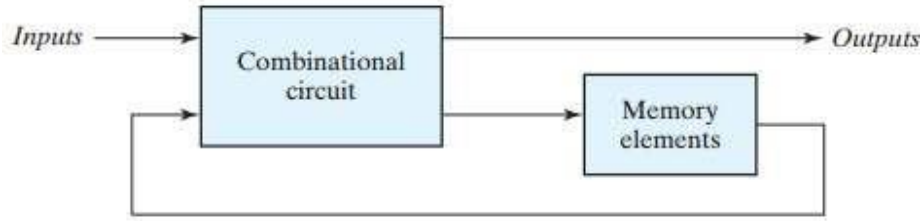
UNIT -4
SEQUENTIAL LOGIC
TOPIC – 1
INTRODUCTION TO SEQUENTIAL CIRCUITS

In a Combinational circuits, the output appears immediately for a change in input, except for the propagation delay through circuit gates.

On the other hand, the logic circuits whose outputs at any instant of time depend on the present inputs as well as on the past outputs are called sequential circuits. In sequential circuits, the output signals are fed back to the input side.

A block diagram of a sequential circuit is shown in Figure below:-



- It consists of a combinational circuit to which storage elements are connected to form a feedback path.
- The storage elements are devices capable of storing binary information.
- The binary information stored in these elements at any given time defines the ***state*** of the sequential circuit at that time.
- The sequential circuit receives binary information from external inputs that, together with the present state of the storage elements, determine the binary value of the outputs.
- These external inputs also determine the condition for changing the state in the storage elements. The block diagram demonstrates that the outputs in a sequential circuit are a function not onlyof the inputs, but also of the present state of the storage elements.
- The next state of the storage elements is also a function of external inputs and the present state. Thus, **a sequential circuit is specified by a time sequence of inputs, outputs, and internal states.**

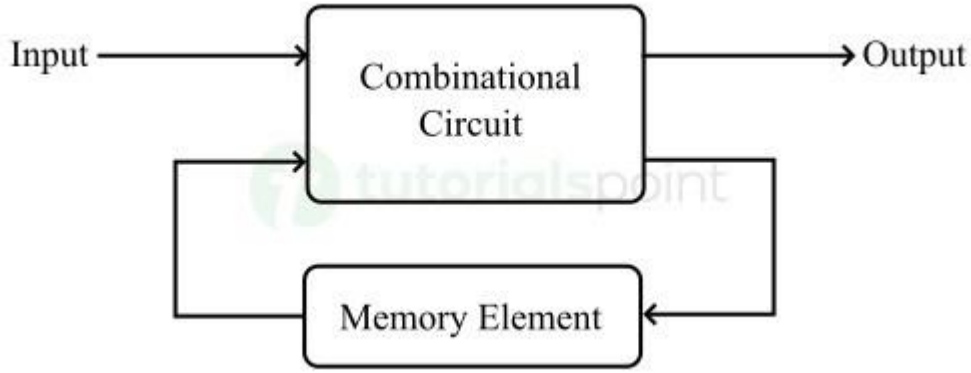
Types Of Sequential Circuits:

There are two types of sequential circuits, and their classification is a function of the timing of their signals.

1. ***Asynchronous sequential circuits***
2. ***Synchronous sequential circuits***

Asynchronous sequential circuit:

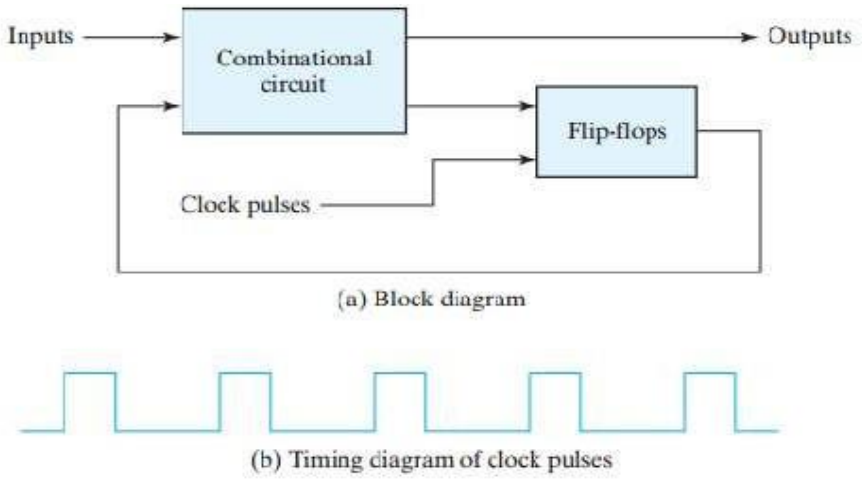
- A sequential circuit whose behavior depends upon the sequence in which the input signals change is referred to as an ***Asynchronous sequential circuit***.
- The output will be affected whenever the input changes.
- The commonly used memory elements in these circuits are time-delay devices.



- There is no need to wait for a clock pulse. Therefore, in general, asynchronous circuits are faster than synchronous sequential circuits.
- However, in an asynchronous circuit, events are allowed to occur without any synchronization and in such a case, the system becomes unstable.
- Since the designs of asynchronous circuits are more tedious and difficult, their uses are rather limited. The memory elements used in sequential circuits are flip-flops which are capable of storing binary information.

Synchronous sequential circuit:

- A sequential circuit whose behavior can be defined from the knowledge of its signal at discrete instants of timeis referred to as a ***synchronous sequential circuit***.
- In these systems, the memory elements are affected only at discrete instants of time.
- The synchronization is achieved by a timing device known as a system clock, which generates a periodic train of lock pulses. The outputs are affected only with the application of a clock pulse.



(a) Block diagram

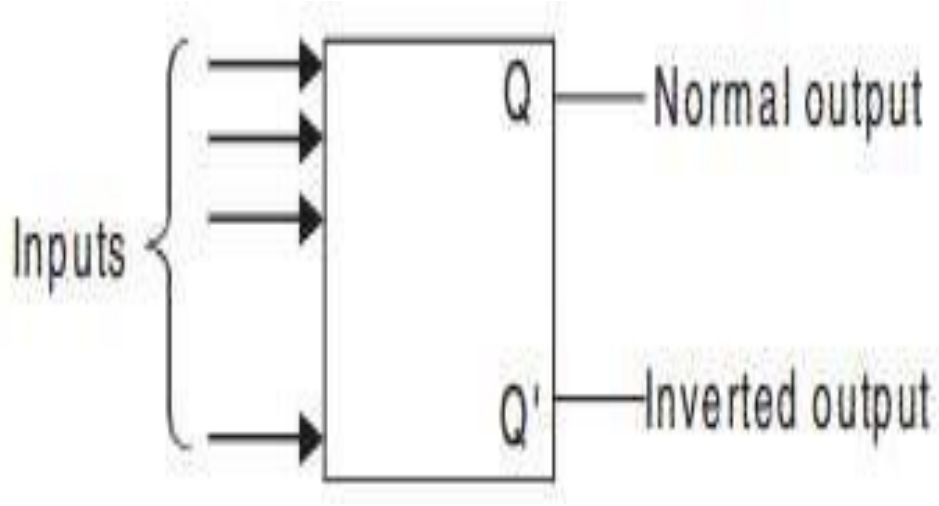
(b) Timing diagram of clock pulses

Synchronous clocked sequential circuit

- The storage elements (memory) used in clocked sequential circuits are called ***flip-flops***

Flip-Flop:

- The basic 1-bit digital memory circuit is known as a flip-flop.
- It can have only two states, either the 1 state or the 0 state. Flip-flops can be obtained by using NAND or NOR gates.
- It has one or more inputs and two outputs. The two outputs are complementary to each other.
- If Q is 1 i.e., Set, then Q' is 0; if Q is 0 i.e., Reset, then Q' is 1. That means Q and Q' cannot be at the same state simultaneously.



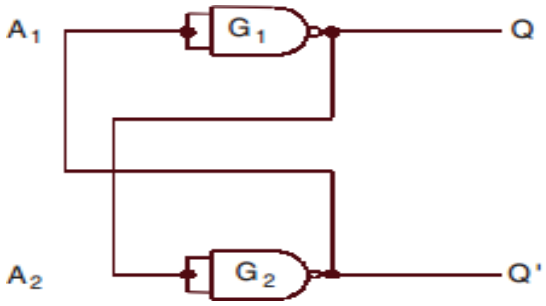
- If it happens by any chance, it violates the definition of a flip-flop and hence is called an undefined condition.
- Normally, the state of Q is called the state of the flip-flop, whereas the state of Q' is called the complementary state of the flip- flop.
- When the output Q is either 1 or 0, it remains in that state unless one or more inputs are excited to effect a change in the output.
- Since the output of the flip-flop remains in the same state until the trigger pulse is applied to change the state, it can be referred as a memory device to store one binary bit.

UNIT -4
SEQUENTIAL LOGIC
TOPIC – 2
LATCHES

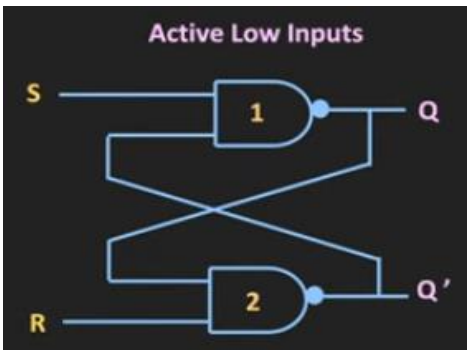
The basic difference between a latch & flip-flop is, Storage elements that operate with signal levels (rather than signal transitions) are referred to as **latches**; those controlled by a clock transition are **flip-flops**. Latches aresaid to be level sensitive devices; flip-flops are edge-sensitive devices.

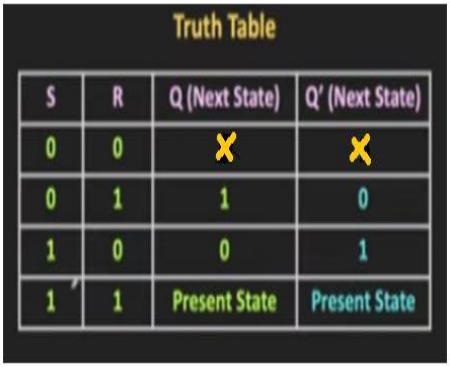
The two types of storage elements are related because latches are the basic circuits from which all flip-flops are constructed.

It consists of two inverters G_1 and G_2 (NAND gates are used as inverters). The output of G_1 is connected to the input of G_2 (A_2) and the output of G_2 is connected to the input of G_1 (A_1).



- Let us assume the output of G1 to be Q = 0, which is also the input of G2 (A2 = 0).
- So, the output of G2 will be Q'= 1, which makes A1 = 1 and consequently Q = 0 which is according to our assumption.
- Similarly, we can demonstrate that if Q = 1, then Q' = 0 and this is also consistent with the circuit connections. Hence we see that Q and Q' are always complementary and if the circuit is in 1 state, it continues to remain in this state and vice versa is also true.
- Since this information is locked or latched in this circuit, therefore, this circuit is also referred to as a latch.
- In this circuit there is no way to enter the desired digital information to be stored in it. To make that possible we have to modify the circuit by replacing the inverters by NAND gates and then it becomes a flip-flop.



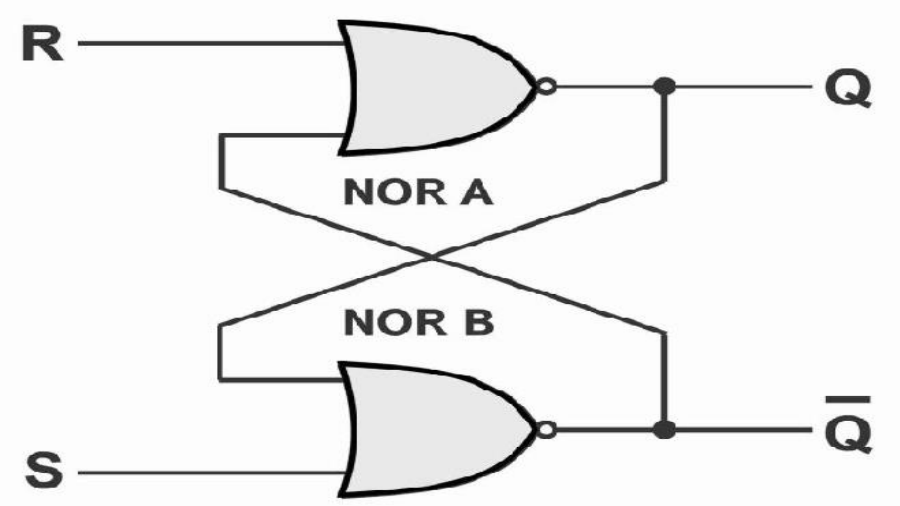


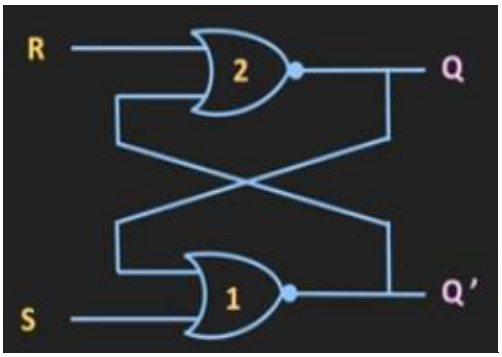
- The SR latch has two inputs (SET and RESET) for setting and resetting the 1-bit of information.

- It has two complementary outputs Q and Q'.
- When S = 0 and R = 1 then latch Sets the output to logic '1'.
- When S = 1 and R = 0 then latch Resets the output to logic '0'.
- And when both S = R= 1 then latch retain the currently stored information.
- In the active low SR latch, S = 0 and R = 0 input combination is forbidden.

NOR Latch:

- An S-R Latch can be constructed with NOR gates at ease by connecting the NOR gates back to back as shown in Figure .
- The cross-coupled connections from the output of gate 1 to the input of gate 2 constitute a feedback path.
- This circuit is not clocked and is classified as an asynchronous sequential circuit.





Truth Table

S	R	Q (Next State)	Q'(Next State)
0	0	Present State	Present State
0	1	0	1
1	0	1	0
1	1	✕	✕

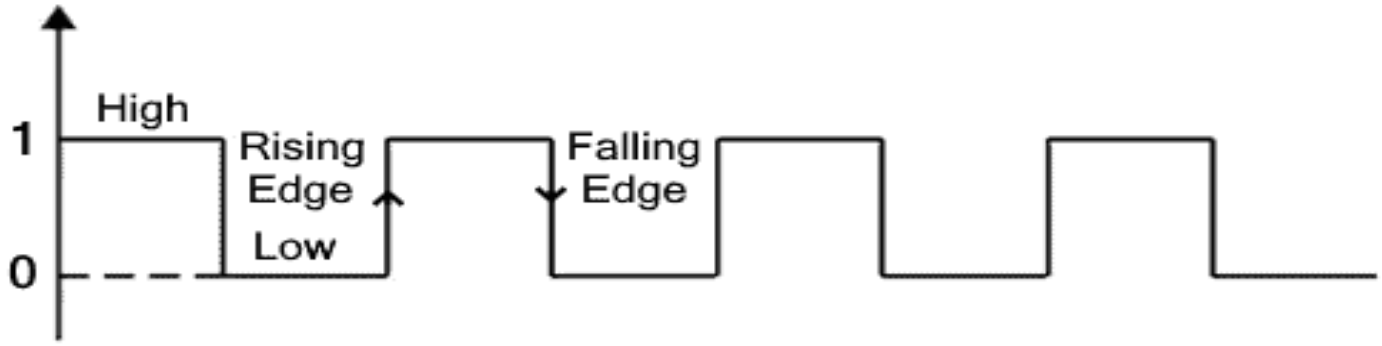
- S = 0 and R = 0 Remains in whatever state it was in, prior to the input condition. Memory state.
- S = 0 and R = 1 Sets the latch to Q = 0 and Q' = 1. It remains in this state even after R returns to 0. Called the Reset State.
- S = 1 R = 0 Sets the latch to Q = 1 and Q' = 0 Remains in this state even after S returns to 0. Called the Set State.
- S =1 and R =1 Not used as both outputs go low and the complementary feature of the device is lost.
- If both inputs are simultaneously made low, the gates race each other and the output cannot be predicted.

UNIT -4
SEQUENTIAL LOGIC
TOPIC – 3
FLIP-FLOP(S-R)

Clock Pulse:

A clock pulse is a periodic pulse signal that continuously oscillates between high state and low state. Usually, it is a square-type signal with a 50% duty cycle. Whereas the frequency of the clock may vary depending on the application.

A clock pulse has two levels i.e. high level and low level. It also has two edges i.e. low to high (rising or positive edge) and high to low (falling or trailing or negative edge).



Clock Pulse

Types of Triggering:

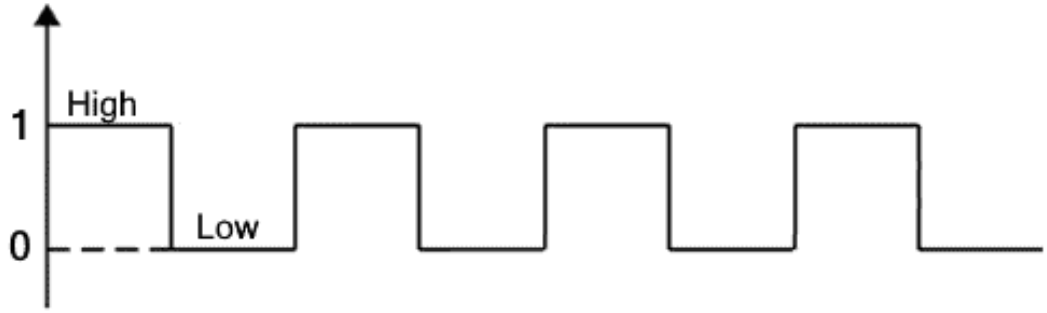
There are two types of Triggering the flip-flops.

- Level Triggering
- Edge Triggering

Level Triggering:

In level triggering, the circuit is triggered during the high level or low level of the clock pulse. Its output changes when the clock level is maintained.

Level triggering is divided into two types



Level Triggering

Positive Level Triggering

In positive-level triggering, the circuit is triggered only at the high level of the clock pulse

Negative Level Triggering

In negative-level triggering, the circuit triggers at the low level of the clock pulse

Edge Triggering

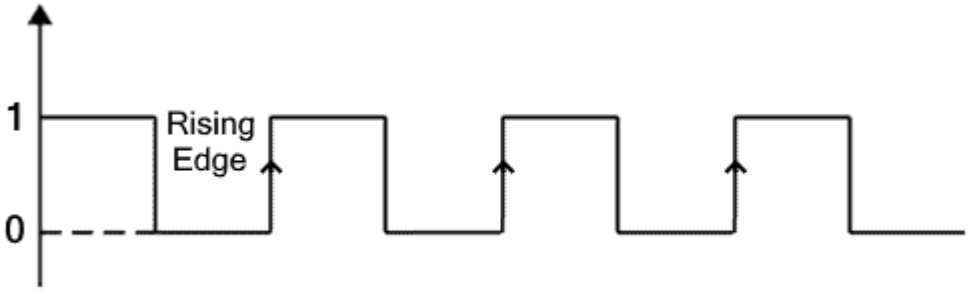
In edge triggering, the sequential circuit can change its state only when a clock edge occurs.

There are two types of edge-triggering

Positive edge triggering

The circuit only changes its state when there is a positive or rising edge at the clock input.

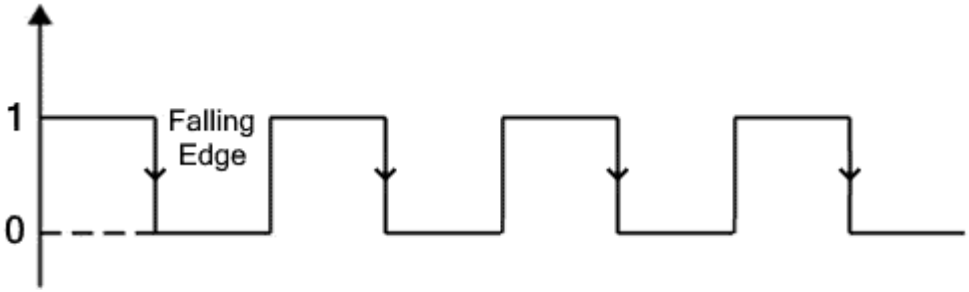
The clock pulse goes from a low state to a high state. The circuit is said to be a positive edge-triggered circuit.



Positive Edge Triggering

Negative Edge Triggering

In negative edge triggering, the circuit only activates when there is a negative or falling clock edge. The pulse goes from a high to a low state. Such circuits are called negative or falling edge-triggered circuits.



Negative Edge Triggering

TYPES OF FLIP-FLOPS:

There are different types of flip-flops depending on how their inputs and clock pulses cause transition between two states.We will discuss four different types of flip-flops

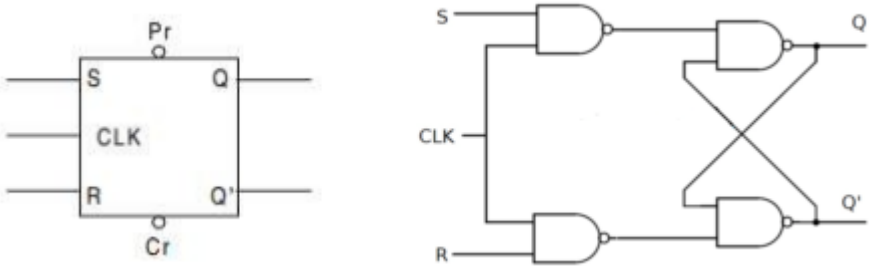
- S-R flip-flop
- D flip-flop
- J-K flip-flop
- T flip-flop

Basically D, J-K, and T are three different modifications of the S-R flip-flop.

CLOCKED S-R FLIP-FLOP:

The logic symbol of the S-R flip-flop is shown below. It has three inputs: S, R, and CLK. The CLK input is marked with a small triangle. The triangle is a symbol that denotes the fact that the circuit responds to an edge or transition at CLK input.

