

SEQUENTIAL LOGIC CIRCUITS

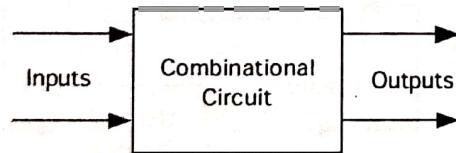
Chapter

4

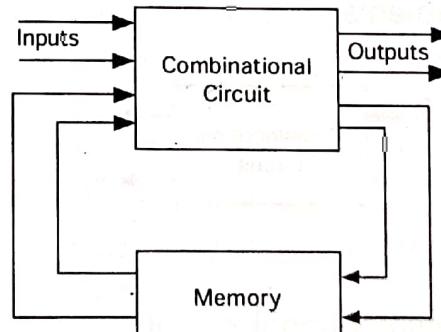
4.0 INTRODUCTION

The digital circuits discussed in previous chapter were all combinational circuits. In these circuits, the input at any particular instant of time solely determines the output at that instant of time. In other words, combinational circuits are those circuits that do not contain any memory elements. On the other hand, sequential circuits are those circuits whose output depends on both the present and previous (past) inputs. Thus, sequential circuits contain at least one memory element, which can store binary information in it.

Fig. 4.1 (a) and (b) shows the block diagram of two types of digital logic circuits, combinational and sequential circuits respectively.



(a) Combinational Circuit



(b) Sequential Circuit

Fig. 4.1

In the combinational circuit of Fig. 4.1 (a), it gives a particular output with a combined set of inputs applied to it simultaneously.

In the sequential circuit of Fig. 4.1 (b), it contains a memory in addition to the present inputs. Sequential circuits thus work in sequence by taking into account, the past input states and the present inputs.

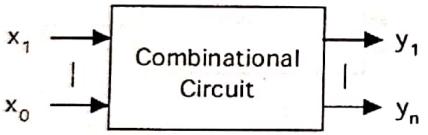
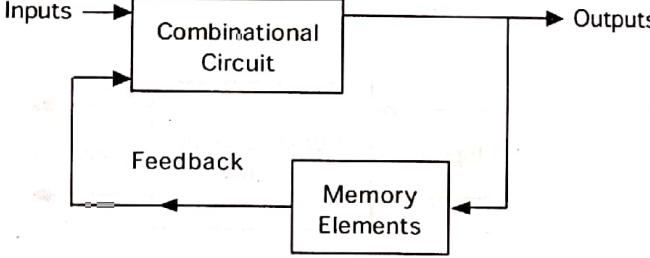
There are basically two types of sequential circuits in digital systems clocked or synchronous and unclocked or asynchronous. In the clocked or synchronous sequential circuits, a master oscillator provides the regular timing pulses. It is

only during these timing pulses that events are allowed to occur. This circuit is otherwise inert and has only propagation delays. A magnetic tape recorder is an example of this type of circuit. On the other hand, unclocked or asynchronous sequential circuits do not wait for the timing pulses. Here an event takes place or occurs only after the completion of a previous event. The telephone dial system is an example of this type of circuit.

In this chapter, we will study different sequential circuits like flip-flops, registers, counters, etc.

4.1 COMPARISON OF COMBINATIONAL AND SEQUENTIAL LOGIC CIRCUITS

The comparison between combinational and sequential logic circuits is presented in the table below.

S. No.	Combinational Circuits	Sequential Circuits
1.	A circuit whose output is dependent only on the inputs at that instant (or) the circuit that does not contain any memory elements. 	1. A circuit whose output depends not only on the present inputs, but also on the past history of the inputs (or) A circuit that contains at least one memory element. 
2.	Combinational circuit output = f (present) inputs (No memory elements so no states)	2. Output = f(present input, present state) Next state = f (present input, present state)
3.	Combinational circuits are easy to design but require more hardware.	3. Sequential circuits are comparatively harder to design but require less hardware.
4.	Combinational circuits are faster in speed, because, delay between input and output is due to propagation delay of gates.	4. Sequential circuits are slower than the combinational circuits because of memory elements (or) delay elements.

- | | | | |
|----|--|----|---|
| 5. | More expensive circuit. | 5. | Cheaper circuit. |
| 6. | Designer has less flexibility since the output depends only on the present inputs. | 6. | Designer has more flexibility because the output depends on both present input and past history of input. |
| 7. | Example : Parallel adder. | 7. | Example : Serial adder. |
| 8. | The behaviour is defined by the set of output functions only. | 8. | The behaviour is defined by the set of output functions and next state functions. |

4.2 COMPARISON OF SYNCHRONOUS SEQUENTIAL AND ASYNCHRONOUS SEQUENTIAL CIRCUITS

A comparison between synchronous sequential and asynchronous sequential circuits is given in the table below :

Synchronous Sequential Circuits	Asynchronous Sequential Circuits
<ol style="list-style-type: none"> 1. A synchronous sequential circuit is a system whose behaviour is defined from the knowledge of its signals at discrete instants of time, i.e., the input signals can affect the memory elements upon the activation of the clock pulse. 2. Here the memory elements are clocked flip-flops. 3. Synchronization is employed by the help of clock pulses. 4. The maximum operating speed of clock depends on time delays involved. 5. Easier to design. 	<ol style="list-style-type: none"> 1. The behaviour of asynchronous sequential circuit depends upon the order in which the inputs change, and the state of the circuits can be affected at any instant of time, i.e., the change in the input signals can affect memory element at any instant of time. 2. The memory elements are either unclocked flip-flops (or) time delay elements. 3. Here no synchronization, hence it is a combinational circuit with feedback. 4. Because of absence of clock, asynchronous circuits can operate faster than synchronous circuits. 5. More difficult to design.

4.3 FLIP-FLOPS

We know that a sequential circuit needs at least one memory element to store the present state of the circuit. This information must be stored at least one

clock signal period (in case of clocked sequential circuit) or until the input signals change (in case of unlocked sequential circuit). As soon as we said memory device what we will get into mind is ROM. But ROM is combinational, according to the above discussion of storage element needed in sequential circuits, ROM has no memory. (The memory of a ROM refers the fact that it "memorizes" the functions relationship between the output variables and the input variables).

A flip-flop is a circuit that maintains the binary state indefinitely until directed by an input signal to switch states. Therefore, the storage device must retain the stored information until we are ready to use it. The basic unit for this storage is a flip-flop (abbreviated as FF). It is a device with two stable states i.e., output is either to 0V (logic 0) or +5V d.c. (logic 1). A flip-flop maintains its output state (either 0 or 1) until directed by an input signal to change its state. It always stores 1-bit information. So it is also called 1-bit memory unit.

4.4 NAND AND NOR LATCHES

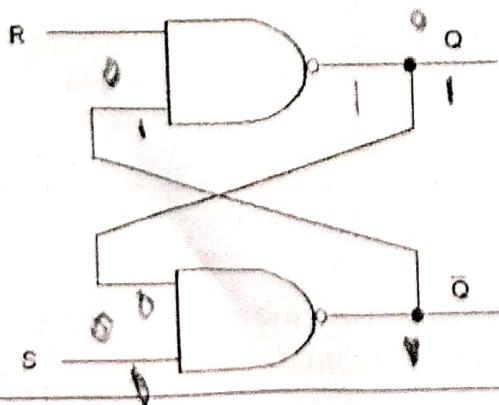
(Oct. 2017)

A flip-flop is a basic memory element used to store one bit of information. A flip-flop, also called a bistable multivibrator (BMV) or a latch, is a circuit which can be triggered into either of the two stable states.

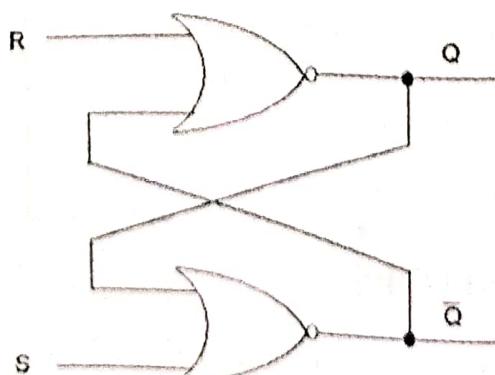
Latch is a type of bistable circuit, similar to a flip-flop, that can reside in either of two stable states by virtue of feedback arrangement. The main difference between latches and flip-flops is in the method used for changing their state.

Consider RS (Reset Set) latches, formed with two cross coupled NAND gates and NOR gates; as shown in Fig. 4.2 (a) and (b) respectively. In these latches,

R and S are the two inputs; and the two outputs are Q and \bar{Q} . Notice that the output of each gate is connected to the input of the other gate. This produces the feedback, which is the characteristic of all multivibrators.



