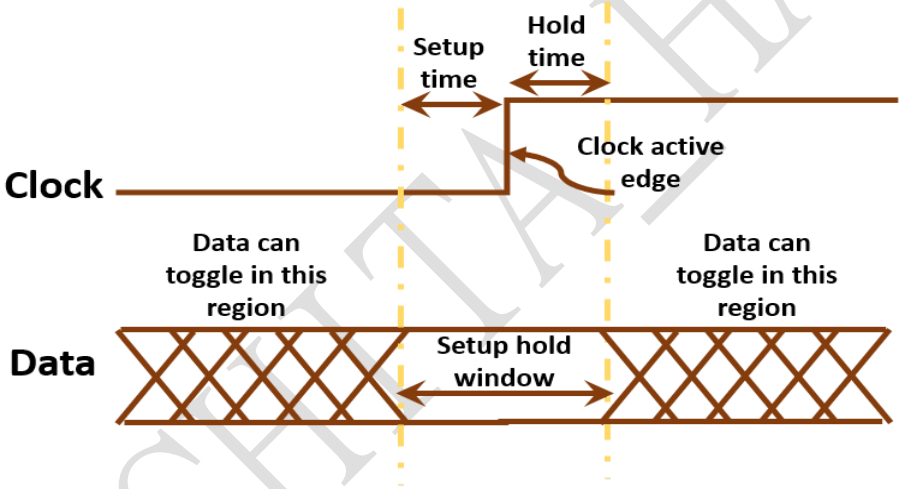
STATIC TIME ANALYSIS

A single timing failure in a 7nm processor can lead to critical system crashes, causing millions in losses

Static Timing Analysis (STA) is a fundamental process in VLSI design used to verify that a digital circuit meets timing constraints.

- ✓ **Contamination delay (t_{cq})** - The **minimum time** after the clock edge when the output **changes** from its previous state.
- ✓ **Propagation delay (t_{pd})** - The **maximum time** after the clock edge when the output **settles** to a stable final value.
- ✓ **Setup Time (t_s):** The **minimum time before the active clock edge** that the input data must **remain stable** to ensure the flip-flop correctly latches it.
- ✓ **Hold Time (t_h):** The **minimum time after the active clock edge** that the input data must **remain stable** to ensure that the flip-flop correctly retains the captured data **without corruption**.



Crucial Question 🤔

Why can hold time be negative, but setup time cannot?

- ✓ **Setup Violation**: A **Setup Violation** occurs when the data is **not stable for at least the setup time** before the active clock edge, the flip-flop **may fail to latch the correct value**, potentially leading to **metastability**.

$$T_{data\_arrival} \geq T_{data\_required}$$

- ✓ **Hold Violation**: A **Hold Violation** occurs when the input data **changes too soon after the active clock edge before the required hold time has passed**. This can cause the flip-flop to **capture incorrect data**, leading to **corruption of the sampled value** and potentially causing **metastability** in the circuit.

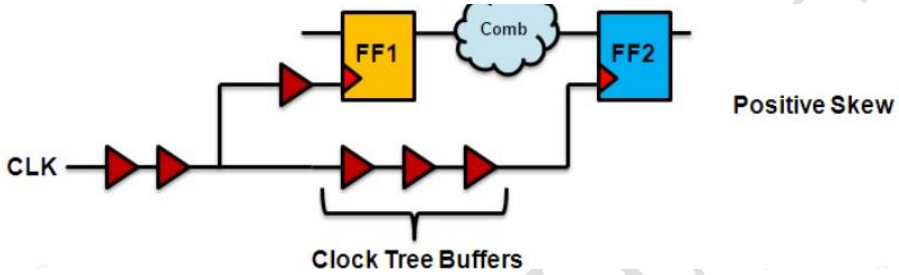
$$T_{data\_arrival} < T_{data\_required}$$

Setup violations can slow you down, but hold violations can break everything!

