

- 1 T flip-flop:
- Another basic flip-flop, called the T or trigger or toggle flip-flop, has only a single data (T) input, a clock input and two outputs Q and \bar{Q} .
 - The T-type flip-flop is obtained from a J-K flip-flop by connecting its J and K inputs together. The designation T comes from the ability of the flip-flop to "toggle" or complement its state.
 - The block diagram of a T flip-flop and its circuit implementation using a J-K flip-flop are shown in fig 11. The J and K inputs are wired together. The truth-table for T flip-flop is shown in table 14.

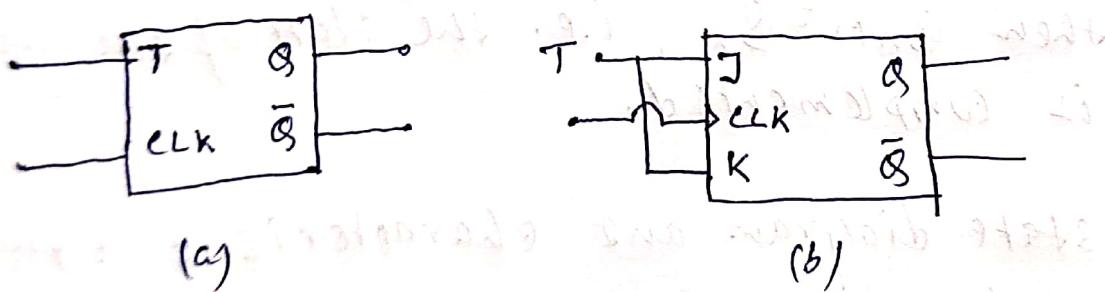


Fig 11 (a) Block diagram of T flip-flop (b) T flip-flop using a J-K flip-flop

Table 14: Truth Table of T flip-flop

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

- When the T input is in the 0 state (i.e. $J=K=0$) prior to a clock pulse pulse, the Q output will not change with clocking.
- When the T input is at a 1 (i.e. $J=K=1$) level prior to clocking, the output will be in the \bar{Q} state after clocking.
- In other words, if the T input is a logical 1 and the device is clocked, the output will change state regardless of what output was prior to clocking. This is called toggling hence the name T flip-flop.
- The truth table shows that when $T=0$, then $Q_{n+1} = Q_n$, i.e. the next state is the same as the present state and no change occurs. When $T=1$, then $Q_{n+1} = \bar{Q}_n$, i.e. the state of the flip-flop is complemented.

□ State diagram and characteristic equation of T flip-flop:

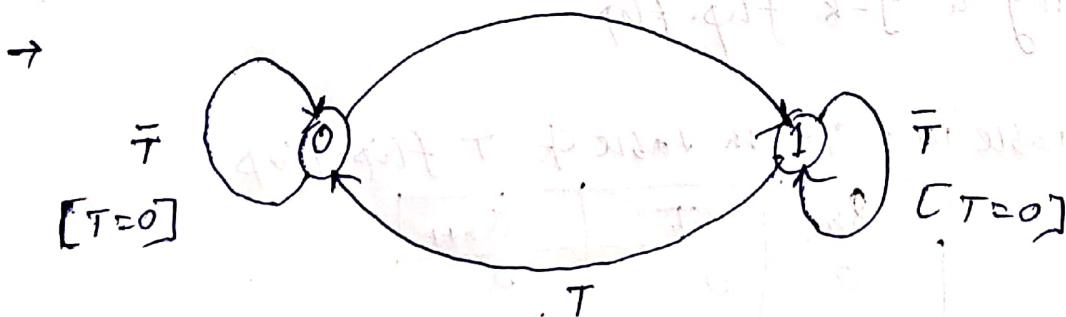


Fig 12: State diagram of trigger flip-flop

→ From the above state diagram, it is clear that when $T=1$, the flip-flop changes or toggles its state irrespective of its previous state; when $T=1$ and $Q_n = 0$, the next state will be 1; when $T=1$ and $Q_n = 1$, the next state will be 0.

