SEQUENTIAL CIRCUITS

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

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

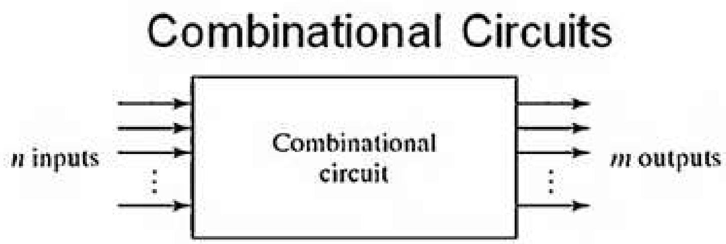
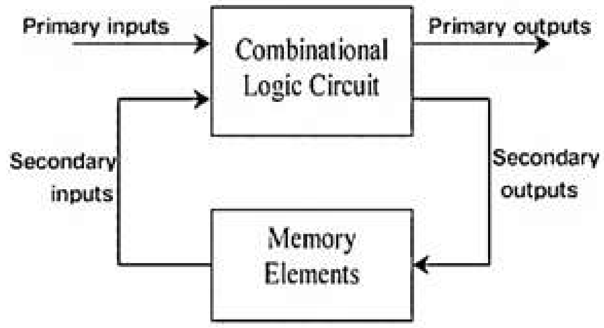


Fig. Block Diagram of Combinational Circuit

Sequential Circuits



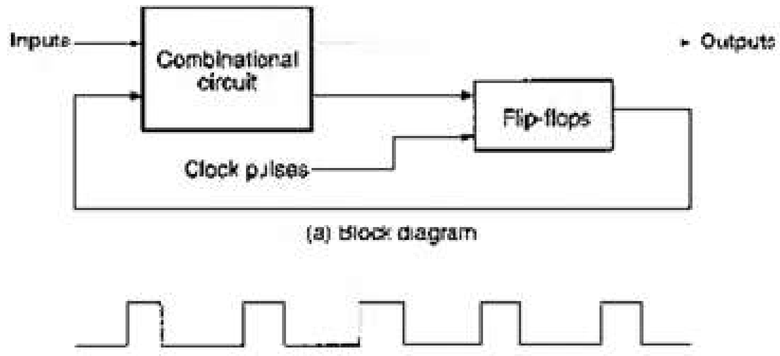
The memory elements are devices capable of storing binary info. The binary info stored in the memory elements at any given time defines the state of the sequential circuit. The input and the present state of the memory element determine the output. Memory elements next state is also a function of external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

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

- ❖ Synchronous sequential circuits
- ❖ Asynchronous sequential circuits

Asynchronous sequential circuit

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



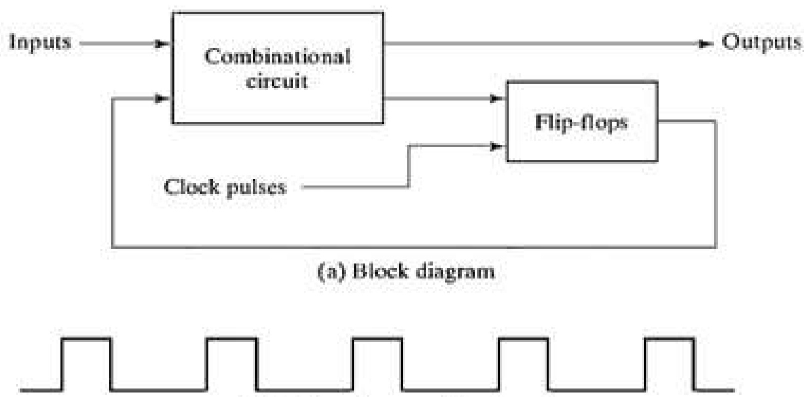
(a) Block diagram

(b) Timing diagram of clock pulses

Synchronous sequential circuits

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

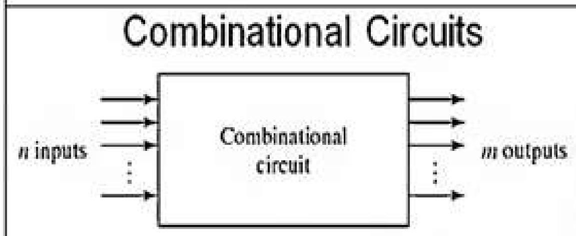
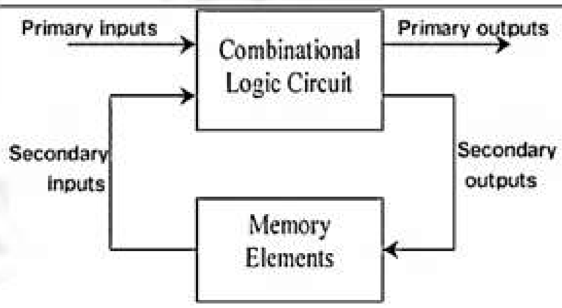
Synchronous sequential circuits use logic gates and flip-flop storage devices. Sequential circuits have a clock signal as one of their inputs. All state transitions in such circuits occur only when the clock value is either 0 or 1 or happen at the rising or falling edges of the clock depending on the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



