

**K.T.S.P.Mandal's**

**Hutatma Rajguru Mahavidyalaya, Rajgurunagar**

**Tal-Khed, Dist.-Pune 410505.**

**F.Y.BSc (Computer Science)**

**Semester-II**

**Electronics Paper-II**

**Subject- ELC-122 – Basics of Computer Organization**

**According to new CBCS syllabus w.e.f.2019-2020**

**Prof.Aparna P.Kulkarni**

**Department of Computer Science**

**Hutatma Rajguru Mahavidyalaya, Rajgurunagar.**

**Unit-I- Flip-Flops**

- RS Flip-Flop using NAND Gate.
- Clocked RS Flip-Flop.
- D-Latch, J-K flip-Flop.
- T Flip-Flop.

## Introduction:

The digital systems are generally classified into systems:

1. **Combinational Circuits:** Combinational Circuit is the type of circuit in which output is independent of time and only relies on the input present at that particular instant.  
**Example:** Encoder, Decoder, Multiplexer, Demultiplexer etc.
2. **Sequential Circuits:** Sequential circuit is the type of circuit where output not only relies on the current input but also depends on the previous output.  
**Example:** Flip-Flops, registers, counters etc.

## Flip-Flop:

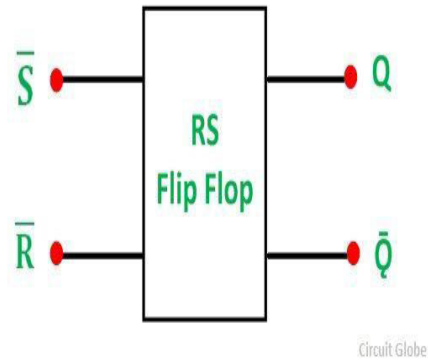
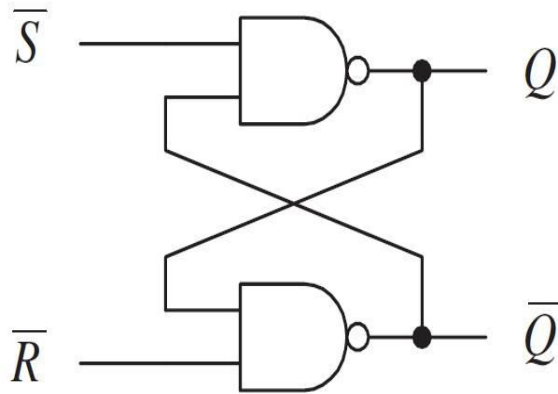
A flip flop is a sequential electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops are edge triggered or edge sensitive.

Flip-flop can store 1-bit of digital information. It is also referred as a 1-bit register. A register contains group of flip-flop, the number of flip-flop in a register being equal to the number of bits present in the data.

## RS Flip-Flop using NAND Gate:

An important point about NAND gate is that its dominating input is 0 i.e., if any of its input is Logic '0', the output is Logic '1', irrespective of the other input. The output is 0, only if all the inputs are 1.

The circuit of RS flips – flop using NAND gates is shown in below figure:



**Case 1: When  $S = 0, R = 0,$**

Both the gates outputs go high. But this cannot be allowed, as in a flip-flop one output ( $Q$ ) will always have to be complement of the other ( $Q$ ).

So  $S=0, R = 0$ ) is not allowed.

**Case 2: When  $S = 0, R = 1,$**

The output of gate 1 is 1. This is given as input to gate 2. Thus gate 2 gets both its inputs as 1. Its output goes 0, which acts as input to gate 1. Thus the output of gate 1 remains 1.

So we find that when  $S = 0, R = 1, Q = 1, Q = 0$ .

**Case 3: When  $S = 1, R = 0,$**

Output of gate 2 is 1. This is given as input to gate 1. So gate 1 receives both inputs as 1 and its output goes to 0.

Thus  $S = 1, R = 0, Q = 0, Q = 1$ .

**Case 4: When  $S = 1, R = 1$**

We cannot decide the outputs unless we know the previous state or last state. Let  $Q=1, Q = 0$ .

