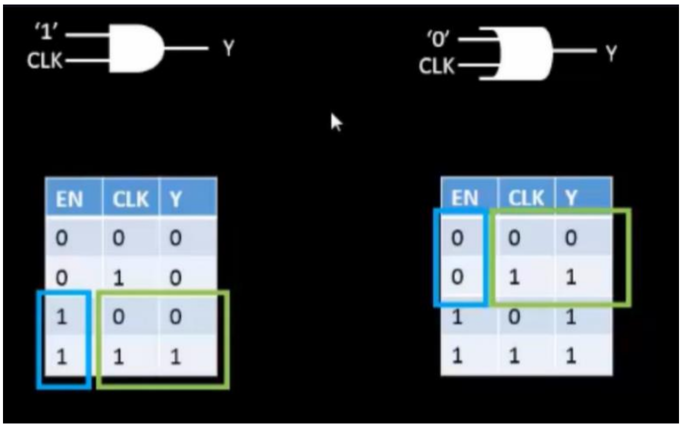
DAY-4 PRE-LAYOUT TIMING ANALYSIS AND IMPORTANCE OF GOOD CLOCK TREE.

1. Timing modelling using delay tables

1: Introduction to delay tables

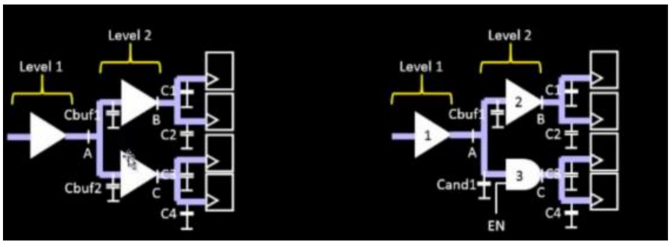
- In the below picture we can see the enable and disable for AND and OR gate.
- It determines if CLK has its influence of the output or not .
- Example: take AND gate enable is 1 and disable is 0 because in enable output is reflected by the clock and disable the output is not reflected by clock



EN	CLK	Y
0	0	0
0	1	0
1	0	0
1	1	1

EN	CLK	Y
0	0	0
0	1	1
1	0	1
1	1	1

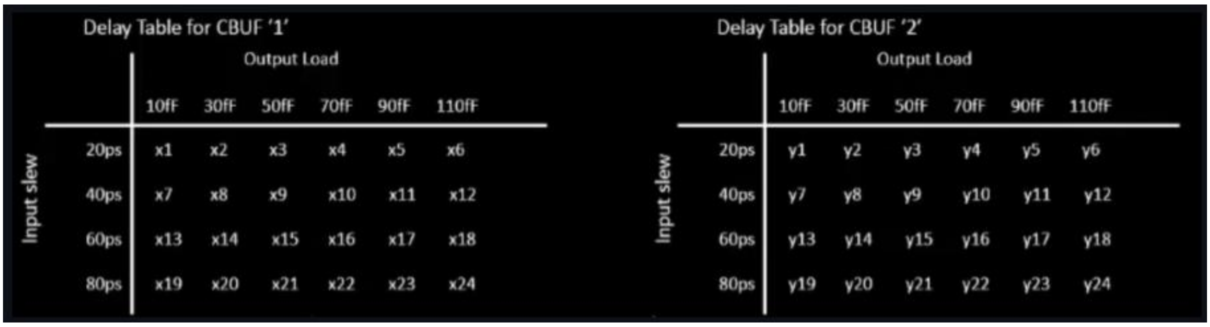
- In the below picture we're comparing a **normal clock buffer tree** (left) vs. a **clock-gated buffer tree** (right), analyzing how replacing a buffer with an AND gate (for gating) affects performance — particularly delay



- **Assumptions Made:**
  - Capacitances:
    - C1 = C2 = C3 = C4 = 25fF (load at the leaf nodes)
    - Cbuf1 = Cbuf2 = 30fF (internal buffer output loads)

