Unit – V

Objectives:

- To familiarize with the concepts of different sequential circuits.

Syllabus: Sequential Circuits-I

Classification of sequential circuits (synchronous and asynchronous); basic flip-flops, truth tables and excitation tables (nand RS latch, nor RS latch, RS flip-flop, JK flip-flop, T flip-flop, D flip-flop with reset and clear terminals). Conversion from one flip-flop to flip-flop. Design of ripple counters, design of synchronous counters, Johnson counter, ring counter. Design of registers - Buffer register, control buffer register, shift register, bi-directional shift register, universal shift register.

Outcomes:

Students will be able to

- understand the functionality of different latches and flip-flops..
- distinguish the working of latch and flip-flop.
- convert from one flip-flop to another flip-flop
- classify various types of registers.
- design synchronous and asynchronous counters.

Learning Material

- Combinational circuits are those whose output at any instant of time is entirely dependent on the input present at that time.

- Sequential circuits are those in which output at any given time is not only dependent on the input, present at that time but also on previous outputs. Naturally, such circuits must record the previous outputs. This gives rise to memory.

- Often, there are requirements of digital circuits whose output remain unchanged, once set, even if the inputs are removed. Such devices are referred as –memory elements‖, each of which can hold 1-bit of information. These binary bits can be retained in the memory indefinitely (as long as power is delivered) or until new information is feeded to the circuit.

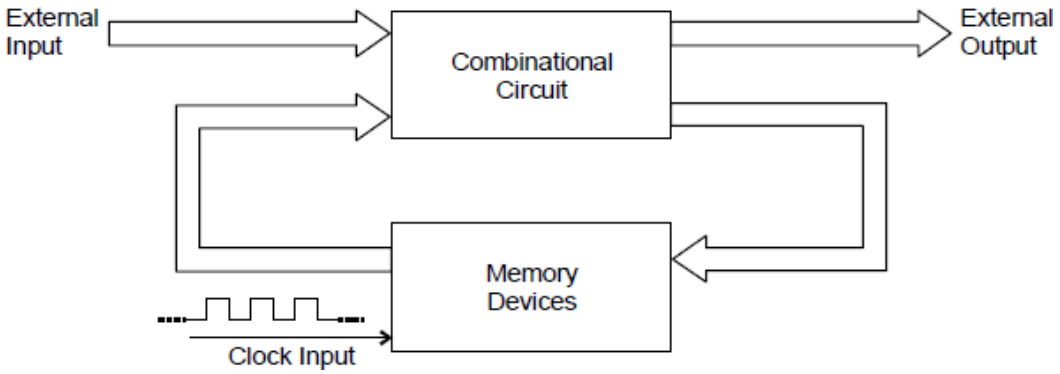


Fig 1: Block diagram of a sequential circuit

A block diagram of a sequential circuit is shown in above Fig.1 A **Sequential circuit** can be regarded as a collection of memory elements and combinational circuit, as shown in above Fig.1. A feedback path is formed by using memory elements, input to which is the output of combinational circuit. The binary information stored in memory element at any given time is defined as the **state** of sequential circuit at that time. Present contents of memory elements are referred as the **present state.** The combinational circuit receives the signals from external input and from the memory output and determines the external output. They also determine the condition and binary values to change the state of memory. The new contents of the memory elements are referred as **next state** and depend upon the external input and present state. Hence, a sequential circuit can be completely specified by a time sequence of inputs, outputs and the internal states. In general, clock is used to control the operation. The clock frequency determines the speed of operation of a sequential circuit.

Classification of sequential circuits

There exist two main categories of sequential circuits, namely synchronous and asynchronous sequential circuits.

- **Asynchronous Sequential Circuits:**
  - Sequential circuits whose behavior depends upon the sequence in which the inputs are applied, are called **Asynchronous Sequential Circuits**.
  - In these circuits, outputs are affected whenever a change in inputs is detected. Memory elements used in asynchronous circuits mostly are time delay devices. The memory capability of time delay devices is due to the propagation delay of the devices. Propagation delay produced by the logic gates is sufficient for this purpose.
  - Hence –An Synchronous sequential circuit can be regarded as a combinational circuit with feedback‖. However feedback among logic gates makes the asynchronous sequential circuits, often susceptible to instability. As a result they may become unstable. This makes the design of asynchronous circuits very tedious and difficult.
- **Synchronous Sequential Circuit:**
  - **Synchronous Sequential Circuit** may be defined as a sequential circuit, whose state can be affected only at the discrete instants of time. The synchronization is achieved by using a timing device, termed as **System Clock Generator**, which generates a periodic train of clock pulses. The clock pulses are feed to entire system in such a way that internal states (i.e. memory contents) are affected only when the clock pulses hit the circuit.

Storage Elements: Latches

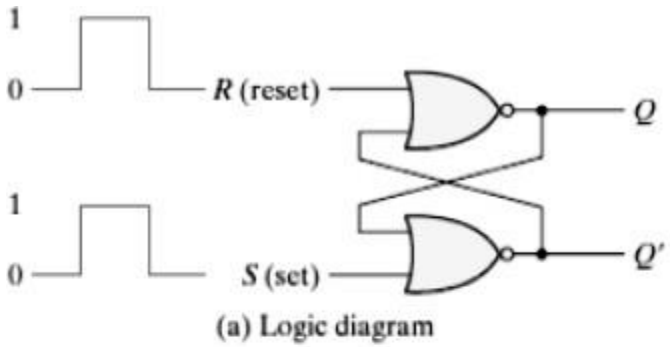
- A storage element in a digital circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit), until directed by an input signal to switch states.
- The major differences among various types of storage elements are in the number of inputs they possess and in the manner in which the inputs affect the binary state.
- 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 are said 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. Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use as storage elements in synchronous sequential circuits.