(a) Block diagram

(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

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

A sequential circuit can further be categorized into **Synchronous** and **Asynchronous**. Here is the difference between synchronous and asynchronous sequential circuits:

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

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

Basic Flip Flops:

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

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

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

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

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

1. RS Latch

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

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

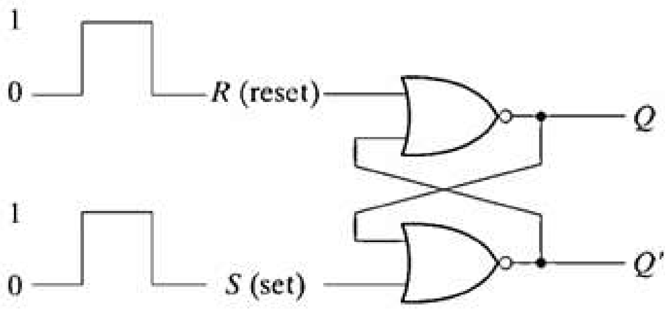
1. **The basic Flip Flop or S-R Flip Flop**
2. **Delay Flip Flop [D Flip Flop]**
3. **J-K Flip Flop**
4. **T Flip Flop**

2. S-R Flip Flop:

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

S-R Flip Flop using NOR Gate

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



(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

Fig. 5-3 *SR* Latch with NOR Gates

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

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

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

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

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

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

In this state When R=1 it resets the output. So this state is known as the RESET state. In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the value of S.

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

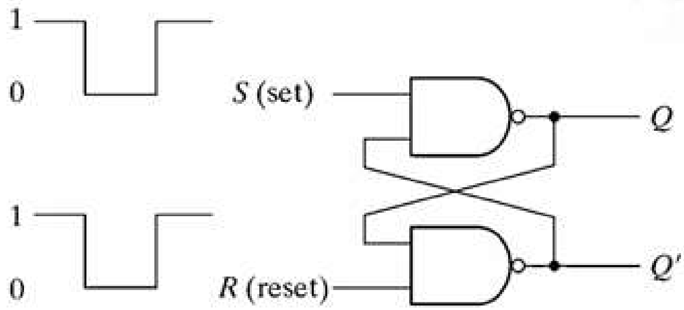
If both the values of S and R are switched to 0, then the circuit remembers the value of S and R in their previous state.

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

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

S-R Flip Flop using NAND Gate

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



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

(b) Function table

Fig. 5-4 *SR* Latch with NAND Gates

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

1. **S=1, R=0, Q=0, Q'=1**
   This state is also called the SET state.

2. **S=0, R=1, Q=1, Q'=0**
   This state is known as the RESET state.
   In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the compliment value of S.

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

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

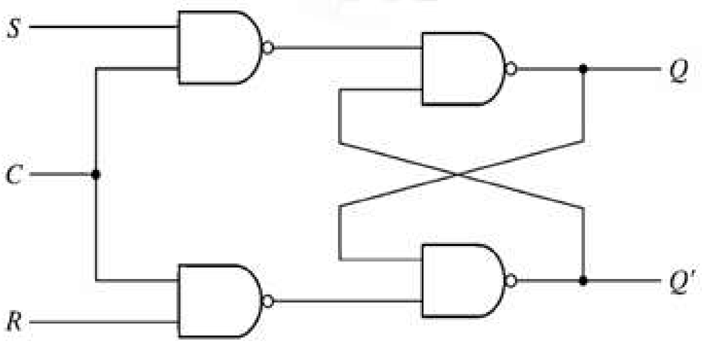
4. S=1, R=1, Q & Q'= Remember

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

Clocked S-R Flip Flop

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

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



(a) Logic diagram

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

(b) Function table

Fig. 5-5 SR Latch with Control Input

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

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

Excitation Table of the SR Latch

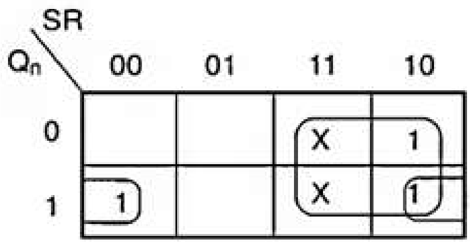
- During the design process we usually know the transition from present state to next state and wish to find the latch input conditions that will cause the required transition.
- For this reason, we need a table that lists the required inputs for a given change of state. Such a table is called an excitation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