BLOCK DIAGRAM **LOGIC DIAGRAM**



Case 1. If $S_n = R_n = 0$, and the clock pulse is not applied, the output of the fl ip-fl op remains in the present state. Even if $S_n = R_n = 0$, and the clock pulse is applied, the output at the end of the clock pulse is the same as the output before the clock pulse, *i.e.*, $Q_{n+1} = Q_n$. The first row of the table indicates that situation. **Case 2.** For $S_n = 0$ and $R_n = 1$, if the clock pulse is applied (*i.e.* CLK = 1), the output of NAND gate 1 becomes 1; whereas the output of NAND gate 2 will be 0. Now a 0 at the input of NAND gate 4 forces the output to be 1

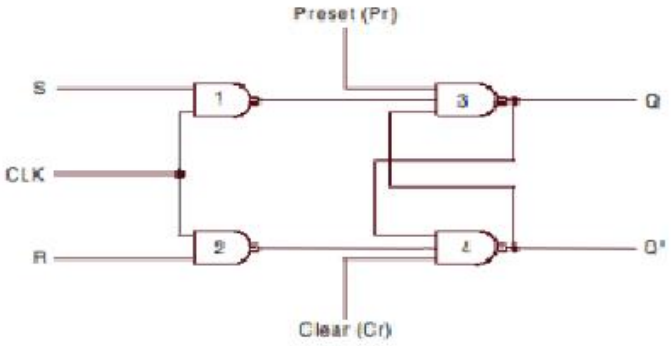
i.e. Q' = 1. This 1 goes to the input of NAND gate 3 to make both the inputs of NAND gate 3 as 1, which forces the output of NAND gate 3 to be 0, i.e., Q = 0.

Case 3. For Sn = 1 and Rn = 0, if the clock pulse is applied (i.e., CLK = 1), the output of NAND gate 2 becomes 1; whereas the output of NAND gate 1 will be 0. Now a 0 at the input of NAND gate 3 forces the output to be 1, i.e., Q = 1. This 1 goes to the input of NAND gate 4 to make both the inputs of NAND gate 4 as 1, which forces the output of NAND gate 4 to be 0, i.e., Q' = 0.

Case 4. For Sn = 1 and Rn = 1, if the clock pulse is applied (i.e. CLK = 1), the outputs of both NAND gate 2 and NAND gate 1 becomes 0. Now a 0 at the input of both NAND gate 3 and NAND gate 4 forces the outputs of both the gates to be 1, *i.e.*, Q = 1 and Q' = 1. When the CLK input goes back to 0 (while S and R remain at 1), it is not possible to determine the next state, as it depends on whether the output of gate 1 or gate 2 goes to 1 first.

Preset and Clear

Till now in the flip-flops when the power is switched on, the state of the circuit is uncertain. It may come to reset (Q = 0) or set (Q = 1) state. But in many applications it is required to initially set or reset the flip-flop., *i.e.*, the initial state of the flip-flop is to be assigned. This is done by using the direct or asynchronous inputs. These inputs are referred to as ***preset*** **(Pr)** and ***clear*** **(Cr)** inputs. These inputs may be applied at any time between clock pulses and is not in synchronism with the clock. Such an S-R flip-flop containing preset and clear inputs is shown in Figure below.



S	R	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

Invalid states

Truth table of SR flip flop

From the above Figure, we see that if Pr = Cr = 1, the circuit operates according to the table of clocked S-R flip-flop.

If Pr = 1 and Cr = 0, the output of NAND gate 4 is forced to be 1, *i.e.*, Q' = 1 and the flip-flop is reset, overwriting the previous state of the flip-flop.

If Pr = 0 and Cr = 1, the output of NAND gate 3 is forced to be 1, *i.e.*, Q = 1 and the flip-flop is set, overwriting the previous state of the flip-flop. Once the state of the flip-flop is established asynchronously, the inputs Pr and Cr must be connected to logic 1 before the next clock is applied.

The condition Pr = Cr = 0 must not be applied, since this leads to an uncertain state.

Characteristic Equation of an S-R Flip-flop:

The *characteristic table* of a flip-flop actually gives us an idea about the character, *i.e.*, the working of the flip-flop. We know that the next state flip-flop output (Q_{n+1}) depends on the present inputs as well as the

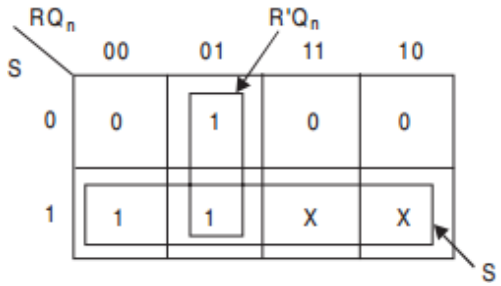
present output (Q_n). So in order to know the next state output of a flip-flop, we have to consider the present state output also. The characteristic table of an S-R flip-flop is given in the table below. From the characteristic table we have to find out the characteristic equation of the S-R flip-flop.

S	R	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Invalid states

Truth table of SR flip flop

Now we will find out the characteristic equation of the S-R flip-flop from the characteristic table with the help of the Karnaugh map:-



$Q_{n+1}(S,R,Q_n)=\Sigma m(1,4,5)+\Sigma d(6,7)$

From the Karnaugh map above we find the expression for Q_{n+1} as

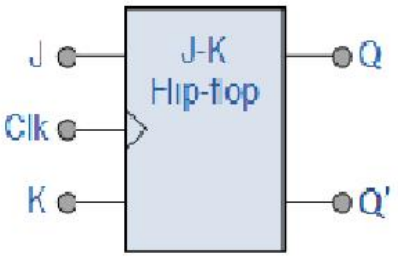
$$Q_{n+1} = S + R'Q_n$$

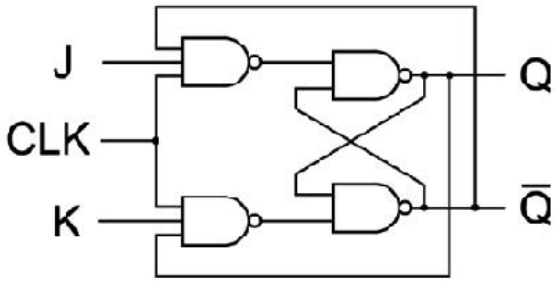
UNIT -4
SEQUENTIAL LOGIC
TOPIC – 4
FLIP-FLOPS(J-K & T)

J-K FLIP-FLOP

A J-K flip-flop has very similar characteristics to an S-R flip-flop. The only difference is that the undefined condition for an S-R flip-flop, *i.e.*, $S_n = R_n = 1$ condition, is also included in this case. Inputs J and K behavelike inputs S and R to set and reset the flip-flop respectively. When J = K = 1, the flip-flop is said to be in a ***toggle state***, which means the output switches to its complementary state every time a clock passes.
A J-K flip-flop using NAND gates:-

BLOCK DIAGRAM **LOGIC DIAGRAM** **TRUTH TABLE**





J	K	Present state Q_n	Next state Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Truth table of JK flip flop

Case 1. When the clock is applied and J = 0, whatever the value of Q'_n (0 or 1), the output of NAND gate 1 is 1. Similarly, when K = 0, whatever the value of Q_n (0 or 1), the output of gate 2 is also 1. Therefore, when J = 0 and K = 0, the inputs to the basic flip-flop are S = 1 and R = 1. This condition forces the flip-flop to remain in the same state.

Case 2. When the clock is applied and J = 0 and K = 1 & the previous state of the flip-flop is reset (*i.e.*, $Q_n = 0$ and $Q'_n = 1$), then S = 1 and R = 1. Since S = 1 and R = 1, the basic flip-flop does not alter the state and remains in the reset state. But if the flip-flop is in set condition (*i.e.*, $Q_n = 1$ & $Q'_n = 0$), then S = 1 and R = 0. Since S = 1 and R = 0, the basic flip-flop changes its state and resets.

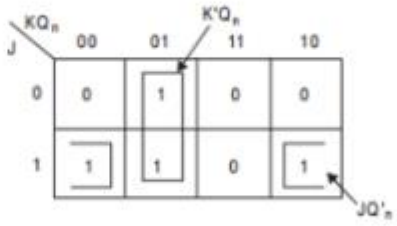
Case 3. When the clock is applied and J = 1 and K = 0 and the previous state of the flip-flop is reset (*i.e.*, $Q_n = 0$ and $Q'_n = 1$), then S = 0 and R = 1. Since S = 0 and R = 1, the basic flip-flop changes its state and goes to the set state. But if the flip-flop is already in set condition (*i.e.*, $Q_n = 1$ and $Q'_n = 0$), then S = 1 and R = 1. Since S = 1 and R = 1, the basic flip-flop does not alter its state and remains in the set state.

Case 4. When the clock is applied and J = 1 and K = 1 and the previous state of the flip-flop is reset (*i.e.*, $Q_n = 0$ and $Q'_n = 1$), then S = 0 and R = 1. Since S = 0 and R = 1, the basic flip-flop changes its state and goes to the set state. But if the flip-flop is already in set condition (*i.e.*, $Q_n = 1$ and $Q'_n = 0$), then S = 1 and R = 0. Since S = 1 and R = 0, the basic flip-flop changes its state and goes to the reset state. So we find that for J = 1 and K = 1, the flip-flop toggles its state from *set* to *reset* and vice versa. Toggle means to switch to the opposite state.

Characteristic Table of a J-K Flip-flop

The characteristic table of a J-K flip-flop is given in the table below. From the characteristic table we have to find out the characteristic equation of the J-K flip-flop.

Flip-flop inputs		*Present output*	*Next output*
J	*K*	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



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

Excitation table of JK flip flop

From the Karnaugh map, we obtain $\mathbf{Q_{n+1} = JQ'_n + K'Q_n}$.
Hence, the characteristic equation of a J-K flip-flop is $\mathbf{Q_{n+1} = JQ'_n + K'Q_n}$

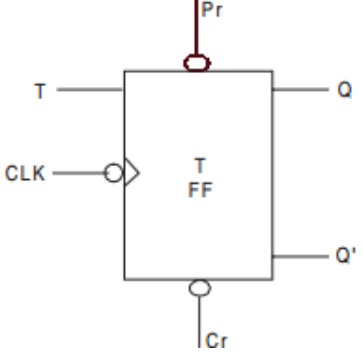
T Flip-flop:

With a slight modification of a J-K flip-flop, we can construct a new flip-flop called a T flip-flop. If the two inputs J and K of a J-K flip-flop are tied together it is referred to as a T flip-flop. Hence, a T flip-flop has onlyone input T and two outputs Q and Q'. The name T flip-flop actually indicates the fact that the flip-flop has the ability to toggle. It has actually only two states—***toggle state*** **and** ***memory state***. Since there are only two states, a T flip-flop is a very good option to use in counter design and in sequential circuits design where switching an operation is required. The truth table of a T flip-flop is given below:-

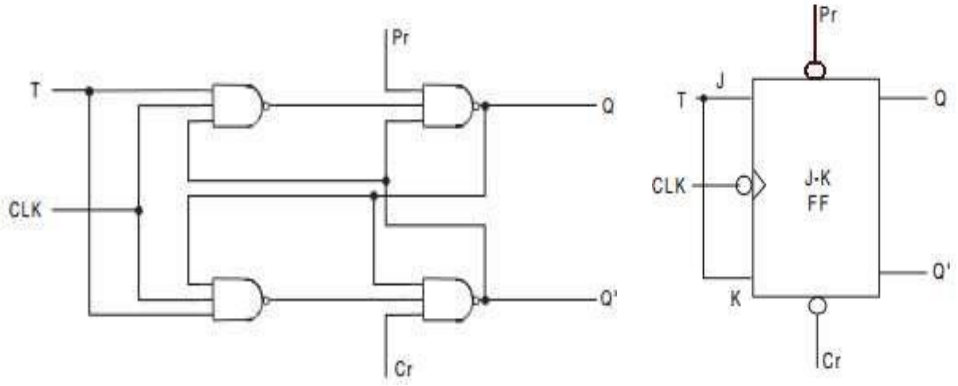
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

If the T input is in 0 state (*i.e.*, J = K = 0) prior to a clock pulse, the Q output will not change with the clockpulse. On the other hand, if the T input is in 1 state (*i.e.*, J = K = 1) prior to a clock pulse, the Q output will change to Q' with the clock pulse. In other words, we may say that, if T = 1 and the device is clocked, then the output toggles its state.

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 $Q_{n+1} = Q'_n$, *i.e.*, the state of the flip-flop is complemented. The circuit



diagramof a T flip-flop and the block diagram of the T flip-flop is shown below:-

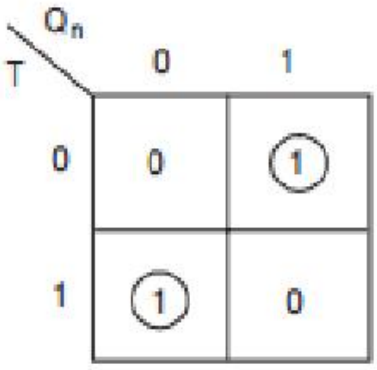


Characteristic Table of a T Flip-flop:

As we already discussed the characteristic equation of a J-K flip-flop, we can similarly find out the characteristic equation of a T flip-flop. The characteristic table of a T flip-flop is given below. From the characteristic table we have to find outthe characteristic equation of the T flip-flop.

T	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of T flip flop



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

Excitation table of T flip flop

Now we will find out the characteristic equation of the T flip-flop from the characteristic table with the help ofthe Karnaugh map below:-

From the Karnaugh map, the Boolean expression of Q_{n+1} is derived as $Q_{n+1} = TQ'_n + T'Q_n$. Hence,

$$Q_{n+1} = TQ'_n + T'Q_n$$

the characteristic equation of a T flip-flop is

UNIT -4
SEQUENTIAL LOGIC
TOPIC –5
CONVERSION OF FLIP -FLOPS

EXCITATION TABLE OF A FLIP-FLOP:

The truth table of a flip-flop is also referred to as the characteristic table of a flip-flop, since this table refers to the operational characteristics of the flip-flop. But in designing sequential circuits, we often face situationswhere the present state(PS) & the next state(NS) of the flip-flop is specified, and we have to find out the input conditions that must prevail for the desired output condition. By present and next states we mean to say the conditions before and after the clock pulse respectively. For example, the output of an S-R flip-flop before the clock pulse is $Q_n = 1$ and it is desired that the output does not change when the clock pulse is applied.

Now from the characteristic table of an S-R flip-flop, we obtain the following conditions:

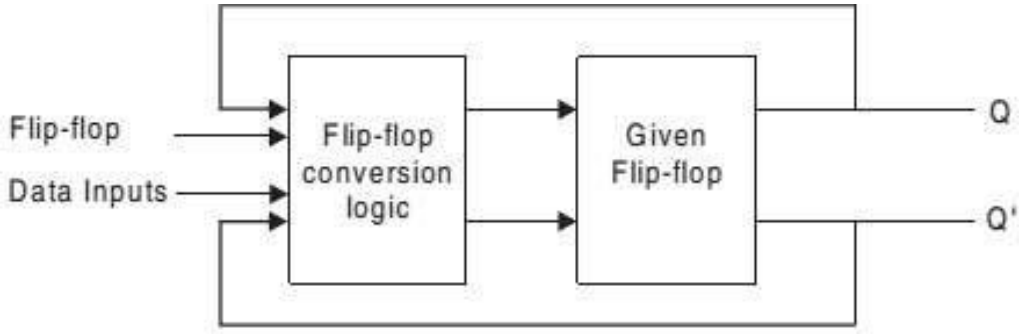
1. S = R = 0 (second row)
2. S = 1, R = 0 (sixth row).

We come to the conclusion from the above conditions that the R input must be 0, whereas the S input may be 0 or 1 (*i.e.*, don't-care). Similarly, for all possible situations, the input conditions can be found out. A tabulation of these conditions is known as an *excitation table*. The table below gives the excitation table for S-R, D, J-K, & T flip-flops. These conditions are derived from the corresponding characteristic tables of the flip-flops.

Present	*Next*	*S-R FF*		*D-FF*	*J-K FF*		*T-FF*
State (Q_n)	*State* (Q_{n+1})	S_n	R_n	D_n	J_n	K_n	T_n
0	0	0	X	0	0	X	0
0	1	1	0	1	1	X	1
1	0	0	1	0	X	1	1
1	1	X	0	1	X	0	0

CONVERSIONS OF FLIP-FLOPS:

In many applications, we are being given a type of flip-flop, whereas we may require some other type. In such cases we may have to convert the given flip-flop to our required flip-flop. From the below model we see that it is required to design the conversion logic for converting new input definitions into input codes that will cause the given flip- flop to work like the desired flip-flop. To design the conversion logic we need to combine the excitation table for both flip-flops and make a truth table with data input(s) and Q as the inputs and the input(s) of the given flip-flop as the output(s).

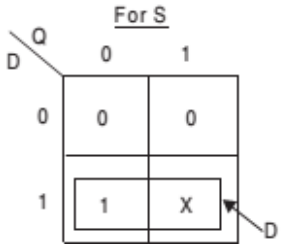


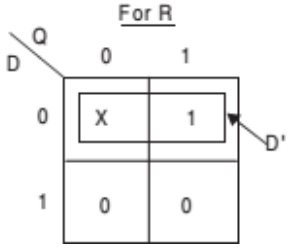
Conversion of an S-R Flip-flop to a D Flip-flop:

The excitation tables of S-R and D flip-flops are given below from which we make the truth table given

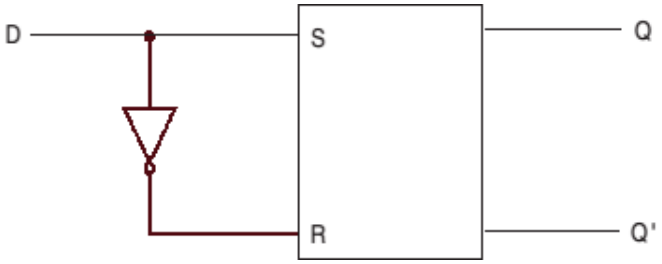
FF data inputs	*Output*	*S-R FF inputs*	
D	*Q*	*S*	*R*
0	0	0	X
1	0	1	0
0	1	0	1
1	1	X	0

From the above table, we make the Karnaugh maps for inputs S and R as shown in Figure below:-





Simplifying with the help of the Karnaugh maps, we obtain S = D and R = D'. Hence the circuit may be designed as in Figure below:-

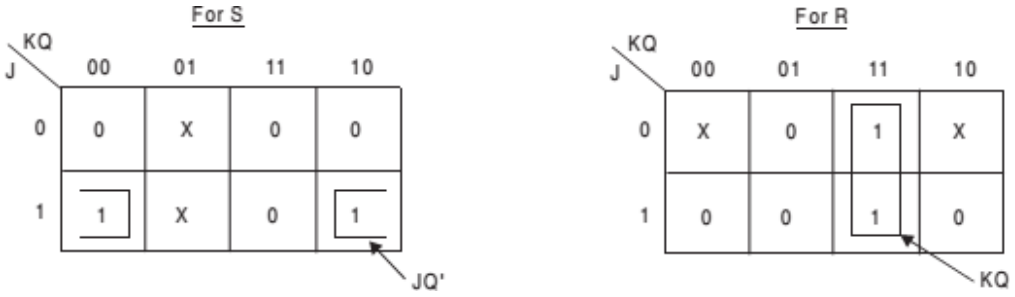


Conversion of an S-R Flip-fl op to a J-K Flip-flop:

The excitation tables of S-R and J-K flip-flops, as we studied before, from which we make the truth table given in below.

FF data inputs		*Output*	*S-R FF inputs*	
J	*K*	*Q*	*S*	*R*
0	0	0	0	X
0	1	0	0	X
1	0	0	1	0
1	1	0	1	0
0	1	1	0	1
1	1	1	0	1
0	0	1	X	0
1	0	1	X	0

From the above truth table, the Karnaugh map is prepared as shown in Figure below:-

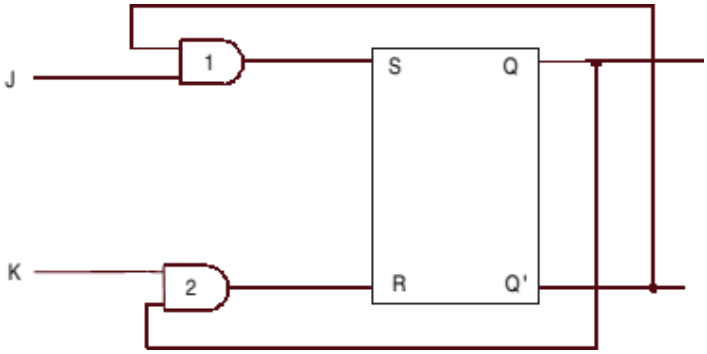


Hence we get the Boolean expression for S and R as

S = JQ'

& R = KQ.

