CASPER

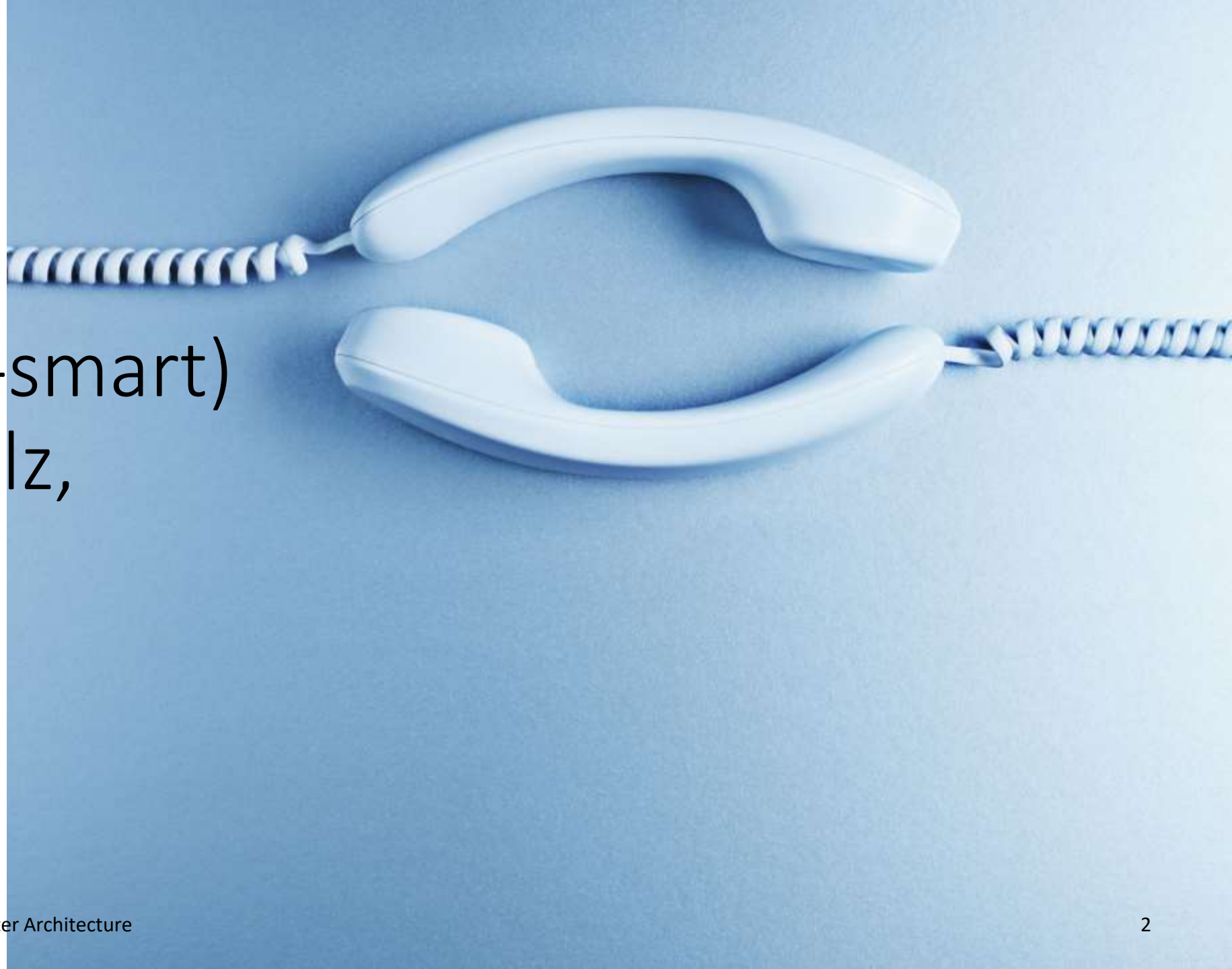
CS230: Digital Logic Design and Computer Architecture

Lecture 5: The D-Day@Digital logic

https://www.cse.iitb.ac.in/~biswa/courses/CS230/autumn23/main.html

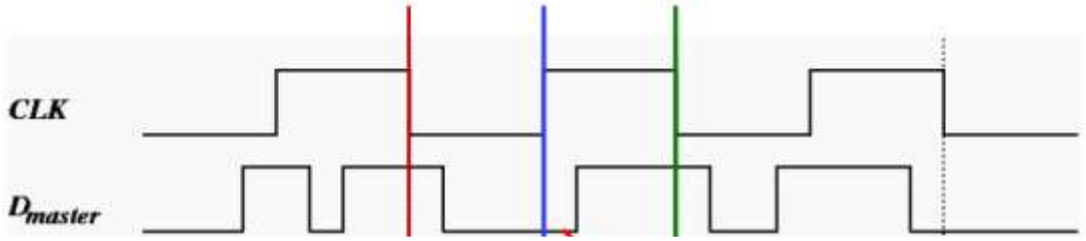
https://www.cse.iitb.ac.in/~biswa/

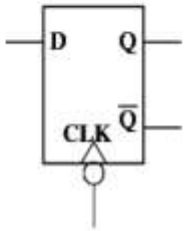
Phones (smart/non-smart) on silence plz, Thanks

