

UNIT-III

Synchronous Sequential Logic Circuits

SEQUENTIAL CIRCUITS

Digital electronics is classified into **combinational logic** and **sequential logic**.

Combinational logic output depends on the present inputs levels, whereas sequential logic output not only depends on the input levels, but also stored levels (previous output history).

Combinational Circuits

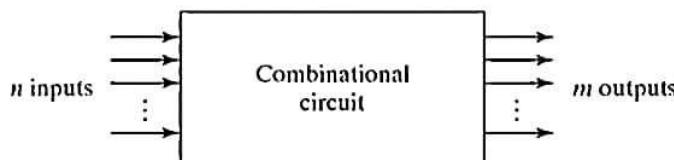
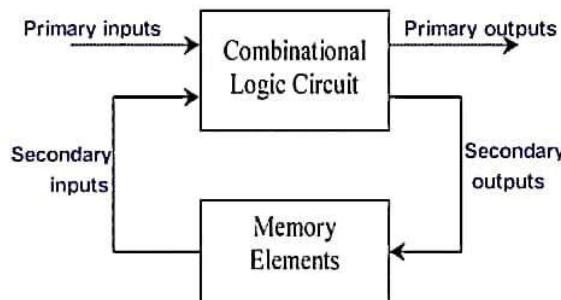


Fig. Block Diagram of Combinational Circuit

Sequential Circuits



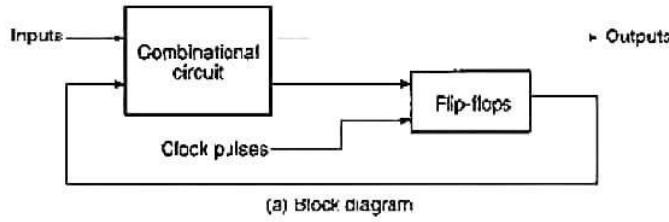
The memory elements are devices capable of storing binary info. The binary info stored in the memory elements at any given time defines the state of the sequential circuit. The input and the present state of the memory element determine the output. Memory elements next state is also a function of external inputs and present state. A sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

There are two types of sequential circuits. Their classification depends on the timing of their signals:

- ❖ Synchronous sequential circuits
- ❖ Asynchronous sequential circuits

Asynchronous sequential circuit

This is a system whose outputs depend upon the order in which its input variables change and can be affected at **any instant of time**.



(a) Block diagram

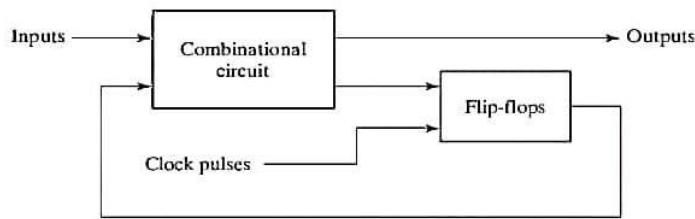


(b) Timing diagram of clock pulses

Synchronous sequential circuits

This type of system uses storage elements called flip-flops that are employed to change their binary value only **at discrete instants of time**.

Synchronous sequential circuits use logic gates and flip-flop storage devices. Sequential circuits have a clock signal as one of their inputs. All state transitions in such circuits occur only when the clock value is either 0 or 1 or happen at the rising or falling edges of the clock depending on the type of memory elements used in the circuit. Synchronization is achieved by a timing device called a clock pulse generator. Clock pulses are distributed throughout the system in such a way that the flip-flops are affected only with the arrival of the synchronization pulse. Synchronous sequential circuits that use clock pulses in the inputs are called **clocked-sequential circuits**.

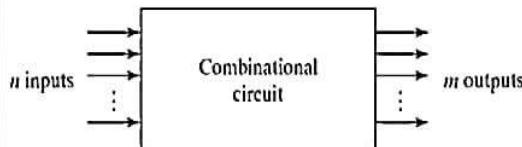
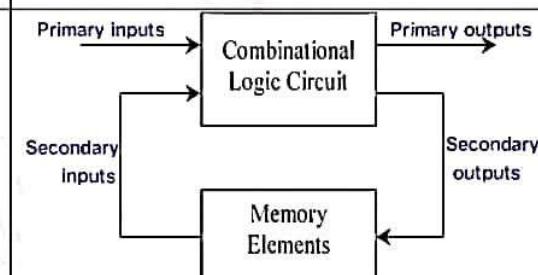


(a) Block diagram



(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

Combinational Circuits	Sequential Circuits
1. The circuit whose output at any instant depends only on the input present at that instant only is known as combinational circuit.	1. The circuit whose output at any instant depends not only on the input present but also on the past output is known as sequential circuit
2. This type of circuit has no memory unit.	2. This type of circuit has memory unit for store past output.
3. Examples of combinational circuits are half adder, full adder, magnitude comparator, multiplexer, demultiplexer e.t.c.	3. Examples of sequential circuits are Flip flop, register, counter e.t.c.
4. Faster in Speed	Slower compared to Combinational Circuit
<p style="text-align: center;">Combinational Circuits</p>  <p>Fig. Block Diagram of Combinational Circuit</p>	

A sequential circuit can further be categorized into **Synchronous** and **Asynchronous**.

Here is the difference between synchronous and asynchronous sequential circuits:

Synchronous Sequential Circuit: Output changes at discrete interval of time. It is a circuit based on an equal state time or a state time defined by external means such as clock. Examples of synchronous sequential circuit are Flip Flops, Synchronous Counter.

Asynchronous Sequential Circuit: Output can be changed at any instant of time by changing the input. It is a circuit whose state time depends solely upon the internal logic circuit delays. Example of asynchronous sequential circuit is Asynchronous Counter.

Basic Flip Flops:

A circuit that changes from 1 to 0 or from 0 to 1 when current is applied. It is one bit storage location.

Flip flops are actually an application of logic gates. When a certain input value is given to them, they will be remembered and executed, if the logic gates are designed correctly. A higher application of flip flops is helpful in designing better electronic circuits.

The most commonly used application of flip flops is in the implementation of a feedback circuit. As a memory relies on the feedback concept, flip flops can be used to design it.

Latches and flip-flops are the basic elements for storing information. One latch or flip-flop can store one bit of information. The main difference between latches and flip-flops is that for latches, their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately when their inputs change. Flip-flops, on the other hand, have their content change only either at the rising or falling edge of the enable signal. This enable signal is usually the controlling clock signal. After the rising or falling edge of the clock, the flip-flop content remains constant even if the input changes.

