before CTS so after clock tree build in CTS stage we consider hold timings and try to meet all hold violations.

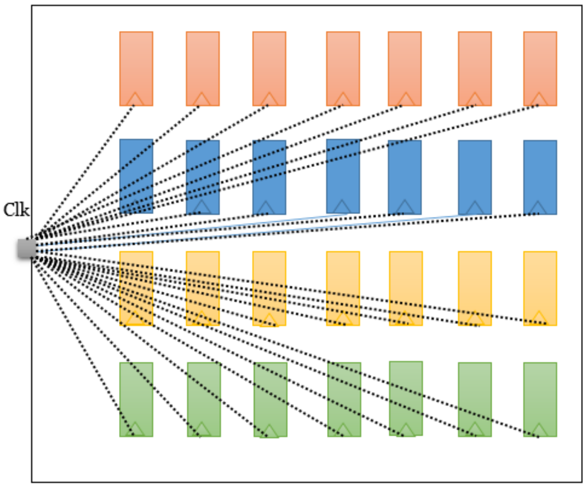


fig: before the clock tree is not build

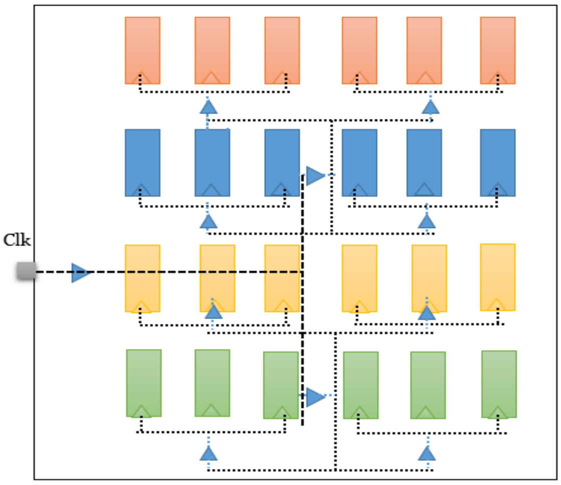


fig: CTS structure after clock tree build

❖ Clock Tree Routing Algorithms :

The main idea behind using these algorithms to minimize the skew. So how we minimize the skew by using these algorithms. Distribute the clock signal in such a way that the interconnections (routing wires) carrying the clock signal to the other sub-blocks that are equal in length.

Several algorithms exist that are trying to achieve this goal (to minimize the skew).

- H-Tree
- X-Tree
- Method of Mean and Median
- Geometric Matching Algorithms
- Zero skew clock routing

The first to four algorithm techniques are trying to make minimize the length and the last one is to use the actual interconnect delay in making the skew is zero.

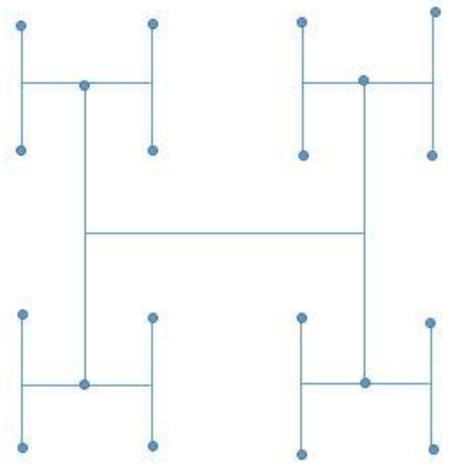
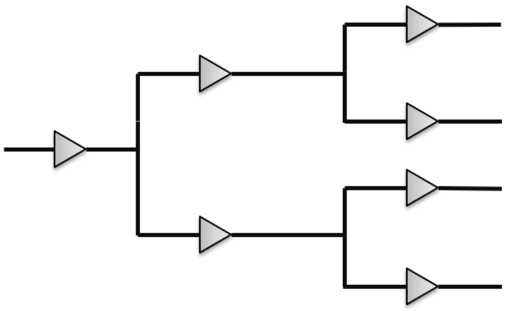


