Lecture 6
EE 421 / CS 425
Digital System Design

Fall 2024

Shahid Masud

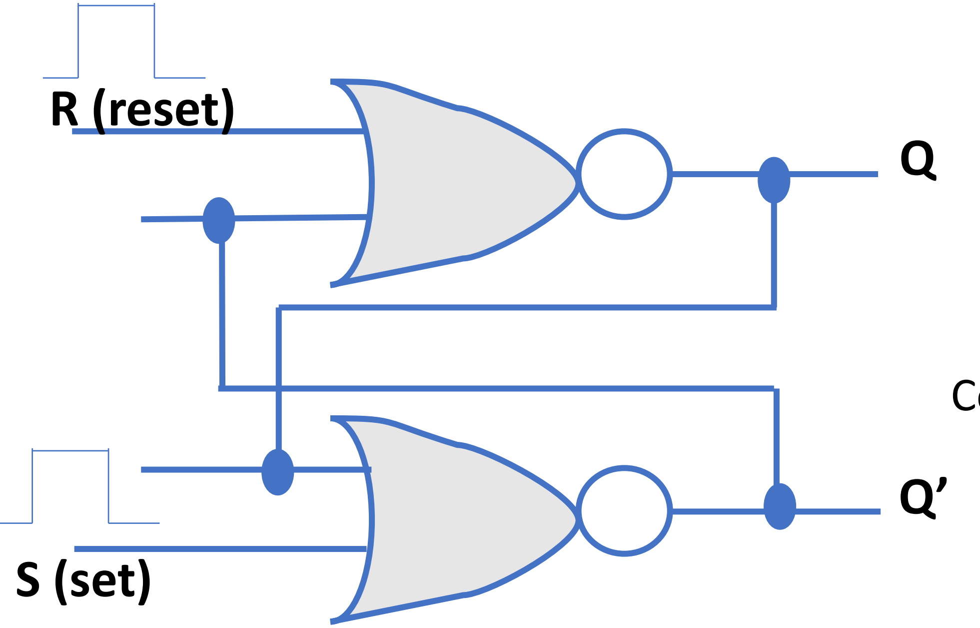
Topics

- **Topics remaining from previous lecture on Glitches and Hazards**
- ----------
- Review of Sequential Digital Logic Circuits
- Basics of Latches
- Basics of Flipflops
- Setup and Hold Times
- D Flipflop
- Clock Issues

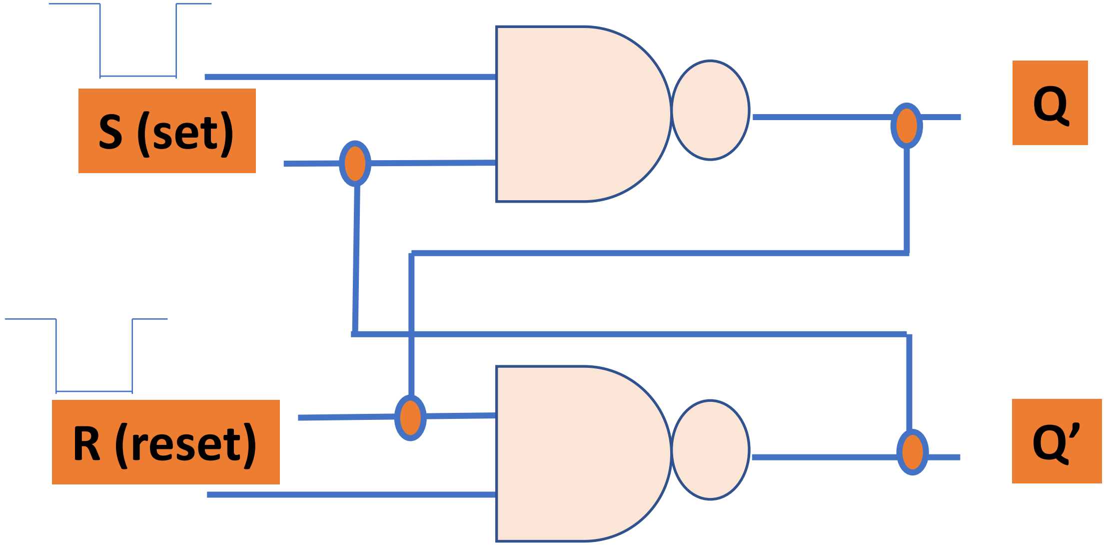
Latches and Flipflops - Definitions

- **Latches:** Level Sensitive storage elements; that can transition at any point in time
- **Flipflops:** Edge Sensitive storage elements, that transition at clock edges, either rising or falling
- **Asynchronous Sequential Logic:** Based on latches and un-clocked transitions
- **Synchronous Sequential Logic:** Based on flipflops and only clocked transitions

Fundamental Element – SR Latch



Complement



SR Latch Using 2 Input NOR Gate – Function Table

S (Set)	R (Reset)	Q	Condition
0	0	Q (=1 after S=1, R=0)	Hold
0	0	Q (=0 after S=0, R=1)	Hold
0	1	0	Reset
1	0	1	Set
1	1	0	Not Allowed

