

## MODULE-IV:

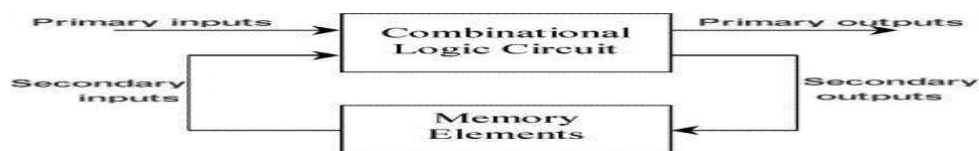
### Sequential Logic Circuits - I

#### Sequential circuits

**Classification of sequential circuits:** Sequential circuits may be classified as two types.

1. Synchronous sequential circuits
2. Asynchronous sequential circuits

Combinational logic refers to circuits whose output is strictly depended on the present value of the inputs. As soon as inputs are changed, the information about the previous inputs is lost, that is, combinational logics circuits have no memory. Although every digital system is likely to have combinational circuits, most systems encountered in practice also include memory elements, which require that the system be described in terms of sequential logic. Circuits whose output depends not only on the present input value but also the past input value are known as sequential logic circuits. The mathematical model of a sequential circuit is usually referred to as a sequential machine.



#### Comparison between combinational and sequential circuits

Combinational circuit	Sequential circuit
1. In combinational circuits, the output variables at any instant of time are dependent only on the present input variables	1. in sequential circuits the output variables at any instant of time are dependent not only on the present input variables, but also on the present state
2. memory unit is not requires in combinational circuit	2. memory unit is required to store the past history of the input variables
3. these circuits are faster because the delay between the i/p and o/p due to propagation delay of gates only	3. sequential circuits are slower than combinational circuits
4. easy to design	4. comparatively hard to design

### Level mode and pulse mode asynchronous sequential circuits:

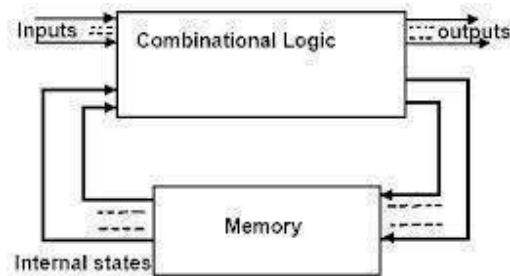


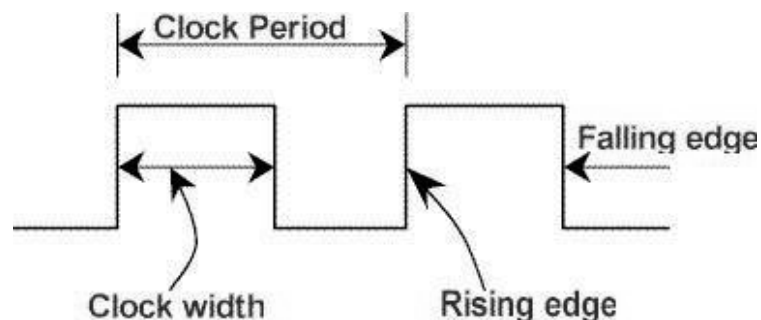
Figure 1: Asynchronous Sequential Circuit

Fig shows a block diagram of an asynchronous sequential circuit. It consists of a combinational circuit and delay elements connected to form the feedback loops. The present state and next state variables in asynchronous sequential circuits called secondary variables and excitation variables respectively..

There are two types of asynchronous circuits: fundamental mode circuits and pulse mode circuits.

### Synchronous and Asynchronous Operation:

Sequential circuits are divided into two main types: synchronous and asynchronous. Their classification depends on the timing of their signals. Synchronous sequential circuits change their states and output values at discrete instants of time, which are specified by the rising and falling edge of a free-running clock signal. The clock signal is generally some form of square wave as shown in Figure below.



From the diagram you can see that the clock period is the time between successive transitions in the same direction, that is, between two rising or two falling edges. State transitions in synchronous sequential circuits are made to take place at times when the clock is making a transition from 0 to 1 (rising edge) or from 1 to 0 (falling edge). Between successive clock pulses there is no change in the information stored in memory.