There are basically four main types of latches and flip-flops: SR, D, JK, and T. The major differences in these flip-flop types are the number of inputs they have and how they change state. For each type, there are also different variations that enhance their operations. In this chapter, we will look at the operations of the various latches and flip-flops.

1. RS Latch

- ❖ RS latch have two inputs, S and R. S is called set and R is called reset.
- ❖ The S input is used to produce HIGH on Q (i.e. store binary 1 in flip-flop).
- ❖ The R input is used to produce LOW on Q (i.e. store binary 0 in flip-flop).
- ❖ Q' is Q complementary output, so it always holds the opposite value of Q.
- ❖ The output of the S-R latch depends on current as well as previous inputs or state, and its state (value stored) can change as soon as its inputs change.

There are mainly four types of flip flops that are used in electronic circuits.

1. The basic Flip Flop or S-R Flip Flop
2. Delay Flip Flop [D Flip Flop]
3. J-K Flip Flop
4. T Flip Flop

2. S-R Flip Flop:

The SET-RESET flip flop is not designed with the help of two NOR gates and also two NAND gates. These flip flops are also called S-R Latch.

S-R Flip Flop using NOR Gate

The design of such a flip flop includes two inputs, called the SET [S] and RESET [R]. There are also two outputs, Q and Q'. The diagram and truth table is shown below.

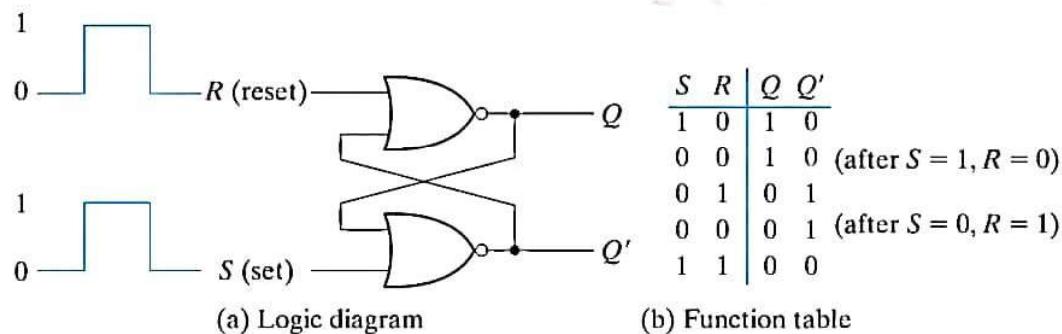


Fig. 5-3 SR Latch with NOR Gates

The operation has to be analyzed with the 4 inputs combinations together with the 2 possible previous states.

From the diagram it is evident that the flip flop has mainly four states. They are

1. When $S=1, R=0$ the output becomes $Q=1, Q'=0$

This SR flip flop function table is constructed based on the XOR gate. In XOR gate if any of the input is 1 the output becomes 1.

In this state when $S=1$ and $R=0$ the output Q becomes set (1). So this state is also called the SET state.

2. When S=0, R=1, the output becomes Q=0, Q'=1

In this state When R=1 it resets the output. So this state is known as the RESET state.

In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the value of S.

3. When S=0, R=0 the output is Q & Q' = Remember (memory)

If both the values of S and R are switched to 0, then the circuit remembers the value of S and R in their previous state.

4. When S=1, R=1 the output Q=0, Q'=0 [Invalid]

This is an invalid state because the values of both Q and Q' are 0. They are supposed to be compliments of each other. Normally, this state must be avoided.

S-R Flip Flop using NAND Gate

The above SR flip flop can be constructed using NAND gate.

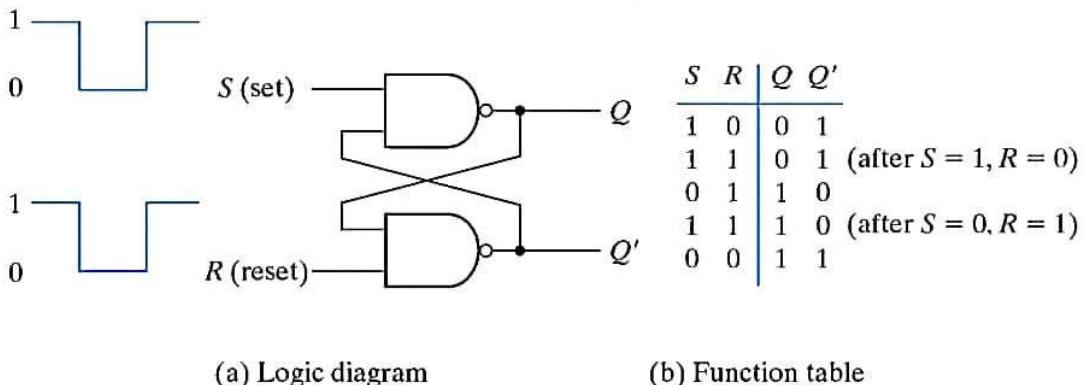


Fig. 5-4 SR Latch with NAND Gates

Like the NOR Gate S-R flip flop, this one also has four states. They are

1. S=1, R=0, Q=0, Q'=1

This state is also called the SET state.

2. S=0, R=1, Q=1, Q'=0

This state is known as the RESET state.

In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the compliment value of S.

3. S=0, R=0, Q=1, & Q' =1 [Invalid]

If both the values of S and R are switched to 0 it is an invalid state because the values of both Q and Q' are 1. They are supposed to be compliments of each other. Normally, this state must be avoided.

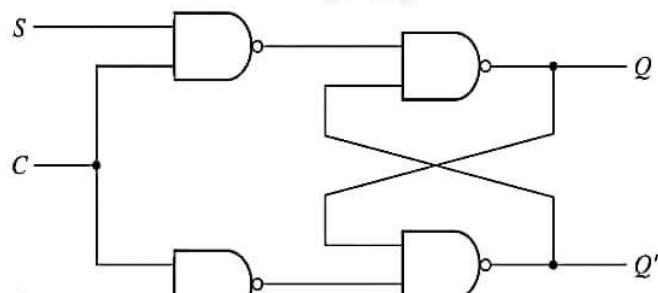
4. S=1, R=1, Q & Q'= Remember

If both the values of S and R are switched to 1, then the circuit remembers the value of S and R in their previous state.