Similarly, one can understand that when $T=0$, the flip-flop retains its previous state. From the above state diagram, one can draw the present state - next state table and application table for the T flip-flop as shown in Table 15 and Table 16, respectively.

Table 15: Present State - Next State Table

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

Table 16: Application or excitation table for T flip-flop

Q_n	Q_{n+1}	Excitation i/p (T)
0	0	0
0	1	1
1	0	1
1	1	0

- From the Table 15, the K-map for the next state (Q_{n+1}) of a trigger flip-flop can be drawn as shown below:
- | | | |
|-----------|-------------|-------|
| | \bar{Q}_n | Q_n |
| \bar{T} | 0 | 1 |
| T | 0 | 1 |
- So, $Q_{n+1} = \bar{T}\bar{Q}_n + TQ_n$.
- so, in a trigger flip-flop, the next state will be the complement of the previous state when ~~$T=1$~~ $T=1$.

□ Application of T flip-flop:

- T-type flip-flop is most often seen in counters and sequential counting networks because of its inherent divide-by-2 capability. When a clock pulse is applied, the output changes state once every input cycle, thus repeating one cycle for every two input cycles.

This is the action required in many cases for binary counters.

Triggering of flip-flops:

- Flip-flops are synchronous bistable devices. The term synchronous means that the changes in the output occur at a specified point on a triggering input called called the clock; that is, changes in the output occur in synchronisation (in time) with the clock.
- Based on the specific interval or point in the clock during or at which triggering of flip-flop takes place, it can be classified into two different types.
 - Level triggering
 - Edge triggering

A clock pulse starts from an initial value of 0, goes momentarily to 1, and after a short time, returns to its initial 0 value.

(i) Level triggering in flip-flops:

- When the clock pulse goes HIGH, ~~and~~ the flip-flop is said to be level triggered flip-flop. Since the flip-flop changes its state when the clock is positive, it is termed as positive ~~and the input of AND gate~~ level triggered flip-flop.

If a NOT gate is introduced in between the clock input and the input of AND gate (Fig. 5, for example) of S-R flip-flop, the flip-flop changes its state only when the clock is negative (i.e. when clock = low), so it is called negative level triggered flip-flop.

→ The main drawback of level triggering is that, as long as the clock is positive or negative, the flip-flop changes its state more than once or many times for each change in inputs.

On other hand, if the frequency of input change is higher than the input clock frequency, the output of the flip-flop undergoes multiple changes when the clock is positive or negative.

This can be overcome in Master-slave flip-flops and Edge-triggered flip-flops where the flip-flops change state only once for a clock.

(ii) Edge Triggering in flip-flops:

→ A clock pulse goes through two signal transitions from 0 to 1 and returns from 1 to 0. As shown in fig. 13, a positive transition is defined as the positive edge and a negative transition as the negative edge. This definition applies also to negative pulses.

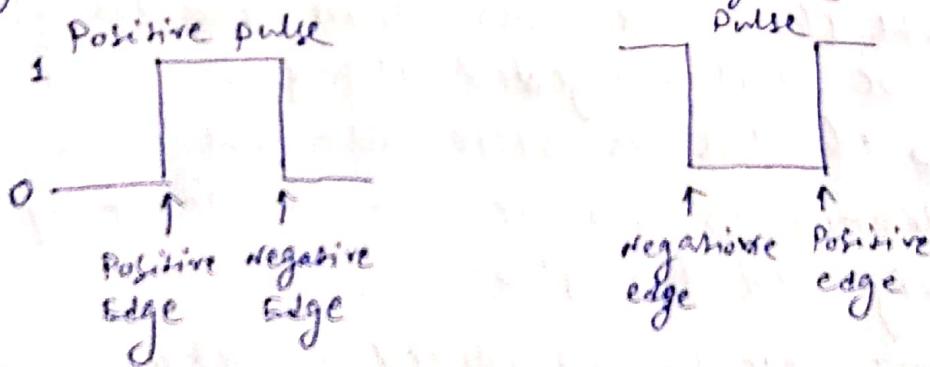


Fig. 13 : Definition of clock pulse transition

→ The term edge-triggered means that the flip-flop changes its state either at the positive edge (rising or leading edge) or at the negative edge.