The reciprocal of the clock period is referred to as the clock frequency. The clock width is defined as the time during which the value of the clock signal is equal to 1. The ratio of the clock width and clock period is referred to as the duty cycle. A clock signal is said to

be active high if the state changes occur at the clock's rising edge or during the clock width. Otherwise, the clock is said to be active low. Synchronous sequential circuits are also known as clocked sequential circuits.

The memory elements used in synchronous sequential circuits are usually flip-flops. These circuits are binary cells capable of storing one bit of information. A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it. Binary information can enter a flip-flop in a variety of ways, a fact which give rise to the different types of flip-flops. For information on the different types of basic flip-flop circuits and their logical properties, see the previous tutorial on flip-flops.

In asynchronous sequential circuits, the transition from one state to another is initiated by the change in the primary inputs; there is no external synchronization. The memory commonly used in asynchronous sequential circuits are time-delayed devices, usually implemented by feedback among logic gates. Thus, asynchronous sequential circuits may be regarded as combinational circuits with feedback. Because of the feedback among logic gates, asynchronous sequential circuits may, at times, become unstable due to transient conditions. The instability problem imposes many difficulties on the designer. Hence, they are not as commonly used as synchronous systems.

**Fundamental Mode Circuits assumes that:**

1. The input variables change only when the circuit is stable
2. Only one input variable can change at a given time
3. Inputs are levels are not pulses

**A pulse mode circuit assumes that:**

1. The input variables are pulses instead of levels
2. The width of the pulses is long enough for the circuit to respond to the input
3. The pulse width must not be so long that is still present after the new state is reached.

### **Latches and flip-flops**

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. the flip-flops has two outputs, labeled Q and Q'. the Q output is the normal output of the flip flop and Q' is the inverted output.

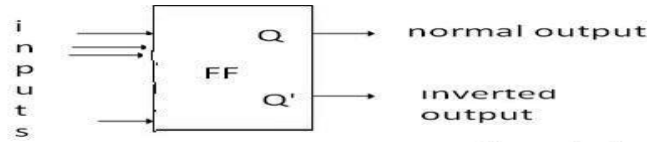


Figure: basic symbol of flipflop

A latch may be an active-high input latch or an active –LOW input latch. active – HIGH means that the SET and RESET inputs are normally resting in the low state and one of them will be pulsed high whenever we want to change latch outputs.

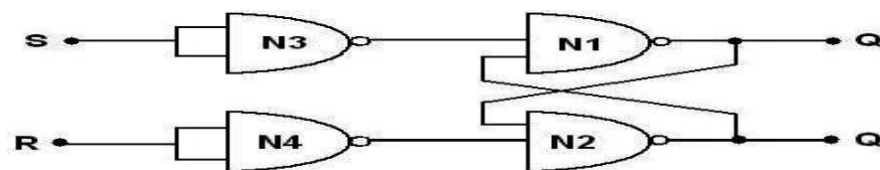
### SR latch:

The latch has two outputs Q and Q'. When the circuit is switched on the latch may enter into any state. If Q=1, then Q'=0, which is called SET state. If Q=0, then Q'=1, which is called RESET state. Whether the latch is in SET state or RESET state, it will continue to remain in the same state, as long as the power is not switched off. But the latch is not an useful circuit, since there is no way of entering the desired input. It is the fundamental building block in constructing flip-flops, as explained in the following sections

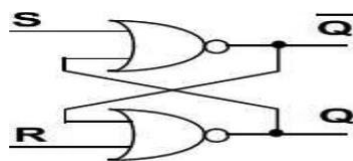
### NAND latch

NAND latch is the fundamental building block in constructing a flip-flop. It has the property of holding on to any previous output, as long as it is not disturbed.

The operation of NAND latch is the reverse of the operation of NOR latch. if 0's are replaced by 1's and 1's are replaced by 0's we get the same truth table as that of the NOR latch shown



### NOR latch



S	R	Q	Q'	Function
0	0	Q <sup>+</sup>	Q' <sup>+</sup>	Storage State
0	1	0	1	Reset
1	0	1	0	Set
1	1	0-?	0-?	Indeterminate State

The analysis of the operation of the active-HIGH NOR latch can be summarized as follows.