fig: H tree with 16 sink points



π *Configuration*

Zero Skew Clock routing

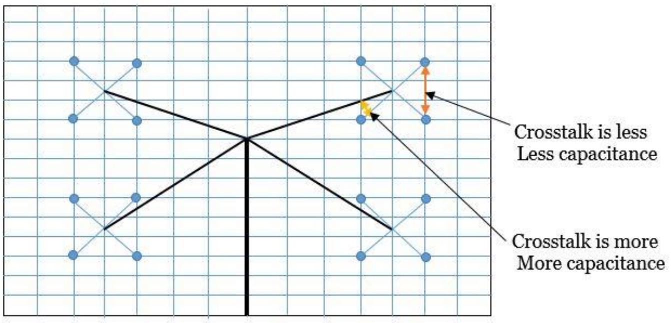


Fig: X tree with 16 points

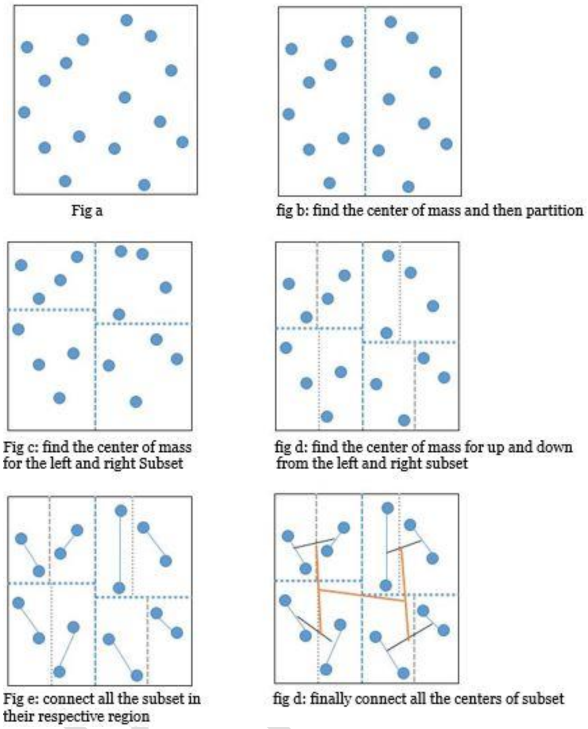


Fig a

fig b: find the center of mass and then partition

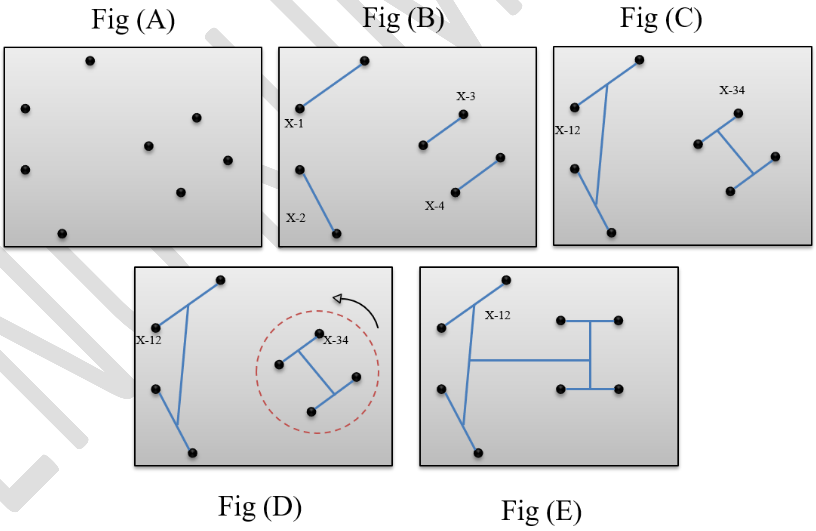
Fig c: find the center of mass for the left and right Subset

fig d: find the center of mass for up and down from the left and right subset

Fig e: connect all the subset in their respective region

fig d: finally connect all the centers of subset

Fig : MMM algorithm



Geometric Matching Algorithms

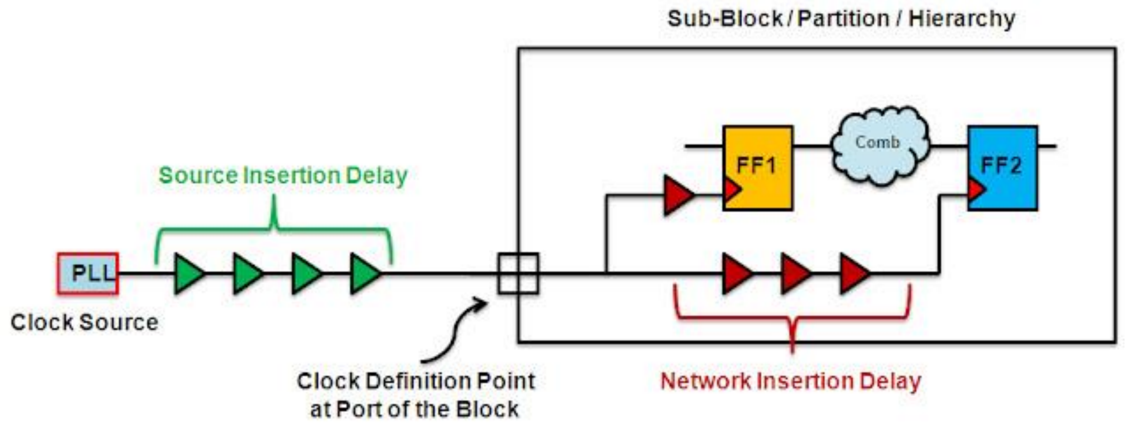
❖ Clock Latency/Insertion Delay:

The time taken by the clock to reach the sink point from the clock source is called Latency.

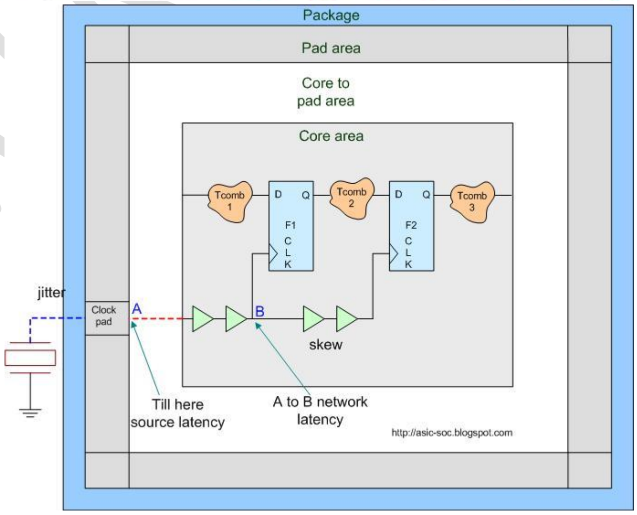
- It is divided into two parts – Clock Source Latency and Clock Network Latency.

➢ **Clock Source Latency**: - The delay between the clock waveform origin point(clock source) to the definition point/ clock port.

➢ **Clock Network Latency**: - The delay from the clock definition point(clock port) to the destination/sink point.



Total Latency = Source Insertion Delay + Network Insertion Delay



- **Set_clock_latency 0.8 [get_clocks clk_name1] ----> network latency**
- **Set_clock_latency 1.9 –source [get_clocks clk_name1] -------> source latency**

❖ Skew : report_clock_timing -type skew

- Skew is the difference in arrival of clock at two consecutive pins of a sequential element is called skew.
- Clock skew is the variation at arrival time of clock at destination points in the clock network. The difference in the arrival of clock signal at the clock pin of different flops.

