VLSI DESIGN FLOW: RTL TO GDS

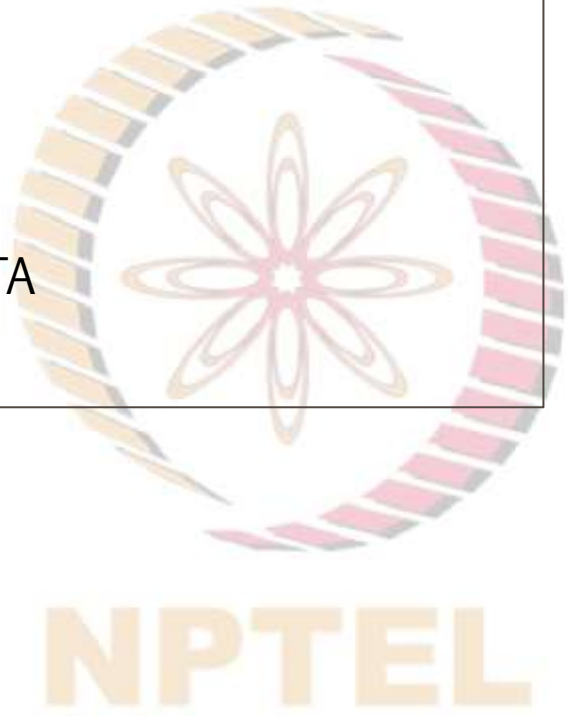
Lecture 22
Static Timing Analysis – Part I

Sneh Saurabh
Electronics and Communications Engineering
IIIT Delhi

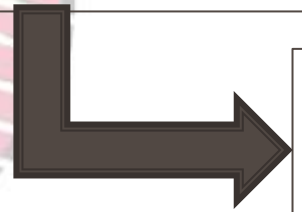
Lecture Plan

Static Timing Analysis (STA)

- Basics of STA (this lecture)
- Subsequent Lectures:
  - Mechanics of STA
  - Advanced concepts of STA
  - Constraints



Static Timing Analysis



Basics

Static Timing Analysis (STA): What is done?

- Ensures that the circuit is in a valid state at each clock cycle
- Verifies that the design is capable of operating at the *given frequency*
  - Information of *frequency* comes from *constraints*
- Ensures that the design does not have *setup* or *hold* violation at flip-flop

- The analysis is based on *worst case* scenario and takes a pessimistic view wherever possible
  - Verification is done without test vector and simulation (therefore **static**)
  - Ensures that design will not have setup or hold violations for **any test vector**

STA	Simulation
No test-vector required	Test vector required
No check of functionality	Checks functionality
Analysis performed taking pessimistic view of delays and other attributes of the design	Simulation done based on specified test vectors and delays

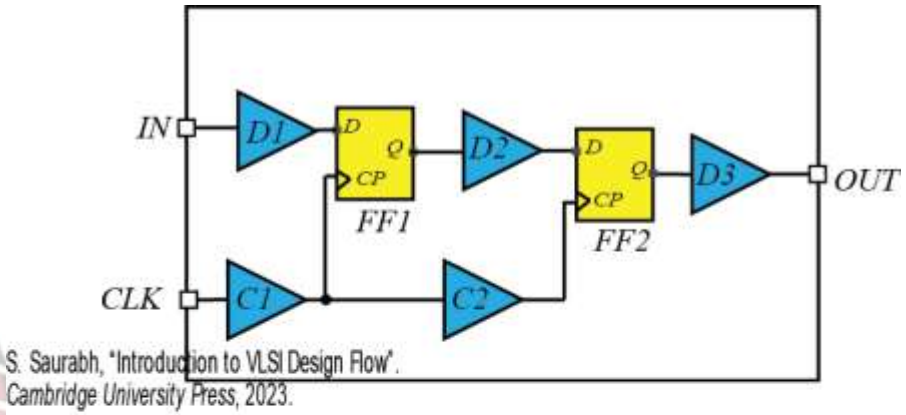
Synchronous Circuit: Data Propagation

Consider a synchronous circuit shown alongside.