1. **SET=0, RESET=0:** this is normal resting state of the NOR latch and it has no effect on the output state. Q and Q' will remain in whatever state they were prior to the occurrence of this input condition.
2. **SET=1, RESET=0:** this will always set Q=1, where it will remain even after SET returns to 0
3. **SET=0, RESET=1:** this will always reset Q=0, where it will remain even after RESET returns to 0
4. **SET=1, RESET=1:** this condition tries to SET and RESET the latch at the same time, and it produces Q=Q'=0. If the inputs are returned to zero simultaneously, the resulting output state is erratic and unpredictable. This input condition should not be used.

The SET and RESET inputs are normally in the LOW state and one of them will be pulsed HIGH. Whenever we want to change the latch outputs..

### RS Flip-flop:

The basic flip-flop is a one bit memory cell that gives the fundamental idea of memory device. It is constructed using two NAND gates. The two NAND gates N1 and N2 are connected such that, output of N1 is connected to input of N2 and output of N2 to input of N1. These form the feedback path. The inputs are S and R, and outputs are Q and Q'. The logic diagram and the block diagram of R-S flip-flop with clocked input

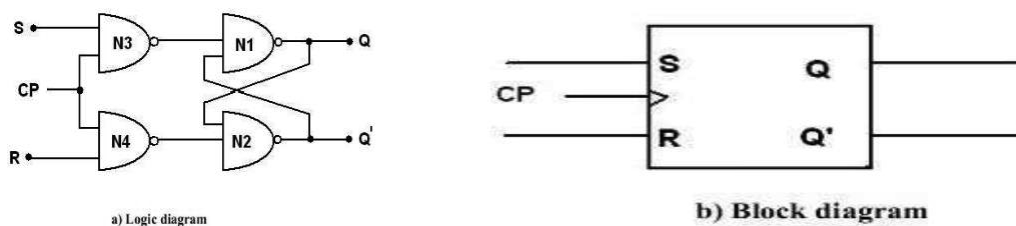


Figure: RS Flip-flop

The flip-flop can be made to respond only during the occurrence of clock pulse by adding two NAND gates to the input latch. So synchronization is achieved. i.e., flip-flops are allowed to change their states only at particular instant of time. The clock pulses are generated by a clock pulse generator. The flip-flops are affected only with the arrival of clock pulse.

### Operation:

1. When CP=0 the output of N3 and N4 are 1 regardless of the value of S and R. This is given as input to N1 and N2. This makes the previous value of Q and Q' unchanged.
2. When CP=1 the information at S and R inputs are allowed to reach the latch and change of state in flip-flop takes place.
3. CP=1, S=1, R=0 gives the SET state i.e., Q=1, Q'=0.

4.  $CP=1, S=0, R=1$  gives the RESET state i.e.,  $Q=0, Q'=1$ .

5.  $CP=1, S=0, R=0$  does not affect the state of flip-flop.

6.  $CP=1, S=1, R=1$  is not allowed, because it is not able to determine the next state. This condition is said to be a —race conditionll.

In the logic symbol CP input is marked with a triangle. It indicates the circuit responds to an input change from 0 to 1. The characteristic table gives the operation conditions of flip-flop.  $Q(t)$  is the present state maintained in the flip-flop at time  $t$ .  $Q(t+1)$  is the state after the occurrence of clock pulse.

Truth table

S	R	$Q_{(t+1)}$	Comments
0	0	$Q_t$	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	*	Not allowed

### Edge triggered RS flip-flop:

Some flip-flops have an RC circuit at the input next to the clock pulse. By the design of the circuit the R-C time constant is much smaller than the width of the clock pulse. So the output changes will occur only at specific level of clock pulse. The capacitor gets fully charged when clock pulse goes from low to high. This change produces a narrow positive spike. Later at the trailing edge it produces narrow negative spike. This operation is called edge triggering, as the flip-flop responds only at the changing state of clock pulse. If output transition occurs at rising edge of clock pulse (0  $\rightarrow$  1), it is called positively edge triggering. If it occurs at trailing edge (1  $\rightarrow$  0) it is called negative edge triggering. Figure shows the logic and block diagram.

