

## Sequential Circuits

### Flip-Flops

- The logic circuits whose outputs at any instant of time depend not only on the present inputs but also on the past outputs are called sequential circuits. In sequential circuits, the output signals are fed back to the input side. Thus, an output signal is a function of the present input signals and a sequence of the past input signals i.e. the past output signals.
- The block diagram of a sequential circuit is shown in fig. 1. It consists of ~~the~~ a combinational circuit to which memory elements are connected to form a feedback path.

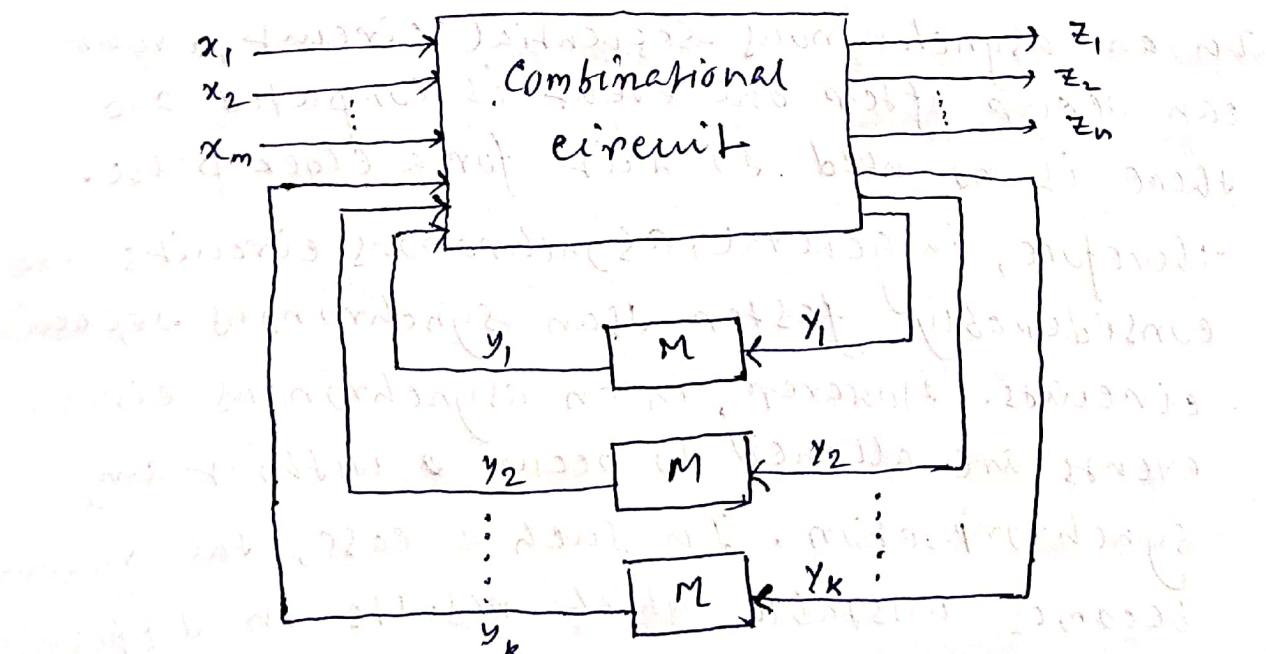


Fig 1: Block diagram of sequential circuit.

- The circuit has  $m$  number of external inputs, denoted by  $x_1, x_2, \dots, x_m$  and  $n$  number of outputs, represented by  $z_1, z_2, \dots, z_n$ . The memory elements denoted by  $M$  are devices capable of storing binary information within them. The signal values at the output of memory elements, denoted by  $y_1, y_2, \dots, y_k$ , are referred to as the present state or simply the state of the sequential circuit.
- Sequential circuits are of two types, viz.,  
(i) Synchronous or clocked and (ii) asynchronous or unclocked.

In the synchronous sequential circuit, synchronisation is achieved by a timing device called a master-clock generator, which generates a periodic train of clock pulses.

In an asynchronous sequential circuit, events can occur after one event is completed and there is no need to wait for a clock pulse.

Therefore, in general, asynchronous circuits are considerably faster than synchronous sequential circuits. However, in an asynchronous circuit, events are allowed to occur ~~at~~ without any synchronisation. In such a case, the system becomes unstable which results in difficulties.

- To have a sequential circuit, a storage device is required to know what has happened in the past. The basic unit of storage is the flip-flop.