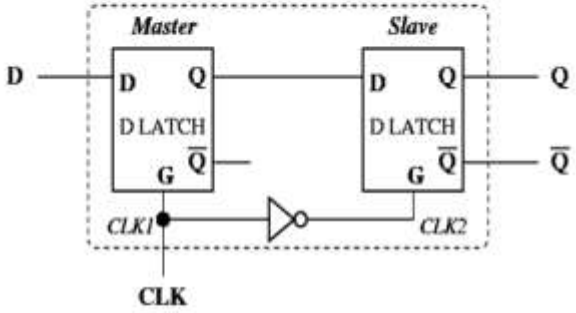


Recap of D Flip-flop

- **Example**: When CLK is high, output of master is allowed to change with D; when CLK is low (falling edge), the output of the master is fixed and propagated through to the output of the slave ⇨ this flip-flop triggers on *falling* or *negative edge*.





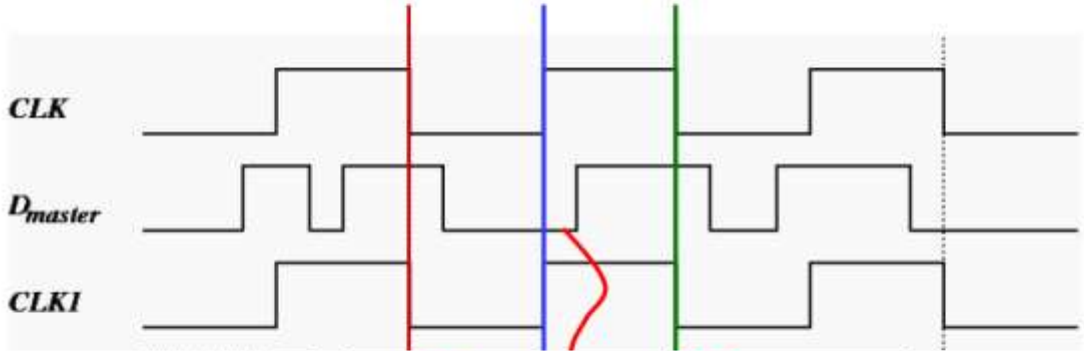


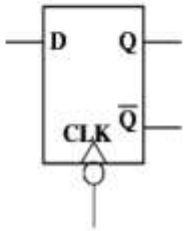
CLK	D	Q*	$\overline{Q}$*
↓	0	0	1
↓	1	1	0

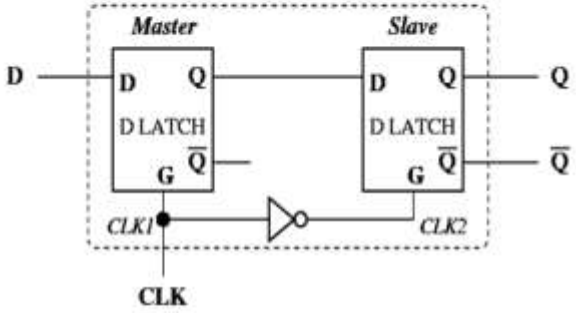
Characteristic Table

Recap of D Flip-flop

- **Example**: When CLK is high, output of master is allowed to change with D; when CLK is low (falling edge), the output of the master is fixed and propagated through to the output of the slave ⇨ this flip-flop triggers on *falling* or *negative edge*.





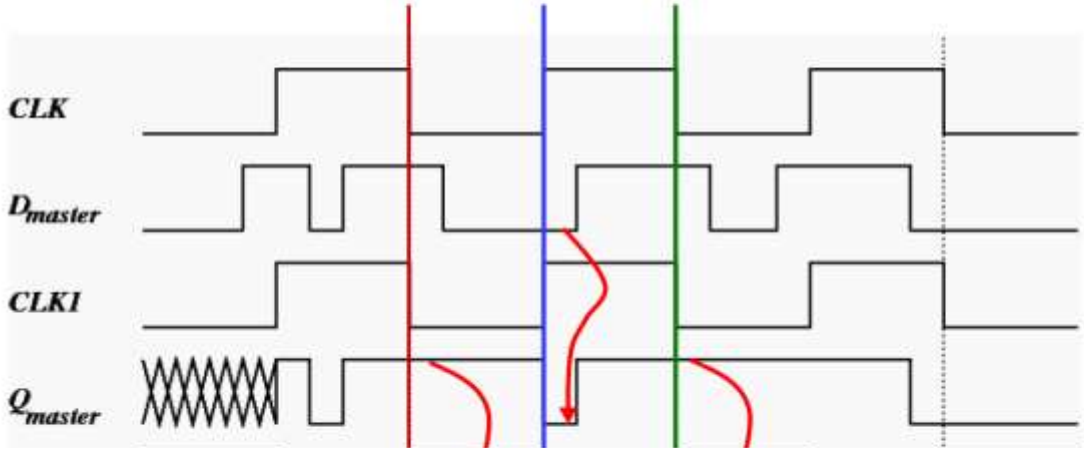


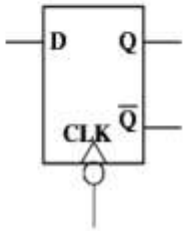
CLK	D	Q*	$\overline{Q}$*
↓	0	0	1
↓	1	1	0

