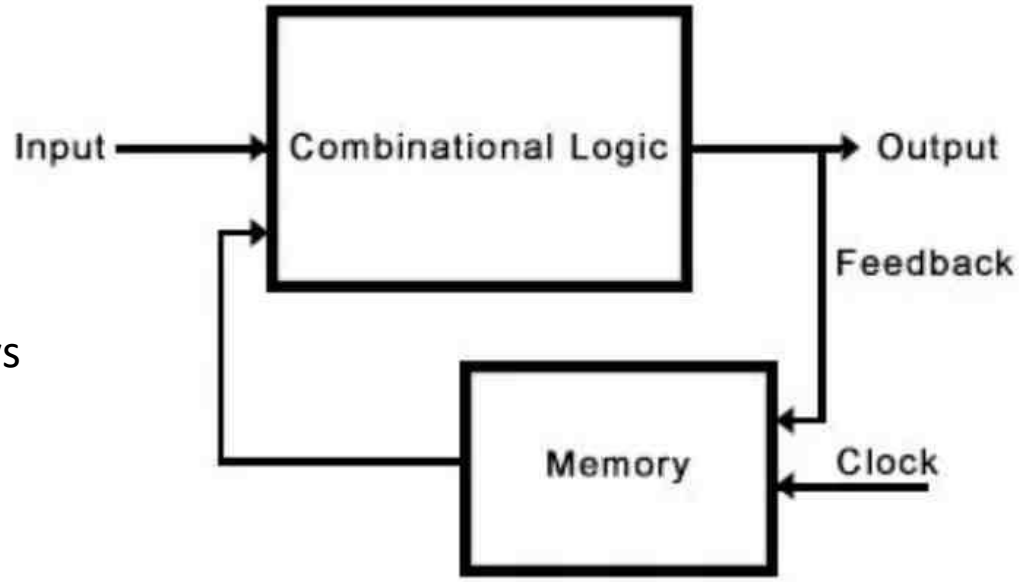
# EEE1024: Fundamentals of Electrical and Electronics Engineering

Dr. Sanchit Khatavkar

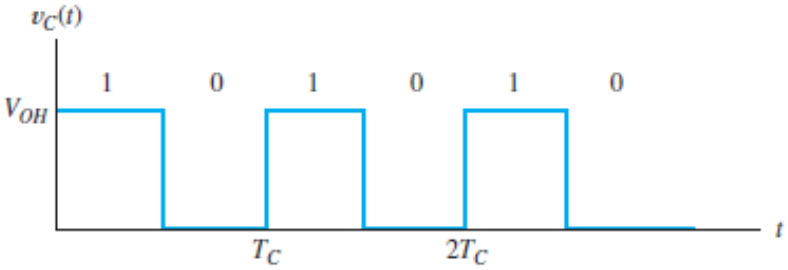
# Sequential circuits

Synchronous Sequential circuit:




Propagation delays

- MEMORY – remember past values
- CLOCK signal – Periodic logic-1 pulses




Asynchronous sequential circuits

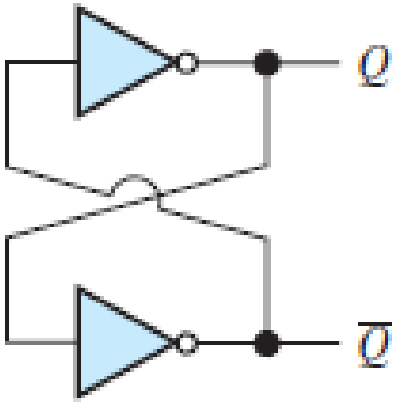
Racing condition

# Sequential circuits – Flip flops

*Basic building block of sequential circuits* – FLIP FLOP (FF)
*It is single bit storage device*

- Two stable operating states possible in a flip flop ⟶ 1 bit of information (0,1) can be stored
- Different types – based on how the clock and input signals control the state (o/p) of the FF.