SR Latch

The *SR* latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates, and two inputs labeled *S* for set and *R* for reset. Both the versions are shown in Fig 2(a) & Fig 2(b). The latch has two useful states. When output $Q = 1$ and $Q' = 0$, the latch is said to be in the *set state*. When $Q = 0$ and $Q' = 1$, it is in the *reset state.* **SR**

SR Latch with Control Input

The operation of the basic SR latch can be modified by providing an additional input signal that determines (controls) when the state of the latch can be changed by determining whether S and R (or S‘ and R‘) can affect the circuit. An SR latch with a control input is shown in Fig 2(c). It consists of the basic SR latch and two additional NAND gates. The control input En acts as an enable signal for the other two inputs. **The outputs of the NAND gates stay at the logic-1 level as long as the enable signal remains at 0.** This is the quiescent condition for the SR latch. When the enable input goes to 1, information from the S or R input is allowed to affect the latch.

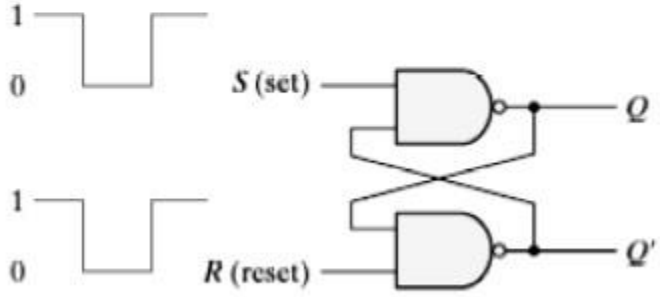


(a) Logic diagram

S	R	Q	Q′	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

Fig 2.a: SR Latch with NOR gates

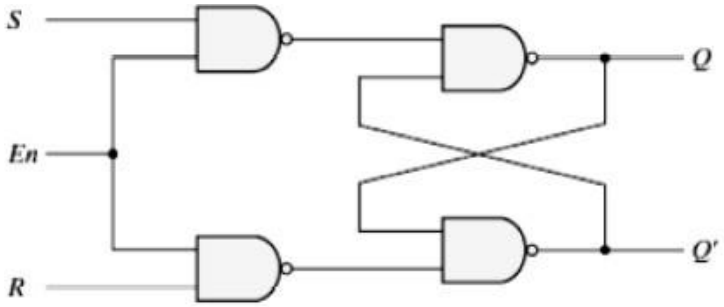


(a) Logic diagram

S	R	Q	Q′	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

Fig 2.b: SR Latch with NAND gates



(a) Logic diagram

En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	*Q* = 0; reset state
1	1	0	*Q* = 1; set state
1	1	1	Indeterminate

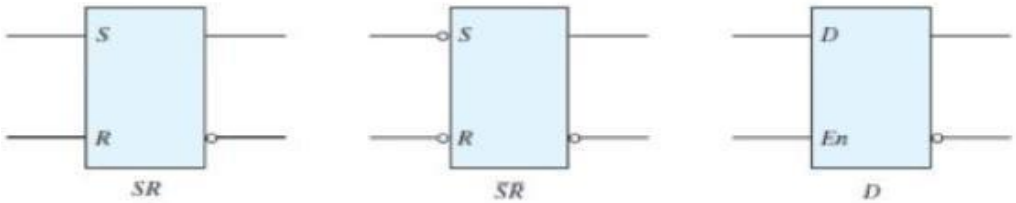
(b) Function table

Fig 2.c: SR Latch with Control Input

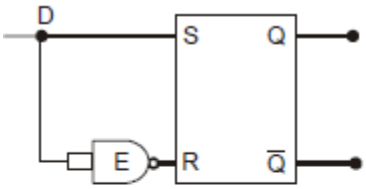
The set state is reached with $S = 1$, $R = 0$, and $En = 1$ (active-high enabled). To change to the reset state, the inputs must be $S = 0$, $R = 1$, and $En = 1$. In either case, when *En* returns to 0, the circuit remains in its current state. The control input disables the circuit by applying 0 to *En*, so that the state of the output does not change regardless of the values of *S* and *R*. Moreover, when $En = 1$ and both the *S* and *R* inputs are equal to 0, the state of the circuit does not change. These conditions are listed in the function table accompanying the diagram. An indeterminate condition occurs when all three inputs are equal to 1. This condition places 0's on both inputs of the basic *SR* latch, which puts it in the undefined state.

D Latch (Transparent latch)

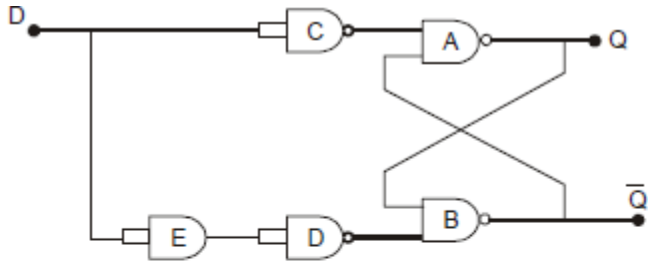
One way to eliminate the undesirable condition of the indeterminate state in the *SR* latch is to ensure that inputs *S* and *R* are never equal to 1 at the same time. This is done in the *D* latch, shown in Fig. below. This latch has only two inputs: *D* (data) and *En* (enable). The *D* input goes directly to the *S* input, and its complement is applied to the *R* input. As long as the enable input is at 0, the cross-coupled *SR* latch has both inputs at the 1 level and the circuit cannot change state regardless of the value of *D*. The *D* input is sampled when $En = 1$. If $D = 1$, the *Q* output goes to 1, placing the circuit in the set state. If $D = 0$, output *Q* goes to 0, placing the circuit in the reset state.