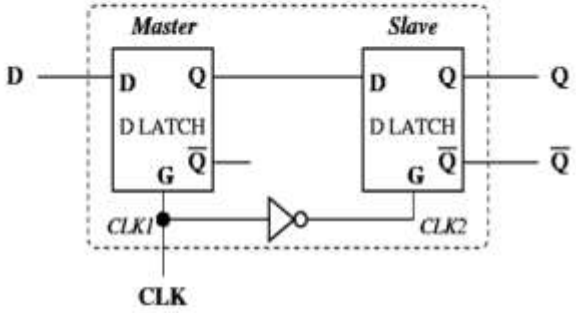
Characteristic Table

Recap of D Flip-flop

- **Example**: When CLK is high, output of master is allowed to change with D; when CLK is low (falling edge), the output of the master is fixed and propagated through to the output of the slave ⇨ this flip-flop triggers on *falling* or *negative edge*.





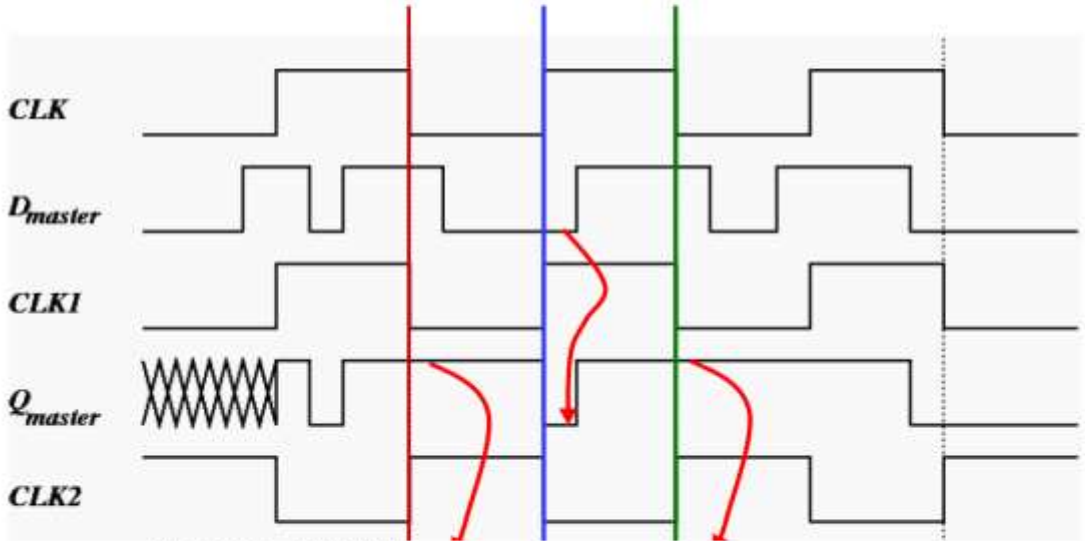


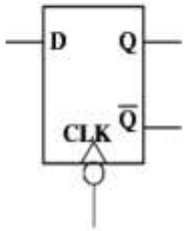
CLK	D	Q*	$\overline{Q}$*
↓	0	0	1
↓	1	1	0

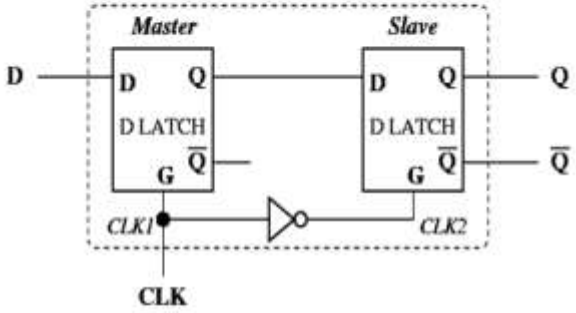
Characteristic Table

Recap of D Flip-flop

- **Example**: When CLK is high, output of master is allowed to change with D; when CLK is low (falling edge), the output of the master is fixed and propagated through to the output of the slave ⇨ this flip-flop triggers on *falling* or *negative edge*.