Clocked S-R Flip Flop

- ❖ It is also called a Gated S-R flip flop.
- ❖ The problems with S-R flip flops using NOR and NAND gate is the invalid state.
- ❖ This problem can be overcome by using a bistable SR flip-flop that can change outputs when certain invalid states are met, regardless of the condition of either the Set or the Reset inputs.
- ❖ For this, a clocked S-R flip flop is designed by adding two AND neither gates to a basic NOR Gate flip flop.
- ❖ The circuit diagram and truth table is shown below.

The circuit of the S-R flip flop using NAND Gate and its truth table is shown below.



(a) Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; Reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

Fig. 5-5 SR Latch with Control Input

- A clock pulse [CP] is given to the inputs of the AND Gate.
- When the value of the clock pulse is '0', the outputs of both the AND Gates remain '0'.

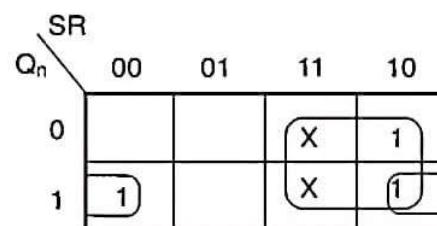
- As soon as a pulse is given the value of CP turns '1'.
- This makes the values at S and R to pass through the NOR Gate flip flop. But when the values of both S and R values turn '1', the HIGH value of CP causes both of them to turn to '0' for a short moment.
- As soon as the pulse is removed, the flip flop state becomes intermediate.
- Thus either of the two states may be caused, and it depends on whether the set or reset input of the flip-flop remains a '1' longer than the transition to '0' at the end of the pulse. Thus the invalid states can be eliminated.

Excitation Table of the SR Latch

- During the design process we usually know the transition from present state to next state and wish to find the latch input conditions that will cause the required transition.
- For this reason, we need a table that lists the required inputs for a given change of state. Such a table is called an excitation table, and it specifies the excitation behavior of the sequential circuits. These are used in the synthesis (design) of sequential circuits, which we shall see later.
- The excitation of the SR latch is as follows:

Excitation Table:

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

K Map for Q_{n+1} :

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

Note: Indeter = not used

3. D Flip Flop

- D flip flop is actually a slight modification of the above explained clocked SR flip-flop.
From the figure you can see that the D input is connected to the S input and the complement of the D input is connected to the R input.
- The D input is passed on to the flip flop when the value of CP is '1'.
- When CP is HIGH, the flip flop moves to the SET state. If it is '0', the flip flop switches to the CLEAR state.
- As long as the clock input C = 0, the SR latch has both inputs equal to 0 and it can't change its state regardless of the value of D
- When C is 1, the latch is placed in the set or reset state based on the value of D.

If D = 1, the Q output goes to 1.

If D = 0, the Q output goes to 0.

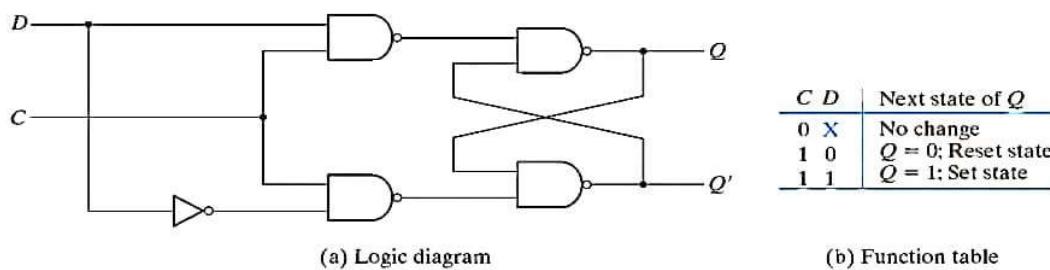
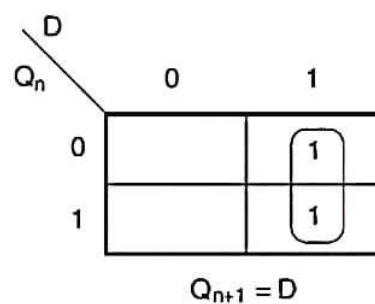


Fig. 5-6 D Latch

Excitation Table:

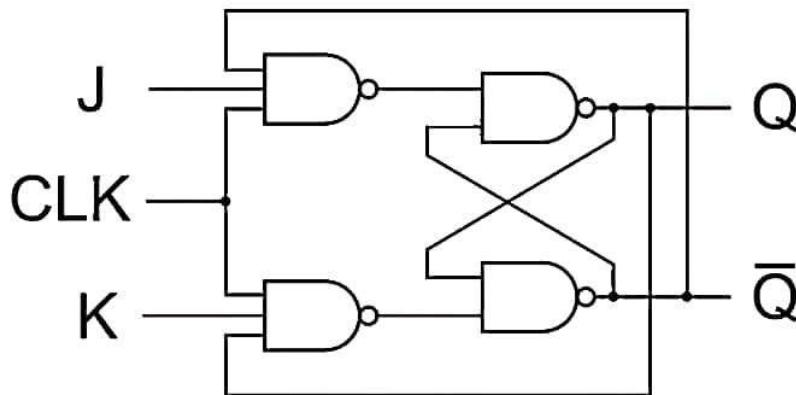
Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

K- Map for Q_{n+1} :



3. J-K Flip Flop

- ❖ A J-K flip flop can also be defined as a modification of the S-R flip flop. The only difference is that the intermediate state is more refined and precise than that of a S-R flip flop.



- ❖ The behavior of inputs J and K is same as the S and R inputs of the S-R flip flop. The letter J stands for SET and the letter K stands for CLEAR.
- ❖ When both the inputs J and K have a HIGH state, the flip-flop switches to the complement state. So, for a value of $Q = 1$, it switches to $Q=0$ and for a value of $Q = 0$, it switches to $Q=1$.
- ❖ The circuit includes two 3-input AND gates. The output Q of the flip flop is returned back as a feedback to the input of the AND along with other inputs like K and clock pulse [CP].
- ❖ So, if the value of CP is '1', the flip flop gets a CLEAR signal and with the condition that the value of Q was earlier 1.
- ❖ Similarly output Q' of the flip flop is given as a feedback to the input of the AND along with other inputs like J and clock pulse [CP].
- ❖ So the output becomes SET when the value of CP is 1 only if the value of Q' was earlier 1.
- ❖ The output may be repeated in transitions once they have been complimented for $J=K=1$ because of the feedback connection in the JK flip-flop.
- ❖ This can be avoided by setting a time duration lesser than the propagation delay through the flip-flop.
- ❖ The restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction.