Graphic Symbols for Latches



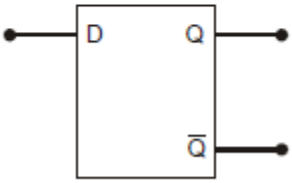
(a) Modification in S R Flip-Flop



(b) Construction

D	Q	Resulting State
0	0	Reset State or Low State
1	1	Set State or High State

(c) Truth Table



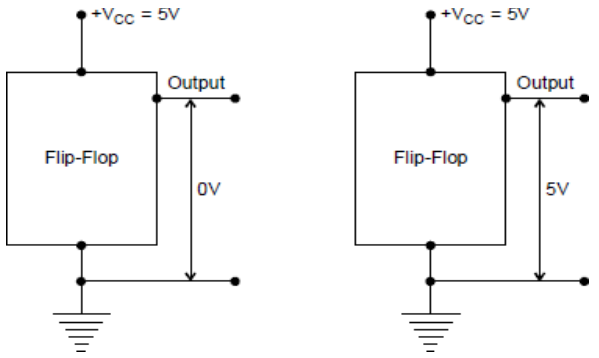
(d) Logic Symbol

STORAGE ELEMENTS: FLIPFLOPS

A synchronous sequential circuit that uses clock at the input of memory elements are referred as **Clocked Sequential circuit**. The clocked sequential circuits use a memory element known as **Flip-Flop**- A flip-flop is an electronic circuit used to store 1-bit of information, and thus forms a 1-bit memory cell. These circuits have two outputs, one giving the value of binary bit stored in it and the other gives the complemented value. The real differences among various flip-flops are the number of inputs and the manner in which binary information can be entered into it. In the next section we examine the most general flip-flops used in digital systems.

The flip-flops are 1-bit memory cells that can maintain the stored bit for desired period of time. A **Bi-stable** device is one in which two well defined states exist and at any time the device could assume either of the stable states. A **stable state** is a state, once reached by a device does not changes until and unless something is done to change it. A toggle switch has two stable states, and can be regarded as a bi-stable device. When it is closed, it remains closed (A stable state) until some one opens it. When it is open, it remains open (2nd stable state) until some one closes it i.e. makes it to return to its first stable state. So it is evident that the switch may be viewed as 1-bit memory cell, since it maintains its state (either open or close). Infact any bistable device may be referred as 1-bit memory cell.

A **Flip-Flop** may also be defined as a bistable electronics device whose two stable states are 0V and + 5V corresponding to Logic 0 and Logic 1 respectively. The two stable states and flip-flop as a memory element is illustrated in below Fig 3(a). shows that the flip-flop is in _State 0‘ as output is 0V. This can be regarded as storing Logic 0. Similarly flip-flop is said to be in _State 1‘, see Fig 3(b), when the output is 5 V. This can be regarded as storing logic 1. Since at any given time flip-flop is in either of two states the flip-flop may also be regarded as Bistable Multivibrator. Since the state once reached is maintained until it is deliberately changed, the flip- flop is viewed as memory element.



(a) State 0 or Low State (b) State 1 or High State

Fig 3: Flip-flop as bistable device

The basic memory circuit or flip-flop can be easily obtained by connecting two inverters (Not gates) in series and then connecting the output of second inverter to the input of first inverter through a feedback path, as shown in below Fig. 4(a).

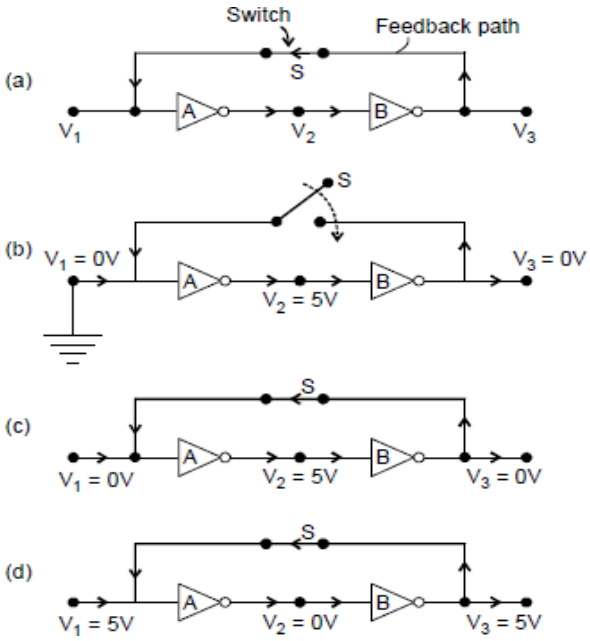


Fig 4: Basic flip-flop or latch _logic 0‘ = 0V, and _logic 1‘ = 5 V

It is evident from Fig that V1 and V3 will always be same, due to very nature of inverters.