Hence the circuit may be realized as in below:-

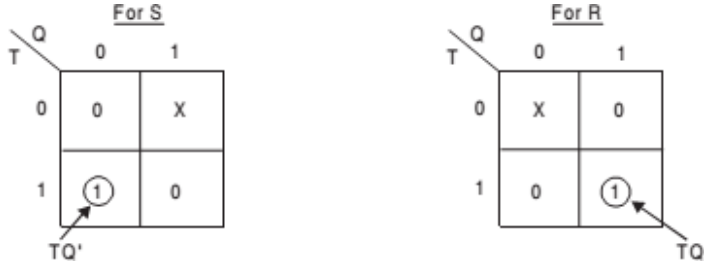


Conversion of an S-R Flip-flop to a T Flip-flop

The excitation tables of S-R and T flip-flops, from which we make the truth table given in below:-

FF data inputs	*Output*	*S-R FF inputs*	
T	*Q*	*S*	*R*
0	0	0	X
1	0	1	0
1	1	0	1
0	1	X	0

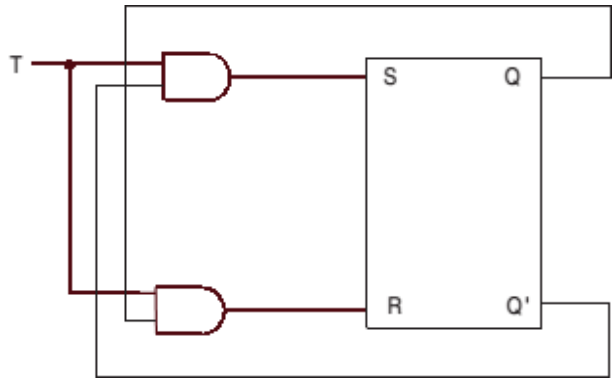
From the above truth table, the Karnaugh map is prepared as shown in Figure below:-



Hence we get the Boolean expression for S and R as:-

S = TQ' and R = TQ

Hence the circuit may be realized as in below:-



Conversion of SR to D FlipFlop:

Actual Flip-Flop:S-R-Flip-Flop
Required Flip-Flop:D-Flip-Flop
S(D,Qn)=Σm(2)+Σd(3)
R(D,Qn)=Σm(1)+Σd(0)

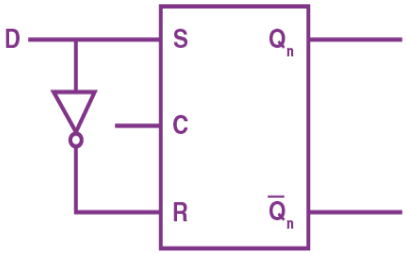
D	Q_N	Q_{N+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

S:

D,Q_N		
	0	0
	1	X

R:

D,Q_N		
	X	1
	0	0



Logic Diagram

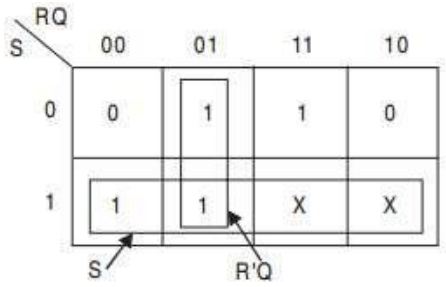
S = D R = D‘

Conversion of a D Flip-flop to an S-R Flip-flop

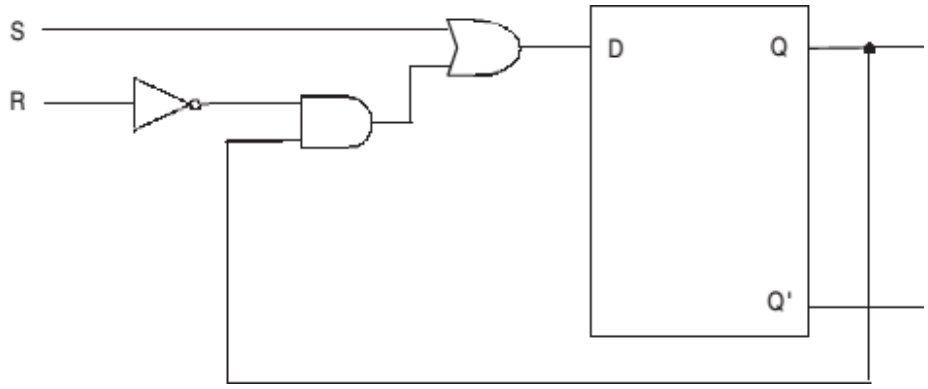
The excitation tables of S-R and D flip-flops, from which we make the truth table given in below:-

FF data inputs		*Output*	*D FF inputs*
S	*R*	*Q*	*D*
0	0	0	0
0	1	0	0
1	0	0	1
0	1	1	0
0	0	1	1
1	0	1	1

From the above truth table, the Karnaughmap is prepared as shown in Figure below:-

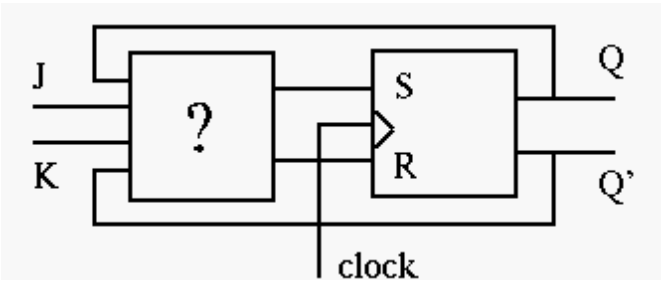


Hence we get the Boolean expression for S and R as:- **D = S + R'Q**
Hence the circuit may be realized as in below:-



Convert a SR-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:

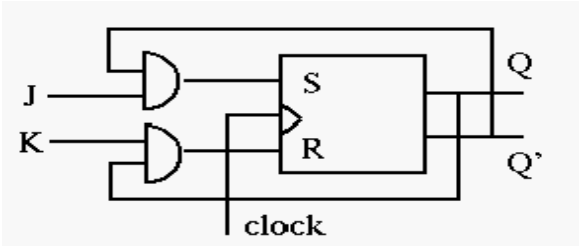
K \ QJ	00	01	11	10
0	0	1	X	X
1	0	1	0	0

S=Q'J

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

R=QK

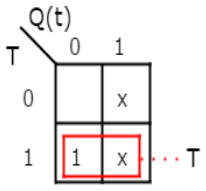
$$S = Q'J, \quad R = QK$$



JK Flip-Flop to T Flip-Flop:

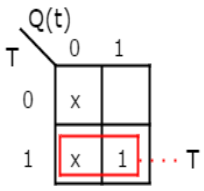
Actual Flip-Flop:J-K-Flip-Flop
Required Flip-Flop:T-Flip-Flop
$J(T,Q_t)=\Sigma m(2)+\Sigma d(1,3)$
$K(T,Q_t)=\Sigma m(3)+\Sigma d(0,2)$

K-Map for J



J = T &

K-Map for K



K = T

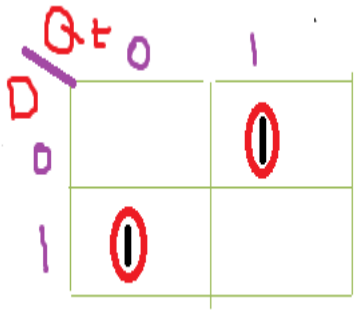
T Flip-Flop Input	Present State	Next State	JK Flip-Flop Inputs	
T	Q(t)	Q(t + 1)	J	K
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

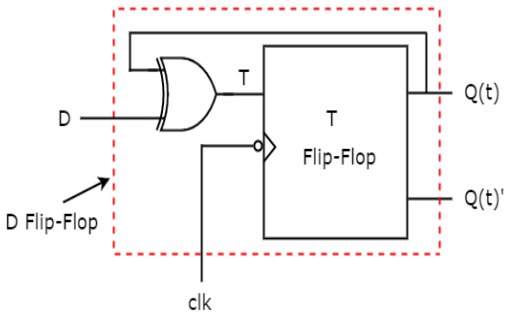
T Flip-Flop to D Flip-Flop conversion:

Actual Flip-Flop:T-Flip-Flop
Required Flip-Flop:D-Flip-Flop
$T(D,Qt)=\Sigma m(1,2)$

$$T=D'Q_t+DQ_t'=D \oplus Q_t$$

D Flip-Flop Input	Present State	Next State	T Flip-Flop Input
D	Q(t)	Q(t + 1)	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0





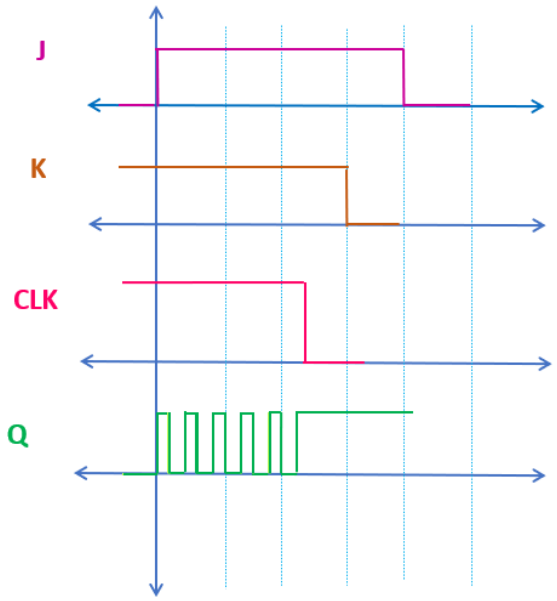
UNIT -4
SEQUENTIAL LOGIC
TOPIC – 6
Analysis of Clocked Sequential Circuits

Master-Slave J-K Flip-flop:

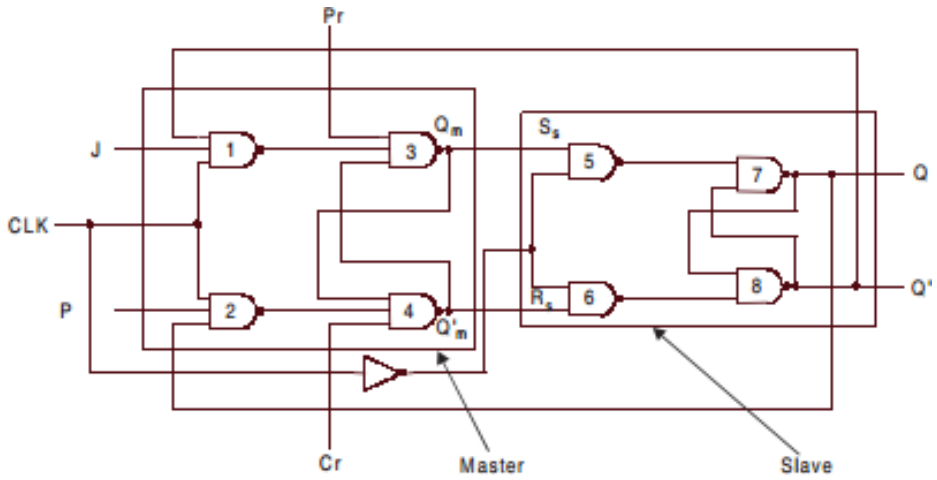
A master-slave J_K flip-flop is a system of two flip-flops—one being designated as *master* and the other is the *slave*. From the figure below we see that a clock pulse is applied to the master and the inverted form of the same clock pulse is applied to the slave.

Race Around Condition in JK Flip-Flop:

In the level triggered JK flip-flop, when both J and K input are 1 and when the ON time of the clock is more than the propagation delay of the flip-flop then the output of the flip-flop will toggle continuously between '1' and '0'. And because of that, it is difficult to predict the output of the flip-flop once the clock becomes low. The race around condition in undesirable condition in the flip-flop, and should be avoided to get the reliable flip-flop output.Using the JK flip-flop in master slave configuration, this race around condition can be avoided.

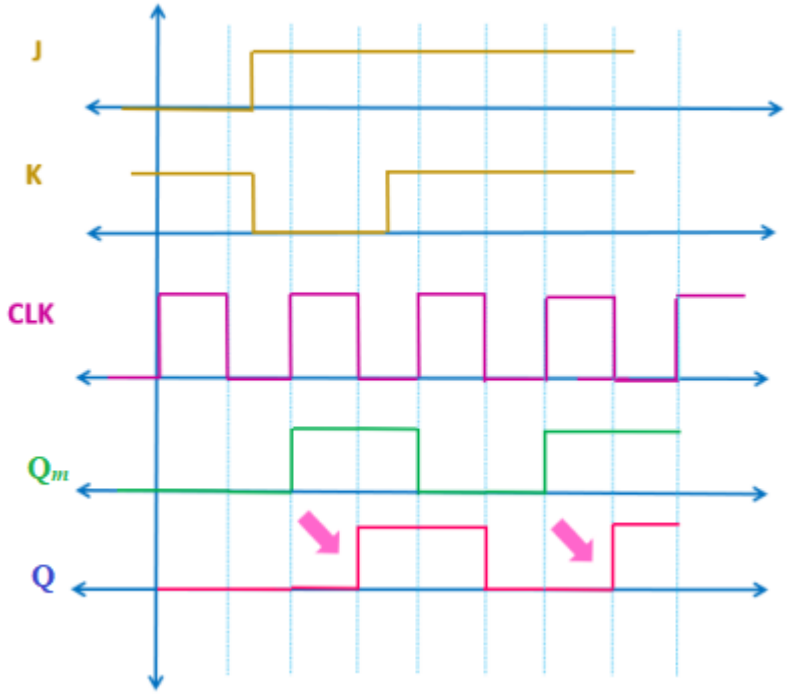


When CLK = 1, the first flip-flop (*i.e.*, the master) is enabled and the outputs Q_m and Q'_m respond to the inputs J and K. At this time the second flip-flop (*i.e.*, the slave) is disabled because the CLK is LOW to the second flip-flop. Similarly, when CLK becomes LOW, the master becomes disabled and the slave becomes active. Therefore, the outputs Q and Q' follow the outputs Q_m and Q'_m respectively. Since the second flip-flop just follows the first one, it is referred to as a slave and the first one is called the master. Hence, the configuration is referred to as a master-slave (M-S) flip- flop.



Working of Master Slave JK Flip-Flop:

Let's understand the working of Master Slave JK flip-flop using timing diagram.

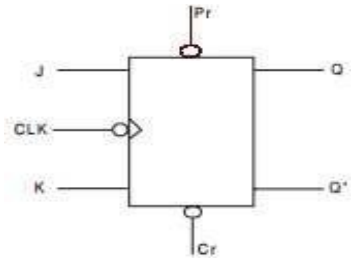


Timing Diagram of Master Slave JK Flip-Flop

Here, initially it has been assumed that the outputs of both master and slave are 0. (Both Q_m and Q are 0). When the clock signal is high then master latch will get enabled and will respond to the input signals. In this case, initially during the first clock, J = 0 and K = 1. So, output of the master latch will be 0. During ON time of the clock, the slave latch will remain disabled and it will hold its current state. Similarly, during the OFF time of the clock, the master will get disabled and it will hold its current state. And at the same time, the slave latch will become active and will follow the master output. That means, the slave latch is following the master output during the off time of the clock. Or in other words, the slave latch follows the master output after the delay of T_{ON} . Where T_{ON} is the ON time of the clock signal. This cycle repeats at every clock cycle.

During the ON time of the second clock, since J = 1 and K = 0, so master output $Q_m = 1$. And the slave output Q will remain 0. (Since it is disabled) The slave latch will follow the master output during the OFF time of the clock. From the timing diagram, we can see that, the slave is following the master output after the delay of T_{ON}.

In this type of circuit configuration the inputs to the gates 5 and 6 do not change at the time of application of the clock pulse. Hence the race-around condition does not exist. The state of the master-slave flip-flop, changes at the negative.The logic symbol of a negative edge master-slave is shown in Figure below.



UNIT -4

SEQUENTIAL LOGIC

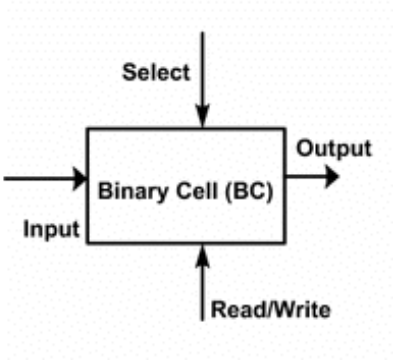
TOPIC – 7

BINARY STORAGE AND REGISTERS

THE BINARY CELL:

The binary cells are kind of storage cells that are capable of storing one binary digit or binary character. They are used to store information in digital electronics.

When a cell is read, it saves one bit of binary data, and before it can be accessed, it must be set to store a 1 and then reset to a 0.



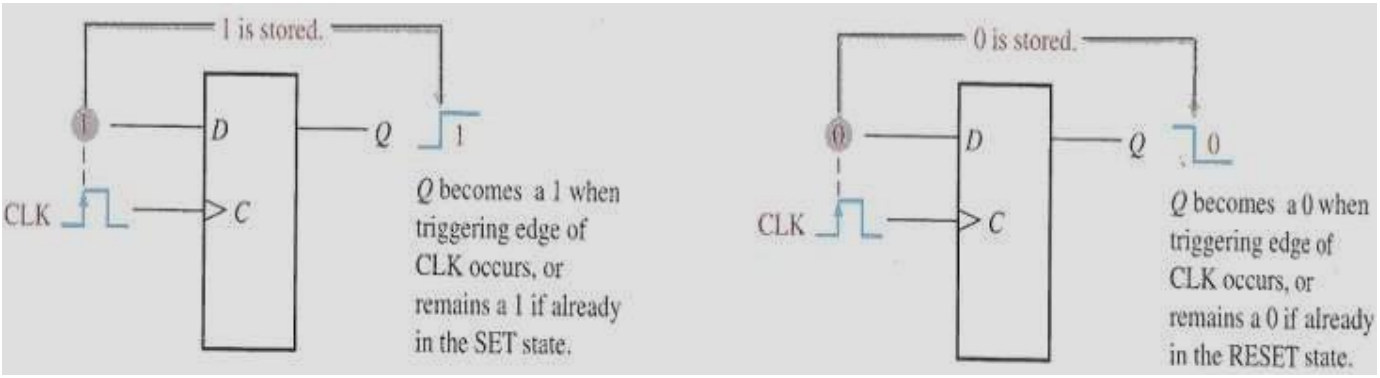
REGISTERS

A register is a group of binary storage cells capable of holding binary information. A group of flip-flops constitutes a register, since each flip-flop can work as a binary cell. An *n*-bit register, has n flip-flops and is capable of holding n-bits information. In addition to flip-flops a register can have a combinational part that performs data-processing tasks.

Register:

- A set of n flip-flops
- Each flip-flop stores one bit
- Two basic functions: data storage and data movement.

Storage Capacity of a register



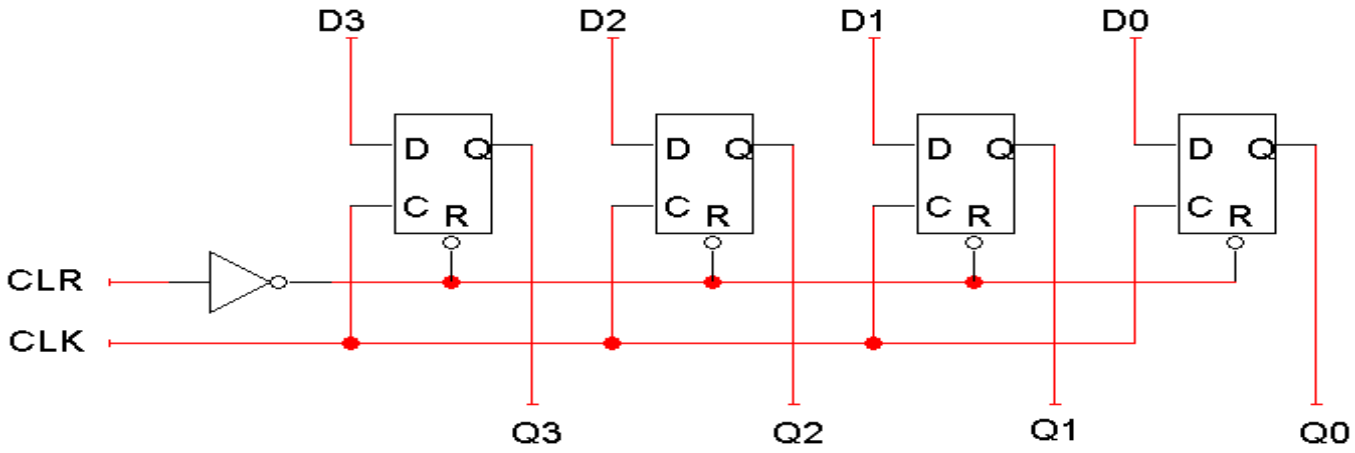
Registers are commonly used as temporary storage in a processor.

- They are faster and more convenient than main memory.
- More registers can help speed up complex calculations.

Basic registers are easy to build. We can store multiple bits just by putting a bunch of flip-flops together!

A 4-bit register is shown on the right, and its internal implementation is below.

- This register uses D flip-flops
- it's easy to store data without worrying about flip-flop input equations.
- All the flip-flops share a common CLK and CLR signal.



CONTROLLED BUFFER REGISTERS:

Buffer registers are used to store binary words

The input D_3-D_0 is copied to the output Q_3-Q_0 on every clock cycle.

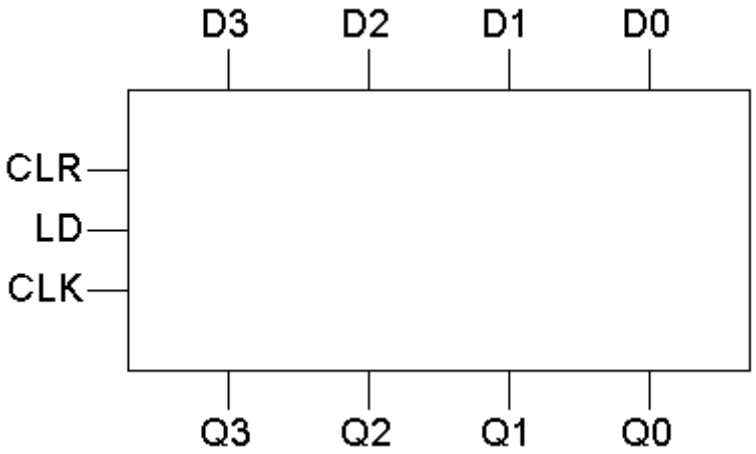
How can we store the current value for more than one cycle?