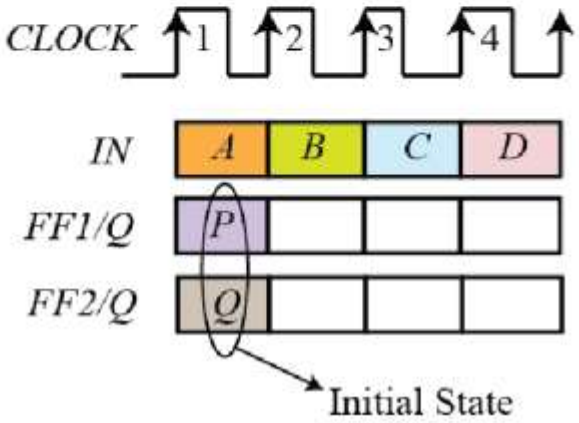
Assume that:

- Flip-flops are ideal
- Buffers have some delay
- Inputs at the port IN are applied as shown [using identifiers for clarity]
- State defined by the combination of values at the Q-pin
- Initial state is {PQ}

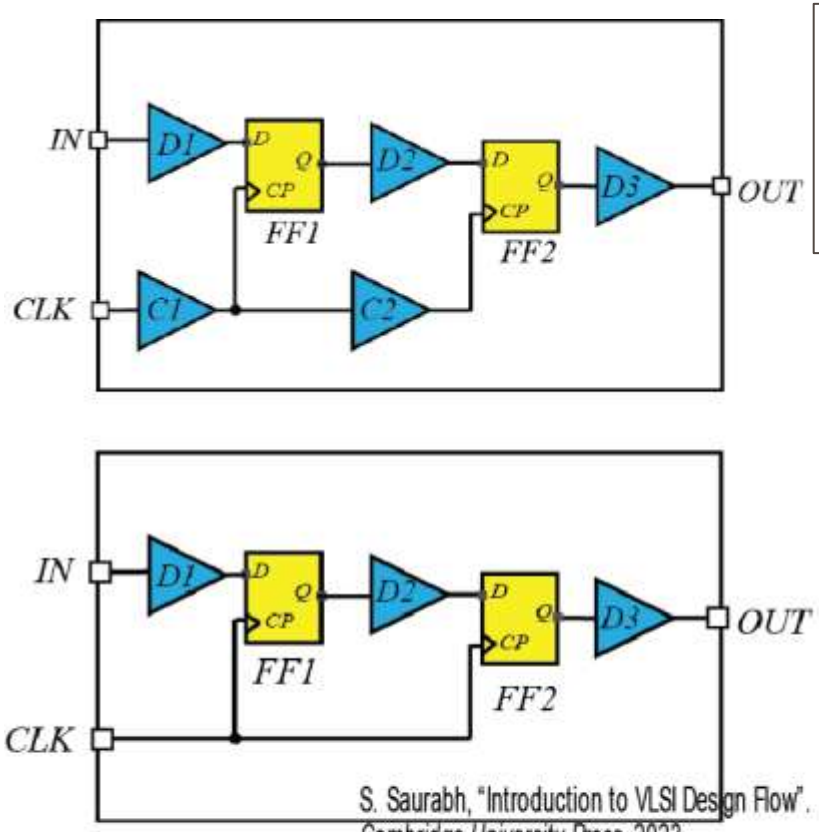
Let us understand the behavior of the circuit in different clock cycles for various cases of delay of the buffers.



S. Saurabh, "Introduction to VLSI Design Flow". Cambridge University Press, 2023.

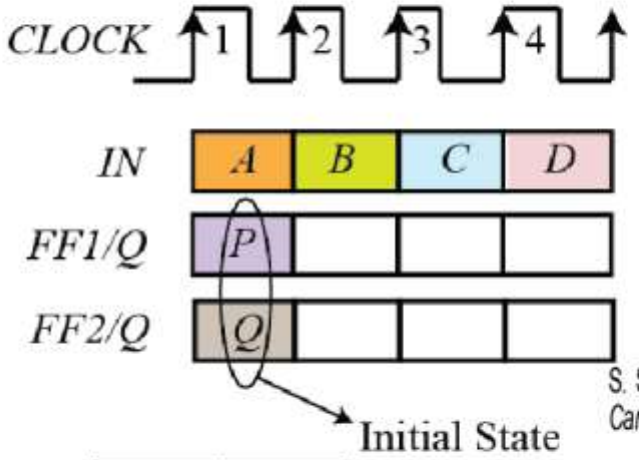


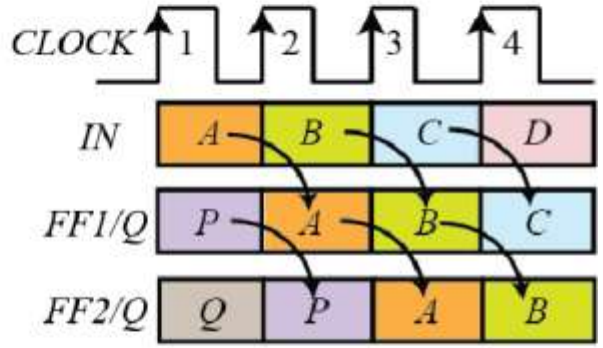
Synchronous Circuit: Synchronous Behavior