(falling or trailing edge) of the clock pulse and is sensitive to its inputs only at this transition of the clock.

- One way to make the flip-flop respond only to a pulse transition is to use capacitive coupling.

An R-C (resistor-capacitor) circuit as shown in fig 14 must be inserted in the clock input of the flip-flop. By deliberate design, the ~~RC time~~ RC time constant is much smaller than the clock's pulse width. Because of this, the capacitor can charge fully when the clock goes high; this exponential charging produces a narrow positive voltage spike across the ~~capacitor~~ resistor. Later, the trailing edge of the pulse results in a narrow negative spike. The circuit that generates a spike in response to a momentary change of input signal is called R-C differentiator.

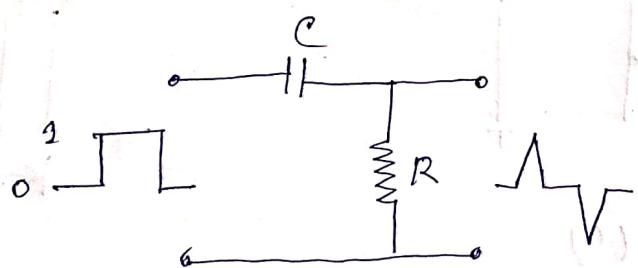


Fig. 14 : R-C differentiator circuit.

D Master-slave flip-flops:

The Race-around condition:

- The timing problem in level-triggered flip-flops is often handled by master-slave flip-flop.
- The condition $S=1$ and $R=1$ we have seen earlier is not allowed in an S-R flip-flop. This is eliminated in the J-K flip-flop by using the feedback connection. Because of the feedback connection ~~from~~ Q (\bar{Q}) at the input to $K(J)$, the inputs will change during the clock pulse ($CLK=1$), if the output changes to state.

Consider, for example, that the inputs are $J=K=1$ and $Q=0$. When a clock pulse of width t_p , as shown in fig. 15(a) is applied, the output will change from 0 to 1 after a time interval Δt , where Δt is the propagation delay through two NAND gates in series.

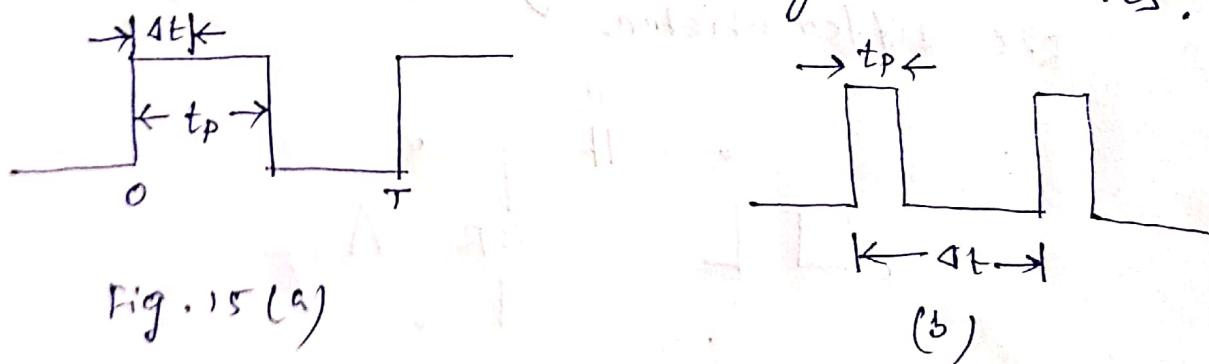


Fig. 15 (a)

(b)

Now, after time Δt , we have $J=K=1$ and $Q=1$ and after another interval of Δt , output Q will become 0. Hence, the output will oscillate back and forth between 0 and 1 in the duration t_p of the clock pulse width. So, at the end of the clock pulse, the value of Q is ambiguous. This situation is known as a race-around condition.

