Sequential Circuit

In combinational circuits outputs depend only on its current inputs". In these circuits, there is no ability to retain the information regarding the state of the circuit ie no memory and any prior input level conditions have no effect on the present outputs.

In sequential logic circuit, the present values of outputs depend on both present values of the inputs and the past values of inputs. It means that it needs memory. A sequential logic circuit consists of two parts.

- *the memory elements . ie a flip-flop which is made up of an assembly of logic gates.*
- *the combinational logic circuits*

Sequential circuits find applications in digital systems as counters, registers, control logic, memories *and other complex functions.* Examples: -The elevator control, the traffic light system.
A general model of a sequential circuit is shown in the following Fig1.

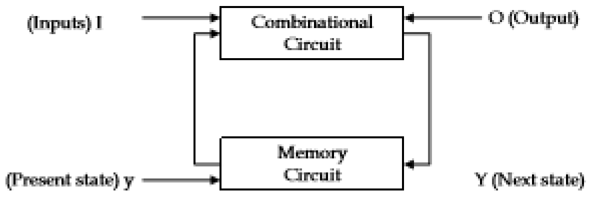


Fig. 1. Block Diagram of Sequential Circuit

Flip Flop (FF) and latches

The important memory elements in a sequential circuit are flip flops which are made up of logic gates. A gate itself has no storage capability, but several gates can be connected together in different ways to store information. These circuits are binary cells capable of storing one bit of data at a time. FF is a bistable multivibrator which has two stable states 1 and 0. It can remain in a state indefinitely and the state can be changed by applying proper triggering signal.

A latch is also a bistable circuit similar to a flip flop that can store. It exists in one of the two states (1 and 0), and in the absence of the input, it remains in that state. the ouput changes accordingly with changes in input as there is no control signal. It is the fundamental building block of flip flops.

The basic difference between a flip flop and a latch is that a latch doesn't contain any clock signal whereas flip-flops consist of a clock signal. That means the output of a latch changes as soon as its input changes but in a flip flop the change in output corresponding to a new input happens only when the control input (CLOCK signal) is enabled. Thus a control signal or otherwise called as clock signal is also there in a FF. Fig 2. Shows the general type of symbol used for a FF. It has fixed no .of inputs, and two outputs, normal ouput O and inverted output O'. There can be 'n' inputs to a flip-flop that is responsible for the simultaneous changes in their states. These signals are known as "excitation's".

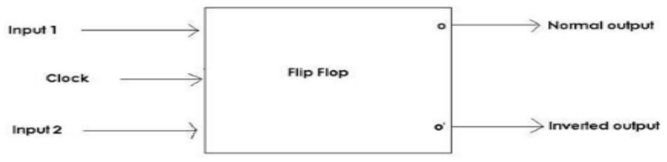


Fig 2. General block diagram of a Flip Flop

Clock Signal

Clock is a periodic sequence of pulses as shown in Fig 3, applied to control the operation (triggering) of a flip flop.

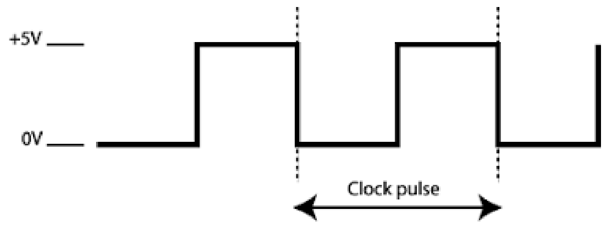


Fig 3. Clock Signal

The purposes of clock are to synchronize the over-all action and to prevent the flip-flop from changing states until the right time. The movement of a clock pulse is always from a 0 to 1 and then 1 to 0 of a signal. Thus it takes two transitions in a single signal. When it moves from 0 to 1 it is called a positive transition and when it moves from 1 to 0 it is called a negative transition. To understand more take a look at fig 4.

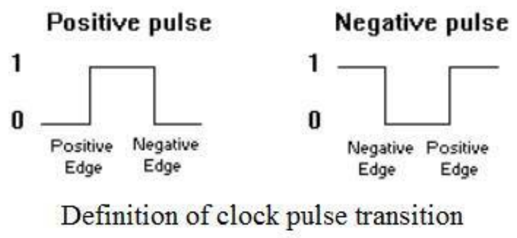


Fig 4. Clock Transition

If the clocked flip-flops are triggered during the 0 to 1 transition of the pulse, and the state transition starts as soon as the pulse reaches the HIGH level. If the other inputs change while the clock is still 1, a new output state may occur. The multi-transition problem can be stopped is the flip flop is made to respond to the positive or negative edge transition only, other than responding to the entire pulse duration.

