MODULE-3

FLIP FLOPS

CLOCK WAVEFORMS -

* A clock is frequently used as a basis for timing all operation in a digital system.

→ The electronic circuit used to generate this square wave is referred to as "System Clock".

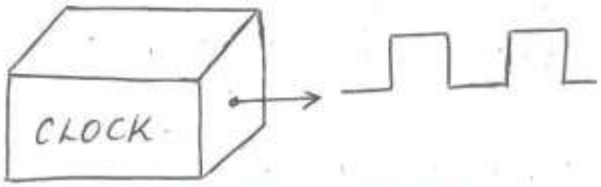


figure : System Clock

→ The square wave shown in fig (a) below is a typical clock waveform used in a digital system.

→ It should be noted that the clock need not be perfectly symmetrical waveform as shown.

→ It could simply be a series of positive (or negative) pulses as shown in the fig (b) below.

* But the main requirement is that the clock must be perfectly periodic, steady & stable.

→ Notice that each signal in the figure defines a basic timing interval during which logic operations must be performed. This basic timing interval is defined as "CLOCK CYCLE TIME" & its equal to one period of the clock waveform.

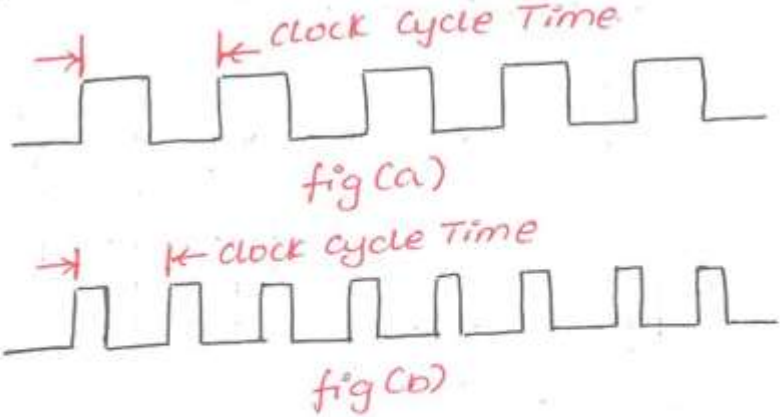


FIGURE : Ideal Clock Waveforms.

FLIP FLOPS -

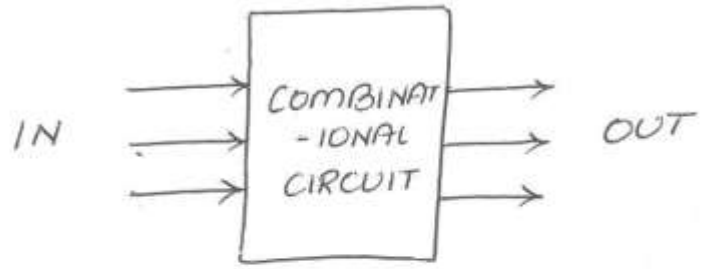
→ Until now we have been seeing circuits which are called as Combinational Circuits.

* A combinational circuit is defined as a circuit where the outputs are influenced only with the present inputs.

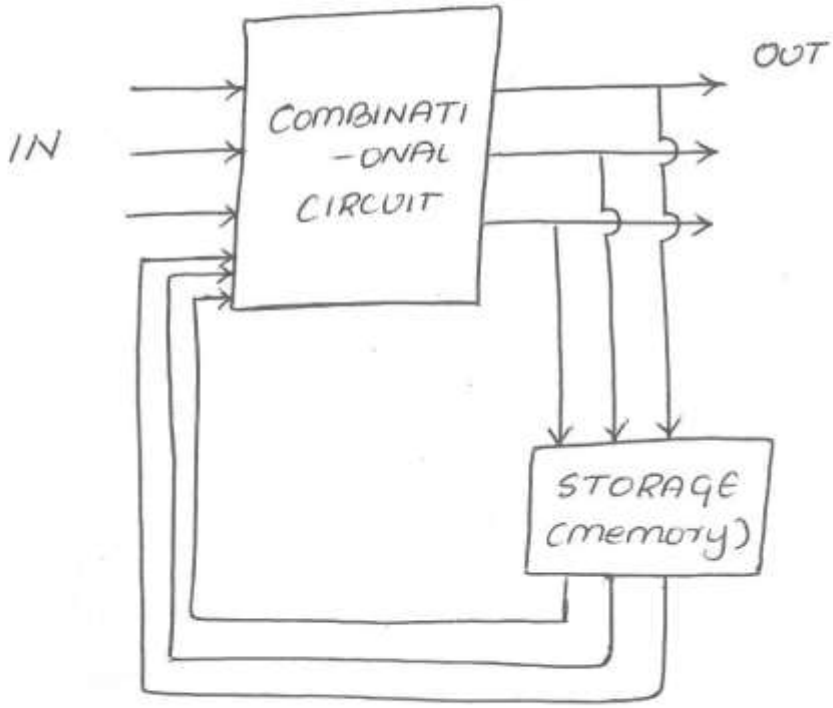
→ on the other hand, we have another class of circuits called as Sequential Circuits.

* A sequential circuit is defined as a circuit where the outputs depend on the present input as well as on the past behaviour of the circuit.

Combinational Circuit Model.



Sequential Circuit Model.



Difference between a Combinational and Sequential Circuit

COMBINATIONAL CIRCUITS	SEQUENTIAL CIRCUITS
* In combinational circuits, the output variables are at all times dependent on the combination of input variables.	* In Sequential circuits, the output variables are not only dependent upon the present input variables but they are also dependent upon the past history of these input variables.
* Memory unit is not required in combinational circuits.	* Memory unit is required to store the past history of input variables in sequential circuits.
* Combinational circuits are faster in speed, because the delay between the input & output is due to propagation delay of gates.	* Sequential circuits are slower than the combinational circuits.
* Combinational circuits are easy to design.	* Sequential circuits are comparitively harder to design.
* Example : Parallel Adder	* Example : Serial Adder.

→ The outputs of the digital circuits considered previously are dependent entirely on their inputs.
That is, if an input changes state, then the output may also change state.

→ However, there are requirements for a digital circuits whose output will remain unchanged once set, even if there is a change in the input levels.
Such a device could be used to store a binary no.
A flipflop is one such circuit, & the characteristics of the most common types of flipflops used in digital systems are considered in this chapter.

Any device (or) circuit that has two stable states is said to be "Bistable"

Example

① Toggle Switch -

It is either up (or) down, depending on the position of the switch.

* The switch is said to have a 'MEMORY' since it will remain as set until someone changes its position.

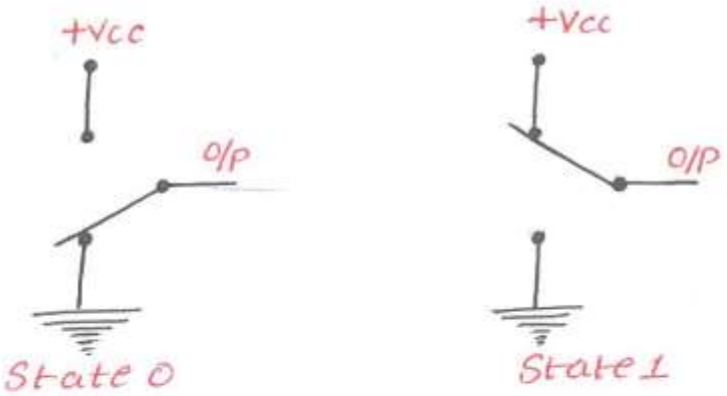


Fig: Toggle Switch.

② Flip-flop -

It is also a bistable electronic circuit that has two stable states ie, output is '0' (or) '+5 Vdc'.

* The flip flop also has memory since its output will remain set until something is done to change its state.
* As such flip flops can be regarded as a memory Device.
* The flip flop is often called as latch, since it will hold or latch to either of the two stable states.

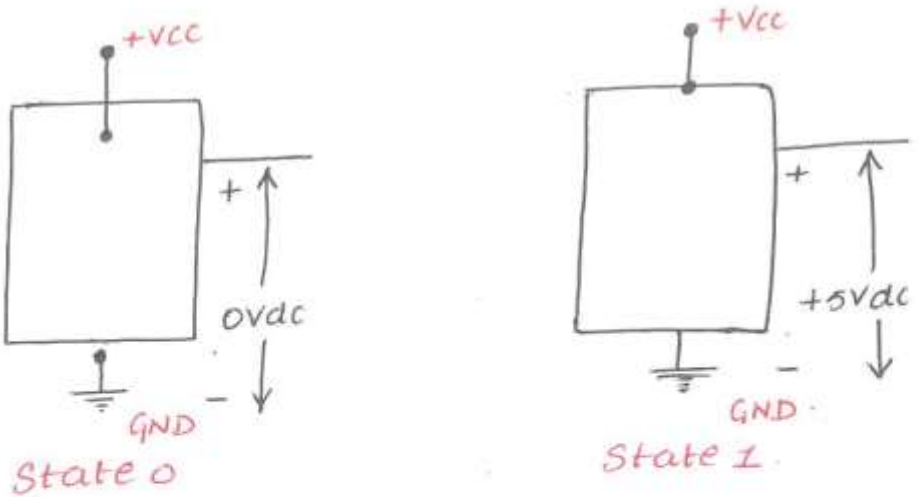
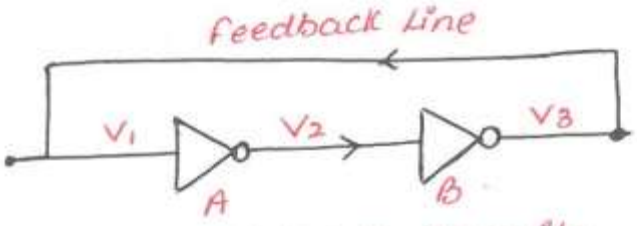


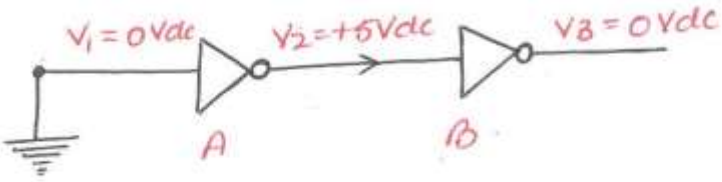
FIG: Flip flops

BASIC IDEA

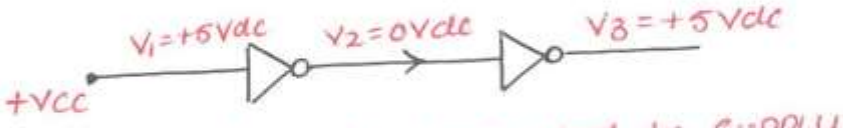
one of the easiest way to construct a flip flop is to connect two inverters in series as shown below.