Characteristic table:

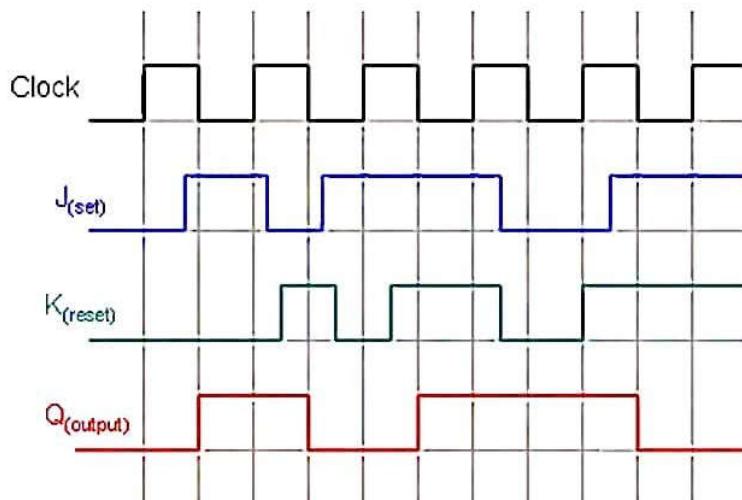
Clk	J	K	Q_{n+1}
0	X	X	Memory
1	0	0	Memory
1	0	1	0
1	1	0	1 (Set)
1	1	1	Toggle

Excitation table for JK Flipflop K map for Q_{n+1} :

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

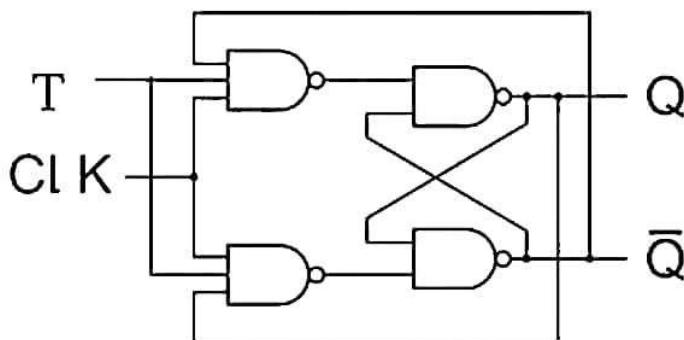
$Q_n \setminus Q_n$	00	01	11	10
0			1 1	
1	1			1

$$Q_{n+1} = J \cdot \overline{Q_n} + \overline{K} \cdot Q_n$$

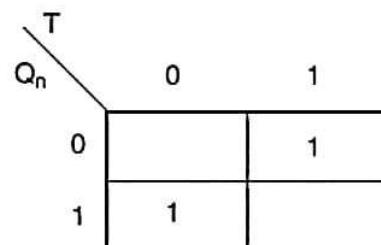
Timing Diagram:

4. T Flip Flop

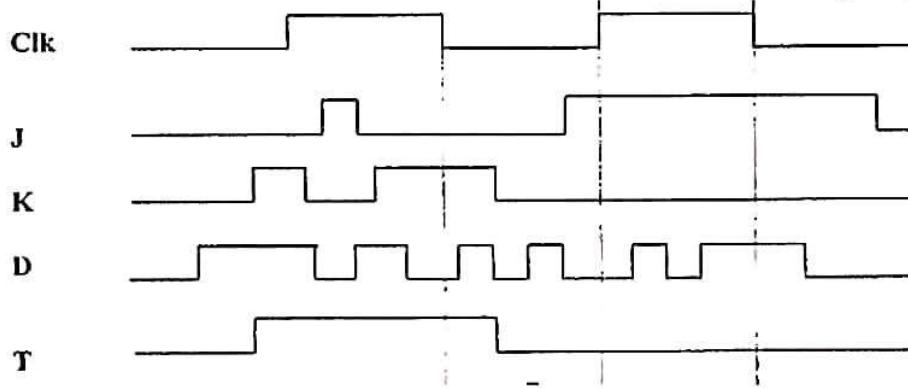
- ❖ This is a much simpler version of the J-K flip flop.
- ❖ Both the J and K inputs are connected together and thus are also called a single input J-K flip flop.
- ❖ When clock pulse is given to the flip flop, the output begins to toggle.
- ❖ Here also the restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction. Take a look at the circuit and truth table below.

Excitation Table for T Flip Flop:K map for T Flip Flop:

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

Characteristic Equation:

$$Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$$

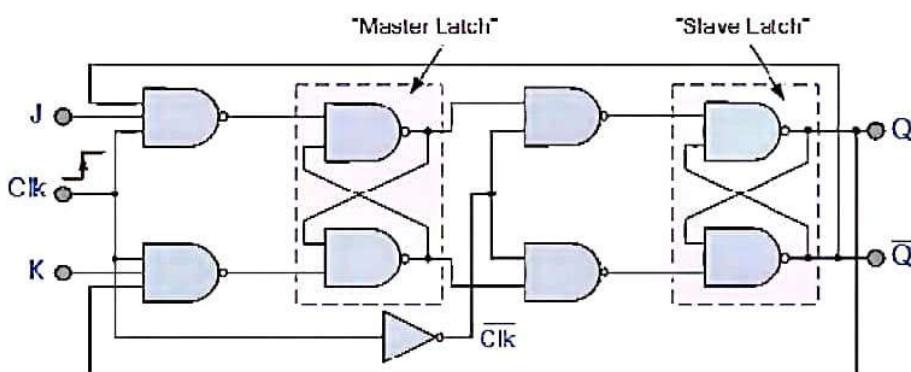


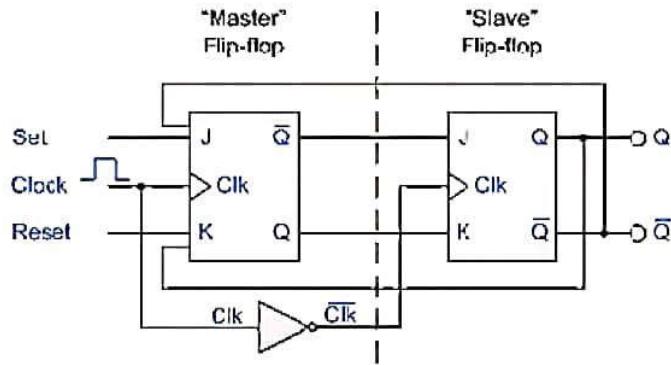
Master-Slave Flip Flop Circuit

Before knowing more about the master-slave flip flop you have to know more on the basics of a J-K flip flop and S-R flip flop. To know more about the flip flops, click on the link below.