(A.P)



(T.S)

Inputs		Output
R	S	Q
0	0	unpredictable state
0	1	1
1	0	0
1	1	No change

(a) NAND Latch

Inputs		Output
R	S	Q
0	0	No change
0	1	1
1	0	0
1	1	unpredictable state

(b) NOR Latch

Fig. 4.2 NAND / NOR Latches Along with their Truth Tables

Fig. 4.2 (a) shows a pair of NAND gates that drive a NAND latch and its truth table.

Operation :

- (i) When R and S inputs are both at logic 0, both outputs go to a logic 1. This is known as 'unpredictable state' (or) 'prohibited state' for the latch and is not used.
- (ii) When R = 0 and S = 1; the output Q is set logical 1. This is called the set condition.
- (iii) When S = 0 and R = 1; the output Q is reset (cleared) to 0. This is called the reset condition.
- (iv) When both inputs R and S are at 1; this is the idle or at rest condition and leaves the outputs Q and \bar{Q} in their previous complementary states. This is called the hold condition or no change state.

By observing the above four operations, we can rewrite the truth table for NAND latch with the complete details as shown in the table below.

Table : The complete truth table for NAND latch

Mode of Operation	Inputs		Outputs		Effect on Q
	R	S	Q	\bar{Q}	
Prohibited	0	0	1	1	Do not use
Set	0	1	1	0	For setting Q to 1
Reset	1	0	0	1	For setting Q to 0
Hold	1	1	Q	\bar{Q}	Depends on previous state

From the above truth table, we must note that whenever a NAND latch, we must avoid the condition when both inputs at LOW at the same time.

4.5 NEED FOR CLOCK

In digital computing circuits, all the operations are precisely timed by means of a **master clock** which sends out the timing signals to different sections of the computer. This coordinates and synchronises the overall action and prevents different sections to respond to an input signal at their own will. A **clocked** or **synchronous** system has output that does not change according to the input conditions until a clock pulse is applied. The clock pulse initiates action upon the input signal conditions.

A clock circuit produces pulses called square waves as shown in Fig. 4.3. Among the important portions of a clock pulse are the **leading edge**, **raising edge** or **positive going edge** and the **trailing edge**, **falling edge** or **negative going edge**.

Most of the flip-flops respond to either (but not both) of the two edges. A circuit is called a **leading edge triggered** or **trailing edge triggered** depending upon which of the two square wave edges initiates action in the circuit. A flip-flop of this type is called a **clocked flip-flop** and the clock input is applied at a point known as the **clock point**. The clock input in a flip-flop is marked by a small triangle. For positive edge triggered circuits, a line is drawn from the middle of the triangle to represent the clock point. For negative edge triggered circuits, a small circle is put between the clock point (line) and the triangle. Fig. 4.4 shows the symbols for positive and negative edge triggered flip flops.

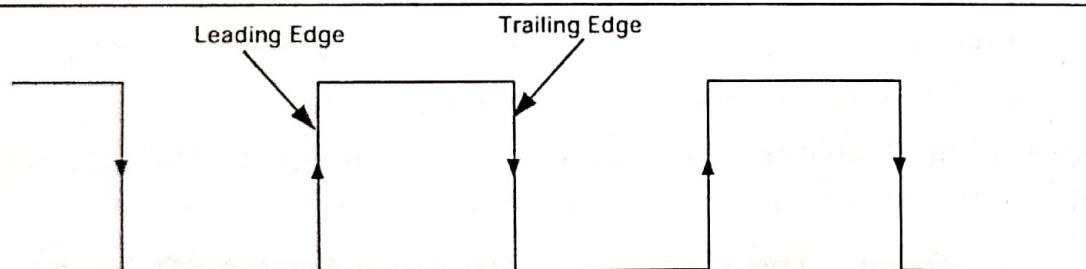
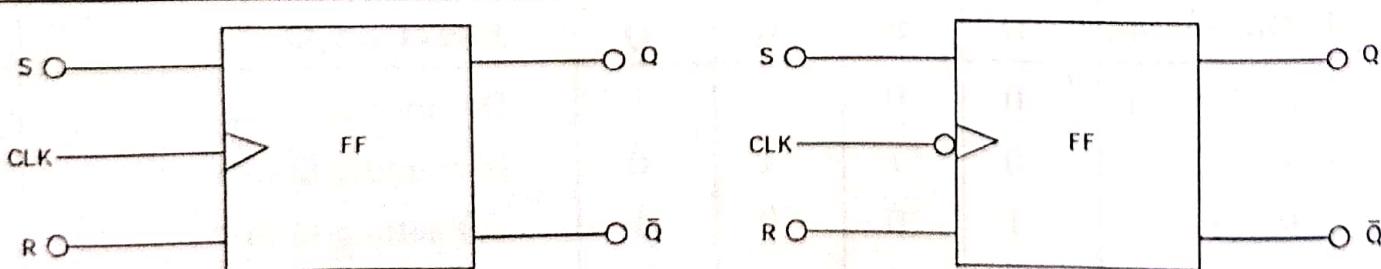


Fig. 4.3 Clock Signal



(a) Positive Edge Triggered

(b) Negative Edge Triggered

Fig. 4.4 Clocked Flip-Flop

4.6 TRIGGERING IN FLIP-FLOPS

(Oct./Nov., 2012)

By a momentarily change in the input signal the state of a flip-flop is switched. (0-1-0). This momentarily change in the input signal is called a trigger. There are two types by which flip-flops can be triggered.

- Edge Triggering

- Level (Pulse) Triggering

An **edge-triggered flip-flop** responds only during a clock pulse transition i.e., clock pulses switches from one level (reference voltage) to another. In this type of flip-flops, output transitions occur at a specific level of the clock pulse. When the pulse input level exceeds this reference level, the inputs are locked out and flip-flop is therefore unresponsive to further changes in inputs until the clock pulse returns to 0 and another clock pulse occurs. An edge-triggered flip-flop changes states either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse on the control input.

When the triggering occurs on the positive going edge of the clock, it is called **positive edge triggering** and when the triggering occurs at the trailing edge of the clock this is called as **negative edge triggering**.

The term **pulse-triggered** or **level triggered** means that data are entered into flip-flop on the rising edge of the clock pulse, but the output does not reflect the input state until the falling edge of the clock pulse. As this kind of flip-flops are sensitive to any change of the input levels during the clock pulse is still HIGH, the inputs must be set up prior to the clock pulse's rising edge and must not be changed before the falling edge. Otherwise, ambiguous results will happen.

(Oct./Nov., 2007)

4.7 CLOCKED RS FLIP-FLOP

The basic RS flip-flop is an asynchronous transparent sequential circuit. This means that any change in the input of R or S is transmitted immediately to the

output at Q and \bar{Q} according to the truth table. The operation of the basic flip-flop can be modified by providing an additional control input that determines when the state of the circuit is to be changed. This additional control input is called "clock" or "clock pulse". With this clock, the flip-flop is called a 'synchronous' device. The term synchronous indicates that the output changes its state only at a specified point according to an input, i.e., the changes in the output occur in synchronous with the clock. By adding gates to the input of the basic circuit, the flip-flop can be made to respond to input levels during the occurrence of a clock pulse.

IT S

An RS-flip-flop with a clock pulse is shown in Fig. 4.5 (a). It consists of basic NOR flip-flop circuit and two AND gates at the input. The AND gates remain at '0' as long as the clock pulse (CP) is '0', regardless of the R and S inputs. When the clock pulse goes to 1, information from R and S inputs is allowed to reach the basic flip flop. The logic symbol and truth table for clocked RS-flip-flop are shown in Fig. 4.5 (b) and 4.5 (c) respectively.

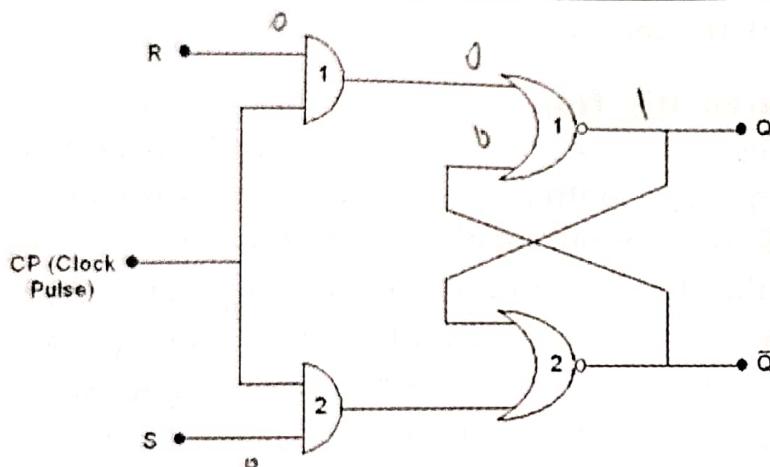


Fig. 4.5 (a)

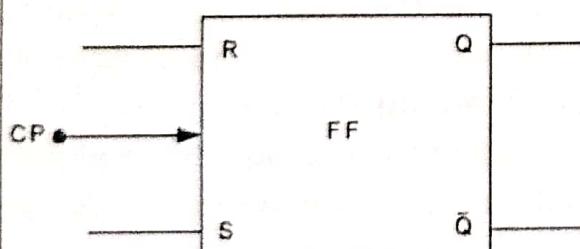


Fig. 4.5 (b) Logic Symbol

CP	R	S	Q	\bar{Q}	State
0	0	0	NC	NC	No change
0	0	1	NC	NC	No change
0	1	0	NC	NC	No change
0	1	1	NC	NC	No change
1	0	0	NC	NC	No change
1	0	1	1	0	Set
1	1	0	0	1	Reset
1	1	1	x	x	Indeterminate

Fig. 4.5 (c) Truth Table

An example of input and output waveforms of flip-flop are shown in Fig. 4.5 (d).