□ Latches :-

- The simplest kind of a sequential circuit has only two states. It has a memory cell, which is capable of storing one bit of information, i.e. logic 1 or 0. This sequential circuit is also called a latch, since one bit of information can be locked or latched.
- The basic latch consists of two inverters as shown in fig 2.

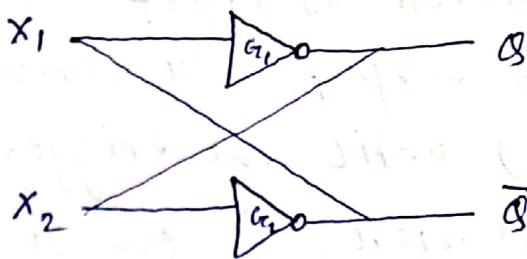


Fig 2: Basic latch - cross coupled inverters

- The output  $Q$  of inverter  $G_1$ , is connected to the input  $X_2$  of  $G_2$  and the output  $\bar{Q}$  of  $G_2$  is connected to the input  $X_1$  of  $G_1$ .
- Let us assume that the output of  $G_1$ , i.e.  $Q=1$ . Then, the output of  $G_2$  i.e.  $\bar{Q}=0$ , which is the complement of  $Q$ . similarly when  $Q=0$ ,  $\bar{Q}=1$ . Thus the output outputs  $Q$  and  $\bar{Q}$  are always complementary to each other.

Also, if the circuit is in state 1 or 0 at  $Q$  and  $\bar{Q}$  respectively, it continues to remain

latched in the same state. This property shows that it can store one bit of digital information. The basic latch shown in fig. 2 has no provision to get any desired digital information.

- The inverters  $G_1$  and  $G_2$  can be replaced with two input NAND/NOR gates and the second ~~gate~~ input of the gates can be used to enter the ~~the~~ digital information.

When the latch output  $Q=0$  or  $1$ , it will remain in the same state until one or more of the inputs are excited to effect a change in the output. Since the latch output will remain set (i.e.  $Q=1$ ) / reset (i.e.  $Q=0$ ) until the trigger pulse is given to change the state, it can be regarded as a memory device with the capability of storing one binary digit of information.

#### □ NOR based S-R Latch:

- The S-R latch has two inputs, namely SET(S) and RESET(R), and two outputs  $Q$  and  $\bar{Q}$ . The outputs are complemented to each other. The S-R latch can be easily implemented using NOR gates or NAND gates.
- NOR based S-R latch is shown in fig. 3. The cross-coupled connections from the output of one gate to the inputs of the other gate constitute a feedback path. For this reason, the circuit

are classified as ~~syn~~ asynchronous sequential circuits. The truth table for the NOR-based S-R latch is shown in fig. table 1.

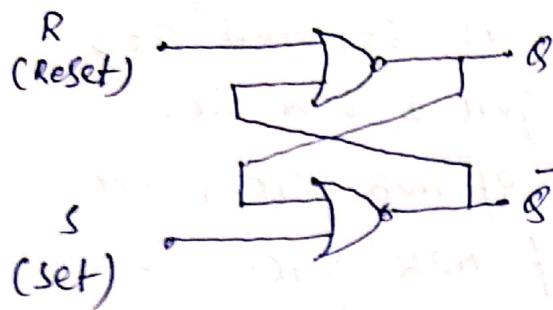


Fig. 3: NOR-based S-R latch

Table 1: Truth table of NOR-based S-R latch

Inputs		Outputs		Action
S	R	$Q_{n+1}$	$\bar{Q}_{n+1}$	
0	0	$Q_n$	$\bar{Q}_n$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Forbidden

→ To analyse the circuit of Fig. 3, one must remember that the output of a NOR gate is 0 if any input is 1 and the output is 1 only when all inputs are 0. From the truth table, the outputs for these four possible input combinations are described below.

Case 1: For  $S=0$  and  $R=0$ , the latch simply remains in its present state ( $Q_n$ ). i.e. the next state of the latch will be  $Q_{n+1}=0$  if  $Q_n=0$  and  $Q_{n+1}=1$  if  $Q_n=1$ .

Case 2: The second input condition is  $S=0$  and  $R=1$ . The 1 at the RESET input forces the of NOR gate-1 LOW (i.e.  $Q_{n+1}=0$ ). Now both the inputs of NOR gate-2

are 0 and the output is 1 ( $Q_{n+1} = 1$ ).