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.

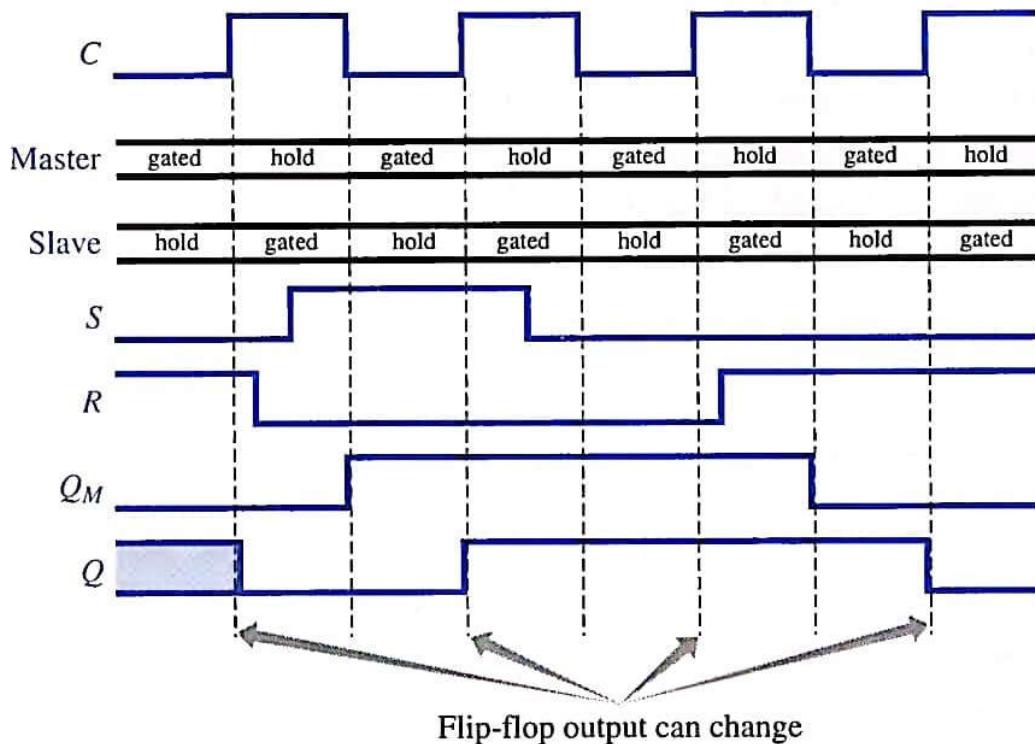
From the below figure you can see that both the J-K flip flops are presented in a series connection. The output of the master J-K flip flop is fed to the input of the slave J-K flip flop. The output of the slave J-K flip flop is given as a feedback to the input of the master J-K flip flop. The clock 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.





Working

When Clock=1, the master J-K flip flop gets disabled. The Clock 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 Clock value becomes 0. Thus, when the clock pulse makes 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.



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.

Triggering of Flip Flops

The output of a flip flop can be changed by bring a small change in the input signal. This small change can be brought with the help of a clock pulse or commonly known as a trigger pulse.

When such a trigger pulse is applied to the input, the output changes and thus the flip flop is said to be triggered. Flip flops are applicable in designing counters or registers which stores data in the form of multi-bit numbers. But such registers need a group of flip flops connected to each other as sequential circuits. And these sequential circuits require trigger pulses.

The number of trigger pulses that is applied to the input of the circuit determines the number in a counter. A single pulse makes the bit move one position, when it is applied onto a register that stores multi-bit data.

In the case of SR Flip Flops, the change in signal level decides the type of trigger that is to be given to the input. But the original level must be regained before giving a second pulse to the circuit.

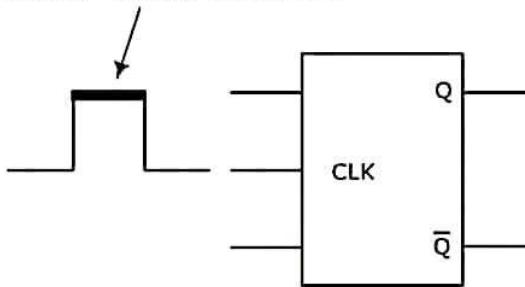
If a clock pulse is given to the input of the flip flop at the same time when the output of the flip flop is changing, it may cause instability to the circuit. The reason for this instability is the feedback that is given from the output combinational circuit to the memory elements. This problem can be solved to a certain level by making the flip flop more sensitive to the pulse transition rather than the pulse duration.

There are mainly four types of pulse-triggering methods. They differ in the manner in which the electronic circuits respond to the pulse. They are

1. High Level Triggering

When a flip flop is required to respond at its HIGH state, a HIGH level triggering method is used. It is mainly identified from the straight lead from the clock input. Take a look at the symbolic representation shown below.

Triggers on high clock level

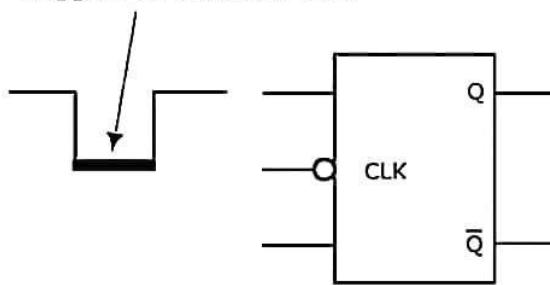


High Level Triggering

4. Low Level Triggering

When a flip flop is required to respond at its LOW state, a LOW level triggering method is used.. It is mainly identified from the clock input lead along with a low state indicator bubble. Take a look at the symbolic representation shown below.

Triggers on low clock level

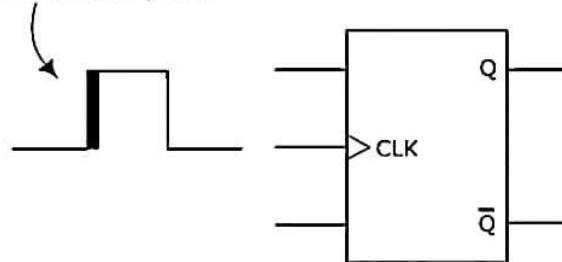


Low Level Triggering

3. Positive Edge Triggering

When a flip flop is required to respond at a LOW to HIGH transition state, POSITIVE edge triggering method is used. It is mainly identified from the clock input lead along with a triangle. Take a look at the symbolic representation shown below.