RACE Condition

SR Latch Using 2 Input NAND Gate – Function Table

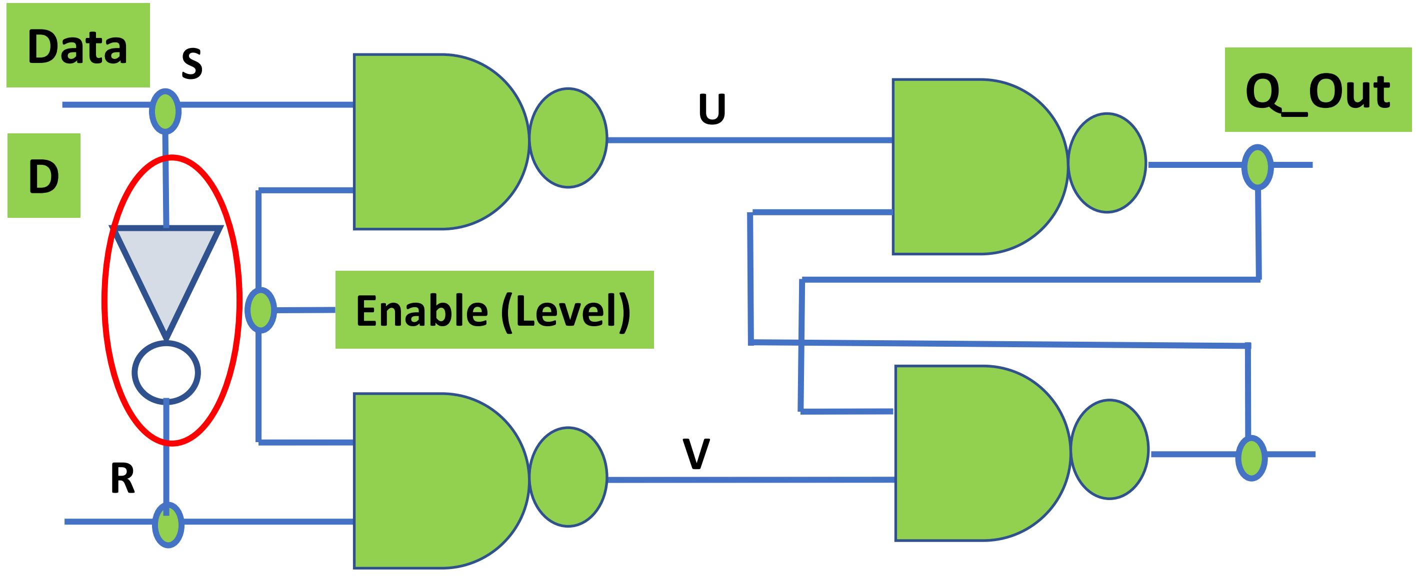
S	R	Q	Condition
0	0	1	Not Allowed
0	1	1	Set
1	0	0	Reset
1	1	Q (=0 after S=1, R=0)	Hold
1	1	Q (=1 after S=0, R=1)	Hold