We find that  $Q$  acts as input to gate 1. So inputs to gate 1 are 1, 0 and its output goes to 1.

However  $Q$  acts as input to gate 2. Gate 2 receives both inputs as 1, 1 and its output become 0.

So  $S = 1, R = 1, Q = 1, Q = 0$  which is nothing but the previous state.

The working can thus be illustrated by a truth table.

**Truth Table:**

R	S	Q	$\bar{Q}$
0	0	Not allowed	
0	1	1	0
1	0	0	1
1	1	Previous state	

RS flip-flop is called the set reset flip-flop because when Set is activated (S = 0, R = 1), output Q=1 (set). When Reset is activated (S = 1, R = 0), output Q = 0) (reset).

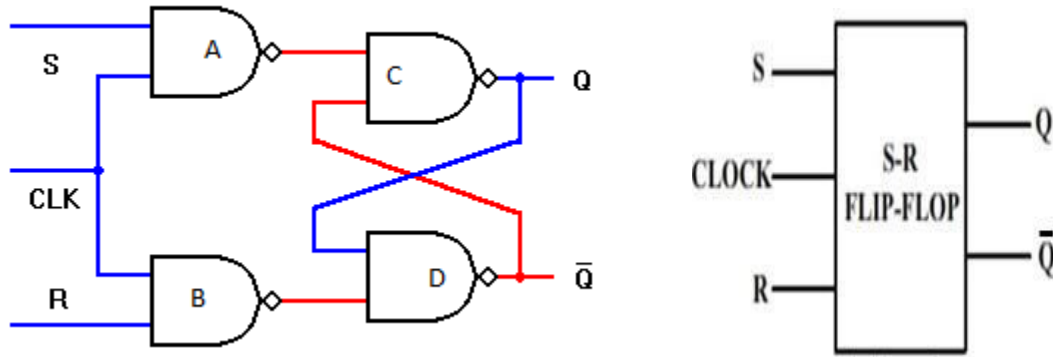
**Clocked RS Flip-Flop:**

The flipflop discussed in the previous section has a problem. There is no restriction on how fast the inputs can change. So it is possible that the inputs may vary faster than the average propagation delay of the flipflop.

So before a particular input can cause some change in the output, the input changes again. We will not get a satisfactory behaviour from the flip-flop. So the flip-flop circuit is modified to include a clock.

The behaviour of a circuit must be defined by knowing its signals at discrete intervals of time. The outputs of the flip-flop must change only at discrete instants of time and so must its input. When the inputs are being accepted the previous output should not change and when the output is changing no new inputs must be accepted. This synchronization is possible with a clock. A clock is a periodic train of 1s and 0s. Generally, we can conclude that by increasing clock speed (its frequency), the circuits can be made to work faster.

However all the time  $T_{Clock} > T_{propagation\ time}$ .



The inputs will be accepted only when clock goes high. When clock goes low, we get a stable corresponding to the applied inputs.

**Case I: Clock = 1, S = 0, R=0.**

Here output of gates A and B are 1. So we cannot work out further unless we know the previous state. Let the previous state be  $Q = 0$  and  $\bar{Q} = 1$ . Now inputs to gate C are 1,  $\bar{Q}$ , i.e., 1. So output of gate C ( $Q$ ) = 0. This is fed to gate D. Inputs to gate D are 1,  $Q = 0$ . So output of gate D = 1. We get  $Q = 0, \bar{Q} = 1$  which is the previous state.

**Case II: Clock = 1, S = 1, R = 0**

Here output of gate A = 0 and that of gate B = 1. Gate C receives one of the inputs as 0. So its output is 1 irrespective of other input. This output is fed as input to gate D. Inputs to gate D are 1, 1. Thus output of gate D ( $Q$ ) = 0 which is fed as input to gate C. The output of gate C ( $\bar{Q}$ ) is thus maintained at 1.

We get  $Q = 1, \bar{Q} = 0$

**Case III: Clock = 1, S = 0, R = 1**

Here output of gate A = 1 and that of gate B = 0. The gate D receives one input as 0. So its output ( $\bar{Q}$ ) goes high. This is fed to gate C. Inputs to gate C are 1, 1. The output of gate C ( $\bar{Q}$ ) = 0. This is fed back as input to gate D keeping the output of gate D ( $Q$ ) as 1 therefore we get  $Q = 0, \bar{Q} = 1$ .