Thus the input condition  $S=0$  and  $R=1$  will always 'Reset' the latch to 0. When the reset input returns to 0, the latch will remain in the 0 state.

case 3: The third input condition is  $S=1$  and  $R=0$ , which forces the output of NOR gate-2 LOW, i.e.

$\bar{Q}_{n+1} = 0$ . Now, both the inputs of NOR gate-1 are 0 and therefore the output of NOR gate-1 is HIGH, i.e.  $Q_{n+1} = 1$ . Hence, the condition  $S=1$  and  $R=0$  will always set the latch to 1.

case 4: The input condition is  $S=1$  and  $R=1$ . This condition will provide 0 at the output of both the NOR gates. Hence,  $Q_{n+1} = 0$  and  $\bar{Q}_{n+1} = 0$ .

This condition violates the fact that the outputs  $Q_{n+1}$  and  $\bar{Q}_{n+1}$  are the complements of each other.

In normal operation, this condition must be avoided by making sure that 1s are not applied to both inputs simultaneously.

D State diagram and characteristic equation of S-R latch:

→ The state transition diagram for S-R latch can be drawn as shown in Fig 4.

From the state diagram, one can easily understand that an S-R latch is with two stable states 0 and 1. On applying SET input; the latch is set to 1; on applying RESET input, the latch is reset to 0 irrespective of its previous state.

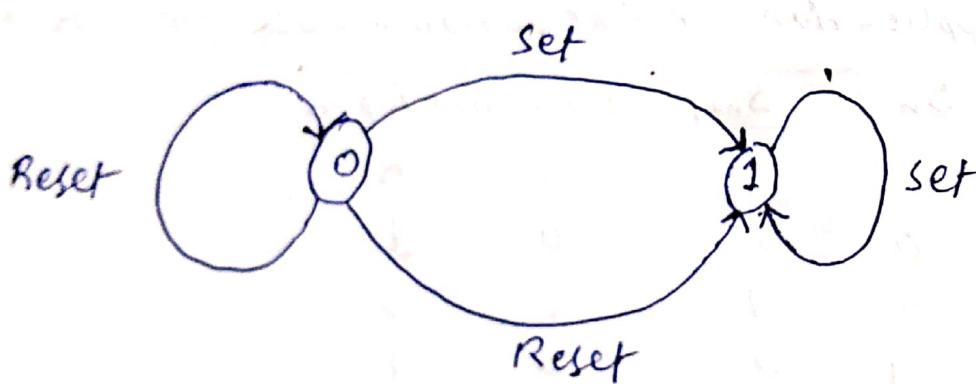


Fig. 4: State diagram of S-R latch

→ From the above state diagram and truth table of S-R latch, the present-state ~~and~~ next-state table and application table or excitation table for S-R latch can be drawn as shown in Table-2 and Table-3 respectively.

Table 3: Present state - next state for S-R latch

present state $Q_n$	SET input S	RESET input R	Next state $Q_{n+1}$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	d
1	0	1	0
1	1	0	1
1	1	1	d

$d \rightarrow$  don't care

Table 4: Application or excitation table for S-R latch

$Q_n$	$Q_{n+1}$	Excitation inputs	
		S	R
0	0	0	d
0	1	1	0
1	0	d	1
1	1	d	0

→ Using the above present state - next state table of S-R latch, a K-map for the next state transition ( $Q_{n+1}$ ) can be drawn as shown in Table 5, and a logic expression that describes the operation of the S-R latch can be found. Such a logic expression is called the characteristic equation of the S-R latch.

$Q_n \backslash S\bar{R}$	00	01	11	10
0	0	0	d	L
1	1	0	d	L

Table 5: Next state  
( $Q_{n+1}$ ) map for  
S-R latch

From the above K-map,  $Q_{n+1} = S\bar{R} + \bar{R}Q_n$

$$\text{or } Q_{n+1} = (S + Q_n)\bar{R}$$

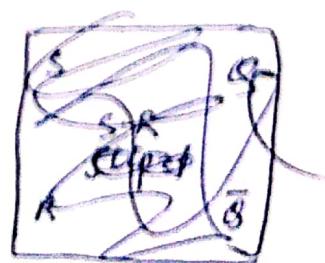
- In the above K-map simplification, the don't care combinations are not considered because they correspond to invalid input combinations of S-R latch.