- Node capacitances:
    - Node A: 60fF (drives both Cbuf1 and Cbuf2)
    - Node B: 50fF
    - Node C: 50fF

- **What This Means:**
    - **Fanout Is Balanced**:
        - Each buffer at a given level drives equal load → better signal integrity & timing.
    - **Replacing Buffer with AND Gate (Clock Gating)**:
        - In the gated version, a buffer is **replaced by an AND gate** (at node C).
        - This change **alters the delay behavior** due to:
    - Different drive strength of the AND gate.
    - Different logical effort and internal capacitance.



Delay Table for CBUF '1'

Input slew	Output Load 10fF	30fF	50fF	70fF	90fF	110fF
20ps	x1	x2	x3	x4	x5	x6
40ps	x7	x8	x9	x10	x11	x12
60ps	x13	x14	x15	x16	x17	x18
80ps	x19	x20	x21	x22	x23	x24

Delay Table for CBUF '2'

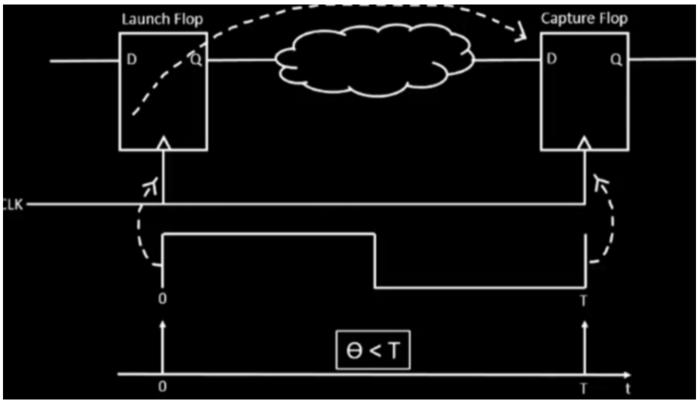
Input slew	Output Load 10fF	30fF	50fF	70fF	90fF	110fF
20ps	y1	y2	y3	y4	y5	y6
40ps	y7	y8	y9	y10	y11	y12
60ps	y13	y14	y15	y16	y17	y18
80ps	y19	y20	y21	y22	y23	y24

2. Timing analysis with ideal clocks using openSTA.

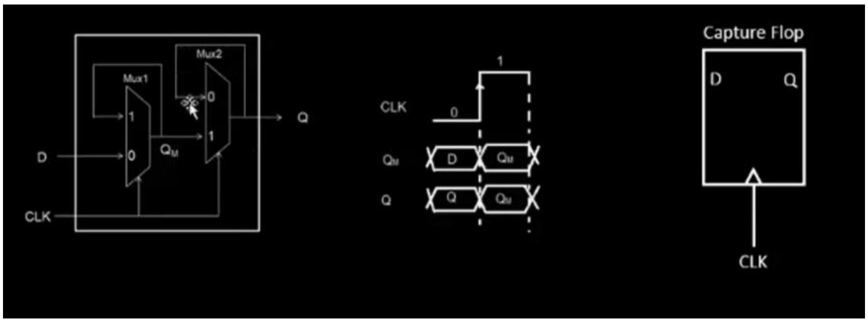
1: Setup timing analysis and introduction to flip-flop setup time

- **System Context**
    - **Clock Frequency**: 1 GHz
    - **Clock Period (T)**: 1 ns
    - **Launch Flop**: Sends data on the rising edge at time 0 ns
    - **Capture Flop**: Receives data on the next rising edge at time T = 1 ns
    - **Combinational Path Delay**: θ (theta)
    - **Clock Assumption**: **Ideal clock**, so **no skew or jitter.**
- **Setup Timing Analysis (Ideal Clock)**
    - Clock frequency: **1 GHz**, so clock period **T = 1 ns**
    - At time 0 ns, a rising clock edge launches data from the **Launch Flop**

    - Data propagates through a **combinational logic path** with delay **θ**
    - The **Capture Flop** samples the data at the next rising edge at time **T = 1 ns**
    - For correct operation:
        - $\Theta < T = 1$ ns
    - This ensures the signal reaches and stabilizes before the capture edge.