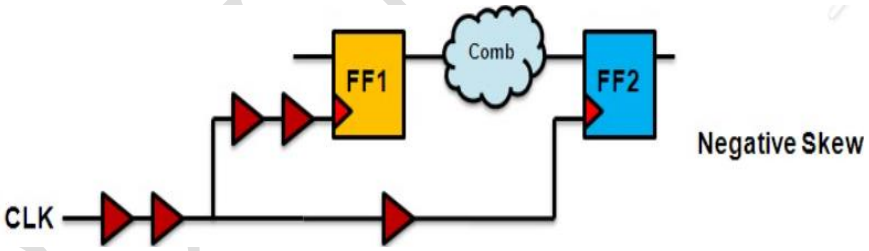
@harshita_harshi
www.linkedin.com/in/harshita-harshi-9377891bb

✓ **Clock Skew**: The difference in arrival time of the clock signal at different flip-flops.

- **Positive skew**: Occurs when the clock reaches the receiving flip-flop (FF2) later than the launching flip-flop (FF1).
  Impact:
  Helps setup timing ☑ (Gives data more time to propagate)
  Worsens hold timing ✕ (Increases the risk of hold violations)



- **Negative skew**: Occurs when the receiving register (FF2) receives the clock pulse earlier than the launching register (FF1).
  Impact:
  Improves hold timing ☑ (Because FF2 captures data early, reducing the risk of new data arriving too soon).
  Worsens setup timing ✕ (Because FF2 needs data earlier, increasing the risk of setup violations).



Violation	Cause	Fixes
Setup Violation	Violation can cause metastability, leading to unpredictable behavior.	Pipeline insertion, clock skew management, Retiming, and use of HVT cells.
Hold Violation	Violation can cause incorrect data to be latched.	Buffer insertion, reducing clock skew, increasing path delays.
Clock Skew	Uneven clock distribution	Optimize clock tree synthesis (CTS)
High Negative Slack	Insufficient optimization	Pipeline, logic restructuring

- SLACK: Difference between the required time and the arrival time of the data at the endpoint.

Maximum Operating Frequency (max)

$$f_{\max} = \frac{1}{T_{\text{clk}}}$$

$$Tclk = Tsetup + Tclk - q + Tcombinational + Tskew + Tjitter$$

$$T_{hold} > T_{clk\text{-}q} + T_{combinational}$$

Types of Timing Paths

Single-Cycle Path

- Data is expected to propagate from **within one clock cycle**.
- Checked for **setup and hold violations**.

Multi-Cycle Path (MCP)

- Data propagation is allowed **over multiple clock cycles**.
- Used in **low-power** or **long combinational logic paths**.
- Requires **multi-cycle path constraints** in STA.

False Path

- A path that **functionally does not contribute** to design operation.
- STA tools analyze all paths, so designers **explicitly define false paths** to avoid unnecessary timing violations.

Asynchronous Paths

- Paths between **different clock domains** (Clock Domain Crossing - CDC).
- Require **synchronization techniques** like **FIFO, handshake, or metastability prevention**.

Critical Path

- The **longest timing path** that determines the **maximum operating frequency** of the design.
- Must meet **setup timing constraints** for proper chip performance.

➢ STA Process Flow

1. **Parsing the Netlist:** Read the synthesized Netlist and library constraints.
2. **Extracting Timing Paths:** Identify all critical paths in the design.
3. **Performing Timing Calculations:** Compute delays, setup/hold margins, and clock skews.
4. **Analysing Slack Values:** Check if the design meets the required timing constraints.
5. **Optimizing for Violations:** Apply fixes such as buffering, resizing, or retiming.

➢ STA Constraints and Timing Analysis

To perform STA, designers define constraints using **SDC (Synopsys Design Constraints) files**, which include:

- Clock definitions (frequency, waveform, jitter)
- Input/output delays
- False path and multi-cycle path constraints (**STA cannot automatically detect false paths**)
- Operating conditions (PVT – Process, Voltage, Temperature variations)

➢ Limitations of STA

- False Paths Are Not Automatically Identified
- No Functional Awareness
- Ignores Data Dependencies
- No Coverage for Asynchronous Paths
- Pessimistic Timing Assumptions
- Limited Handling of Process Variations (PVT)
- Cannot Model Aging Effects (BTI, EM, HCI)
- Clock Jitter & Noise Effects

In STA, a nanosecond can make or break a billion-dollar chip—because in VLSI, timing isn't just everything, it's the only thing!