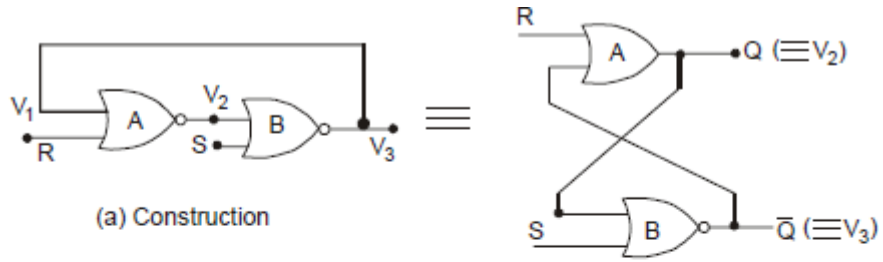
Let us define Logic 0 = 0V and Logic 1 = 5 V. Now open the switch _S‘ to remove the feedback and connect V1 to ground, as shown in Fig. 4(b). Thus input to inverter A is Logic 0 and its output would be Logic 1 which is given to the input of inverter B. Since input to inverter B is Logic 1, its output would be Logic 0. Hence input of inverter A and output of inverter B are same. Now if we close the switch S feedback path is reconnected, then ground can be removed from V1 and V3 can still be at 0V i.e. Logic 0. This is shown in Fig. 4(c). This is possible because once the V1 is given 0V (i.e. Logic 0) the V3 will also be at 0V and then it can be used to hold the input to inverter A at 0V, through the feedback path. This is first stable state. In the simpler way if we connect the V1 to 5 V and repeat the whole process, we reach to second stable state because V3 = 5V. Essentially the V3 holds the input to inverter. A (i.e. V1), allowing + 5V supply to be removed, as shown in Fig. 4(*d*). Thus V3 = 5 V can be maintained untill desired period time. A simple observation of the flip-flop shown in Fig.4(*a*) reveals that V2 and V3 are always complementary, i.e. $V_2 = \overline{V_3}$ or $V_3 = \overline{V_2}$. This does mean that ‒at any point of time, irrespective of the value of V1, both the stable states are available‖, see Fig.4(*c*) (*d*). This is fundamental condition to Flip-Flop. Since the information present at the input is locked or latched in the circuit, it is also referred or **Latch**. When the output is in low state (i.e. V3 = 0 V), it is frequently referred as **Reset State**. Whereas when the output is in high state (i.e. V3 = 5 V), it is conveniently called as **Set State**. Fig. 4(*c*) and (*d*) shows the reset and set states, respectively.

RS Flip-Flop

Although the basic latch shown by the Fig. 5(*a*) was successful to memorize (or store) 1-bit information, it does not provide any convenient mean to enter the required binary bit. Thus to provide a way to enter the data circuit of Fig. 5(*a*) can be modified by replacing the two inverters by two 2-input NOR gate or NAND gates, discussed in following articles.

The NOR LATCH: The NOR latch is shown by Fig. 5(*a*) and 5(*b*). Notice that if we connect the inputs, labeled as R and S, to logic 0 the circuit will be same as the circuit shown in Fig. 5(*a*)

and thus behave exactly same as the NOT gate latch of Fig. 5(*a*). The voltage V2 and V3 are now labeled as Q and Q and are declared as output. Regardless of the value of Q, its complement is Q (as V3 = $\overline{V2}$). The two inputs to this flip-flop are R and S, stand for RESET and SET inputs respectively. A _1‘ on input R switches the flip-flop the inputs are low the flip-flop maintain its last state. Thats why the truth table has entry Q in first row in reset state i.e. Q = _0‘ and $\bar{Q}$= _1‘. A _1‘ on inputs (SET input) will bring the latch into set state i.e. Q = _1‘ and $\bar{Q}$ = _0‘. Due to this action it is often called **set-reset latch**. The operation and behavior is summarized in the truth table shown by Fig. 5(*c*). Fig.5(*d*) displays the logic symbol of RS (or SR) flip-flop. To understand the operation of this flip-flop, recall that a _1‘ at any input of a NOR gate forces its output to _0‘ where as _0‘ at an input does not affect the output of NOR gate. When inputs are S = R = 0, first row of truth tables it does not affect the output. As a result the Latch maintains its state. For example if before application of the inputs S = R= 0, the output was Q = 1, then it remains 1 after S = R = 0 are applied. Thus, when both the inputs are low the flip-flop maintain its last state. Thats why the truth table has entry Q in first row.

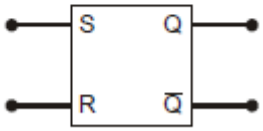


(a) Construction

(b) Construction

S	R	Q	Resulting State
0	0	Q	Last State or No Change at output
0	1	0	Reset State
1	0	1	Set State
1	1	?	Indeterminate or Forbidden State

(c) Truth Table



(d) Logic Symbol

Fig 5: Basic NOR gate latch or RS (or SR) flip-flop

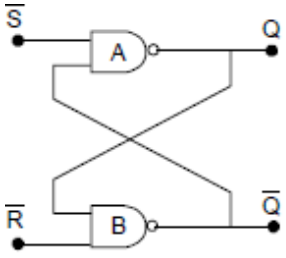
Now, if S = 0 and R = 1, output of gate-A goes low i.e. Q = 0. The Q is connected to input of gate-B along with S input. Thus with Q = 0 both the inputs to NOR gate B are LOW. As a result $\bar{Q}$= 1. This Q and $\bar{Q}$ are complementary. Since Q = 0 the flip-flop is said to be in ‖reset state‖.

This is indicated by the second row of the truth table. Now if S = 1 and R = 0 output of gate-B is LOW, making both the inputs of gate-A LOW, consequently Q = 1. This is –set state‖. As Q = 1 and $\bar{Q}$=0 the two outputs are complementary. This is shown in third row of truth table. When S = 1 and R = 1, output of both the gates are forced to Logic 0. This conflicts with the definition that both Q and $\bar{Q}$ must be complementary. Hence this condition must not be applied to SR flip-flop. But if due to some reasons S = 1 and R = 1 is applied, then it is not possible to predict the output and flip-flop state is said to be indeterminate. This is shown by the last row of truth table.