Assume that:

- Delay of *D1*, *D2,* and *D3* be some finite value less than the clock period, and
- Delays of *C1* and *C2* are negligible.



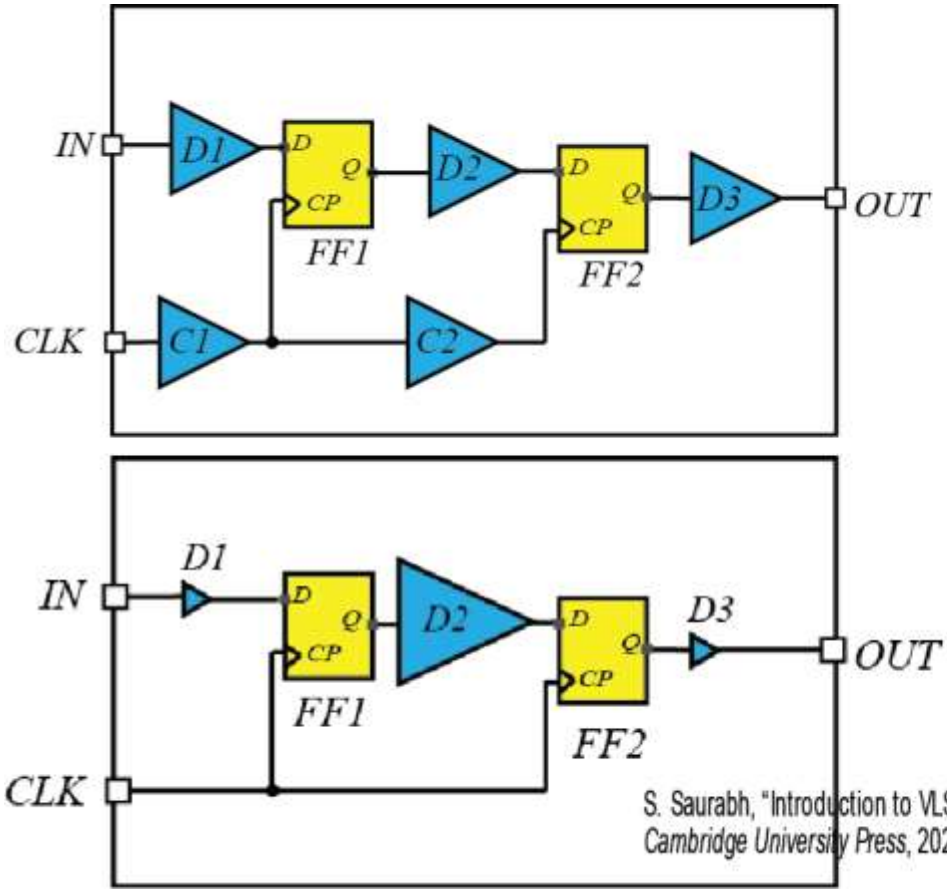


S. Saurabh, "Introduction to VLSI Design Flow".
Cambridge University Press, 2023.

Cycle	State
1	{*PQ*}
2	{*AP*}
3	{*BA*}
4	{*CB*}

These are valid states of a synchronous circuit.

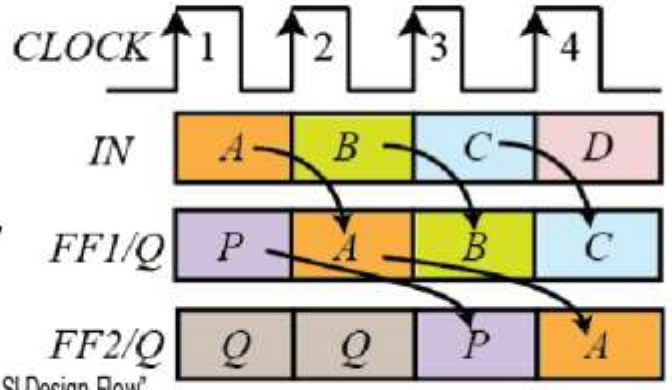
Synchronous Circuit: Zero Clocking



S. Saurabh, "Introduction to VLSI Design Flow". Cambridge University Press, 2023.

Assume that:

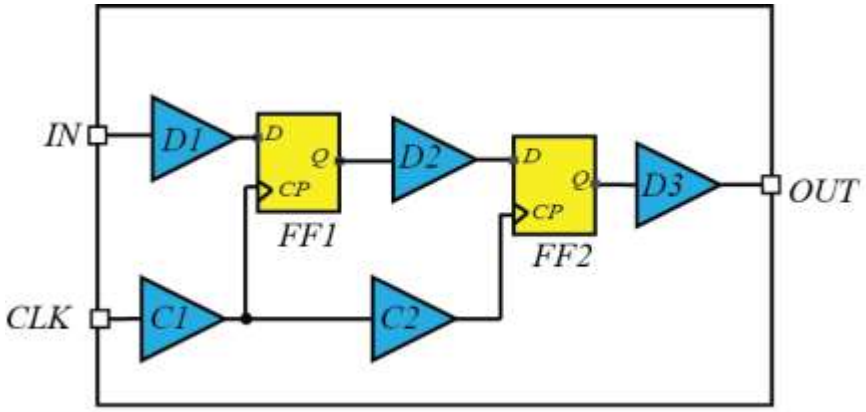
- Delay of *C1* and *C2* are negligible.
- Delay of the circuit element *D2* is more than one clock period, but less than two clock periods.
- Delay of *D1* and *D3* are minimal.

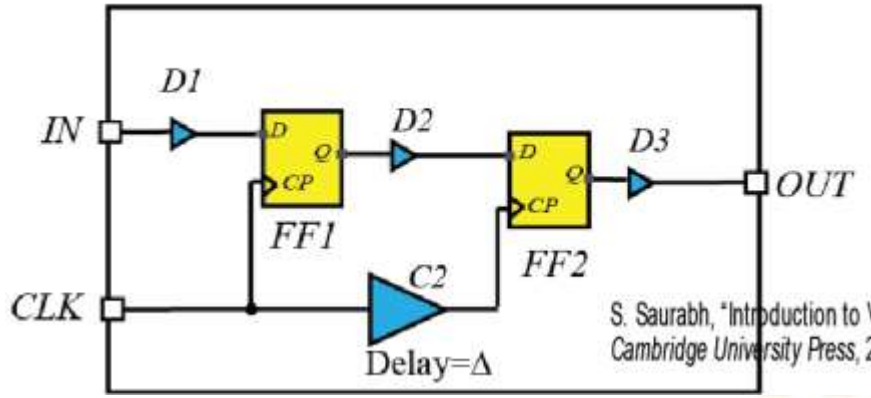


Cycle	State	Valid State
1	{*PQ*}	{*PQ*}
2	{*AQ*}	{*AP*}
3	{*BP*}	{*BA*}
4	{*CA*}	{*CB*}

- In effect clock fails to capture the right data (zero-clocking) due to late data arrival
- The circuit goes into invalid states

Synchronous Circuit: Double Clocking

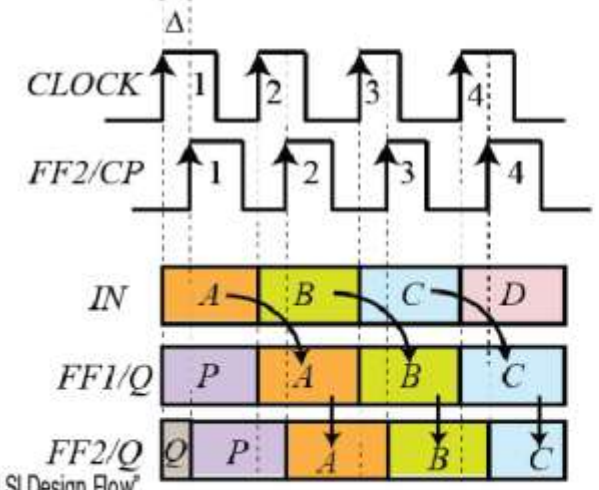




S. Saurabh, "Introduction to VLSI Design Flow", Cambridge University Press, 2023.

Assume that:

