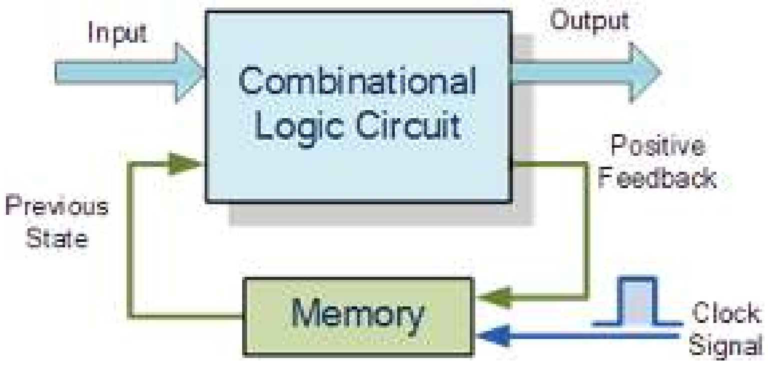
SEQUENTIAL CIRCUITS



The word “Sequential” means that things happen in a “sequence”, one after another and in **Sequential Logic** circuits, the actual clock signal determines when things will happen next. Simple sequential logic circuits can be constructed from standard **Bistable** circuits such as: *Flip-flops*, *Latches* and *Counters* and which themselves can be made by simply connecting together universal NAND Gates and/or NOR Gates in a particular combinational way to produce the required sequential circuit.

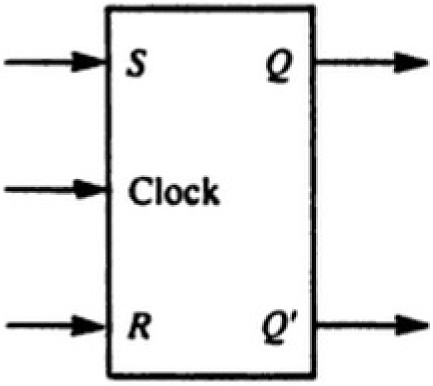
FLIP-FLOPS:

A flip flop is an 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 and latches are used as data storage elements. It is the basic storage element in sequential logic. But first, let’s clarify the difference between a latch and a flip-flop.

A flip has two outputs one for normal value and one for its compliment value of bit stored in it. i.e either one or zero.

That’s why flip flops are termed as bistable multivibrator.

Flip flop v/s Latch



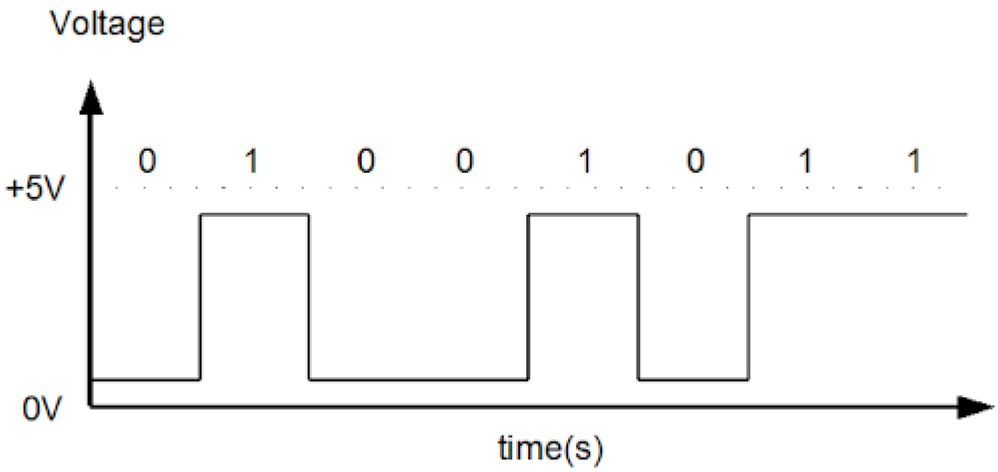
graphic symbol(logic symbol)

The basic difference between a latch and a flip-flop is a gating or clocking mechanism.

There are two types of memory elements based on the type of triggering that is suitable to operate it.