It is worth to devote some time to investigate why S = R = 1 results indeterminate state while we said earlier that output of both the gates go LOW for this input. This is true due to the logic function of NOR gate that if any of the input is HIGH output is LOW. In the circuit of above Fig (b) both Q and $\bar{Q}$ are LOW as long as S and R are High. The problem occurs when inputs S and R go to LOW from High. The two gates cannot have exactly same propagation delay. Now the gate having smaller delay will change its state to HIGH earlier than the other gate. And since this output (i.e. Logic 1) is feeded to the second gate, the output of second gate is forced to stay at Logic 0. Thus depending upon the propagation delays of two gates, the flip-flop attains either of the stable states (i.e. either Q = 1 or Q = 0). Therefore it is not possible to predict the state of flip-flop after the inputs S = R = 1 are applied That‘s why the fourth row of truth table contains a question mark (?). For the above reasons the input condition S = R = 1 is forbidden.

The NAND Gate Flip-Flop

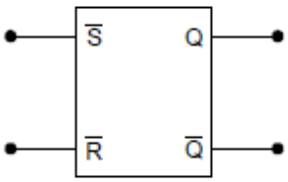
The NOR gate latch shown by Fig. 5(*b*) may also be modified by replacing each inverter by a 2-input NAND gate as shown in below Fig.6(*a*). This is a slightly different latch



(a) Construction

$\bar{S}$	$\bar{R}$	Out-put 'Q'	Resulting State
1	1	Q	Last State or No Change at output
1	0	0	Reset State
0	1	1	Set State
0	0	?	Indeterminate or Forbidden State

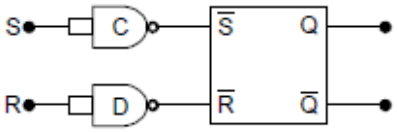
(b) Truth Table



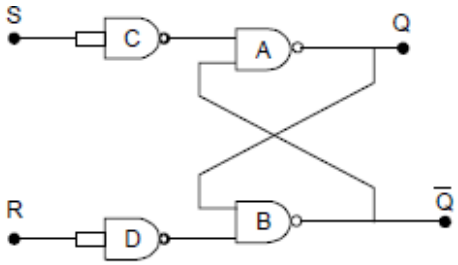
(c) Logic Symbol

Fig 6: NAND gate latch or $\bar{S}, \bar{R}$ flipflop

The SR flip-flop can be modified further by using two additional NAND gates.



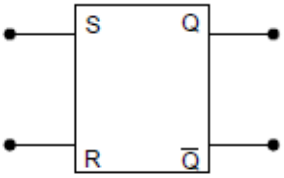
(a) Modification in $\overline{S}\,\overline{R}$ Flip-Flop



(b) Construction

S	R	Out-put Q	Resulting State
0	0	Q	Last State preserved or No Change at output
0	1	0	Reset State or Low State
1	0	1	Set State or High State
1	1	?	Indeterminate or Forbidden State

(c) Truth Table

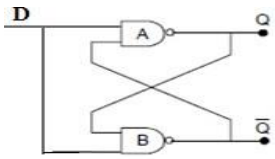


(d) Logic Symbol

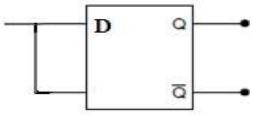
Fig 7: NAND gate latch or SR flip-flop

D-Flip-flop

The SR latch, we discussed earlier, has two inputs S and R. At any time to store a bit, we must activate both the inputs simultaneously. This may be troubling in some applications. Use of only one data line is convenient in such applications. Moreover the forbidden input combination S = R = 1 may occur unintentionally, thus leading the flip-flop to indeterminate state. In order to deal such issues, SR flip-flop is further modified as shown in Fig 8. The resultant latch is referred as D flip-flop or D latch. The latch has only one input labelled D (called as Data input). An external NAND gate (connected as inverter) is used to ensure that S and R inputs are always complement to each other. Thus to store information in this latch, only one signal has to be generated.



(a) Construction



(b) Logic Symbol

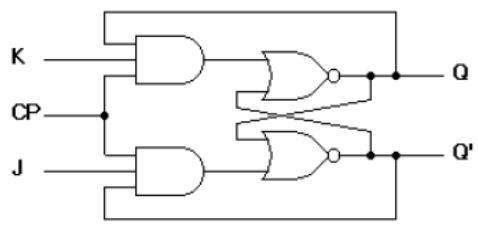
Fig 8: D flip-flop or D latch

Operation of this flip-flop is straight forward. At any instant of time the output Q is same as D (i.e. Q = D). Since output is exactly same as the input, the latch may be viewed as a delay unit. The flip-flop always takes some time to produce output, after the input is applied. This is called propagation delay. Thus it is said that the information present at point D (i.e. at input) will take a time equal to the propagation delay to reach to Q. Hence the information is delayed. For this reason it is often called as **Delay (D) Flip-Flop**.

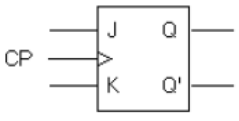
JK FLIPFLOP

A JK flip-flop is a refinement of the SR flip-flop in that the indeterminate state of the SR type is defined in the JK type. Inputs J and K behave like inputs S and R to set and clear the flip-flop (note that in a JK flip-flop, the letter J is for set and the letter K is for clear). When logic 1 inputs are applied to both J and K simultaneously, the flip-flop switches to its complement state, ie., if Q=1, it switches to Q=0 and vice versa.

A clocked JK flip-flop is shown. Output Q is ANDed with K and CP inputs so that the flip-flop is cleared during a clock pulse only if Q was previously 1. Similarly, ouput Q' is ANDed with J and CP inputs so that the flip-flop is set with a clock pulse only if Q' was previously 1. Note that because of the feedback connection in the JK flip-flop, a CP signal which remains a 1 (while J=K=1) after the outputs have been complemented once will cause repeated and continuous transitions of the outputs. To avoid this, the clock pulses must have a time duration less than the propagation delay through the flip-flop. The restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction. The same reasoning also applies to the T flip-flop presented next.