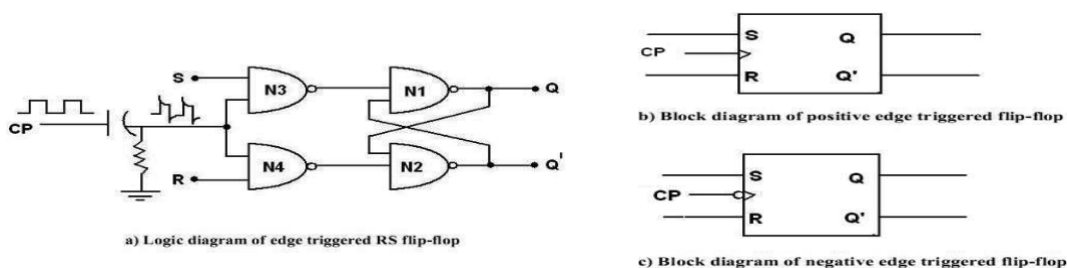
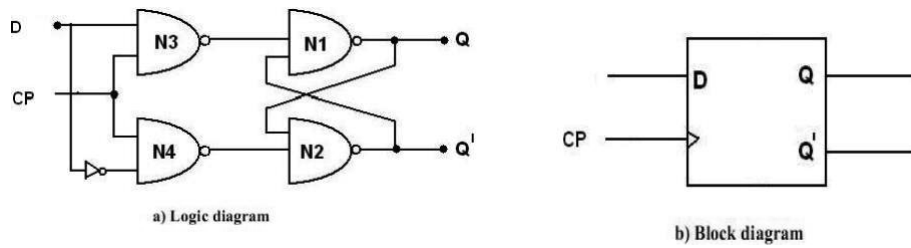


Figure: Edge triggered RS flip-flop

### D flip-flop:

The D flip-flop is the modified form of R-S flip-flop. R-S flip-flop is converted to D flip-flop by adding an inverter between S and R and only one input D is taken instead of S and R. So one input is D and complement of D is given as another input. The logic diagram and the block diagram of D flip-flop with clocked input

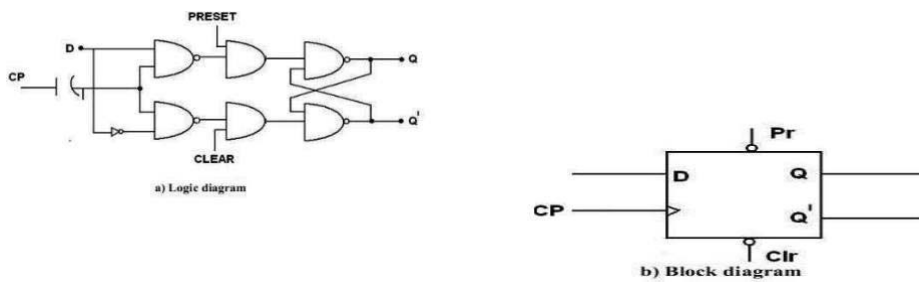


When the clock is low both the NAND gates (N1 and N2) are disabled and Q retains its last value. When clock is high both the gates are enabled and the input value at D is transferred to its output Q. D flip-flop is also called —Data flip-flop.

Truth table

CP	D	Q
0	x	Previous state
1	0	0
1	1	1

**Edge Triggered D Flip-flop:**

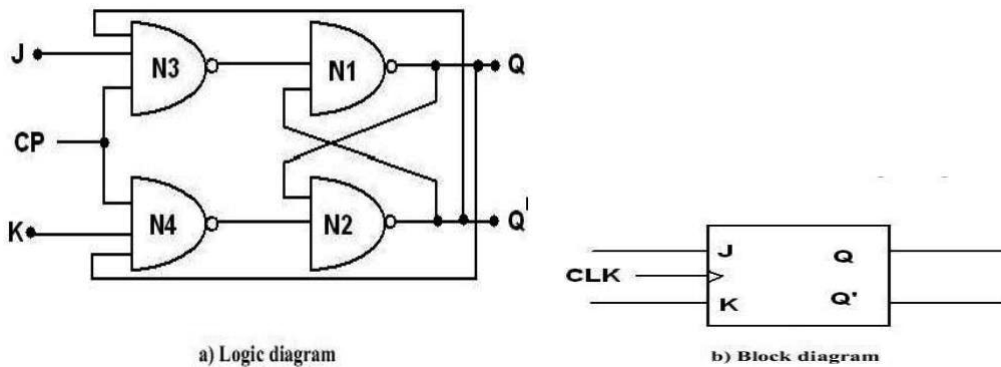


Truth table

PRESET	CLEAR	CP	D	Q
0	0	X	X	*(forbidden)
0	1	X	X	1
1	0	X	X	0
1	1	0	0	Z
1	1	1	0	0
1	1	↓	X	0
1	1	↑	0	1
1	1	↑	1	1