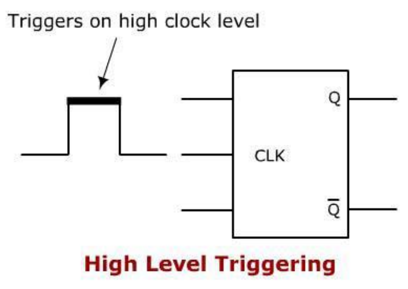
There are mainly two types of triggering methods. They differ in the manner in which the FFs respond to the pulse. They are.

1) **Level Triggering**

The flip flop is enabled during the entire high level(on time) or low level(off time) of clock signal. There are two types of level trigerring. They are

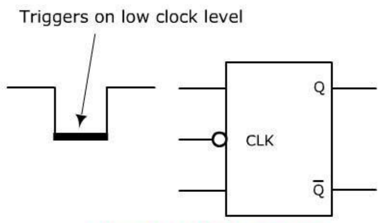
1.1) High level trigerring

Here flip flop changes its output during the HIGH state of clock signal, as highlighted in the figure. Take a look at the symbolic representation shown below. For high level clock input is represented using a straight line.



1.2. Low Level Triggering

Here flip flop changes it output during the LOW state of clock signal as highlighted in the figure. In the symbolic representation shown below clock input is shown using a straight line with a bubble.

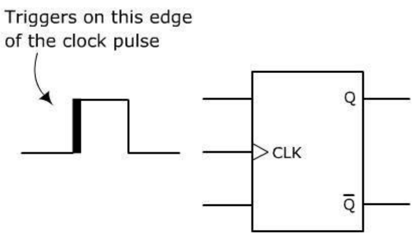


Low Level Triggering

2. **Edge Triggering:** here the flip flop responds to input at either positive edge transition or negative edge transition of the clock signal.

2.1 Positive Edge Triggering

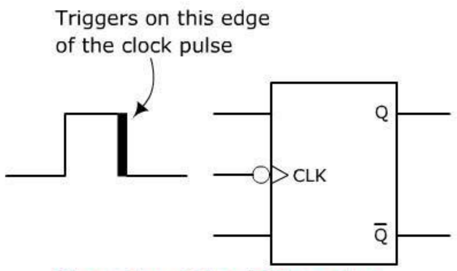
Here flip flop responds at a LOW to HIGH transition state, of the clock signal. These types of FFs are represented using a straight line along with a triangle.



Positive Edge Triggering

2.2 Negative Edge Triggering

Here flip flop responds at a HIGH to LOW transition state, of the clock signal. These types of FFs are represented using a straight line with a bubble along with a triangle.



Negative Edge Triggering

Types of Sequential Logic Circuits

Depending upon the timing of sequential circuit signal, the sequential logic circuits can be divided into two classes.

Synchronous Sequential Circuits

A synchronous sequential circuit is one in which the contents of the memory can change only at discrete instants time or at the of transitions of a clock. Since all the circuit action will take place under the control of a clock, so these circuits are known as clocked sequential circuit.

Advantages: They are easier to troubleshoot and design because its outputs can change only at specific instants of time i.e. every thing is synchronized to the clock signal transition.

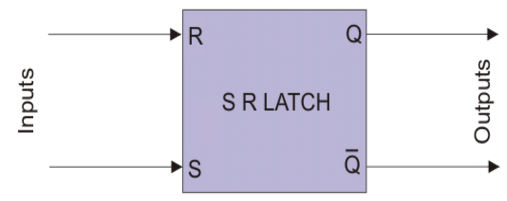
Asynchronous Sequential Logic Circuits

An asynchronous sequential logic circuits is one whose outputs can change state at any instant of time with the change of one or more of the inputs. The memory elements used in these systems are delay type memory elements. It can be regarded as combinational circuit with feed back.

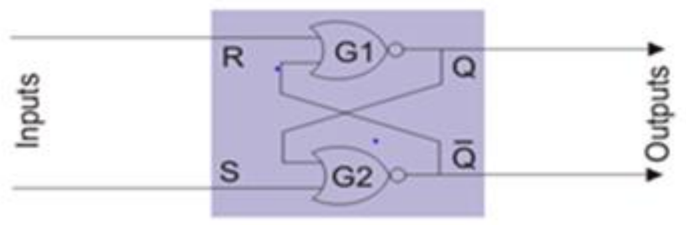
Disadvantage: It is difficult to design and troubleshoot and used only for simple configuration.

