

## **UNIT:- 4**

### **Registers & Counters:**

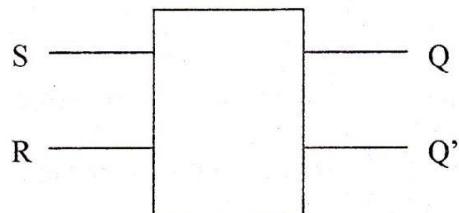
- Flip flops (D, JK, RS, and T).
- Registers (Buffer, Shifting, Control Shift)
- Counters (Up-Down, Ripple)

**Latches :-**

- Latch is a binary cell capable of storing one bit information.
- It is an electronic that has two stable states and therefore can store one bit binary information.
- The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs.
- It is the basic storage element in sequential logic.
- How can we make a circuit out of gates that is not combinatorial? The answer is *feed-back*, which means that we create *loops* in the circuit diagrams so that output values depend, indirectly, on themselves. If such feed-back is *positive* then the circuit tends to have stable states, and if it is *negative* the circuit will tend to oscillate.
- A latch has positive feedback.

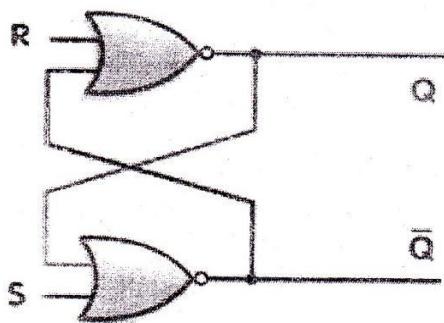
**SR Latch/ RS Latch:-**

- This latch is called *SR-latch*, which stands for *set* and *reset*.
- When using static gates as building block , the most fundamental latch is the simple SR Latch.
- It can be constructed from a pair of cross-coupled NOR logic gates.
- The cross coupled connection from the output of one gate to the input of the other gate constitute a feedback path.
- SR Latch has two input lines and two output lines.
- The output Q and Q' always follow this logic equation  $Q = \text{not } Q$ .
- SR Latch also known as Direct Coupled RS flip-flop or Latch.

**Logic Symbol Of SR Latch:-****Truth Table Of SR Latch:-**

S	R	Q	$Q'$	State
0	0	Q	$Q'$	No Change
0	1	0	1	Set
1	0	1	0	Reset
1	1	X	X	Restricted

Circuit Of SR Latch using NOR gate:-



- $S=1$  and  $r=0$ : the output of  $Q$  is Low (set) and  $s=0$  and  $r=1$  then the output of  $Q$  is high (Reset).
- When  $S$  and  $R$  inputs both are Low, and feedback maintains the  $Q$  and  $Q'$  outputs in a constant state.(no change).
- The  $R = S = 1$  combination is called a restricted combination or a forbidden state because, as both NOR gates then output zeros, it breaks the logical equation  $Q = \text{not } Q$ . The combination is also inappropriate in circuits where *both* inputs may go low *simultaneously* (i.e. a transition from *restricted* to *keep*). The output would lock at either 1 or 0 depending on the propagation time relations between the gates (a race condition).

### D latch:-

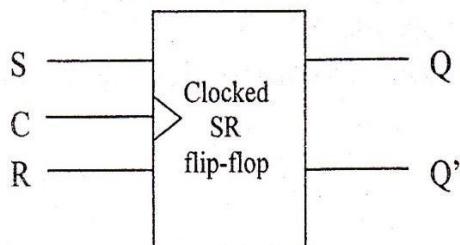
- It is called Data latch or D latch.
- D latch is constructed using the inverted S input as the R input signal.
- The single remaining input designated as "D" to distinguish its operation from other type of latches.

### Flip-Flop:-

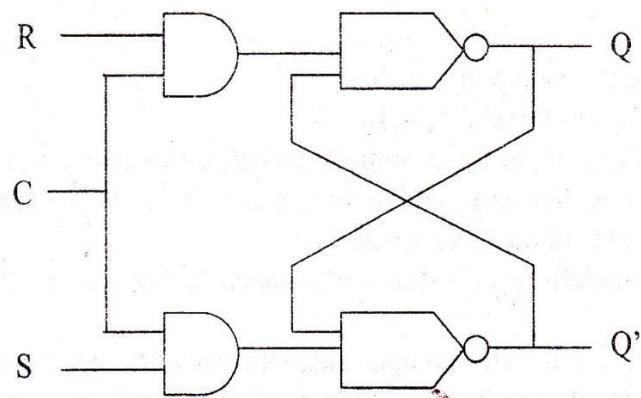
- In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information.
- A flip flop is said to be in state SET 1 when flip flop output is 1. When flip flop output is 0, it is said to be in CLEAR/RESET state.
- A flip-flop is usually controlled by one or two control signals and gate or clock signal.
- The output often stores normal as well as complement values.
- Flip-flop can be applicable in both asynchronous and synchronous( clocked) sequential systems.