*Simplest FF – using inverters*



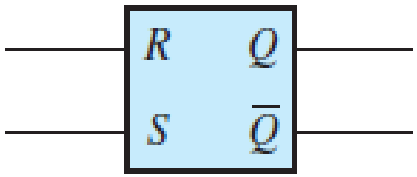

- **O/p of top inverter – HIGH, o/p of bottom inverter – LOW.**
- **Hence labeled as $\overline{Q}$**

- **O/p of top inverter – LOW, o/p of bottom inverter – HIGH.**
- **Any of these states possible and then can remain in this state indefinitely**

*How to control its state?*

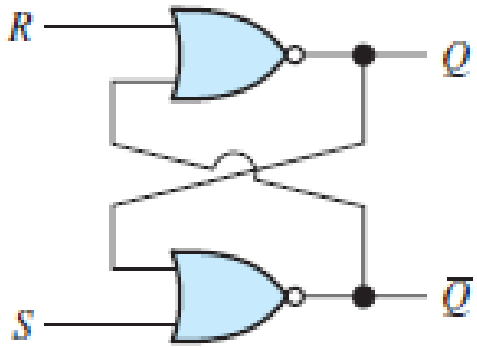
# SR FLIP FLOP

## SET (S) RESET (R) flipflop




Circuit symbol

- *Inverter replaced by* ***NOR gate***
- ***Latch: Unclocked flip-flop***




SR flipflop implementation

❖ *When S is HIGH and R is LOW, $\overline{Q}$ is forced to LOW and Q to HIGH (1)*

❖ *When S returns to LOW, Q stays HIGH: FF remains in SET state*

❖ *Now, S is LOW and if R becomes HIGH, Q is forced to LOW*

❖ *When R returns LOW, Q stays LOW: FF remains in the RESET state*

❖ *Both R and S not allowed to remain HIGH at the same time*

❖ *With both R and S LOW, FF remembers PREVIOUS STATE*

| A | B | C= A+B | C= $\overline{A+B}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

Truth table of NOR gate:

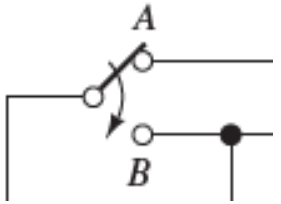
* *When any of the i/p is 1, o/p is 0*

| R | S | $Q_n$ | |
|---|---|---|---|
| 0 | 0 | $Q_{n-1}$ | Normal operation |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 1 | Not allowed | |