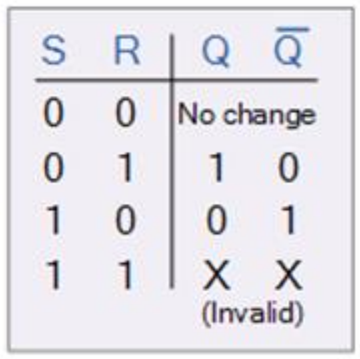
S R Latch(Set Reset latch)

Most simple type of latch is **S R latch**. It has two inputs S and R and two outputs Q and $\overline{Q}$. The state of this latch is determined by condition of Q. If Q is 1 the latch is said to be SET and if Q is 0 the latch is said to be RESET. This **S R Latch** or Flip flop can be designed either by two cross-coupled NAND gates or two-cross coupled NOR gates. When we design this latch by using NOR gates, it will be an active high S-R latch. That means it is SET when S = 1. When we design this latch by using NAND gates, it will be an active low S-R latch. That means it is SET when S = 0. **S R Flip Flop** is also called **SET RESET Flip Flop**.



Two-cross coupled NOR gates SR latch





S	R	Q	$\overline{Q}$
0	0	No change	
0	1	1	0
1	0	0	1
1	1	X	X
		(Invalid)	

In the above logic circuit if S = 1 and R = 0, Q becomes 1. Let us explain how.

- NOR gate always gives output 0 when at least one of the inputs is 1.
- So when S is applied as 1 the output of gate G2 i.e. $\overline{Q}$ is 0 irrespective of the condition of second input Q to the gate G2.
- Now $\overline{Q}$ is input of gate G1 so both the inputs of G1 become 0 as R is already 0. So, output of G1 is now $\overline{0+0}$ or 1.

- So whatever may be the previous condition of Q, it always becomes Q = 1 and $\overline{Q}$ = 0 when, S = 1 and R = 0. This is called SET condition of the latch.

In the above logic circuit if S = 0 and R = 1, Q becomes 0. Let us explain how.

- As we already said, a NOR gate always gives output 0 when at least one of the inputs is 1.
- So when R is applied as 1, the output of gate G1 i.e. Q is 0 irrespective of the condition of second input $\overline{Q}$ to the gate.
- So, whatever may be the previous condition of Q, it always becomes 0 this 0 is then fed back to input of gate G2. As here S is already 0, both inputs of G2 are 0. Hence output of G2 i.e. $\overline{Q}$ will be 1. So, Q = 0 and $\overline{Q}$ = 1 when, S = 0 and R = 1. This is called RESET condition of the latch.

In the above logic circuit if S = 0 and also R = 0, Q remains same as it was. Let us explain how.

- First suppose Q is previously 1.
- Now the inputs of G2 are 0 and 1 as S=0 and Q=1. So output of G2 i.e. $\overline{Q}$ is $\overline{0+1}$ or 0.
- Now both inputs of G1 are 0 as R=0 and $\overline{Q}$=0. So output of G1 i.e. Q is $\overline{0+0}$ or 1.
- Now suppose Q is previously 0.
- Now both inputs of G2 are 0 and 1 as S = 0 and Q = 0. So output of G2 i.e. $\overline{Q}$ is $\overline{0+0}$ or 1.
- Now the inputs of G1 are 0 and 1 as R=0 and $\overline{Q}$ = 1. So output of G1 i.e. Q is $\overline{0+1}$ or 0.
- So it is proved that Q remains same as it is when S = 0 and also R = 0 in S R latch or flip flop.

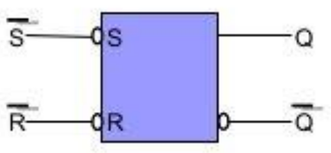
In the above logic circuit if S = 1 and also R = 1, the condition of Q is totally unpredictable. Let us explain how.

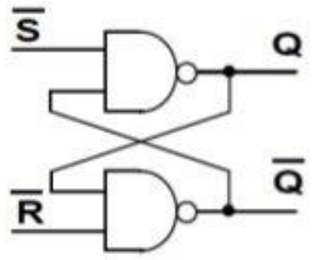
- First suppose Q is previously 1.
- Now both inputs of G2 are 1 as S = 1 and Q = 1. So output of G2 i.e. $\overline{Q}$ is $\overline{1+1}$ or 0.
- Now the inputs of G1 are 1 and 0 as R = 1 and $\overline{Q}$ = 0. So output of G1 i.e. Q is $\overline{1+0}$ or 0. That means Q is changed.
- Now Q is 0. So inputs of G2 are 1 and 0 as S = 1 and Q = 0. So output of G2 i.e. $\overline{Q}$ is $\overline{1+0}$ or 0. That means $\overline{Q}$ is unchanged.
- Now the inputs of G1 are 1 and 0 as R = 1 and $\overline{Q}$ = 0. So output of G1 i.e. Q is $\overline{1+0}$ or 0. That means Q is unchanged.