(a) Bistable Circuit



(b) Input connected to GND



(C) Input connected to SUPPLY

NOR Gate Latch -

The basic storage element is a 'LATCH' and it is a very simple circuit.

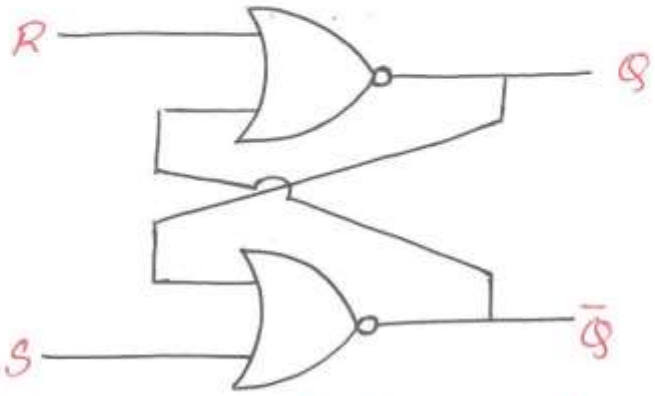


FIG: Circuit diagram of a NOR-Gate Latch

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

NOR Gate Truth Table

Case 1 $S=1$, $R=0$ $Q=1$, $\bar{Q}=0$

$S=0$, $R=0$ $Q=1$, $\bar{Q}=0$

Case 2 $S=0$, $R=1$ $Q=0$, $\bar{Q}=1$

$S=0$, $R=0$ $Q=0$, $\bar{Q}=1$.

Case 3 $S=1$, $R=1$ $Q=0$, $\bar{Q}=0$

$S=0$, $R=0$ "o/p's are unpredictable"

In the above NOR Gate latch circuit, one of the inputs for the 2 NOR Gates are R and S respectively.

The other two inputs are fed back from the outputs of those gates. Hence a Feedback is been introduced here.

Memory is feeding back, ie; putting the values back into the system.

The feedback is built, hence it should act as a 'latch' to store a bit either '0' or '1'.

The outputs are called as Q and $\bar{Q}$, that means that Q and $\bar{Q}$ are always complementing.

Applying all the possible inputs to the above latch

CASE 1: Let $S=1$ and $R=0$.

Since the previous outputs of Q and $\bar{Q}$ are not known, irrespective of the previous outputs, when we introduce an input of $S=1$ and $R=0$.

Since we know that when one of the input to the NOR gate is 1 the output is 0.

Hence the moment $S=1$, the output $\bar{Q}=0$. Then $R=0$ and $\bar{Q}=0$ will make $Q=1$.

ie; $S=1$, $R=0$. $Q=1$, $\bar{Q}=0$

CASE 2: $S=0$ and $R=0$.

When $S=0$ and $R=0$, by feeding the previous outputs to the NOR Gates, the outputs of the next state for Q will continue to remain 1 and that of $\bar{Q}$ will continue to be 0.

Even though I made $S=1$ and $R=0$ earlier to make $Q=1$ and $\bar{Q}=0$, when I remove S and made $S=0, R=0$ already we can see the effect of the previous output

ie; $S=0$, $R=0$ $Q=1$, $\bar{Q}=0$

CASE 2: Let $S=0$ and $R=1$.

The moment I make $R=1$, the output $Q=0$, and then $S=0$ and $Q=0$ will make $\bar{Q}=1$.

ie; $S=0$, $R=1$ $Q=0$, $\bar{Q}=1$

Now I am going to make $S=0$ and $R=0$, so the outputs Q will continue to be 0 and $\bar{Q}$ will continue to be 1.

ie; $S=0$, $R=0$. $Q=0$, $\bar{Q}=1$

Even though I made $S=0$ and $R=1$ earlier to make $Q=0$ and $\bar{Q}=1$, when I remove R and make $S=0$ and $R=0$ again we can see the effect of the previous output.

CASE 3 : Let S=1 and R=1.

Now we shall see the final combination ie; S=1 and R=1

So the outputs Q will become 0, since one input to the NOR gate is 1 and the output $\bar{Q}$ we also become 0, since the input S=1.

So here the complement condition of Q and $\bar{Q}$ is being violated.

Furthermore, when S=0 and R=0 what happens?

The outputs of Q and $\bar{Q}$ becomes unpredictable.

ie; S=1, R=1 Q=0, $\bar{Q}$=0

S=0, R=0 outputs unpredictable.

when S=0 and R=0, the outputs depend on whichever gate I start first to analyze.

In reality. we cannot have two gates which are identical in terms of delay.

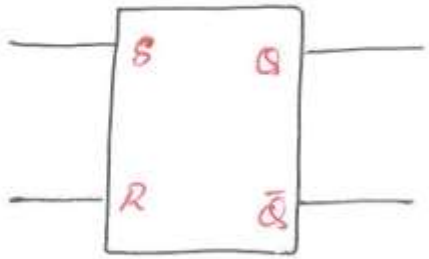
Hence, due to this time delay; whichever gate is faster that gate gets the output as 1. and the other gates gets the output as 0.

Such dependencies make the job of the designer very difficult.

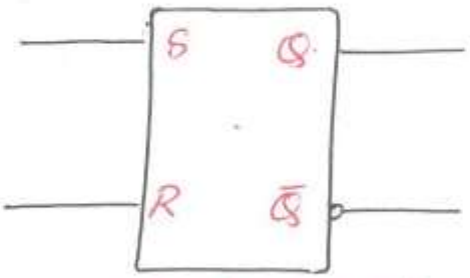
That is why, R=1 and S=1 condition is FORBIDDEN

(OR) UNRELIABLE.

The above figure is called a SR Latch.



IEEE Symbol



Logic Symbol

GATED FLIP FLOPS

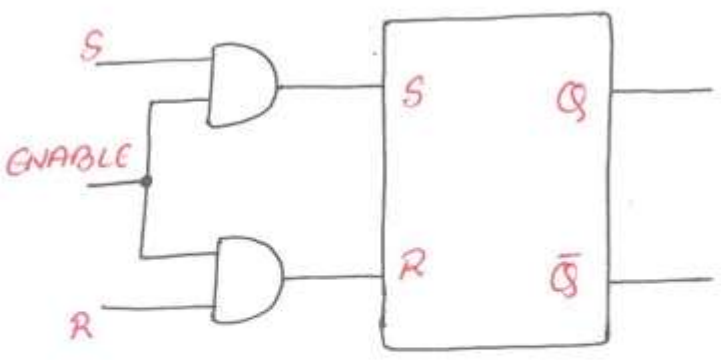
* Clocked RS Flipflops

The figure below shows the logic diagram of a SR/RS flipflop. It is similar to the RS latch, except that we need to provide one more input (ie; CLOCK) to that latch.

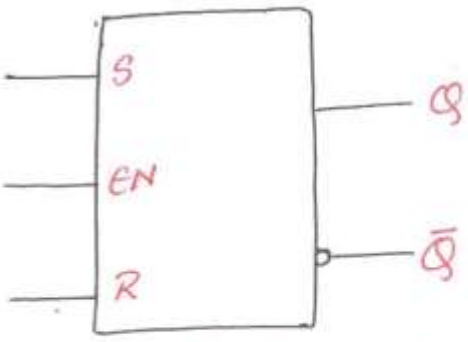
The clock input cannot be provided directly, so the addition of two AND Gates at the RS inputs along with the clock will result in a FLIP FLOP that can be enabled (or) disabled.

* When the ENABLE input is LOW, the AND gate outputs must both be LOW and changes in neither R nor S will have no effect on the flipflop output Q.
* When the ENABLE input is HIGH, the information at R & S inputs will be transmitted directly to the outputs.
  - → The output will change in response to the input changes as long as the ENABLE is high.
  - → When the ENABLE input goes LOW, the output will retain the information that was present on the input.

In this way it is possible to STROBE (or) CLOCK the flip flop in order to store the information at any time.



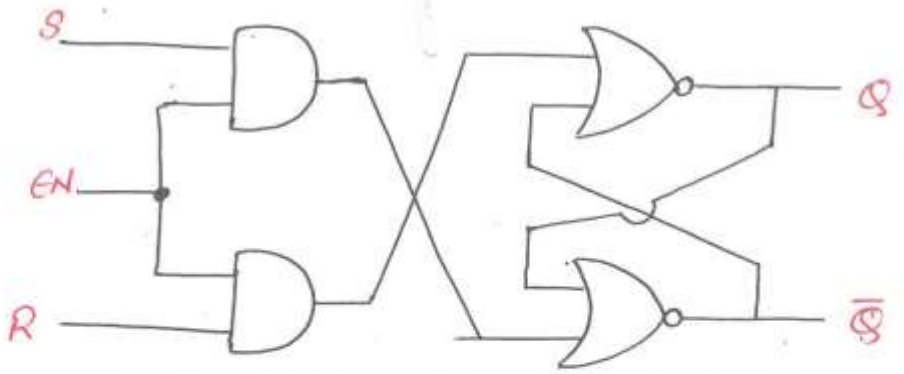
(a) Logic Diagram of Clocked RS Flipflop.