Truth table of SR flipflop

# SR flipflop - Application

*Eliminate effects of a switch bounce*

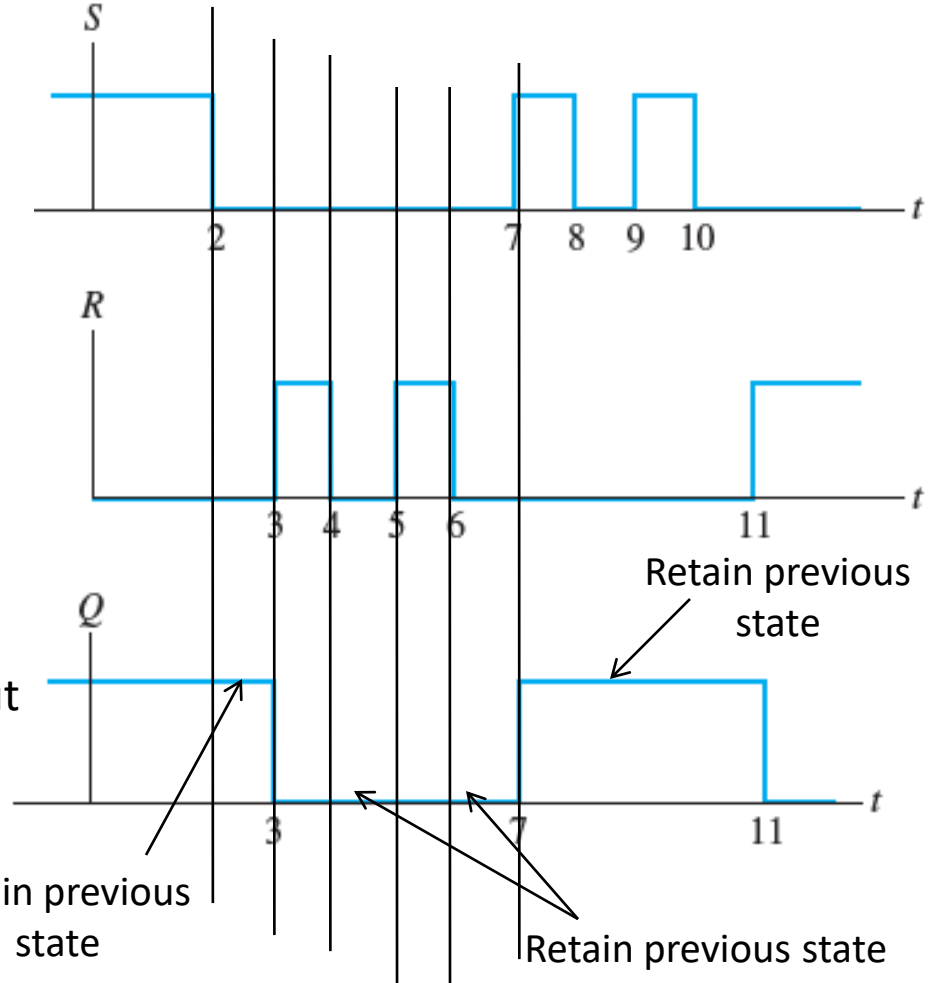



Q) The waveforms present at the input terminals of a SR flip flop are shown. Sketch the waveforms for Q versus time

S input pulse

R input pulse

Q- output pulse

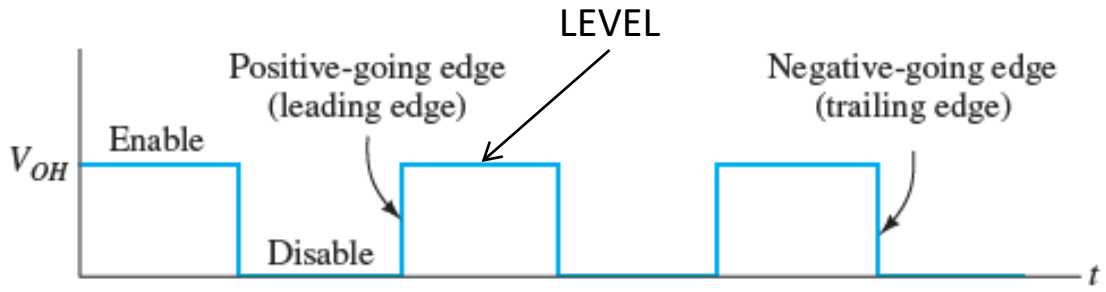



| $R$ | $S$ | $Q_n$ |
|---|---|---|
| 0 | 0 | $Q_{n-1}$ |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | Not allowed |

# Edge-triggered FLIP FLOPs

**Clocked SR flipflop – Latch :**

**Level triggered – *High clock level enables the inputs & low clock level disables it***




- **Edge triggered circuits – Respond to the inputs only on the *transition* of the *clock signal***
- **If clock signal is steady, i.e. either a HIGH or a LOW, inputs are disabled.**
- **At the clock transition, the flip-flop responds to just prior to the transition**

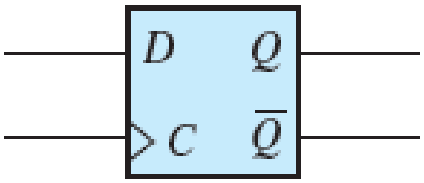
*Edge-triggered circuits*

***Positive** Edge triggered*:
Respond on ***leading* edge-** when clock goes from ***LOW to HIGH***

***Negative** Edge triggered*:
Respond on ***trailing* edge-** when clock goes from ***HIGH to LOW***

# D FLIP FLOP

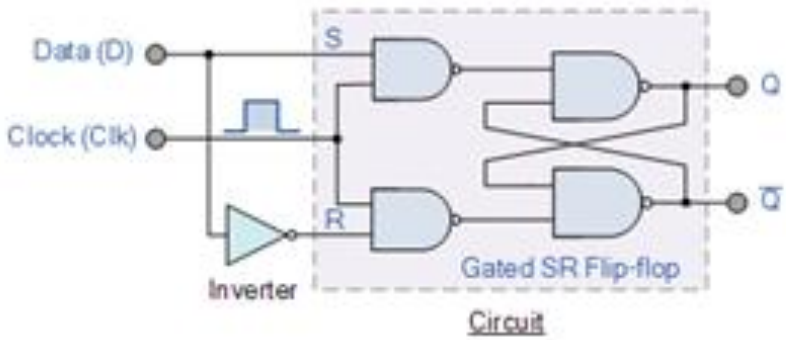
**D flip flop or Delay flip flop :**




Circuit symbol

| $C$ | $D$ | $Q_n$ |
|---|---|---|
| 0 | × | $Q_{n-1}$ |
| 1 | × | $Q_{n-1}$ |
| ↑ | 0 | 0 |
| ↑ | 1 | 1 |