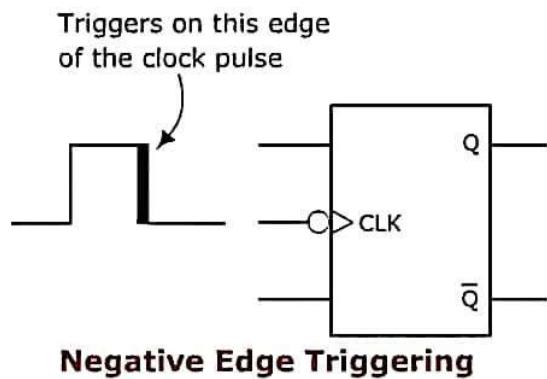
Triggers on this edge
of the clock pulse



Positive Edge Triggering

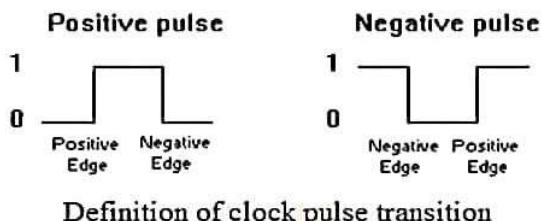
4. Negative Edge Triggering

When a flip flop is required to respond during the HIGH to LOW transition state, a NEGATIVE edge triggering method is used. It is mainly identified from the clock input lead along with a low-state indicator and a triangle. Take a look at the symbolic representation shown below.



Clock Pulse Transition

The movement of a trigger 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 the images below.



The clocked flip-flops already introduced 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. If the flip-flop is made to then the multiple-transition problem can be eliminated.

The multi-transition problem can be stopped if the flip flop is made to respond to the positive or negative edge transition only, other than responding to the entire pulse duration.

Flip Flop Conversion

For the conversion of one flip flop to another, a combinational circuit has to be designed first. If a JK Flip Flop is required, the inputs are given to the combinational circuit and the output of the combinational circuit is connected to the inputs of the actual flip flop. Thus, the output of the actual flip flop is the output of the required flip flop. The following flip flop conversions will be explained.

- **SR Flip Flop to JK Flip Flop**
- **JK Flip Flop to SR Flip Flop**
- **SR Flip Flop to D Flip Flop**
- **D Flip Flop to SR Flip Flop**
- **JK Flip Flop to T Flip Flop**
- **JK Flip Flop to D Flip Flop**
- **D Flip Flop to JK Flip Flop**

SR Flip Flop to JK Flip Flop

As told earlier, J and K will be given as external inputs to S and R. As shown in the logic diagram below, S and R will be the outputs of the combinational circuit.

The truth tables for the flip flop conversion are given below. The present state is represented by Q_p and Q_{p+1} is the next state to be obtained when the J and K inputs are applied.

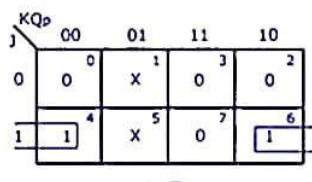
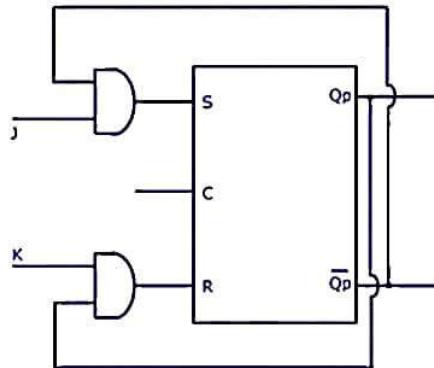
For two inputs J and K, there will be eight possible combinations. For each combination of J, K and Q_p , the corresponding Q_{p+1} states are found. Q_{p+1} simply suggests the future values to be obtained by the JK flip flop after the value of Q_p . The table is then completed by writing the values of S and R required getting each Q_{p+1} from the corresponding Q_p . That is, the values of S and R that are required to change the state of the flip flop from Q_p to Q_{p+1} are written.

S-R Flip Flop to J-K Flip Flop

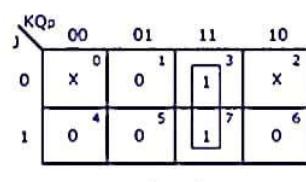
Conversion Table

J-K Inputs		Outputs		S-R Inputs	
J	K	Q _p	Q _{p+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Logic Diagram



K-Map

R = KQ_pJK Flip Flop to SR Flip Flop

This will be the reverse process of the above explained conversion. S and R will be the external inputs to J and K. As shown in the logic diagram below, J and K will be the outputs of the combinational circuit. Thus, the values of J and K have to be obtained in terms of S, R and Q_p. The logic diagram is shown below.

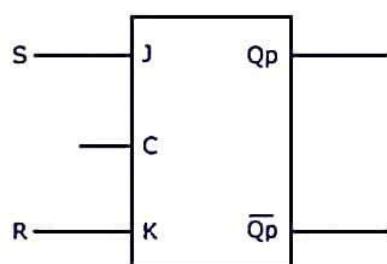
A conversion table is to be written using S, R, Q_p, Q_{p+1}, J and K. For two inputs, S and R, eight combinations are made. For each combination, the corresponding Q_{p+1} outputs are found out. The outputs for the combinations of S=1 and R=1 are not permitted for an SR flip flop. Thus the outputs are considered invalid and the J and K values are taken as "don't cares".

J-K Flip Flop to S-R Flip Flop

Conversion Table

S-R Inputs		Outputs		J-K Inputs	
S	R	Q _p	Q _{p+1}	J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	Invalid		Dont care	
1	1	Invalid		Dont care	

Logic Diagram



S	RQ _p			
	00	01	11	10
0	0	X	X	0
1	1	X	X	0

J=S

S	RQ _p			
	00	01	11	10
0	0	1	1	X
1	X	0	X	X

K-maps

K=R

SR Flip Flop to D Flip Flop

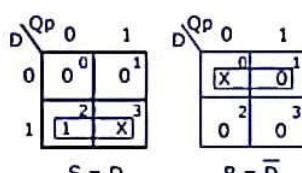
As shown in the figure, S and R are the actual inputs of the flip flop and D is the external input of the flip flop. The four combinations, the logic diagram, conversion table, and the K-map for S and R in terms of D and Q_p are shown below.