- Delay of circuit elements *D1*, *D2*, *D3,* and *C1* are insignificant.
- Delay of the circuit element *C2* is Δ



Cycle	State	Valid State
1	{*PP*}	{*PQ*}
2	{*AA*}	{*AP*}
3	{*BB*}	{*BA*}
4	{*CC*}	{*CB*}

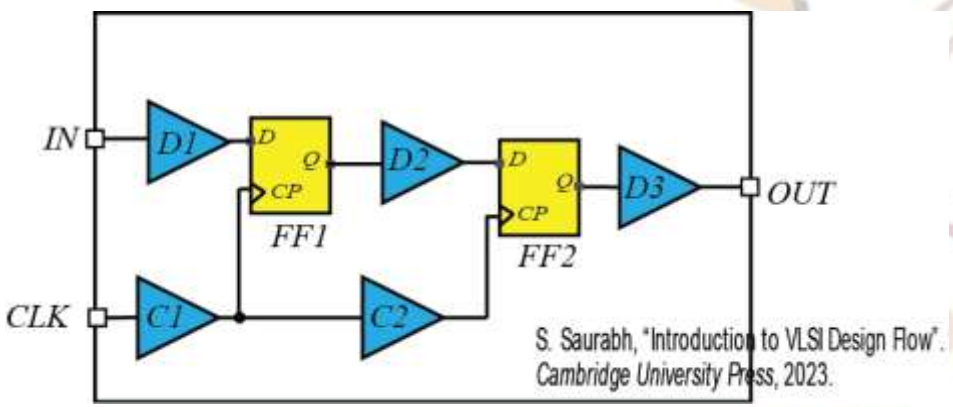
- In effect, the data gets captured by two flop-flops by the same clock edge (double clocking)
- The circuit goes into invalid states

Synchronous Circuit: Verification

To ensure synchronous behavior:

- **Avoid Zero Clocking**: setup analysis or late analysis
- **Avoid Double Clocking**: hold analysis or early analysis

- A synchronous circuit can contain many flip-flops
  - ➢ Data can propagate sequentially through a pipeline before reaching the output
- Examine each pair of launch and capture flip-flops separately



S. Saurabh, "Introduction to VLSI Design Flow". Cambridge University Press, 2023.

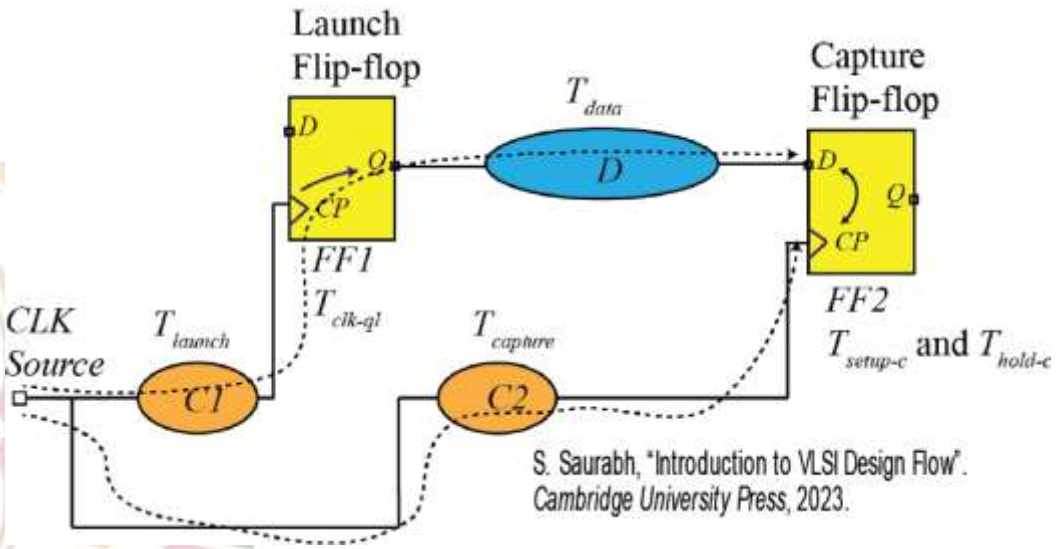
- Various types of combinational gates can be encountered in a path
- Add the delay of all the combinational circuit elements (and also the wire delay) in the path and check for delay requirements

- Real flip-flops have ST, HT, CK-Q delay
- Account for them (make the verification a bit more pessimistic)

Static Timing Analysis (STA) : Setup Requirement (1)