- Latches
- Flip-flops

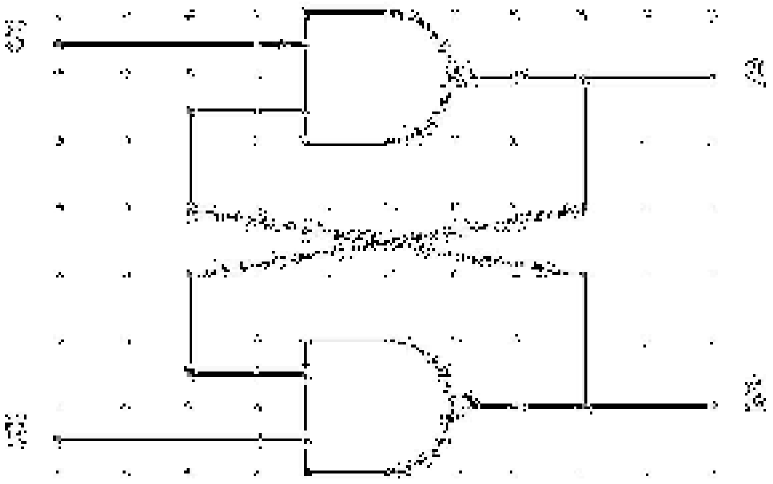
Latches operate with enable signal, which is **level sensitive**. Whereas, flip-flops are edge sensitive.



Read the full comparison of Flip Flop v/s latch here

For example, let us talk about SR latch and SR flip-flops. In this circuit when you Set S as active the output Q would be high and Q' will be low. This is irrespective of anything else. (This is an active-low circuit so active here means low, but for an active high circuit active would mean high)

((**properties of NAND gate is either input is zero output should be one))**



SR Latch(Basic flip-flop)

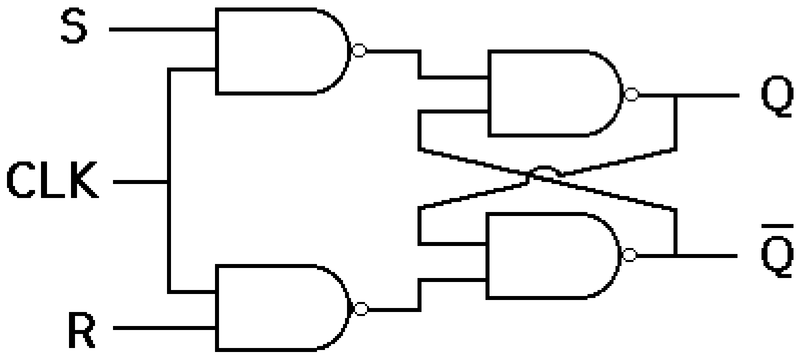
A flip flop, on the other hand, is synchronous and is also known as gated or clocked SR latch.

SR Flip Flop using NAND configuration

There are majorly 4 types of flip flops, with the most common one being SR flip flop. This simple flip flop circuit has a set input (S) and a reset input (R). In this circuit when you Set “S” as active the output “Q” would be high and “Q‘” will be low. Once the outputs are established, the wiring of the circuit is maintained until “S” or “R” go high, or power is turned off. As shown above, it is the simplest and easiest to understand. The two outputs, as shown above, are the inverse of each other. The truth table of SR Flip Flop is highlighted below.

Circuit diagram:

properties of NAND gate is either input is zero output should be one))



SR Flip-Flop

In all conditions CLK =1

Eg: S=1 R=0   top gate 0 bottom gate 1   Q=1, Q’=0 -> set state(stored value=1)
R=1 S=0   top gate 1   “   “   0   Q=0   Q’=1 -> Reset state(“ “   =0)
S=0 R=0   “   “   1 “   “   1   Q=0   Q’=1   -> Reset   (pervious state)
S=1 R=0   top gate 0 bottom gate 1   Q=1, Q’=0 -> set state(stored value=1)

S=0 R=0   “   “   1 “   “   1   Q=1, Q’=0   -> set(previous state)
S=1 R=1 “   “   0   “   “   0   Q= 1, Q’=1-> (invalid condition)

Truth Table:

S	**R**	**Q**	**Q’**
0	0	0	1
0	1	0	1
1	0	1	0
1	1	∞invalid state	∞

In this circuit diagram, the output is changed (i.e. the stored data is changed) only when you give an active clock signal. Otherwise, even if the S or R is active the data will not change. Let’s look at the types of flip-flops to understand better.