S-R Flip Flop to D Flip Flop

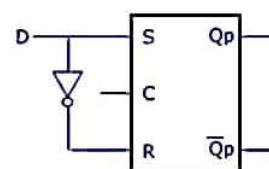
Conversion Table

D Input	Outputs		S-R Inputs	
	Q _p	Q _{p+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

K-maps



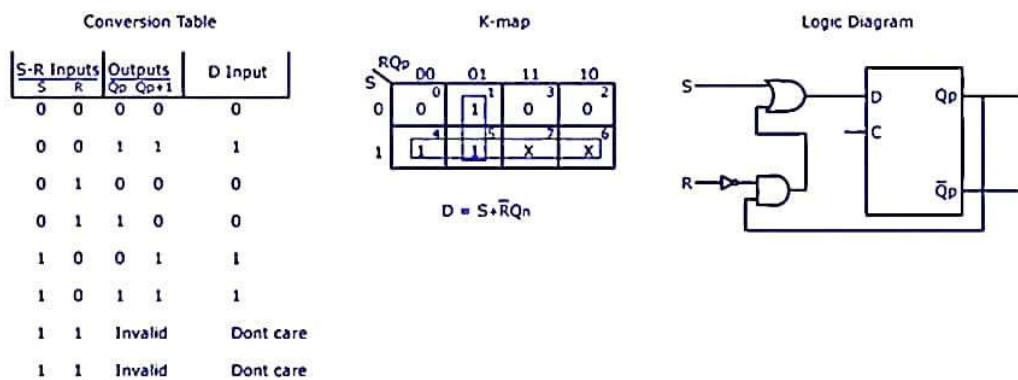
Logic Diagram



D Flip Flop to SR Flip Flop

D is the actual input of the flip flop and S and R are the external inputs. Eight possible combinations are achieved from the external inputs S, R and Q_p. But, since the combination of S=1 and R=1 are invalid, the values of Q_{p+1} and D are considered as “don’t cares”. The logic diagram showing the conversion from D to SR, and the K-map for D in terms of S, R and Q_p are shown below.

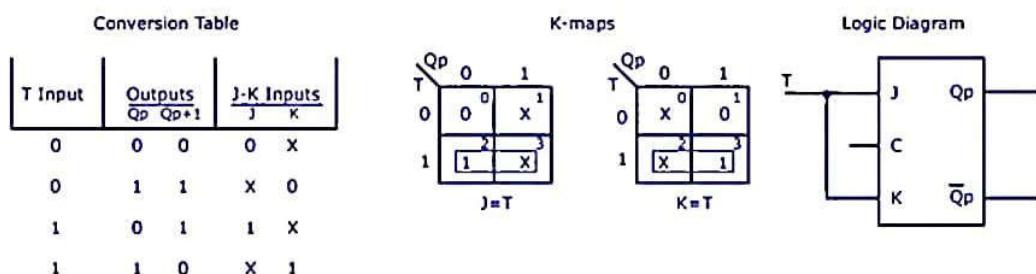
D Flip Flop to S-R Flip Flop



JK Flip Flop to T Flip Flop

J and K are the actual inputs of the flip flop and T is taken as the external input for conversion. Four combinations are produced with T and Q_p. J and K are expressed in terms of T and Q_p. The conversion table, K-maps, and the logic diagram are given below.

J-K Flip Flop to T Flip Flop



JK Flip Flop to D Flip Flop

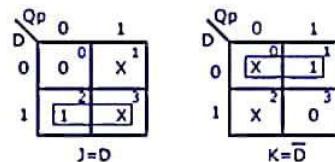
D is the external input and J and K are the actual inputs of the flip flop. D and Q_p make four combinations. J and K are expressed in terms of D and Q_p. The four combination conversion table, the K-maps for J and K in terms of D and Q_p, and the logic diagram showing the conversion from JK to D are given below.

J-K Flip Flop to D Flip Flop

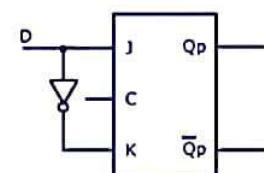
Conversion Table

D Input	Outputs		J-K Inputs	
	\bar{Q}_p	$Q_p + 1$	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	0	X	0

K-maps



Logic Diagram



D Flip Flop to JK Flip Flop

In this conversion, D is the actual input to the flip flop and J and K are the external inputs. J, K and Q_p make eight possible combinations, as shown in the conversion table below. D is expressed in terms of J, K and Q_p .

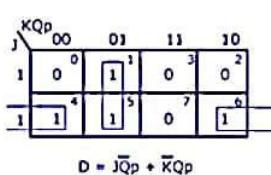
The conversion table, the K-map for D in terms of J, K and Q_p and the logic diagram showing the conversion from D to JK are given in the figure below.

D Flip Flop to J-K Flip Flop

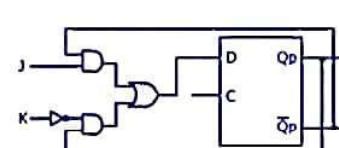
Conversion Table

J-K Input	Outputs		D Input
	\bar{Q}_p	$Q_p + 1$	
0 0	0 0	0	0
0 0	1 1	1	1
0 1	0 0	0	0
0 1	1 0	0	0
1 0	0 1	1	1
1 0	1 1	1	1
1 1	0 1	1	1
1 1	1 0	0	0

K-map



Logic Diagram



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.

- The master slave JK flip flop is a combination of a clocked JK latch and a clocked SR latch. The clocked JK latch acts as the master and the clocked SR latch acts as the slave.
 - Master is positive level triggered and due to the presence of an inverter in the clock line, the slave is negative level edge triggered. Hence when $\text{clock}=1$, the master is active and slave is inactive. Vice versa happens when $\text{clock}=0$.

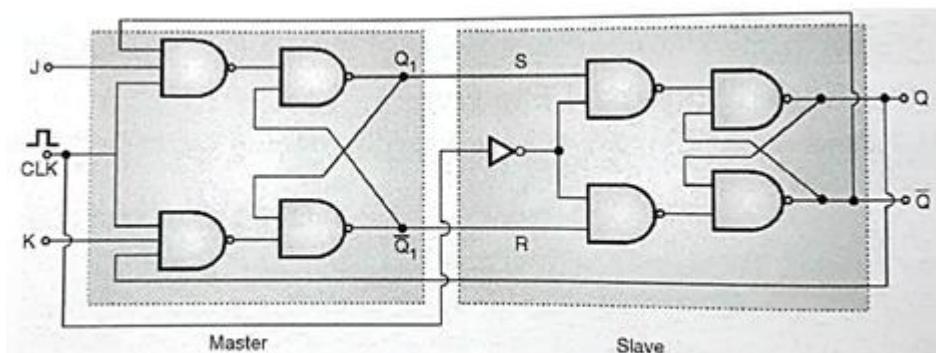


Fig6. Master slave JK FF

- The following is truth table of master slave flip flop.