## Flip - Flips:

- Synchronous circuits changes their ~~stage~~ states only when clock pulses are present.
- The operation of the basic latch can be modified, by providing an additional control input that determines, when the state of the circuit is to be changed.
- The latch with the additional control input is called the flip-flop. The additional control input is either the clock or enable input.
- Flip-flops are of different types depending on how their inputs and clock pulses cause transition between two states. There are four basic types namely, S-R, J-K, D and T flip-flops.

### S-R flip-flop:

- The S-R flip-flop consists of two additional AND gates at the S and R inputs of S-R latch as shown in fig. 5.



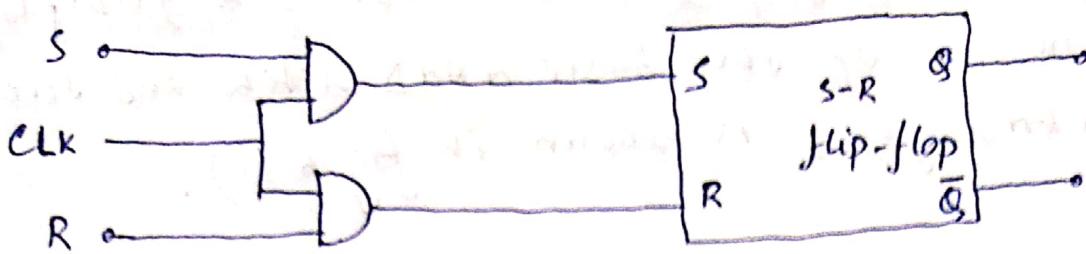


Fig. 5: Block diagram of S-R flip-flop.

- In this circuit, when the clock input is 'low', the output of both the AND gates are 'Low' and the changes in S and R inputs will not affect the output ( $Q$ ) of the flip-flop.  
When the clock input becomes 'high', the value at S and R inputs will be passed to the outputs of the AND gates and the outputs ( $Q$ ) of the flip-flop will change according the changes in S and R inputs as long as the clock input is 'HIGH'.
- In this manner, one can clock the flip-flop so as to store either a '1' by applying  $S=1, R=0$  (i.e. set) or a '0' by applying  $S=0, R=1$  (reset) at any time and then hold that bit of information for any desired period of time by applying a 'low' at the clock input.  
This flip-flop is called clocked S-R flip flop.

→ NAND based S-R flip-flop: The S-R flip-flop which consists of the basic NAND latch and two other NAND gates is shown in fig 6 (a).

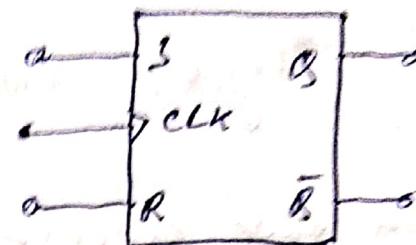
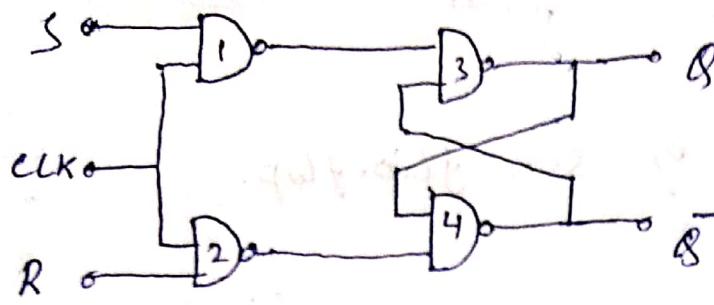


Fig 6(a): NAND-based S-R flip flop (b) graphic symbol

→ The S and R inputs control the same state of the flip-flop in the same manner as described earlier for the basic (unlocked) S-R latch.

However, the flip-flop does not respond to these inputs until the rising edge of the clock signal occurs.

The clock pulse input acts as an enable signal for the other two inputs. The outputs of NAND gates 1 and 2 stay at the logic '1' level as long as the clock input remains at 0. This 1 level of the inputs of NAND-based basic S-R latch retains the present state, i.e. no change occurs.

→ The characteristic table of the S-R flip-flop is shown in truth table of table 6. This table shows the operation of the flip-flop in tabular form.

