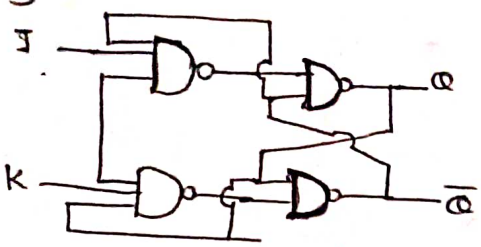
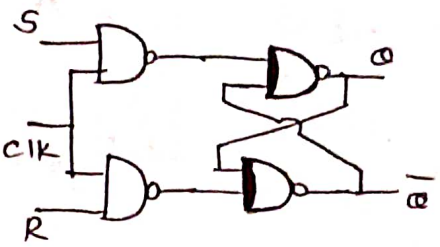
Ajinkya More07

1] Explain functioning of JK flipflop and SR flipflop

→

JK	SR
fig - 	fig - 
- It is an Enhancement of SR FF, designed to overcome the indeterminate state problem	It is Basic Type of FF that stores single bit of data
- operat^n mode i] Hold state when J=0, K=0, previous state ii] Reset when J=0 & K=1 iii] Set when J=1 & K=0 iv] Toggle when J=1 & K=1	operation mode 1] Hold state when S=0, R=0 2] Reset when S=0 & R=1 3] Set when S=1 & R=0 4] Invalid when S=1 & R=1
- Application Counter Register memory Freq Division	Basic memory storage control circuit Debounce ckt

2] Difference bet^n FlipFlop and Latches

→

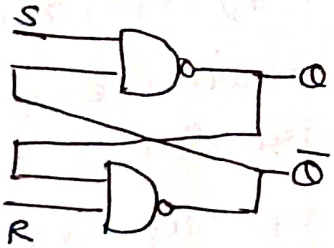
	FlipFlop	Latch
1]	FlipFlops are bistable device, it has 2 stable states , 0,1	Latch is also Bistable device states Represent as 0,1
2]	o/p changes on Rising edge or falling edge of clock when i/p is changed	o/p changes immediately w.r.t input, it monitors i/p continuosly
3]	It is edge triggered device	It is level triggered device
4]	It is made up of gates and latches	It is made up of only gates
5]	FlipFlops are slow Capared to latches	latches are faster
6]	more power is consumed	less power is consumed
7]	FlipFlops are clocked	Latches can be clocked or clockless
8]	Latches can generate faults or responsive to any fault on enable pin	FlipFlops are protected toward fault
9]	Types JK, SR, D, T FF	JK, SR, D, T latches
10]	It takes more area	It takes less area
11]	FF are synchronous	latches are asynchronous
12]	Flip flop are more Robust	latches are less Robust than FF
13	latches are used for Temporal data storage	Flip-Flops are used for permanent data storage

3] Why are Latches faster than FlipFlops

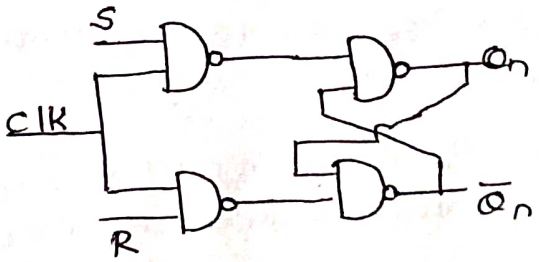
→

- Considering FlipFlop it's internal circuit has more logic level or layers, thus delay are larger in FF that mean delays get added.
- Also Latches are transparent during half cycle period & any input changes will reflect immediately on o/p side after combinational delay
- while in case of FlipFlop it gets updates on Rising or falling edges only
  It Req twice hardware as that of latch to implement

SR Nand latch



SR FF

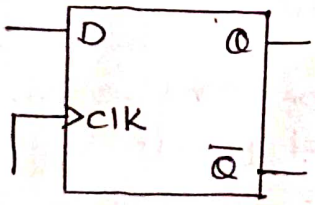


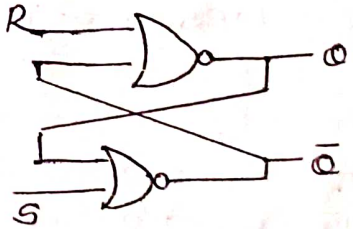
Ajinkya More07

4] Explain the use of
i] Flipflop
ii] Latches

→

- flipflops and Latches are fundamental building blocks in Digital Electronics.
- They are used for storing and controlling the data
- They serve as memory element for holding data and enabling sequential logic operation
- ff provides Timing control and are used in complex sequential logic application like counter shift Register, state machine
- Latches are simpler and are used for basic data storage & control application
- Flipflops ability to store and transfer data reliably makes essential components in memory devices, processors





S & Nor Latch

5] Why is the Gated SR FF called Asynchronous latch

- Gated SR FF are sometime referred to as asynchronous because they can change their o/p state immediately when their input change

- Gated S-R FF are level-sensitive, meaning their o/p can change as long as the enable signal is active It is different from edge-triggered FF.

- when the enable signal is active, any changes in the S & R inputs will immediately affect the o/p This means the o/p responds to the input as soon as they change, not waiting for a clock edge

- The output can change in direct response to input changes while the enable signal is active, this behavior is considered asynchronous.

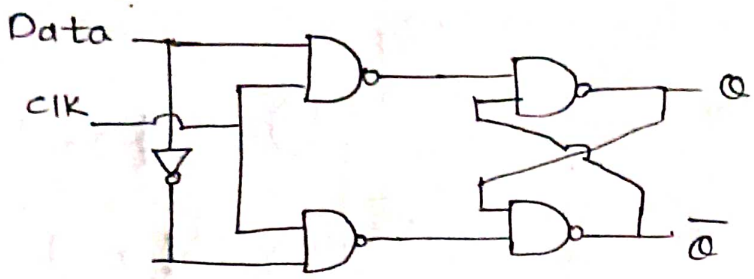
Example of Asynchronous Behaviour Gated SR FF

Consider a gated SR FF with enable-E, & i/p S,R

* Enable Active (E=1) The FF is open to i/p changes
  - i] If S=1 & R=0, the o/p Q is set to 1
  - ii] If S=0 & R=1, the o/p Q is set to 0
  - iii) If S=0 & R=0, the o/p Q remain in current state
  - iv) If S=1, R=1, the state is usually considered Invalid.

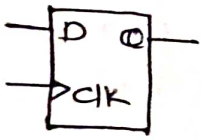
Enable Inactive :- The FF ignore i/p changes, & o/p remain unchanged.

6] Implement D-FF using Nand Gate



- The edge triggered D FlipFlop has only one i/p terminal
- The D FF may be obtained from S-R FF by inverting or putting inverter between the S and R terminal

Below Fig shows logic symbol and Truth table

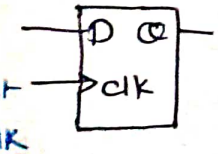


D	clk	Q	
0	↑	0	Reset
1	↑	1	Set

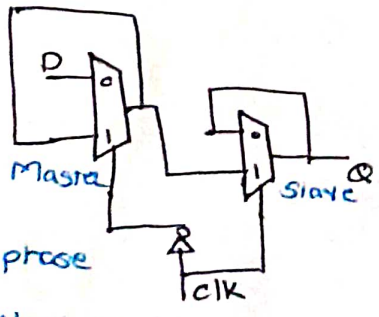
7] Design D-FF using 2:1 Mux

→

As shown in fig



master takes input on low phase of clk and slave takes input on high phase



slave always follows the data that master capture