Case	Inputs			Outputs		Remark
	CLK	J	K	Q_{n+1}	\bar{Q}_{n+1}	
I	x	0	0	Q_n	\bar{Q}_n	No change
II		0	0	Q_n	\bar{Q}_n	No change
III		0	1	0	1	Reset
IV		1	0	1	0	Set
V		1	1	\bar{Q}_n	Q_n	Toggle

Fig7. Truth table of Master slave JK FF

- Operation:

Case I: When clock is not given, both master and slave are inactive and there will be no change in outputs.

Case II: For clock=1, master is active, slave inactive. As J=K=0, output of master ie Q and Q' will not change. As soon as clock goes to 0, slave becomes active, and master inactive. But since input to slave S and R is same, output of slave will also remain same.

Case III: For clock=1, master is active and slave is inactive. When J=0 and K=1, outputs of master will be $Q=0$, $Q'=1$, which will be inputs to slave. When clock=0, slave becomes active and takes inputs 0,1 to give output $Q=0$, $Q'=1$. This output will not change if clock is again made 1 and then 0. Hence we get a stable output from master and slave.

Case IV: For clock=1, master is active and slave is inactive. When J=1 and K=0, outputs of master will be $Q=1$, $Q'=0$, which will be inputs to slave. When clock=0, slave becomes active and takes inputs 1,0 to give output $Q=1$, $Q'=0$. This output will not change if clock is again made 1 and then 0. Hence we get a stable output from master and slave.

Case V: When clock =1, $J=K=1$, master output will toggle. So S and R will invert. But slave remains inactive all this time since clock is 1. As soon as clock becomes 0, slave becomes active and master becomes inactive. So slave will also toggle. These changed outputs are returned through feedback to the master, but master does not respond to them because clock is now 0 and

master is inactive. Thus, in one clock period, master and slave both toggle only once, avoiding race condition caused by multiple toggling.

characteristic table & excitation table for SR flipflop

Truth table:

CLK	S	R	Q _{n+1}
0	X	X	Q _n
1	0	0	Q _n (Memory)
1	0	1	0
1	1	0	1
1	1	1	Invalid

characteristic table

clk=1

Q _n	S	R	Q _{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Excitation table

Q _n	Q _{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Q_{n+1} = next state

Q_n = present state

Assume clk=1
for all states

Find equation for Q_{n+1}

Q _n	SR	00	01	11	10
0	00	0	0	(X)	
1	10	D	0	X	1

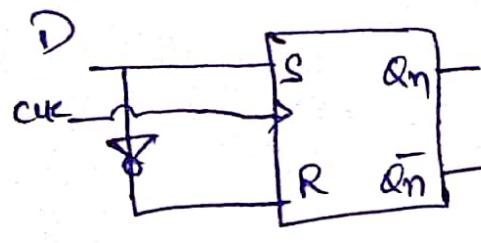
$$Q_{n+1} = S + Q_n \bar{R}$$

D - flopflip

Truth table for S.R flopflip:

CK	S	R	Q_{n+1}
0	x	x	Q_n
0	0	0	Q_n
0	1	0	0
1	0	1	1
1	1	1	Invalid

Q_n & Q_{n+1} memory



Truth table for D FF

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

Truth table

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

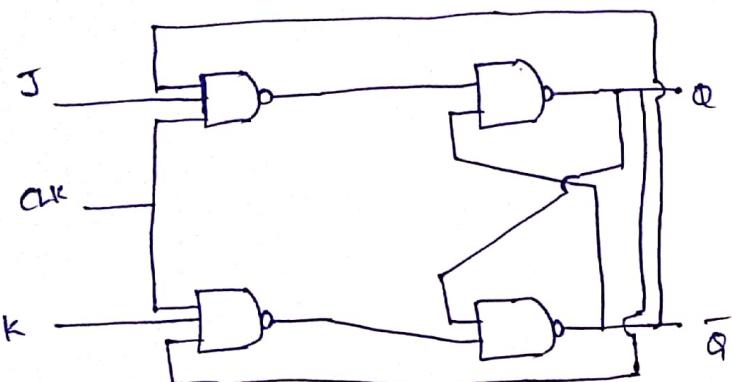
Characteristic table

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

Excitation table

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

J-K flip flop



CLK = 0 memory

CLK = 1, J=1, K=0, Q=1, $\bar{Q}=0$

→ form NAND
SR latch
we can say
easily T↑

CLK=1 J=0, K=1, Q=0, $\bar{Q}=1$

CLK=1 J=1, K=1 Q=0, $\bar{Q}=1$ (Assume)

J	K	Q	\bar{Q}
0	0	memory	
0	1	0	1
1	0	1	0
1	1		

Race around
(010101---)

T. table

CLK	J	K	Q _{out}
0	X	X	Q _{in} only memory
1	0	0	
1	0	1	0
1	1	0	1
1	1	1	Toggle (Q _{in})

Characteristic table

Q _{in}	J	K	Q _{out}
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Excitation table

Q _{in}	Q _{out}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

for J =

Q _{in}	Q _{out}
0	0
1	X

Q _{in}	Q _{out}
0	X
1	0

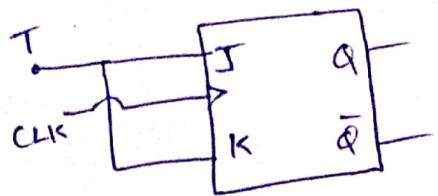
$$J = \overline{Q_{out}}$$

$$K = \overline{Q_{out}}$$

$$\begin{aligned} \text{for } Q_{out} \\ Q_{in} &= 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \\ Q_{out} &= 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \end{aligned}$$

$= \overline{Q_{in}} + \overline{Q_{in}}J$

T - flip flop (Toggle)



Truth table

CLK	T	Qntl
0	X	Qn
1	0	Qn
1	1	Qn (Toggle)

Truth table

CLK	T	Qntl
0	X	Qn
1	0	Qn
1	1	Qn (Toggle)

Characteristic table

Qn	T	Qntl
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table

Qn	Qntl	T
0	0	0
0	1	1
1	0	1
1	1	0