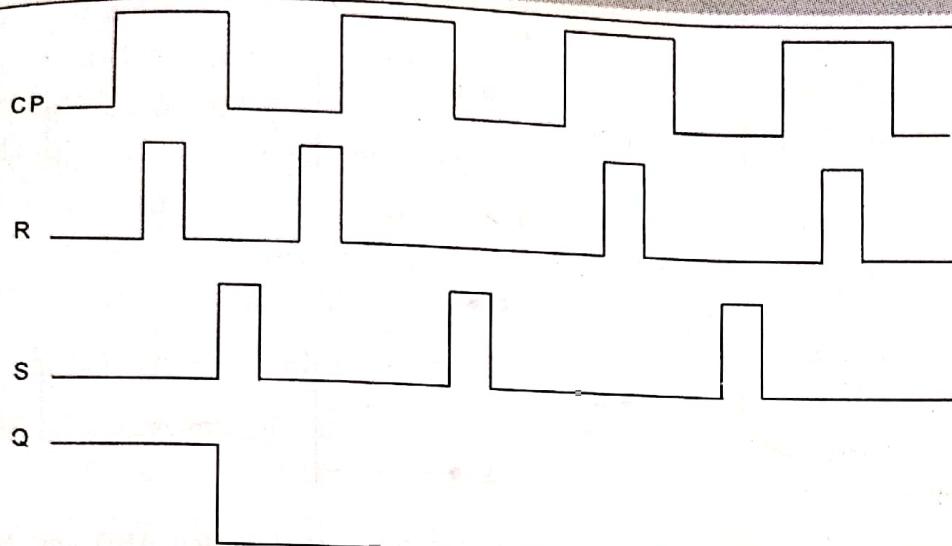


Fig. 4.5 (d)

Operation :

1. When clock is absent (or) $CLK = 0$, the circuit will work as a RS-flip-flop.
2. In the presence of clock i.e., $CLK = 1$.
 - (a) **If $R = 0$ and $S = 0$ and assume that $Q = 1$.** The output of both AND gates are '0' which are given as one of inputs to the NOR gate. If $Q = 1$ and $\bar{Q} = 0$; the output of NOR gate 1 = 1 and NOR gate 2 = 0, i.e., same as the previous values. This state is called 'no change' state.
 - (b) **If $R = 1$, $S = 0$.** The output of AND gate 1 = 1 and AND gate 2 is '0'. With these inputs, the NOR gate 1 output = 0 and NOR gate 2 output is 1. This state is called "reset" state.
 - (c) **If $R = 0$, $S = 1$.** The output of AND gate 1 is '0' and AND gate 2 is 1. With these inputs the NOR gate 1 output = 1 and NOR gate 2 output is 0. This state is called "set state".
 - (d) **If $R = 1$, $S = 1$.** The output of both AND gates is 1. With these inputs, irrespective of other input, the output of both NOR gates is '0', i.e., $Q = 0$ and $\bar{Q} = 0$, which is an invalid output. This state is called 'indeterminate state'.

For all the 4-conditions, the RS flip-flop and clocked RS flip-flop outputs are same, but in clocked RS flip-flop, the clock must be present.

Clocked RS Flip-flop using NAND Gates : We can construct the clocked RS flip-flop using NAND gates as shown in Fig. 4.6. It consists of basic RS flip-flop circuit and two additional NAND gates. The clock pulse input acts as an enable input for the other two inputs.

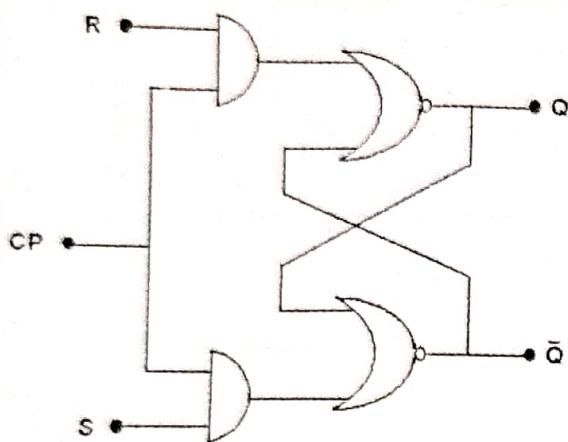


Fig. 4.6 (a) Basic Circuit

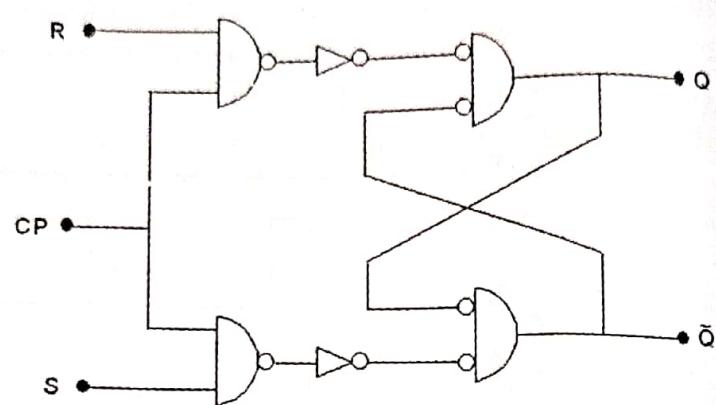


Fig. 4.6 (b) With Bubbled AND and NAND Gates

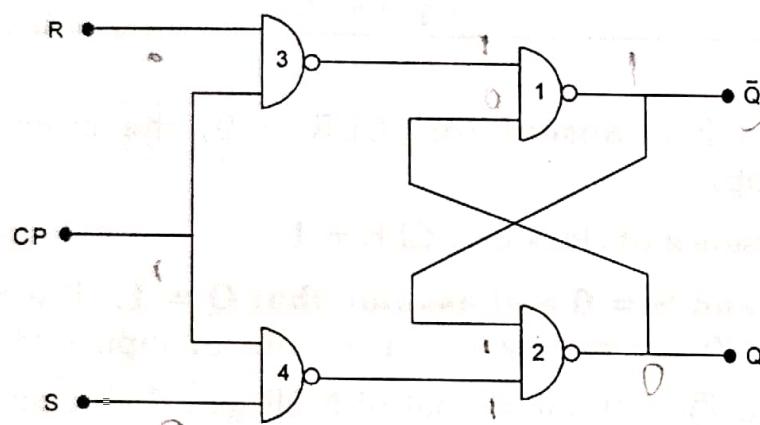


Fig. 4.6 (c) Logic Diagram with NAND Gates

Operation : 1. When Clock pulse = 0. The output of NAND gates 3 and 4 stay at logic 1 level. This is the 'no change' condition of the basic flip-flop.

2. When clock pulse (CP = 1). In this condition, the S or R input is allowed to react the output.

(a) **When R = 0 and S = 0 (Assume Q = 1 and $\bar{Q} = 0$).** In this condition, the output of NAND gate 3 and 4 are at logic 1, with $Q = 1$. The both inputs of NAND gate 1 are at logic 1, then its output i.e., $\bar{Q} = 0$. The NAND gate 2 output i.e., $Q = 1$. Hence the output remains same. This state is called "no change" state.

(b) **R = 1 and S = 0.** In this condition, the output of NAND gate 3 is at logic 0 and output of NAND gate 4 is at logic 1. With these inputs, the output of NAND gate 1 is at logic 1 i.e., $\bar{Q} = 1$ and the output of the NAND gate 2 is at logic 0 i.e., $Q = 0$. This state is called "reset" state.

(A.P)

(T.S)

(c) **If $R = 0$ and $S = 1$.** In this condition, the output of NAND gate 4 is at logic 0 and output of NAND gate 3 is at logic 1. With these inputs, the output of NAND gate 1 and NAND gate 2 are logic 0 and logic 1 respectively. This state is called "set" state.