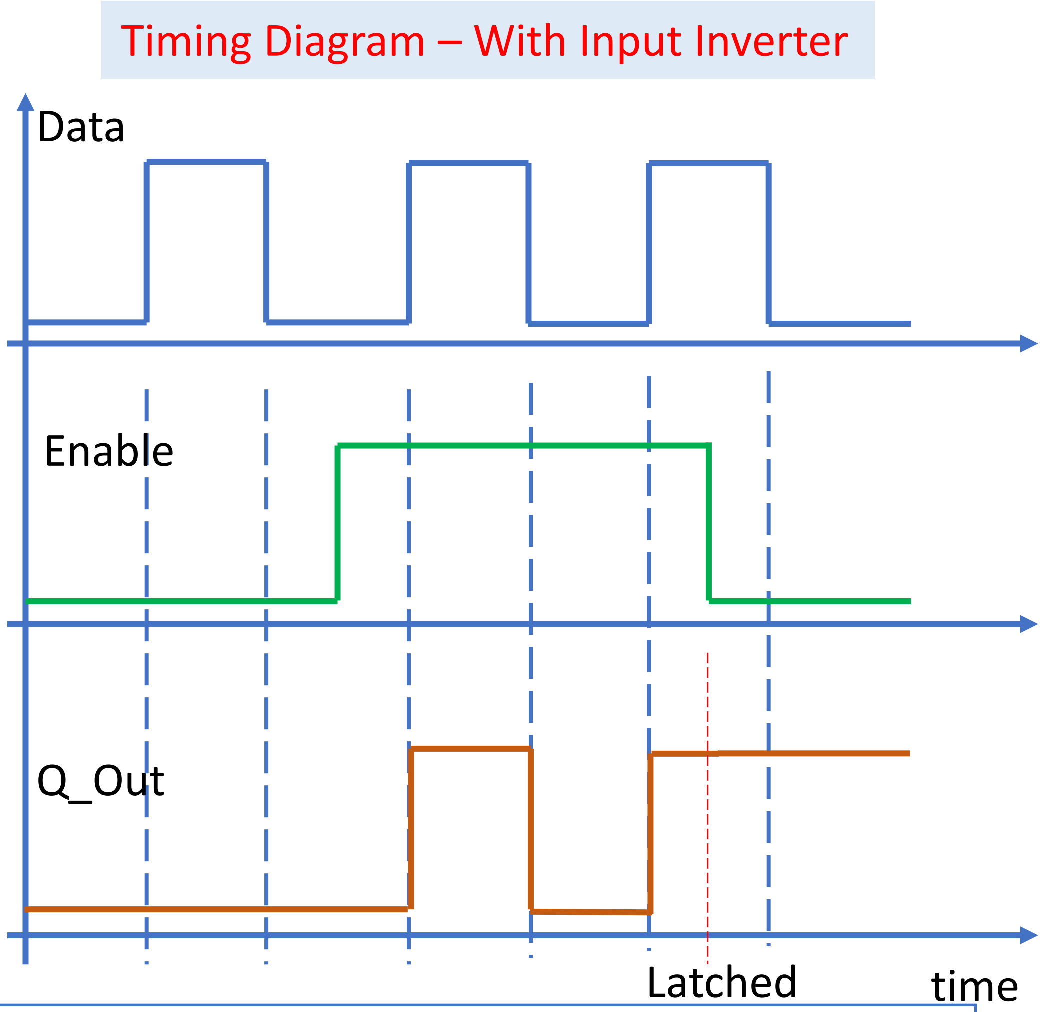
RACE Condition

Transparent Latch

- An additional ‘Enable’ input is provided
- Definition: Output of Transparent Latch changes in response to the data input only when the Latch is ‘Enabled’.
- Changes in input are straight away visible at the output

Transparent Latch Circuit





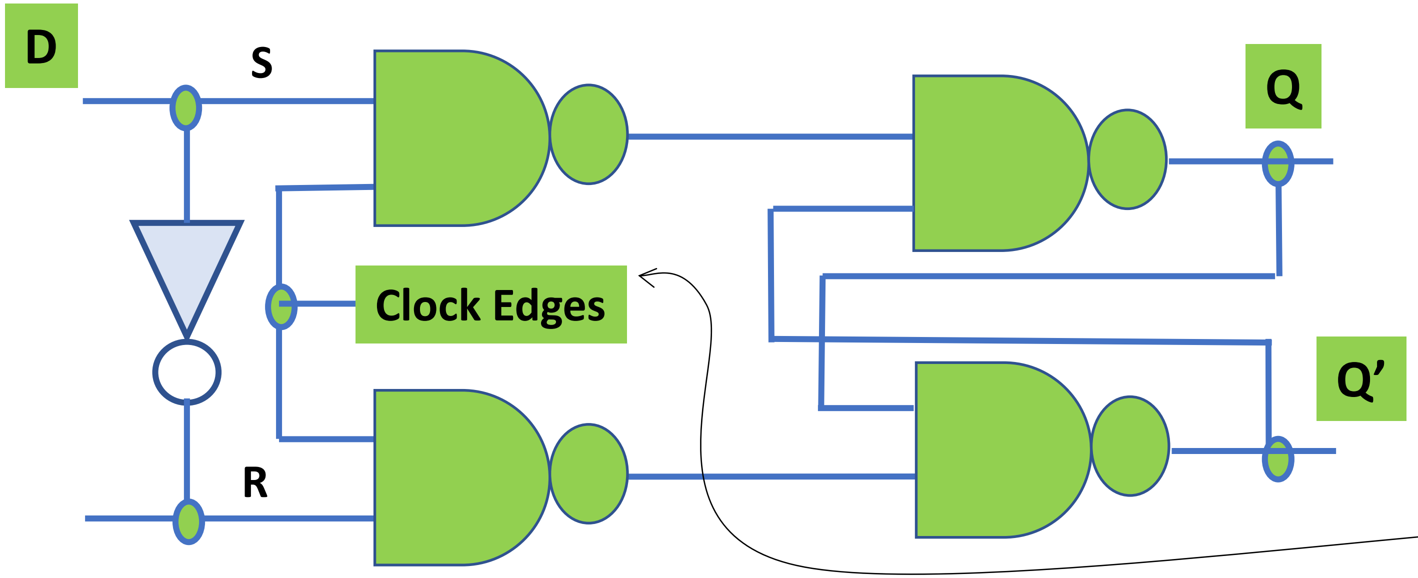
SR Latch With Enable (Without Inverter)