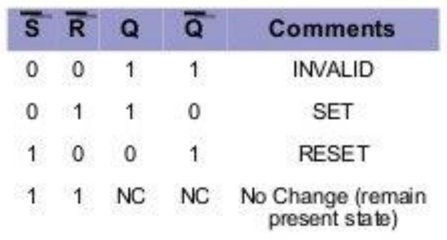
So, when both S and R are 1, it becomes unpredictable whether the value of output Q will be changed or unchanged. This condition of S R latch normally avoided.

As the latch is SET when S = 1(HIGH), the latch is called Active High S R Latch. There is other type of latch which is SET when, S = 0 (LOW), and this latch is known as Active Low S R Latch.

Two-cross coupled NAND gates SR latch(Active Low)







$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	Comments
0	0	1	1	INVALID
0	1	1	0	SET
1	0	0	1	RESET
1	1	NC	NC	No Change (remain present state)

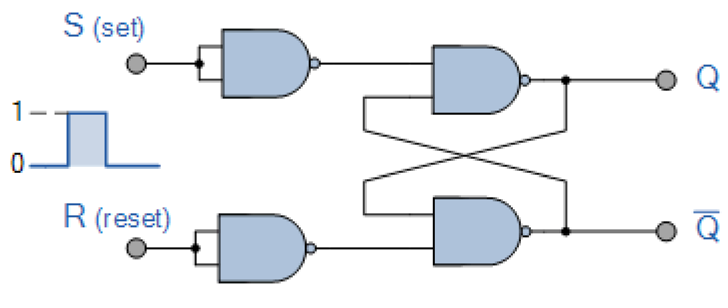
Case 1: Here when both inputs are 0, output of two NAND gates becomes 1. But it is invalid here as these two should be complementary as per definition.

Case 2: when $\overline{S} = 0$, $\overline{R} = 1$,the upper gate output goes to 1, ie the new output Q_{new} becomes 1, and sets the flip flop irrespective of the previous output Q. Now this Q is fed back to lower gate so that its output becomes 0. Thus the output becomes 1 irrespective of the previous input.

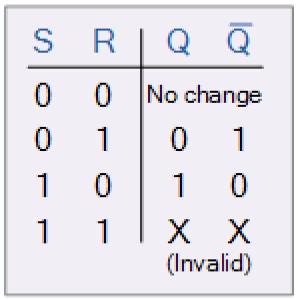
Case 3: when $\overline{S} = 1$, $\overline{R} = 0$, the lower gate output goes to 1, and this is fed back to the upper gate. The other input to upper gate is also 1, which makes the new output Q as 0, ie resets the FF.

Case 4: when $\overline{S} = 1$, $\overline{R} = 1$,the output of the FFs depends on the second input. If the previous output Q was 0. Then $\overline{Q} = 1$. The input to lower gates are 0,1. The new $\overline{Q}$, ie $\overline{Q}_{new}$ becomes 1 as same as previous $\overline{Q}$. This $\overline{Q}_{new} = 1$ is fed back to upper gate so that its input are both 1s and outputs new Q as 0, same as that of previous ouput Q. ie $Q_{new} = Q$. thus for this input condition there is no change in the output condition.

The active low SR latch can be changed to a an active high SR latch by introducing two simple NAND gates for inverting the normal S and R input signals as shown in figure.

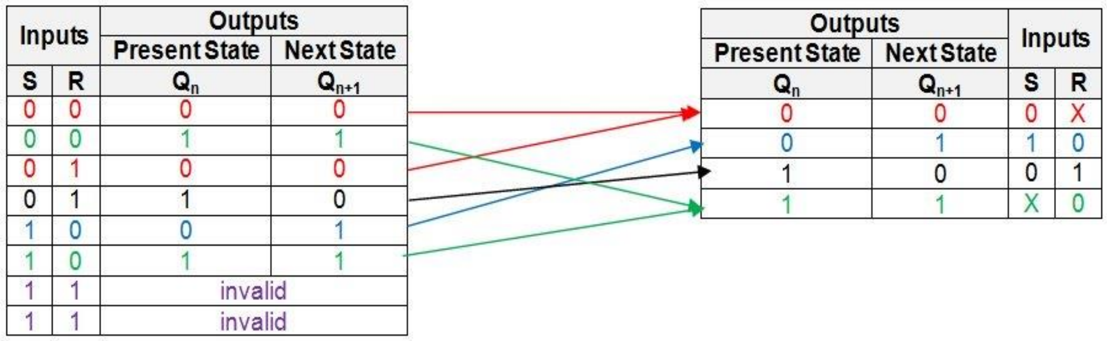


NAND Circuit



S	R	Q	$\overline{Q}$
0	0	No change	
0	1	0	1
1	0	1	0
1	1	X	X (Invalid)