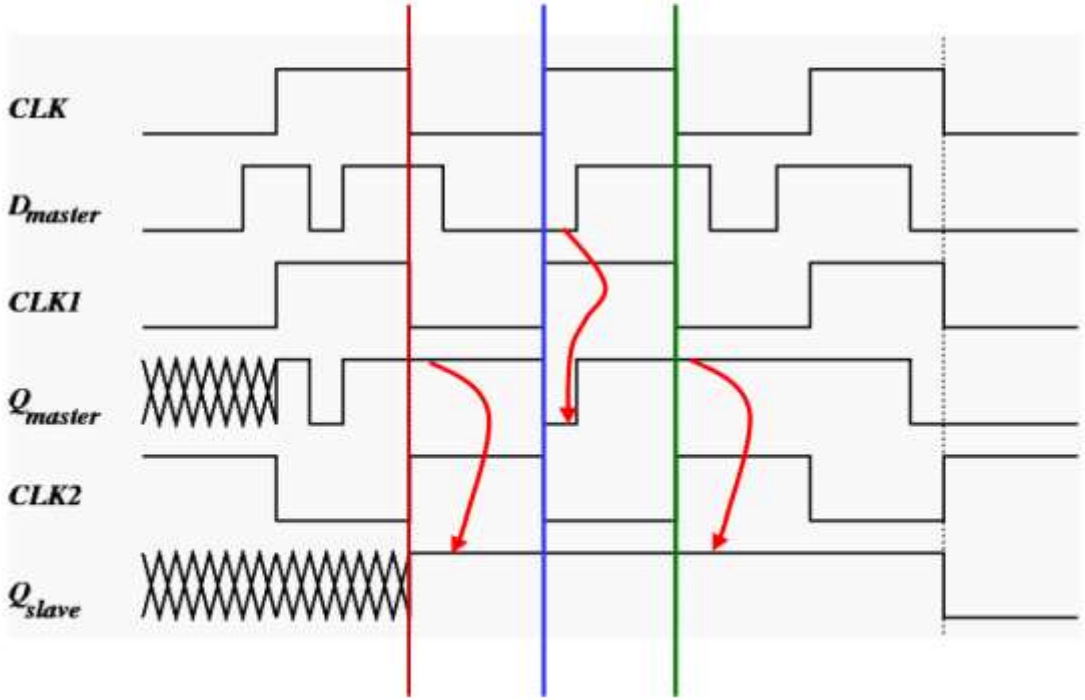


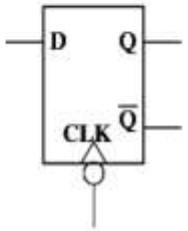
CLK	D	Q*	$\overline{Q}$*
↓	0	0	1
↓	1	1	0

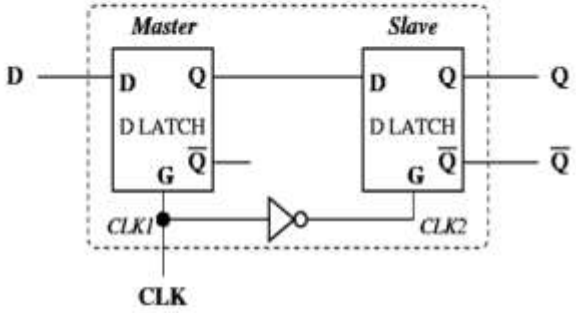
Characteristic Table

Recap of D Flip-flop

- **Example**: When CLK is high, output of master is allowed to change with D; when CLK is low (falling edge), the output of the master is fixed and propagated through to the output of the slave ⇨ this flip-flop triggers on *falling* or *negative edge*.







CLK	D	Q*	$\overline{Q}$*
↓	0	0	1
↓	1	1	0

Characteristic Table

Delay to make sure all is well

- **Setup time**, t_{su}, is the time period prior to the clock becoming active (edge or level) during which the flip-flop inputs must remain stable.

- **Hold time**, t_h, is the time after the clock becomes inactive during which the flip-flop inputs must remain stable.

- Setup time and hold time define a *window of time during which the flip-flop inputs cannot change* – quiescent interval.