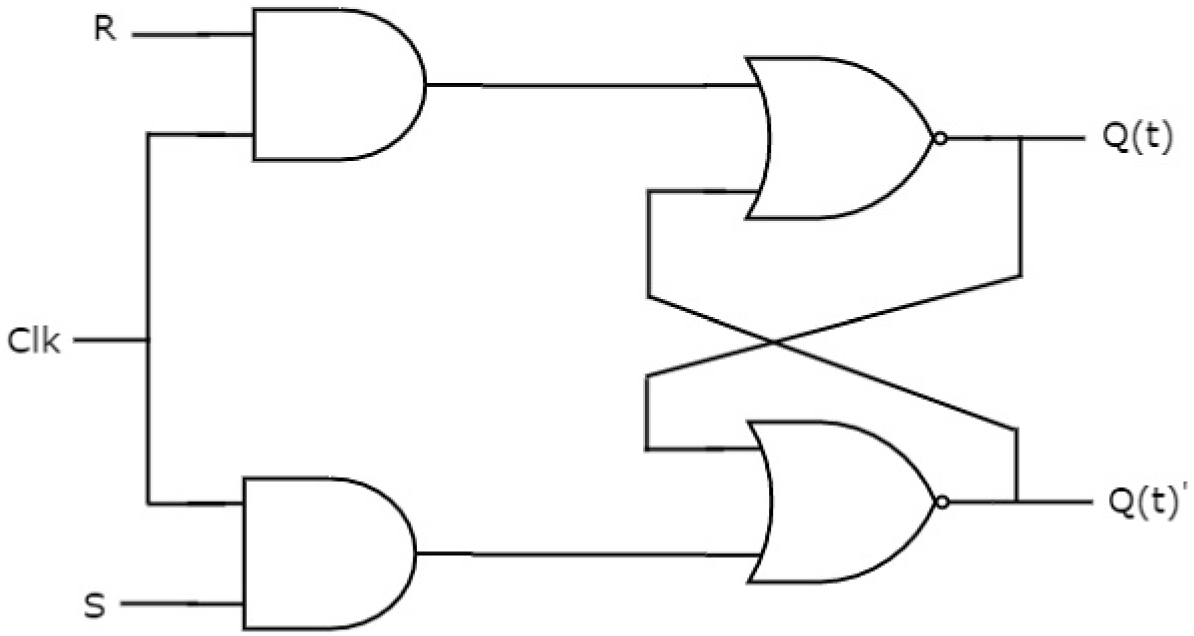
Truth Table:

n all conditions clock is one if clock pulse goes to zero the latch maintain the previous state.

SR Flip-Flop using NOR configuration

SR flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, SR latch operates with enable signal. The **circuit diagram** of SR flip-flop is shown in the following figure.

NOR gate: Either input is one output should be zero



This circuit has two inputs S & R and two outputs Qt & Qt’. The operation of SR flipflop is similar to SR Latch. But, this flip-flop affects the outputs only when positive transition of the clock signal is applied instead of active enable.

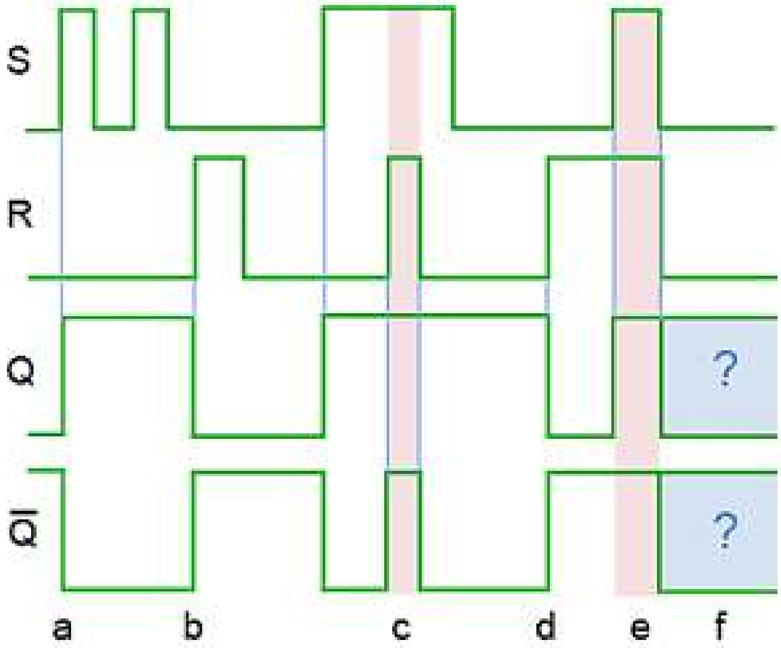
The following table shows the **state table** of SR flip-flop.

in all conditions clock is one if clock pulse goes to zero the latch maintain the previous state.

S	R	Q(t)
0	0	No change
0	1	0
1	0	1
1	1	Invalid condition-

So, SR flip-flop can be used for one of these three functions such as Hold, Reset & Set based on the input conditions, when positive transition of clock signal is applied.