- The race-around condition can be avoided if ~~$t_p < t_f$~~ as shown in fig 15(b).
- A master-slave flip-flop is constructed using two separate flip-flops connected serially. The first flip-flop serves as the master and the second as a slave, and the overall circuit is referred to as a Master-Slave flip-flop.

□ Master-Slave J-K flip-flop:

- A master-slave flip-flop can be constructed using two J-K flip-flops as shown in fig. 16. The first flip-flop, called the master, is driven by the positive edge of the clock pulse; the second flip-flop, called the slave, is driven by the negative edge of the clock pulse. Therefore, when the clock input has a positive edge, the master acts according to its J-K inputs, but the slave does not respond since it requires a negative edge at the clock input.

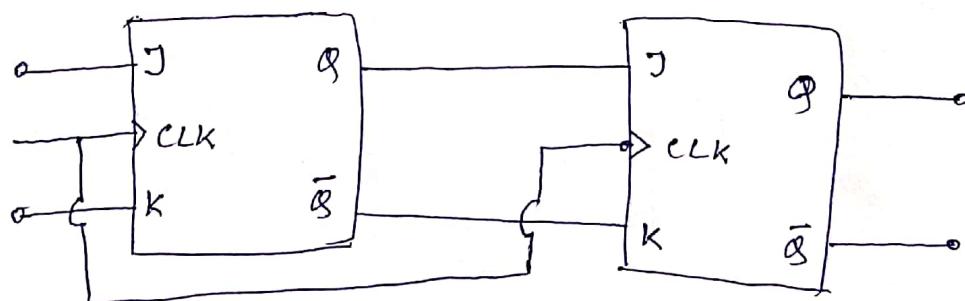


Fig. 16 : Master-Slave J-K flip-flop .

- When the clock input has a negative edge, the slave flip-flop copies the master outputs. But the master

does not respond to the feedback from S and \bar{S} . Since it requires a positive edge at its clock input. Thus, the master-slave flip-flop does not have race-around problem.

- Master-slave flip-flops operate from a complete clock-pulse, and the outputs change on the negative transition.

- Realization of one flip-flop using other flip-flops:
- It is possible to implement a flip-flop circuit using any other flip-flop. The general block diagram connection for such realisation is shown in fig. 17.

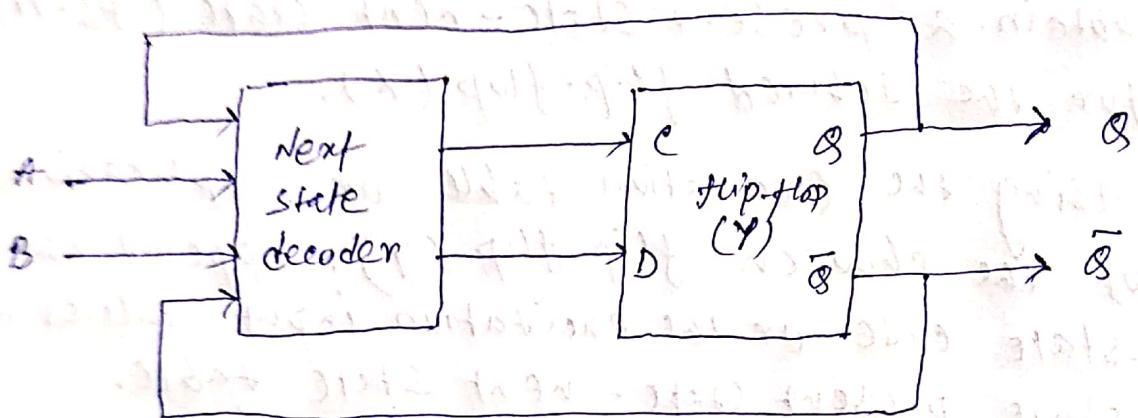


Fig. 17: Block diagram to realise flipflop (X) using flip-flop (Y)