**Case IV: Clock = 1, S= 1, R= 1**

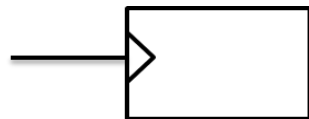
Here output of gates A and B are 0. So both gates C and D receive one input as 0. Hence their outputs go to 1. This means Q and  $\bar{Q} = 1$  which is not possible. Q and  $\bar{Q}$  should be complementary. Thus, this condition is not allowed.

The working can be summarized by a truth table.

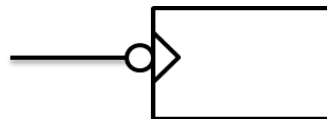
Clock	S	R	Output	
			Q	$\bar{Q}$
1	0	0	Previous state	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not allowed	

**Edge and Level Triggered Flip-flop:**

If a flip-flop responds to changing edges of a clock, it is called edge-triggered flip-flop. If the output changes at the rising edge of a clock, the flip-flop is called positive edge triggered. If the output changes at the falling edge of the clock, the flip flop is called negative edge triggered flip-flop.

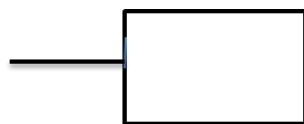


Positive edge triggered

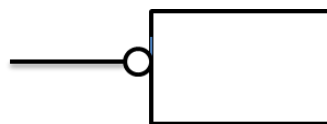


Negative edge triggered

If the flip-flop responds only to a particular state of the clock, i.e. 1 or 0 and not the clock transition, then it is called level triggered flip-flops.



Positive level triggered



Negative level triggered

### Preset/Set and Clear/Reset:

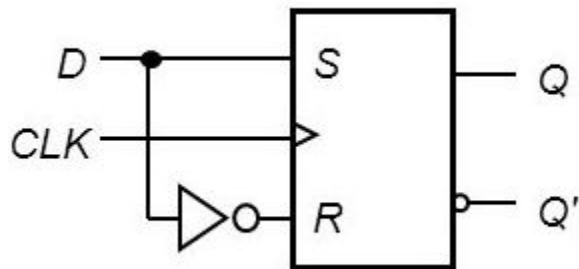
These are asynchronous inputs to a flip-flop. They may or may not be present. If present, they cause the output of the flip-flop (Q) to change, irrespective of the clock. They are clock independent.

They can be active high or active low. Whenever present or set input is active, the output is forced to become 1, irrespective of the clock and the other flip-flop inputs. Whenever reset or clear input is active, the output is forced to become 0, irrespective of the clock and the other flip-flop inputs.

At a time only preset and clear or set and reset may be present in a flip-flop.

### D Flip-Flop (D Latch):

The D Flip Flop is by far the most important of the clocked flip-flops as it ensures that ensures that inputs S and R are never equal to one at the same time. The D-type flip flops are constructed from a gated SR flip-flop and JK flip-flop with an inverter added between the S and the R inputs as well as J and K inputs to allow for a single D (Data) input.



D	Q
0	0
1	1

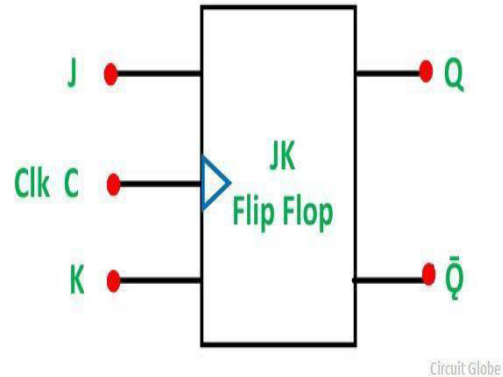
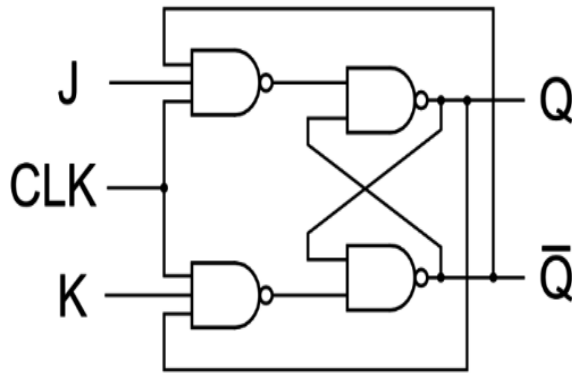
D flip-flop is also called as delay flip-flop. D flip-flop is used as a 1 bit latch or memory. For storing several bits, many D flip-flops are cascaded to form a register.