Excitation Table:

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

Note: Indeter = not used

K Map for Q_{n+1}:



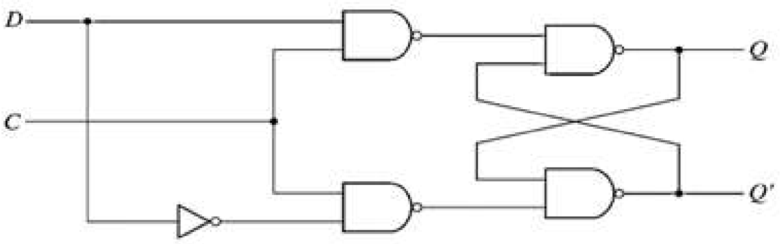
$$Q_{n+1} = S + \overline{R}.Q_n$$

3. D Flip Flop

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

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

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



(a) Logic diagram

C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

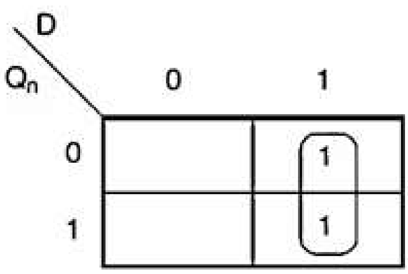
(b) Function table

Fig. 5-6 D Latch

Excitation Table:

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

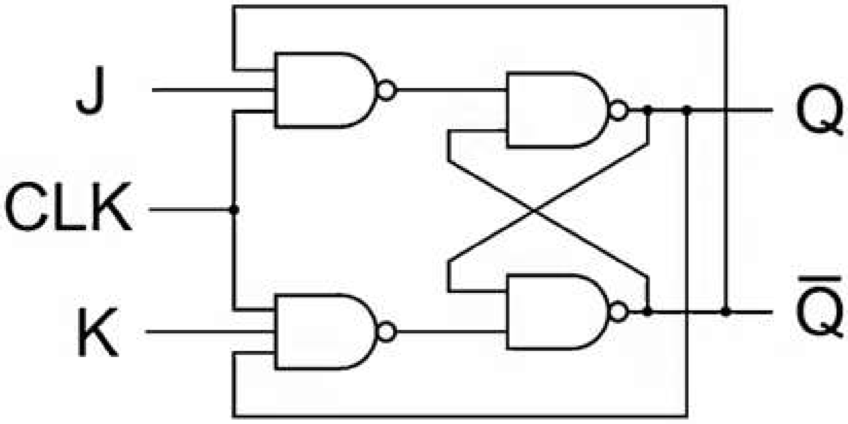
K- Map for Q_{n+1}:



$Q_{n+1} = D$

3. J-K Flip Flop

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



❖ The behavior of inputs J and K is same as the S and R inputs of the S-R flip flop. The letter J stands for SET and the letter K stands for CLEAR.

❖ When both the inputs J and K have a HIGH state, the flip-flop switches to the complement state. So, for a value of Q = 1, it switches to Q=0 and for a value of Q = 0, it switches to Q=1.

❖ The circuit includes two 3-input AND gates. The output Q of the flip flop is returned back as a feedback to the input of the AND along with other inputs like K and clock pulse [CP].

❖ So, if the value of CP is '1', the flip flop gets a CLEAR signal and with the condition that the value of Q was earlier 1.

❖ Similarly output Q' of the flip flop is given as a feedback to the input of the AND along with other inputs like J and clock pulse [CP].

❖ So the output becomes SET when the value of CP is 1 only if the value of Q' was earlier 1.

❖ The output may be repeated in transitions once they have been complimented for J=K=1 because of the feedback connection in the JK flip-flop.

❖ This can be avoided by setting a time duration lesser than the propagation delay through the flip-flop.

❖ The restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction.