(d) **If $R = 1$ and $S = 1$.** In this condition, the output of both NAND gate 3 and NAND gate 4 are at logic 0. With these inputs, the output of both NAND gate 1 and NAND gate 2 is at logic 1, i.e., $Q = 1$ and $\bar{Q} = 1$, which is invalid. This state is called 'indeterminate state'. When CP input goes back to 0 (while S and R are maintained at logic 1), it is not possible to determine the next state, as it depends on whether the output of gate 3 or gate 4 goes to 1 first. This indeterminate state makes the circuit difficult to manage and it is seldom used in practice.

This is an important circuit because all other flip-flops are constructed from it. The characteristic table of the flip-flop is shown below.

Truth Table for Clocked RS Flip-Flop

$Q(t)$	R	S	$Q(t + 1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	Indeterminate
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	Indeterminate

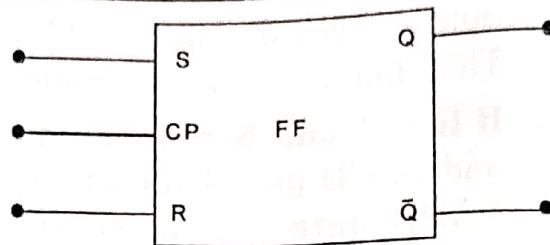
In the table, $Q(t)$ stands for binary state of the flip-flop before the application of a clock pulse, referred to as the "present state". The R and S columns give the possible values of the inputs, and $Q(t + 1)$ is the state of the flip-flop after the application of a single pulse, referred to as the "next state". Note that the CP input is not included in the characteristic table. If the present state is Q and the inputs S and R, the application of a single clock pulse causes the flip-flop to go to the next state, $Q(t + 1)$.

The characteristic equation is derived from the k-map shown in Fig. 4.7(a). This equation specifies the value of the next state as a function of the present state and the inputs.

$$Q(t+1) = S + \bar{R}Q \text{ and } SR = 0$$

RS	00	01	11	10
0	0	1	x	0
1	1	1	x	0

(a) Characteristic Equation



(b) Graphic Symbol

Fig. 4.7

The two indeterminate states are marked with don't care conditions in the map. The relation $RS = 0$, must be included as a part of the characteristic equation to specify that both R and S cannot be equal to 1 simultaneously. The graphic/ logic symbol of the RS flip-flop is shown in Fig. 4.7 (b). It is a rectangular shape block with three inputs S, R and CP and two outputs Q and \bar{Q} .

4.8 JK FLIP-FLOP

[March 2008, 2009, April 2012, 2015]

The JK flip-flop is a refinement of the RS flip-flop. In JK flip-flop, the indeterminate state in RS flip-flop is defined. The inputs J and K behave like inputs S and R to set and reset the flip-flop, respectively. When both inputs J and K are equal to 1, the flip-flop switches to its complement state, that is, if $Q = 0$, it switches to $Q = 1$, and vice versa. This condition can be written as

$$Q_{n+1} = \bar{Q}_n \text{ (or) } Q(t+1) = \bar{Q}(t)$$

Where Q_{n+1} = next state = $Q(t+1)$

Q_n = present state = $Q(t)$

Here, the indeterminate state in the RS flip-flop can be eliminated by using the feedback technique.

A clocked JK flip-flop can be constructed with two cross coupled NOR gates and two AND gates are shown in Fig. 4.8 (a). The logic diagram of it is shown in Fig. 4.8 (b). Fig. 4.8 (c) shows the logic symbol for clocked JK flip-flop and its truth table is shown in Fig. 4.8 (d).

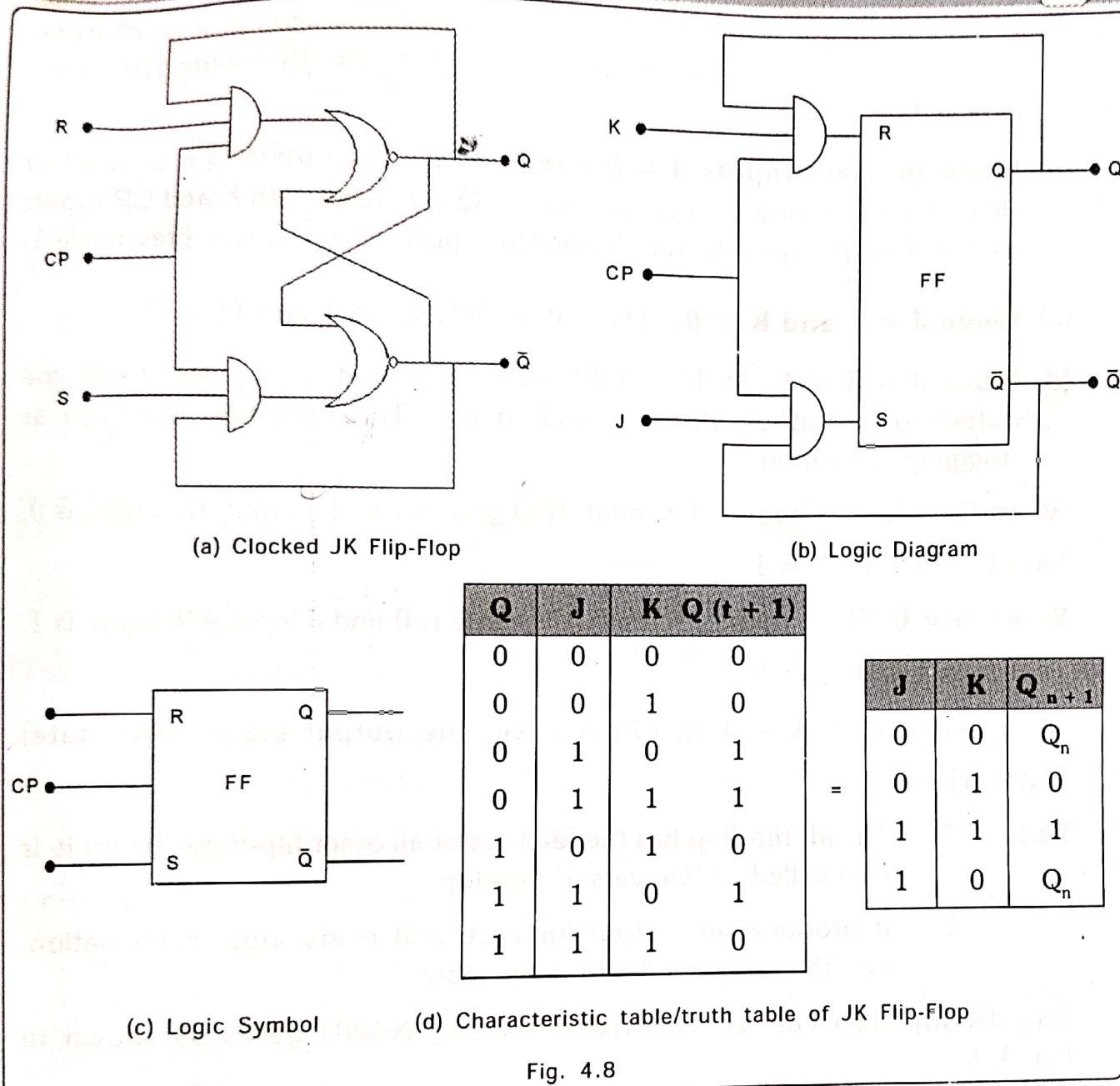


Fig. 4.8

- The output Q is ANDed with K and CP inputs so that the flip-flop is cleared during a clock pulse only if \bar{Q} was previously 1.
- Similarly the output \bar{Q} is ANDed with J and CP inputs so that the flip-flop is set with a clock pulse only when \bar{Q} was previously 1.
- When both J and K are 1, the input pulse is transmitted through one AND gate only; the one whose input is connected to the flip-flop. Output that is previously equal to 1.

Operation : 1. As long as the CP input is at LOW level, the data inputs have no effect on the outputs. Hence, the output holds the previous data. This state is called "no change state".

2. When CP is HIGH level.

(A.P)

(T.S)

- (a) **If data inputs $J = K = 0$.** The flip flop remains in "no change or hold" state, in which data inputs have no effect on the outputs, i.e., $Q(t+1) = Q$.
- (b) **When the data inputs $J = 0$ and $K = 1$.** The output Q is in reset or cleared to 0. In other words, the output Q is ANDed with K and CP inputs so that the flip-flop is cleared during clock pulse only if Q was previously 1.
- (c) **When $J = 1$ and $K = 0$.** The output Q is set to 1 and $\bar{Q} = 0$.
- (d) **When $J = K = 1$.** In this condition, the repeated clock pulses cause the output to turn-off-on-off-on ... and so on. This condition is known as "toggling" operation.