### JK Flip-Flop:

RS flip-flops suffer from one important drawback. Suppose R and S inputs are provided from some logic circuit. What would happen when the inputs become 1, 1? Obviously this condition is

not allowed because we will not be sure whether output is 1 or 0. The flip-flop begins to function randomly.

To eliminate this condition and to make the flip-flops function even if inputs are 1, 1, the RS flip-flop is modified to get the JK flip-flop.



**Case I:  $J=0, K=0, Q=0, \bar{Q}=1, \text{Clock}=1$**

Output of A = 1, output of gate B = 1. Inputs to gate C are 1,  $\bar{Q}$ , i.e., 1. So output  $\bar{Q} = 0$ . Inputs to gate D are 1,  $Q = 1, 0$ . So output  $Q = 1$ . We finally get  $\bar{Q} = 0, Q = 1$  which is the previous state; which is same as that of SR flip-flop.

**Case II:  $J=1, K=0, Q=0, \bar{Q}=1, \text{Clock}=1$**

Input to gate A = J,  $\bar{Q}$ , Clock = 1, 1, 1. So its output is 0. Input to gate B are K,  $\bar{Q}$ , Clock = 0, 0, 1. So its output is 1.

Input to gate C = 0,  $\bar{Q}$ , i.e., 1. So its output  $Q = 1$ . Input to gate D = 1,  $\bar{Q}$ , i.e., 1, 1 ( $\bar{Q}$  has become 1) so its output ( $\bar{Q}$ ) = 0. We get  $Q = 1, \bar{Q} = 0$ . This is same as the set state in a SR flip-flop.

**Case III:  $J=0, K=1, Q=1, \bar{Q}=0, \text{Clock}=1$**

Inputs to gate A = J,  $\bar{Q}$ , Clock = 0, 0, 1. So its output is 1. Inputs to gate B = K,  $\bar{Q}$ , Clock = 1, 1, 1. So its output is 0.

Now inputs to gate D = 0,  $\bar{Q}$ , i.e., 0, 1. So its output ( $\bar{Q}$ ) = 1. Inputs to gate C = 1,  $\bar{Q}$ , i.e., 1, 1 ( $\bar{Q}$  has changed to  $\bar{1}$ ). Its output ( $Q$ ) = 0. So we get  $Q = 0, \bar{Q} = 1$  which is same as Reset state of S R flip-flop.

**Case IV:  $J=1, K=1, Q=0, \bar{Q}=1, \text{Clock}=1$**



Inputs to gate A = J, Q, Clock = 1, 1, 1. So its output is 0. Inputs to gate B are K, Q, Clock = 1, 0, 1. So its output is 1.

Inputs to gate C = 0, Q, i.e., 0, 1. So its output is 1. Inputs to gate D = 1, Q, i.e., 1, 1 (Q has become 1). So its output Q = 0.

We get Q = 1,  $\overline{Q} = 0$ , i.e., the outputs have got complemented. This condition is called toggling.

In the next clock pulse again Q = 0,  $\overline{Q} = 1$ .

Thus we can summarize the working of a JK flip-flop with the truth table.