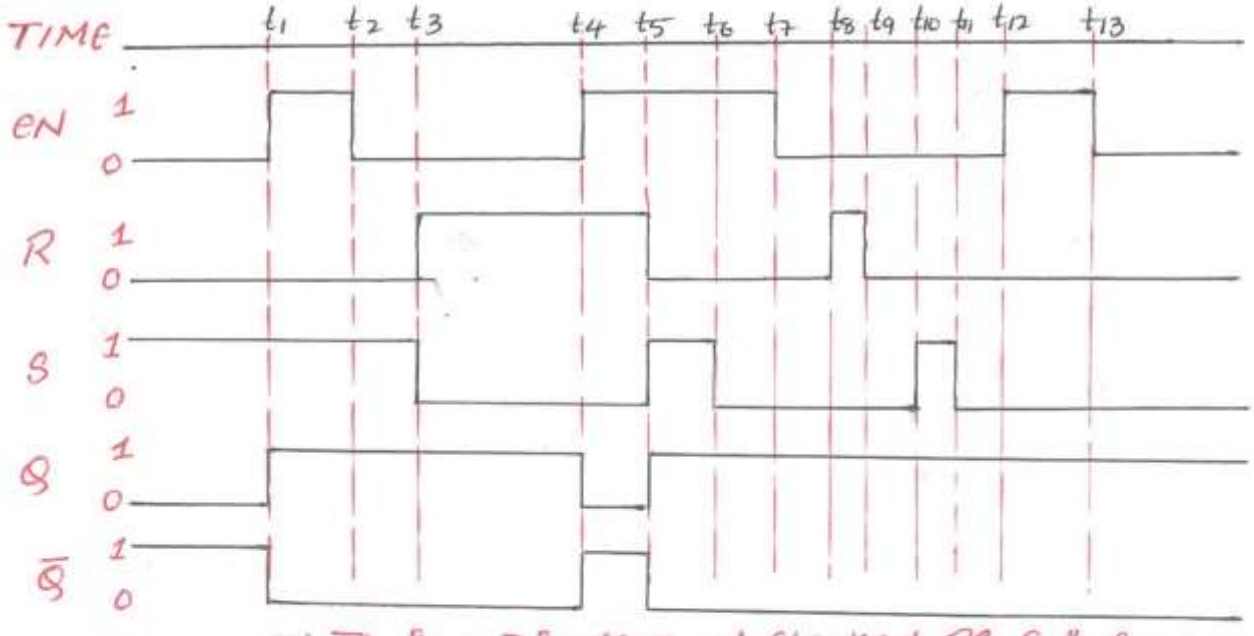
(b) Logic Symbol of clocked RS flipflop.

EN	S	R	Q_{n+1}	Action
0	X	X	Q_n	NO change
1	0	0	Q_n	NO change
1	0	1	0	RESET
1	1	0	1	SET
1	1	1	?	FORBIDDEN

(c) Characteristic Table of Clocked RS Flip Flop.



(d) Realization of clocked RS Flip Flop using NOR.

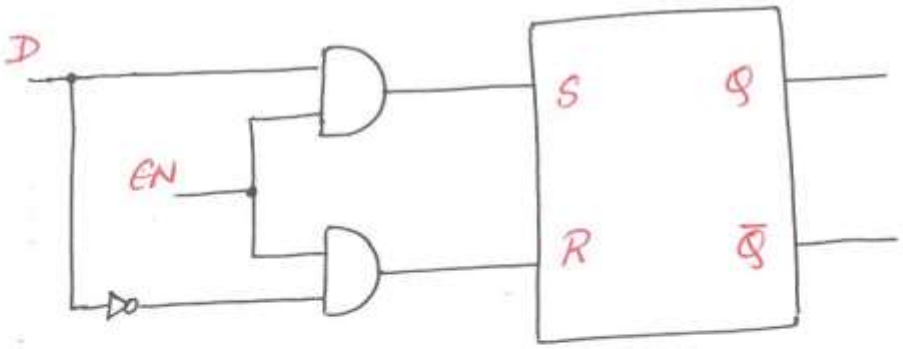


(e) Timing Diagram of Clocked RS Flip Flop.

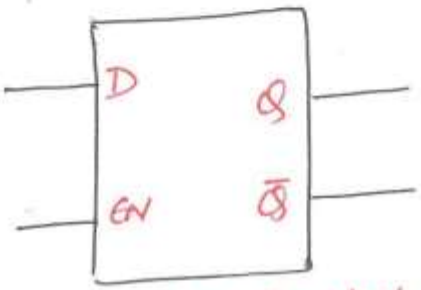
* CLOCKED D FLIPFLOP.

Generation of 2 signals to drive a flipflop is a disadvantage in many applications. Furthermore, the forbidden condition of both R & S high may occur inadvertently. This has led to the D flip flop, a circuit that needs only single data input.

The figure below shows a simple way to build a D flipflop. This flipflop is disabled when EN is low, but is transparent when EN is high. The action of this circuit is straight forward.



(a) Logic Diagram of clocked D-flipflop.



(b) Logic Symbol

EN	D	Q_{n+1}	Action
0	X	Q_n	No change
1	0	0	D
1	1	1	D

(c) Characteristic Table of clocked D flipflop.

→ To store the input, I give the input and enable the clock, to keep it all the same, all I have to do is to remove the clock (ie; make the clock low).

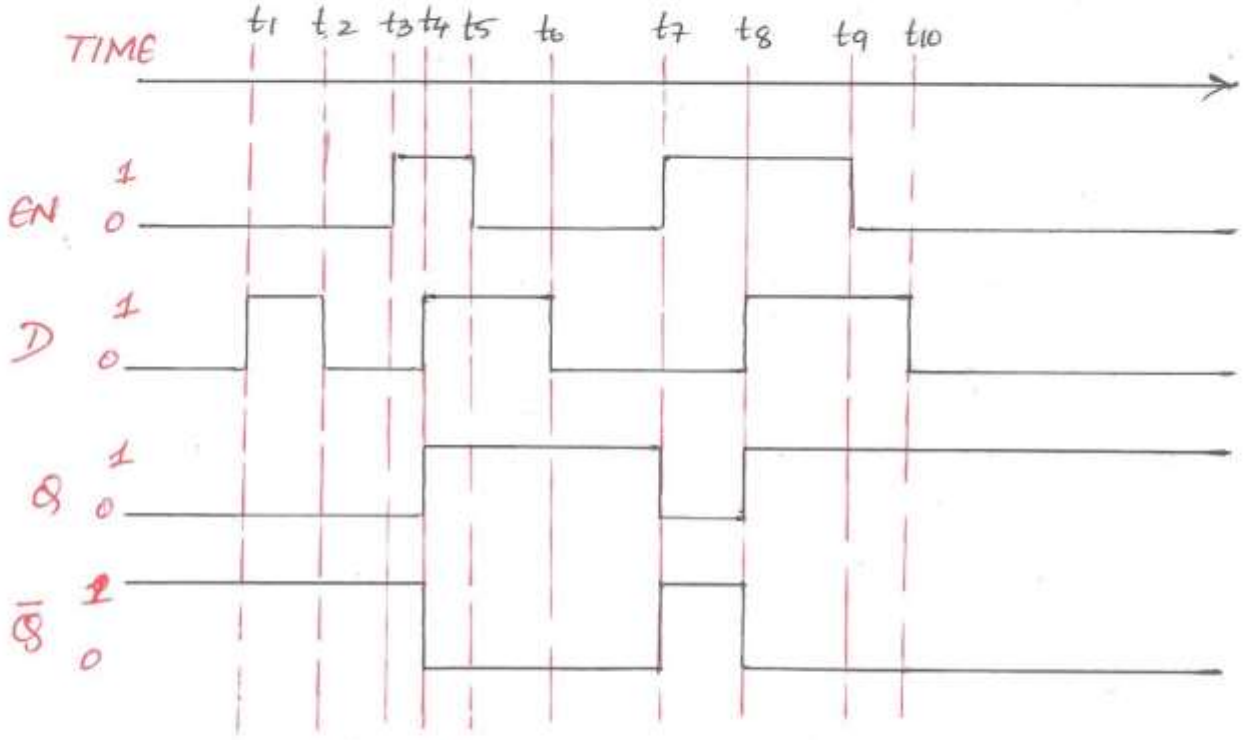
→ The data storage is always mostly done using the D flipflop.

→ The fig (a) shows a simple way to build a D flipflop. This flipflop is disabled when EN is low, but is transparent when EN is high.