- To realise a flip-flop (X) using another flip-flop (Y), the flip-flop (Y) along with a combinational circuit, called 'next state decoder', is used which functions like flip-flop (X). For this to be realised, for each set of inputs of flip-flop (X), i.e. A & B and present state (Q_n), we have to find the inputs to the flip-flop (Y), i.e. C & D, that will cause the flip-flop (Y) to make transition into the proper next state (Q_{n+1}) of the flip-flop (X). These inputs to flip-flop (Y) are called the next state code.

→ Realisation The design procedure for such transition realisation can be summarised in the following steps:

- (i) obtain a clear word description of the desired flip-flop (X).
- (ii) obtain a present state - next state ($PS-NS$) table for the desired flip-flop (X).
- (iii) Using the excitation table or application table of the chosen flip-flop (Y), appends the next state code or the excitation input values to the above present state - next state table.
- (iv) Using K-maps, simplify the logic expressions for excitation inputs of flip-flop (Y) and design the next state decoder logic.
- (v) Draw a circuit for the desired flip-flop (X) using next state decoder logic and chosen flip-flop (Y) as shown in the block diagram.

□ Realisation of D flip-flop using S-R flip-flop:

→ Here, the desired flip-flop (X) is Delay flip-flop and the chosen flip-flop (Y) is S-R flip-flop. The basic block diagram is shown in fig. 17.

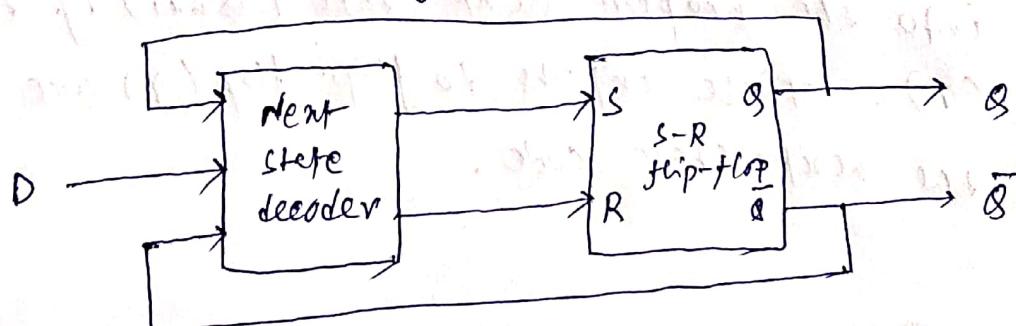


Fig. 17: Block diagram of a D flip-flop using S-R flip-flop

Step 1: ~ the operation of Delay flip-flop is well known, and hence there is no need for word description.

Step 2: The present state - Next state table for D flip-flop can be drawn as shown in Table 17.

Table 17: PS-NS table for D flip-flop.

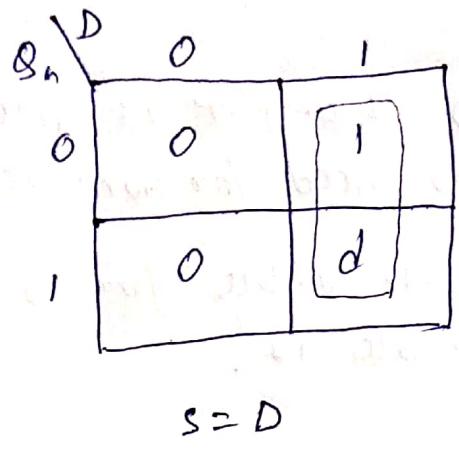
Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

Step 3: Using the application table of S-R flip-flop given in Table 40, the next state codes, i.e. S & R values, can be augmented in the above PS-NS table as shown in Table 18.