**Figure: truth table, block diagram, logic diagram of edge triggered flip-flop JK flip-flop (edge triggered JK flip-flop)**

The race condition in RS flip-flop, when R=S=1 is eliminated in J-K flip-flop. There is a feedback from the output to the inputs. Figure 3.4 represents one way of building a JK flip-flop.



Truth table

J	K	$Q_{(t+1)}$	Comments
0	0	$Q_t$	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	$Q_t'$	Complement/ toggle.

Figure: JK flip-flop

The J and K are called control inputs, because they determine what the flip-flop does when a positive clock edge arrives.

Operation:

1. When  $J=0$ ,  $K=0$  then both N3 and N4 will produce high output and the previous value of Q and Q' retained as it is.

2. When  $J=0$ ,  $K=1$ , N3 will get an output as 1 and output of N4 depends on the value of Q. The final output is  $Q=0$ ,  $Q'=1$  i.e., reset state

3. When  $J=1$ ,  $K=0$  the output of N4 is 1 and N3 depends on the value of Q'. The final output is  $Q=1$  and  $Q'=0$  i.e., set state

4. When  $J=1$ ,  $K=1$  it is possible to set (or) reset the flip-flop depending on the current state of output. If  $Q=1$ ,  $Q'=0$  then N4 passes '0' to N2 which produces  $Q'=1$ ,  $Q=0$  which is reset state. When  $J=1$ ,  $K=1$ , Q changes to the complement of the last state. The flip-flop is said to be in the toggle state.

The characteristic equation of the JK flip-flop is:

$$Q_{next} = J\bar{Q} + \bar{K}Q$$



JK flip-flop operation<sup>[28]</sup>

<u>Characteristic table</u>		<u>Excitation table</u>						
J	K	Q <sub>next</sub>	Comment	Q	Q <sub>next</sub>	J	K	Comment
0	0	Q	hold state	0	0	0	X	No change
0	1	0	reset	0	1	1	X	Set
1	0	1	set	1	0	X	1	Reset
1	1	Q	toggle	1	1	X	0	No change

## T flip-flop:

If the T input is high, the T flip-flop changes state ("toggles") whenever the clock input is strobed. If the T input is low, the flip-flop holds the previous value. This behavior is described by the characteristic equation

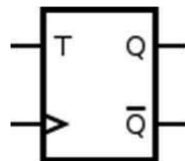


Figure : symbol for T flip flop

$$Q_{next} = T \oplus Q = T\bar{Q} + \bar{T}Q \text{ (expanding the XOR operator)}$$

When T is held high, the toggle flip-flop divides the clock frequency by two; that is, if clock frequency is 4 MHz, the output frequency obtained from the flip-flop will be 2 MHz. This "divide by" feature has application in various types of digital counters. A T flip-flop can also be built using a JK flip-flop (J & K pins are connected together and act as T) or D flip-flop (T input and P<sub>previous</sub> is connected to the D input through an XOR gate).

T flip-flop operation<sup>[28]</sup>

T flip-flop operation <sup>[28]</sup>							
<u>Characteristic table</u>				<u>Excitation table</u>			
<i>T</i>	<i>Q</i>	<i>Q<sub>next</sub></i>	Comment	<i>Q</i>	<i>Q<sub>next</sub></i>	<i>T</i>	Comment
0	0	0	hold state (no clk)	0	0	0	No change
0	1	1	hold state (no clk)	1	1	0	No change
1	0	1	toggle	0	1	1	Complement
1	1	0	toggle	1	0	1	Complement

## Flip flop operating characteristics:

The operation characteristics specify the performance, operating requirements, and operating limitations of the circuits. The operation characteristics mentions here apply to all flip-flops regardless of the particular form of the circuit.

**Propagation Delay Time:** is the interval of time required after an input signal has been applied for the resulting output change to occur.

**Set-up Time:** is the minimum interval required for the logic levels to be maintained constantly on the inputs (J and K, or S and R, or D) prior to the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop.