→ The action of the circuit is straightforward as follows

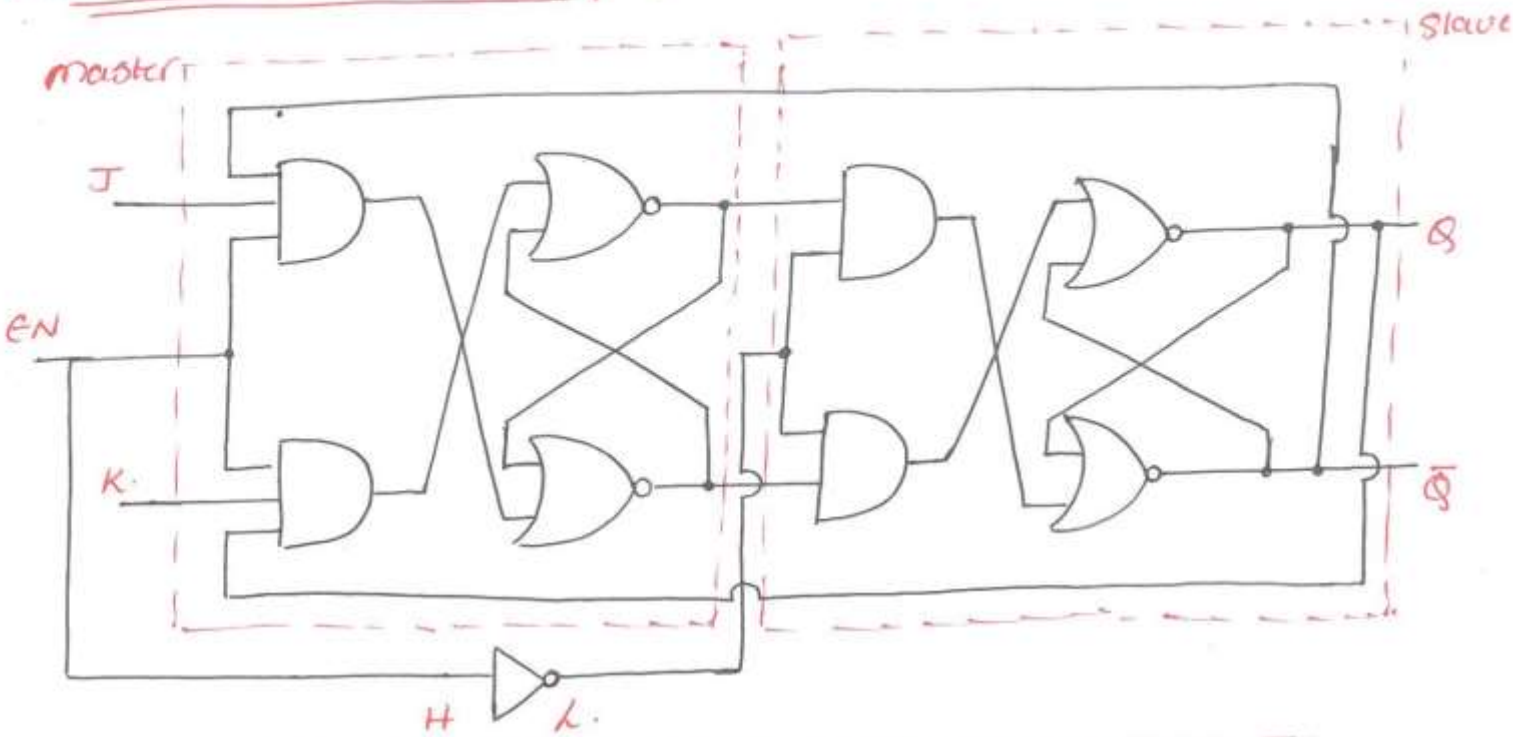
- When EN is low, both AND gates are disabled; therefore, D can change value without affecting the value of Q.
- On the other hand, when EN is high, both AND gates are enabled.
  In this case, Q is forced to equal the value of D.
  When EN again goes low, Q retains or stores the last value of D.

→ The fig(c) shows the truth table for a D latch.
When EN is low, D is a dorit care (X); Q will remain latched in its last state.
When EN is high, Q takes on the last value of D.

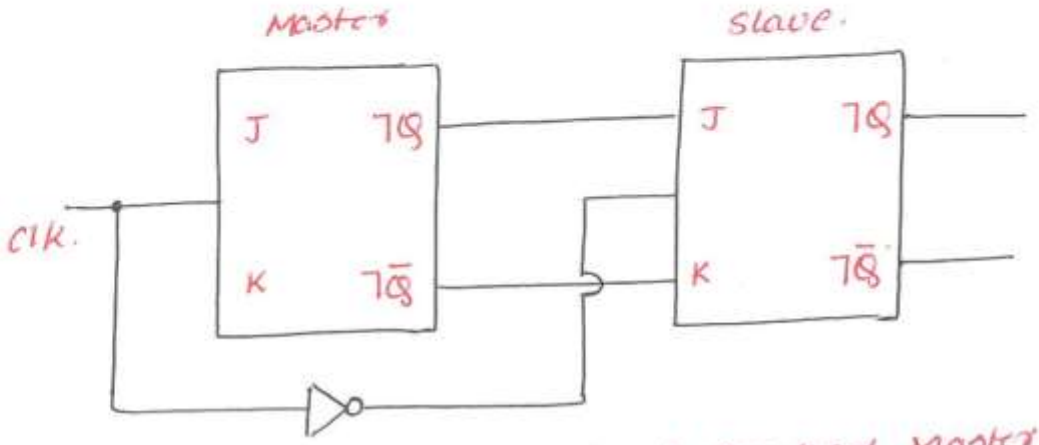


(d) Timing Diagram of D flip flop.

* Master Slave JK flipflop -



(a) Logic Diagram of Clocked Master Slave JK flipflop.



(b) Logic Symbol of Clocked Master Slave JK flipflop.

→ The above circuit has two flipflops, the clock is common but one of them is fed directly and the other clock is fed through an inverter.

→ Now the above circuit will behave exactly like a JK flipflop, except that this output will change only once in a clock cycle for the condition S=1 and R=1.

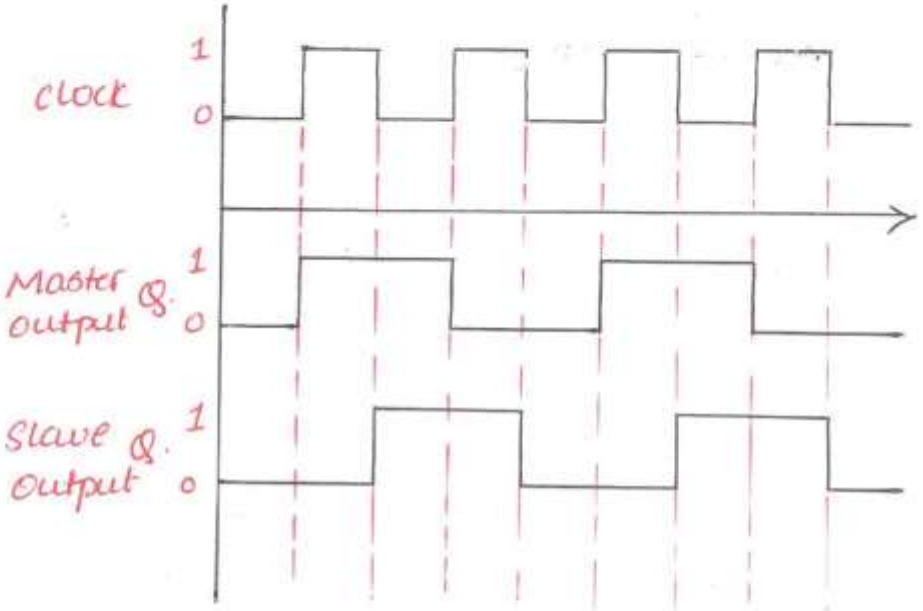
→ When you apply the clock, it will be high for the Master and low for the Slave.
The output of the Master FF is fed as input to the Slave FF.

→ Therefore, when the clock is high for this Master flipflop, this master FF will change state but the feedback is not going to happen because feedback must be given by the slave flipflop. So the constantly changing & the racing condition is not going to happen.

→ Constant toggling is not there, but it changes only once in a clock period.

EN	J	K	Qn+1	Action
0	X	X	Qn	last state
1	0	0	Qn	last state
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	$\overline{Q_n}$	Toggle

(c) Characteristic Table for Master Slave JK Flipflop.



(d) Timing Diagram of Master Slave JK Flip Flop.

→ The problem that existed in the SR flipflop for S=1 & R=1 has been eliminated using the Master Slave JK flip flop.

ADVANTAGE -
- Racing is avoided & toggling happens only once in a clock cycle.

DISADVANTAGE -
- There is extra H/w involved, ie; the slave H/w is almost double the effort in terms of no. of gates used.

Edge Triggered Flipflops

* Edge Triggered RS Flipflop

→ The simple latch-type flipflops presented before are completely "transparent"; ie, the output 'Q' immediately follows any change of state at the input.

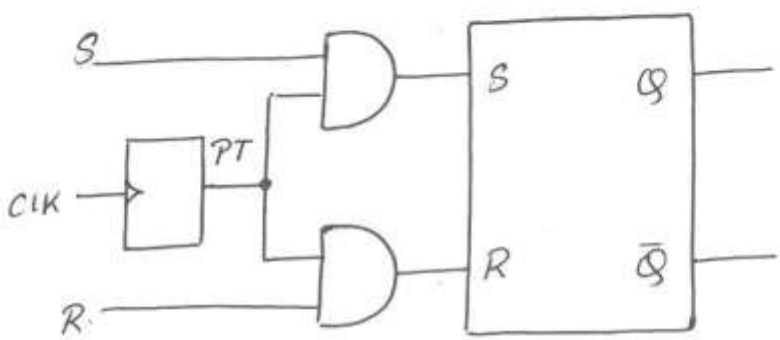
→ The gated (or) clocked RS & D flipflops can be considered "Semi Transparent", ie; the output 'Q' will change state immediately provided that the EN input is high.

→ If any of these flipflops are used in synchronous systems, care must be taken to ensure that all the flipflop inputs change state in synchronous with the clock.

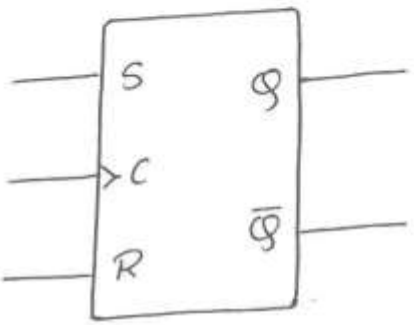
* One way of resolving the problem for gated flipflop is to allow changes in the input R, S and D levels only when the EN is low.

→ Thus the EDGE TRIGGERED FLIPFLOP was developed to overcome these restrictions.