When $Q = 1$. The output of K input AND gate is 1 and J -input AND gate is 0. Thus $Q = 0$ and $\bar{Q} = 1$.

When $Q = 0$. The output of K input AND gate is 0 and J input AND gate is 1. Thus $Q = 1$ and $\bar{Q} = 0$.

Thus, when $J = K = 1$ in either case, the output state (next state) $Q(t+1) = \bar{Q}(t)$.

- Note :**
- 1. The JK flip-flop has the features of all other flip-flops, hence it is also called as "Universal" flip-flop.
 - 2. It produces an output for each and every input combination, i.e., there is no indeterminate state.

The JK flip-flop can be constructed using NAND gates as shown in Fig. 4.9.

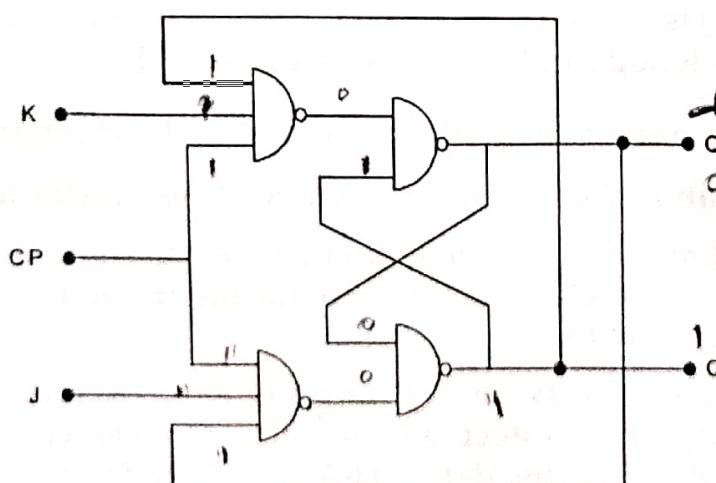
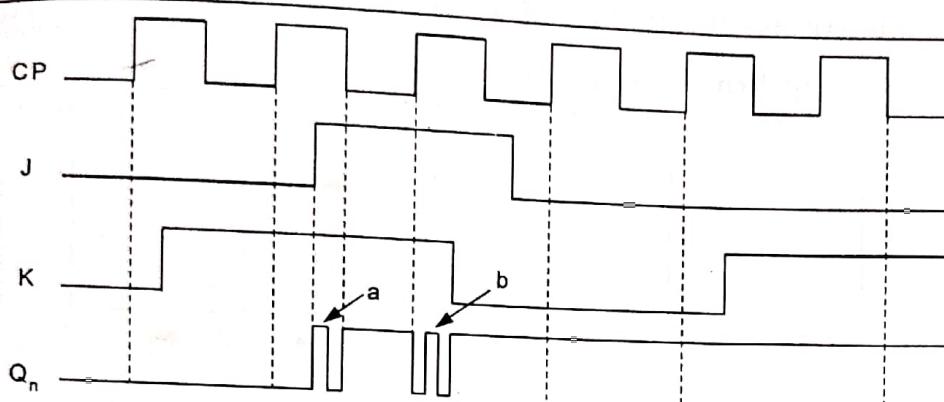


Fig. 4.9 JK Flip-Flop using NAND Gates Only

The relation between input and output waveforms for clocked JK flip flop is shown in Fig. 4.10.



Note : a and b are toggle condition output.

Fig. 4.10 Input and Output Waveforms of Clocked JK Flip-Flop

4.9 RACE AROUND CONDITION

[Oct. 2007, March 2009, Oct. 2017]

It is important to note that in JK flip-flop, the output is feedback to the input, and therefore change in the output results change in the input, if the inputs $J = K = 1$ and $Q = 0$.

When a clock pulse with a width ' t_p ' as shown in Fig. 4.11 (a) is applied, the output will change from 0 to 1 after the time interval Δt ,

Where Δt = propagation delay of two level NAND gates

$$t_p = \text{pulse width}$$

Now, after Δt , we have $J = K = 1$ and $Q = 1$ and after another 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 when $t_p < \Delta t$ as shown in Fig. 4.11 (b). This condition can be obtained by two ways.

1. If t_p is reduced.
2. If Δt is increased.

Reduction of t_p means, we have to use a pulse generator to produce less pulse width waveform, but it is difficult to get such type of circuit.

The value of Δt can be increased by using the lumped delay lines in series with the feedback connection, which is worthless.

There are two solutions for reducing race around condition.

1. Use of edge triggering, which we have already studied.
2. Master-slave flip-flop configuration.

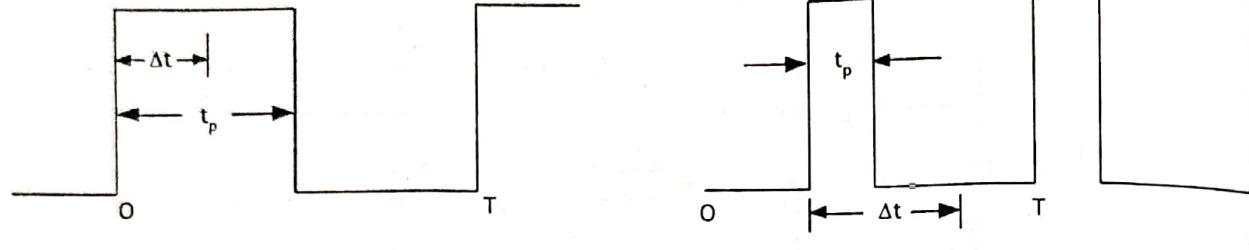


Fig. 4.11

4.10 MASTER-SLAVE JK FLIP-FLOP

[March 2008, Oct. 2007, 2011, 2012, 2013]

A master-slave flip-flop can be constructed using two JK flip-flops connected serially. The first flip-flop serves as the "master" and second as a "slave", and the overall circuit is referred to as a "master-slave flip-flop". The logic diagram of a JK master-slave flip-flop is shown in Fig. 4.12. It consists of a master flip-flop, a slave flip-flop, and an inverter. The first flip-flop i.e., master is driven by the positive edge of the clock pulse, the second flip-flop, called slave is driven by the negative edge of the clock pulse.

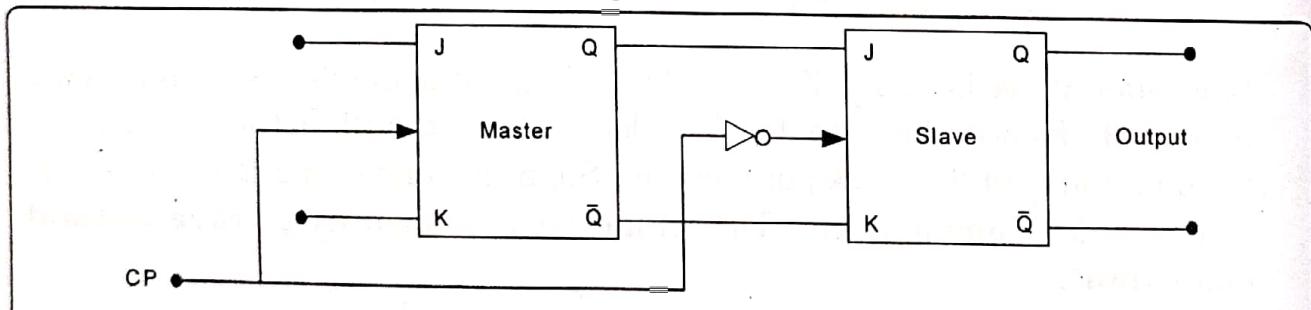


Fig. 4.12 Master Slave Flip-Flop

- Operation :**
1. When $CP = 1$, i.e., clock input of master is 1, the master acts according to its JK input but the slave does not respond, since it requires a negative edge at the clock input.
 2. When $CP = 0$, i.e., clock input of slave is 1, the slave flip flop copies the mast outputs, but the master does not respond, since it requires a positive edge at its clock input. Thus, master slave flip-flop does not have race around problem.

The output state of the master flip-flop is determined by the J and K inputs at the positive clock pulse. The output state of the master is then transferred as

an input to the slave flip-flop. The slave flip-flop uses these inputs at the negative clock pulse to determine the output state. Fig. 4.13 illustrates the operation of the master-slave flip-flop.