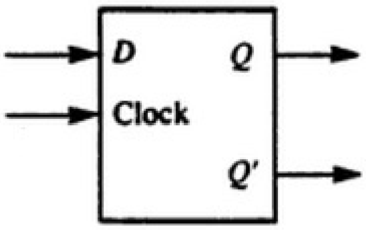
Timing diagram:

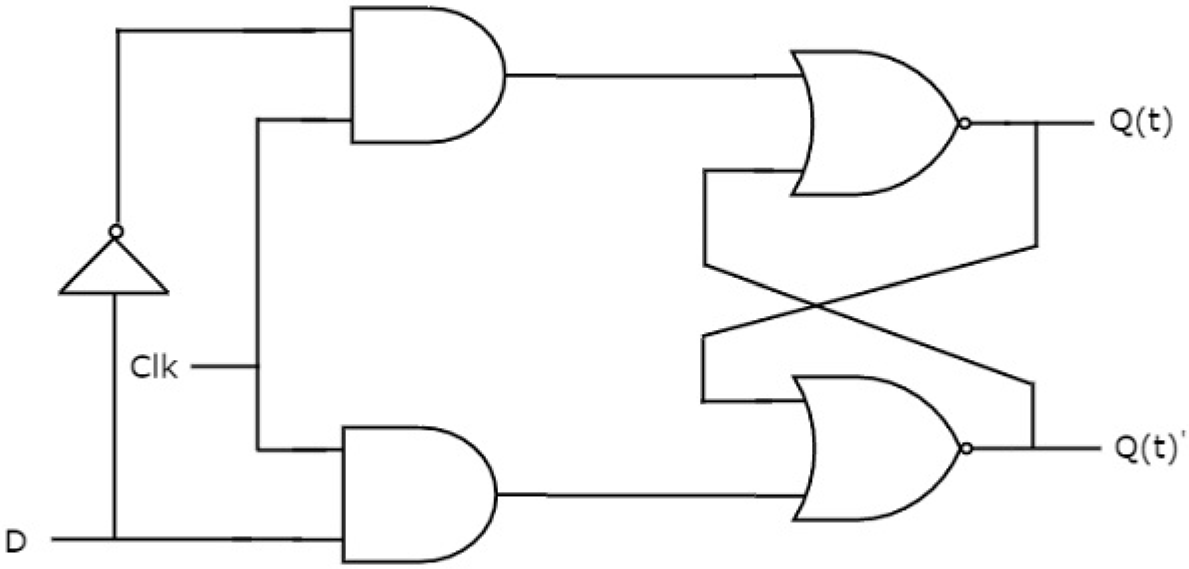


D Flip-Flop(D->Data)

D flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal. That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal. The **circuit diagram** of D flip-flop is shown in the following figure.

graphic symbol





This circuit has single input D and two outputs Qtt & Qtt'. The operation of D flip-flop is similar to D Latch. But, this flip-flop affects the outputs only when positive transition of the clock signal is applied instead of active enable.

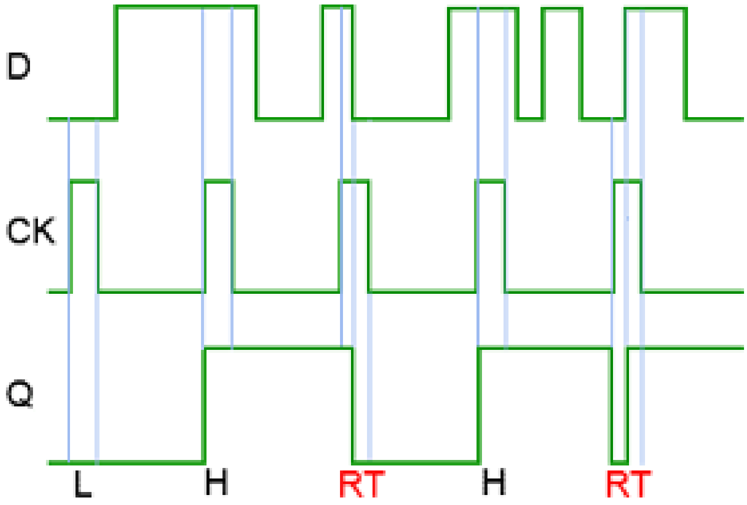
The following table shows the **state table** of D flip-flop.