POSITIVE-EDGE TRIGGERED RS-Flip Flop's



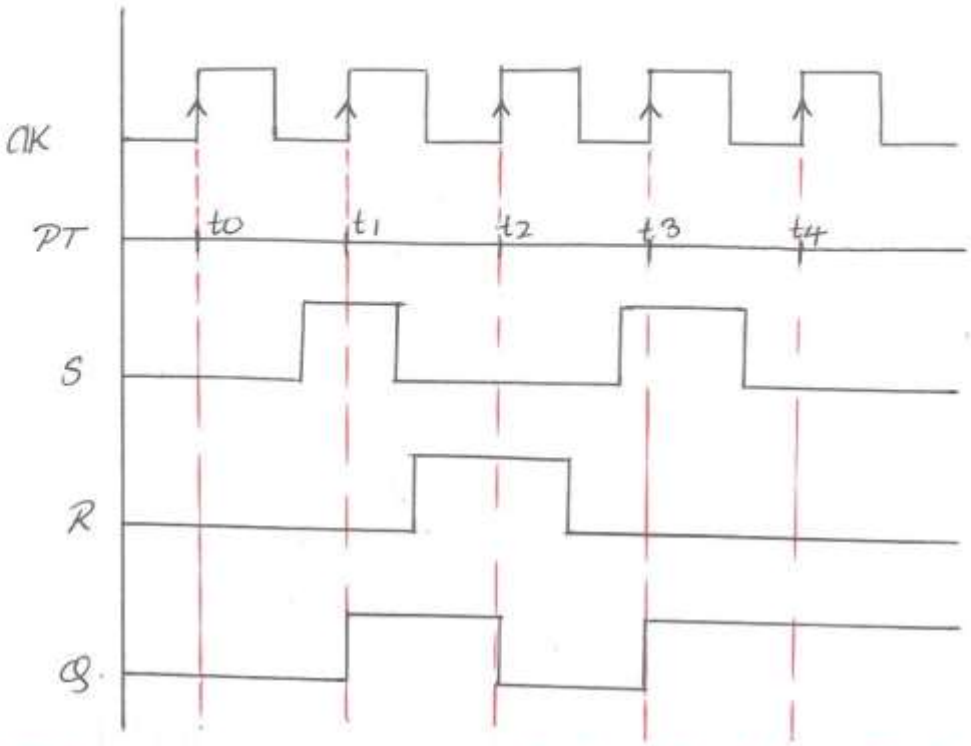
(a) Logic Diagram



(b) Logic Symbol

CLK	S	R	Q_{n+1}	Action
↑	0	0	Q_n	No change
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	?	ILLEGAL

(c) Characteristic Table



(d) Timing Diagram of Positive Edge Triggered RS Flip Flop.

→ In the above fig (a), the clock is applied to a POSITIVE Pulse forming circuit

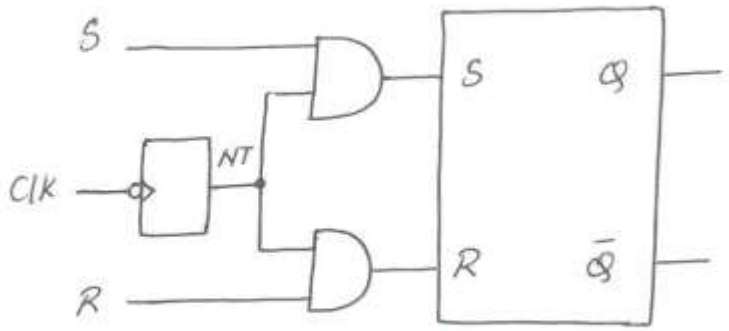
→ The Positive Transitions (PT's) developed are then applied to a RS-flipflop. The result is a positive edge triggered RS FF.

→ The fig (b) shows the logic symbol. The small triangle inside the symbol indicates that Q can change state only with the PT's of the clock.

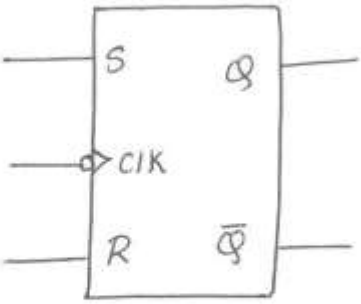
→ Each PT of the clock are applied to the AND gates. The AND gates are active only while the PT is high and thus Q can change state only during this short time period. In this manner "Q changes state in synchronous with the PT's of the clock".

→ Another way of expressing its behaviour is to say that the flipflop is transparent only during PT's.

NEGATIVE EDGE TRIGGERED RS - Flipflops



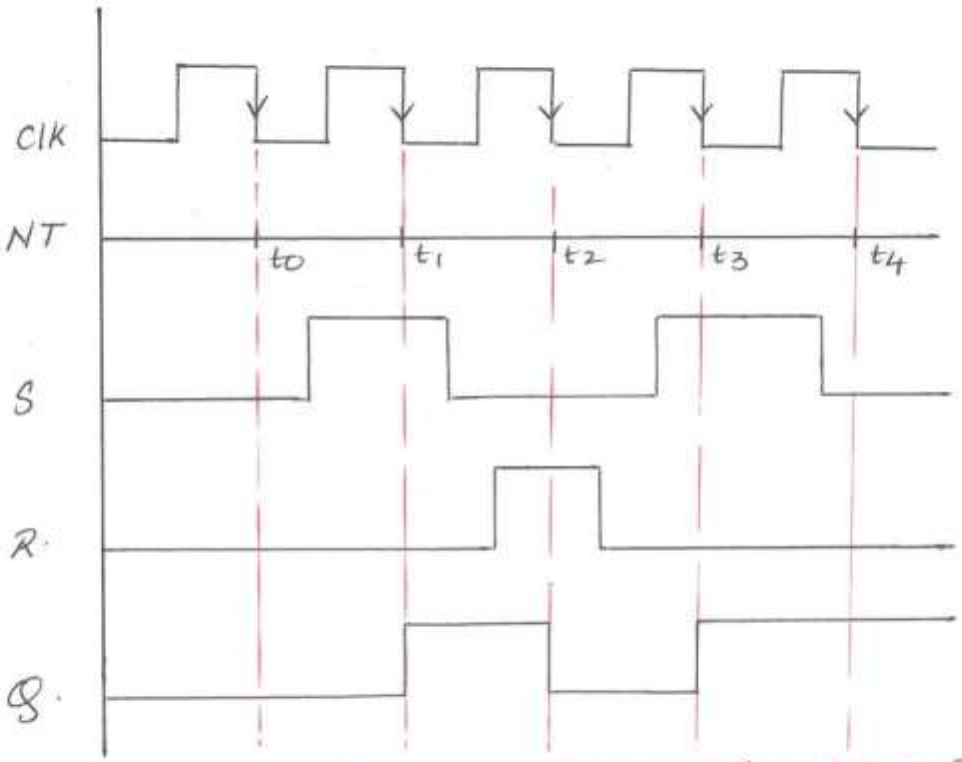
(a) Logic Diagram



(b) Logic Symbol.

CIK	S	R	Q_{n+1}	Action
↓	0	0	Q_n	No change
↓	0	1	0	RESET
↓	1	0	1	SET
↓	1	1	?	ILLEGAL

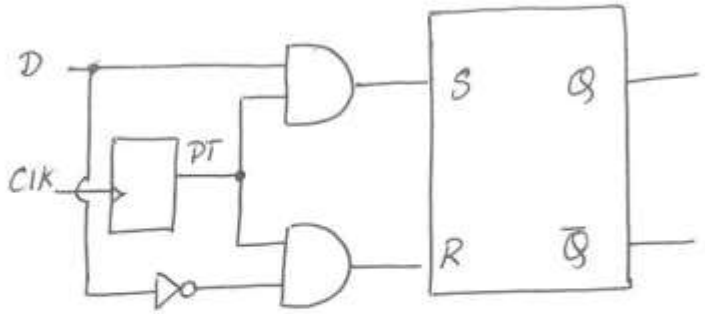
(c) Characteristic Table.



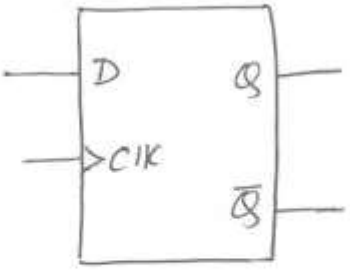
(d) Timing Diagram of Negative Edge Triggered RS flipflop.

NOTE : Give the Explanation for Negative Edge Triggered RS ff just the same way given for Positive Edge Triggered RS ff.

* Edge Triggered D-Flipflops.



(a) Logic Diagram of +ve Edge Triggered D-Flip flop.



(b) Logic symbol of +ve Edge Triggered D Flipflop.

"POSITIVE EDGE TRIGGERED D FLIP FLOP".

CIK	D	Q_{n+1}
0	X	Q_n (no change)
↑	0	0
↑	1	1

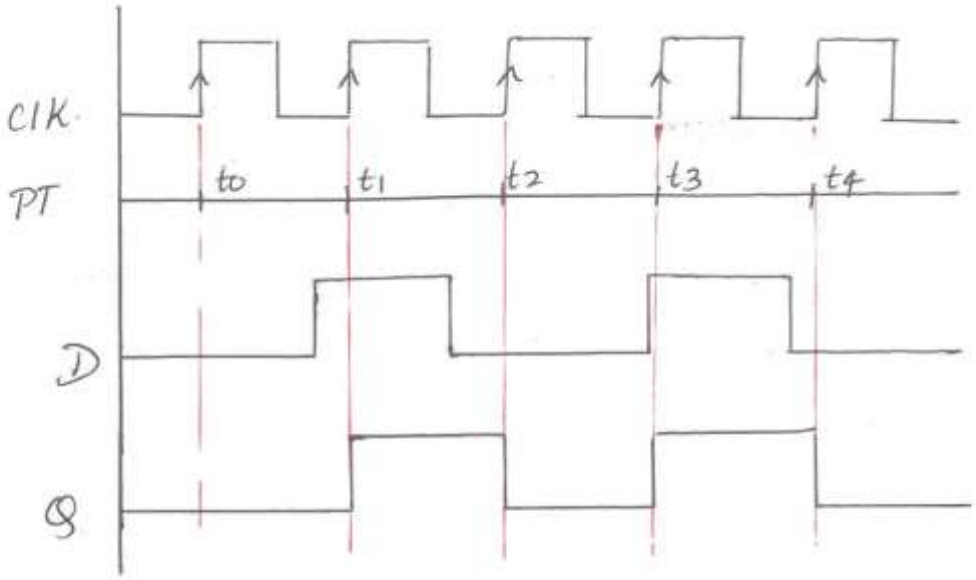
(c) Characteristic Table for +ve Edge Triggered D-Flipflop

→ The triggering of the above circuit occurs on the positive going edge of the clock; that is why its reffered to as +ve edge triggering.