Truth table of D flip flop




Logical Circuit implementation

- **Output takes the value of input that was present just prior to the triggering clock transition**

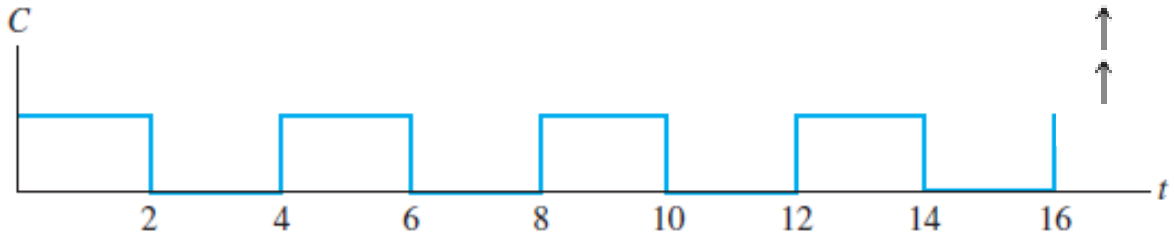
- **Up arrow in truth table – indicates – positive edge of the clock signal**

# D FLIP FLOP - Operation

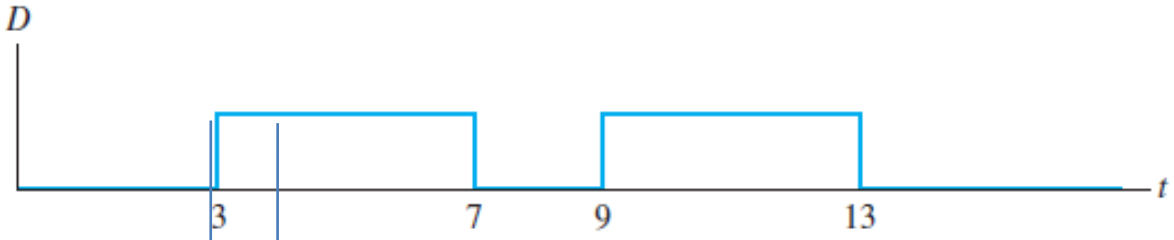
Q) The input signal to a positive edge triggered flip flop is shown.
Sketch the output Q versus time t

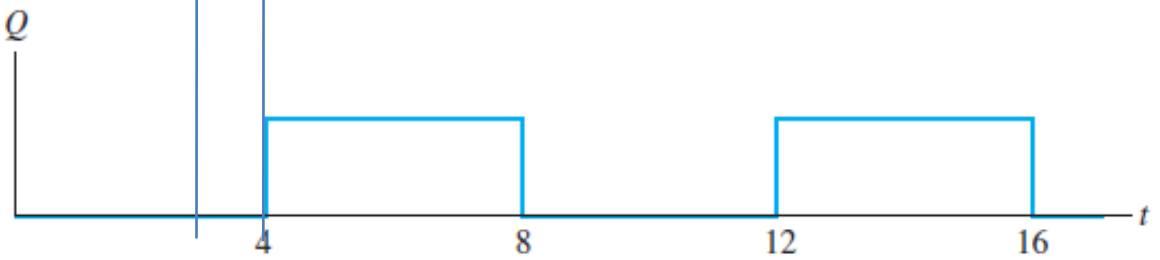
| $C$ | $D$ | $Q_n$ |
|---|---|---|
| 0 | × | $Q_{n-1}$ |
| 1 | × | $Q_{n-1}$ |
| ↑ | 0 | 0 |
| ↑ | 1 | 1 |