D	Qt
0	0
1	1

Therefore, D flip-flop always Hold the information, which is available on data input, D of earlier positive transition of clock signal. From the above state table, we can directly write the next state equation as

$$Qt = D$$

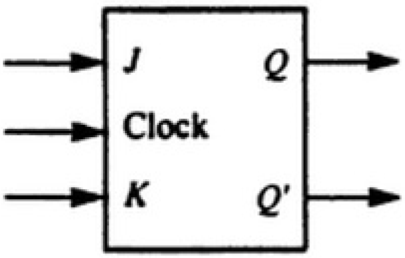
Next state of D flip-flop is always equal to data input, D for every positive transition of the clock signal. Hence, D flip-flops can be used in registers, **shift registers** and some of the counters.



Drawback of RS flip-flop:

if S=1 and R=1 invalid state or forbidden condition.(forbidden->not allowed)

The JK Flip Flop:



The basic S-R NAND flip-flop circuit has many advantages and uses in sequential logic circuits but it suffers from two basic switching problems.

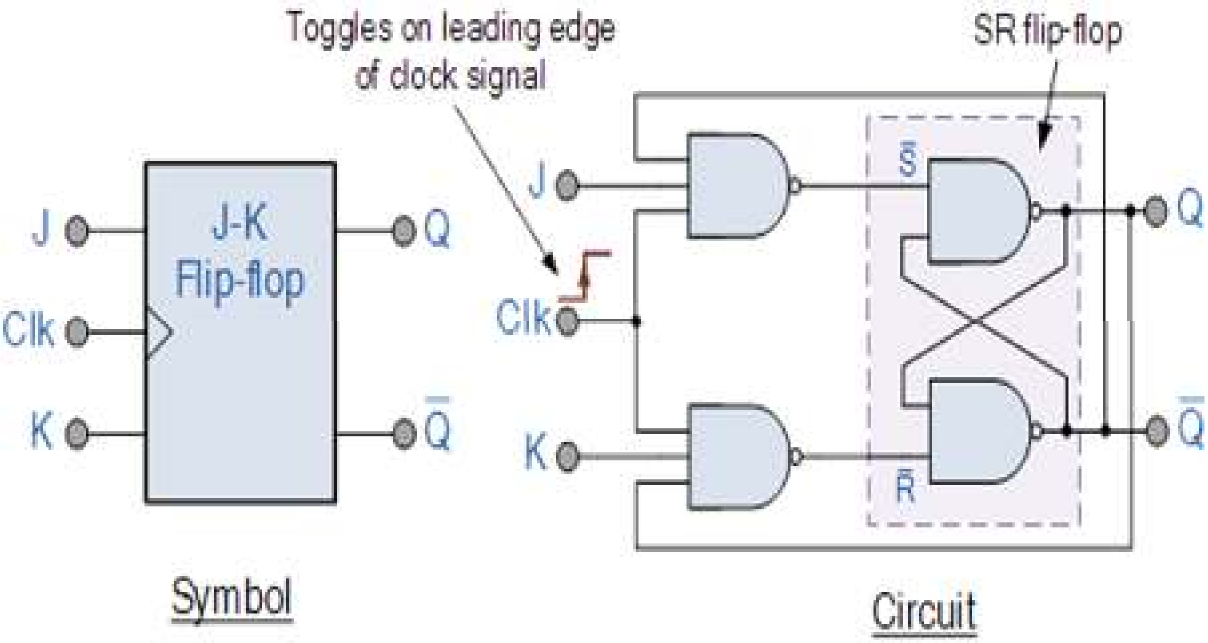
- 1. the Set = 1 and Reset = 1 condition (S = R = 1) must always be avoided

Then to overcome these two fundamental design problems with the SR flip-flop design, the **JK flip Flop** was developed.

This simple **JK flip Flop** is the most widely used of all the flip-flop designs and is considered to be a universal flip-flop circuit. The two inputs labelled “J” and “K” are not shortened abbreviated letters of other words, such as “S” for Set and “R” for Reset, but are themselves autonomous letters chosen by its inventor Jack Kilby to distinguish the flip-flop design from other types.

The sequential operation of the JK flip flop is exactly the same as for the previous SR flip-flop with the same “Set” and “Reset” inputs. The difference this time is that the “JK flip flop” has no invalid or forbidden input states of the SR Latch even when S and R are both at logic “1”.

The **JK flip flop** is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level “1”. Due to this additional clocked input, a JK flip-flop has four possible input combinations, “logic 1”, “logic 0”, “no change” and “toggle”. The symbol for a JK flip flop is similar to that of an *SR Bistable Latch* as seen in the previous tutorial except for the addition of a clock input.



Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs, respectively after its inventor Jack Kilby. Then this equates to: J = S and K = R.