Characterisic Table\ Truth Table and Excitation table



Inputs		Outputs	
		Present State	Next State
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	invalid	
1	1	invalid	

Truth Table of SR Flip-Flop

Outputs		Inputs	
Present State	Next State		
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table of SR Flip-Flop

The above figure shows the truth table (characteristic table) and excitation table of an SR latch.

While writing TT, write inputs first S & R in the first columns and write the outputs. Here Q represents the current output state.

The characteristic equation for the SR latch is obtained from this detailed truth table. By k map reduction method as follows

S\R$Q_n$	00	01	11	10
0	0	1	0	0
1	[1	1	x	X]

The characteristics equation is $Q_{n+1} = S + R'Q_n$

An **excitation table** shows the minimum inputs that are necessary to generate a particular next state In other words, we should find the required inputs to a FF so as to **"excite"** it to the next state when the current state is known. So the inputs are current output, Q and the next output Q_{next}. The table should be written for all possible state changes from 0 to 0, from 0 to 1, from 1 to 0 and from 1 to 1.

Case 1: Q = 0 to Q_{next} = 0 is No change condition. it happens when S=0 and R = 0. But output changes to 0 during S=0 and R=1{reset condition) also. ie S must be 0 and R can be 0 or 1. So can be written as S=0 and R = x(don't care).

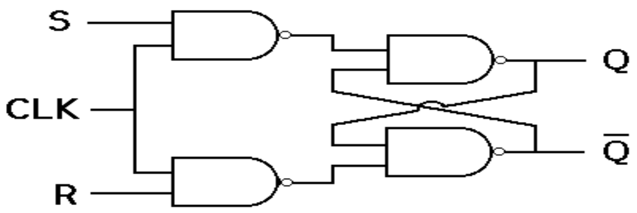
Case 2: Q = 0 to Q_{next} = 1, the new output should be 1 can be achieved by applying set condition. ie S=1 and R = 0.

Case 3: Q = 1 to Q_{next} = 0, the new output should be 0 can be achieved by applying reset condition. ie S=0 and R = 1.

Case 4: Q = 1 to Q_{next} = 1 is No change condition. it happens when S=0 and R = 0. But output changes to 1 when S=1 and R=0{set condition) also. ie R must be 0 and S can be 0 or 1. So can be written as S= x(don't care) and R = 0.

SR (Set - Reset) Flip-Flop

The SR latch which is synchronized to a clock pulse, is SR FF. i.e. changes occur only when a clock pulse occurs. This is done by adding two inputs NAND gates with the clock pulse input. In order to operate these devices effectively, the following conditions must be met:
Note : The circuit action can occur only when CLK=1, when CLK=0, the FF outputs do not change.



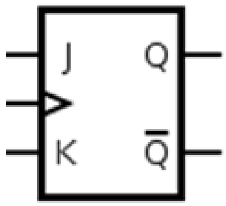
SR Flip Flop

Input			Current output	Output	
CLK	S	R	Q_n	Mode	Q_{n+1}
O	-	-	-	No action	-
1	0	0	0	Hold	0
1	0	0	1		1
1	0	1	0	Reset	0
1	0	1	1		0
1	1	0	0	Set	1
1	1	0	1		1
1	1	1	0	Invalid	-
1	1	1	1		-

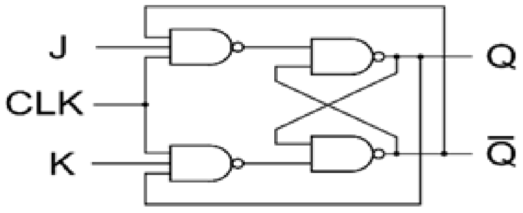
Truth Table for clocked S-R FF.