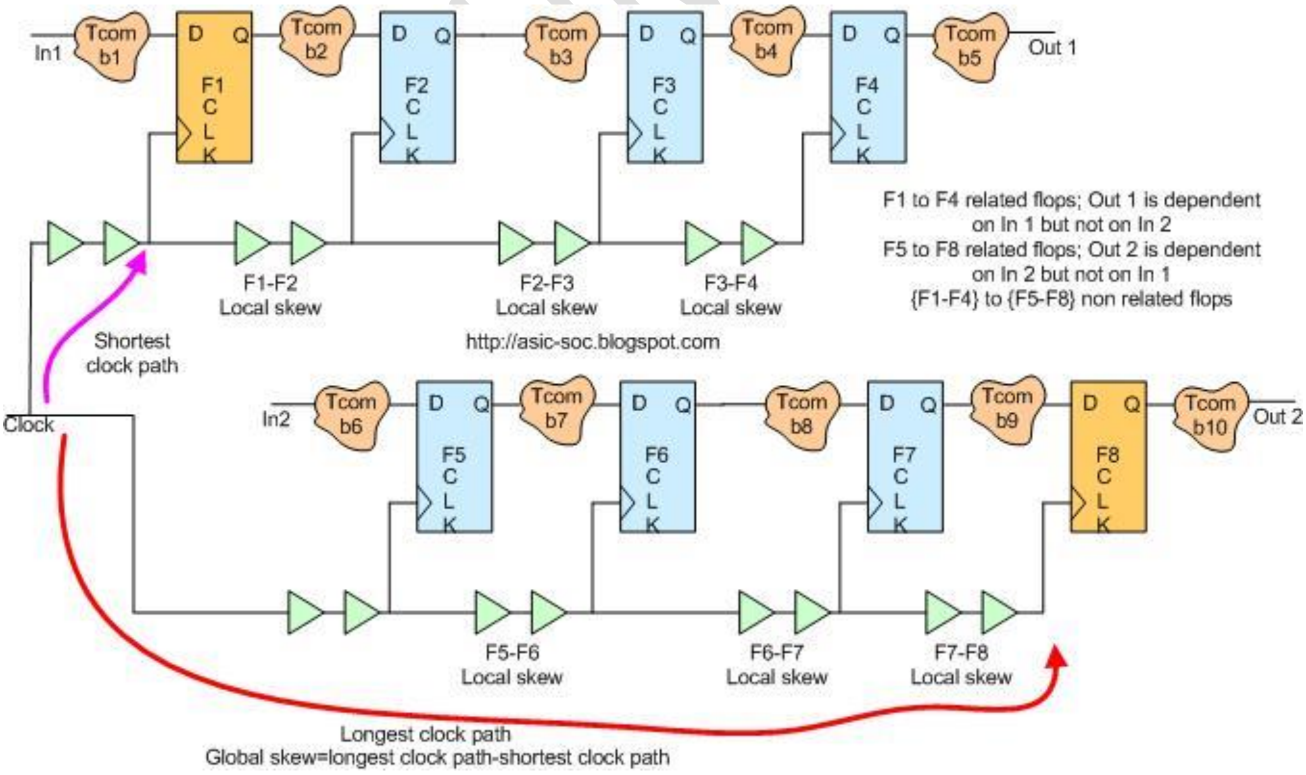
Useful skew: if the clock is skewed intentionally to resolve setup violations.

Two types of skews are defined: Local skew and Global skew.

➢ **Local skew :**
Local skew is the difference in the arrival of clock signal at the clock pin of related flops.

➢ **Global skew :**

- Global skew is the difference in the arrival of clock signal at the clock pin of non related flops.
- This also defined as the difference between shortest clock path delay and longest clock path delay reaching two sequential elements.



F1 to F4 related flops; Out 1 is dependent on In 1 but not on In 2
F5 to F8 related flops; Out 2 is dependent on In 2 but not on In 1
{F1-F4} to {F5-F8} non related flops

http://asic-soc.blogspot.com

Global skew=longest clock path-shortest clock path

➢ Positive Skew :

- If capture clock comes late than launch clock then it is called +ve skew.
- Clock and data both travel in same direction.
- When data and clock are routed in same direction then it is **Positive skew.**
- +ve skew can lead to hold violation.
- +ve skew improves setup time.

➢ Negative Skew

- If capture clock comes early than launch clock it is called –ve skew.
- Clock and data travel in opposite direction.
- When data and clock are routed in opposite then it is **negative skew.**
- -ve skew can lead to setup violation. -ve skew improves hold time.

❖ Uncertainty :

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

It can define for both setup and hold.

```
Set_clock_uncertainty –setup 0.2 [get_clocks clk_name1]
```

```
Set_clock_uncertainty –hold 0.05 [get_clocks clk_name1]
```