The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q. This cross coupling of the SR flip-flop allows the previously invalid condition of S = “1” and R = “1” state to be used to produce a “toggle action” as the two inputs are now interlocked.

If the circuit is now “SET” the J input is inhibited by the “0” status of Q through the lower NAND gate. If the circuit is “RESET” the K input is inhibited by the “0” status of Q through the upper NAND gate.
As Q and Q are always different we can use them to control the input. When both inputs J and K are equal to logic “1”, the JK flip flop toggles as shown in the following truth table.

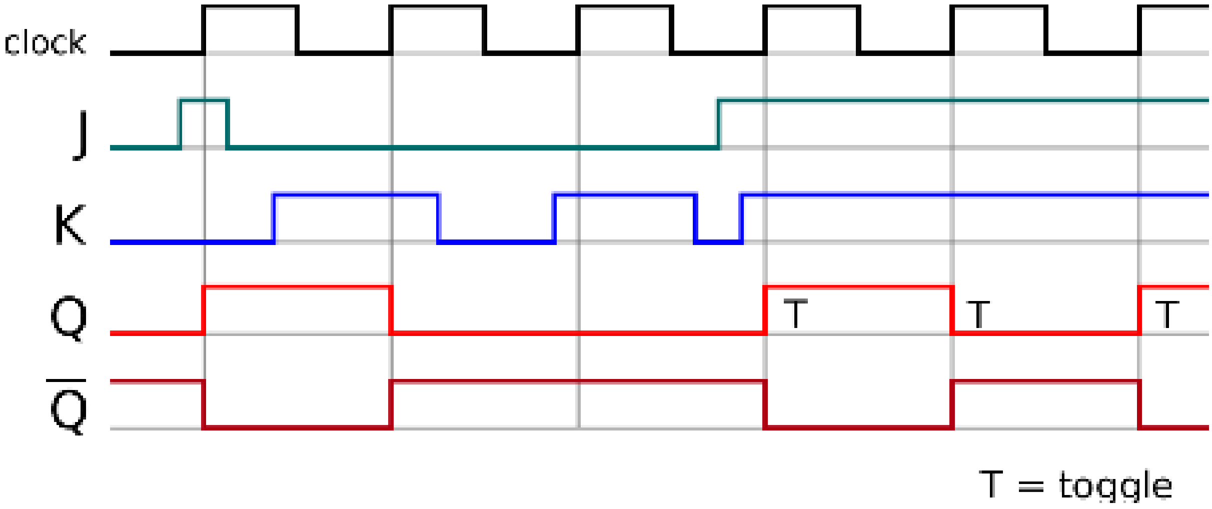
The Truth Table for the JK Function

	Clock	Input		Output		Description
	Clk	J	K	Q	Q	
same as for the SR Latch	X	0	0	1	0	Memory no change
	X	0	0	0	1	
	¯↓_	0	1	1	0	Reset Q » 0
	X	0	1	0	1	
	¯↓_	1	0	0	1	Set Q » 1
	X	1	0	1	0	
toggle action	¯↓_	1	1	0	1	Toggle
	¯↓_	1	1	1	0	

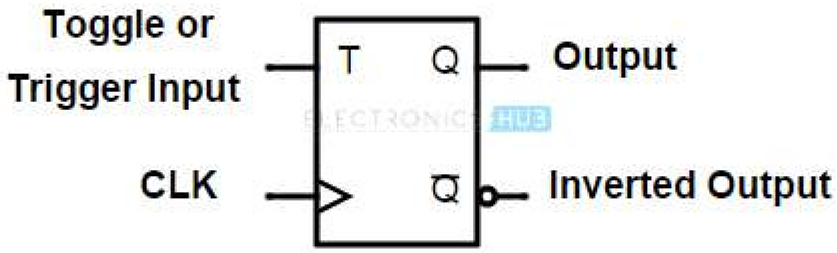
Then the JK flip-flop is basically an SR flip flop with feedback which enables only one of its two input terminals, either SET or RESET to be active at any one time under normal switching thereby eliminating the invalid condition seen previously in the SR flip flop circuit.

However, if both the J and K inputs are HIGH at logic “1” (J = K = 1), when the clock input goes HIGH, the circuit will “toggle” as its outputs switch and change state complementing each other. This results in the JK flip-flop acting more like a T-type toggle flip-flop when both terminals are “HIGH”. However, as the outputs are fed back to the inputs, this can cause the output at Q to oscillate between SET and RESET continuously after being complemented once.