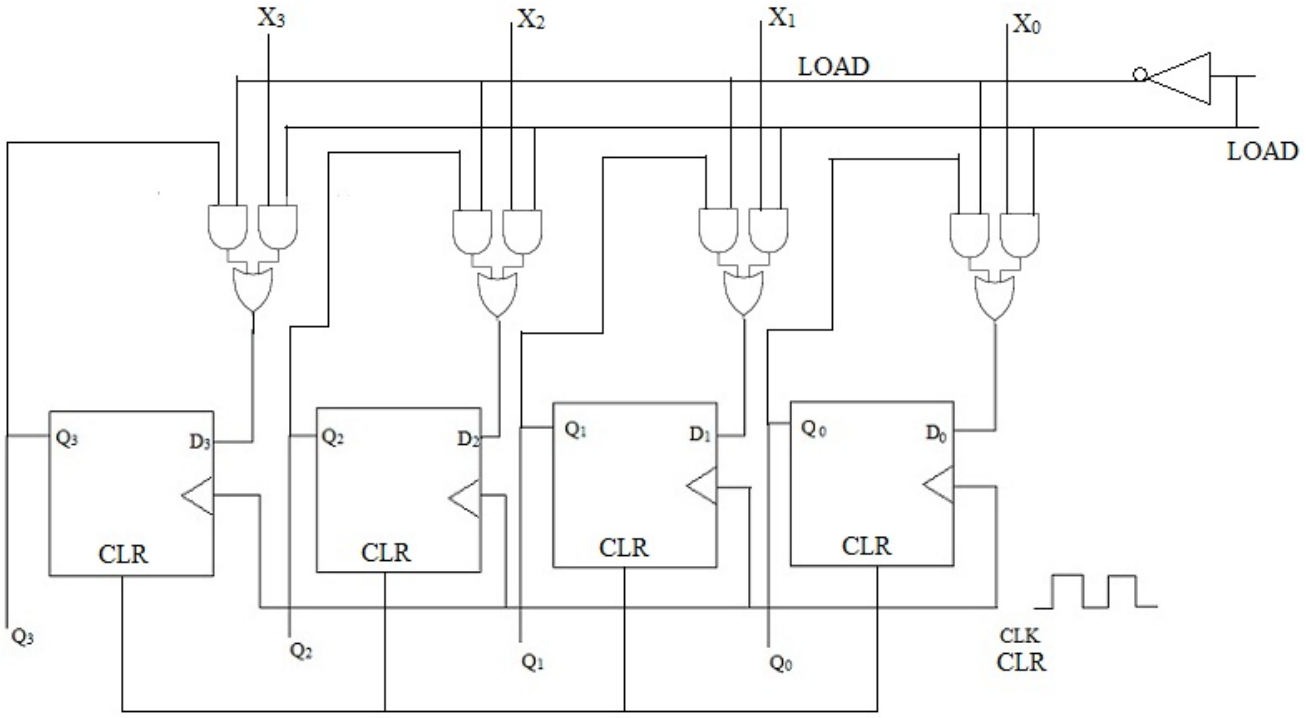
Let's add a load input signal LD to the register.

If LD = 0, the register keeps its current contents.

If LD = 1, the register stores a new value, taken from inputs D_3-D_0.

LD	Q(t+1)
0	Q(t)
1	D_3-D_0





Control terminal LOAD determines circuit function. When LOAD is HIGH, data X is permitted to reach flip-flop.

when LOAD is LOW, that permits Q outputs to go to D inputs

That is, contents of register continue to remain unchanged so long as LOAD is LOW.

UNIT -4

SEQUENTIAL LOGIC

TOPIC – 8

SHIFT REGISTERS

SHIFT REGISTERS:

A shift register is a storage device that used to store binary data.

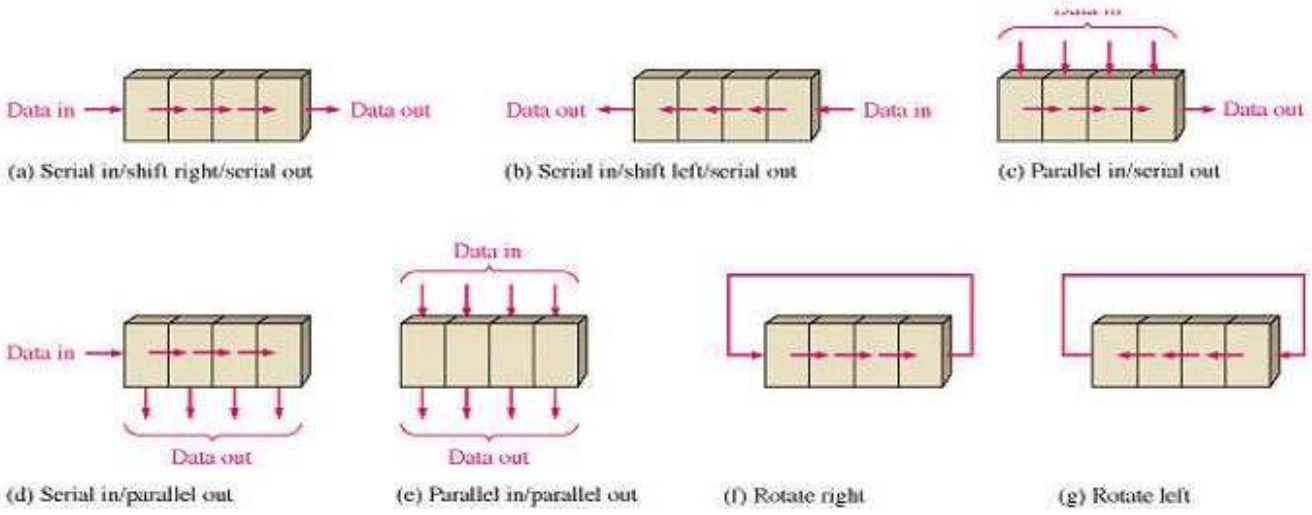
In these registers the connection is done in such a way that the output of one of the flip flop forms as input to other, it is known as a shift register. The data in a shift register is moved serially (one bit at a time).

There are two ways to shift data into a register (serial or parallel) and similarly two ways to shift the data out of the register.

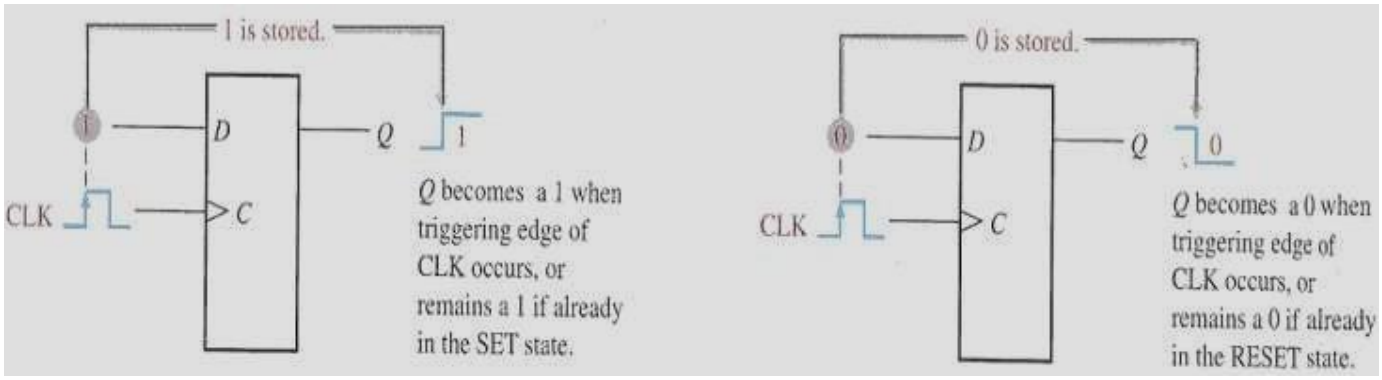
This leads to the construction of four basic types of registers:-

1. Serial in/Serial out (SISO)
2. Serial in/Parallel out (SIPO)
3. Parallel in/Serial out (PISO)
4. Parallel in/Parallel out (PIPO)

Basic data movement operation in shift registers

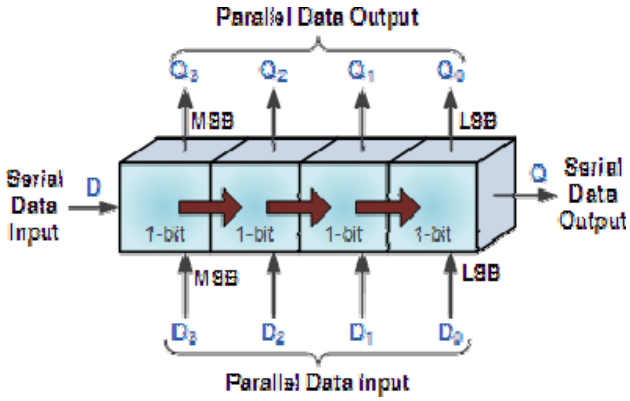


Storage Capacity of a register



The storage capacity of a register is the total number of bits (1 or 0) of digital data it can retain. Each stage (flip flop) in a shift register represents one bit of storage capacity. Therefore the number of stages in a register determines its storage capacity.

The effect of data movement from left to right through a shift register can be presented graphically as:



The shift register can be built using RS, JK or D flip-flops various types of shift registers are available some of them are given as under.

1. Shift Left Register
2. Shift Right Register
3. Shift Around Register
4. Bi-directional Shift Register

SERIAL-IN–-SERIAL-OUT SHIFT REGISTER:

From the name itself it is obvious that this type of register accepts data serially, *i.e.*, one bit at a time at the single input line. The output is also obtained on a single output line in a serial fashion. The data within the register may be shifted from left to right using *shift-left* register, or may be shifted from right to left using *shift- right* register.

Shift-right Register:

A shift-right register can be constructed with either J-K or D flip-flops as shown in Figure 8.3. A J-K flip-flop based shift register requires connection of both J and K inputs. Input data are connected to the J and K inputs of the left most (lowest order) flip-fl op. To input a 0, one should apply a 0 at the J input, *i.e.*, J = 0 and K = 1 and vice versa. With the application of a clock pulse the data will be shifted by one bit to the right.

In the shift register using D flip-flop, D input of the left most flip-flop is used as a serial input line. To input 0,one should apply 0 at the D input and vice versa.

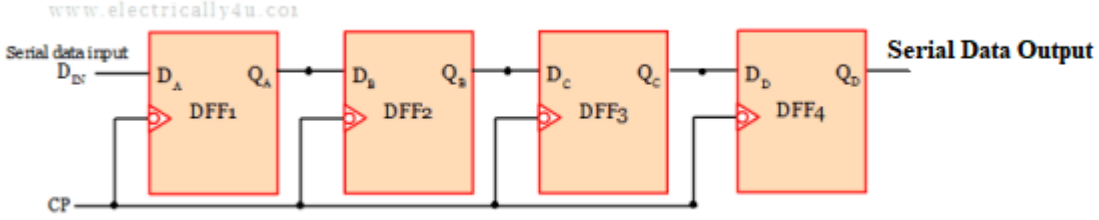


Figure of Shift-right register using D flip-flops

The clock pulse is applied to all the flip-flops simultaneously. When the clock pulse is applied, each flip-flop is either set or reset according to the data available at that point of time at the respective inputs of the individual flip-flops. Hence the input data bit at the serial input line is entered into flip-flop A by the first clock pulse. At

the same time, the data of stage A is shifted into stage B and so on to the following stages. For each clock pulse, data stored in the register is shifted to the right by one stage. New data is entered into stage A, whereas the data present in stage D are shifted out (to the right).

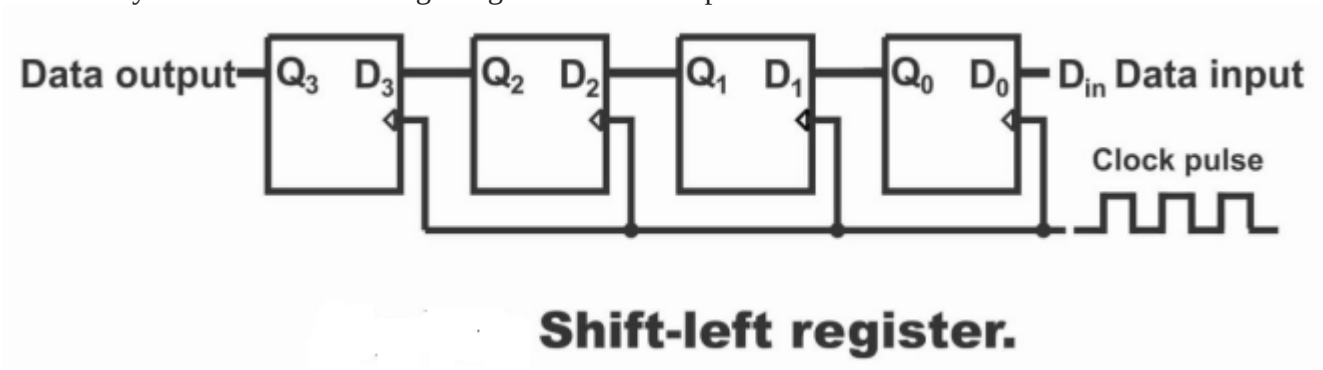
OPERATION OF THE SHIFT-RIGHT REGISTER:-

Timing pulse	Q_A	Q_B	Q_C	Q_D	*Serial output at* Q_D
Initial value	0	0	0	0	0
After 1st clock pulse	1	0	0	0	0
After 2nd clock pulse	1	1	0	0	0
After 3rd clock pulse	0	1	1	0	0
After 4th clock pulse	1	0	1	1	1

SHIFT-LEFT REGISTER:

A shift-left register can also be constructed with either J-K or D flip-flops as shown in Figure below. Let us now illustrate the entry of the 4-bit number 1110 into the register, beginning with the right-most bit. A 0 is applied at the serial input line, making D = 0. As the first clock pulse is applied, flip-fl op A is RESET, thus storing the 0. Next a 1 is applied to the serial input, making D = 1 for flip-flop A and D = 0 for flip-flop B, because the inputof flip-flop B is connected to the Q_A output.

When the second clock pulse occurs, the 1 on the data input is "shifted" to the flip-flop A and the 0 in the flip-flop A is "shifted" to flip-flop B. The 1 in the binary number is now applied at the serial input line, and the third clock pulse is now applied. This 1 is entered in flip-flop A and the 1 stored in flip-flop A is now "shifted" to flip-flop B and the 0 stored in flip-flop B is now "shifted" to flip-flop C. The last bit in the binary number that is the 1 is now applied at the serial input line and the fourth clock pulse is now applied. This 1 now enters the flip-flop A and the 1 stored in flip-flop A is now "shifted" to flip-flop B and the 1 stored in flip-flop B is now "shifted" to flip-flop C and the 0 stored in flip-flop C is now "shifted" to flip-flop D. Thus the entry of the 4-bit binary number in the shift-right register is now completed.



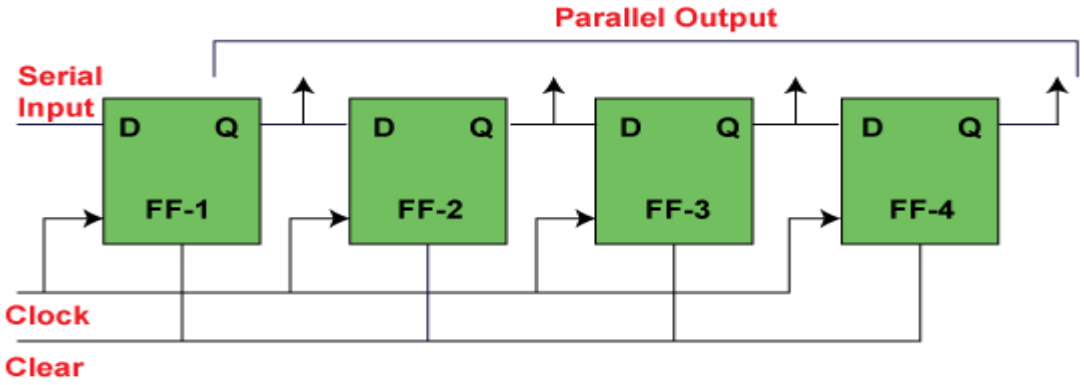
Shift-left register.

Timing pulse	Q_D	Q_C	Q_B	Q_A	*Serial output at* Q_D
Initial value	0	0	0	0	0
After 1st clock pulse	0	0	0	0	0
After 2nd clock pulse	0	0	0	1	0
After 3rd clock pulse	0	0	1	1	0
After 4th clock pulse	0	1	1	1	0

SIPO SHIFT REGISTER:

A Serial-In Parallel-Out shift register is a sequential logic device that can store and shift data bits. It consists of a chain of flip-flops connected in series, with data input and output terminals. The data is shifted from one flip-flop to the next, either in a serial or parallel fashion, depending on the mode of operation.

In the "Serial IN Parallel OUT" shift register, the data is passed serially to the flip flop, and outputs are fetched in a parallel way. The data is passed bit by bit in the register, and the output remains disabled until the data is not passed to the data input. When the data is passed to the register, the outputs are enabled, and the flip flops contain their return value.



The circuit having four D flip-flops contains a clear and clock signal to reset these four flip flops. In SIPO, the input of the second flip flop is the output of the first flip flop, and so on. The same clock signal is applied to each flip flop since the flip flops synchronize each other. The parallel outputs are used for communication.

PARALLEL-IN–SERIAL-OUT REGISTER:

In the preceding two cases the data was shifted into the registers in a serial manner. Here we develop an idea for the parallel entry of data into the register. Here the data bits are entered into the flip-flops simultaneously, rather than a bit-by-bit basis.A 4-bit parallel-in–serial-out register is illustrated in Figure below. A, B, C, and D are the four parallel data input lines and *SHIFT* /LOAD is a control input that allows the four bits of data at A, B, C, and D inputs to enter into the register in parallel or shift the data in serial. When *SHIFT* / $\overline{LOAD}$ is HIGH, AND gates G_1, G_3 & G_5 are enabled, allowing the data bits to shift right from one stage to the next. When *SHIFT* / $\overline{LOAD}$ is LOW, AND gates G_2, G_4, and G_6 are enabled, allowing the data bits at the parallel inputs. When a clock pulse is applied, the flip-flops with D = 1 will be set and the flip-flops with D = 0 will be reset, thereby storing all the four bits simultaneously. The OR gates allow either the normal shifting operation or the parallel data-entry operation, depending on which of the AND gates are enabled by the level on the *SHIFT* / $\overline{LOAD}$ input.

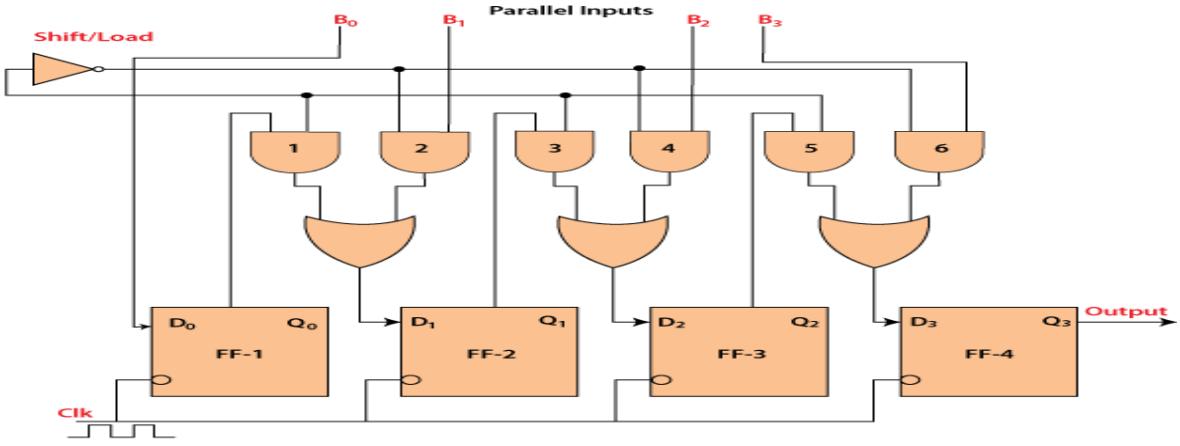
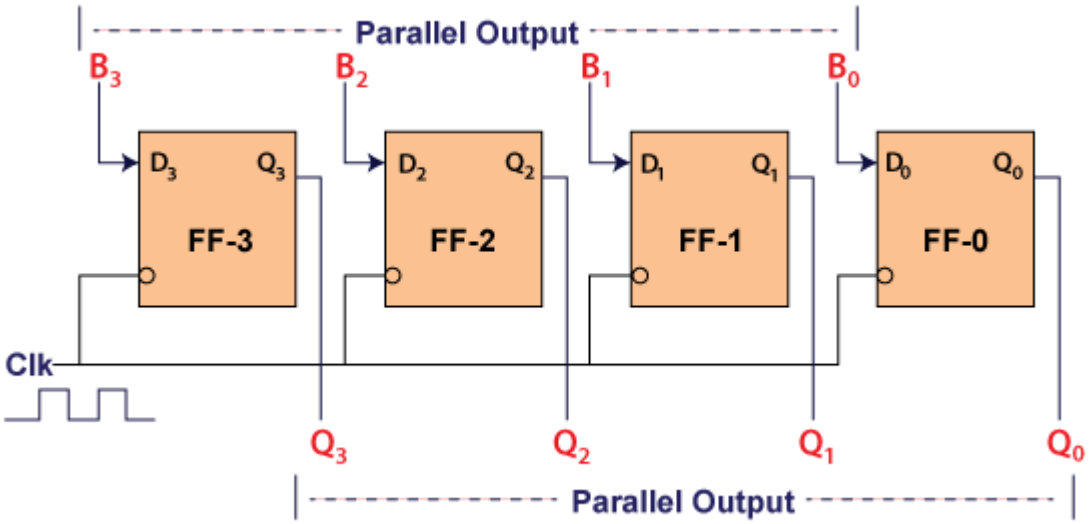


Figure:- A 4-bit parallel-in–serial-out shift register.

PARALLEL-IN–PARALLEL-OUT REGISTER:

The parallel input of data has already been discussed in the preceding section of parallel-in–serial-out shift register. Also, in this type of register there is no interconnection between the flip-flops since no serial shifting is required. Hence, the moment the parallel entry of the data is accomplished the data will be available at the parallel outputs of the register. A simple parallel-in–parallel out shift register is shown in Figure below.