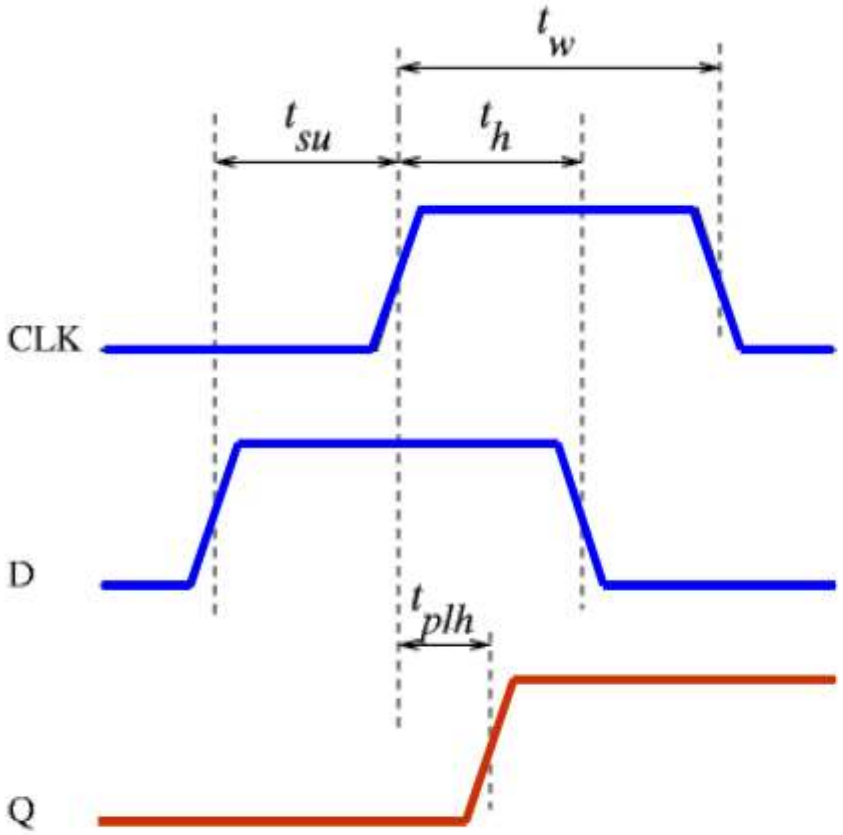
More Delay

- **Propagation delay,** t_{pHL} and t_{pLH} , has the same meaning as in combinational circuit – beware propagation delays usually will not be equal for all input to output pairs. There can be two propagation delays: $t_{C\text{-}Q}$ (*clock*→*Q* delay) and $t_{D\text{-}Q}$ (*data*→*Q* delay).

- For a level or pulse triggered latch:
  - Data input should remain stable till the clock becomes inactive.
  - Clock should remain active till the input change is propagated to Q output. That is, active period of the clock,