→ The narrow positive pulse (PT) enables the AND Gates for an instant. The effect is to activate the AND gates during the PT of the clock.
At this unique point in time D and its complement hit the flip flop inputs, forcing the output Q to SET or to RESET.

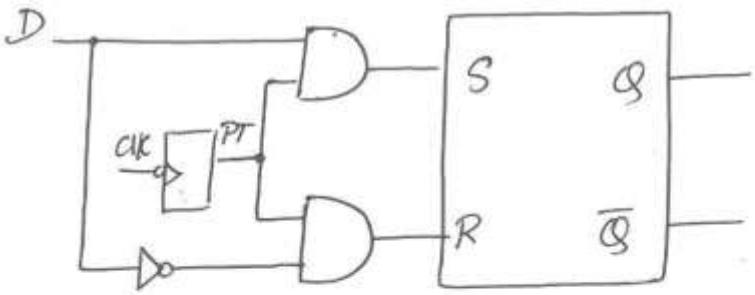
→ The truth table in fig (c) summarizes the action of the positive edge triggered D flipflop.
When the clock is low, D is a don't care and Q is latched to its last state.

→ On the leading edge of the clock (PT), designated by the up arrow, the data bit is loaded into the flipflop & and Q takes the value of D.

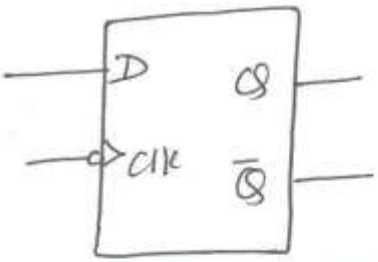


(a) & Timing Diagram of Positive Edge Triggered D-flip flop.

NEGATIVE EDGE TRIGGERED D flip flop.



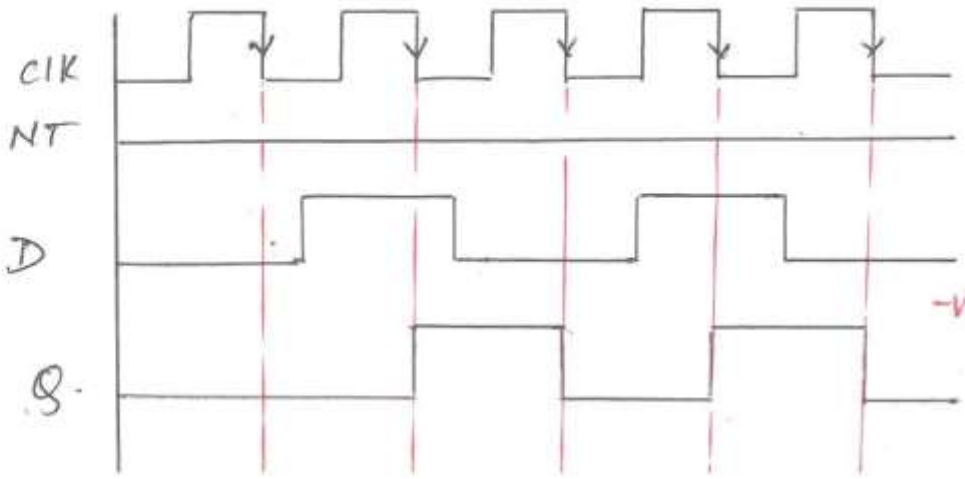
(a) Logic Diagram.



(b) Logic symbol.

CIK	D	Q_{n+1}
0	X	Q_n (No change)
↓	0	0
↓	1	1

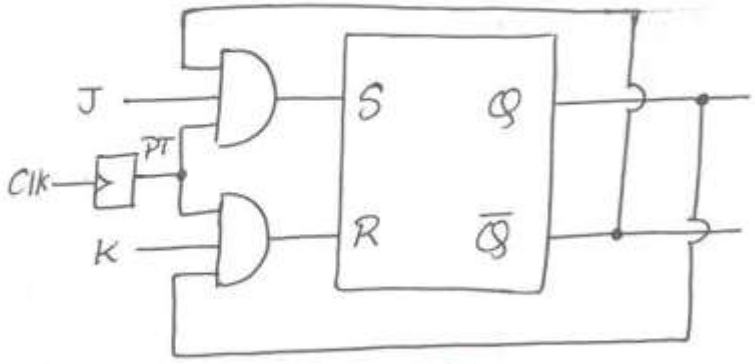
(c) Characteristic Table.



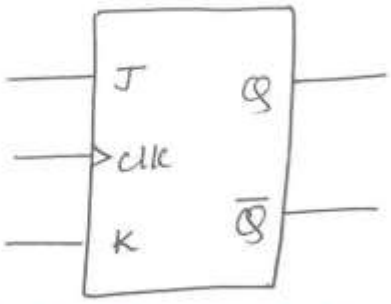
(d) Timing Diagram for -ve Edge Triggered D flip flop.

* Edge Triggered JK Flip Flop -

POSITIVE EDGE TRIGGERED JK FLIPFLOP.



(a) Logic Diagram



(b) Logic Symbol.

CIK	J	K	Q_{n+1}	Action
↑	0	0	Q_n	NoChange
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	$\overline{Q_n}$	TOGGLE.

(c) Characteristic Table.

In the fig(a), the pulse forming box changes the clock into a series of positive pulses & thus the circuit will be sensitive to PT's of the clock. The basic circuit is identical to the previous positive edge triggered RS-flipflop, with two important additions:

i) The o/p Q is connected back to the i/p of the lower AND Gate.

ii) The o/p $\overline{Q}$ is connected back to the i/p of the upper AND Gate.

This cross coupling from o/p's to i/p's changes RS flipflop into JK flipflop.

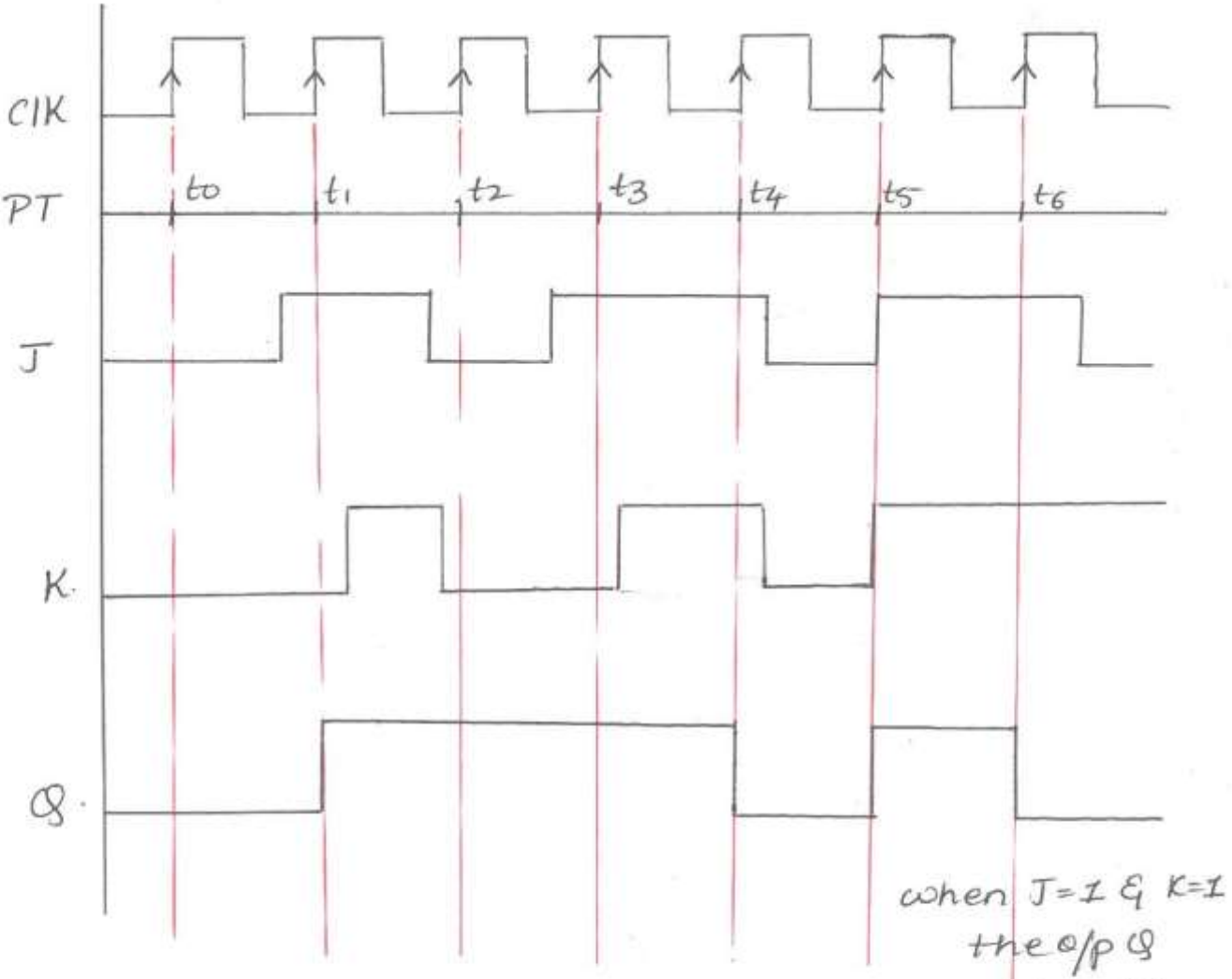
Here is how it works:

i> when J=0, K=0 ; Both AND gates are disabled. This is the initial entry in the truth table. Q retains its last value.

ii> when J=0, K=1 ; means that the next PT of the clock resets the flip flop. (unless Q is already Reset).

iii> when J=1, K=0; means that the next PT of the clock sets the flip flop (unless Q is already set).