While this JK flip-flop circuit is an improvement on the clocked SR flip-flop it also suffers from timing problems called “race” if the output Q changes state before the timing pulse of the clock input has time to go “OFF”. To avoid this the timing pulse period (T) must be kept as short as possible (high frequency). As this is sometimes not possible with basic JK flip-flops built using basic NAND or NOR gates, far more advanced master-slave (edge-triggered) flip-flops were developed which are more stable.

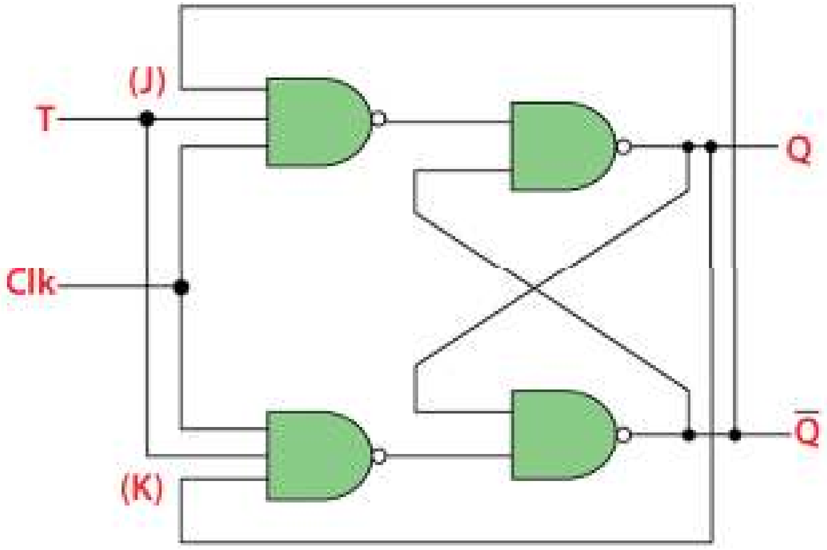


T-FLIP-FLOP



Block diagram of the "T-Flip Flop" is given where T defines the "Toggle input", and CLK defines the clock signal input.

We can construct the "T Flip Flop" by making changes in the "JK Flip Flop". The "T Flip Flop" has only one input, which is constructed by connecting the input of JK flip flop. This single input is called T. In simple words, we can construct the "T Flip Flop" by converting a "JK Flip Flop". Sometimes the "T Flip Flop" is referred to as single input "JK Flip Flop".



Truth Table of T Flip Flop

	Previous		Next	
T	Q	Q’	Q	Q’
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

The upper NAND gate is enabled, and the lower NAND gate is disabled when the output Q To is set to 0. make the flip flop in "set state(Q=1)", the trigger passes the S input in the flip flop.

The upper NAND gate is disabled, and the lower NAND gate is enabled when the output Q is set to 1. The trigger passes the R input in the flip flop to make the flip flop in the reset state(Q=0).

Operations of T-Flip Flop

The next sate of the T flip flop is similar to the current state when the T input is set to false or 0.

- If toggle input is set to 0 and the present state is also 0, the next state will be 0.
- If toggle input is set to 0 and the present state is 1, the next state will be 1.

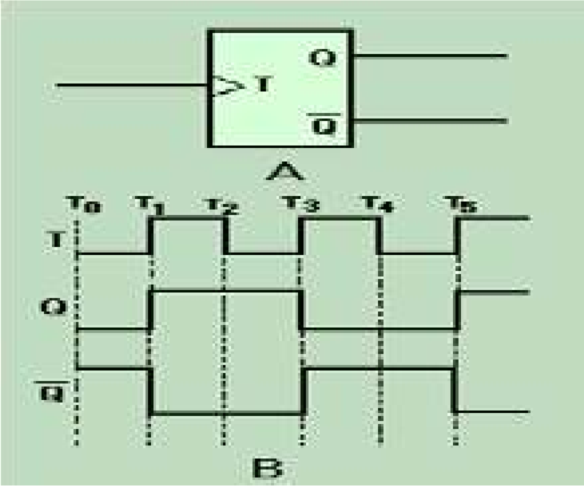
The next state of the flip flop is opposite to the current state when the toggle input is set to 1.

- If toggle input is set to 1 and the present state is 0, the next state will be 1.
- If toggle input is set to 1 and the present state is 1, the next state will be 0.

The "T Flip Flop" is toggled when the set and reset inputs alternatively changed by the incoming trigger. The "T Flip Flop" requires two triggers to complete a full cycle of the output waveform. The frequency of the output produced by the "T Flip Flop" is half of the input frequency. The "T Flip Flop" works as the "Frequency Divider Circuit."