Clock signal



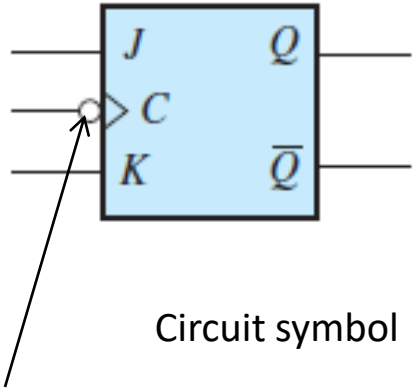

D input pulse



# JK FLIP FLOP

**JK flip flop : Most widely used!**

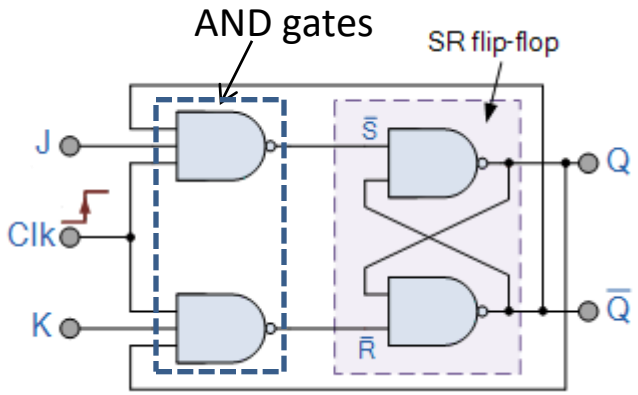



Circuit symbol

Negative edge triggered symbol

| C | J | K | $Q_n$ | Comment |
|---|---|---|---|---|
| 0 | × | × | $Q_{n-1}$ | Memory |
| 1 | × | × | $Q_{n-1}$ | Memory |
| ↓ | 0 | 0 | $Q_{n-1}$ | Memory |
| ↓ | 0 | 1 | 0 | Reset |
| ↓ | 1 | 0 | 1 | Set |
| ↓ | 1 | 1 | $\overline{Q}_{n-1}$ | Toggle |

Truth table of JK flip flop



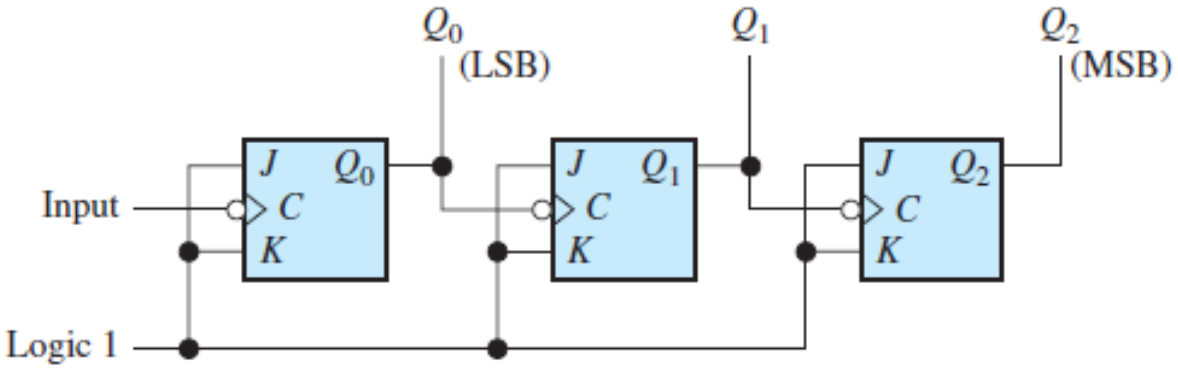

Logical Circuit implementation

- **Operation similar to an SR flipflop except that when both inputs J & K are HIGH, state changes on the next negative going clock pulse**
- **When both J & K are HIGH, output of JK flipflop toggles on each cycle of clock – Low to High on one –ve edge transition and again High to Low on the next.**