Here the parallel inputs to be applied at B_0, B_1,B_2,B_3 and D inputs are directly connected to the D inputs of the respective flip-flops. On applying the clock transitions, these inputs are entered into the register and are immediately available at the outputs Q_0,Q_1, Q_2, and Q_3.

UNIT -4

SEQUENTIAL LOGIC

TOPIC – 9

2-BIT RIPPLE COUNTER

Counters:

Counter is a device which stores (and sometimes displays) the number of times particular event or process has occurred, often in relationship to a clock signal. A Digital counter is a set of flip flops whose state change in response to pulses applied at the input to the counter. Counters may be asynchronous counters or synchronous counters. Asynchronous counters are also called ripple counters

In electronics counters can be implemented quite easily using register-type circuits such as the flip-flops and a wide variety of classifications exist:

- Asynchronous (ripple) counter – changing state bits are used as clocks to subsequent state flip-flops
- Synchronous counter – all state bits change under control of a single clock
- Decade counter – counts through ten states perstage
- Up/down counter – counts both up and down, under command of a controlinput
- Ring counter – formed by a shift register with feedback connection in a ring

Each is useful for different applications. Usually, counter circuits are digital in nature, and count in natural binary Many types of counter circuits are available as digital building blocks, for example a number of chips in the 4000 series implement different counters.

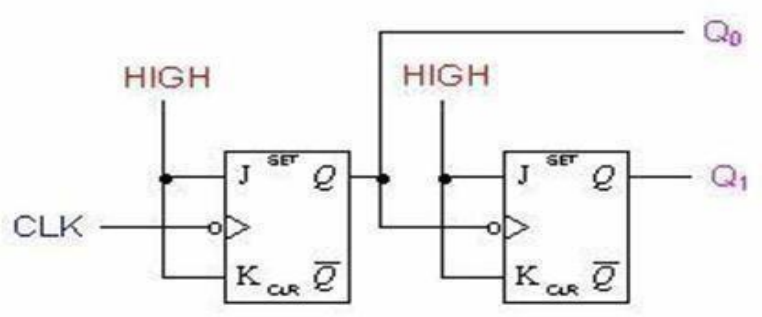
Occasionally there are advantages to using a counting sequence other than the natural binary sequence such as the binary coded decimal counter, a linear feed-back shift register counter, or a gray-code counter.

Counters are useful for digital clocks and timers, and in oven timers, VCR clocks, etc.

Asynchronous counters(RIPPLE COUNTER):

An asynchronous (ripple) counter is a single JK-type flip-flop, with its J (data) input fed from its own inverted output. This circuit can store one bit, and hence can count from zero to one before it overflows (starts over from 0). This counter will increment once for every clock cycle and takes two clock cycles to overflow, so every cycle it will alternate between a transition from 0 to 1 and a transition from 1 to 0. Notice that this creates a new clock with a 50% duty cycle at exactly half the frequency of the input clock. If this output is then used as the clock signal for a similarly arranged D flip-flop (remembering to invert the output to the input), one will get another 1 bit counter that counts half as fast. Putting them together yields a two-bit counter:

Two-bit ripple up-counter using negative edge triggered flip flop:



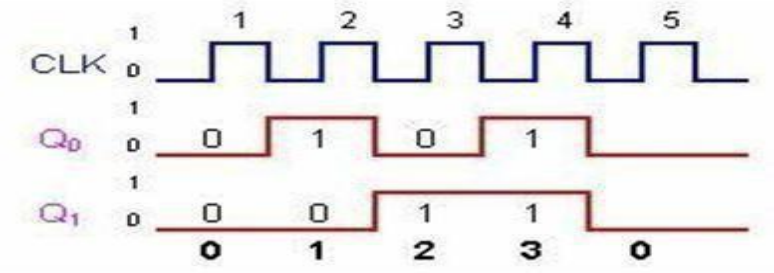
- Two bit ripple counter used two flip-flops. There are four possible states from 2 – bit up- countingI.e. 00, 01, 10 and 11.
- The counter is initially assumed to be at a state 00 where the outputs of the tow flip-flops arenoted as Q_1Q_0. Where Q_1 forms the MSB and Q_0 forms the LSB.

- For the negative edge of the first clock pulse, output of the first flip-flop FF_1 toggles itsstate. Thus Q_1 remains at 0 and Q_0 toggles to 1 and the counter state are now read as 01.

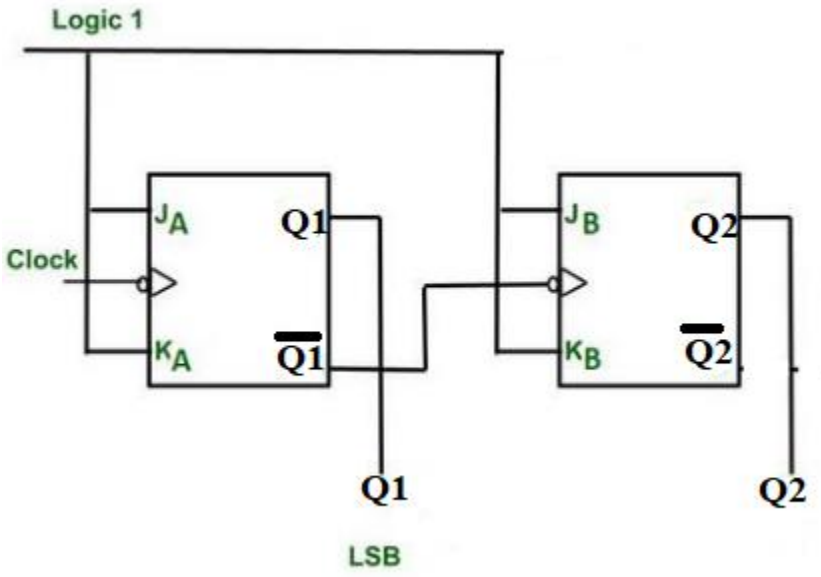
- During the next negative edge of the input clock pulse FF_1 toggles and $Q_0 = 0$. The output Q_0 being a clock signal for the second flip-flop FF_2 and the present transition acts as a negative edge for FF_2 thus toggles its state $Q_1 = 1$. The counter state is now read as10.

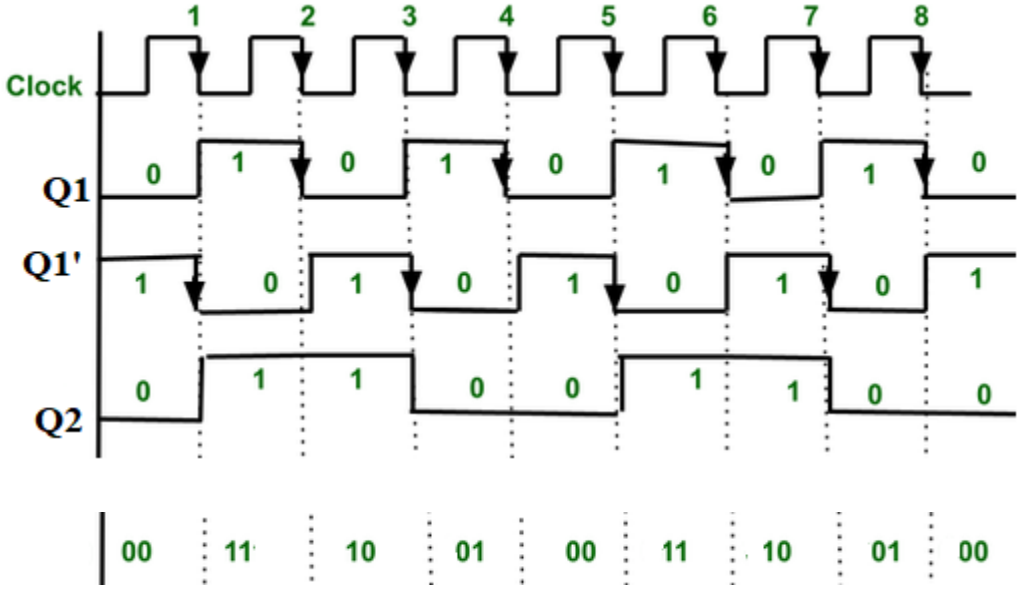
- For the next negative edge of the input clock to FF_1 output Q_0 toggles to 1. But this transition from 0 to 1 being a positive edge for FF_2 output Q_1 remains at 1. The counter state is now read as 11.

- For the next negative edge of the input clock, Q_0 toggles to 0. This transition from 1 to 0 acts as a negative edge clock for FF_2 and its output Q_1 toggles to 0. Thus the starting state 00 is attained. Figure shown below.



Two-bit ripple down-counter using negative edge triggered flip flop:





A 2-bit down-counter counts in the order 0,3,2,1,0,3……..,i.e, 00,11,10,01,00,11 …..,etc. the above fig. shows ripple down counter, using negative edge triggered J-K FFs and its timing diagram.

- For down counting, Q_1' of FF1 is connected to the clock of FF2. Let initially all the FF1 toggles, so, Q_1 goes from a 0 to a 1 and Q_1'goes from a 1 to a 0.The negative-going signal at Q_1 is applied to the clock input of FF2, toggles Ff2 and, therefore, Q2 goes from a 0 to a 1.so, after one clock pulse Q_2=1 and Q_1=1, I.e., the stateof the counter is 11.
- At the negative-going edge of the second clock pulse, Q1 changes from a 1 to a 0 and Q1_ from a 0 to a 1.
- This positive-going signal at Q_1' does not affect FF2 and, therefore, Q2 remains at a 1. Hence , the state of the counter after second clock pulse is 10
- At the negative going edge of the third clock pulse, FF1 toggles. So Q1, goes from a 0 to a 1 and Q_1' from 1 to 0. This negative going signal at Q_1' toggles FF2 and, so, Q2 changes from 1 to 0, hence, the state of the counter after the third clock pulse is01.
- At the negative going edge of the fourth clock pulse, FF1 toggles. So Q1, goes from a 1 to a 0 and Q_1' from 0 to 1. . This positive going signal at Q_1' does not affect FF2 and, so, Q2 remains at 0, hence, the state of the counter after the fourth clock pulse is 00.

UNIT -4

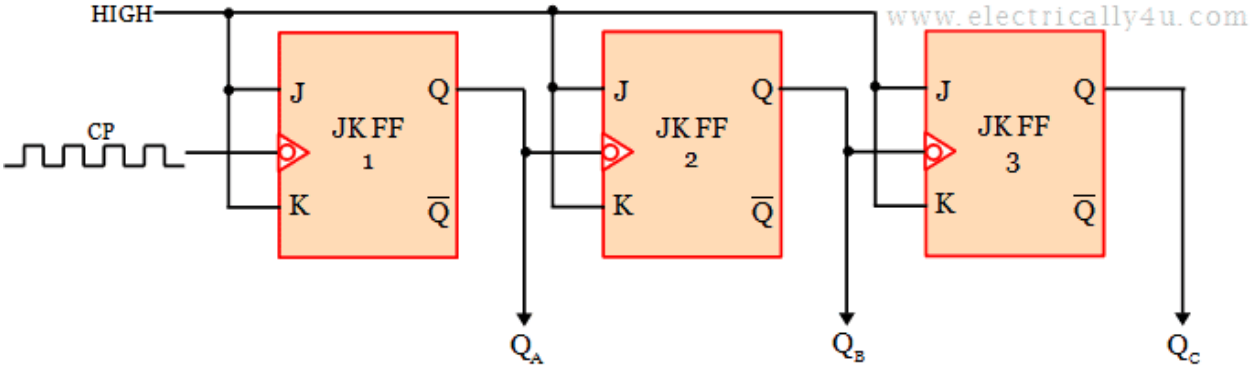
SEQUENTIAL LOGIC

TOPIC 10

3-BIT RIPPLE COUNTER

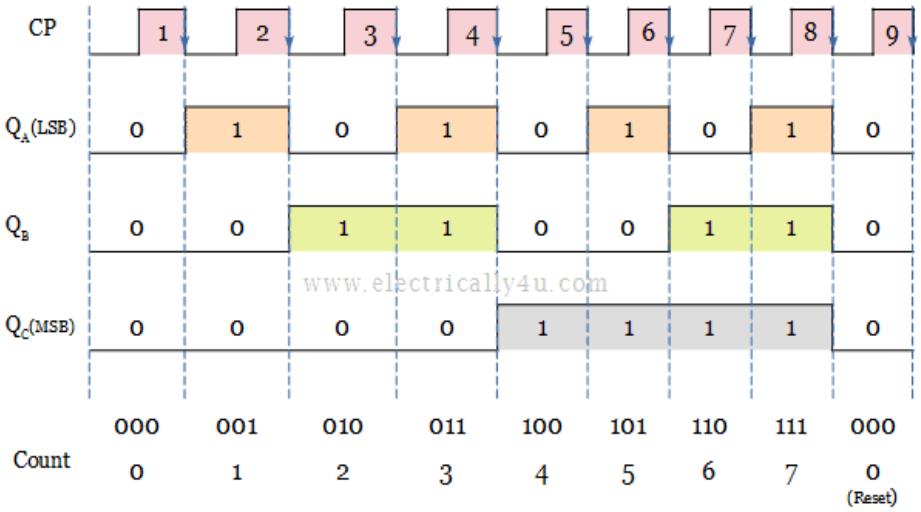
3-bit asynchronous up counter:

The 3-bit asynchronous or ripple up counter is similar to the 2-bit ripple up counter. Here for a 3-bit counter, an additional flip-flop is added. Thus for the 3-bit asynchronous counter, 3 T-flip-flops are used.
This counter consists of 23 = 8 count states(000, 001, 010, 011, 100, 101, 110, 111). The counter counts the incoming pulses starting from 0 to 7.



In a 3-bit asynchronous up counter, in which the clock pulse is given as clock input for JK FF1. For the other flip-flops, the clock input is fed from the output of previous flip-flops.

The clock pulse count is noted at the output of each flip-flop(QCQBQA), where QA is the LSB and QC is the MSB.
The below figure shows the timing diagram of the 3-bit ripple counter, which shows the change of state of each flip-flop during each clock pulse. In this type, the counter resets to 000, after counting up to 111.

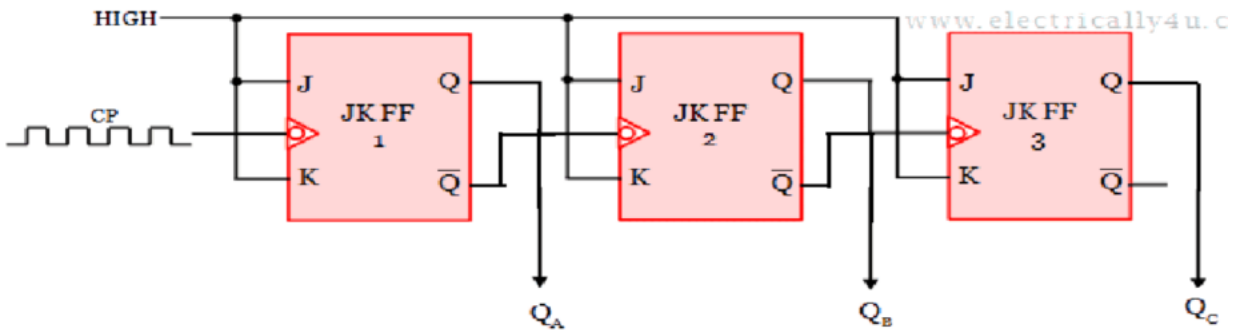


3-bit asynchronous down counter:

The down counter will count the clock pulses from maximum value to zero. In other words, for each clock pulse, the count value is decremented.

Since it is a 3-bit counter, 3 negative edge-triggered flip-flops are used. The clock pulse input is given only to the first flip-flop. The clock input of the remaining flip-flops is triggered by the Q' output of the previous flip-flop.

Since it is down counter, the 3-bit count value is measured from the (QAQBQC), where QA is the MSB and QC is the LSB.

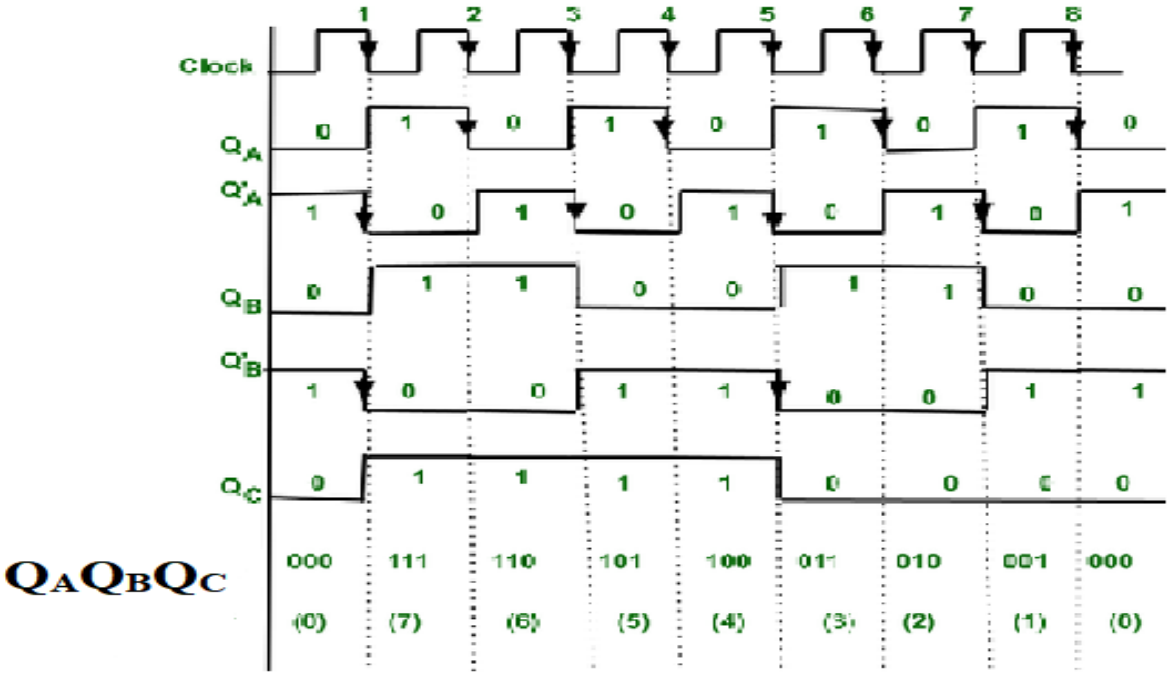


For the negative edge of the first clock pulse, output of the first flip-flop FF1 toggles its state. Thus Q1 remains at 0 and Q0 toggles to 1 and the counter state are now read as 01.

During the next negative edge of the input clock pulse FF1 toggles and Q0 = 0. The output Q0 being a clock signal for the second flip-flop FF2 and the present transition acts as a negative edge for FF2 thus toggles its state Q1 = 1. The counter state is now read as10.

For the next negative edge of the input clock to FF1 output Q0 toggles to 1. But this transition from 0 to 1 being a positive edge for FF2 output Q1 remains at 1. The counter state is now read as 11.

For the next negative edge of the input clock, Q0 toggles to 0. This transition from 1 to 0 acts as a negative edge clock for FF2 and its output Q1 toggles to 0. Thus the starting state 00 is attained. Figure shown below.



The operation is the same as that of the 3-bit asynchronous up counter. If the output is taken at the normal Q output of each flip flop, then it is an up counter. If the output is taken at the complemented output (Q') of each flip flop, it is said to be the down counter.

The change of state of each flip flop with respect to the clock pulse and the count value is shown in the below timing diagram.

In this diagram, the output waveform QC, QB, QA represents the normal Q output of JK FF3, JK FF2 and JK FF1 respectively. In the below waveform, QC‘, QB‘, QA‘ represents the complemented output from of JK FF3, JK FF2 and JK FF1 respectively.

The count value can be observed from the complemented outputs. The counter value starts from 111 and decrements its value for each clock pulse. After reaching 000, the counter resets to the maximum value(111) and starts to decrement again for the next clock pulse.

UNIT -4

SEQUENTIAL LOGIC

TOPIC 11
2-BIT SYNCHRONOUS UP COUNTER

Synchronous Counter:

Asynchronous counters are serial counters. They are slow because each Flip-Flops can change state only if all the preceding Flip-Flops have changed their state. If the clock frequency is very high, the asynchronous counter may skip some of the states. This problem is overcome in synchronous counters or parallel counters. Synchronous counters are counters in which all the flip flops are triggered simultaneously by the clock pulses Synchronous counters have a common clock pulse applied simultaneously to all flip-flops.

Design of synchronous Counter:

The synchronous counters count the number of clock pulses received at its input. It uses the same clock signal from the same source and at exactly the same time. Generally, it is constructed using either JK flip flop or T flip flop. Synchronous counters use edge-triggered flip-flops. These flip-flops change the state during the next clock pulse.

Design steps of synchronous counter:

The design steps are somewhat similar for both synchronous counter and asynchronous counter but differ slightly. Follow the below-given steps to design the synchronous counter.

- Find the number of flip flops using $2^n \geq N$, where N is the number of states and n is the number of flip flops.
- Choose the type of flip flop.
- Draw the state diagram of the counter.
- Draw the excitation table of the selected flip flop and determine the excitation table for the counter.
- Use K-map to derive the flip flop input functions.

Design 2-bit synchronous up counter using JK flip flops.

Step 1: Find the number of flip flops.