**SR flip-flop:-**

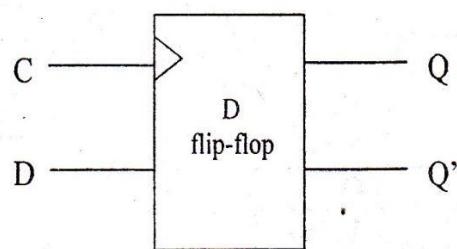
- A *flip-flop* is a synchronous version of the latch.
- It has three inputs S(set), R(reset) and C(clock).
- Like SR latches, SR flip-flops are useful in control applications where we want to be able to set or reset the data bit. However, unlike SR latches, SR flip-flops change their content only at the active edge of the clock signal.
- SR flip-flops can enter an undefined state when both inputs are asserted simultaneously.
- Operation of SR flip-flop:
  - If no clock signal i.e.  $C = 0$  then output can not change irrespective of R and S.
  - When clock signal changes from 0 to 1 and  $S=1$  and  $R=0$  then output  $Q=1$  and  $Q'=0$
  - If  $R=1$   $S=0$  and clock signal changes from 0 to 1 then output  $Q=0$  and  $Q'=1$ (reset)

**Logic diagram of SR filp-flop:-****Truth table of SR flip-flop:-**

C	S	R	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

**Circuit of SR flip-flop:-****D Flip-flop:-**

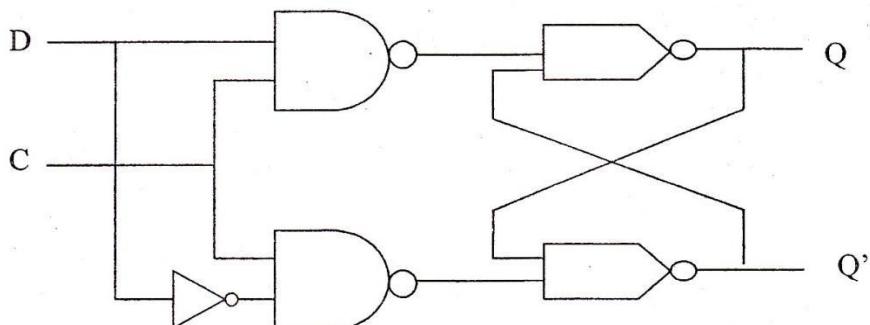
- An RS filp-flop is rarely used in actual sequential logic. It is a modification of RS flip-flop.
- A master-slave D-flip-flop is built from two SR-latches and some other gates.
- However it is fundamental building block and the very useful.
- It has single D data input and a clock signal
- The leftmost SR-latch is called the *master* and the rightmost is called the *slave*.

**Block Diagram of D flip-flop:-****Truth table of D flip-flop:-**

C	D	Q
0	0	0
0	1	1
1	0	0
1	1	1

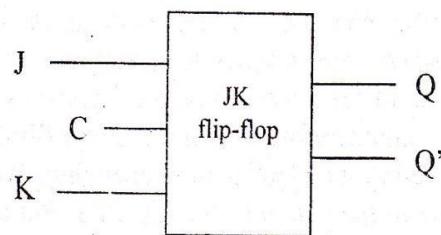
- Let us first consider what happens when the clock signal is 1. In this case, the two *and*-gates in front of the input of the master are *open*, i.e., they let the value of the D input through to the s input of the master, and the inverse of the D input to the r input of the master. Thus, the value of the D input will go straight through the master to the x output of the master. But the two *and*-gates of the slave are *closed*, i.e., their outputs are always 0, so the slave keeps its old value.
- When instead the clock signal is 0, the reverse is true, i.e., the *and*-gates at the input of the master are *closed*, whereas the ones at the input of the slave are *open*. In this case, the flip-flop is completely insensitive to changes in input.
- Now, let us consider what happens when the clock goes from 1 to 0. For this to work, we have to assume that the input remains the same during a brief period from right before to right after the clock signal changes. The first thing that happens is that the *and*-gates at the input of the master *turn off*, i.e., they become insensitive to further changes in input. The value of the x output of the master is now the value of the D input right before the clock started changing. A brief moment later, the clock signal transition has traversed the *inverter* and reaches the *and*-gates of the slave. These gates *open*, allowing the x output of the master to be propagated to the x value of the slave. The x value of the slave, and therefore that of the entire flip-flop now contains the value of the D input right before the clock started changing. We can say that *the clock transition copied the input to the output of the flip-flop*. But at no point in time is there a direct path from input to output. The output changes only as a result of clock transitions from 1 to 0.
- Finally, let us see what happens when the clock goes from 0 to 1. First, the *and*-gates of the master open, letting the value of the D input into the master. By the time the D value reaches the master, the clock signal transition reaches the *and*-gates of the slave, and turns them off before the possibly modified output of the master reaches the slave. Thus, the slave keeps its old value. From the outside, nothing seems to happen, since the output does not change. From now on, however, the master is open to changes in the input.