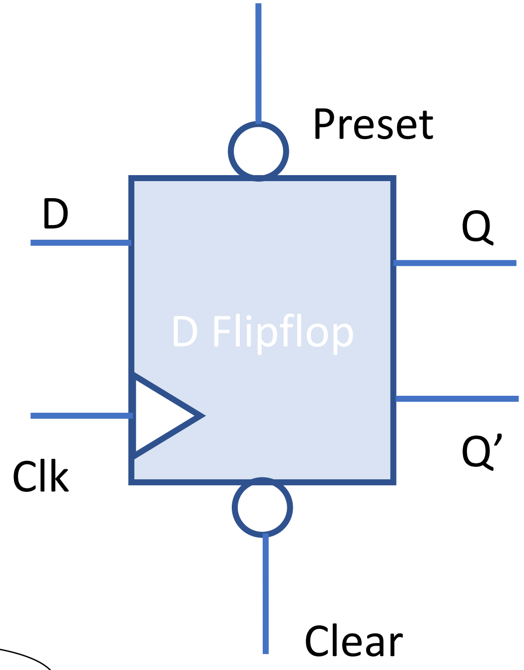
En	S	R	Next State of Q
0	X	X	No Change
1	0	0	No Change
1	0	1	Q=0; Reset State
1	1	0	Q=1; Set State
1	1	1	Not Allowed

When Enable is ‘0’; U=‘1’ and V=‘1’; Hold Condition in SR Latch

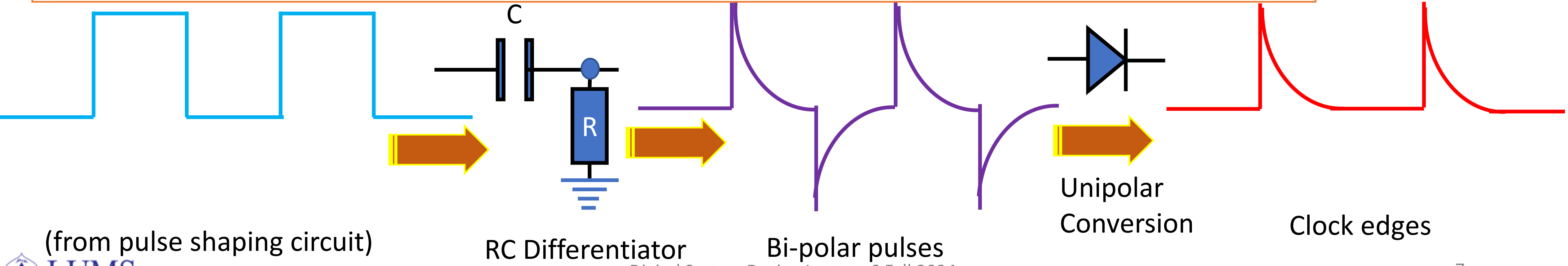
Only these two states are valid when inverter is inserted

D Flipflop from Transparent Latch





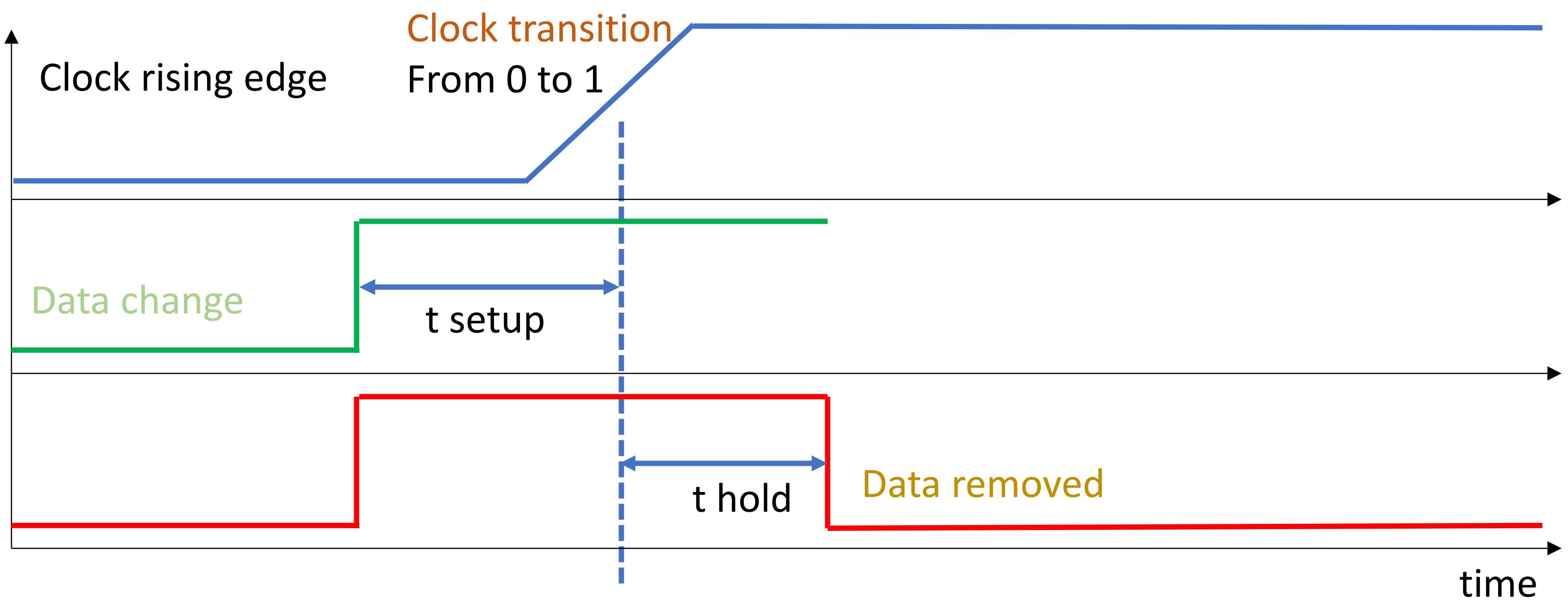
NOTE: Usually a Master-Slave Arrangement is used where output is available after clock goes 0 to 1 to 0