**Hold Time:** is the minimum interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop.

**Maximum Clock Frequency:** is the highest rate that a flip-flop can be reliably triggered. **Power Dissipation:** is the total power consumption of the device. It is equal to product of supply voltage ( $V_{cc}$ ) and the current ( $I_{cc}$ ).

$$P = V_{cc} \cdot I_{cc}$$

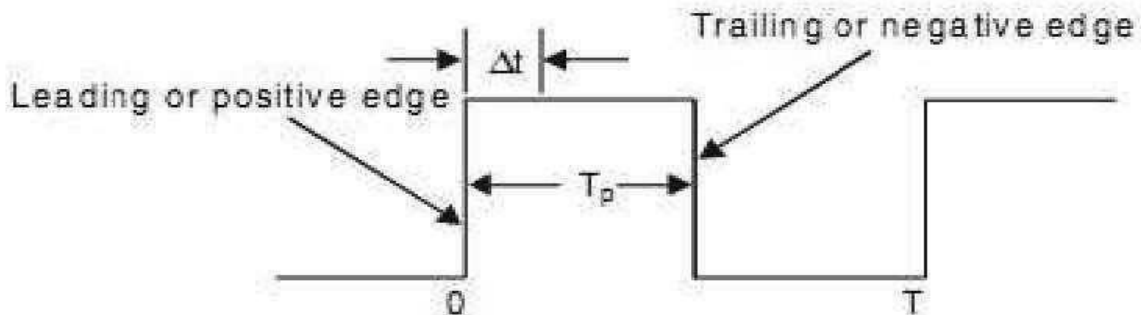
The power dissipation of a flip flop is usually in mW.

**Pulse Widths:** are the minimum pulse widths specified by the manufacturer for the Clock, SET and CLEAR inputs.

**Clock transition times:** for reliable triggering, the clock waveform transition times should be kept very short. If the clock signal takes too long to make the transitions from one level to other, the flip flop may either triggering erratically or not trigger at all.

### Race around Condition

The inherent difficulty of an S-R flip-flop (i.e.,  $S = R = 1$ ) is eliminated by using the feedback connections from the outputs to the inputs of gate 1 and gate 2 as shown in Figure. Truth tables in figure were formed with the assumption that the inputs do not change during the clock pulse ( $CLK = 1$ ). But the consideration is not true because of the feedback connections



Consider, for example, that the inputs are  $J = K = 1$  and  $Q = 1$ , and a pulse as shown in Figure is applied at the clock input.

After a time interval  $t$  equal to the propagation delay through two NAND gates in series, the outputs will change to  $Q = 0$ . So now we have  $J = K = 1$  and  $Q = 0$ .

After another time interval of  $t$  the output will change back to  $Q = 1$ . Hence, we conclude that for the time duration of  $tP$  of the clock pulse, the output will oscillate between 0 and 1. Hence, at the end of the clock pulse, the value of the output is not certain. This situation is referred to as a race-around condition.

Generally, the propagation delay of TTL gates is of the order of nanoseconds. So if the clock pulse is of the order of microseconds, then the output will change thousands of times within the clock pulse.

This race-around condition can be avoided if  $t_p < t < T$ . Due to the small propagation delay of the ICs it may be difficult to satisfy the above condition.

A more practical way to avoid the problem is to use the master-slave (M-S) configuration as discussed below.

### Applications of flip-flops:

**Frequency Division:** When a pulse waveform is applied to the clock input of a J-K flip-flop that is connected to toggle, the Q output is a square wave with half the frequency of the clock input. If more flip-flops are connected together as shown in the figure below, further division of the clock frequency can be achieved.

**Parallel data storage:** a group of flip-flops is called register. To store data of  $N$  bits,  $N$  flip-flops are required. Since the data is available in parallel form. When a clock pulse is applied to all flip-flops simultaneously, these bits will transfer will be transferred to the Q outputs of the flip flops.

**Serial data storage:** to store data of  $N$  bits available in serial form,  $N$  number of D-flip-flops is connected in cascade. The clock signal is connected to all the flip-flops. The serial data is applied to the D input terminal of the first flip-flop.

Transfer of data: data stored in flip-flops may be transferred out in a serial fashion, i.e., bit-by-bit from the output of one flip-flops or may be transferred out in parallel form.