- Inside the Flip Flop we have the below



2: Introduction to clock jitter and uncertainty

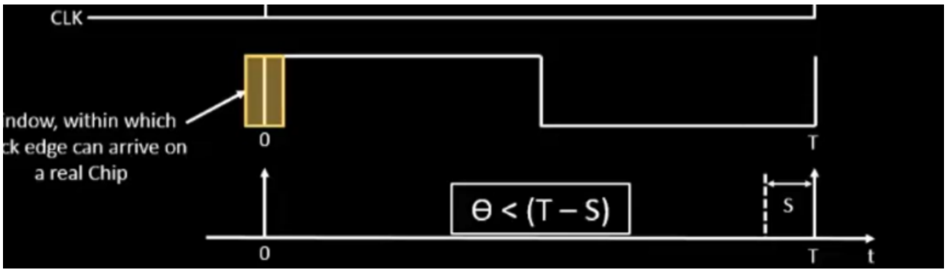
- In our circuit, the clock is generated using a PLL, which is supposed to send clock pulses exactly at 0, T, 2T, etc. But due to internal variations, it might not always hit those exact times — this variation is called jitter.
- Jitter adds **uncertainty** to when the clock edge arrives. We call this **uncertainty time (US)**. So now, when we do setup timing analysis, we need to account for both the setup time of the flip-flop and this uncertainty.
- The new timing condition becomes:

$$\Theta < (T - S - US)$$

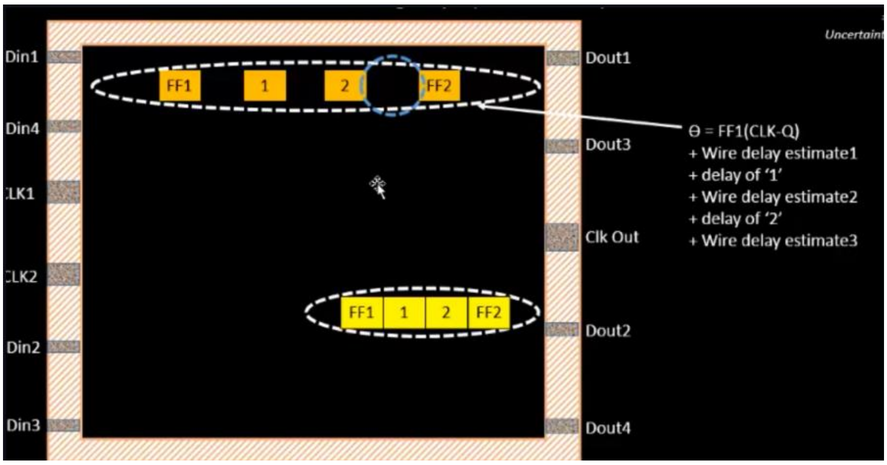
- Assuming:
    - Setup time **S = 0.01 ns**
    - Clock uncertainty **US = 0.09 ns**

- So our max allowed logic delay is:

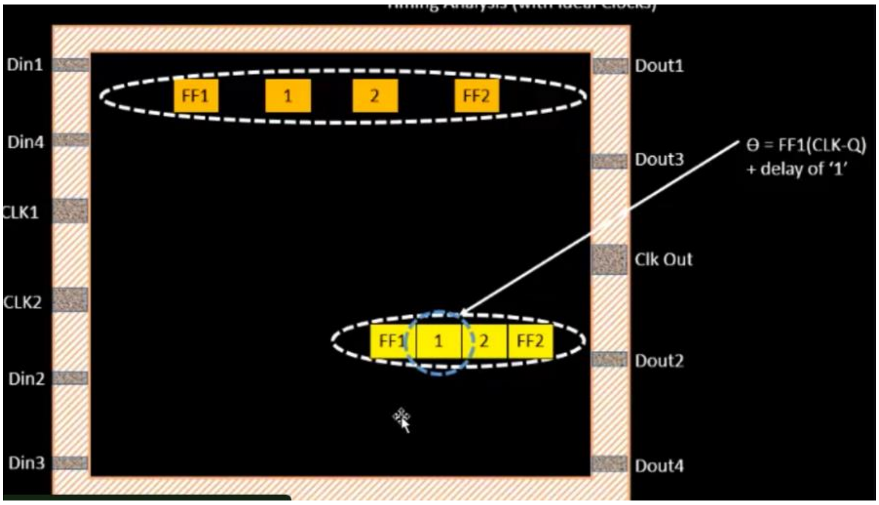
  Θ < 1− 0.01 − 0.09 = 0.9ns



- Identifying combinational path delays for both logics



Θ = FF1(CLK-Q)
+ Wire delay estimate1
+ delay of '1'
+ Wire delay estimate2
+ delay of '2'
+ Wire delay estimate3

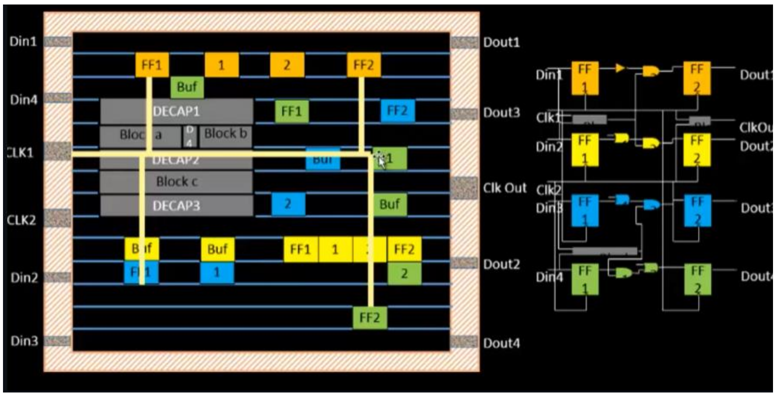


Θ = FF1(CLK-Q)
+ delay of '1'

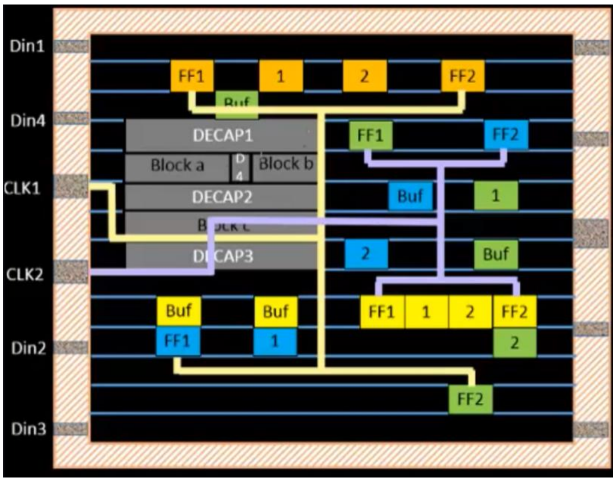
3. Clock tree synthesis TritonCTS and signal integrity.