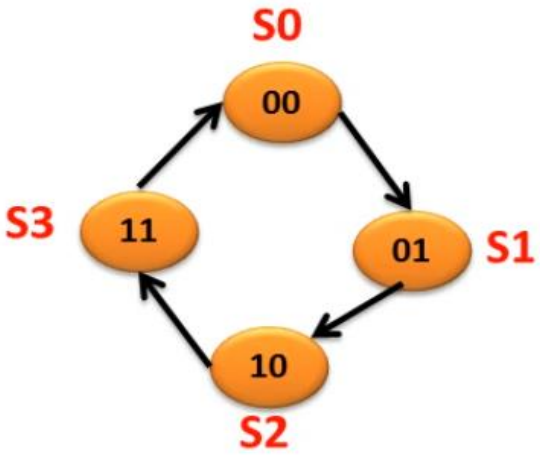
A flip flop stores only one bit, hence for a 2 bit counter, 2 flip flops(n=2) are needed to design the counter.
Number of states = $2^n = 2^2 = 4$ states(00, 01, 10,11)

Step 2: Choose the type of flip flop.

Since the type of flip flop is given in the problem, let us use JK flip flops.

Step 3: Draw state diagram for the counter.

The state diagram for the counter is drawn as below.



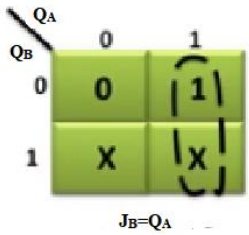
Step 4: Obtain excitation table for the counter.
We know, the excitation table for JK flip flop is given by,

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

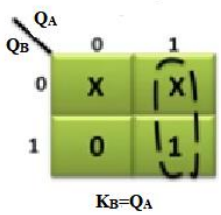
Now the excitation table for the 2-bit synchronous counter is determined from the excitation table of JK flip flop.
The excitation table is framed for 4 states of the counter. Since 2 flip-flops are used in the design, the present state, next state and flip flop inputs for each flip flop are considered.

Q_B	Q_A	Q_B^*	Q_A^*	J_B	K_B	J_A	K_A
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

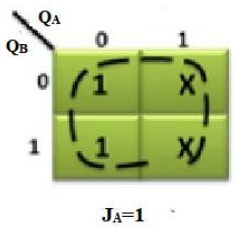
Step 5: Derive the flip flop input functions.
Using Karnaugh maps, the input functions for the 2 flip flops are derived. The present states are the input for all the flip-flops. Since there are three inputs(Q_B, Q_A), 4 cell K-map is used.



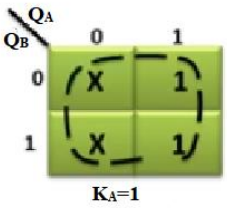
$J_B=Q_A$



$K_B=Q_A$

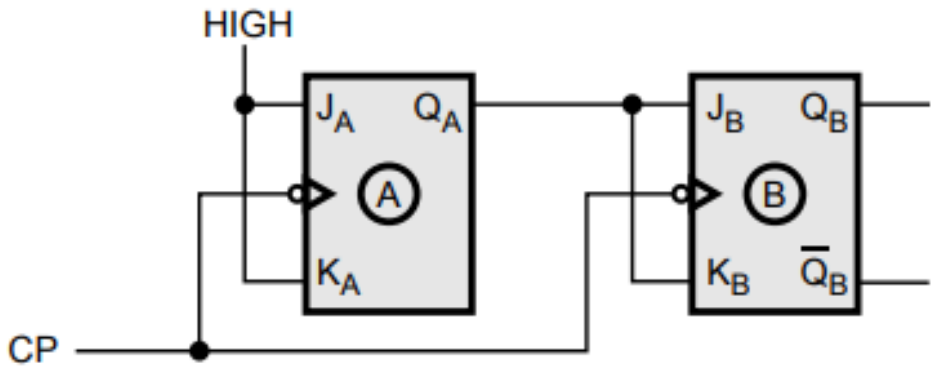


$J_A=1$

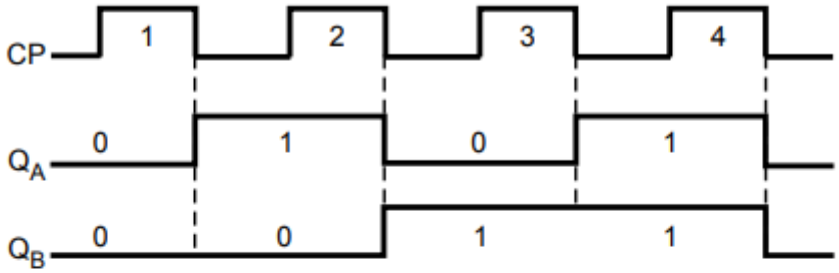


$K_A=1$

Step 6: Draw the logic diagram of the counter.
The logic diagram of the 2-bit synchronous counter is drawn as follows. Draw the 2 JK flip-flops. The common clock pulse input is given to all the flip-flops. The inputs for each flip-flop are drawn as per the logic functions derived in the previous step.



A two-bit synchronous binary counter



(a)

CP	Q_B	Q_A
0	0	0
1	0	1
2	1	0
3	1	1

(b)

Timing diagram and state sequence for the 2-bit synchronous counter

UNIT -4

SEQUENTIAL LOGIC

TOPIC 12
3-BIT SYNCHRONOUS COUNTER

Synchronous Counter:

An Asynchronous counters are serial counters. They are slow because each Flip-Flops can change state only if all the preceding Flip-Flops have changed their state. if the clock frequency is very high, the asynchronous counter may skip some of the states. This problem is overcome in synchronous counters or parallel counters. Synchronous counters are counters in which all the flip flops are triggered simultaneously by the clock pulses Synchronous counters have a common clock pulse applied simultaneously to all flip-flops.

Design of synchronous Counter:

The synchronous counters count the number of clock pulses received at its input. It uses the same clock signal from the same source and at exactly the same time. Generally, it is constructed using either JK flip flop or T flip flop. Synchronous counters use edge-triggered flip-flops. These flip-flops change the state during the next clock pulse.

Design steps of synchronous counter:

The design steps are somewhat similar for both synchronous counter and asynchronous counter but differ slightly. Follow the below-given steps to design the synchronous counter.

Find the number of flip flops using $2^n \geq N$, where N is the number of states and n is the number of flip flops.

- Choose the type of flip flop.
- Draw the state diagram of the counter.
- Draw the excitation table of the selected flip flop and determine the excitation table for the counter.
- Use K-map to derive the flip flop input functions.

Design 3-bit synchronous up counter using JK flip flops.

Step 1: Find the number of flip flops.

A flip flop stores only one bit, hence for a 3 bit counter, 3 flip flops(n=3) are needed to design the counter.

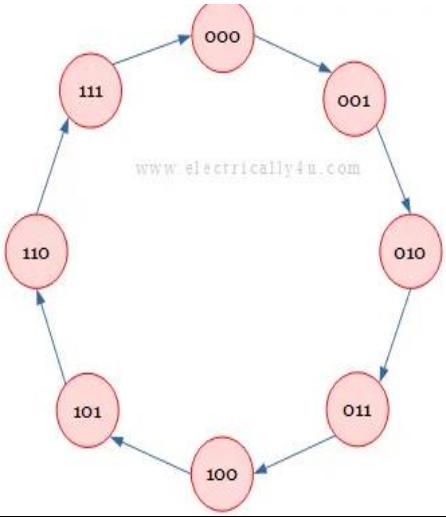
Number of states = $2^n = 2^3 = 8$ states(000, 001, 010, 011, 100, 101, 110, 111)

Step 2: Choose the type of flip flop.

Since the type of flip flop is given in the problem, let us use JK flip flops.

Step 3: Draw state diagram for the counter.

The state diagram for the counter is drawn as below.



Step 4: Obtain excitation table for the counter.
We know, the excitation table for JK flip flop is given by,

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

Now the excitation table for the 3-bit synchronous counter is determined from the excitation table of JK flip flop.
The excitation table is framed for 8 states of the counter. Since 3 flip-flops are used in the design, the present state, next state and flip flop inputs for each flip flop are considered.

Clock	Present State			Next State			Flip flop Inputs					
	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
1	0	0	0	0	0	1	0	x	0	x	1	x
2	0	0	1	0	1	0	0	x	1	x	x	1
3	0	1	0	0	1	1	0	x	x	0	1	x
4	0	1	1	1	0	0	1	x	x	1	x	1
5	1	0	0	1	0	1	x	0	0	x	1	x
6	1	0	1	1	1	0	x	0	1	x	x	1
7	1	1	0	1	1	1	x	0	x	0	1	x
8	1	1	1	0	0	0	x	1	x	1	x	1

Step 5: Derive the flip flop input functions.
Using Karnaugh maps, the input functions for the 3 flip flops are derived. The present states are the input for all the flip-flops. Since there are three inputs(Q_C, Q_B, Q_A), 8 cell K-map is used.

$J_C = \Sigma m(3)+\Sigma d(4,5,6,7)$ $K_C = \Sigma m(7)+\Sigma d(0,1,2,3)$
$J_B = \Sigma m(1,5)+\Sigma d(2,3,6,7)$ $K_B = \Sigma m(3,7)+\Sigma d(0,1,4,5)$
$J_A = \Sigma m(0,2,4,6)+\Sigma d(1,3,5,7)$ $K_A = \Sigma m(1,3,5,7)+\Sigma d(0,2,4,6)$

For J_C

Q_C \ Q_BQ_A	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_C = Q_BQ_A$

For K_C

Q_C \ Q_BQ_A	00	01	11	10
0	X	X	X	X
1	0	0	1	0

$K_C = Q_BQ_A$

For J_B

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	X	X
1	0	1	X	X

$J_B = Q_A$

For K_B

Q_C \ Q_BQ_A	00	01	11	10
0	X	X	1	0
1	X	X	1	0

$K_B = Q_A$

For J_A

Q_C \ Q_BQ_A	00	01	11	10
0	1	X	X	1
1	1	X	X	1

$J_A = 1$

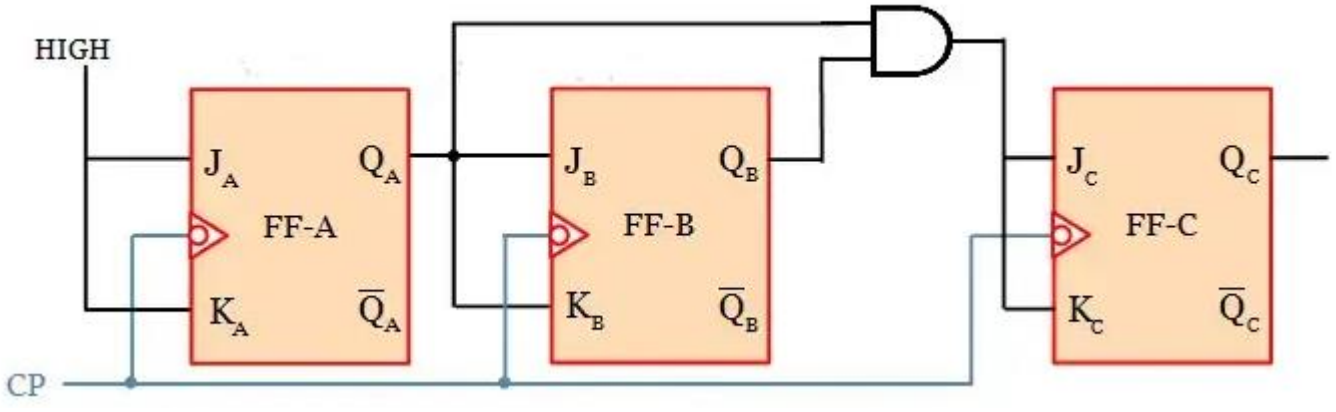
For K_A

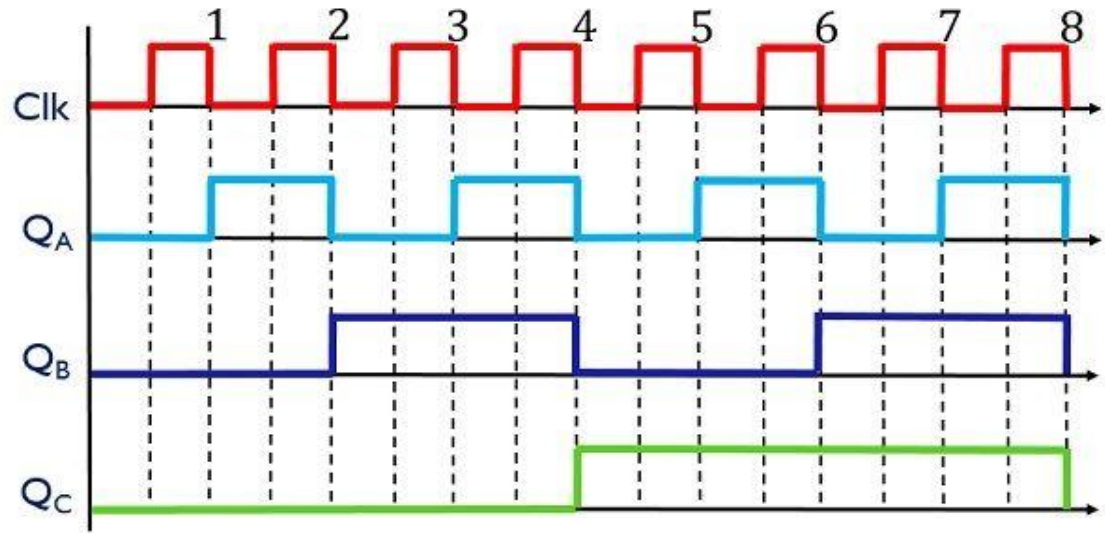
Q_C \ Q_BQ_A	00	01	11	10
0	X	1	1	X
1	X	1	1	X

$K_A = 1$

Step 6: Draw the logic diagram of the counter.

The logic diagram of the 3-bit synchronous counter is drawn as follows. Draw the 3 JK flip-flops. The common clock pulse input is given to all the flip-flops. The inputs for each flip-flop are drawn as per the logic functions derived in the previous step.





Timing Diagram of 3-bit Synchronous Counter

Electronics Coach

3 bit Synchronous Down Counter :

- In synchronous counter clock is provided to all the flip-flops simultaneously.
- Circuit becomes complex as the number of states increases.
- Speed is high.

Design : The steps involves in design are

1. Decide the number of Flip flops –

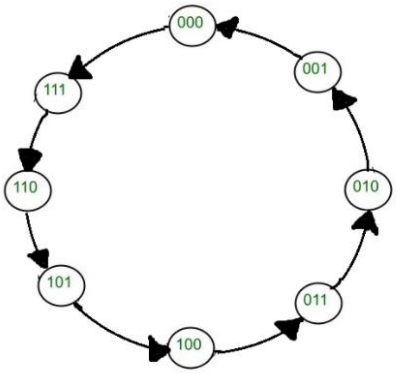
- For 3 bit counter we require 3 FF.
- **Maximum count** = 2^n-1, where n is a number of bits.
- For n= 3, Maximum count = 7.
- Here T FF is used.

2. Write excitation table of FF –

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

Excitation table of T flip flop

2. **Draw State diagram and circuit excitation table –**
Number of states = 2^n, where n is number of bits.

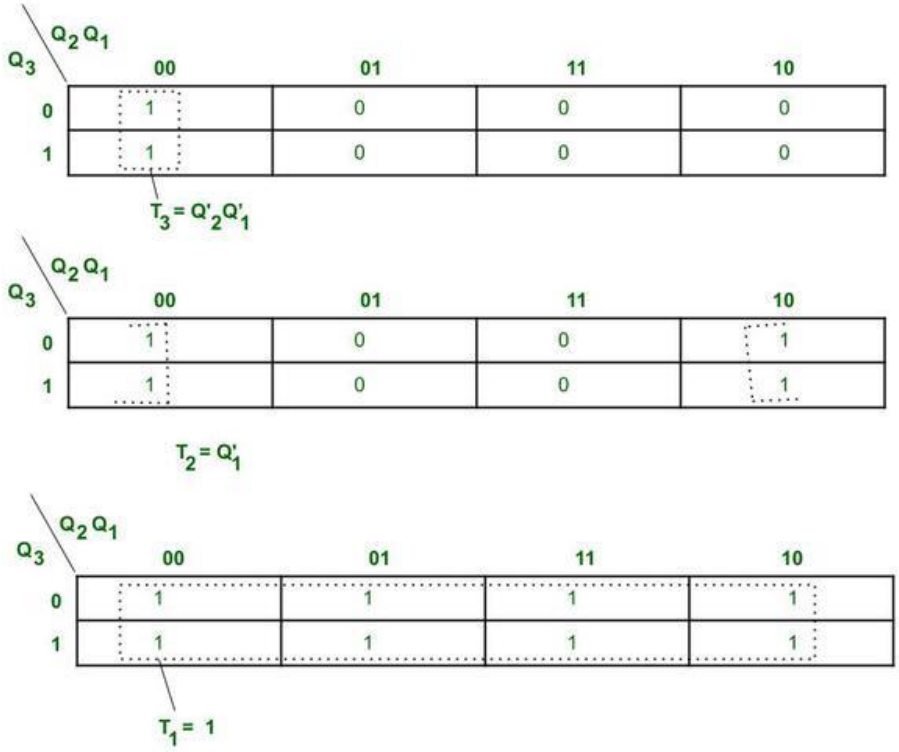


Previous state			Next state					
Q_3	Q_2	Q_1	Q^*_3	Q^*_2	Q^*_1	T_3	T_2	T_1
0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0	1
1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	0	0	1

Here T = 1, then there is output state(next state changes from previous state) changes i.e Q changes from 0 to 1 or 1 to 0

T= 0 then, there is no state output state changes i.e Q remains same .

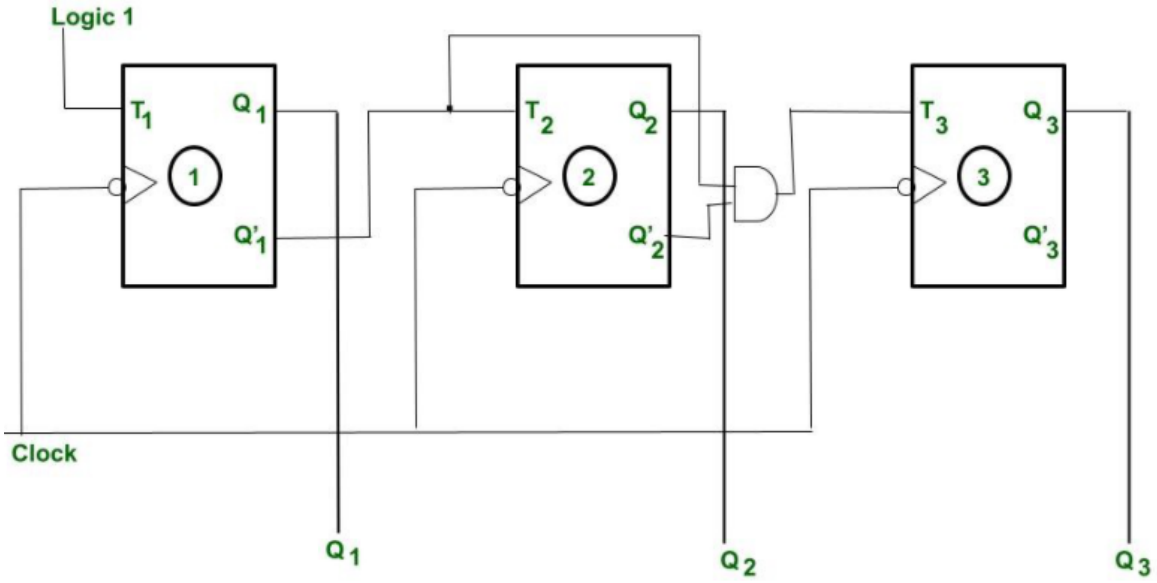
4. Find simplified equation using k map –

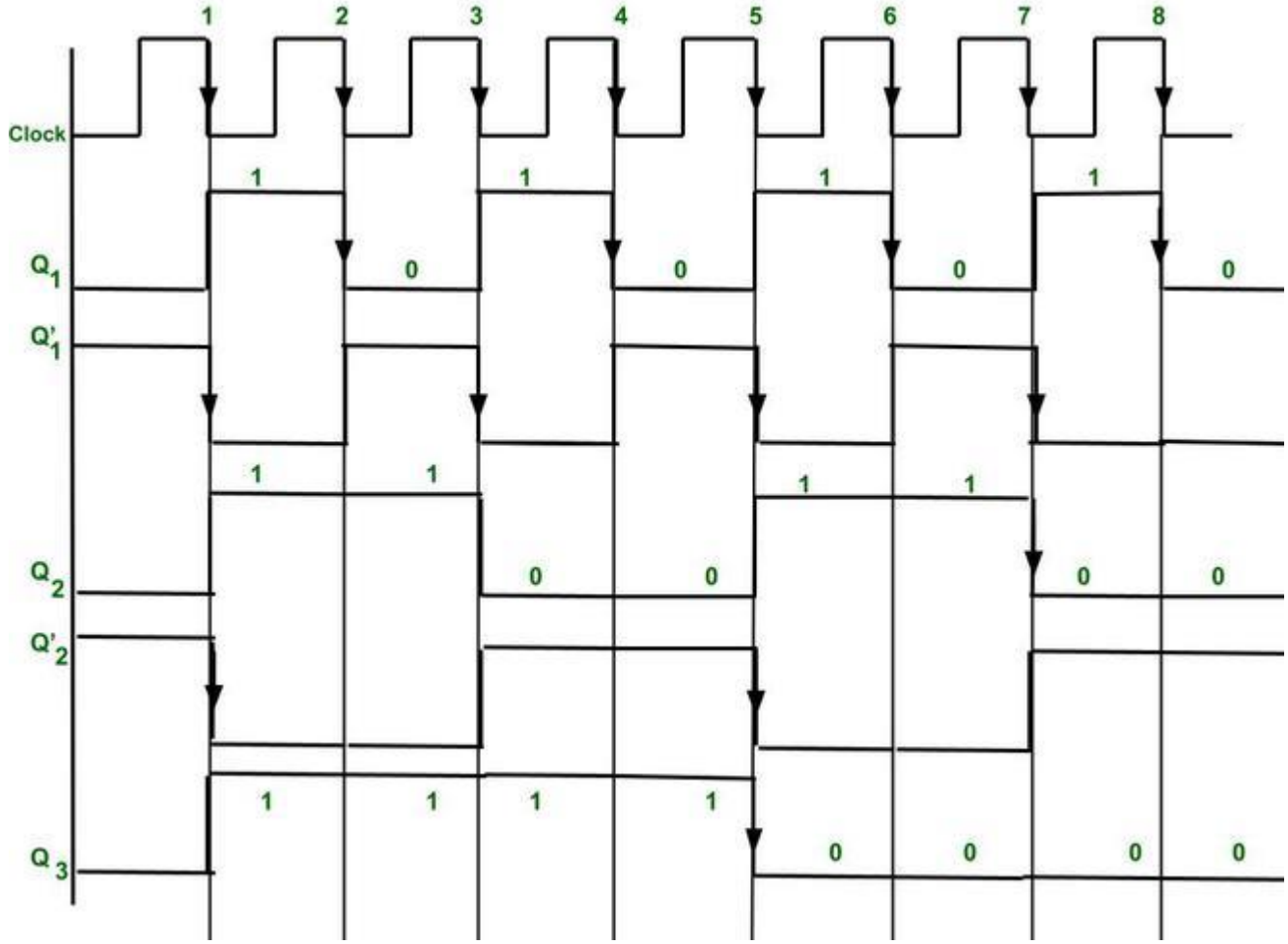


K map for 3 bit synchronous down counter

5. Draw the logic diagram –

The clock is provided to every Flip flop at same instant of time.
The toggle(T) input is provided to every Flip flop according to the simplified equation of K map.