# JK FLIP FLOP – Application - COUNTER

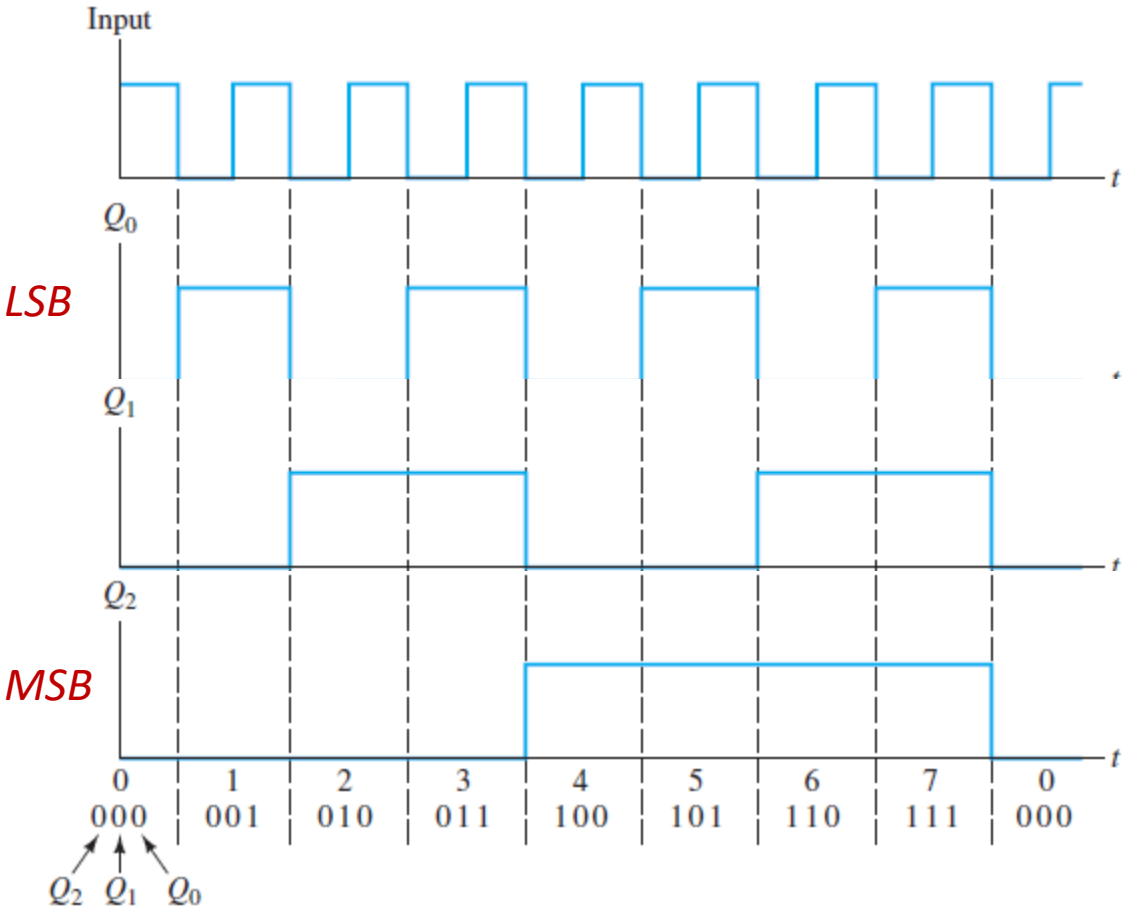
**Counters – used to count the pulses of the input signal – IMP part of the ALU**

**RIPPLE COUNTER: Cascade of J-K flip flops**




- **When both inputs are HIGH, Q output toggles, in a J-K flip flop (FF)**
- **Input pulses to be counted are connected to input of the 1st JK FF and output of the 1st JK FF is connected to the input of the 2nd JK FF - *Cascading***

# RIPPLE COUNTER Waveforms




- Assume – all FF's in the Reset state (Q=0)
- When the falling edge of the 1st i/p pulse occurs, $Q_0$ changes to o/p - **1**
- On the falling edge of the 2nd i/p pulse, $Q_0$ toggles back to **0**
- Falling input to 2nd stage, makes $Q_1$ go HIGH (**1**)
- After 7 pulses, the counter is in 111 state & on the 8th pulse, counter returns to **0.**

# Acknowledgements


1. Allan R. Hambley, 'Electrical Engineering - Principles & Applications, Pearson Education, First Impression, 6/e, 2013
2. https://www.electronics-tutorials.ws/sequential/seq_4.html