#### Circuit of D flip-flop:-



**JK Flip-flop:-**

- JK flip-flops are very similar to SR flip-flops. The  $J$  input is just like the  $S$  input in that when asserted, it sets the flip-flop.
- Similarly, the  $K$  input is like the  $R$  input where it clears the flip-flop when asserted.
- The only difference is when both inputs are asserted. For the SR flip-flop, the next state is undefined, whereas, for the JK flip-flop, the next state is the inverse of the current state. In other words, the JK flip-flop toggles its state when both inputs are asserted.
- When  $J$  and  $K$  are 1, the flip-flop output is complimented with clock signal that is if  $Q=1$ , it switches to  $Q_0$  and vice versa.
- Output  $Q$  is ANDed with  $K$  and C inputs so that the flip flop is cleared during a clock pulse only if  $Q$  was previously 1.  $Q'$  is ANDed with  $J$  and C inputs so that the flip flop is set with a clock pulse only if  $Q'$  was previously 1.

**Block diagram of JK flip flop:-****Truth table of JK flip flop:-**

C	J	K	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

## RACE CONDITION

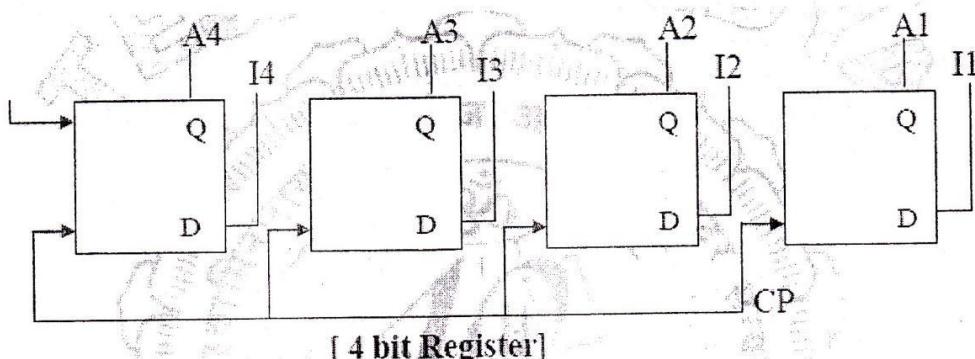
- When 1 is applied to both set and reset I/P in RS flip flop both o/p goes to 0, this stage is undefined and usually avoided.
- If both I/P now goes to zero the state of flip flop is indeterminate and depends on which input remains 1 longer before transition to 0. this is known as race condition.

### Registers

- A *register* is a group of flip-flops capable of storing one bit of information.
- An  $n$ -bit register has a group of  $n$  flip-flops and is capable of storing any binary information of  $n$  bits.
- Various types of register are available in MSI(medium Scale Integrated) circuit.
- The simplest possible register is one that consists of only flip flops without any external gates.
- In addition to flip-flops, registers can have combinational gates that perform certain data-processing tasks. The gates control how and when new information is transferred into the registers.
- The transfer of new information into a register is referred to as a *register load*. If the loading occurs simultaneously at a common clock pulse transition, we say that the load is done in *parallel*.
- The *load input* in a register determines the action to be taken with each clock pulse.
- When the load input is 1, the data from the input lines is transferred into the register's flip-flops. When the load input is 0, the data inputs are *inhibited* and the flip-flop maintains its present state.

### 4 Bit register

- Its basic function is to store the information within a digital system. It is also called buffer register or register.
- Register in which it is possible to give 4 input at a time and to get 4 output is called parallel load register.
- A 4-bit register is shown in the figure below. A clock transition applied to the CP inputs of the register will load all four inputs I1 through I4 in parallel.