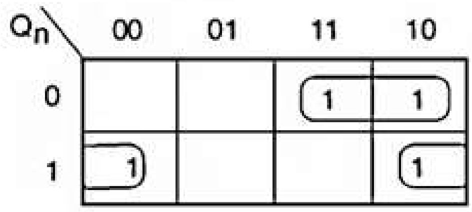
Characteristic table:

Clk	J	K	Qn+1
0	X	X	Memory
1	0	0	Memory
1	0	1	0
1	1	0	1 (Set)
1	1	1	Toggle

Excitation table for JK Flipflop

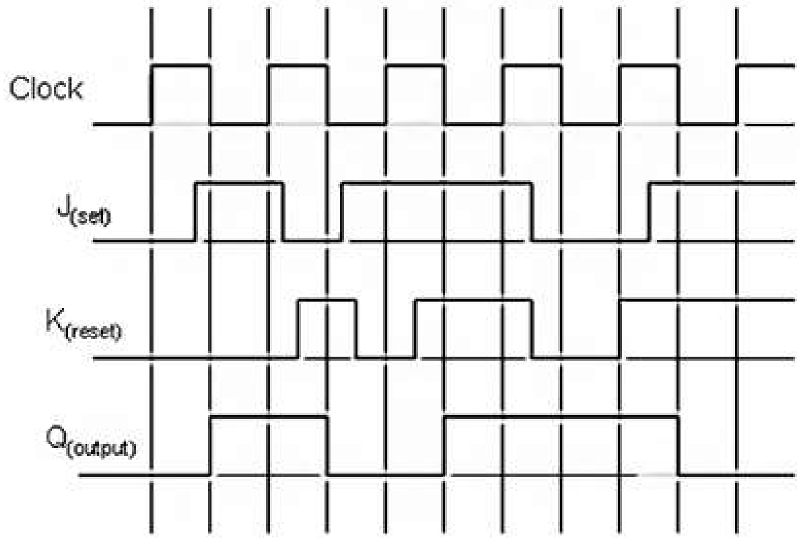
Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

K map for Qn+1:



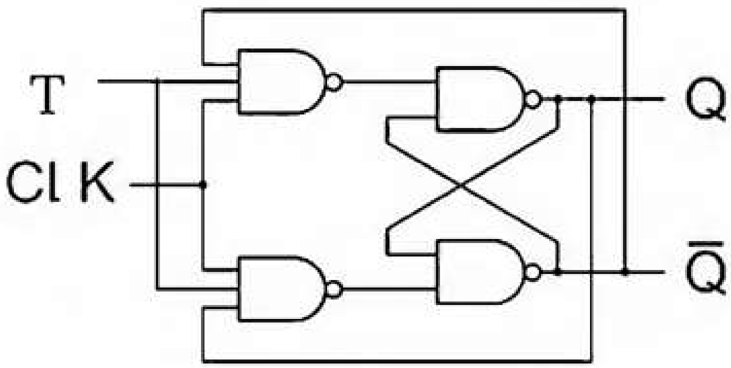
$$Q_{n+1} = J.\overline{Q_n} + \overline{K}.Q_n$$

Timing Diagram:



4. T Flip Flop

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



,Excitation Table for T Flip Flop:

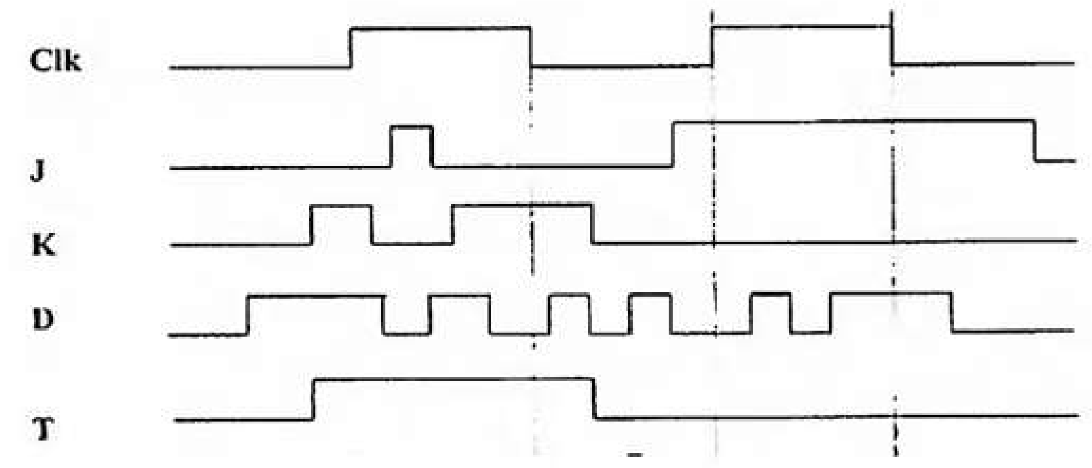
Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

K map for T Flip Flop:

Q_n \ T	0	1
0		1
1	1	

Characteristic Equation:

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


Clk
J
K
D
T