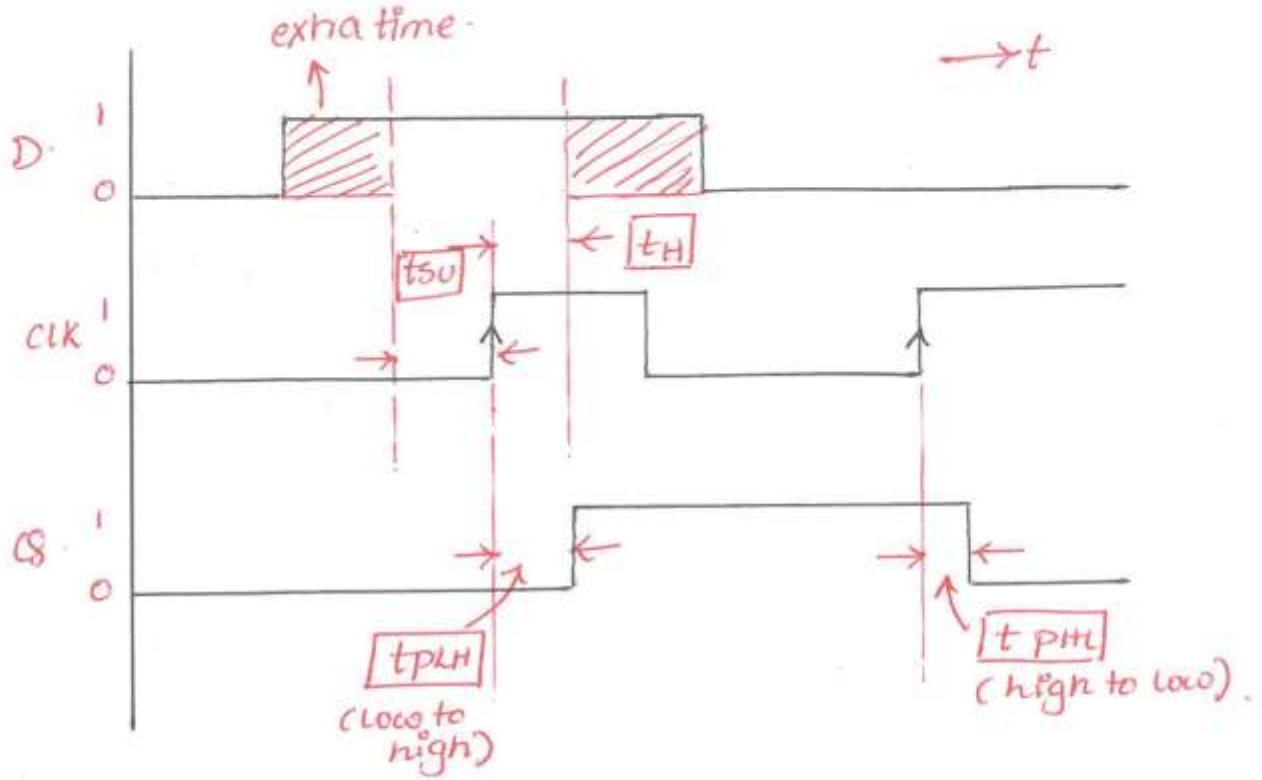
iv> when J=1, K=1 means that the flip flop will toggle. [switch to the opposite state] on the next PT.



(a) Timing Diagram of Edge Triggered (+ve) JK flip flop.

Note: Similarly for -ve flip flop JK Edge Triggered can be asked in the exam.

FLIP FLOP TIMING -



→ This topic deals with the timing issues in the clocked flip flops. I have taken a simple D flip flop (edge trigg) to illustrate this concept.

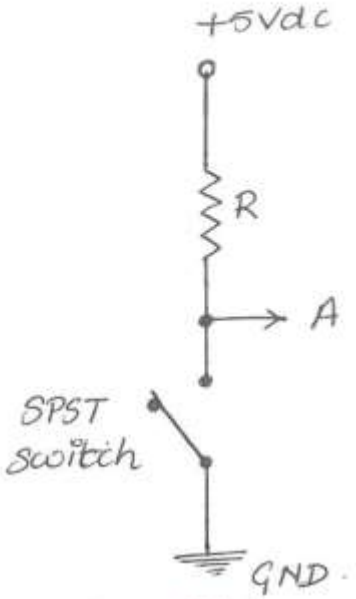
→ t_{PLH} & t_{PHL} are propagation delays through gates.

→ The two other timing considerations are SETUP & HOLD TIME.

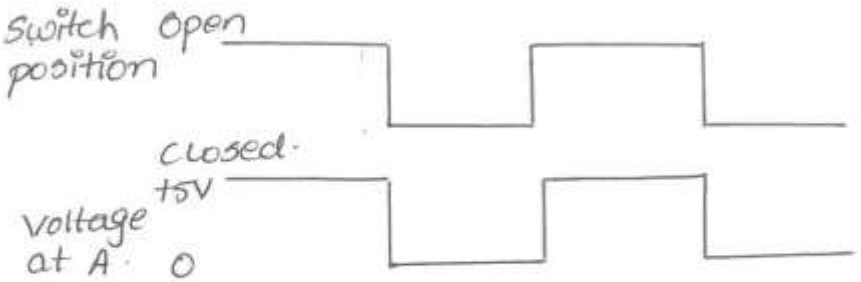
→ Setup time is the "minimum time before the clock transition that the i/p should be stable at the i/p of the ff (t_{SU})"

→ On the same argument, Hold Time is the "minimum time for which the i/p should be stable after the clock transition occurs for the o/p to be guaranteed (t_H)".

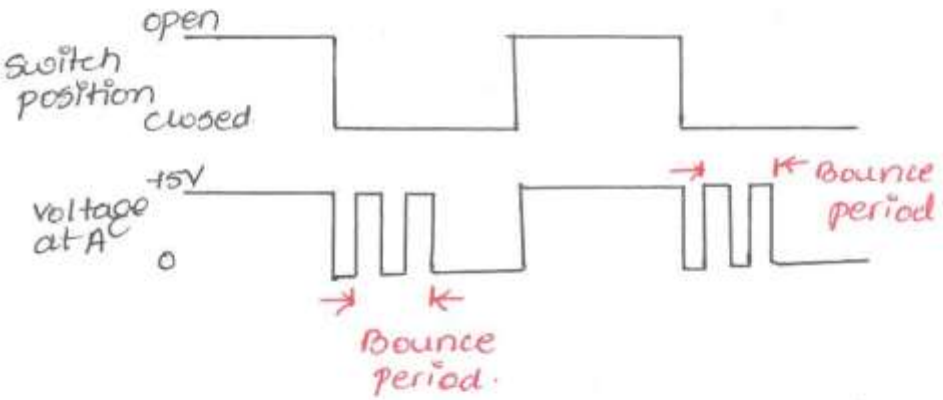
SWITCH CONTACT BOUNCE CIRCUITS -



(a) Switch.



(b) Ideal voltage at A.



(c) Voltage at A showing contact Bounce.

→ A common problem involving switches is the occurrence of CONTACT BOUNCE.

The Single Pole Single Throw (SPST) switch shown in fig (a) is one such example.

Eg: Switches used on the keyboard of a computer system contact bounce may cause a system to respond as though a key was depressed several times in succession.

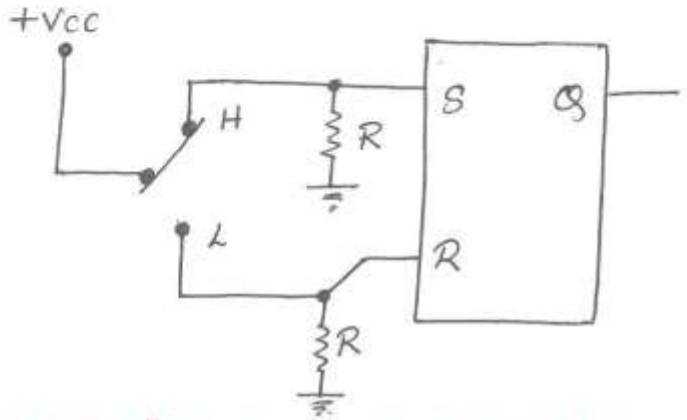
→ When the switch is open, the voltage at point A is +5 Vdc when the switch is closed, the voltage at point A is 0 Vdc. Ideally, the voltage waveform at A should appear as shown in fig (b) as the switch is moved from state 0 to state 1 or vice versa.

→ In actuality, the waveform at point A will appear as shown in fig (c) as a result of the phenomenon known as contact Bounce.