J - K Flip-Flop

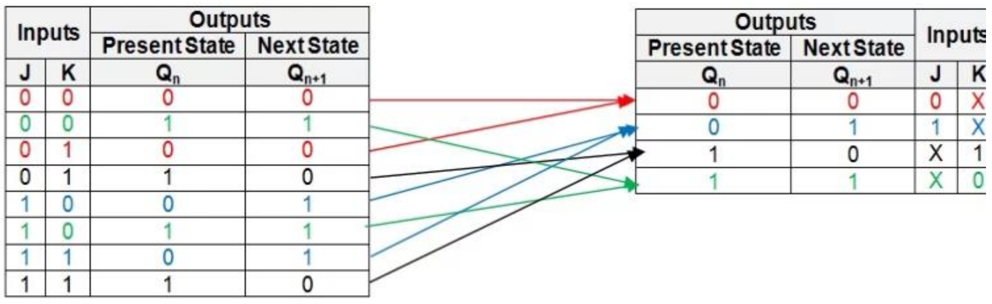
We saw that the clock SR FF has an indeterminate state when S=R=1. This troublesome restrictions S=R=1 can be removed by modifying the SR FF. This refined FF is known as the JK FF. This modifications involves feeding the outputs of the FF back into the inputs of circuit as shown in Fig. The J=1 & K = 0 input performs *set* function causing the output (next state) to be 1. The K=1 & J = 0 input performs *reset* function causing the output (next state) to be 0. When both J and K are 1, the next state of the FF is the complement of the present state i.e. output (next state) is reversed. When J=K=0, then output is unchanged. The block diagram, FF circuit, transition table and K map of JK FF are as shown below.



J K FF logic symbol



J K FF logic circuit



Inputs		Outputs	
		Present State	Next State
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

Truth Table of JK Flip-Flop

Outputs		Inputs	
Present State	Next State		
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table of JK Flip-Flop

Working of JK FF

Case 1: When J=0, K = 0, the outputs of first set of gates remains 1, ie the NAND gates are disabled and the new final output is Q itself.

Case 2: Reset condition, when J=0, K = 1, the upper NAND is disabled with 1 output and lower NAND output depends on current output Q. If Q =1, the first lower gate outputs 0 and it comes to the input of second lower NAND. And the new Q’ becomes 1, which is fed to second upper NAND whose other input is already 1. Then it outputs a 0.
If current output, Q is 0, then it comes to the input of second lower gate and outputs 1,ie Q’= 1 and this is fed to the second upper gate whose other input is already 1. Thus it outputs a 0.

Case 3: Set condition, when J=1, K=0, the first lower NAND is disabled by generating 1. If current output, Q =1, it also comes to the input of second lower NAND. And the new Q’ becomes 0, which is fed to second upper NAND to output a 1.
If current output, Q is 0, then it moves to the input of second lower gate producing the new final Q’ output = 1, and this is fed to the first upper gate, when enabled produces 0 and feeds second upper gate so that it gives next output as 1.

Case 4: when J=1, K=1: toggling condition: the ouputs of first upper & lower NAND depends on current output Q. if Q=0, then first upper gate generates 0 when its clock is enabled. This when passed through the second upper gate, we get the next output as 1, which is the compliment of previous output.
Now if Q=1, output of first lower gate becomes 0 and first upper gate becomes 1 when enabled, and the Q’ output changes to 1. This goes to the input side of second upper NAND, which changes the next output Q to 0. Again compliment of previous. Thus the output toggles when CLOCK gets enabled .

Characteristic Equation of J-K FF .

To determine the characteristic equation, obtain the k-map for next output, Q_{n+1} from the truth table as shown.

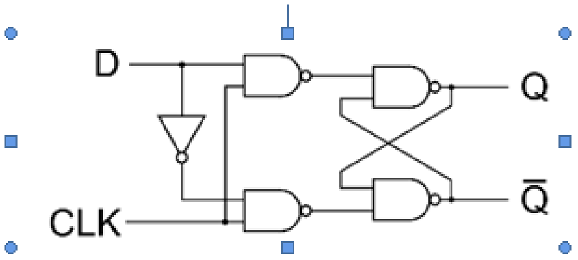
J/ KQn	00	01	11	10
0	0	1	0	0
1	1}	1	0	{1

Fig K- map of J-K Flip Flop

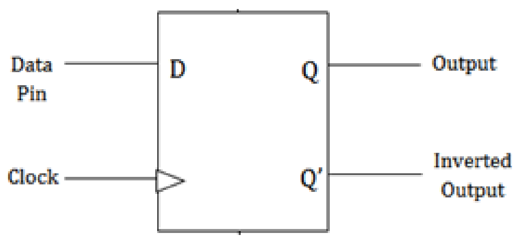
The characteristic equation is Q_{n+1} = JQn'+ K'Qn.