- Ensures that the data sent by launch flip-flop in a given clock cycle is captured reliably by the capture flip-flop in *the next clock cycle*

- Ensures that the setup requirement of the flip-flop is also met



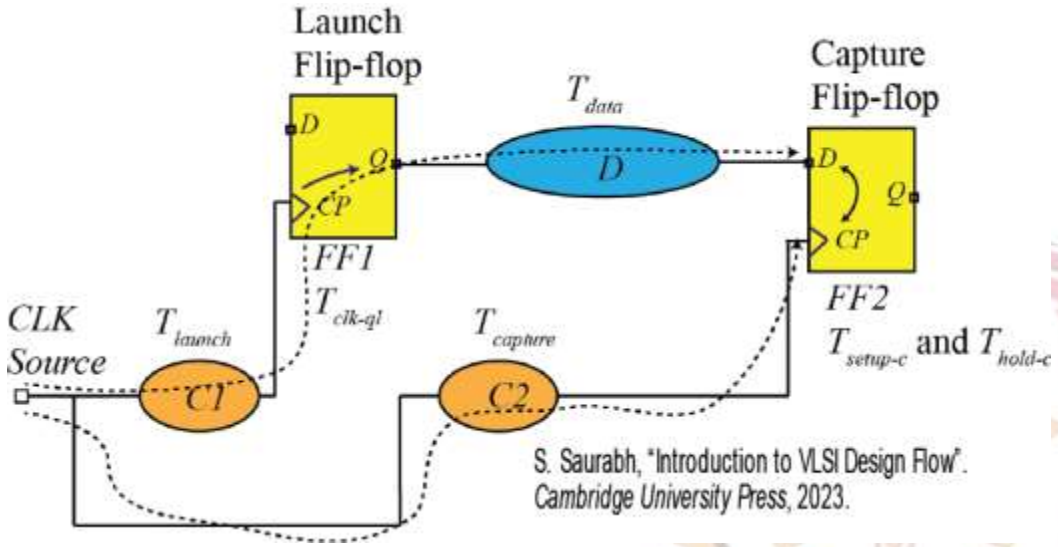
Arrival Time of data at the D-pin: $t_{arrival} = T_{launch} + T_{clk-ql} + T_{data}$

Required time for data to settle at FF2/D: $t_{req,set} = T_{period} + T_{capture} - T_{setup-c}$

To avoid zero clocking and setup-time constraints of flip-flops: $t_{req,set} > t_{arrival}$

Setup requirement: $T_{period} + T_{capture} - T_{setup-c} > T_{launch} + T_{clk-ql} + T_{data}$

Static Timing Analysis (STA) : Setup Requirement (2)



Setup requirement:

$$T_{period} + T_{capture} - T_{setup-c} > T_{launch} + T_{clk-ql} + T_{data}$$

$$T_{period} > (T_{launch} - T_{capture}) + T_{clk-ql} + T_{data} + T_{setup-c}$$

$T_{period} > \delta_{lc} + T_{clk-ql} + T_{data} + T_{setup-c}$ [δ_{lc} is the clock skew]

Most Restrictive:

$$T_{period} > \delta_{lc} + T_{clk-ql} + T_{data,max} + T_{setup-c}$$

Setup Violations can occur if:

- Clock Period is decreased (clock frequency is increased)
- Delay of capture clock path is decreased
- Delay of data path is increased
- Delay of launch clock path is increased

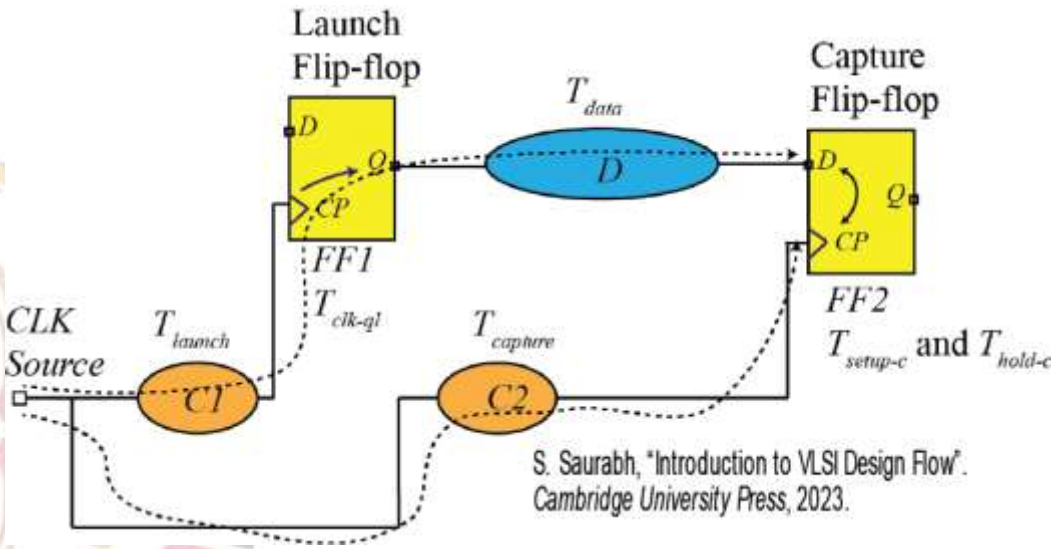
What happens for an ideal flip-flop?