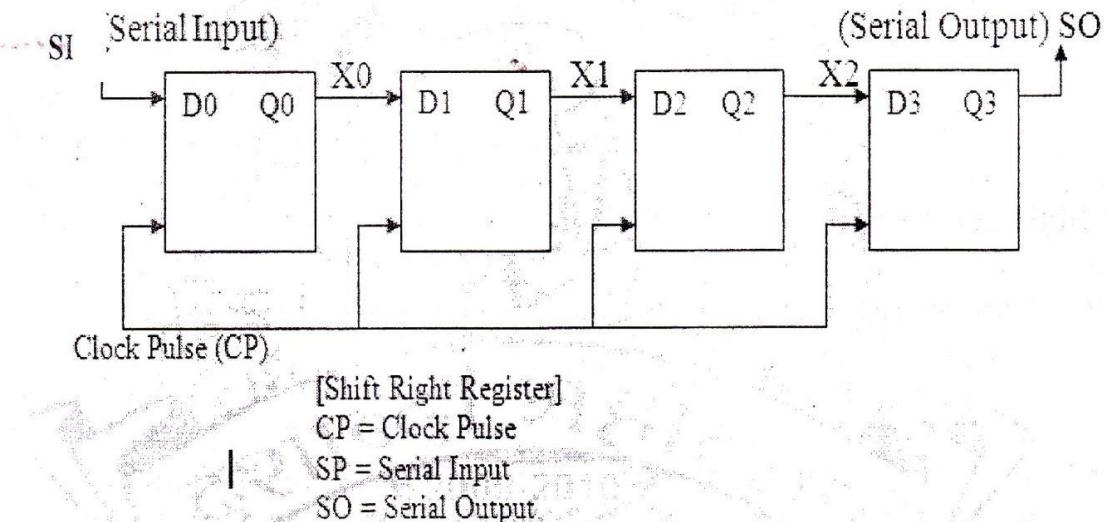
- A register constructed with 4 D – type Flip flop and a common clock pulse input.
- The clock pulse input CP, enables all flip flop so that the information presently available at the four inputs can be transferred into the 4 bit register.
- The four output can be sampled to obtain the information presently stored in the register.
- Sometimes may be possible at a time all inputs not available then this time clock is off, so all inputs are available the clock is on.
- So in register error not possible. That is use register in data storage.
- Now we give all inputs and use clock on position then all inputs are stored. Now clock is off then we change some inputs but clock is off. So this change is not available in stored input.
- Now we store this change in output then use clock on condition.
- Similarly 4 bit register , we can use 8 – bit register, 16 – bit register and so on...

### Shift Registers

- A register capable of shifting its binary information in one either to the right or to the left or in both directions is called a shift register.
- Shift registers are constructed by connecting flip-flops in cascade, where the output of one flip-flop is connected to the input of the next flip-flop.
- All flip-flops receive common clock pulses that initiate the shift from one stage to the next.
- A serial input shift register has a single external input (called the serial input) entering an outermost flip-flop. Each remaining flip-flop uses the output of the previous flip-flop as its input, with the last flip-flop producing the external output (called the serial output).
- A register capable of shifting in one direction is called a unidirectional shift register.
- A register that can shift in both directions is called a bi-directional shift register.
- The most general shift register has the following capabilities:
  - An input for clock pulses to synchronize all operations.
  - A shift-right operation and a serial input line associated with the shift-right.
  - A shift-left operation and a serial input line associated with the shift-left.
  - A parallel load operation and n input lines associated with the parallel transfer.
  - N parallel output lines.
  - A control state that leaves the information in the register unchanged even though clock pulses are applied continuously.
  - A mode control to determine which type of register operation to perform.

### 1. Shift Right Register

- A register capable of shifting its binary information either to the right is called Shift Right register.
- The logical configuration of a shift register consists of a chain of flip flops connected, the output of one flip flop connected to the input of the next flip flop.
- All flip flops receive a common clock pulse which causes the shift from one stage to the next stage.
- The simplest possible shift register is one that uses only flip flops as shown in figure.



- The Q output of a given flip-flop is connected to the D input of the flip flop at its right.
- Each clock pulse shifts the content of the register position to the right. The serial input determines what goes into the leftmost flip flop during the shift.
- The serial output is taken from the output of the right most flip flop prior to the application of a pulse.
- This register shifts its contents to the right, so it's called Shift Right Register.
- Example: we understand in following example, we say one register's output is connected next register input.
  - First time we don't give any input then in register stored garbage value. Now we give input like as 0-1-0-1 then clock in on at this time so register store this value(serial sequence) stored

First Time :	D0	D1	D2	D3	Input
	X0	X1	X2	X3	Garbage Values

Clock on condition and give Input

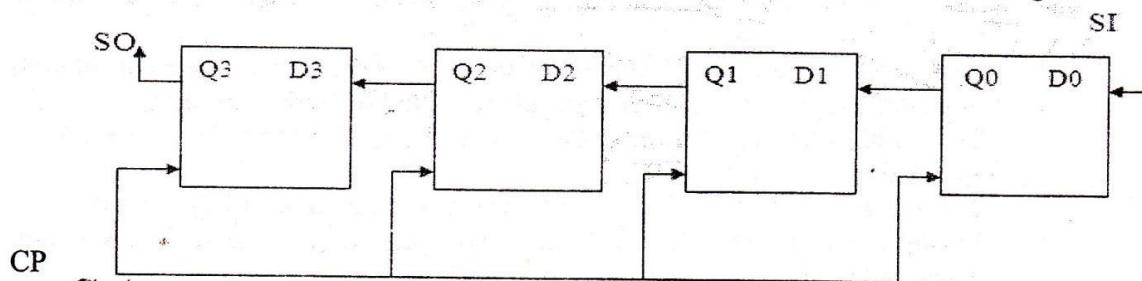
First input	1	X0	X1	X2
Second input	0	1	X0	X1
Third Input	1	0	1	X0
Fourth Input	0	1	0	1

- Above this table, we can say 1<sup>st</sup> input is also store in register and shift this value in next register at that time clock is on condition and similarly so on...