D flip flop (Data FF)

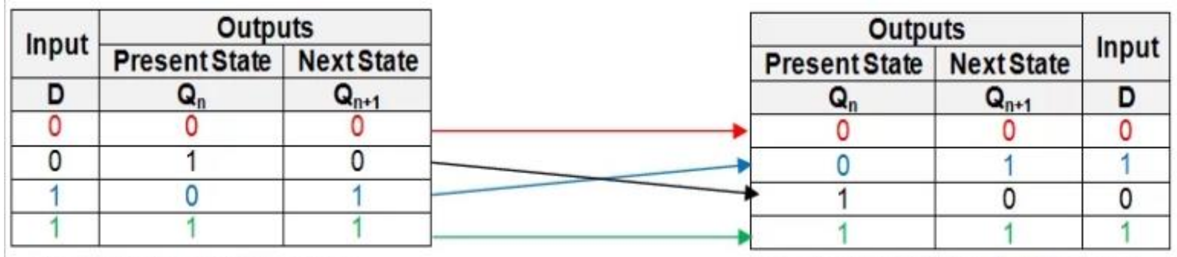
A Data FF or Delay FF is a single input flip flop along with clock signal. When enabled the input D is transferred directly to the output. The logic symbol & logic diagram is shown below.



D flip flop logic diagram



Logic Symbol



Input	Outputs	
	Present State	Next State
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Truth Table of D Flip-Flop

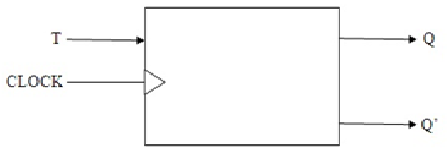
Outputs		Input
Present State	Next State	
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table of D Flip-Flop

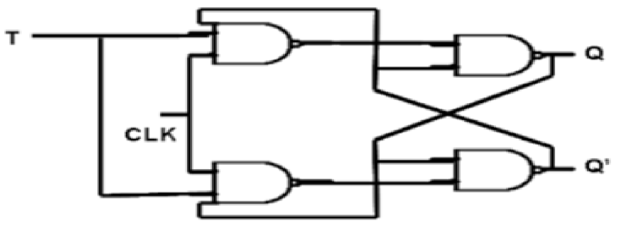
Basically it is a modification of RS flip flop. The D input goes directly to S input and its compliment is applied to the R input. When the clock is low, both gates are disabled and D cannot change the value of Q. on the other hand, when clock is high, both gates are enabled and Q will change to D. Its characteristics equation, from truth table is $Q_{n+1} = DQ_n' + DQ_n$.
Thus $Q_{n+1} = D(Q_n' + Q_n) = D$

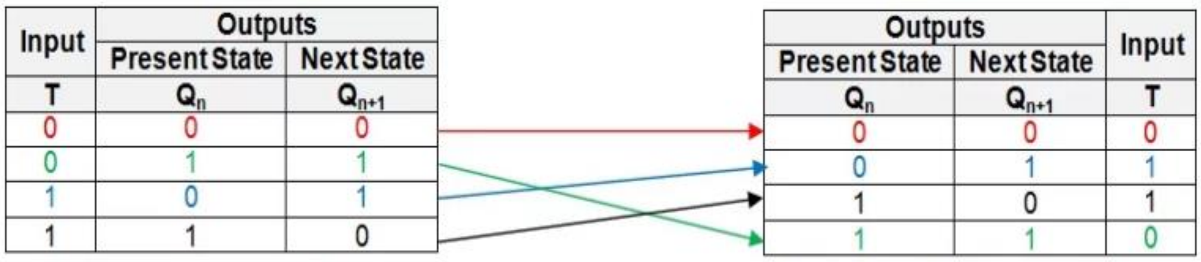
T Flip Flop

It is the highly simplified version of J-K flip-flop. It consists of one controlled input value. T represents TOGGLE. If the two inputs J and K of a J-K flip-flop are tied together it is referred to as a T flip-flop. Hence, a T flip-flop has only one input T and two outputs Q and Q'. The name T flip-flop actually indicates the fact that the flip-flop has the ability to toggle. It has actually only two states—*toggle state* and *memory state*. Since there are only two states, a T flip flop is a very good option to use. When T is at the HIGH state it begins to toggle as soon as it detects the new clock signal, otherwise remains at the same state as it was previously. ie if T = 1 and the device is clocked, then the output toggles (compliments) its state.