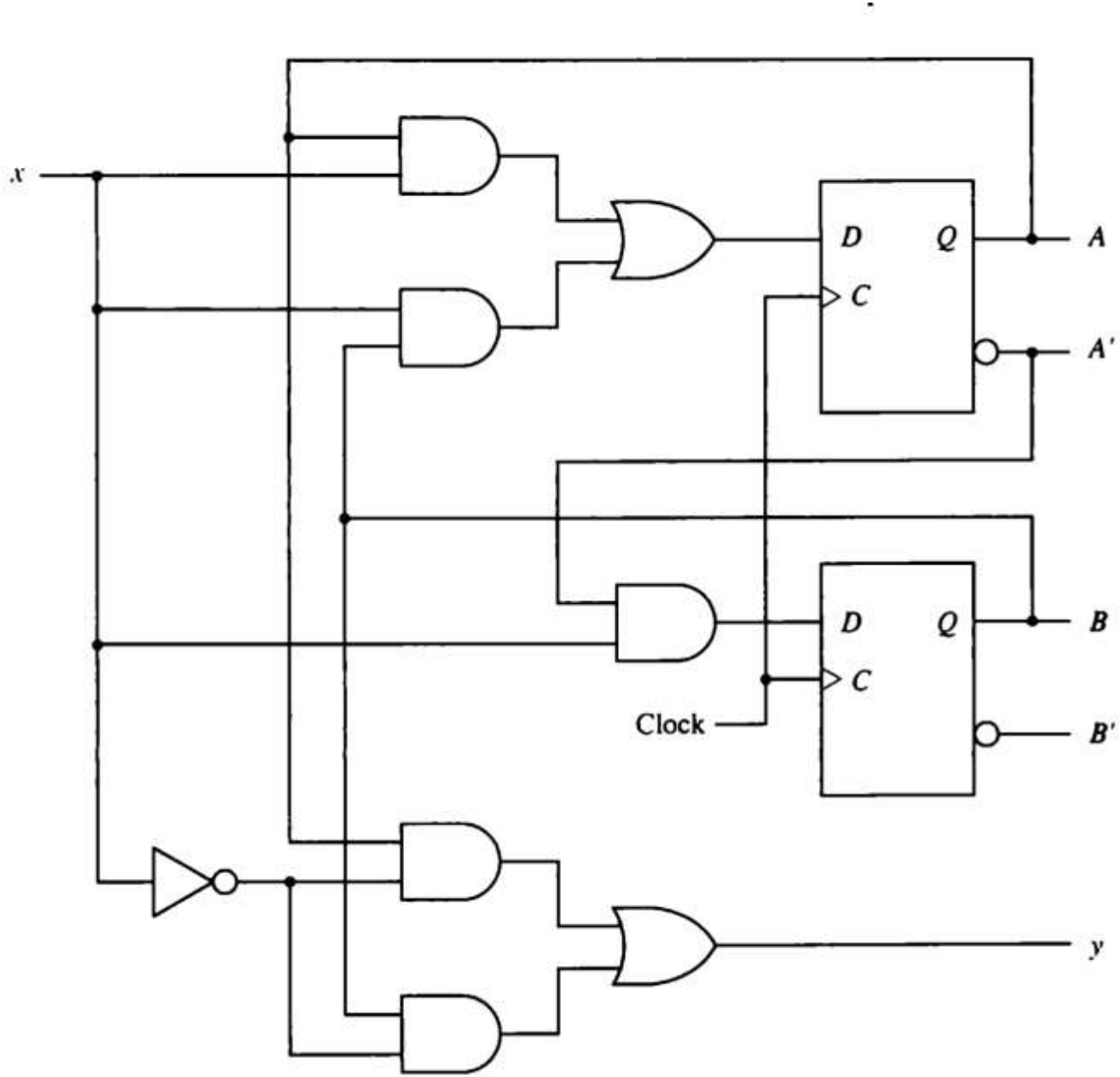
$$t_w > \max \{t_{pLH}, t_{pHL}\}$$

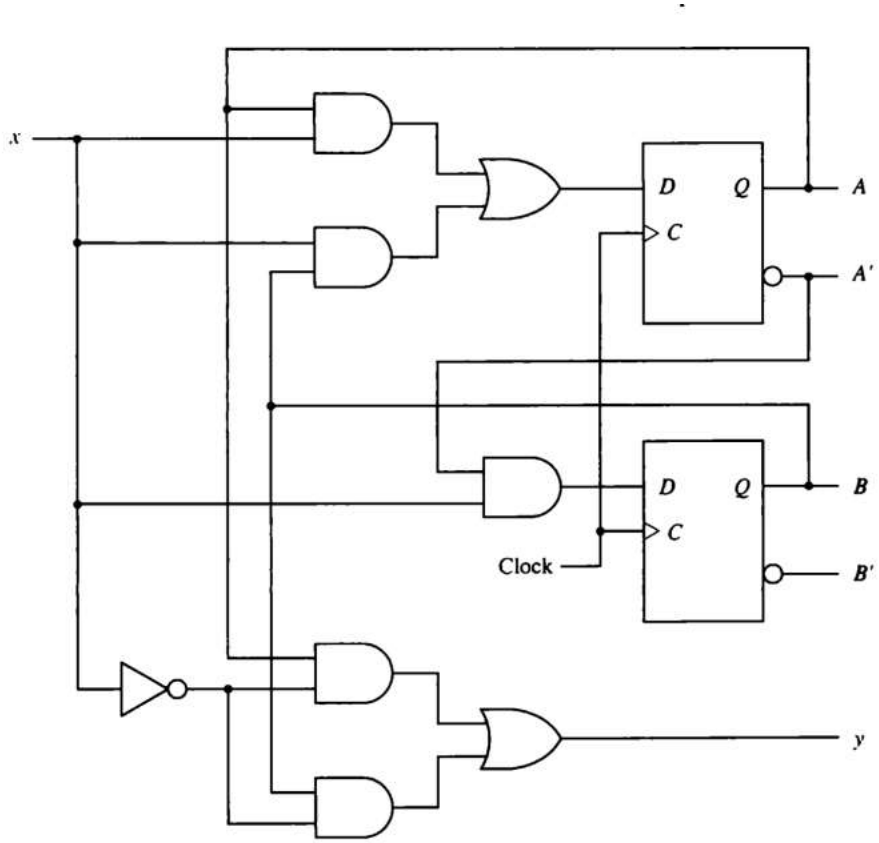
All in One



D Flip–flop (edge–triggered)
(positive edge triggering)