Pre CTS : uncertainty = skew + jitter + design margin

post CTS : for setup : uncertainty = jitter + design margin

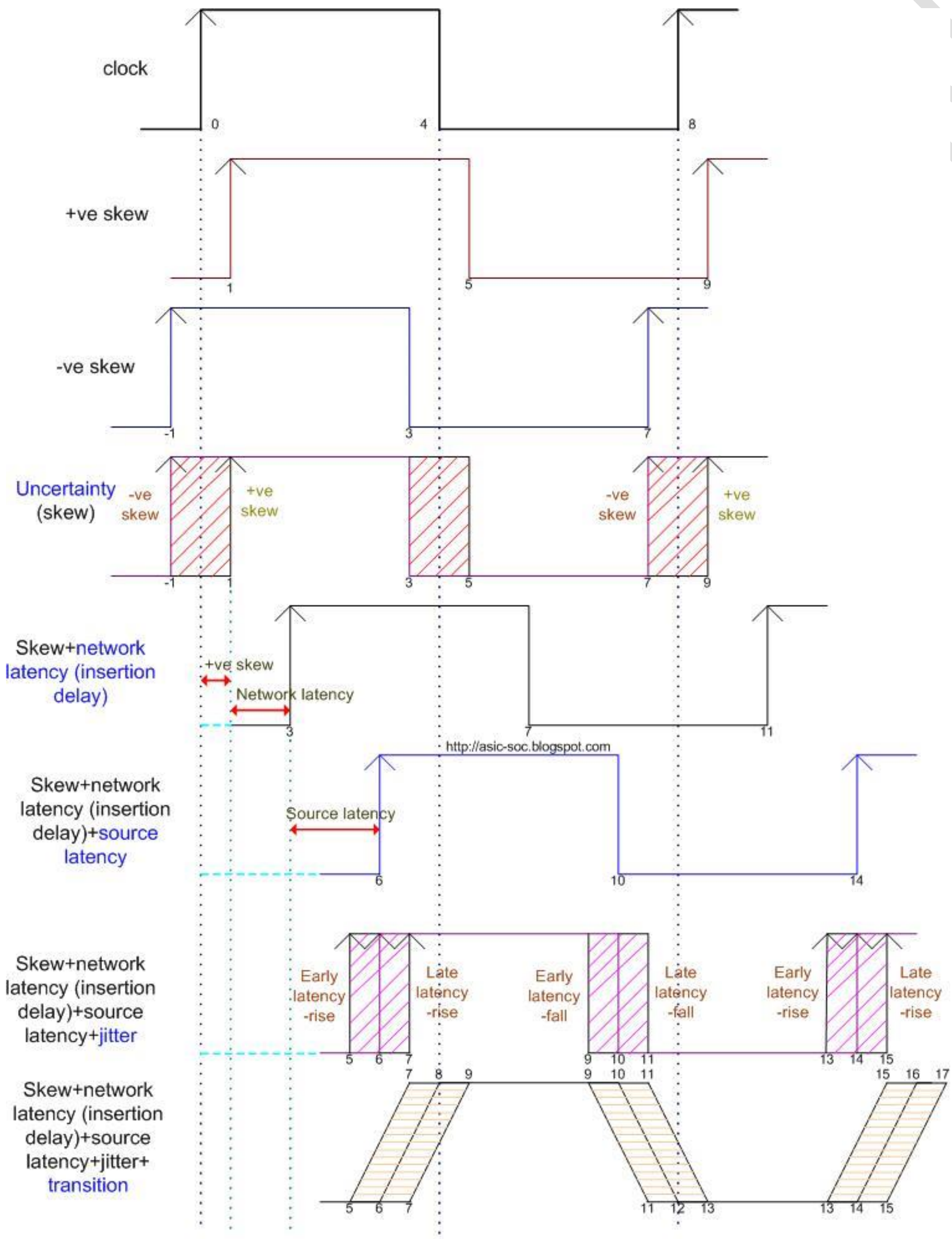
for hold : uncertainty = design margin

❖ Jitter

- Jitter is the short-term variations of a signal with respect to its ideal position in time.

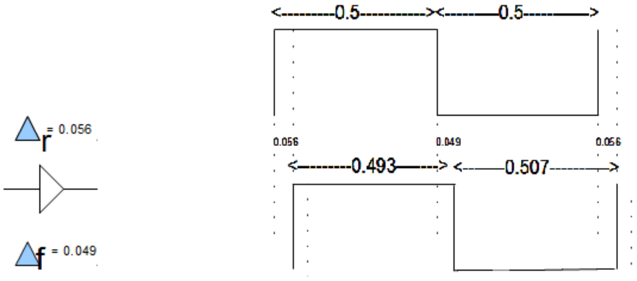
Sources of Jitter Common sources of jitter include:

- Internal circuitry of the phase-locked loop (PLL)
- Random thermal noise from a crystal
- Other resonating devices



Min Pulse Width : report_timing –check_type pulse_width

Time between active (ON) and inactive (OFF) states of same signal.



- **Min High Pulse width:** After raising edge of signal the min amount of time signal must be stable (0.493) .
- **Min Low Pulse Width :** After falling edge of signal the min amount of time signal must be stable (0.0.507) .

In SDC file (.sdc):set_min_pulse_width -high 5 [get_clock clk1]

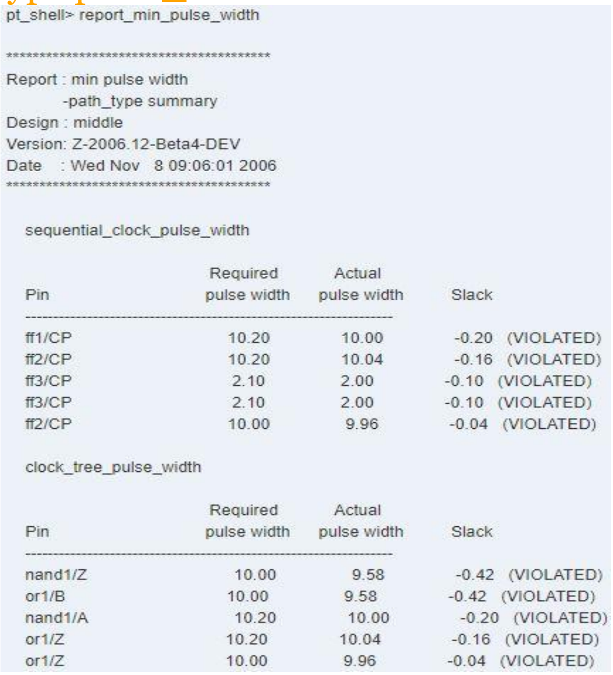
set_min_pulse_width -low 4 [get_clock clk1]

➢ **Why the minimum pulse width violation occurs:**

Due to unequal rise and fall delay of combinational cell. Let’s take an example of buffer and clock signal having 1 GHz frequency (1ns period) is entering into a buffer. So for example, if the rise delay is more than the fall delay than the output of clock pulse width will have less width for high level than the input clock pulse.