Clock	J	K	Output	
			Q	$\overline{Q}$
1	0	0	Previous state	
1	0	1	0	1
1	1	0	1	0
1	1	1	toggling	

The JK flip-flop suffers from the problem of racing of outputs (Race around condition). It can be explained as follows. Let J = K = 1, Q = 0 and a clock input applied. After time  $t_p$  (propagation delay) the output Q = 1 (Toggles). After another time  $t_p$ , J and K are still 1. So Q = 0. This way the output keeps on racing from 1 to 0 and at the end of the clock pulse the state of the output is not clear (whether 1 or 0). This is the race around condition. Here we have assumed that the positive clock period  $T_x \gg t_p$ . The race around condition mainly exists because of the feedback of the outputs Q and  $\overline{Q}$  to gates B and A.

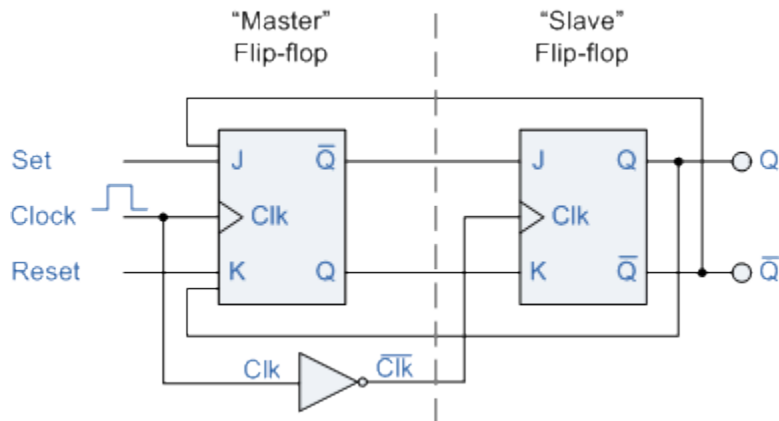
The race around condition can be avoided if  $T_x < t_p$ . However  $t_p$ , is very small. It can also be avoided by making the flip flop edge triggered, rather than level triggered. It can also be avoided by using a master-slave arrangement.

### **JK Master Slave Flip-Flop:**

The problem of racing around is also avoided using a JK master slave flip-flop. The Master-Slave Flip-Flop is basically a combination of two JK flip-flops connected together in a series configuration. Out of these, one acts as the “**master**” and the other as a “**slave**”. While the master is accepting the new inputs, slave is busy maintaining previous outputs as it is. When the master

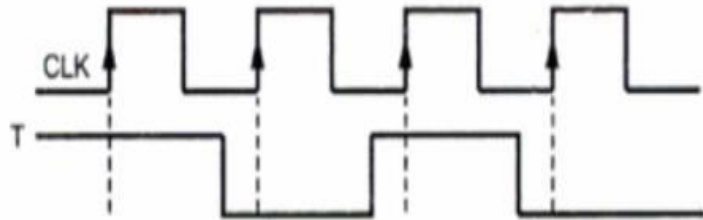
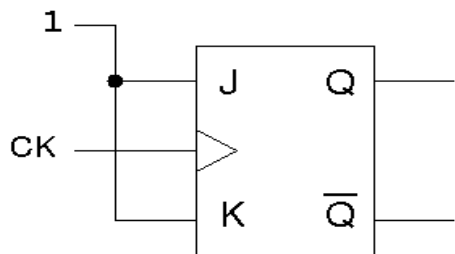
has completed accepting the inputs, the slave will allow the changes caused by the master to reach the output. During this time no new inputs are accepted by the master.

When clock pulse = 1, master accepts new input and causes changes. When clock pulse = 0 slave allows the changes to come at the output but master doesn't accept any new input.



### T Flip-Flop:

T flip-flop is the toggle flip-flop. It causes the flip-flop output to toggle. This is special case of JK flip-flop where  $J=K=1$ . RS flip-flop cannot work as a T flip-flop because 1, 1 condition is not allowed there. T flip flop can be used to divide the input frequency by 2.



$$T_{out} = 2 T_{clock}$$

### Applications of Flip-Flop:

These are the various types of flip-flops being used in digital electronic circuits and the applications of Flip-flops are as specified below.

- Counters

- Frequency Dividers
- Shift Registers
- Storage Registers
- Bounce elimination switch
- Data transfer
- Latch
- Memory

### References:

1. A text book of Basics of Computer Organization of vision publication by  
H.R.Arvind.
2. [www.google.com](http://www.google.com)