## 2. Shift Left Register

### 3.

- A register capable of shifting its binary information either to the left is called Shift left register.
- The logical configuration of a shift register consists of a chain of flip flops connected, the output of one flip flop connected to the input of the next flip flop.
- All flip flops receive a common clock pulse which causes the shift from one stage to the next stage.
- The simplest possible shift register is one that uses only flip flops as shown in figure.



- The Q output of a given flip-flop is connected to the D input of the flip flop at its left.
- Each clock pulse shifts the content of the register position to the left. The serial input determines what goes into the rightmost flip flop during the shift.
- The serial output is taken from the output of the left most flip flop prior to the application of a pulse.
- This register shifts its contents to the left, so it's called Shift Left Register.
- Example: we understand in following example, we say one register's output is connected next register input.

- First time we don't give any input then in register stored garbage value. Now we give input like as 0-1-0-1 then clock in on at this time so register store this value(serial sequence) stored.

First Time :	D0	D1	D2	D3	Input
	X3	X2	X1	X0	Garbage Values

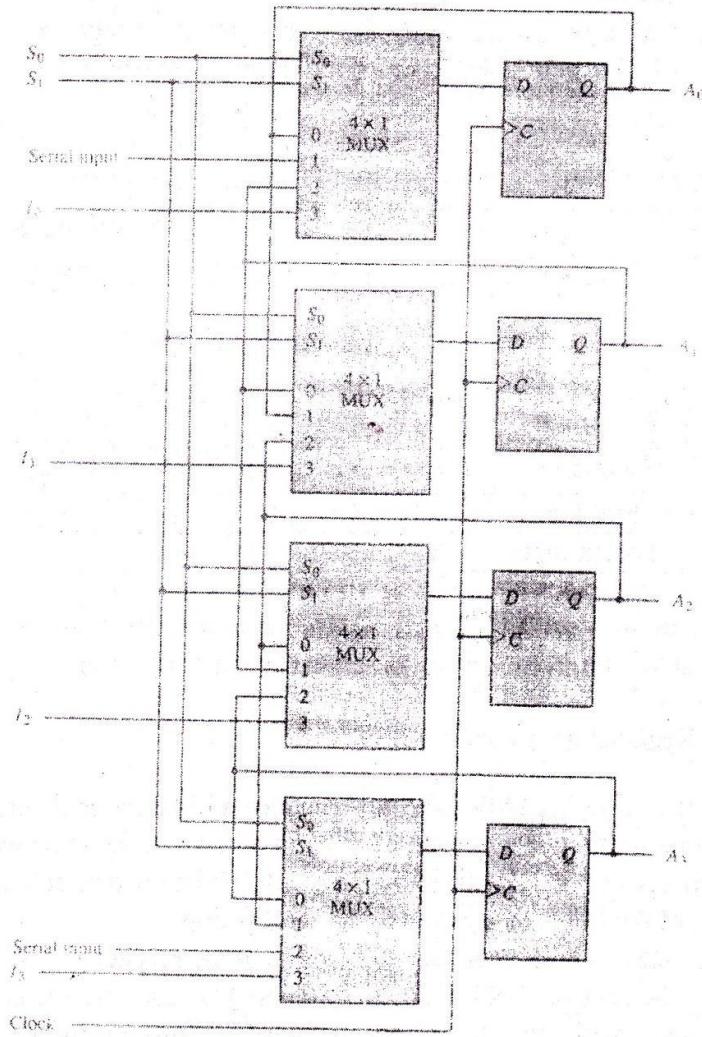
Clock on condition and give Input

First input	X2	X1	X0	0
Second input	X1	X0	0	1
Third Input	X0	0	1	0
Fourth Input	0	1	0	1

- Above this table, we can say 1<sup>st</sup> input is also store in register and shift this value in next register at that time clock is on condition and similarly so on...

#### Bidirectional Shift Register with parallel load

- A register that can shift in both directions is called a bi-directional shift register. A 4-bit bidirectional shift register with parallel load is shown in figure below.
- Each stage consists of a D flip-flop and a 4X1 MUX. The two selection inputs S1 and S0 select one of the MUX data inputs for the D flip-flop.
- The selection lines control the mode of operation of the register.
- When the mode control S1S0 = 00, data input 0 of each MUX is selected. This condition forms a path from the output of each flip-flop into the input of the same flip-flop.
- The next clock transition refers into each flip-flop the binary value it held previously, and no change of state occurs. When S1S0 = 01, the terminal marked 1 in each MUX has a path to the D input of the corresponding flip-flop.
- This causes a shift-right operation, with the serial input data transferred into flip-flop A0 and the content of each flip-flop Ai-1 transferred into flip-flop Ai for i=1,2,3.
- When S1S0 = 10 a shift-left operations results, with the other serial input data going into flip-flop A3 and the content of flip-flop Ai+1 transferred into flip-flop Ai for I=0,1,2.
- When S1S0 = 11, the binary information from each input I0 through I3 is transferred into the corresponding flip-flop, resulting in a parallel load operation.
- In the diagram, the shift-right operation shifts the contents of the register in the down direction while the shift left operation causes the contents of the register to shift in the upward direction.