Table 6: Characteristic table of S-R flip-flop

Present state $Q_n$	clock pulse $clk$	Data inputs $S$ & $R$	Next state $Q_{n+1}$	Action
0	0	0 0	0	No change
1	0	0 0	1	No change
0	1	0 0	0	No change
1	1	0 0	1	No change
0	0	0 1	0	No change
1	0	0 1	1	No change
0	1	0 1	0	Reset
1	1	0 1	0	Reset
0	0	1 0	0	No change
1	0	1 0	1	No change
0	1	1 0	1	Set
1	1	1 0	1	Set
0	0	1 1	0	No change
1	0	1 1	1	No change
0	1	1 1	2	Forbidden
1	1	1 1	2	Forbidden

case 1: For  $S=0$ ,  $R=0$  and  $clk=0$ , the flip-flop simply remains in its present state. That is,  $Q$  remains unchanged. Even for  $S=0$ ,  $R=0$  and  $clk=1$ , the flip-flop remains in its present state. The first four rows of the truth table clearly indicate that the state of the flip-flop remains unchanged, i.e.  $Q_{n+1} = Q_n$ .

case 2: For  $S=0$ ,  $R=1$  and  $CLK=0$ , the flip-flop remains in its present state. But, when  $CLK=1$ , the NAND gate-1 output will go to 1 and the NAND gate-2 output will go to 0. Now a 0 at NAND gate-4 input forces  $\bar{Q}=1$  which in turn ~~sets~~ results in NAND gate-3 output  $Q=0$ . Thus, for  $S=0$ ,  $R=1$  and  $CLK=1$ , the flip-flop RESETS to the 0 state.

case 3: For  $S=1$ ,  $R=0$  and  $CLK=0$ , the flip-flop remains in its present state. But for  $S=1$ ,  $R=0$  and  $CLK=1$ , the set state of the flip-flop is reached. This causes the NAND gate-1 output to go to 0 and the NAND gate-2 output to 1. Now, a 0 at NAND gate-3 input forces  $Q$  to 1 which in turn forces NAND gate-4 output  $\bar{Q}$  to 0.

case 4: An indeterminate condition occurs when all the inputs, namely  $CLK$ ,  $S$  and  $R$ , are equal to 1. This condition results in 0s in the outputs of gate-1 and 2 and 1s in both outputs  $Q$  and  $\bar{Q}$ . When the clock input  $CLK$  goes back to 0 (while  $S$  and  $R$  remain at 1), it is not possible to determine the next state, as it depends on whether the output of gate-1 or gate-2 goes to 1 first.

- ②
- D Flip-Flop:
- The D (delay) flip-flop has only one input called the Delay (D) input and two outputs Q and  $\bar{Q}$ . It can be constructed from an S-R flip-flop by inserting an inverter between S and R and assigning the symbol D to the S input.
- The structure of D flip-flop is shown in fig. 7(a). Basically, it consists of a. NAND flip-flop with a gating arrangement on its inputs. It operates as follows:

1. When the CLK input is 'Low', the D input has no effect, since the set and reset inputs of the NAND flip-flop are kept 'HIGH'.

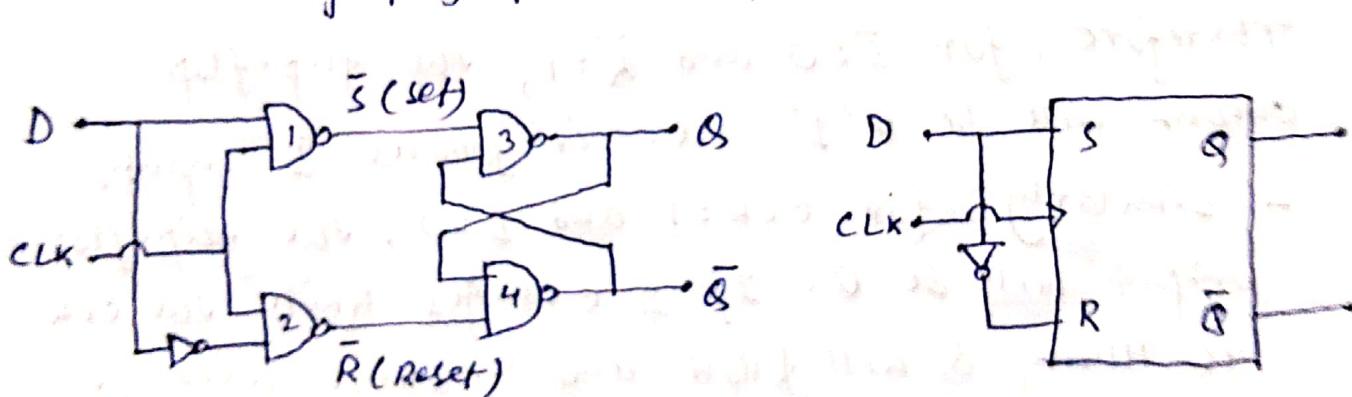
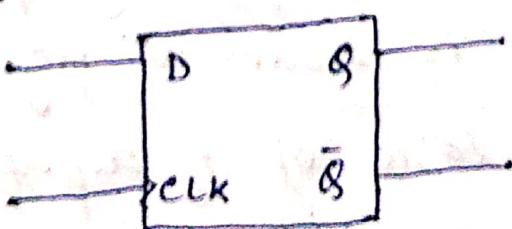