### Excitation Tables:

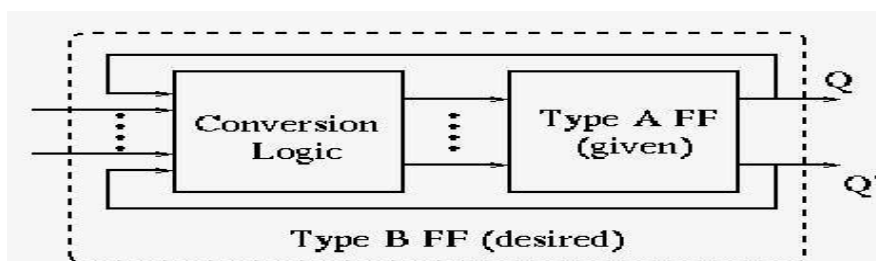
Previous State -> Present State	D
0 -> 0	0
0 -> 1	1
1 -> 0	0
1 -> 1	1

Previous State -> Present State	J	K
0 -> 0	0	X
0 -> 1	1	X
1 -> 0	X	1
1 -> 1	X	0

Previous State -> Present State	S	R
0 -> 0	0	X
0 -> 1	1	0
1 -> 0	0	1
1 -> 1	X	0

Previous State -> Present State	T
0 -> 0	0
0 -> 1	1
1 -> 0	1
1 -> 1	0

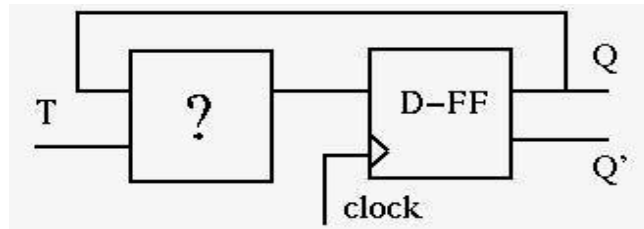
### Conversions of flip-flops:



The key here is to use the excitation table, which shows the necessary triggering signal (S,R,J,K, D and T) for a desired flip-flop state transition :

$Q_t$	$Q_{t+1}$	S	R	J	K	D	T
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

Convert a D-FF to a T-FF:



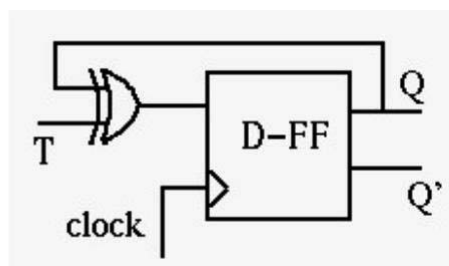
We need to design the circuit to generate the triggering signal D as a function of T and Q:  
 . Consider the excitation table:

$$D = f(T, Q).$$

$Q_t$	$Q_{t+1}$	T	D
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

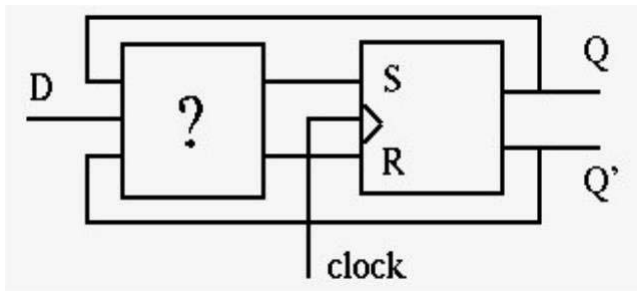
Treating as a function of and current FF state , we have

$$D = T'Q + TQ' = T \oplus Q$$



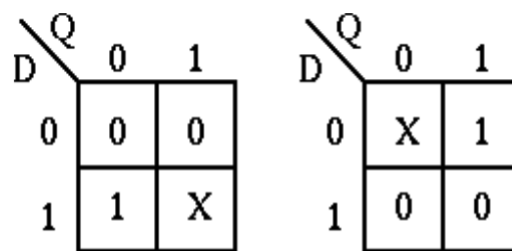
Convert a RS-FF to a D-FF:

We need to design the circuit to generate the triggering signals S and R as functions of and consider the excitation table:



$Q_t$	$Q_{t+1}$	D	S	R
0	0	0	0	x
0	1	1	1	0
1	0	0	0	1
1	1	1	x	0

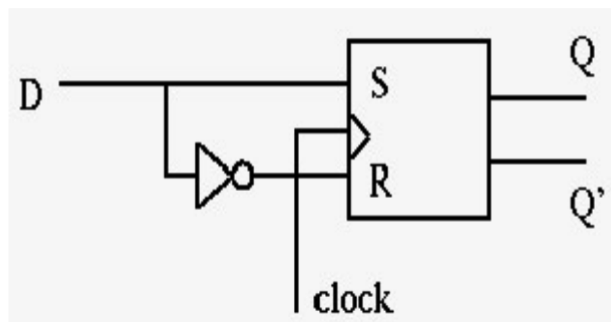
The desired signal and can be obtained as functions of and current FF state from the Karnaugh maps:



$S = D$

$R = D'$

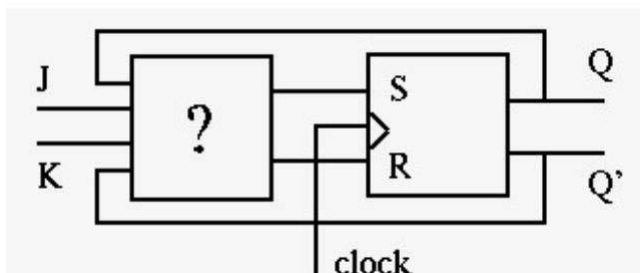
$S = D, \quad R = D'$



Convert a RS-FF to a JK-FF:

We need to design the circuit to generate the triggering signals S and R as functions of, J, K.

Consider the excitation table: The desired signal and as functions of, and current FF state can be obtained from the Karnaugh maps:



$Q_t$	$Q_{t+1}$	J	K	S	R
0	0	0	x	0	x
0	1	1	x	1	0
1	0	x	1	0	1
1	1	x	0	x	0

K-maps:

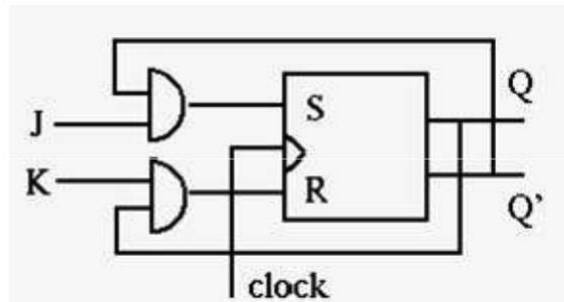
		QJ			
		00	01	11	10
K	0	0	1	X	X
	1	0	1	0	0

$$S = Q'J$$

		QJ			
		00	01	11	10
K	0	X	0	0	0
	1	X	0	1	1

$$R = QK$$

$$S = Q'J, \quad R = QK$$



The Master-Slave JK Flip-flop:

The Master-Slave Flip-Flop is basically two gated SR flip-flops connected together in a series configuration with the slave having an inverted clock pulse. The outputs from Q and Q' from the "Slave" flip-flop are fed back to the inputs of the "Master" with the outputs of the "Master" flip-flop being connected to the two inputs of the "Slave" flip-flop. This feedback configuration from the slave's output to the master's input gives the characteristic toggle of the JK flip-flop as shown below.

The input signals J and K are connected to the gated "master" SR flip-flop which "locks" the input condition while the clock (Clk) input is "HIGH" at logic level "1". As the clock input of the "slave" flip-flop is the inverse (complement) of the "master" clock input, the "slave" SR flip-flop does not toggle. The outputs from the "master" flip-flop are only "seen" by the gated "slave" flip-flop when the clock input goes "LOW" to logic level "0". When the clock is "LOW", the outputs from the "master" flip-flop are latched and any additional changes to its inputs are ignored. The gated "slave" flip-flop now responds to the state of its inputs passed over by the "master" section. Then on the "Low-to-High" transition of the clock pulse the inputs of the "master" flip-flop are fed through to the gated inputs of the "slave" flip-flop and on the "High-to-Low" transition the same inputs are reflected on the output of the "slave" making this type of flip-flop edge or pulse-triggered. Then, the circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal. In other words, the Master-Slave JK Flip-flop is a "Synchronous" device as it only passes data with the timing of the clock signal.