Timing diagram of 3 bit synchronous Down counter.

Working Principle of Synchronous 3-Bit down Counter :

Here -ve edge triggered clock is used for toggling purpose.

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

Excitation table of T flip flop

As we see from characteristics table when T = 1, then toggling takes place and T =0 then it stores the output state.

- Initially $Q_3 = 0$, $Q_2 = 0$, $Q_1 = 0$.
- In simplified equation of K-map we get $T_1 = 1$, therefore Flip flop 1 output Q_1 is toggle for every negative edge(because clock is negative edge triggered). Flip-flop(FF) 2, toggle input(T_2) is connected to Q_1'. Therefore, Flip Flop 2 output state Q_2 is toggle only when there is clock falling edge (i.e negative edge triggering) and Q_1' =1.
- Similarly, Flip flop 3 toggle input(T_3) is connected to Q_2' and Q_1'. Therefore, Flip flop 3 output is toggle when there is clock falling edge and Q_2'=1 and Q_1' = 1 .(as you can see from timing diagram)
- Therefore, we get output as down counting Q_3(MSB) Q_2 Q_1(LSB) after 8th negative edge triggered clock the output of the three Flip flops again becomes to initial state $Q_3 = 0$, $Q_2 = 0$, $Q_1 = 0$.
- We get output(state changes) after every negative edge clock pulse.
- With 3 T- Flip flops we get output as 2^3-1= 7 to 0.

UNIT -4
SEQUENTIAL LOGIC
TOPIC – 13
MOD-N-COUNTER

Design a Synchronous Mod-6 up Counter:

Mod represents number of states in a Counter. To design a Mod counter the following steps are followed.

Step 1: Find the number of flip flops.

Mod-6 counter represents that the counter will have 6 states. Thus, N =6.

The number of flip-flops used for counter design is determined using the formula, $2^n \geq N$.

By trial and error method, the value of n is found to be 3. That is the number of flip-flops, n = 3.

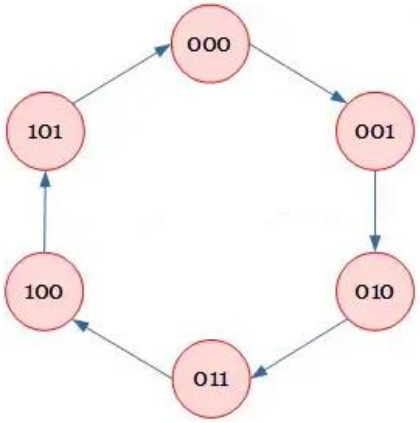
Hence the 6 counter states are 000, 001, 010, 011, 100, 101.

Step 2: Choose the type of flip flop.

Let us choose the JK- flip flop to design the Mod-6 synchronous up counter.

Step 3: Draw state diagram for the counter.

The state diagram for the mod-6 counter with 6 states is drawn as below.



Step 4: Obtain an excitation table for the counter.

The excitation table for JK flip flop is given by,

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

Now the excitation table for the mod-6 synchronous counter is determined from the excitation table of JK flip flop.

The excitation table is framed for 6 states of the counter. Since 3 flip-flops are used in the design, the present state, next state and flip flop inputs for each flip flop are considered.

Clock	Present State			Next State			Flip flop Inputs					
	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
1	0	0	0	0	0	1	0	x	0	x	1	x
2	0	0	1	0	1	0	0	x	1	x	x	1
3	0	1	0	0	1	1	0	x	x	0	1	x
4	0	1	1	1	0	0	1	x	x	1	x	1
5	1	0	0	1	0	1	x	0	0	x	1	x
6	1	0	1	0	0	0	x	1	0	x	x	1
-	1	1	0	x	x	x	x	x	x	x	x	x
-	1	1	1	x	x	x	x	x	x	x	x	x

Step 5: Derive the flip flop input functions.
Using K-maps, the input functions for the 3 flip flops are derived. The present states are the input for all the flip-flops. Since there are three inputs(QC, QB, QA), 8 cell K-map is used.

For J_C

Q_C \ Q_BQ_A	00	01	11	10
0	0	0	1	0
1	x	x	x	x

$J_C = Q_BQ_A$

For K_C

Q_C \ Q_BQ_A	00	01	11	10
0	x	x	x	x
1	0	1	x	x

$K_C = Q_A$

For J_B

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	x	x
1	0	0	x	x

$J_B = \overline{Q}_C Q_A$

For K_B

Q_C \ Q_BQ_A	00	01	11	10
0	x	x	1	0
1	x	x	x	x

$K_B = Q_A$

For J_A

Q_C \ Q_BQ_A	00	01	11	10
0	1	x	x	1
1	1	x	x	x

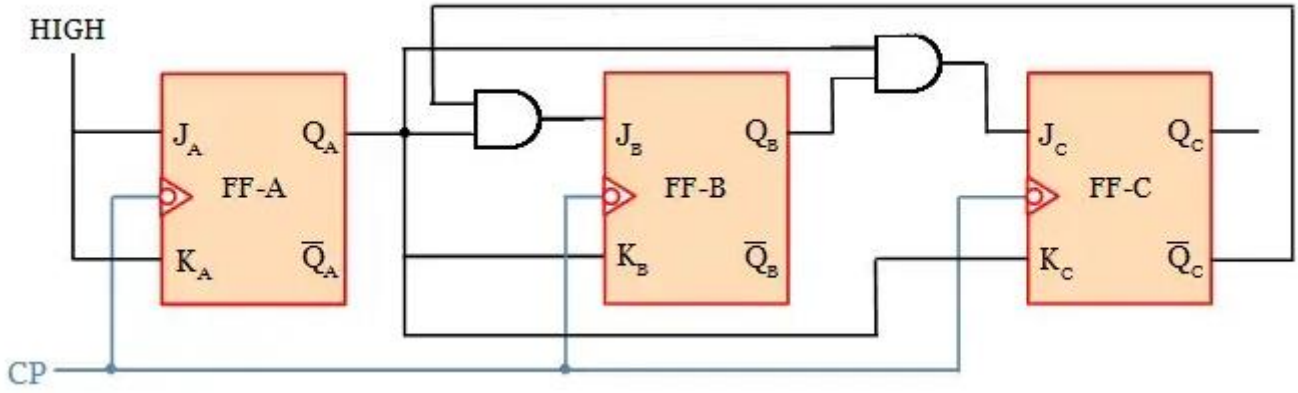
$J_A = 1$

For K_A

Q_C \ Q_BQ_A	00	01	11	10
0	x	1	1	x
1	x	1	x	x

$K_A = 1$

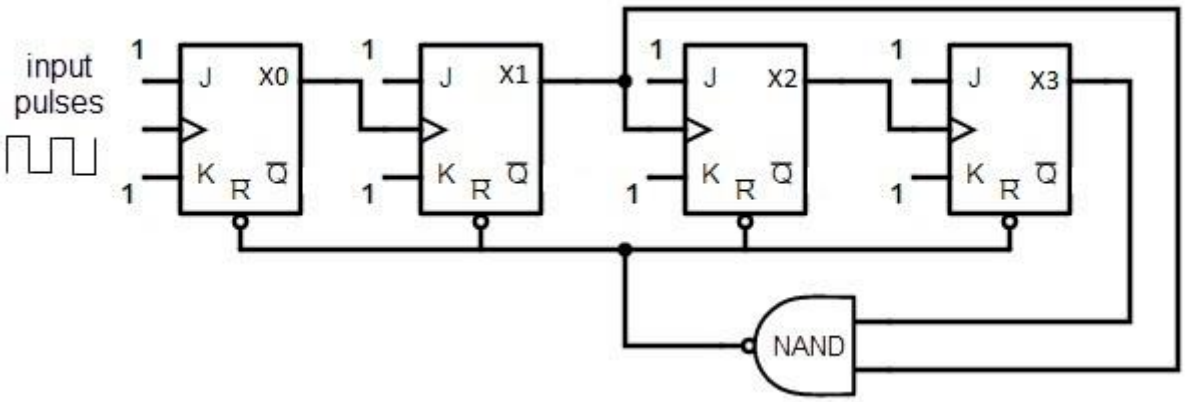
Step 6: Draw the logic diagram of the counter.
The logic diagram of the mod-6 synchronous up counter is drawn as follows. Draw the 3 JK flip-flops. The inputs for each flip-flop are drawn as per the logic functions derived in the previous step. The clock pulse is given to all the flip-flops.



Asynchronous BCD or Decade Counter Circuit:

A binary coded decimal (BCD) is a serial digital counter that counts ten digits .And it resets for every new clock input. As it can go through 10 unique combinations of output, it is also called as "Decade counter". A BCD counter can count 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, and 1001 and so on.

A 4 bit binary counter will act as decade counter by skipping any six outputs out of the 16 (2^4) outputs. There are some available ICs for decade counters which we can readily use in our circuit, like 74LS90. It is an asynchronous decade counter.



The above figure shows a decade counter constructed with JK flip flop. The J output and K outputs are connected to logic 1. The clock input of every flip flop is connected to the output previous flip flop, except the last one.

The output of the NAND gate is connected in parallel to the clear input 'CLR' to all the flip flops. This ripple counter can count up to 16 i.e. 2^4.

Decade Counter Operation

When the Decade counter is at REST, the count is equal to 0000. This is first stage of the counter cycle. When we connect a clock signal input to the counter circuit, then the circuit will count the binary sequence. The first clock pulse can make the circuit to count up to 9 (1001). The next clock pulse advances to count 10 (1010).

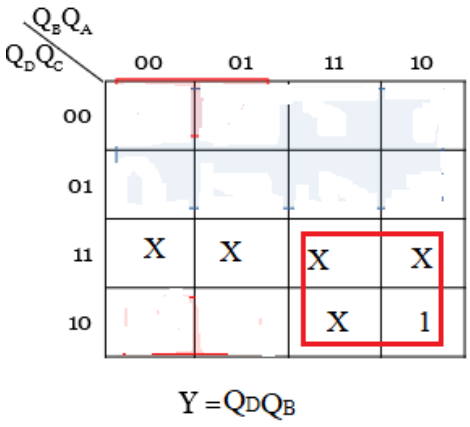
Then the ports X1 and X3 will be high. As we know that for high inputs, the NAND gate output will be low. The NAND gate output is connected to clear input, so it resets all the flip flop stages in decade counter. This means the pulse after count 9 will again start the count from count 0.

Truth Table of Decade Counter

Input Pulses	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0(resets)

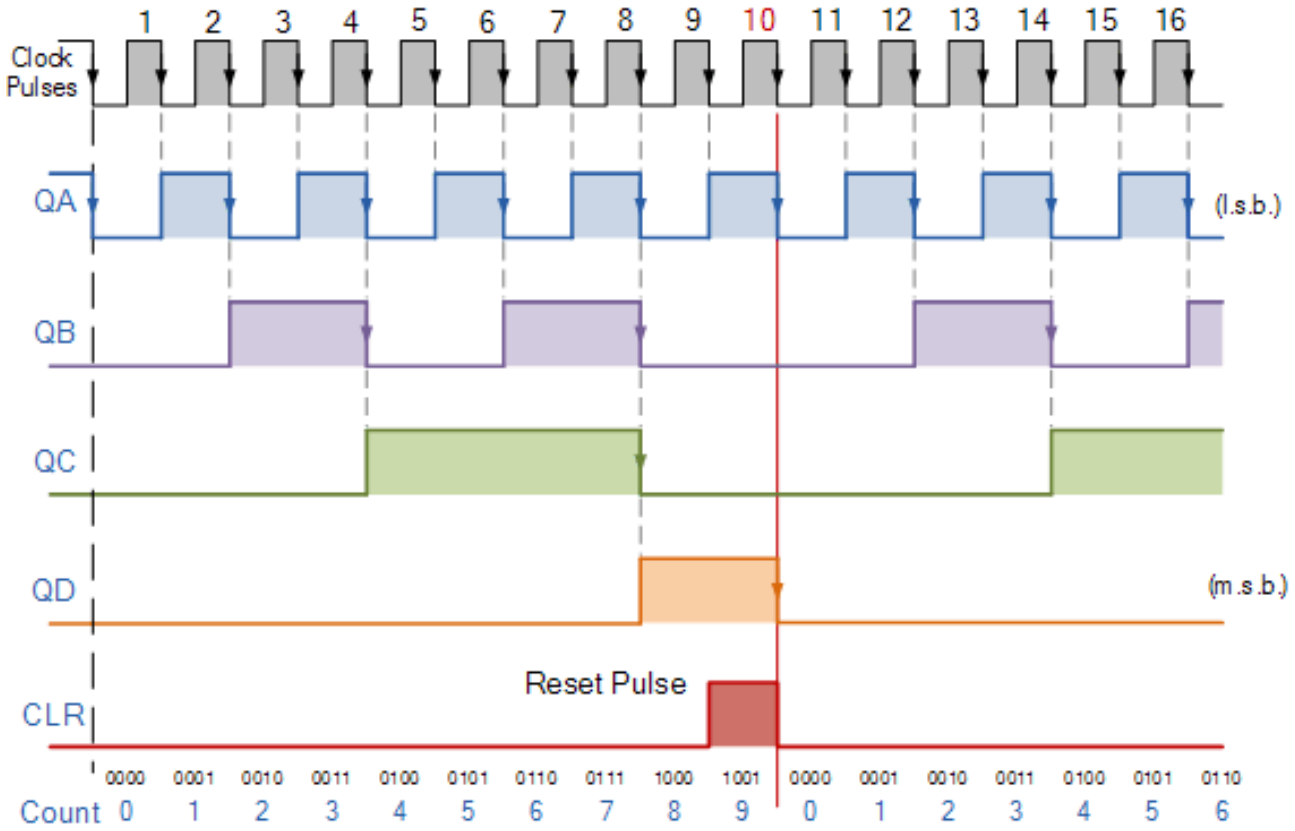
The above table describes the counting operation of Decade counter. It represents the count of circuit for decimal count of input pulses. The NAND gate output is zero when the count reaches 10 (1010).

The count is decoded by the inputs of NAND gate X1 and X3. After count 10, the logic gate NAND will trigger its output from 1 to 0, and it resets all flip flops.



$Y = Q_D Q_B$

Clock	BCD Counter				Output of Reset Logic Y
	Q_D	Q_C	Q_B	Q_A	
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1



UNIT -4
SEQUENTIAL LOGIC
TOPIC – 14
JOHNSON -COUNTER

Shift register counters:

One of the applications of shift register is that they can be arranged to form several types of counters. The most widely used shift register counter is ring counter as well as the twisted ring counter.

Ring counter: this is the simplest shift register counter. The basic ring counter using D flip- flops is shown in fig. the realization of this counter using JK FFs. The Q output of each stage is connected to the D flip-flop connected back to the ring counter.

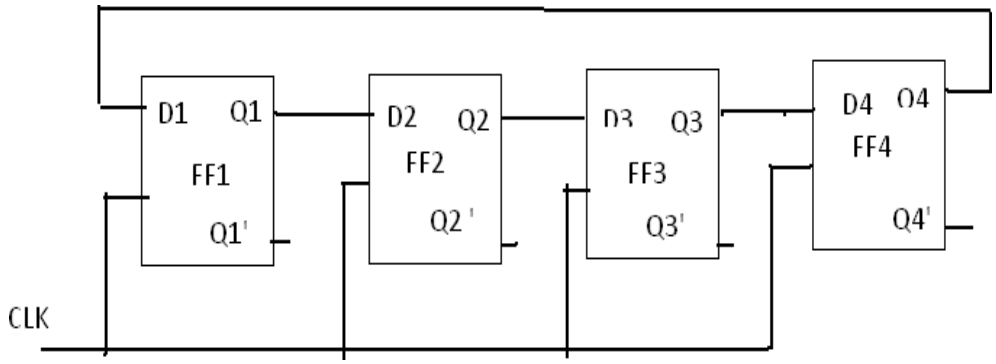
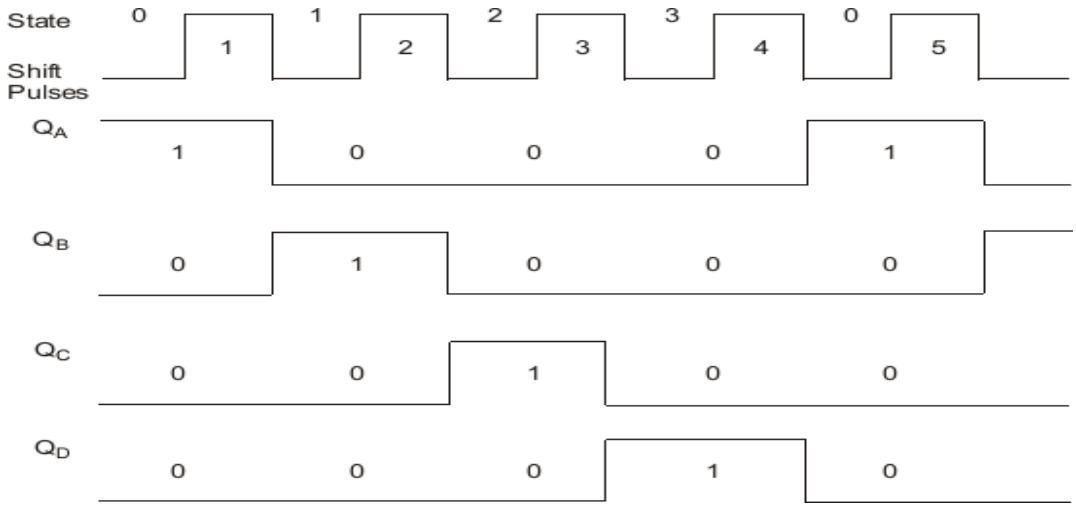


fig: logic diagram of 4-bit ring counter using D flip-flops

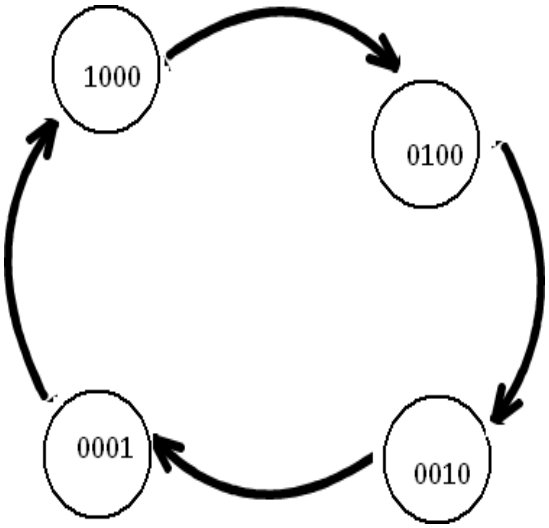
Only a single 1 is in the register and is made to circulate around the register as long as clock pulses are applied. Initially the first FF is present to a 1. So, the initial state is 1000, i.e., Q1=1, Q2=0,Q3=0,Q4=0. After each clock pulse, the contents of the register are shifted to the right by one bit and Q4 is shifted back to Q1. The sequence repeats after four clock pulses.

The number of distinct states in the ring counter, i.e., the mod of the ring counter is equal to number of FFs used in the counter. An n-bit ring counter can count only n bits, where as n-bit ripple counter can count 2^n bits. So, the ring counter is uneconomical compared to a ripple counter but has advantage of requiring no decoder, since we can read the count by simply noting which FF is set. Since it is entirely a synchronous operation and requires no gates external FFs, it has the further advantage ofbeing very fast.