1. Clock tree routing and buffering using H-Tree algorithm

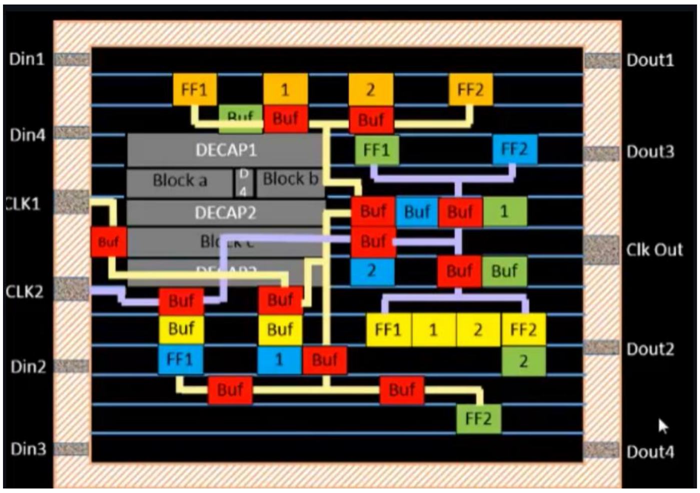
- Using **Clock tree routing and buffering using H-Tree algorithm:** Clock tree routing is the process of distributing the clock signal from the clock source (usually a PLL) to all sequential elements (like flip-flops) in a chip with minimal skew and balanced delay.
- The below picture represents the connection to the FF from one stage to another as follows:
  - Clk1 → stage 1 of FF1 and FF2 is connected.
  - Clk1 → stage 3 FF1 and Stage 2 FF2 is connected.



- But the above picture the connection is too long and becomes a problem leading to increase the skew resulting in bad tree network.
- To make a good clock Tree reduce the physical distance as follows:

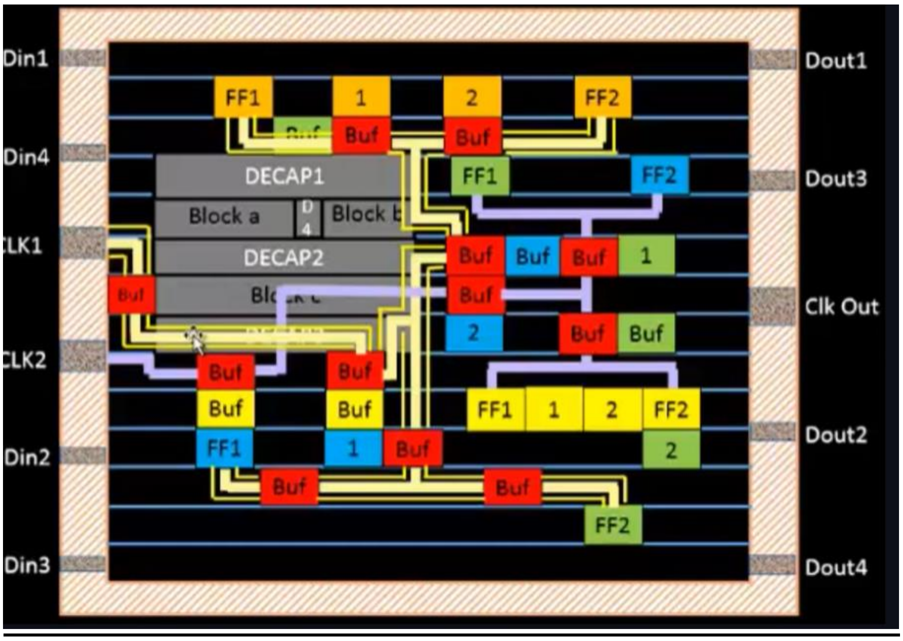


- But this might still create a problem for generating the current output so we connect the buffers/ repeaters to get the optimal output.