Fig. 7(a): D flip-flop using  
NAND gates      (b) Using S-R flip-flop



(c) Logic symbol.

2. When the CLK goes HIGH, the Q output will take on the value of the D input.
- If  $\text{CLK}=1$  and  $D=1$ , the NAND gate-1 output goes 0 which is the  $\bar{S}$  input of the basic NAND-based S-R flip-flop and NAND gate-2 output goes 1 which is the  $\bar{R}$  input of the basic NAND-based S-R flip-flop. Truth table of NAND-based  $\bar{S}-\bar{R}$  latch shown below.

Table: Truth table of NAND-based S-R latch

Inputs		Outputs		Action
$\bar{S}$	$\bar{R}$	$Q_{n+1}$	$\bar{Q}_{n+1}$	
0	0	?	?	Forbidden
0	1	1	0	Set
	0	0	1	Reset
1	1	$Q_n$	$\bar{Q}_n$	No change

D flip-flop  
action on  
 $\bar{S}-\bar{R}$  latch

Therefore, for  $\bar{S}=0$  and  $\bar{R}=1$ , the flip-flop output will be  $Q=1$  i.e. it follows D input.

→ Similarly, for  $\text{CLK}=1$  and  $D=0$ , the flip-flop output will be 0. If D changes while the CLK is HIGH, Q will follow and change quickly.

→ The logic symbol for the D flip-flop is shown in fig 7(c). A simple way of building a delay D flip-flop is shown in fig. 7(b).

→ The truth table of D flip-flop is given in table 7. from which it is clear that the

Table 7: Truth table of D flip-flop

CLK	Input D	Output $Q_{n+1}$
1	0	0
1	1	1
0	X	NO change

→ From the truth table it is clear that the next state of the flip-flop at time ( $Q_{n+1}$ ) follows the value of the input D when the clock pulse is applied.

As transfer of data from the input to the output is delayed, it is known as delay (D) flip-flop. The D-type flip-flop is either used as a delay device or as a latch to store 1 bit of binary information.

D state diagram and characteristic equation of D flip-flop:-

→ The state diagram for the D flip-flop is shown in fig 8.

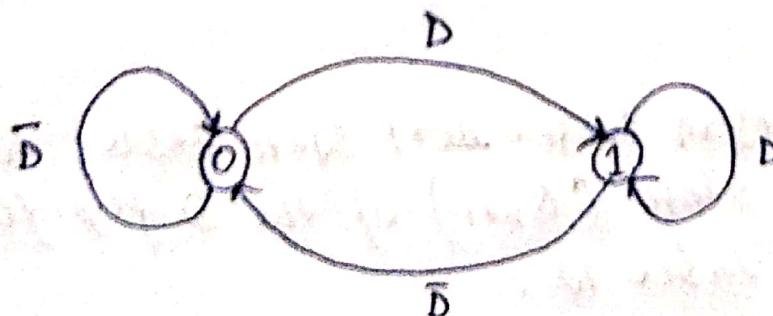


Fig 8: State diagram of delay flip-flop.

From the above state diagram, it is clear that when  $D=1$ , the next state will be 1; when  $D=0$ , the next state will be 0, irrespective of its previous state. From the state diagram, one can draw the present state-next state table and the application or excitation table for the delay flip-flop as shown in Table 8 and Table 9, respectively.