How to report:

report_timing –check_type pulse_width



```
pt_shell> report_min_pulse_width

****************************************
Report : min pulse width
         -path_type summary
Design : middle
Version: Z-2006.12-Beta4-DEV
Date   : Wed Nov  8 09:06:01 2006
****************************************

  sequential_clock_pulse_width

                          Required      Actual
  Pin                     pulse width   pulse width     Slack
  ------------------------------------------------------------
  ff1/CP                    10.20         10.00          -0.20  (VIOLATED)
  ff2/CP                    10.20         10.04          -0.16  (VIOLATED)
  ff3/CP                     2.10          2.00         -0.10  (VIOLATED)
  ff3/CP                     2.10          2.00         -0.10  (VIOLATED)
  ff2/CP                    10.00          9.96          -0.04  (VIOLATED)

  clock_tree_pulse_width

                          Required      Actual
  Pin                     pulse width   pulse width     Slack
  ------------------------------------------------------------
  nand1/Z                    10.00          9.58          -0.42  (VIOLATED)
  or1/B                     10.00          9.58         -0.42  (VIOLATED)
  nand1/A                    10.20         10.00           -0.20  (VIOLATED)
  or1/Z                     10.20         10.04          -0.16  (VIOLATED)
  or1/Z                     10.00          9.96         -0.04  (VIOLATED)
```

❖ Clock Buffer VS Normal Buffer

Clock net is one of the High Fanout Net(HFN)s.

- **Clock Buffers** are designed with some special property like high drive strength and less delay.Clock buffers have equal rise and falltime. This prevents duty cycle of clock signal from changing when it passes through a chain of clock buffers.

- **Normal buffers** are designed with W/L ratio such that sum of rise time and fall time is minimum.They too are designed for higher drive strength.

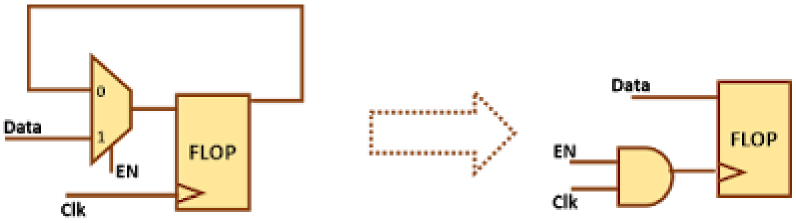
➢ Synchronous & Asynchronous Clocks:

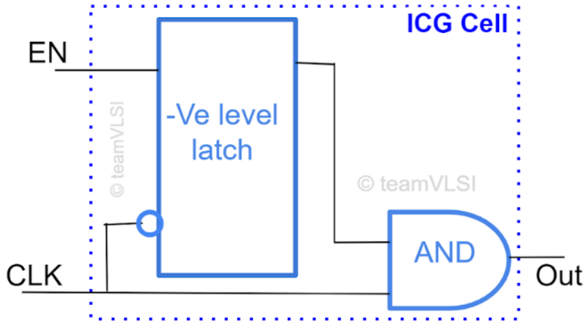
• Two clocks are synchronous with respect to each other if they share a common source and have a fixed phase relationship and a common base period(should have a common multiple).
Ex: time period of two clocks : 2 and 6, here the common base period is 2.

• Two clocks are said to be asynchronous if they do not have a fixed phase relationship with each other in the design and don't have a common base period.

❖ clock gating:

Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation. Clock gating saves power by adding more logic to a circuit to prune the clock tree. Pruning the clock disables portions of the circuitry so that the flip-flops in them do not have to switch states. Switching states consumes power. When not being switched, the switching power consumption goes to zero, and only leakage currents are incurred.



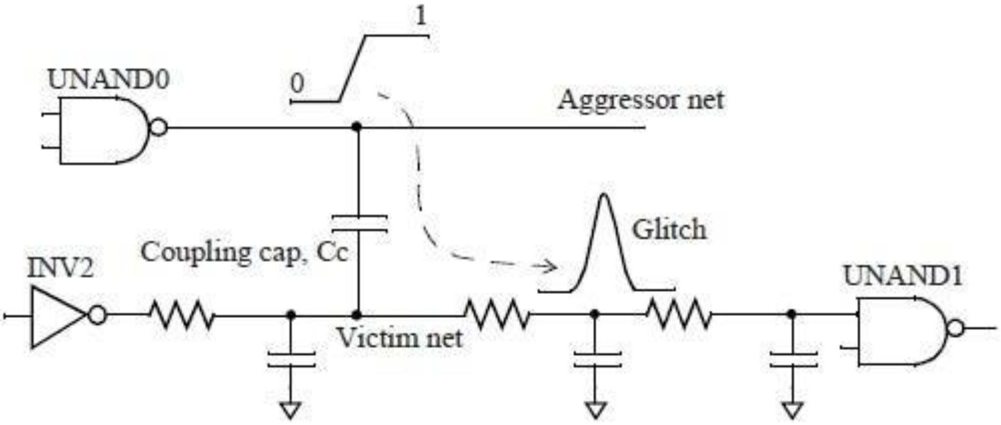


❖ CROSSTALK :

- Crosstalk is the udesirable electrical intraction between two or more adjacent nets due to coupling capcitance.
- Switching of the signal in one net (aggressor) can interfere neighbouring net (victim) due to cross cross coupling capacitance this is called crosstalk.