(a) Logic diagram



(b) Graphical symbol

Q J K	Q(t+1)
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	0

(c) Transition table

Fig 9: Clocked JK Flip Flop

T Flip-Flop

The T flip-flop is a single input version of the JK flip-flop. As shown, the T flip-flop is obtained from the JK type if both inputs are tied together. The output of the T flip-flop "toggles" with each clock pulse.

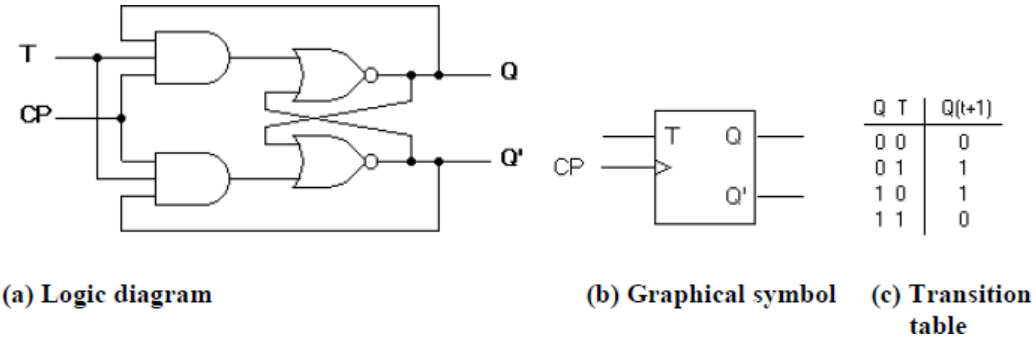


Fig 10: Clocked T Flip Flop

Clocked Flip-Flops

All the flip-flops discussed earlier are said to be **transparent**, because any chance in input is immediately accepted and the output changes accordingly. Since they consist of logic gates along with feedback they are also regarded as **asynchronous flip-flops.** However, often there are requirements to change the state of flip-flop in synchronism with a train of pulses, called as **Clock**. In fact we need a control signal through which a flipflop can be instructed to respond to input or not. Use of clock can serve this purpose. A clock signal can be defined as a train of pulses. Essentially each pulse must have two states, ON state and OFF state. Fig 11 shows two alternate representation of a pulse, and shows a clock signal. The clock pulses are characterized by the **duty cycle**, which is representative of ON time in the total time period of pulse, and is given as:

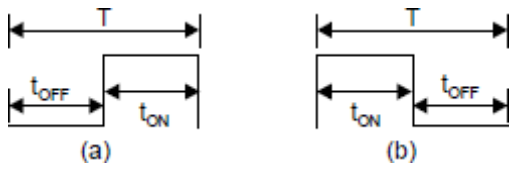


Fig 11: Representation of Pulse

$$\text{Duty Cycle} = D = \frac{t_{ON}}{t_{ON} + t_{OFF}} \text{ or } D = \frac{t_{ON}}{T}$$

In the digital systems we need a clock with duty cycle D ≤ 50%. The OFF time of a pulse is also referred as **bit-time**. This is the time in which flip-flop remains unaffected in either of two stable states. The state of latch during this time is due to the input signals present during ON time of pulse. State Q*n* + 1 is due to the inputs present during ON time of (*n* + 1)th pulse i.e at *t* = *n*T. In the analysis and discussion we adopt the designation Qn to represent **"present state"** which is the state before the ON time of (*n* + 1)th pulse or state just before the time *t* = *n*T in Fig 11(a) and Q*n* + 1 as **"next state"** i.e. the state just after the ON time of (*n* + 1)th clock pulse. Thus Q*n* represents the state (or output) in bit time *n* and Q*n* + 1 represents the output Q in bit time *n* + 1, as shown in Fig 11(b).

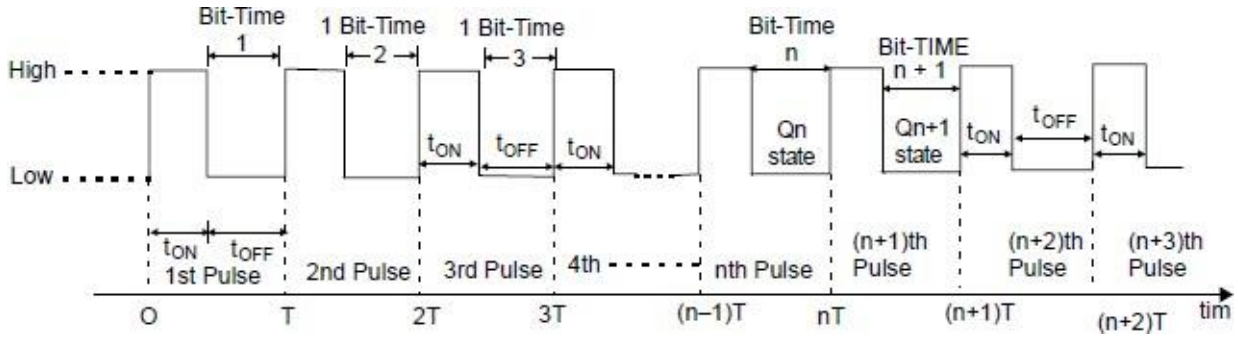
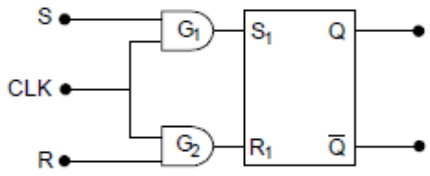


Fig 12: Clock signal-pulse train of shape

Clocked SR Flip-Flop: A simple way to get a clocked SR flip-flop is to AND the inputs signals with clock and then apply them to S and R inputs of flip-flop as shown in Fig.13(a).