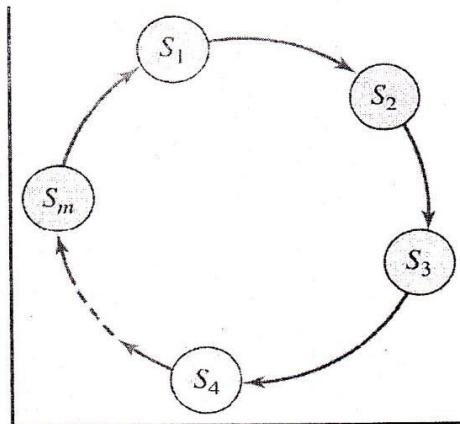


### Application of Shift Registers

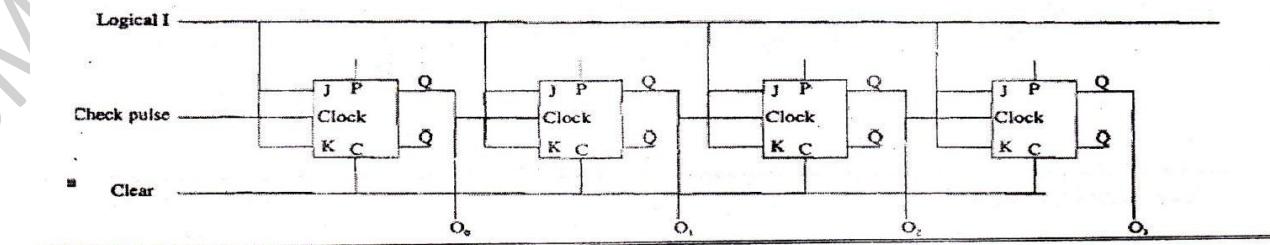
Shift registers are often used to interface digital systems situated remotely from each other. For example, suppose that it is necessary to transmit an  $n$ -bit quantity between two points. If the distance between the source and the destination is too far, it will be expensive to use  $n$  lines to transmit the  $n$  bits in parallel. It may be more economical to use a single line and transmit the information serially one bit at a time. The transmitter loads the  $n$ -bit data in parallel into a shift register and then transmits the data from the serial output line. The receiver accepts the data serially into a shift register through its serial input line. When the entire  $n$  bits are accumulated they can be taken from the outputs of the register in parallel. Thus the transmitter performs a parallel-to-serial conversion of data and the receiver converts the incoming serial data back to parallel data transfer.

**Counters:-**

- A counter is a mechanism , which calculate a step ahead and step behind current value
- A Counter is a mechanism , that calculate any value one stepwise backward or forward.
- A counter is a register, which goes through a predetermined sequence of states when clock pulse is applied. In principle, the value of counters is incremented by 1 module the capacity of register i.e. when the value stored in a counter reaches its maximum value, the next incremented value becomes zero. The counters are mainly used in circuits of digital systems where sequence and control operations are performed, for example, in CPU we have program counter (PC).
- The counting sequence is often depicted by a graph called a state diagram. A modulus-m counter (i.e., a counter with m states) has the following state diagram:
- Each node  $S_i$  denotes the states of the counter and the arrows in the graph denote the order in which the states occur.
- Counters can be classified into two categories, based on the way they operate: Asynchronous and synchronous counters. In Asynchronous counters, the change in state of one flip-flop triggers the other flip-flops. Synchronous counters are relatively faster because the state of all flip-flops can be changed at the same time.

**ASYNCHRONOUS (RIPPLE) COUNTERS**

- This is more often referred to as ripple counter, as the change, which occurs in order to increment the counter ripples through it from one end to the other.
- An implementation of 4-bit ripple counter using J-K flip-flops. This counter is incremented on the occurrence of each clock pulse and counts from 0000 to 1111 (i.e. 0 to 15).
- This circuit is a 4-bit binary ripple counter. All the JK flip-flops are configured to toggle their state on a downward transition of their clock input, and the output of each flip-flop is fed into the next flip-flop's clock. So, when each bit changes from 1 to 0, it "carries the one" to the next higher bit.
- The input line to J & K of all flip-flops is kept high i.e. logic1. Each time a clock pulse occurs the value of flip-flop is complemented (Refer to characteristic table of J K flip-flop in Figure. Please note that the clock pulse is given only to first flip-flop



and second flip-flop onwards, the output of previous flip-flop is fed as clock signal. This implies that these flip-flops will be complemented if the previous flip-flop has a value 1. Thus, the effect of complement will ripple through these flip-flops.