Table 8: Present state-next state table for D flip-flop

Present state $Q_n$	Delay input $D$	Next state $Q_{n+1}$
0	0	0
0	1	1
1	0	0
1	1	1

Table 9: Application or excitation table for D flip-flop

$Q_n$	$Q_{n+1}$	Excitation input $D$
0	0	0
0	1	1
1	0	0
1	1	1

Using the Present state-Next State table, the k-map for the next state ( $Q_{n+1}$ ) of the D flip-flop can be drawn as in Table 10.

Table 10: next state ( $Q_{n+1}$ ) map for D flip-flop

$S_n$	0	1
0	0	1
1	0	1

The simplified expression for  $Q_{n+1}$  can be found as follows.

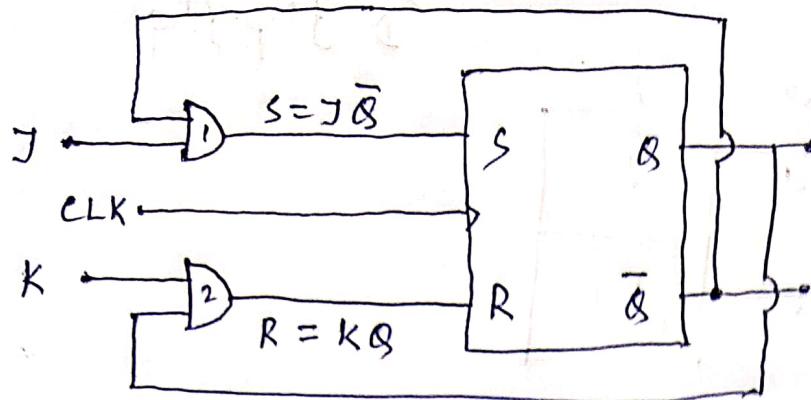
$$Q_{n+1} = D$$

This is the characteristic equation for Delay flip-flop. Hence, in a delay flip-flop, the next state follows the delay input as shown by the characteristic equation.

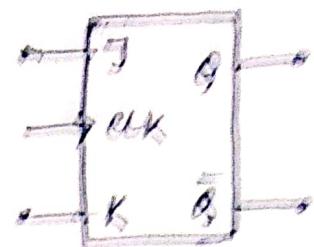
### J-K flip-Flop

→ A J-K flip-flop has a characteristic similar to that of an S-R flip-flop. In addition, the intermediate indeterminate condition of the S-R flip-flop is permitted in it.

→ A J-K flip-flop can be obtained from the clocked S-R flip-flop by augmenting two AND gates as shown in Fig 9. The data input  $J$  and the output  $\bar{Q}$  are applied to the first AND gate, and its output ( $J\bar{Q}$ ) is applied to the S input of S-R flip-flop.



(a)



(b)

Fig 9(a): J-K flip-flop using S-R flip-flop      (b) Graphic symbol of J-K flip-flop.

→ Similarly, the data input K and the output Q are connected to the ~~J input~~ second AND gate and its output ( $K\bar{Q}$ ) is applied to R input of S-R flip-flop.

The graphic symbol of J-K flip-flop is shown in fig 9(b) and the truth table is shown in table 11. The output for the four possible input sequences are described below.

Table 11: Truth table of J-K flip-flop

CLK	Inputs		Output $Q_{n+1}$	Action
	J	K		
X	0	0	Q_n	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	$\bar{Q}_n$	Toggle

Case 1: When  $J=0$ , whatever be the value of  $Q_n$  (0 or 1), the output of the first AND gate is 0. Similarly, when  $K=0$ , whatever be the value of  $Q_n$  (0 or 1), the output of the second AND gate is also 0.

Therefore, when  $J=0$  and  $K=0$ , the inputs to the basic flip-flop are  $S=0$  and  $R=0$ . This condition forces the flip-flop into the same state.

Case 2: When  $J=0$ ,  $K=1$  and previous state of flip-flop is SET (i.e.  $Q_n=1$ ), then  $S=J\bar{Q}_n=0 \times 0=0$  and  $R=KQ_n=1 \times 1=1$ . Since  $S=0$  and  $R=1$ , the flip-flop RESETS on the application of a clock pulse. But if the flip-flop is already RESET (i.e.  $Q_n=0$  and  $\bar{Q}_n=1$ ), then the application of  $J=0$  and  $K=1$  will not alter the state of the flip-flop ( $S=J\bar{Q}_n=0 \times 1=0$  and  $R=KQ_n=1 \times 0=0$ ), and it remains in the RESET state.