Timing diagram:



State diagram:



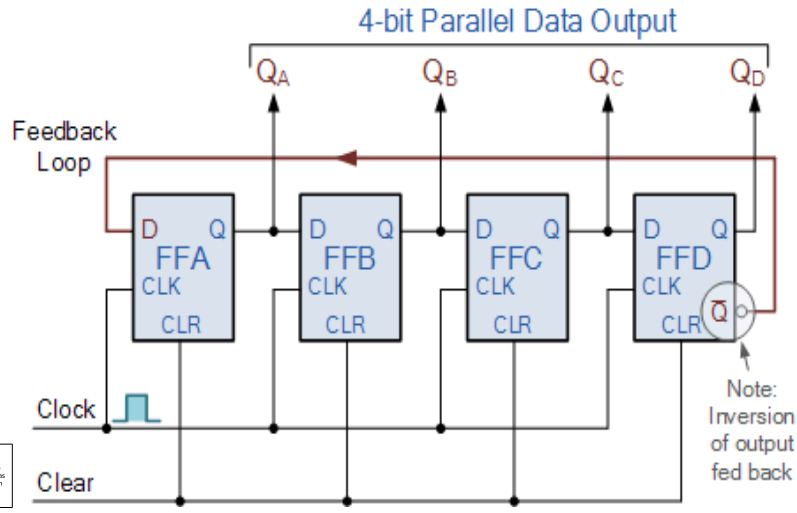
Johnson Ring Counter/Twisted Ring Counters:

- Johnson ring counter is a type of counter created using shift registers,the Johnson Ring Counter or "Twisted Ring Counters", is another shift register with feedback exactly the same as the standard Ring Counter , except that this time the inverted output Q of the last flip-flop is now connected back to the input D of the first flip-flop as shown below.
- The main advantage of this type of ring counter is that it only needs half the number of flip-flops compared to the standard ring counter then its modulo number is halved. So a "n-stage" Johnson counter will circulate a single data bit giving sequence of 2n different states and can therefore be considered as a "mod-2n counter".
- This inversion of Q before it is fed back to input D causes the counter to "count" in a different way.
- Instead of counting through a fixed set of patterns like the normal ring counter such as for a 4-bit counter, "0001"(1), "0010"(2), "0100"(4), "1000"(8) and repeat, the Johnson counter counts up and then down as the initial logic "1" passes through it to the right replacing the preceding logic "0".

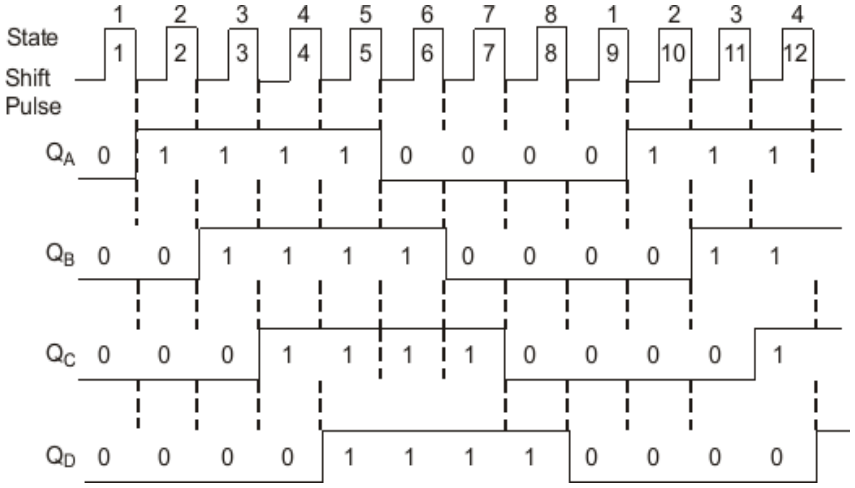
- A 4-bit Johnson ring counter passes blocks of four logic "0" and then four logic "1" thereby producing an 8-bit pattern. As the inverted output Q is connected to the input D this 8-bit pattern continually repeats. For example, "1000", "1100", "1110", "1111", "0111", "0011", "0001", "0000" .

Clock pulse	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

Four-bit Johnson sequence



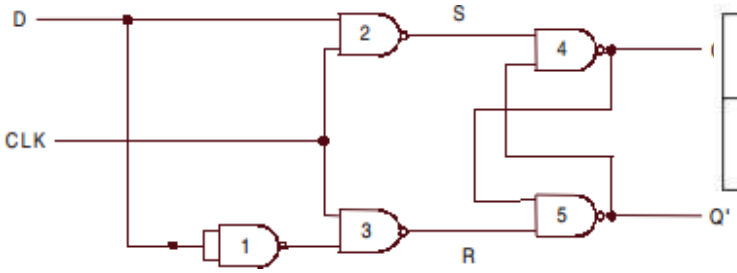
Timing diagram:



UNIT -4
SEQUENTIAL LOGIC
TOPIC – 3.2
FLIP-FLOP(D)

CLOCKED D FLIP-FLOP

One way to eliminate the undesirable condition of the indeterminate state in the *SR* latch is to ensure that inputs *S* and *R* are never equal to 1 at the same time. This is done in the *D* latch. The D flip-flop has only one input referred to as the D (**data**) input & two outputs as usual Q and Q'. It transfers the data at the input after the delay of one clock pulse at the output Q. So in some cases the input is referred to as a delay input and the flip-flop gets the name ***delay*** (D) flip-flop. It can be easily constructed from an S-R flip-flop by simply incorporating an inverter between S and R such that the input of the inverter is at the S end & the output of the inverter is at the R end. We can get rid of the undefined condition, *i.e.*, S = R = 1 condition, of the S-R flip-flop in the D flip flop. The D flip-flop is either used as a delay device or as a latch to store one bit of binary information. The truth table of D flip-flop is given in the table below. The structure of the D flip-flop is shown in Figure below, which is being constructed using NAND gates.



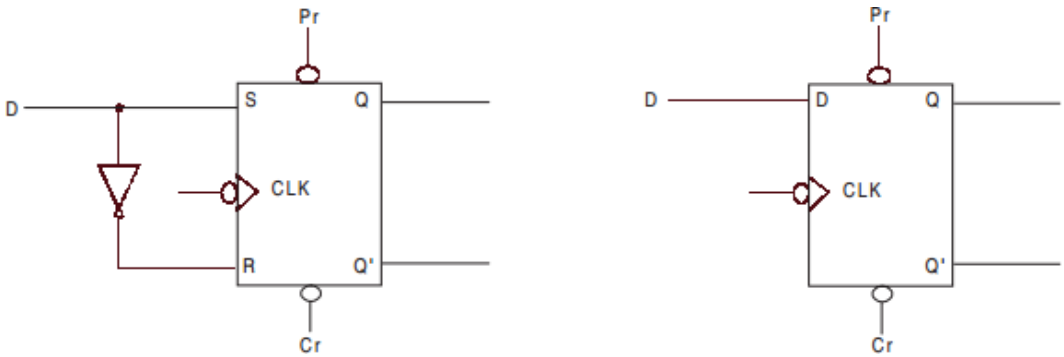
Input D_n	*Output* Q_{n+1}
0	0
1	1

Case 1. If the CLK input is low, the value of the D input has no effect, since the S and R inputs of the basicNAND flip-flop are kept as 1.

Case 2. If the CLK = 1 and D = 1, the NAND gate 1 produces 0, which forces the output of NAND gate 3 as 1.On the other hand, both the inputs of NAND gate 2 are 1, which gives the output of gate 2 as 0. Hence, output of NAND gate 4 is forced to be 1, *i.e.*, Q = 1, whereas both the inputs of gate 5 are 1 and the output is 0, *i.e.*, Q' = 0. Hence, we find that when D = 1, after one clock pulse passes Q = 1, which means the output follows D. **Case 3.** If the CLK = 1, and D = 0, the NAND gate 1 produces 1. Hence both the inputs of NAND gate 3 are 1,which gives the output of gate 3 as 0. On the other hand, D = 0 forces the output of NAND gate 2 to be 1.

Hence the output of NAND gate 5 is forced to be 1, *i.e.*, Q' = 1, whereas both the inputs of gate 4 are 1 and theoutput is 0, *i.e.*, Q = 0. Hence, we find that when D = 0, after one clock pulse passes Q = 0, which means the output again follows D.

A D flip-flop is most often used in the construction of sequential circuits like registers.



Characteristic Table of a D Flip-flop

The characteristic table of a D flip-flop is given in the table below. From the characteristic table we have to find out the characteristic equation of the D flip-flop.

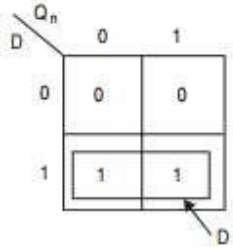
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation table of D flip flop

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

Truth table of D flip flop

Now we will find out the characteristic equation of the D flip-flop from the characteristic table with the help ofthe Karnaugh map:-



Hence, the characteristic equation of a D flip-flop is $Q_{n+1} = D$

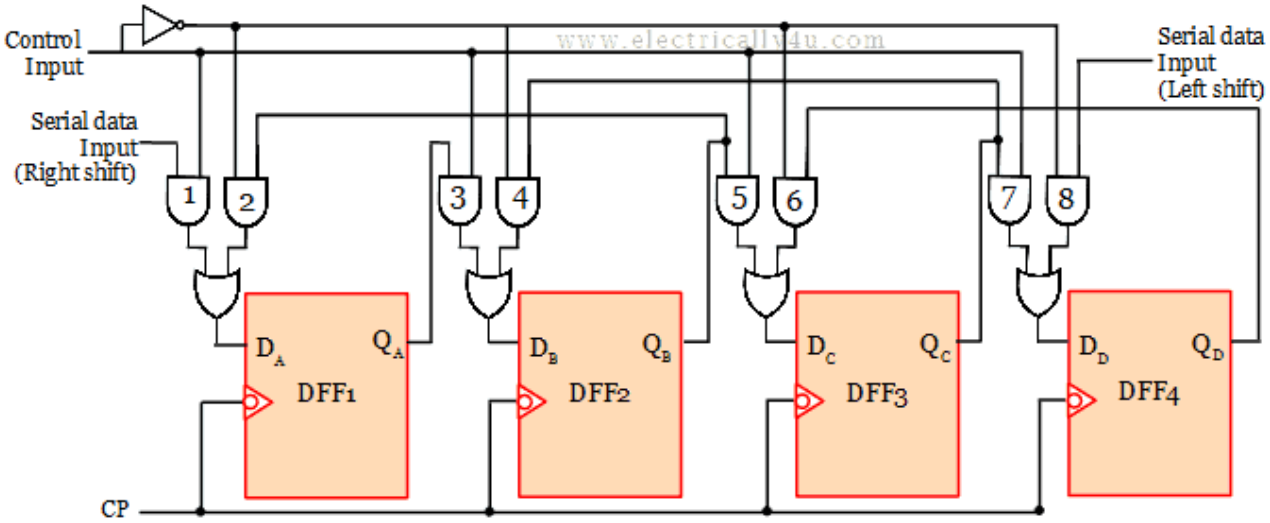
UNIT -4
SEQUENTIAL LOGIC
TOPIC – 8.2
BI DIRECTIONAL AND UNIVERSAL SHIFT REGISTERS

Bidirectional Shift Register:

The Serial In Serial Out shift register(SISO) will either shift the data from left to right or from right to left. But the bidirectional shift register will be capable of shifting the data in both directions based on the selected mode or control input.

If the control input is given logic 0, it will disable AND gates 1, 3, 5, 7 and enable gates 2, 4, 6,8. The

enabled AND gates will make the data be shifted from right to left. The following figure shows the block diagram of the bidirectional shift register.

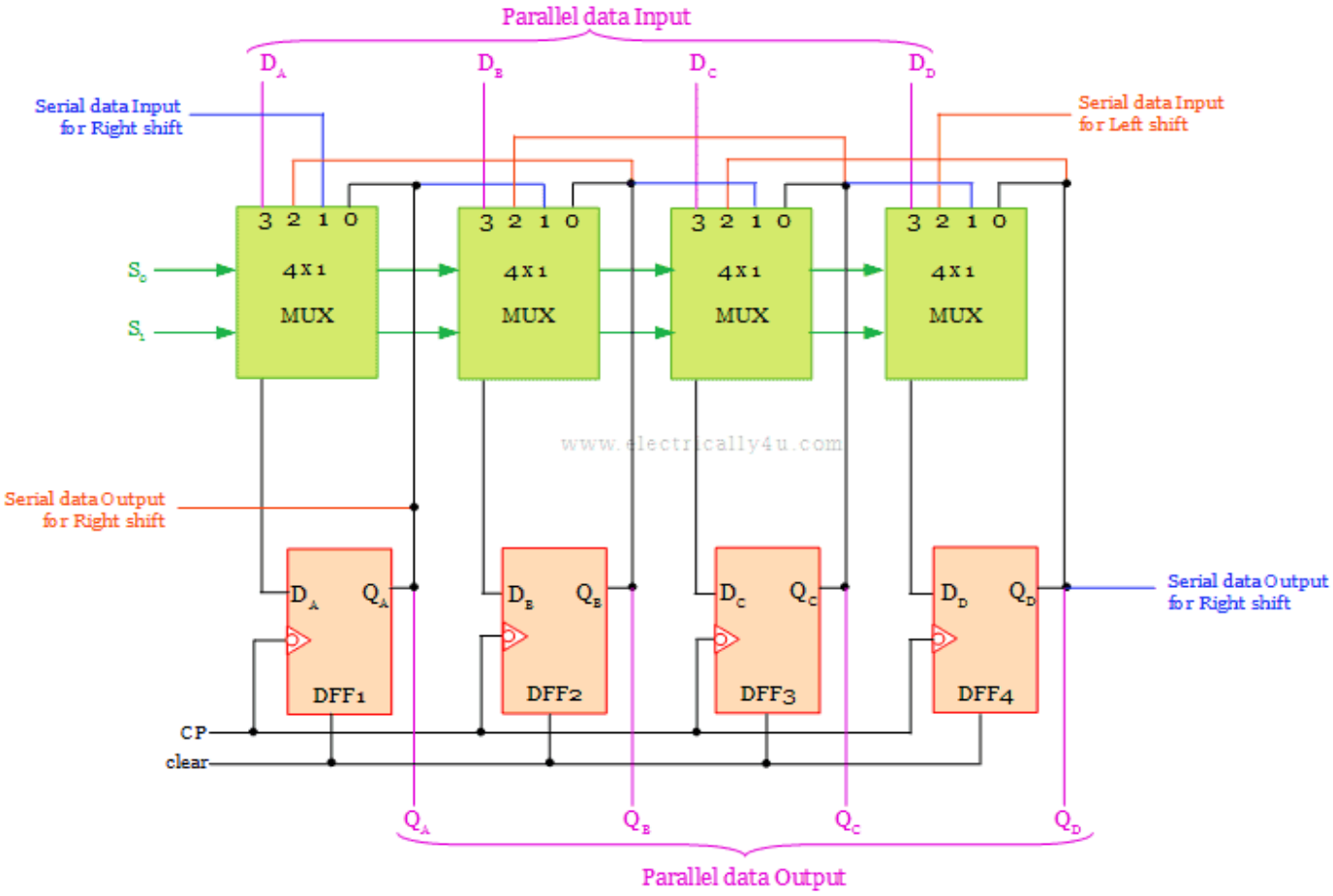


When the control input is given as Logic 1, the AND gates 1, 3, 5, 7 will be enabled and gates 2, 4, 6, 8 will be disabled. This will shift the data from left to right.

Universal Shift Register:

It is a multi-function shift register, which can perform any type of data bit movement across the registers. The data can be loaded and it can be retrieved in either serial or in a parallel way.

Thus it can perform any type of operation like serial to serial operation(both shift left and shift right), parallel to parallel conversion, serial to parallel and parallel to serial conversion. Hence the name Universal shift register.The block diagram of the universal shift register is shown below.



As you can see from the block diagram, it has four D flip-flops and four multiplexers. All the multiplexers have the same select lines(S_0 and S_1, shown in green color). The select line is used to select the mode of the shift register.

Depending upon the selection line input, the shift register operates. If $S_1S_0 = 00$, there will not be any change in the shift register output. If $S_1S_0 = 01$, then serial shift right operation(shown in blue color) will take place.

When the selection input $S_1S_0 = 10$, the data will be shifted left, as shown in red color. When $S_1S_0 = 11$, the data will be loaded parallelly through the inputs $D_AD_BD_CD_D$ and retrieved from each flip-flop at $Q_AQ_BQ_CQ_D$(shown in pink color).

Universal shift registers are used in the arithmetic operation of binary numbers where the shifting of data is necessary.

UNIT -4

SEQUENTIAL LOGIC

TOPIC 11.2
2-BIT SYNCHRONOUS DOWN COUNTER

2 bit Synchronous Down Counter :

- In synchronous counter clock is provided to all the flip-flops simultaneously.
- Circuit becomes complex as the number of states increases.
- Speed is high.

Design : The steps involves in design are

1. Decide the number of Flip flops –

- For 2 bit counter we require 2 FF.
- **Maximum count** = 2^n-1, where n is a number of bits.
- For n= 2, Maximum count = 3.
- Here T FF is used.

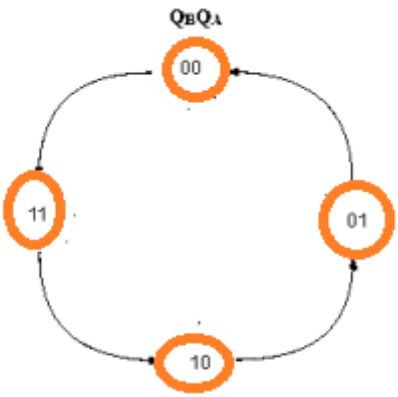
2. Write excitation table of FF –

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

Excitation table of T flip flop

3.Draw State diagram and circuit excitation table –

Number of states = 2^n, where n is number of bits.

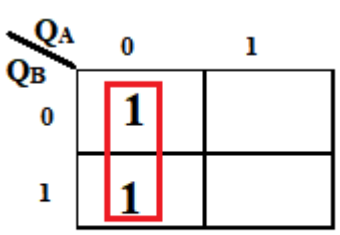


Excitation table:

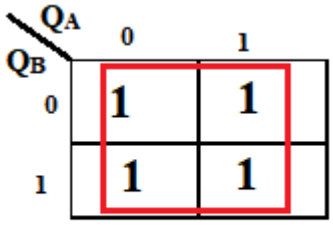
Present State		Next State		Flip Flop	
Q_B	Q_A	Q_{B+1}	Q_{A+1}	T_B	T_A
0	0	1	1	1	1
0	1	0	0	0	1
1	0	0	1	1	1
1	1	1	0	0	1

Here T = 1, then there is output state(next state changes from previous state) changes i.e Q changes from 0 to 1 or 1 to 0
T= 0 then, there is no state output state changes i.e Q remains same .

4. Find simplified equation using k map –



$T_B=Q_A'$

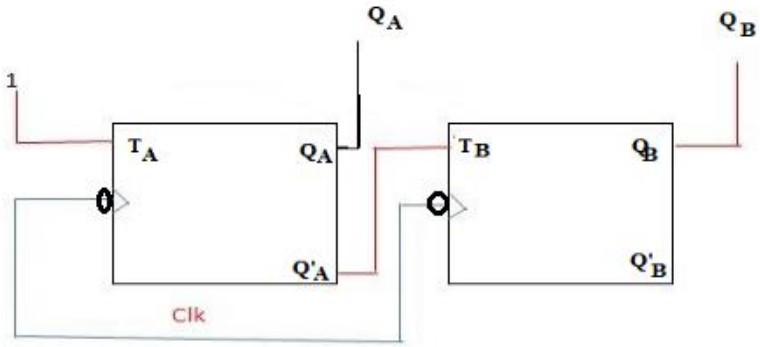


T_A=Logic 1

K map for 2 bit synchronous down counter

5. Draw the logic diagram –

The clock is provided to every Flip flop at same instant of time.
The toggle(T) input is provided to every Flip flop according to the simplified equation of K map.



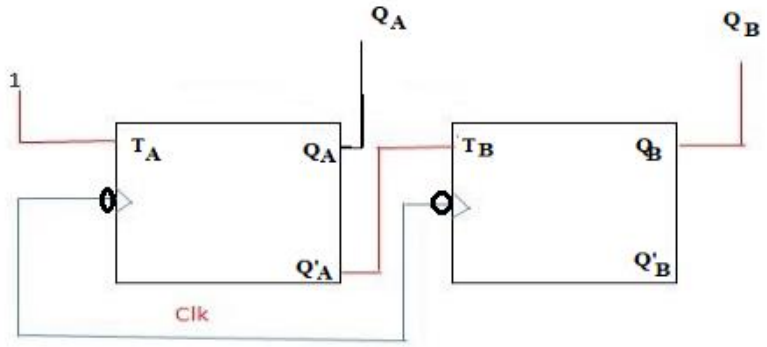
Timing diagram of 2 bit synchronous Down counter.

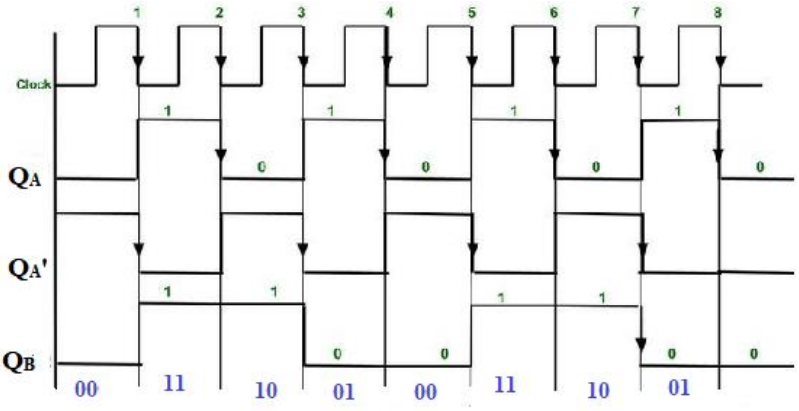
Working Principle of Synchronous 2-Bit down Counter :

Here -ve edge triggered clock is used for toggling purpose.

Previous state(Q_n)	T	Next state(Q_{n+1})
0	0	0
0	1	1
1	0	1
1	1	0

As we see from characteristics table when T = 1, then toggling takes place and T =0 then it stores the output state.





- Initially Q_A= 0, Q_B= 0.
- In simplified equation of K-map we get T_A = 1, therefore Flip flop 1 output Q_A is toggle for every negative edge(because clock is negative edge triggered). Flip-flop(FF) 2, toggle input(T_B) is connected to Q_A'. Therefore, Flip Flop 2 output state Q_B is toggle only when there is clock falling edge (i.e negative edge triggering) and Q_A' =1.
- Therefore, we get output as down counting Q_B(MSB) Q_A(LSB) after 4th negative edge triggered clock the output of the three Flip flops again becomes to initial state Q_B = 0, Q_A =0.
- We get output(state changes) after every negative edge clock pulse.
- With 2 T- Flip flops we get output as 2^2-1= 3 to 0.