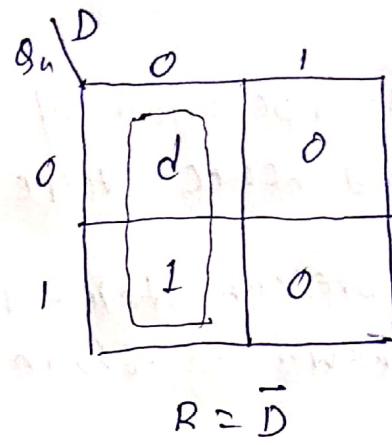
Table 18: Excitation table for D flip-flop realisation using S-R flip-flop

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

Step 4: In this step, one can design the next state decoder, i.e. the simplified expression for S and R from the excitation maps as shown in K-map table (a) and (b).



(a) Excitation map for S



(b) Excitation map for R

Step 5: From the above step, $S=D$ and $R=\bar{D}$.

now the circuit for Delay flip-flop using S-R flip-flop can be drawn, as shown in fig. 18, with a single NOT gate in the next state decoder logic.

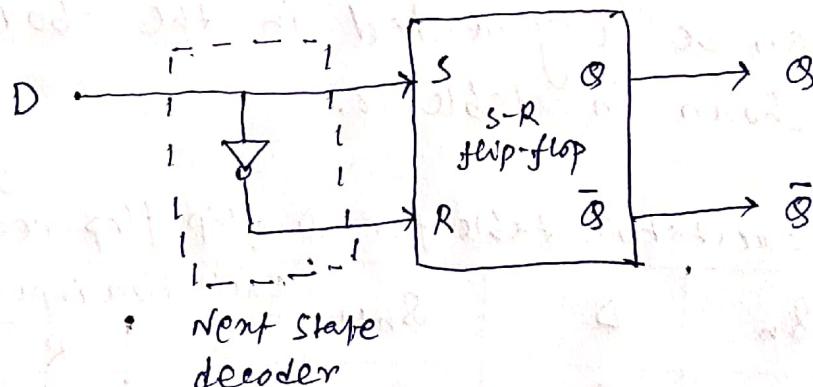


Fig. 18: D flip-flop using S-R flip-flop.

□ Realisation of J-K flip-flop using S-R flip-flop in

→ Here, the desired flip-flop (X) is a J-K flip-flop and the chosen flip-flop (Y) is an S-R flip-flop. The block diagram of J-K flip-flop using S-R flip-flop is shown in fig. 19.

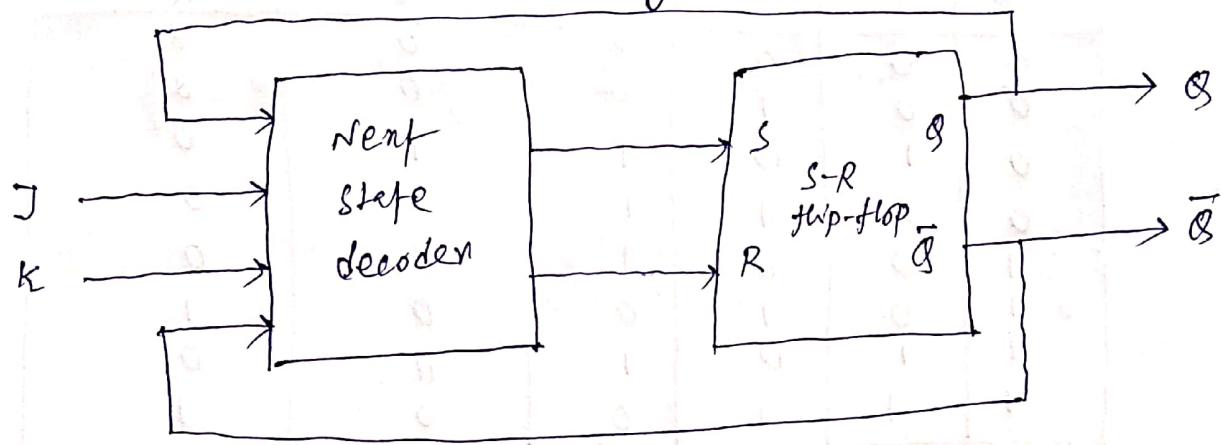


Fig 19: Block diagram of J-K flip-flop using S-R flip-flop.