Case 3: When  $J=1$ ,  $K=0$  and the previous state of the flip-flop is RESET ( $Q_n=0$  and  $\bar{Q}_n=1$ ), then  $S=J\bar{Q}_n=1 \times 1=1$  and  $R=KQ_n=0 \times 0=0$ . Since flip-flop is already SET (i.e.  $Q_n=1$  and  $\bar{Q}_n=0$ ). Then the inputs  $J=1$ ,  $K=0$  will make  $S=0$  and  $R=0$ , and the flip-flop remains in the SET state.

Case 4: The condition  $J=1$ ,  $K=1$  deserves special attention, when  $J=1$ ,  $K=1$  and the previous state

is a SET state (i.e.  $Q_n=1$ ,  $\bar{Q}_n=0$ ), then  $S=J\bar{Q}_n$   
 $= 1 \times 0 = 0$  and  $R=KQ_n=1 \times 1 = 1$ . Since  $S=0$  and  
 $R=1$ , the flip-flop RESETS on the application of  
a clock pulse, i.e. the flip-flop toggles from  
SET to RESET State.

Again for  $J=1, K=1$ , if the previous state is a

RESET state (i.e.  $Q_n=0$  and  $\bar{Q}_n=1$ ), then  $S=J\bar{Q}_n$   
 $= 1 \times 1 = 1$  and  $R=KQ_n=1 \times 0 = 0$ . Since  $S=1$  and

$R=0$ , the flip-flop on the application of a clock  
pulse will toggle to the SET state. Hence, when

$J=1$  and  $K=1$ , the flip-flop toggles on the  
application of the next clock pulse. The flip-flop  
will continuously change its state when  $J=K=1$   
and  $\text{clock} = \text{HIGH}$ , resulting in an unstable output.

Toggle means to switch to the opposite state.

- State diagram and characteristic equation of J-K flip-flop :

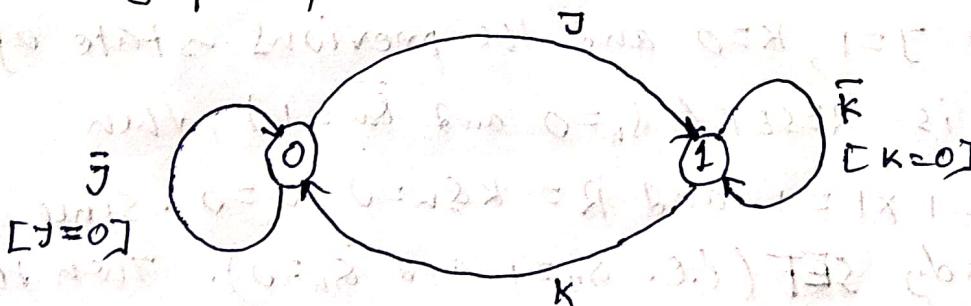


Fig 10 : State diagram of J-K flip-flop.

- From the above state diagram, one can easily understand that the state transition from 0 to 1 takes place whenever J is asserted (i.e.  $J=1$ ) irrespective of K value.
- Similarly, State transition from 1 to 0 takes place whenever K is asserted (i.e.  $K=1$ ) irrespective of the value of J. Also, the state transition from 0 to 0 occurs whenever  $J=0$ , irrespective of the value of K, and the state transition from 1 to 1 occurs whenever  $K=0$ , irrespective of J value.
- From the above state diagram and truth table (Table 11) of J-K flip-flop, the present state-next state table and application table or excitation table for J-K flip-flop as shown in table 12 and table 13, respectively.

Table 12: Present State - next state table for J-K flip-flop

Present state $Q_n$	Inputs		Next state $Q_{n+1}$
	J	K	
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

Table 13: Application or excitation table for J-K flip-flop

$Q_n$	$Q_{n+1}$	Excitation inputs	
		J	K
0	0	0	1
0	1	1	0
1	0	0	1
1	1	1	0

→ From table 12, a K-map for the next state transition ( $Q_{n+1}$ ) can be drawn as shown in table 14.

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

The characteristic equation of J-K flip-flop can be written as

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

Inputs	1	0	1	0	1	0	1
1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1
0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	1
0	0	1	0	1	0	1	0
1	1	1	0	0	1	1	0