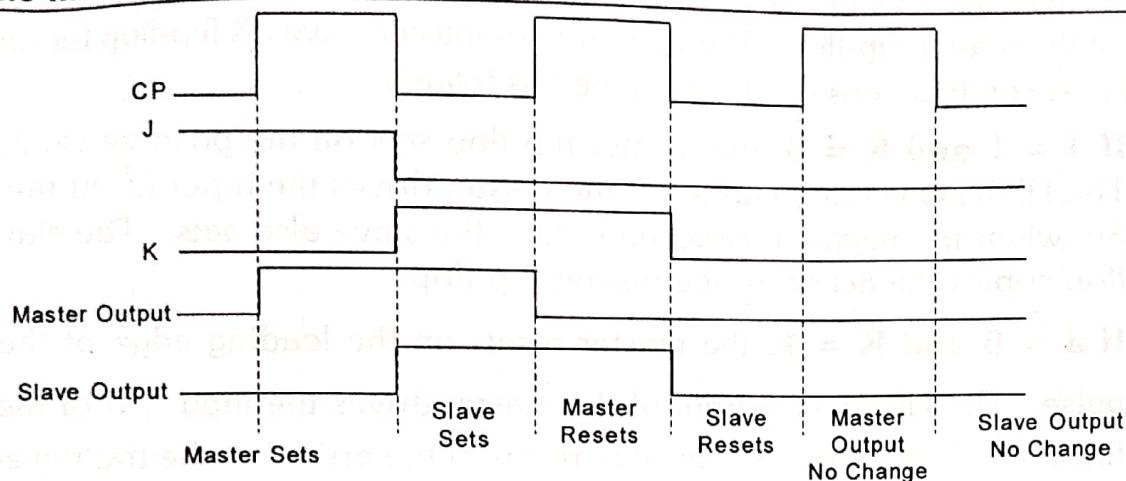


Fig. 4.13 Input and Output Waveforms for Master-Slave Flip-Flop

The master-slave combination can be constructed for any type of flip-flop. The master slave JK flip-flop constructed using NAND gates is shown in Fig. 4.14. It consists of two flip-flops connected in series. NAND gates-1 through 4 form the master flip-flop and NAND gates-5 through 8 form the slave flip-flop. When clock is positive, a change in J and K inputs causes a change of state in the master flip-flop. During this period, the slave remains in its previous state and serves as a buffer between the master and the output.

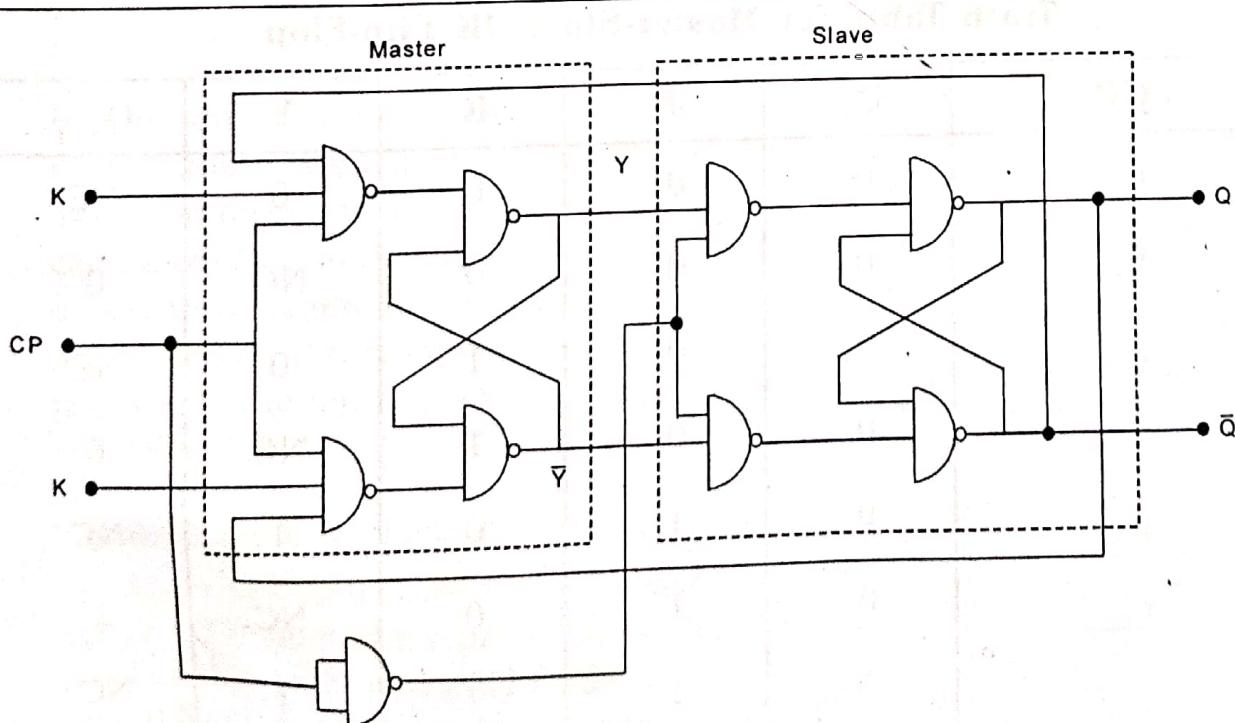


Fig. 4.14 Clocked Master-Slave Flip-Flop using NAND Gates

When the clock goes negative, the master flip-flop does not respond, i.e., it maintains previous state, while the slave flip-flop is enabled and changes its state to that of the master flip-flop. The new state of slave flip-flop is enabled and changes its state to that of the master flip-flop. The operation of master-slave JK flip-flop for different JK inputs combinations can be explained as follows.

- (a) **If $J = 1$ and $K = 0$,** the master flip-flop sets on the positive clock edge. The HIGH Q (1) and Output of the master drives the input (J) of the slave. So, when the negative clock edge hits, the slave also sets. The slave flip-flop copies the action of the master flip-flop.
- (b) **If $J = 0$ and $K = 1$,** the master resets on the leading edge of the clock pulse. The HIGH \bar{Q} output of the master drives the input (K) of the slave flip-flop. Then, the slave flip-flop resets at the arrival of the trailing edge of the clock pulse. Once again, the slave flip-flop copies the action of the master flip-flop.
- (c) **If $J = K = 1$,** the master flip-flop toggles on the positive clock edge and the slave toggles on the negative clock edge.
- (d) **If $J = K = 0$,** this input condition does not produce any change. Thus master-slave flip-flop operates from a complete clock pulse, and the outputs change on the negative transition.

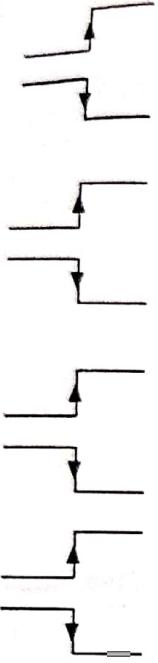
Table below gives the truth table for M/S. JK Flip-flop

Truth Table for Master-Slave JK Flip-Flop

CP	Q_n	J	K	Y	Q_{n+1}
	0	0	0	0	NC
	0	0	0	NC	0
	0	0	1	0	NC
	0	0	1	NC	0
	0	1	0	1	NC
	0	1	0	NC	1
	0	1	1	1	NC
	0	1	1	NC	1

(A.P)

(T.S)



	1	0	0	1	NC
	1	0	0	NC	1
	1	0	1	0	NC
	1	0	1	NC	0
	1	1	0	1	NC
	1	1	0	NC	1
	1	1	1	0	NC
	1	1	1	NC	0

4.11 ASYNCHRONOUS INPUTS OF FLIP-FLOP

[March 2009, Oct 2013]

For the flip-flops discussed so far, the S R, J K, D and T inputs called "control inputs", are also called "**synchronous inputs**". For such synchronous inputs, the flip-flop changes its state in synchronization with the clock input only. Some of the synchronous control inputs must be used along with a clock to trigger a change in the flip-flop.

In addition to the above synchronous inputs, most of the clocked flip-flops have one or more asynchronous inputs, whose effect on the flip-flop output is independent of the synchronous inputs and clock input. Asynchronous inputs can be used to set the flip-flop to the 1 state, or clear the flip-flop to the 0-State at any time, regardless of the condition at the other inputs. These inputs are also called as "direct-inputs". These inputs are connected directly into the latch portion of any flip-flop so that they override the effect of synchronous inputs and the clock.

Fig. 4.15 shows a clocked JK flip flop with PRESET (PR) and CLEAR (CLR) or "direct set" and "direct reset" inputs. These are active LOW inputs. An active LOW level at the preset input will SET the flip-flop, and an active LOW level on the clear input will RESET it. These asynchronous inputs are activated by a 0 level, as indicated by the small bubble on the flip-flop symbol. Both the preset and clear inputs must be kept HIGH for synchronous operation.