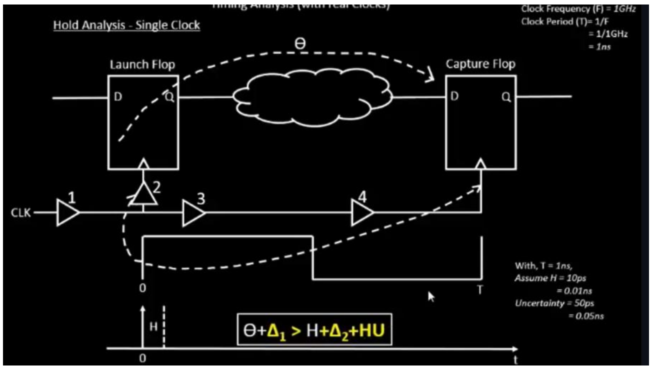
2. Crosstalk and clock net shielding

- Clock net shielding is the critical net scene in the design. We take the particular clock net and shield it means we protect the clock from the outside world



4. Timing analysis with real clocks using openSTA

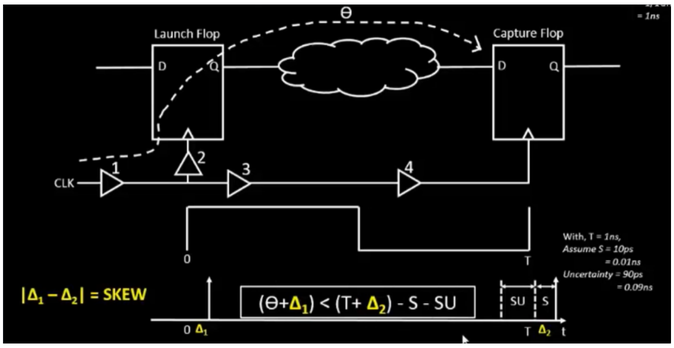
- Hold timing analysis with real clocks



With, T = 1ns,
Assume H = 10ps
= 0.01ns
Uncertainty = 50ps
= 0.05ns

$\Theta+\Delta_1 > H+\Delta_2+HU$

- Setup timing analysis with real clocks

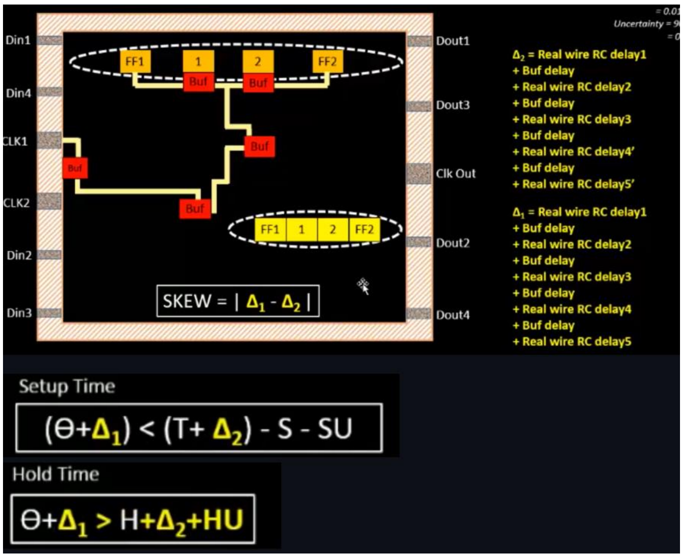


With, T = 1ns,
Assume S = 10ps
= 0.01ns
Uncertainty = 90ps
= 0.09ns

$|\Delta_1 - \Delta_2| = \text{SKEW}$

$(\Theta+\Delta_1) < (T+ \Delta_2) - S - SU$

- The timing paths for single clocks



$\text{SKEW} = |\Delta_1 - \Delta_2|$

Δ_2 = Real wire RC delay1
+ Buf delay
+ Real wire RC delay2
+ Buf delay
+ Real wire RC delay3
+ Buf delay
+ Real wire RC delay4'
+ Buf delay
+ Real wire RC delay5'

Δ_1 = Real wire RC delay1
+ Buf delay
+ Real wire RC delay2
+ Buf delay
+ Real wire RC delay3
+ Buf delay
+ Real wire RC delay4
+ Buf delay
+ Real wire RC delay5

Setup Time

$(\Theta+\Delta_1) < (T+ \Delta_2) - S - SU$

Hold Time

$\Theta+\Delta_1 > H+\Delta_2+HU$