Revisiting sequential circuit: A complete picture





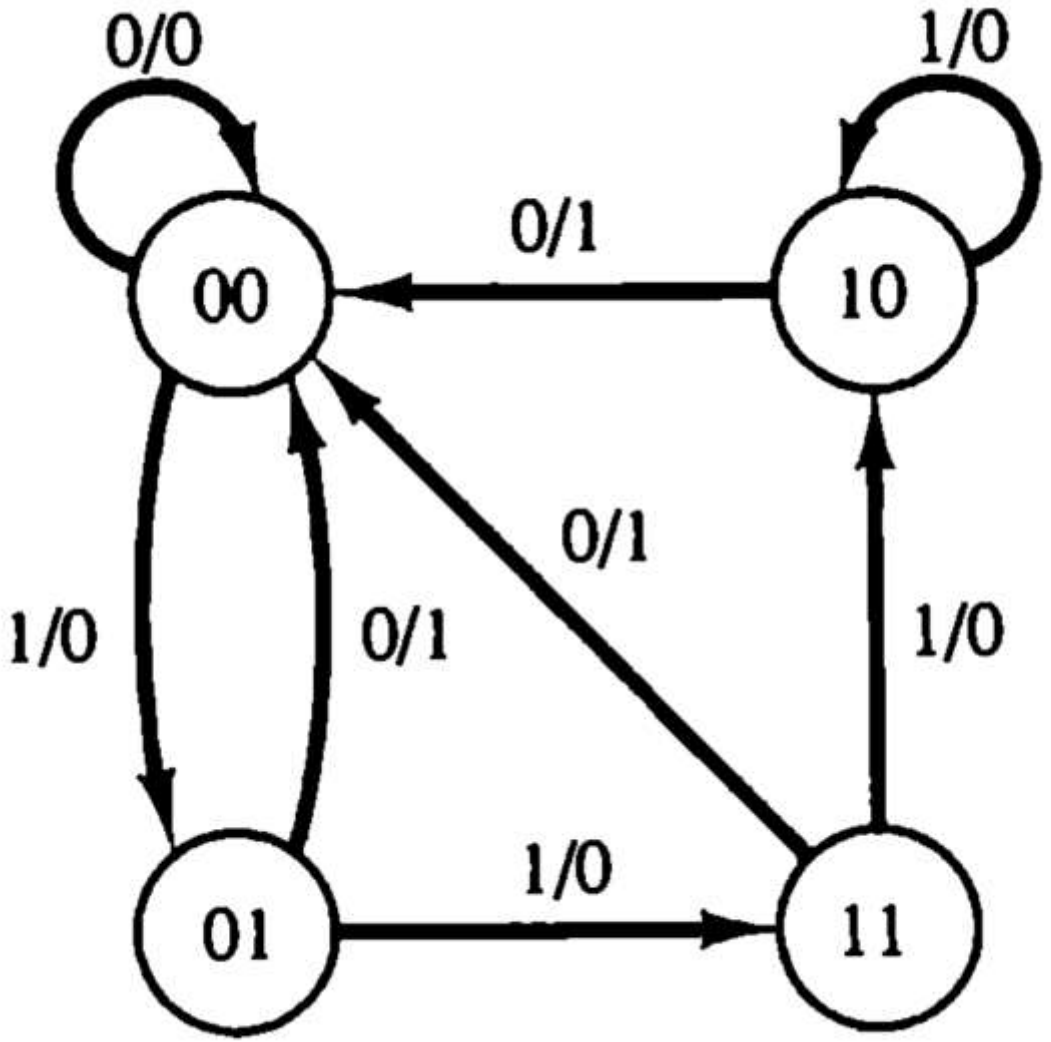
Present state		Input	Next state		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

A (t+1) = Ax + Bx, B (t+1) = A`x ,  y = Ax` + Bx`

State Table

State Diagram

x/y where x is input and y is output after the transition



Try it for a binary counter ☺

Lab-1

World of State machines (FSMs) Moore and Mealy Machines

Moore vs Mealy

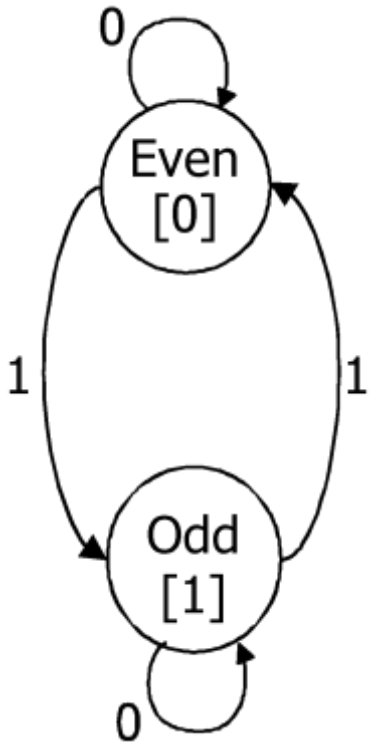
Moore machine: Output depends on the current state

Mealy machine: Output depends on the current state and inputs

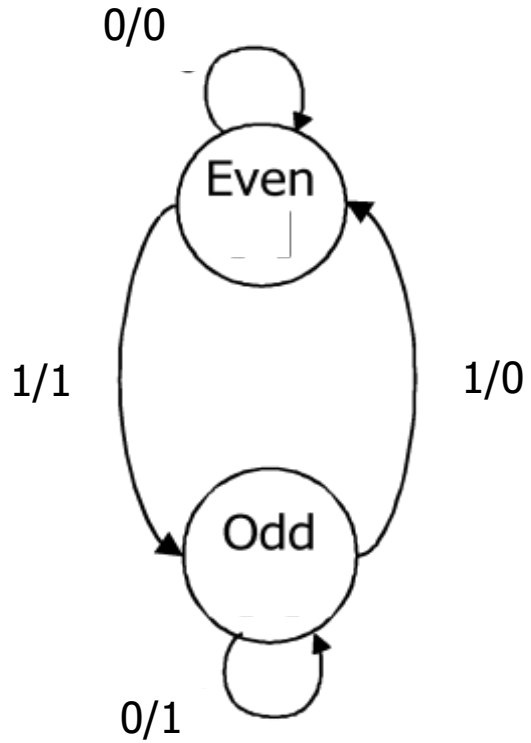
Odd Parity Checker

- Serial input string
  - OUT=1 if odd # of 1s in input
  - OUT=0 if even # of 1s in input
- Let's do this for Moore and Mealy