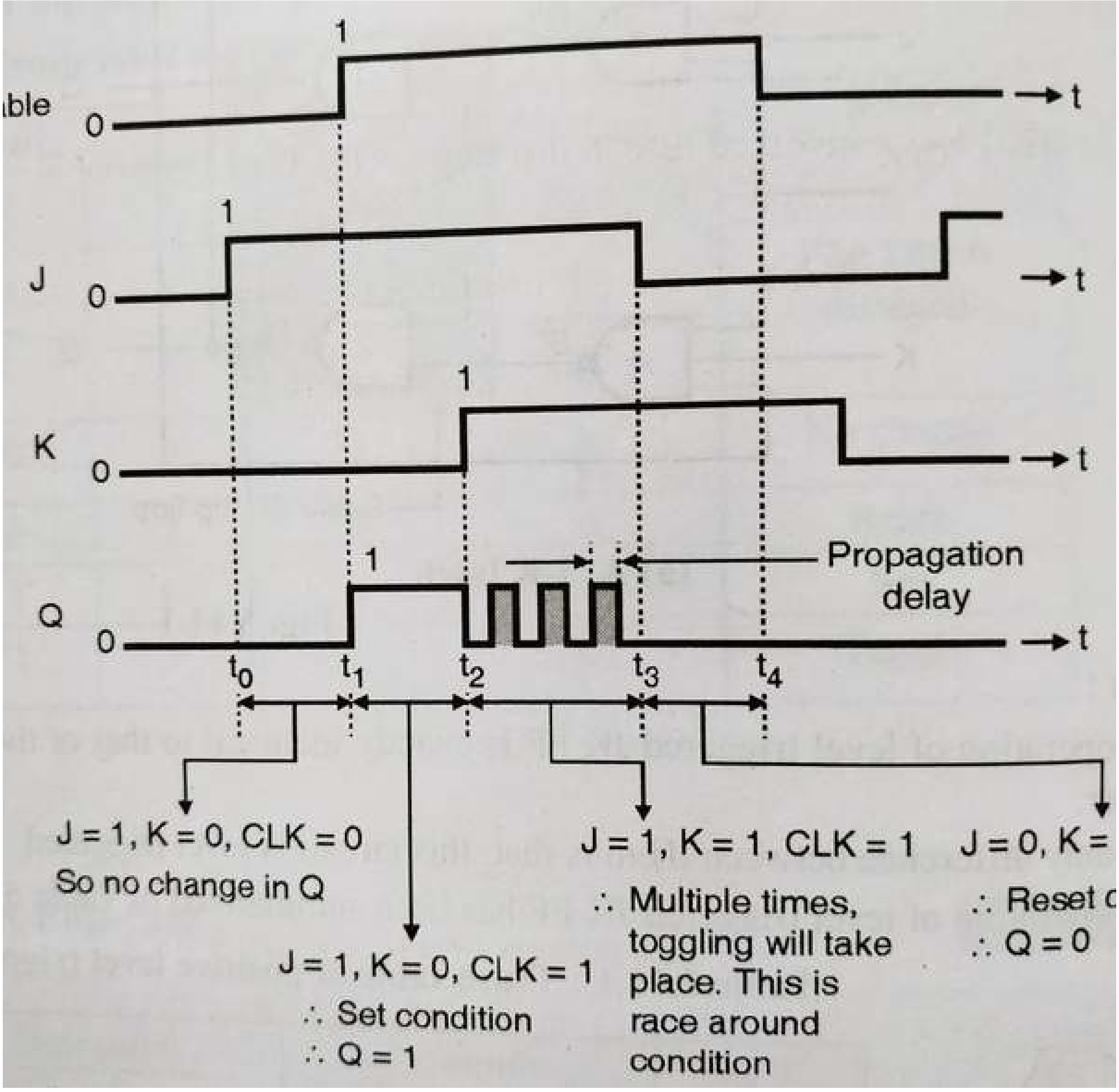
In "T Flip Flop", the state at an applied trigger pulse is defined only when the previous state is defined. It is the main drawback of the "T Flip Flop".

Timing Diagram.



Disadvantage of JK Flip flop:

If clock=1 and j=1 and k=1 the flip flop continuously toggled. That condition is known as race around condition. To avoid this, we can construct Master slave JK flip=flop.



J = 1, K = 0, CLK = 0
So no change in Q

J = 1, K = 0, CLK = 1
∴ Set condition
∴ Q = 1

J = 1, K = 1, CLK = 1
∴ Multiple times, toggling will take place. This is race around condition

J = 0, K =
∴ Reset c
∴ Q = 0