What happens for an ideal clocking structure?

Static Timing Analysis (STA) : Hold Requirement (1)

- The hold check is from one active edge of the clock in the launch flip-flop to the ***same clock edge*** at the capture flip-flop (**independent of clock-period**)

- Ensures that the hold requirement of the flip-flop is also met



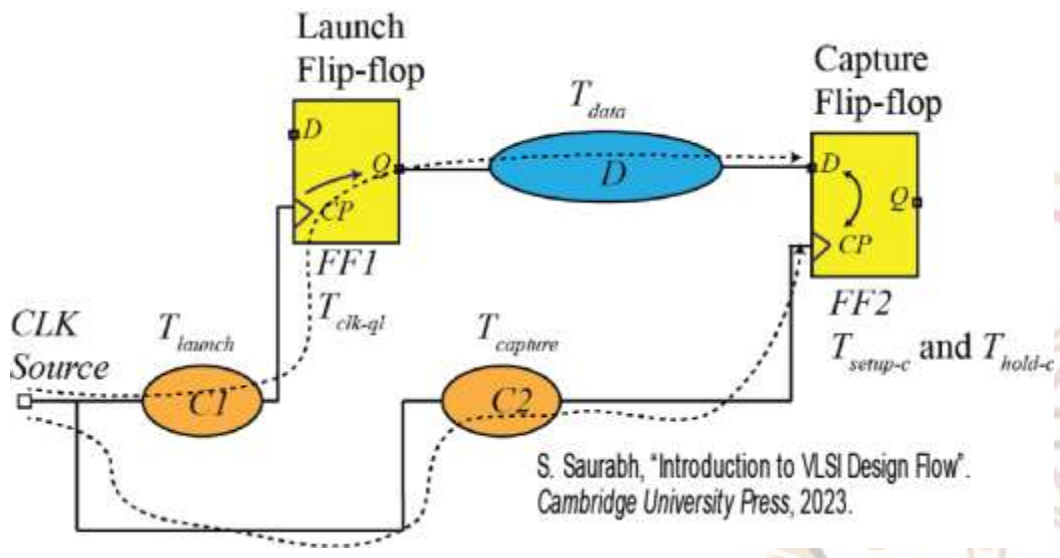
Data Reaches D-pin: $t_{arrival} = T_{launch} + T_{clk-ql} + T_{data}$

Data to arrive at FF2/D after the required time: $t_{req,hold} = T_{capture} + T_{hold-c}$

To avoid double clocking and hold-time constraints of flip-flops: $t_{arrival} > t_{req,hold}$

Hold requirement: $T_{launch} + T_{clk-ql} + T_{data} > T_{capture} + T_{hold-c}$

Static Timing Analysis (STA) : Hold Requirement (2)



Hold requirement:

$$T_{launch} + T_{clk-ql} + T_{data} > T_{capture} + T_{hold-c}$$

$$\delta_{lc} + T_{clk-ql} + T_{data} > T_{hold-c}$$

Most Restrictive requirement:

$$\delta_{lc} + T_{clk-ql} + T_{data,min} > T_{hold-c}$$

- Hold Violations can occur if:
  - Delay of data path is decreased
  - Delay of launch clock path is decreased
  - Delay of capture clock path is increased

What happens for an ideal flip-flop?

What happens for an ideal clocking structure?

References

- S. Saurabh, “Introduction to VLSI Design Flow”. Cambridge: *Cambridge University Press*, 2023.