Step 1: The operation of a J-K flip-flop is well known; hence, description is not given.

Step 2: Present state - next state table for J-K flip-flop can be drawn as shown in table 19.

Table 19: PS-NS table for J-K flip-flop

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	0
1	0	1	1
1	1	0	0
1	1	1	1

Step 3: Using the application table of S-R flip-flop, the next state codes, i.e. S & R values, can be augmented in the above PS-NS table as shown in table 20.

using S-R flip-flop
Table 20: Excitation table for J-K flip-flop realisation

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

Step 4: In this step, one can design the next state decoder, i.e. the simplified expressions for S and R, from the excitation maps as shown below.

Q_n	JK			
	00	01	11	10
0	0	0	1	1
1	d	0	0	d

$$S = J \bar{Q}_n$$

Q_n	JK			
	00	01	11	10
0	d	d	0	0
1	0	1	1	0

$$R = K \bar{Q}_n$$

Step 5: From the above step, $S = J \bar{Q}_n$ and $R = K \bar{Q}_n$. Now, the circuit for J-K flip-flop

using S-R flip-flop can be drawn as shown in fig. 20.

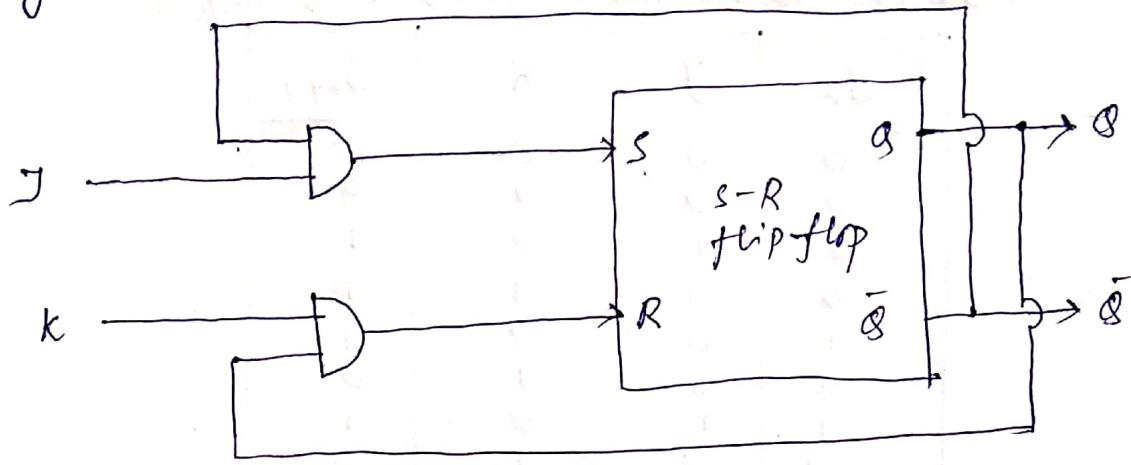


Fig. 20 : JK flip-flop using S-R flip-flop.

□ Realisation of JK flip-flop using D flip-flop:

→ Here, the desired flip-flop (X) is a JK flip-flop and the chosen one (Y) is a D flip-flop. The block diagram of this realization is shown in fig. 21.