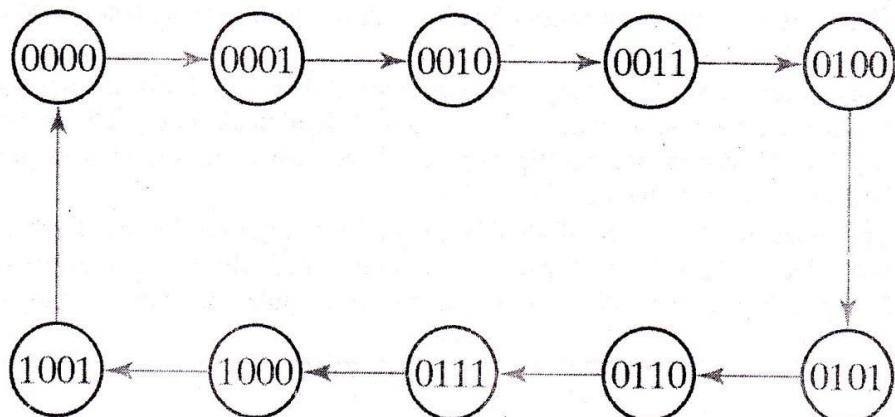
[ 4 bit ripple counter]

➤ Count Sequence for a binary ripple counter

Count Sequence				Conditions for complementing flip flop
A4	A3	A2	A1	
0	0	0	0	Complement A1
0	0	0	1	Complement A1 - A1 will go from 1 to 0 and compl. A2
0	0	1	0	Complement A1
0	0	1	1	Complement A1 - A1 will go from 1 to 0 and compl. A2; A2 will go from 1 to 0 and Compl. A3
0	1	0	0	Complement A1
0	1	0	1	Complement A1 - A1 will go from 1 to 0 and compl. A2
0	1	1	0	Complement A1
0	1	1	1	Complement A1 - A1 will go from 1 to 0 and compl. A2; A2 will go from 1 to 0 and Compl. A3; A3 go from 1 to 0 and Compl. A4
1	0	0	0	And so on...

### BCD RIPPLE COUNTER

- A decimal counter follows a sequence of ten states and returns to 0 after the count of 9. Such a counter must have at least four flip flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits.
- The sequence of states in a decimal counter is dictated by the binary code used to state diagram.
- This is similar to a binary counter, except that the state after 1001 ( code for decimal digit 9) is 0000 (code for decimal digit 0)
- 



[State diagram of a Decimal BCD Counter]

- The four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD code.
- The flip flops trigger on the negative edge, i.e. when the CP signal goes from 1 to 0. Note that the output of  $Q_1$  is applied to the CP inputs of both  $Q_2$  and  $Q_8$ , the output of  $Q_2$  is applied to the CP input of  $Q_4$ .
- A Ripple counter is an asynchronous sequential circuit. Signals that affect the flip flop transition depends on order in which they change from 1 to 0. The operation of the counter can be explained by a list of conditions for flip flop transitions.
- When the CP input goes from 1 to 0, the flip flop is set  $J = 1$ , is cleared if  $K = 1$ , is complemented if  $J = K = 1$ , and is left unchanged if  $J = K = 0$ .
- The following are the conditions for each flip flop state diagram:
  - $Q_1$  is complemented on the negative edge of every count pulse
  - $Q_2$  is complemented if  $Q_8 = 0$  and  $Q_1$  goes from 1 to 0.  $Q_2$  is cleared if  $Q_8 = 1$  and  $Q_1$  goes from 1 to 0
  - $Q_4$  is complemented when  $Q_2$  goes from 1 to 0
  - $Q_8$  is complemented when  $Q_4Q_2$  and  $Q_1$  go from 1 to 0.  $Q_8$  is cleared if either  $Q_4$  or  $Q_2$  is clear and  $Q_1$  goes from 1 to 0.