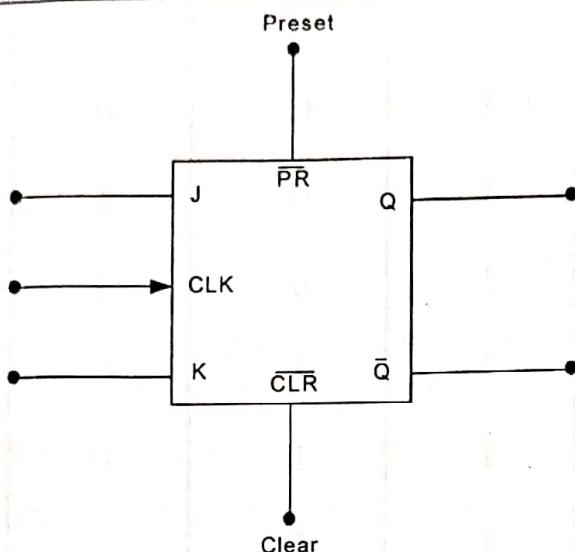


Fig. 4.15 Logic Symbol for JK Flip-Flop with Active LOW Preset and Clear Inputs

Table below shows how these inputs operate.

Note : LOW levels on preset and clear should not be applied simultaneously, since it leads to an ambiguous condition.

Truth Table				
Inputs		Outputs		Action
Preset	Clear	Q_{n+1}	\bar{Q}_{n+1}	
0	0	x	x	Forbidden
0	1	1	0	Set
1	0	0	1	Reset/Clear
1	1	Q_n	\bar{Q}_n	No change

4.12 D FLIP-FLOP

[March 2008, April 2012, Oct. 2013]

Before going to the details of D-flip-flop, we shall study about D-latch and clocked D-latch.

D-Latch :

You must have noticed that in RS latches if both S and R inputs are low at the same time, the output is invalid. To overcome this problem an Inverter can be incorporated ahead of the S input as shown in Fig. 4.16. This modified flip-flop is called a D latch.

The input at D drives the R input of the NAND latch and its complement drives the S input. When the input at D is 1, the S input is 0 and this sets the latch.

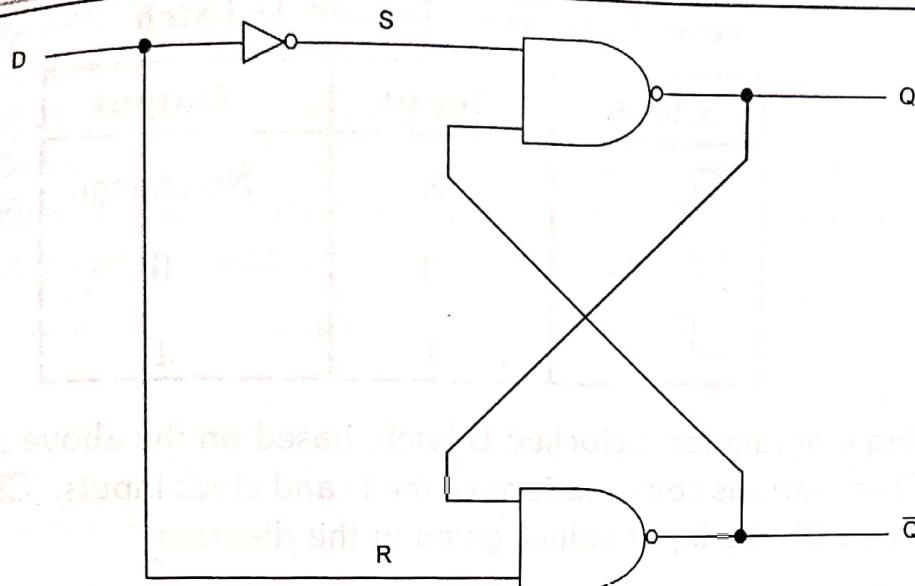


Fig. 4.16 D-Latch

When the D input is 0, the output at R is also 0 and this resets the latch. Since the S and R inputs are always in opposite states, the invalid states does not occur. The truth table for the D-latch is given below.

Truth Table for D Latch

Input		Output	
D		Q	\bar{Q}
0		0	1
1		1	0

Clocked D-Latch :

[Oct. 2013]

A clocked D latch is very similar to an RS latch and, besides, it does not produce an invalid state. A circuit for a clocked D latch is given in Fig. 4.17 and its symbol in Fig. 4.18. The truth table for this latch is given below.

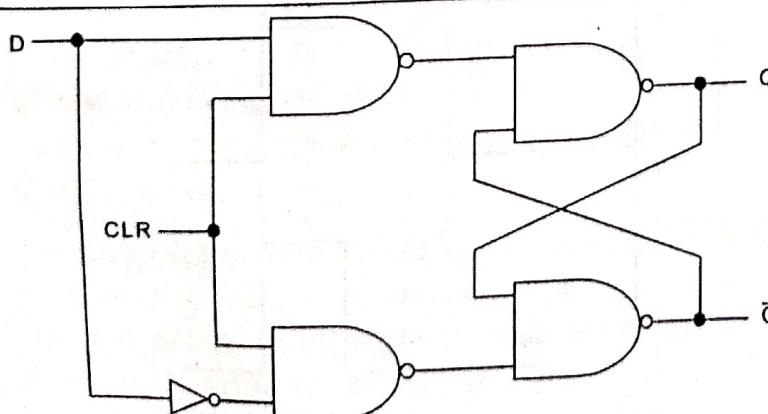


Fig. 4.17 Clocked D Latch

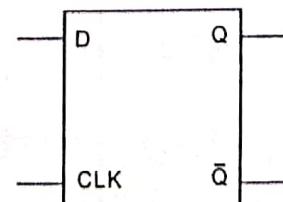


Fig. 4.18 Symbol for Clocked D Latch

Truth Table for Clocked D Latch

Clock	Input	Output
[Waveform: High-to-Low transition]	X	No change
[Waveform: Low-to-High transition]	0	0
[Waveform: Low-to-High transition]	1	1

The timing diagram for a clocked D latch, based on the above table is given in Fig. 4.19 for various combinations of the D and clock inputs. Check the output waveforms with the input values given in the diagram.

D Flip-Flop (Edge-Triggered) :

A disadvantage of level clocked (pulse-triggered) flip flops is that, the input to the flip-flop has to be held constant as long as the clock is active. If the input keeps changing while the clock is active, the output will also keep changing and this is obviously undesirable.

CLK	1	1	0	0	1	1	0	0	1	1	
D	0	1	1	0	1	0	0	1	1	0	
Q	0	1	1	1	1	0	0	0	1	0	

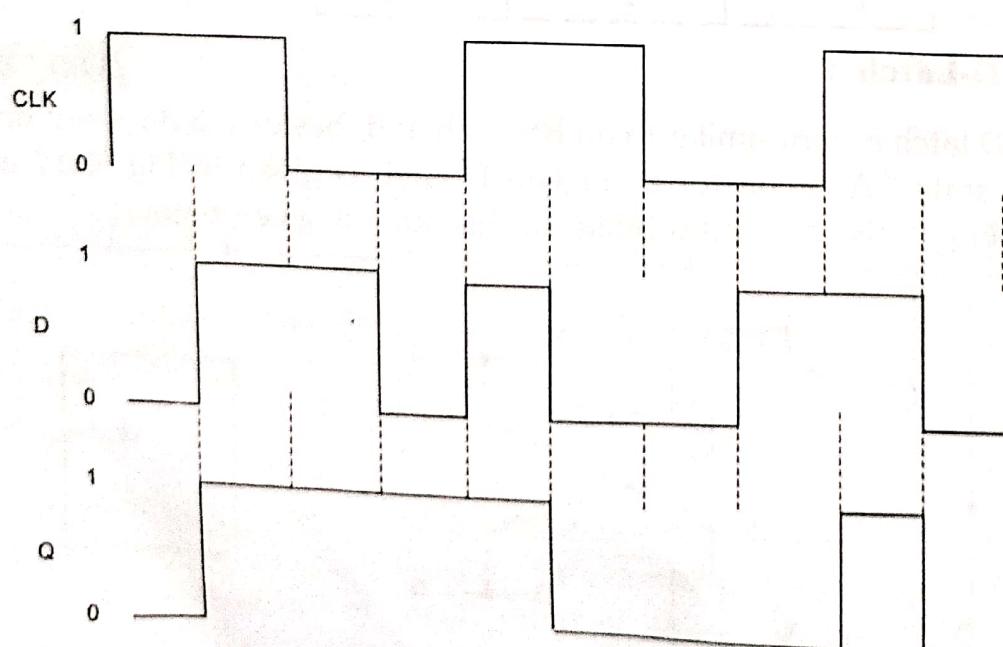


Fig. 4.19 Timing Diagram for Clocked D Latch

(A.P)

(T.S)

Edge-triggering of a flip-flop can be achieved by so designing it, that it is triggered only for an instant either by the rising or the falling edge of a clock pulse. Let us consider the effect incorporating an RC circuit at the clock input device, although this is not a practical way of doing it, as capacitors cannot be conveniently incorporated on chips.