Timing Definitions

- Setup Time: The duration of time the Data needs to be present before the arrival of clock.
- Any Data activity, in tandem with clock edge, violates setup time and will not be recorded in the D flipflop
- Hold Time: The time for which data has to remain stable after the removal of clock edge.
- If the Data changes too soon, the internal circuit may not have settled with the new value.

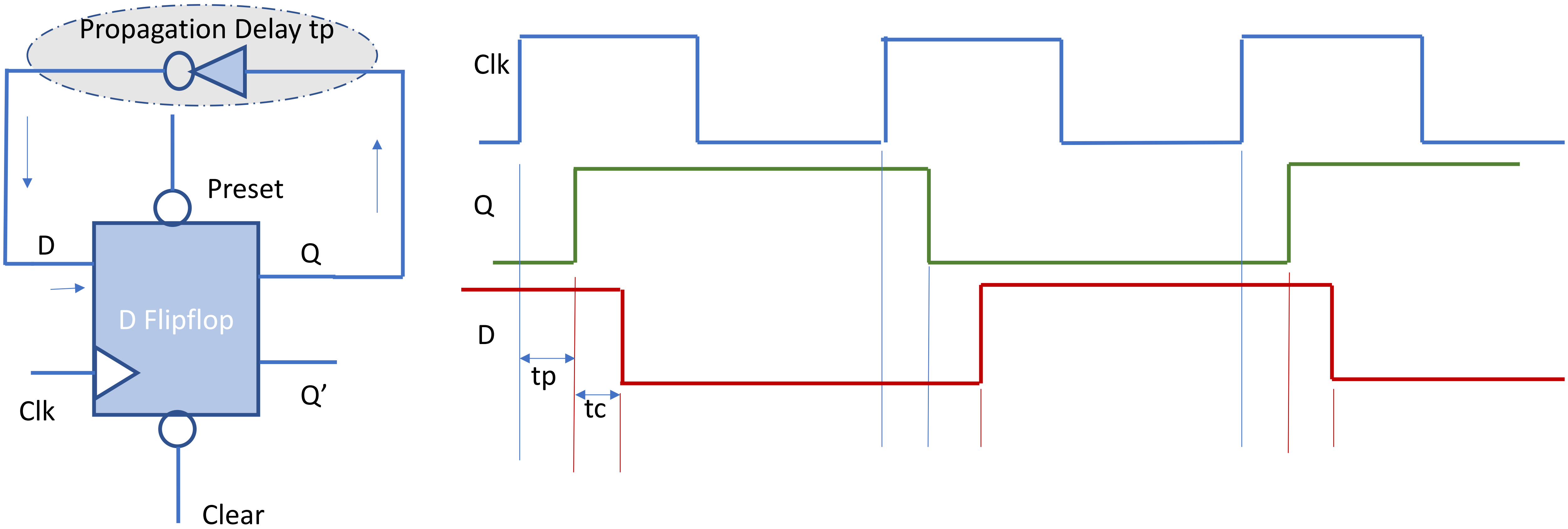
Setup and Hold Times



Maximum clock frequency has to be slower than (t setup + t hold) to maintain correct data

These timings needed both ways, slow data and fast clock; as well as slow clock and fast data

Timing in a simple frequency divider

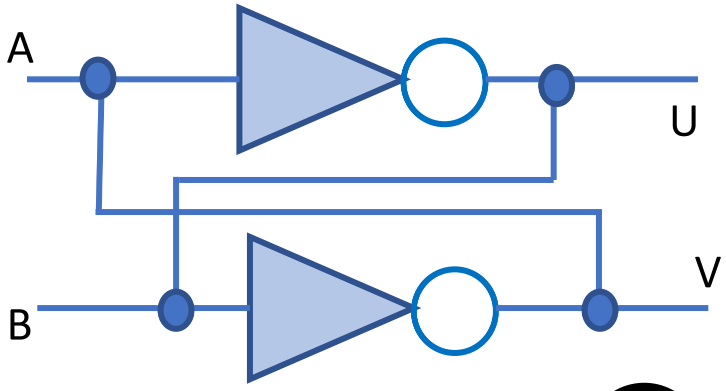


If the current output of flipflop is 1, a value of 0 will appear at D after propagation delay of inverter

Assuming that next active edge of clock arrives after setup time has elapsed, the output of flipflop will go to 0