Moore



Mealy




Try on your own

State Transitions

Output changes only when the state changes

Appears after the state transition takes place

outputs change at clock edge

Even = 0

Odd = 1

Moore

Present State	Input	Next State	Output
Even	0	Even	0
Even	1	Odd	0
Odd	0	Odd	1
Odd	1	Even	1

Output changes when the state and input changes

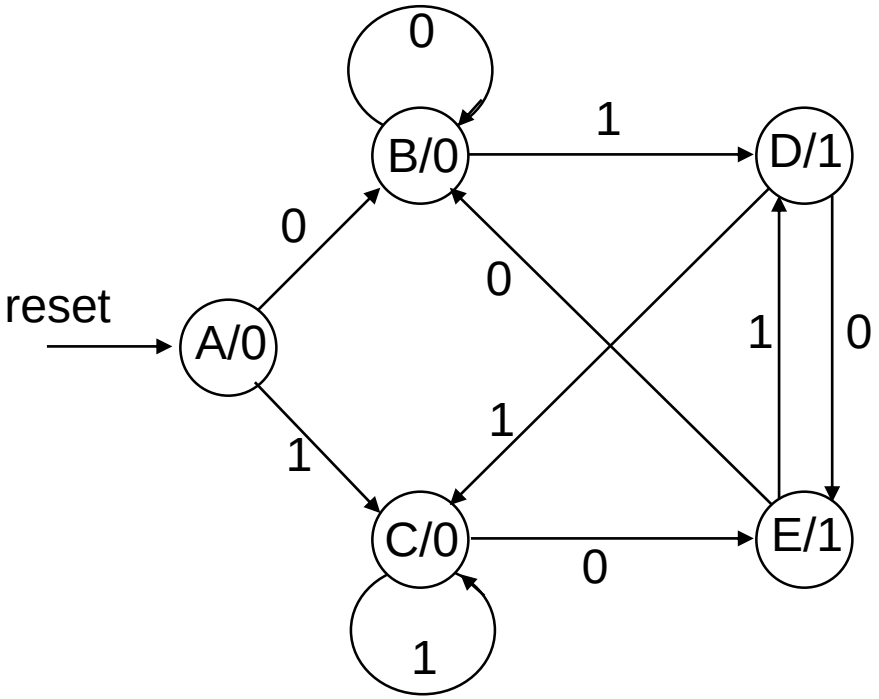
Appears before the state transition is completed

React faster to inputs — don't wait for clock

Mealy

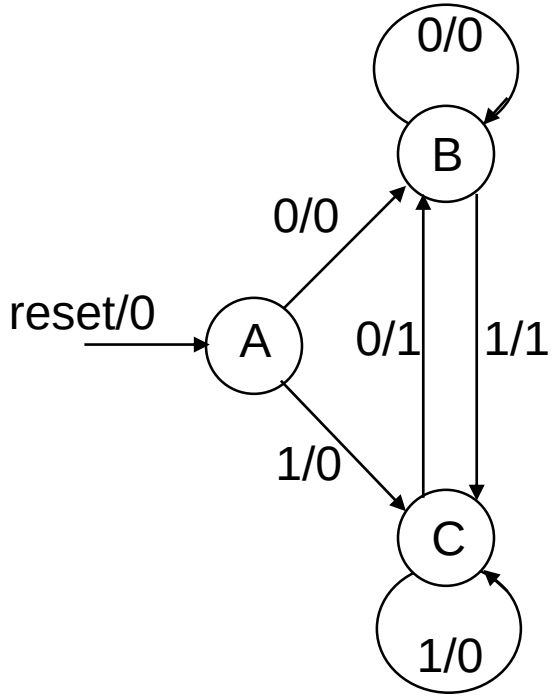
Present State	Input	Next State	Output
Even	0	Even	0
Even	1	Odd	1
Odd	0	Odd	1
Odd	1	Even	0

01/10 detector: Moore Machine



reset	input	current state	next state	current output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	D	0
0	0	C	E	0
0	1	C	C	0
0	0	D	E	1
0	1	D	C	1
0	0	E	B	1
0	1	E	D	1

01/10 detector: Mealy Machine

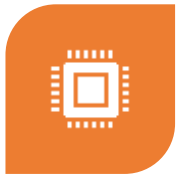


reset	input	current state	next state	current output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	C	1
0	0	C	B	1
0	1	C	C	0

PAUSE

Architecture-101

Next Few Lectures



HOW CAN A PROGRAMMER INTERACT WITH THE PROCESSOR?

THE LANGUAGE OF COMPUTER: INSTRUCTIONS

INSTRUCTIONS HAVE A VOCABULARY CALLED INSTRUCTION SET

DRIVEN BY INSTRUCTION SET ARCHITECTURE (ISA)

ISA: X86, ARM, RISC-V, MIPS

Coffee Credits

- Anirrudh + 1
- Tejas +1