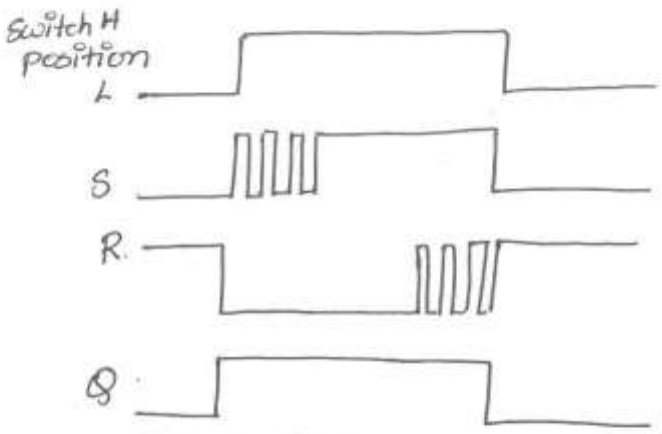
→ As a result, when the arm is moved from one stable state to another, the arm bounces, just like that of a ball which bounces when dropped on a hard surface.

→ Notice that in this particular instance, even though actual physical contact bounce occurs each time the switch is opened (or) closed, contact bounce appears in the voltage level at point 'A' only when the switch is closed.

A Simple RS Latch Debounce Circuit.



(a) Switch contact Bounce Eliminator Circuit



(b) Switch Bounce.

→ The RS latch in the above fig will remove any contact bounce due to the switch. The o/p (Q) is used to generate the desired switch signal.

→ when the switch is moved to position H, R=0 and S=1 Bouncing occurs at the S i/p due to the switch.

→ The flipflop sees this as a series of high & low inputs, settling with a high level.

→ The flipflop will immediately be set with Q=1 at the first high level on 'S' when switch Bounces; losing contact, the i/p signals are R=S=0. ∴ the ff remains set (Q=1).

→ when the switch regains contact, R=0 & S=1 this causes an attempt to again set the ff. But since the ff is already set, no changes occur at Q.

→ The result is that the ff responds to the 1st & only to the 1st high level at its S i/p, resulting in a 'clean' low to high signal at the o/p (Q).

→ when the switch is moved to position L, S=0 & R=1, Bouncing occurs at the R i/p due to the switch. Again the ff sees this as a series of high & low inputs. It simply responds to the 1st high level & ignores the following transitions.

→ The result is a clean high-to-low signal at the ff o/p (Q).

Various Representations of Flip flops -

→ There are various ways a flipflop can be represented, each one suitable for certain application.

i> Characteristic Eqn of Flipflops

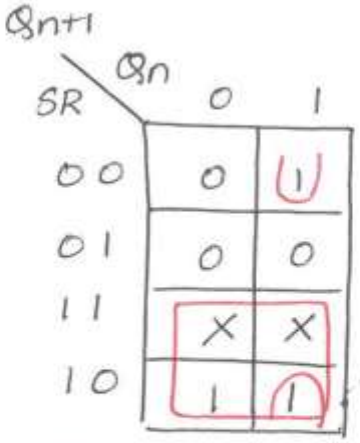
→ The characteristic eqn of FF's are useful in analyzing circuits made of them.

→ Here, next o/p Q_{n+1} is expressed as a function of the present o/p Q_n & I/p to FF's.

→ K-map can be used to get the optimized expression & characteristic table of each FF is mapped into it.

SR Flip flop.

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

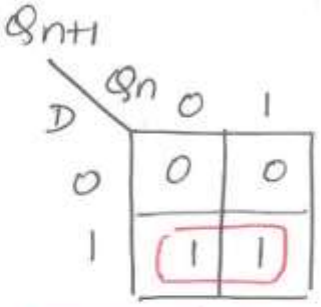


$Q_{n+1} = S + \bar{R}Q_n$

Characteristic Eqn of SR Flip flop.

D Flip flop.

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1



$Q_{n+1} = D$

Characteristic Eqn of D flip flop.

JK flip flop.

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Q_{n+1}

JK \ Q_n	0	1
00	0	1
01	0	0
11	1	0
10	1	1

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

characteristic Eqn of JK flipflop.

T flipflop -

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q_{n+1}

T \ Q_n	0	1
0	0	1
1	1	0

$$Q_{n+1} = T\bar{Q}_n + \bar{T}Q_n$$

characteristic Eqn of T flipflop.

SR - ff : $Q_{n+1} = S + \bar{R}Q_n$

JK - ff : $Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

D - ff : $Q_{n+1} = D$

T - ff : $Q_{n+1} = T\bar{Q}_n + \bar{T}Q_n$

flip flops characteristic Equations.

ii) Flip-flops as Finite State Machine (FSM)

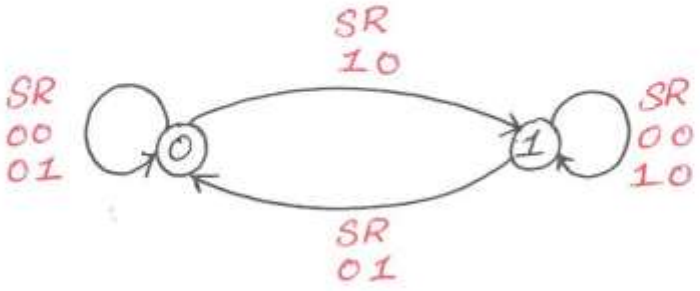
→ In a sequential logic circuit the value of all the memory elements at a given time define the state of that circuit at that time.

→ FSM concept offers a better alternative to the characteristic table in understanding the progress of sequential logic with time.

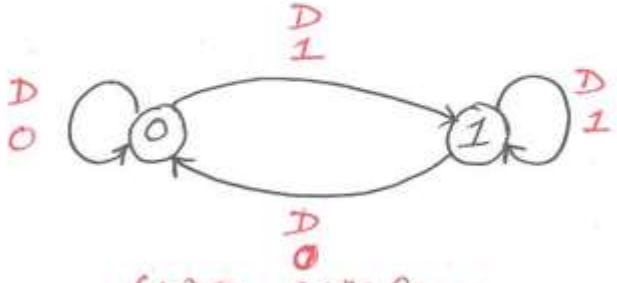
→ In FSM, functional behaviour of a circuit is explained using finite no. of states.

→ State Transition Diagram is a very convenient tool to describe the FSM.

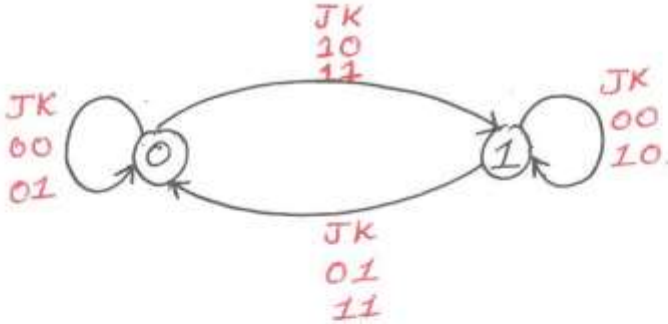
→ In the figure below, all the FF's are represented as FSM through their state transition diagrams.



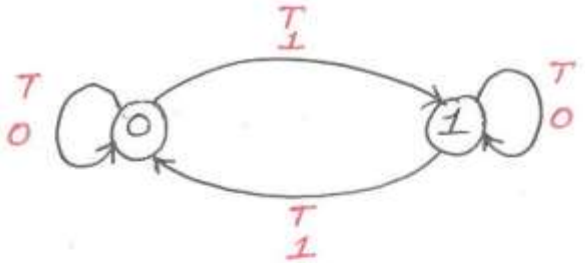
(a) SR flip flop



(b) D - flip flop.



(c) JK flip flop



(d) T - flip flop.

→ Let us see how state transition diagram for SR flip flop is developed from its characteristic table (or) characteristic equation.

→ Each FF can be either 0 (or) 1 state defined by its stored value at any given time.

→ Application of i/p may change the stored value, ie; the state of the ff. This is shown by the directional arrows & the corresponding i/p is written along side.

→ If SR ff stores 0, then for SR = 00 or 01 the stored value does not change. for SR = 10, ff o/p changes to 1. Note that SR = 11 is not allowed.

→ when SR ff stores 1, application of SR = 00 (or) 10 does not change its value & only when SR = 01, o/p changes to 0.

→ The state transition diagrams are developed in a similar way for D, JK & T flipflops.

iii> Flip-flop Excitation Table.

→ In synthesis (or design problem excitation tables are very useful & its importance is analogous to that of the characteristic table in analysis problem.

→ EXCITATION TABLE of a ff is looking at its TRUTH TABLE in a reverse way.

→ Here ff i/p is presented as a dependent function of transition $Q_n \rightarrow Q_{n+1}$

→ This is derived from the ff characteristic table (or) characteristic eqn, but more directly from its State Transition Diagram.

→ The table below gives a summary presentation of the excitation tables of all flipflops.

$Q_n \rightarrow Q_{n+1}$	S	R	J	K	D	T
0 0	0	x	0	x	0	0
0 1	1	0	1	x	1	1
1 0	0	1	x	1	0	1
1 1	x	0	x	0	1	0

Excitation Table of all the flipflops.

HDL Implementation of FLIP FLOPS -

i) Write the HDL code for D flipflop.

```
module DFF (D, EN, Q);
input D, EN;
output Q;
reg Q;
always @ (EN or D)
if (EN) Q = D;
endmodule.
```

ii) Write the HDL code for SR flipflop.

```
module SRFF (S, R, EN, Q);
input S, R, EN;
output Q;
reg Q;
always @ (EN or S or R)
if (EN) Q = S | (~R & Q);
endmodule.
```

iii) Write the HDL code for positive edge triggered D flipflop.

```
module DFFpos (D, C, Q);
input D, C;
output Q;
reg Q;
always @ (posedge C)
    Q = D;
endmodule
```

iv> write the HDL code for Negative Edge Triggered D flip flop.

```
module Dffneg (D, C, Q);
input D, C;
output Q;
reg Q;
always @ (negedge c)
        Q=D;
endmodule
```

v> write the HDL code for Positive Edge Triggered SR flip flop.

```
module SRFFpos (S, R, C, Q);
input S, R, C;
output Q;
reg Q;
always @(posedge c)
        Q = S | (~R & Q);
endmodule
```