A continuous waveform with period twice the period of Clk is observed at Q

Metastable State

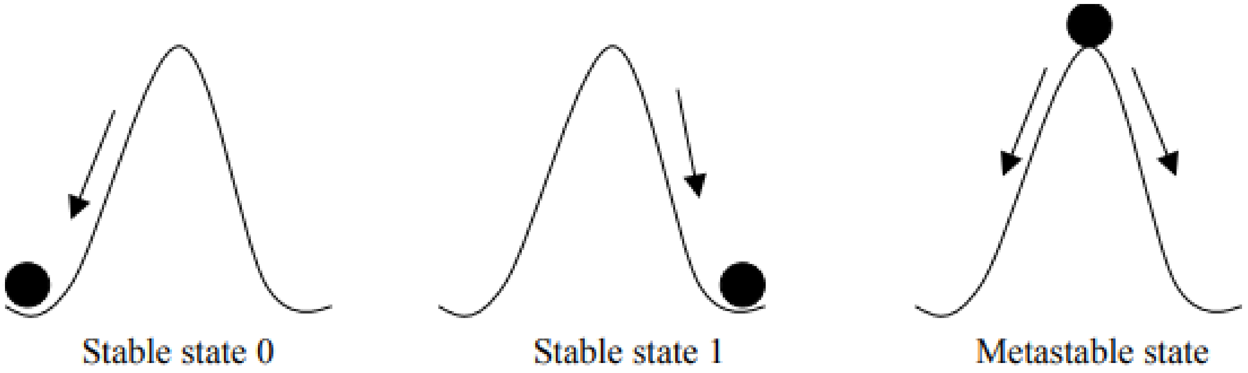


Imagine a circuit like this

Given a pulse input:
Output reaches a value (or oscillates) somewhere in-between '0' and '1' and stays there, neither '0' nor '1', for some noticeable time

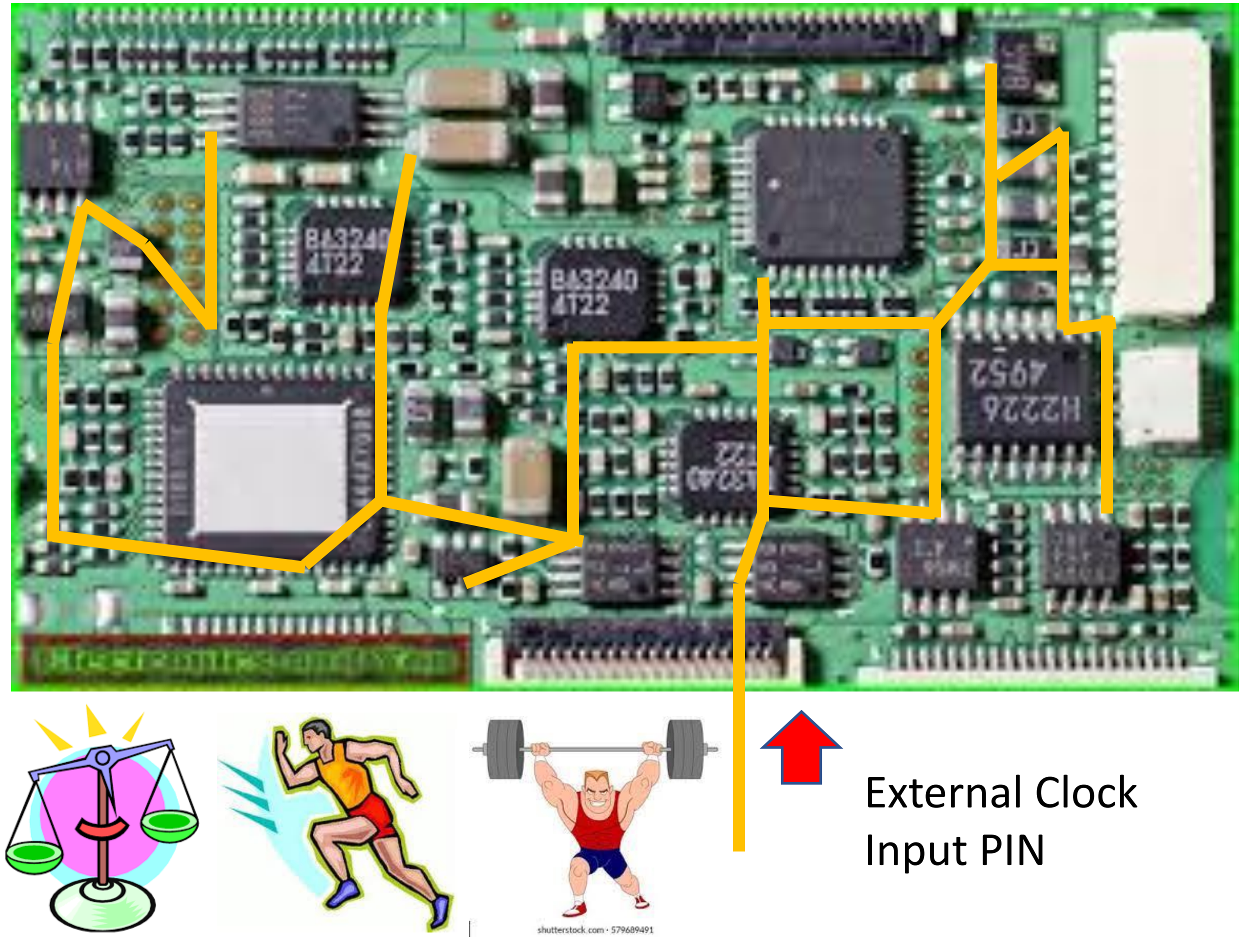
Why?