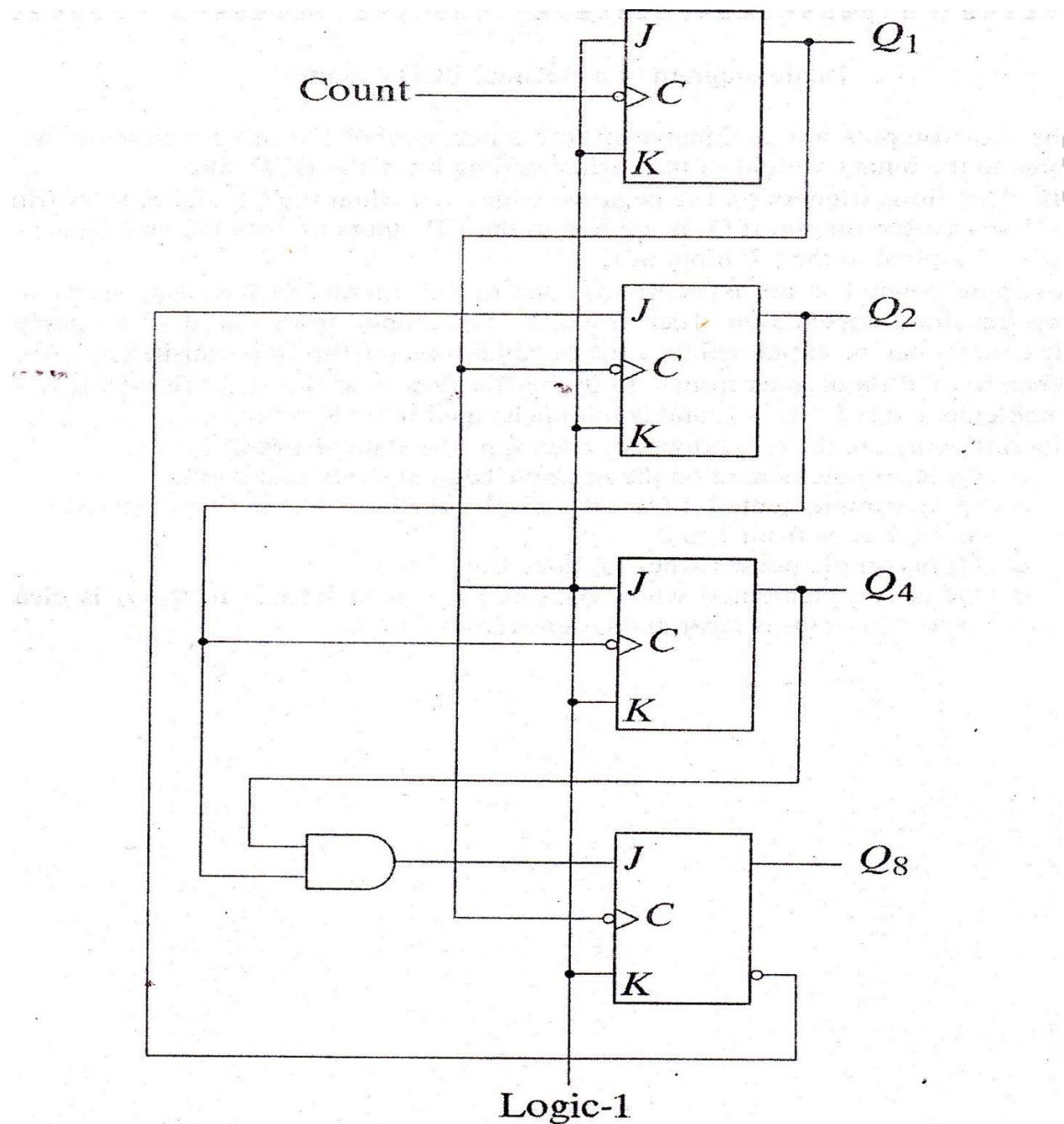
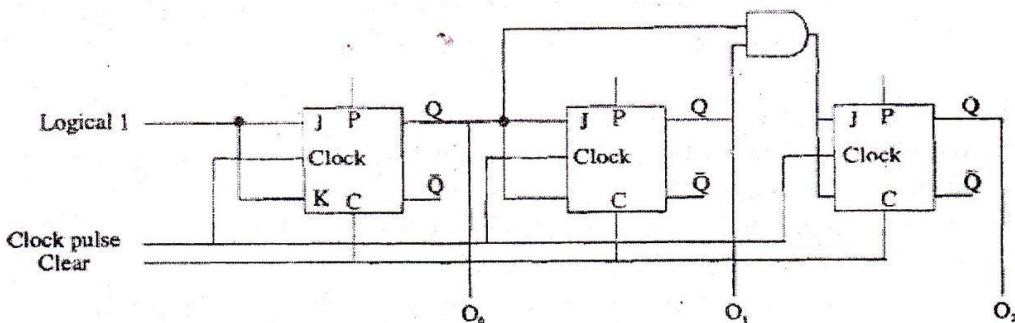


Fig. 6-10 BCD Ripple Counter

**SYNCHRONOUS COUNTER**

- It Take the instance when the state of ripple counter is 01 11 now the next state will be 1000 that means change in the state of all the flip-flops, but will it occur simultaneously in ripple counter? No first left most flip flop will change state from 1 to 0, this will cause the next flip-flop to change state and so on till the last flip-flop changes the state. Thus, a delay is there in changing the state is proportional to the length of the counter. Therefore, to avoid this delay normally synchronous counters are used, in which all the flip-flops change state at the same time.



[Logic diagram of three bit synchronous counter]

- In a synchronous counter:
  - The first flip-flop is always complemented.
  - The second flip-flop is complemented in the next clock pulse if the current state of the first flip flop is set (one).
  - The third flip-flop is fed by an AND gate which is connected with the output of the first and second flip-flops thus, the third flip-flop will be complemented only if the first AND second flip-flops are currently in 1 state. This will be more evident from the following truth table:

$O_0$	$O_1$	$O_2$
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1
1	1	1
0	1	1
1	1	1
0	0	0

Smt K S Kapshi Bca College,Palitana

D.C.O

Unit-4

Msc.IT SEM-1

SMT KS KAPASHI BCA COLLEGE PALITANA