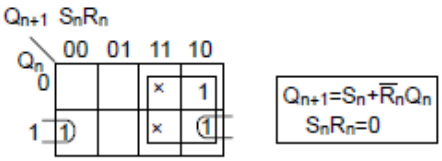
For the simplicity SET and RESET inputs of unclocked SR latch are labelled S1 and R1 respectively. Whereas external inputs are labelled S and R.



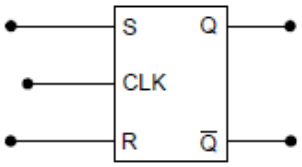
CLK	Qn	Sn	Rn	Q_{n+1}	Comments
0	0	×	×	0	Flip-Flop disabled no change in state and last state maintained
0	1	×	×	1	
1	0	0	0	0	Last State Qn+1=Qn
1	0	0	1	0	Reset state
1	0	1	0	1	Set state
1	0	1	1	?	Indeterminate
1	1	0	0	1	Last state Qn+1=Qn
1	1	0	1	0	Reset state
1	1	1	0	1	Set state
1	1	1	1	?	Indeterminate

(a) Construction of clocked Sr flip-flop.

(b) Truth Table

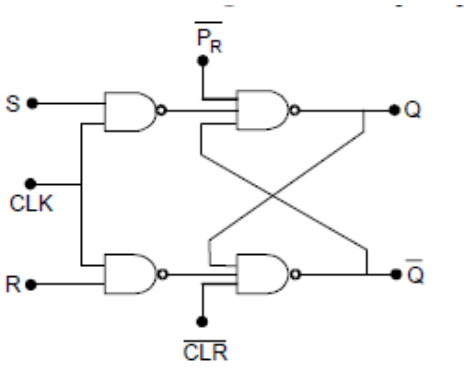


(c) Chare terigtic Equation



(d) Logic Symbol

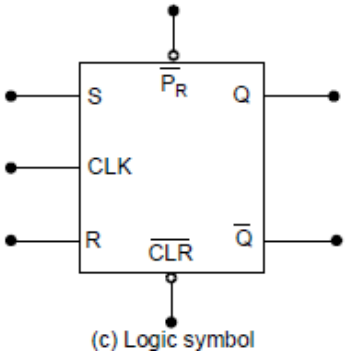
Fig 13: Clocked RS (or SR) flip-flop



(a) Construction

CLK	$\overline{P_R}$	$\overline{CLR}$	Out-Put Q	Resulting State
0	1	0	0	Clear
0	0	1	1	Preset or Set
0	0	0	?	Indeterminate
1	1	1	Q_{n+1}	Normal Flip-Flop Next state is determined by S R inputs:

(b) Truth table

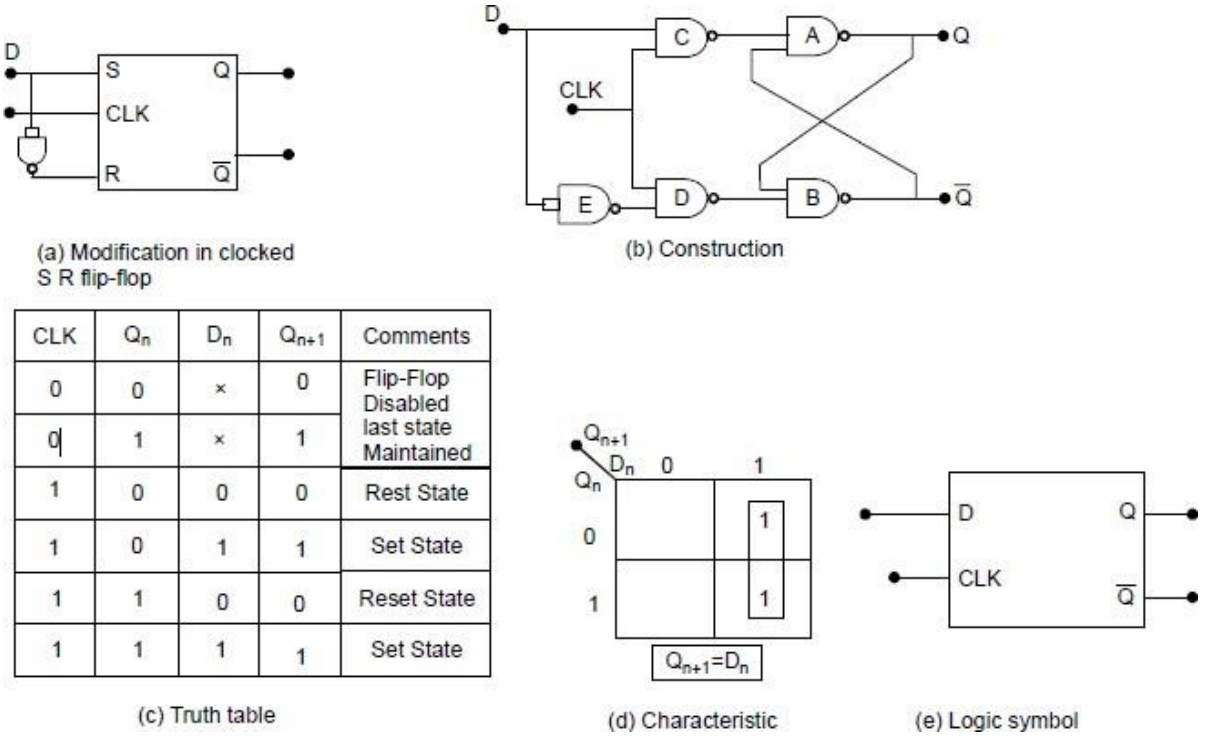


(c) Logic symbol

Fig 14: Realization of _CLEAR‘ and _PRESET‘ with SR flip-flop

Clocked D Flip-Flop

D flip-flop by using an external inverter present at the input of SR latch as shown in Fig 15(*a*). In the similar way a clocked D flip-flop is obtained by using an external inverter at the input of clocked SR flip-flop. The clocked D flip-flop is shown below in Fig. Note that unclocked RS latch of Fig.15(*a*) is replaced by a clocked RS flip-flop shown in Fig. 15(*d*).