- Too small pulse violates setup and hold times
- Too less drive strength available to overcome the intermediate state



Clock - Problems

Clock is an essential Signal in Synchronous Digital Systems



External Clock Input PIN

Issues:
1. Too much load on Clock Input Pin
2. Variable path delay to Different Chips in circuit
3. Path is prone to picking up noise from surrounding

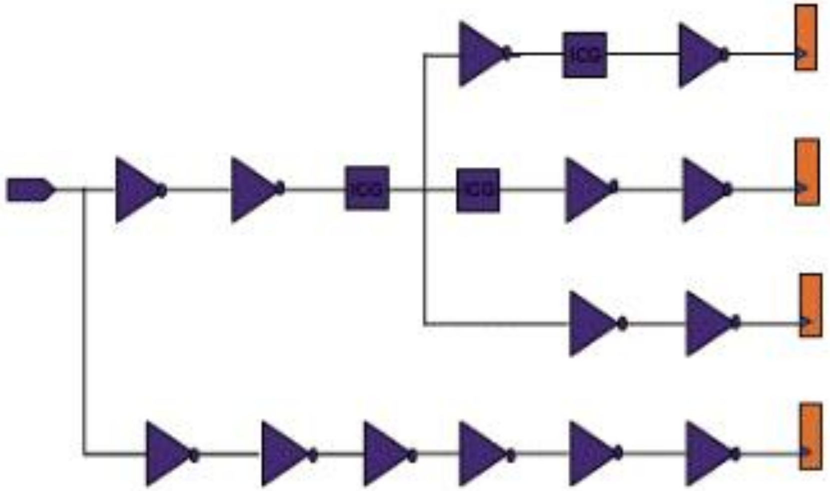
Clock Distribution Circuit

Qualities of a Good Clock Signal:

- Clock has to reach each and every module of a complex circuit
- Clock needs very high signal drive strength
- Clock Edge has to appear at the same instant across all modules (skew)
- Clock should be stable and free from jitter (time variations along stable reference point)

Clock Tree Synthesis

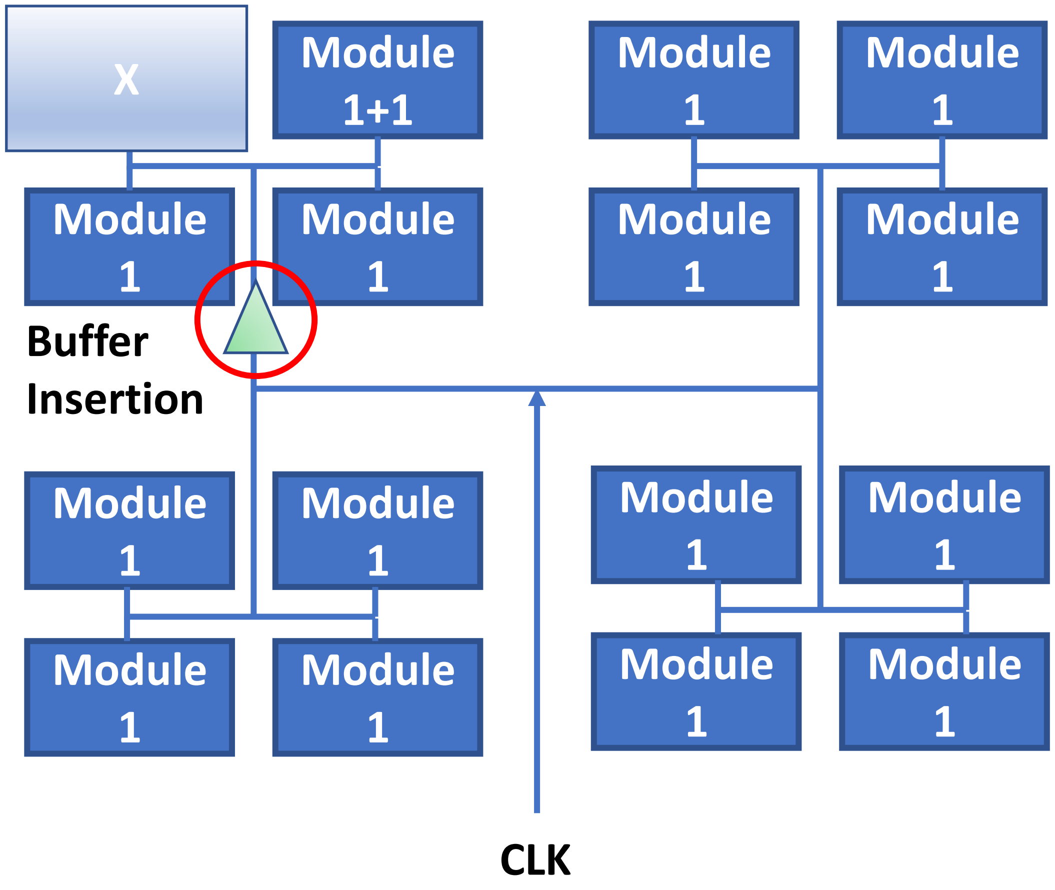
- Clock tree synthesis is performed during the physical design process considering the effects of place and route, channel impedance, parasitic loads, etc.
- Then through the insertion of buffers or inverters along the clock paths, it minimize or balances skew of important clock signal chains, build a clock tree that achieves proper timing across the entire design.