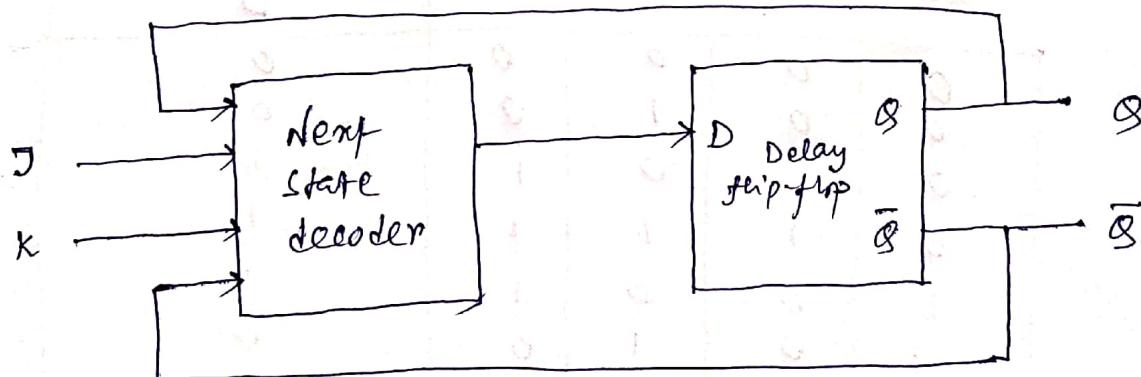


Fig. 21: Block diagram of JK flip-flop using delay flip-flop.

Step 1 : The operation of JK flip-flop is well known. Hence, there is no need for word description.

Step 2: The PS-NS table for J-K flip-flop can be drawn as shown in table 21.

Table 21: PS-NS table for J-K flip-flop

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

Step 3: Using the application table of D flip-flop given in table 9, the next state codes, i.e. D value, can be augmented in the above PS-NS table as shown in table 22.

Table 22: Excitation table for realisation of J-K flip-flop using D flip-flop

Q_n	J	K	Q_{n+1}	Excitation input D
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Step 4: In this step, one can design the next state decoder, i.e. the simplified expression for D, from the excitation map as shown in fig 22. K-map table below.

Q_n	JK	00	01	11	10
0		0	0	(1)	D
1		1	0	0	1

$$D = J\bar{Q}_n + \bar{K}Q_n$$

Step 5: Now, the circuit for J-K flip-flop using D flip-flop is shown in fig. 22 with two AND gates, one OR gate and one NOT gate in the next state decoder logic.

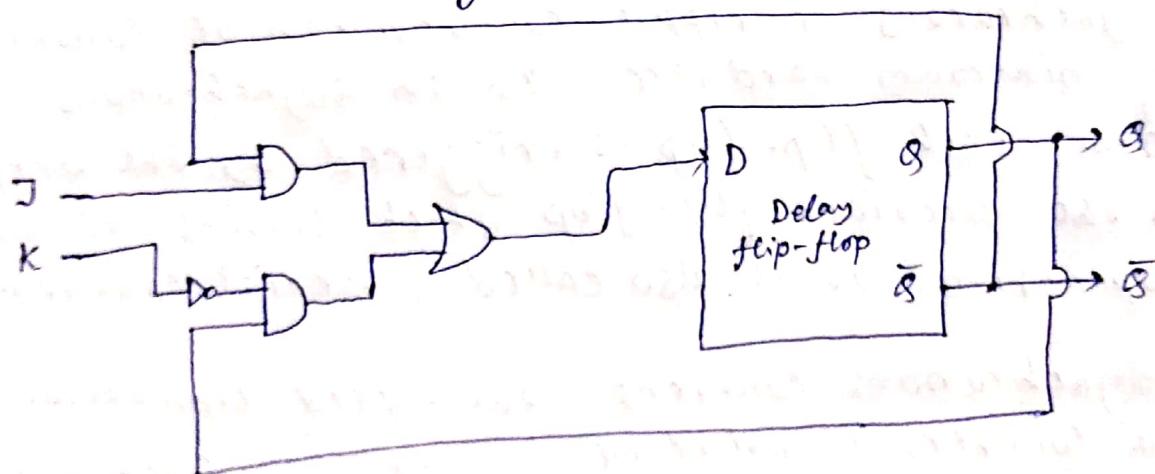


Fig. 22: J-K flip-flop using D flip-flop

Assignment questions :-

- ① Realisation of trigger flip-flop using S-R flip-flop.
- ② Realisation of Delay flip-flop using J-K flip-flop.