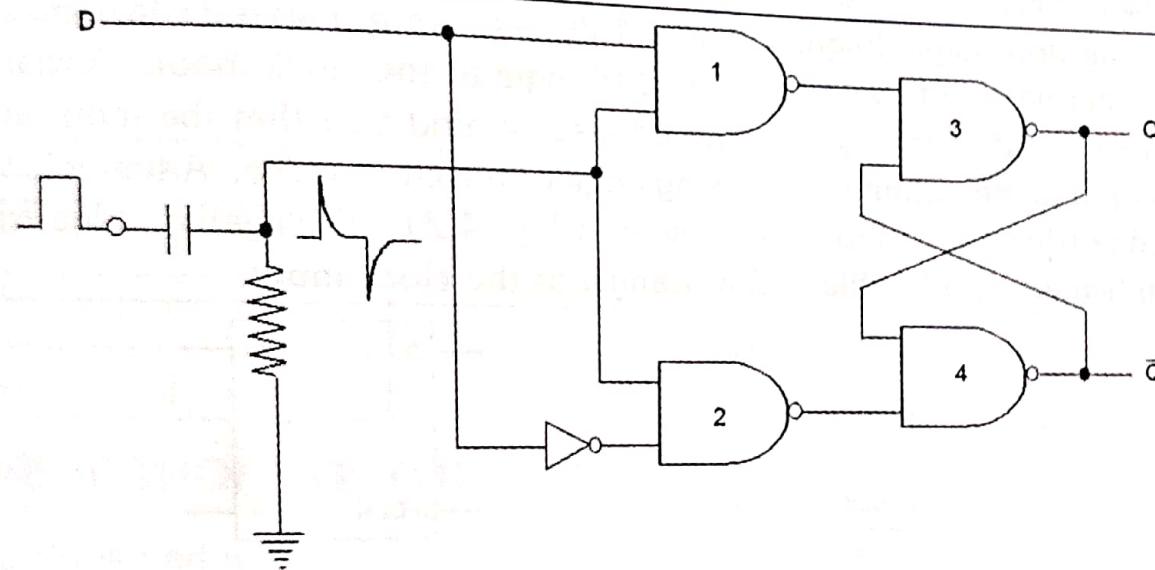


Fig. 4.20 D Flip-Flop (Edge Triggered)

If the RC time constant is much smaller than the width of the clock pulse, the capacitor will get charged rapidly when the clock signal goes high and discharge equally rapidly when the clock signal goes low. Thus the leading edge of the clock pulse will produce a narrow positive voltage spike across the resistor and the trailing edge of the clock pulse will produce a narrow negative spike.

NAND gates 1 and 2 will be enabled by the positive voltage spike and the negative voltage spike will have no effect on the operation of the flip-flop. The input signal at D will be sampled by the flip-flop for only a small fraction of the time, during which the clock pulse is high. The important point to note is that change in the output of the flip-flop will occur only on the rising edge of the clock pulse. Thus the flip-flop will be set or reset depending on the nature of the input signal at D. This is an example of positive edge-triggering, as triggering takes place on the positive edge of the clock pulse, which is also the leading edge.

The important distinction between edge-triggering and pulse-triggering (level-clocking) is that, with pulse triggering the output can change as long as the clock is active (high or low), that is during an entire half cycle. On the other hand with edge-triggering the output can change only during the rising or falling edge of a clock pulse, and that too only at one point in time of the clock cycle.

(T.S)

We have considered a very simple device to explain edge-triggering but, as stated earlier, it is not practical incorporate capacitors on chips. However, many edge-triggered circuits have been developed and edge-triggered flip-flops are commercially available.

As there are many types of flip-flops, it will be useful to acquire familiarity with the distinctive features of their symbols. The symbol for a positive edge-triggered D flip-flop (logic diagram in Fig. 4.20 is given in Fig. 4.21 (a). Positive edge-triggering is indicated by a small triangle at the clock input. A triangle at the clock input of the symbol should also remind you that the input at D will be stored at the output on the rising edge of the clock pulse. A symbol for negative edge-triggered D flip-flop is given in Fig. 4.21 (b) Negative edge-triggering is indicated by a bubble and a triangle at the clock input.

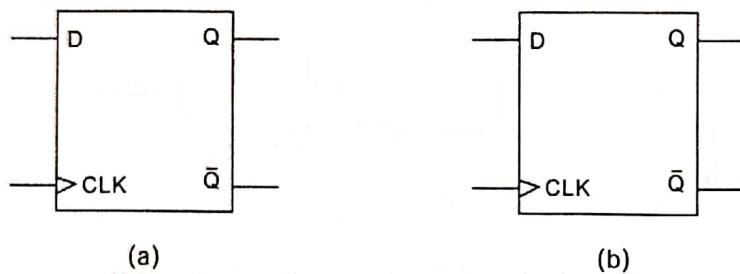


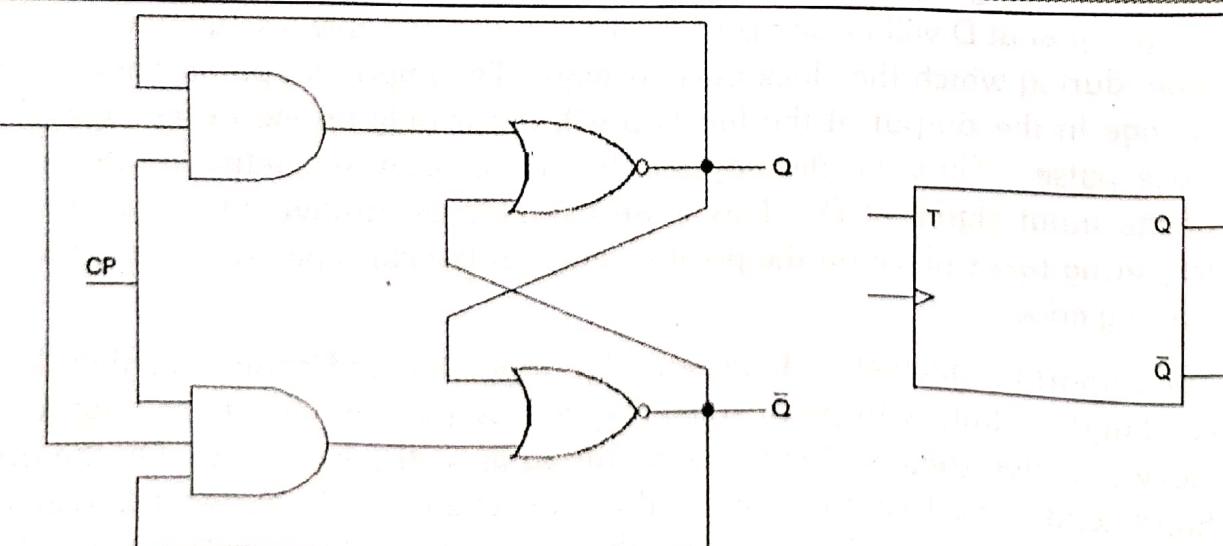
Fig. 4.21

4.13 T FLIP-FLOP

[Oct. 2007]

The T flip-flop is known as "**Toggle flip-flop**". The T flip flop is a modification of the JK flip flop. It is obtained from JK flip flop by connecting both inputs J and K together. Fig. 4.22 (a) shows the logic diagram of T flip flop, Fig. 4.22 (b) shows the logic symbol of T flip flop and the truth table of T flip flop is given below.

[Oct./Nov., 2012]



(a) Logic Diagram for Toggle Flip Flop

(b) Logic Symbol of T Flip Flop

Fig. 4.22

(A.P)

(T.S)

When $T = 0$, both AND gates are disabled and hence there is no change in the previous output. When $T = 1$ ($J = K = 1$) output toggles. Toggles means that the output is 0 when the previous state is 1 otherwise output is 1 when the previous state is 0. So the output is a complement of the previous output.

Truth Table of T Flip Flop

Present State	Flip-Flop Input	Next State
Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

4.14 APPLICATIONS OF FLIP-FLOPS

Flip-flops find wide applications in the following circuits :

- ◆ Counters
- ◆ Frequency dividers
- ◆ Shift and storage registers
- ◆ Serial decoding
- ◆ Comparators
- ◆ Monostable multivibrators
- ◆ Timing signal generation
- ◆ Data transfer
- ◆ Contact bounce elimination, etc.

4.15 INTRODUCTION TO COUNTERS

A counter is a sequential digital circuit whose output states progress in a predictable repeating pattern, advancing by one state for each clock pulse. Counters are one of the simplest types of sequential networks. A counter is usually constructed from two or more flip flops which change their state in a prescribed sequence when input clock pulses are received. This count sequence may be ascending, descending or non-linear. These procedures will be applied to more general types of sequential networks. If a counter has n -flip-flops, then it may count pulses upto 2^n .

(T.S)