Ref: https://www.system-to-asic.com/blog/what-is-asic-design/

Some Techniques for Clock Distribution

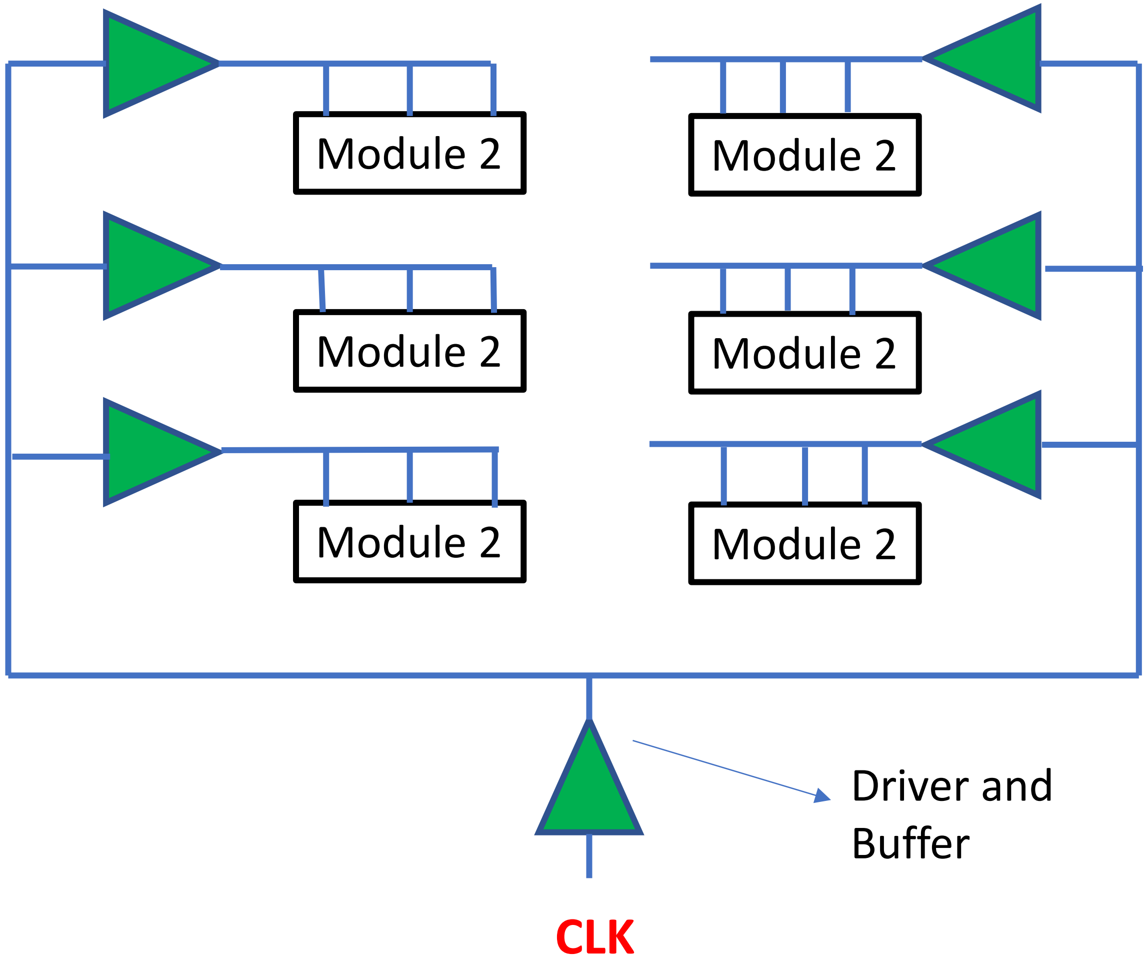
H-Tree Clock Synthesis
Path is balanced through Tree structure



H Tree Clock Distribution Network
Buffer Insertion for Delay Balancing in un-balanced trees

Hierarchical Clock

Buffer based Path Delay Balancing



Two Level Distributed Clock Buffering