T-Flip-Flop Block Diagram





Input	Outputs	
	Present State	Next State
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table of T Flip-Flop

Outputs		Input
Present State	Next State	
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

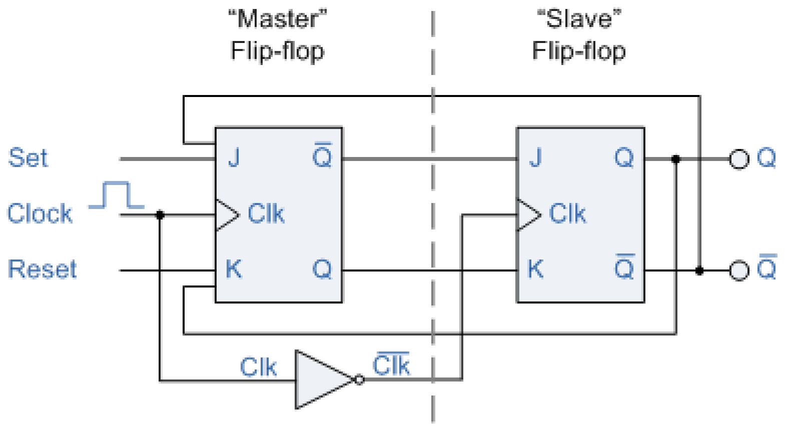
Excitation Table of T Flip-Flop

The characteristic equation is $Q_{n+1} = T'Q_n + TQ_n'$

Master Slave JK Flip Flop

Race Around Condition In JK Flip-flop – For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop unstable or uncertain. This problem is called race around condition in J-K flip-flop. This problem (Race Around Condition) can be avoided by ensuring that the clock input is at logic "1" only for a very short time. This introduced the concept of **Master Slave JK** flip flop.

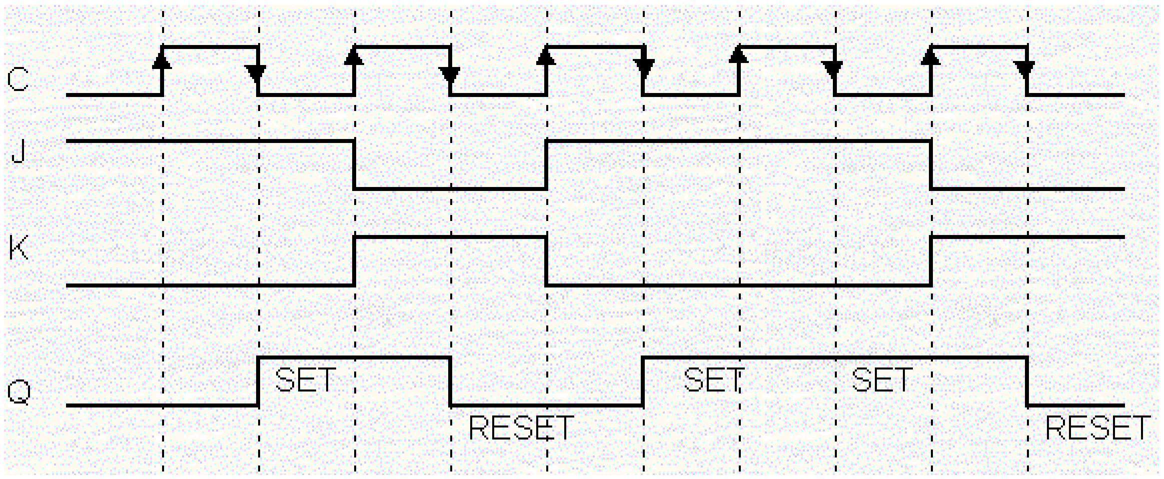
Master-slave flip flop is designed using two separate flip flops. Out of these, one acts as the master and the other as a slave. The figure of a master-slave J-K flip flop is shown below.



Master Slave Flip Flop

From the above figure you can see that both the J-K flip flops are presented in a series connection. The output of the master J-K flip flop is fed to the input of the slave J-K flip flop. The output of the slave J-K flip flop is given as a feedback to the input of the master J-K flip flop. The clock pulse [Clk] is given to the master J-K flip flop and it is sent through a NOT Gate and thus inverted before passing it to the slave J-K flip flop.

When Clk=1, the master J-K flip flop gets disabled. The Clk input of the master input will be the opposite of the slave input. So the master flip flop output will be recognized by the slave flip flop only when the Clk value becomes 0. Thus, when the clock pulse males a transition from 1 to 0, the locked outputs of the master flip flop are fed through to the inputs of the slave flip-flop making this flip flop edge or pulse-triggered. To understand better take a look at the timing diagram illustrated below.



Master Slave J-K Flip Flop Timing Diagram

Thus, the circuit accepts the value in the input when the clock is HIGH, and passes the data to the output on the falling-edge of the clock signal. This makes the Master-Slave J-K flip flop a Synchronous device as it only passes data with the timing of the clock signal.