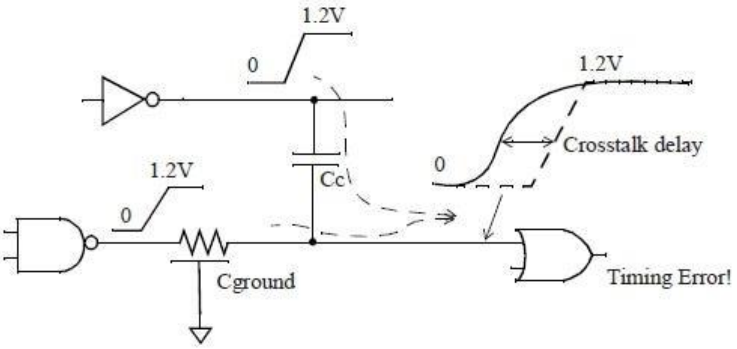
➢ Cross talk is two types :

1. **Crosstalk Noise :** Agresser net is switching and victim net is constant low or high

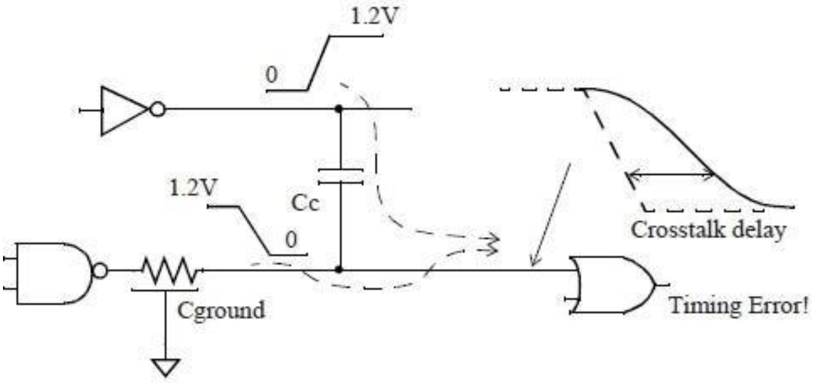


2. **Crosstalk Delay :** Both aggressor and victim are switching .

- **Negative crosstalk delay**: If the aggressor and victim nets are switching in the same direction, it results in a smaller delay for the victim net. The reduction in delay is known as a negative crosstalk delay.



- **Positive crosstalk delay**: If the aggressor net is switching in the opposite direction of the victim net, it results in a larger delay for the victim net. The increase in delay is known as a positive crosstalk delay.



How to reduce the crosstalk :

- Wire spacing (NDR rules) by doing this we can reduce the coupling capacitance between two nets.
- Increased the drive strength of victim net and decrease the drive strength of aggressor net
- Jumping to higher layers (because higher layers have width is more)
- Insert buffer to split long nets
- shielded layers are connects to either VDD or VSS. The coupling capacitance remains constant with VDD or VSS.

CTS

➢ Before CTS we need to check:

- All cells should be legalized.
- All power nets are prerouted.
- All pins should legalized.
- Congestion, timing should control.

➢ GOALS OF CTS:

- Balancing the skew.
- Minimum Insertion Delay
- Redue power consumption
- Reduce crosstalk
- Reduce EM
- Balancing load across different clocks

➢ INPUTS OF CTS:

1. SDC
2. SPEC FILE
3. PLACEMENT DATABASE (netlist, physical and logical library's)

SDC (Synopsys design constraint) :

➢ Clock definition :

- Create clock
- Create virtual clock
- Create generated clock
- Create clock uncertainty

➢ External delays

- Input delays
- Output delays

➢ DRV's

- Max tran
- max cap
- max fanout

➢ Timing path exceptions

- False path
- Multi cycle path
- Max delay
- Min delay

➢ What spec file contains :

spec file contains:

- Clock name
- Clock period
- Max skew and latency target values
- Clock buffers and clock inverters information
- NDR rules
- Clock tree Exceptions
- Information about Metal layers used
- Sink max tran

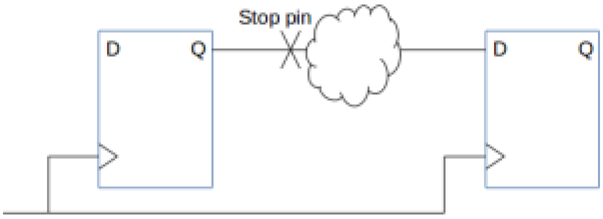
- ✓ create_clock [get_ports{clk}] –name pclk –period 10 –waveform{4 8}
- ✓ Create_generated_clock –name clk –source PLL –divided_by 2 [get_pins uff0/q]

➢ Clock Tree Exceptions:

- Stop Pin
- Non- Stop Pin
- Exclude Pin
- Float Pin
- Don’t Touch pin

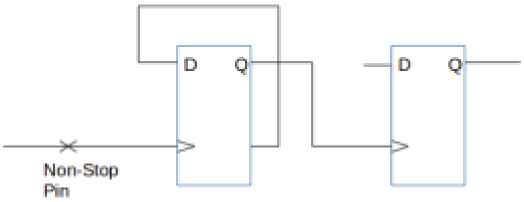
Stop Pin:

- Stop pins are the endpoints of clock tree that are used for delay balancing.
- CTS, the tool uses stop pins in calculation & optimization for both DRC and clock tree timing.
- Example:
  - Clock sink are implicit stop pins.



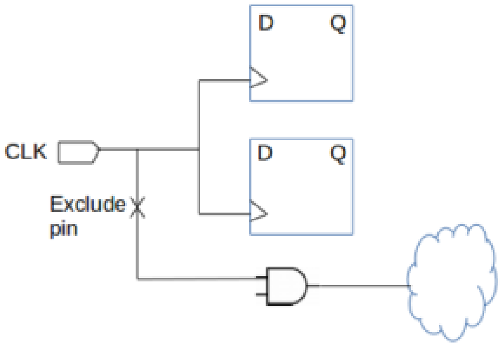
Non- Stop Pin:

- Nonstop 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.



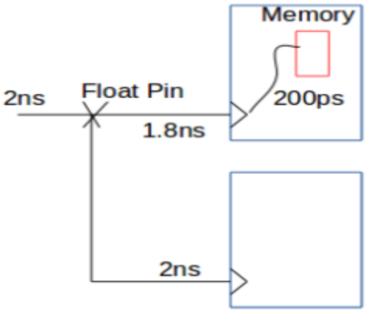
Exclude pin:

- Exclude pin are clock tree endpoints that are excluded from clock tree timing calculation and optimization.
- The tool considers exclude pins only in calculation and optimizations for design rule constraints



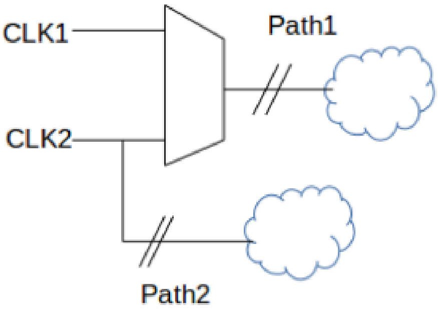
Float Pin:

- Float pins are clock pins that have special insertion delay requirements and balancing is done according to the delay.[Macro modelling].



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.

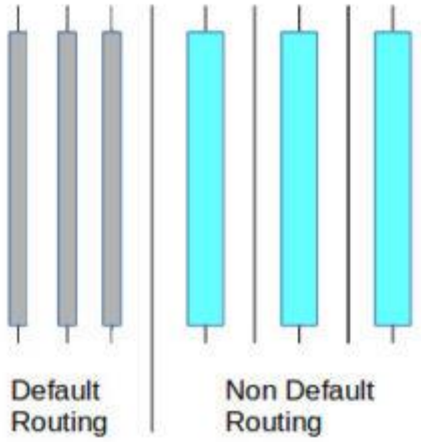


Non Default Rules?

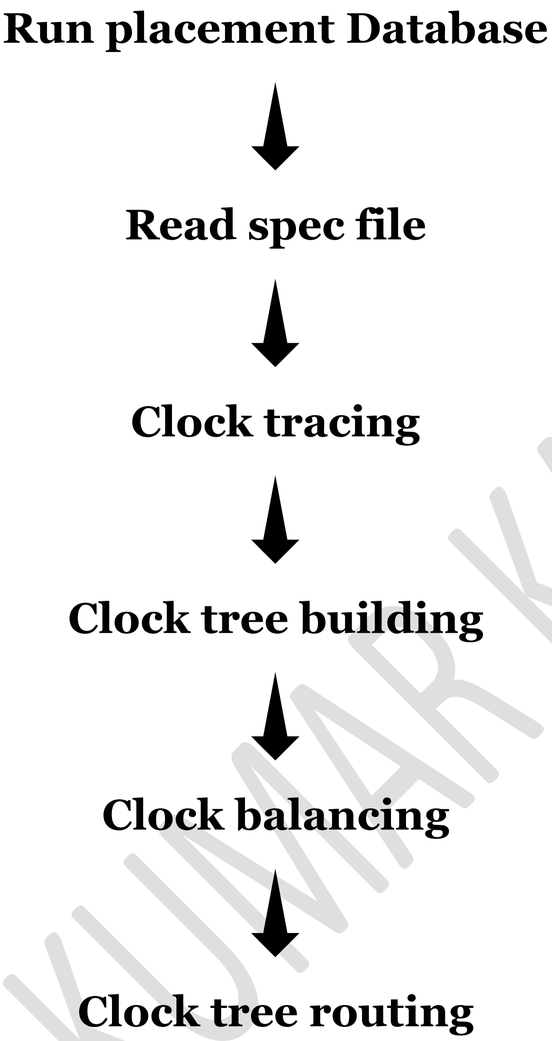
- Double width and double space.
- After PNR stage if u will get timing /crosstalk/noise violations which are difficult to fix at ECO stage we can try this NDR option at routing stage.

Double spacing: It is used to avoid the crosstalk.

Double width: It is used to avoid the EM.



CTS Flow:



Run placement Database

Read spec file

Clock tracing

Clock tree building

Clock balancing

Clock tree routing

Some Commands Used in CTS :

- set a maximum transition time of 0.20 ns on all pins in the CLK clock path for all corners of the current mode, use the following command:

```
icc2_shell> set_max_transition 0.20 -clock_path [get_clocks CLK]
```

- The following example creates a skew group named sg1 consisting of sinks reg1/CP, reg2/CP, and reg3/CP:

```
icc2_shell> create_clock_skew_group -name sg1  -objects {reg1/CP reg2/CP
reg3/CP}
```

- To define a clock balance group named group1 that contains the clocks named clk1 and clk2 in the current mode, use the following command:

```
icc2_shell> create_clock_balance_group -name group1

\ -objects [get_clocks{clk1 clk2}
```

- If you want the insertion delay of clk1 and clk2 to be the same and the insertion delay of clk3 to be 100 less than that of clk1 and clk2, use the following command:

  ```
   icc2_shell> create_clock_balance_group -name group3

     \ -objects [get_clocks{clk1 clk2 clk3}]

    \ -offset_latencies {0 0 -100}
  ```

- **Routing Clock Trees** After synthesizing and optimizing the clocks, you can detail route the clock nets by using the route_group command, as shown in the following example:

```
            icc2_shell> route_group -all_clock_nets -reuse_existing_global_route
true
```

- Synthesize the entire clock tree, from the clock root, by using the following command:

  ```
   icc2_shell> clock_opt -from build_clock -to route_clock
  ```

- the following command creates a clock mesh for the net named clk1_mesh that is bounded by coordinates (0,0) and (1200, 980). The straps are on layers M7 and M8 with a width of 2.4 units and of type stripe.

  ```
  icc2_shell> create_clock_straps-nets [get_nets clk1_mesh] \ -layers
  {M7 M8} -widths {2.4 2.4} -types {stripe stripe} \ -grids {{20 1200 100}
  {60 980 150}}-boundary {{0 0} {1200 980}}
  ```

- To route all clock nets, use the following command:

  ```
  icc2_shell> route_group -all_clock_nets
  ```

- the maximum number of iterations is 40. You can change this limit by setting the -max_number_iterationsoption.

  ```
  icc2_shell> route_detail -max_number_iterations20
  ```

CTS Optimization process:

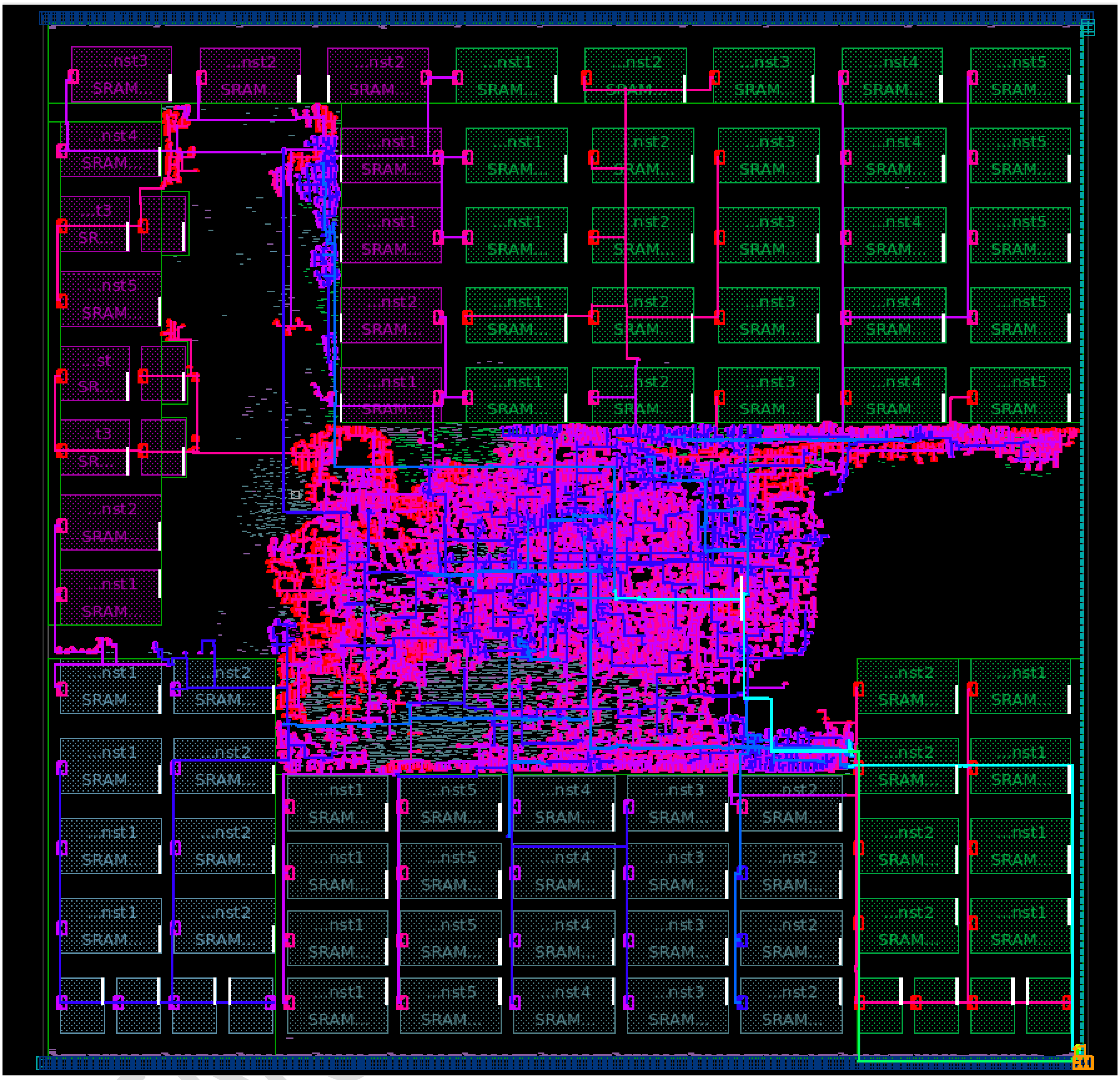
- By buffer sizing
- Gate sizing
- Buffer relocation
- Level adjustment
- HFN synthesis
- Delay insertion
- Fix max transition
- Fix max capacitance
- Reduce disturbances to other cells as much as possible.

➢ After CTS what will you check?

- timing (drv's , setup and hold)
- core area utilization
- Insertion delay (Target have to meet)
- Skew (Target have to meet)
- Routing congestion.
- Placement legality.
- Signal integrity and crosstalk.
- Clock duty cycle.
- Clock tree power consumption.

➢ OutPuts Of CTS :

- CTS def
- Timing reports



- VENU KUMAR KARE

https://www.linkedin.com/in/venu-kumar-kare-465265233?lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base_contact_details%3BN8cQRhOLSaSZ2aBCZEMJXQ%3D%3D