CLK	Q_n	D_n	Q_{n+1}	Comments
0	0	×	0	Flip-Flop Disabled last state Maintained
0	1	×	1	
1	0	0	0	Rest State
1	0	1	1	Set State
1	1	0	0	Reset State
1	1	1	1	Set State

Fig 15: Clocked D flip-flop equation

Similar to clocked SR flip-flops, the clocked D flip-flop may also be accommodated with the asynchronous inputs ‑preset‖ and ‑clear‖. One particular realization is shown in Fig16. realization of clear and preset in D flip-flop.

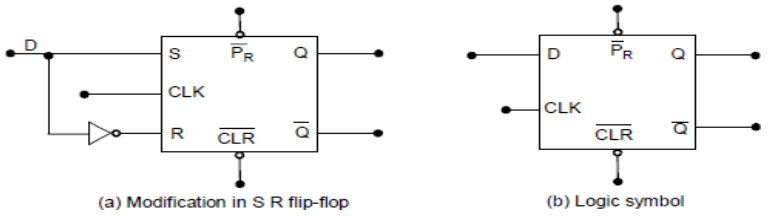
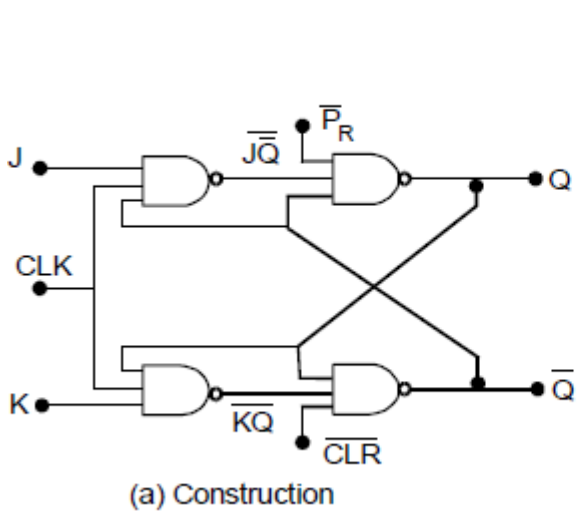
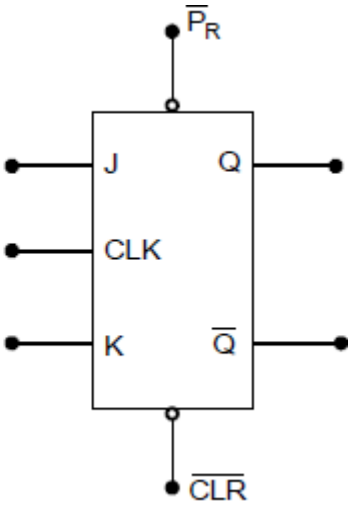


Fig 16: Clocked D flip-flop with PRESET and CLEAR inputs

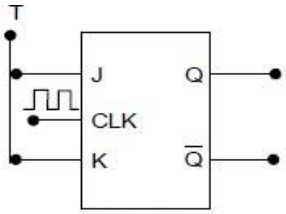
CLOCKED JK FLIPFLOP AND T FLOPFLIP



(a) Construction



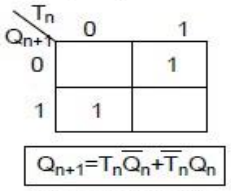
(b) Logic symbol



(a) Realization of T flip-flop from J K

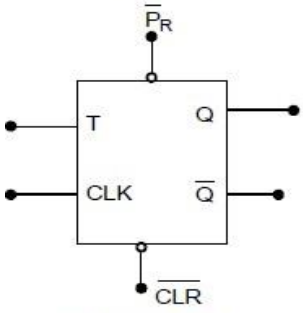
CLK	Present States		Applied Inputs			Next state Out-put		Resulting state
	Q_n	$\overline{Q}_n$	T_n	J_n	K_n	Q_{n+1}		
0	0	1	×	×	×	0	Q_n	Flip-Flop Disabled for CLK=0, no change next state=present
0	1	0	×	×	×	1		
1	0	1	0	0	0	0	Q_n	Maintain Out-put state no change in out-put next state=present state
1	1	0	0	0	0	1		
1	0	1	1	1	1	1	$\overline{Q}_n$	Toggled state or complemented state next state =present state
1	1	0	1	1	1	0		

(b) Detailed truth table

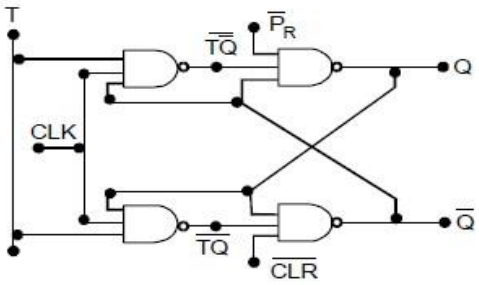


$Q_{n+1}=T_n\overline{Q}_n+\overline{T}_nQ_n$

(c) Characteristic equation



(e) Logic symbol



(d) Construction with active low clear & preset

Fig 17: Clocked JK Flip Flop and T Flip Flop

Clocked T (Toggles) flip-flop with active LOW asynchronous inputs The JK flip-flops are very popular as indeterminate state (as present in SR type) does not exist. Furthermore, due to the toggling capability, when both inputs HIGH, on each arrival of pulse, it forms the basic element for counters. For this purpose JK flip-flop is further modified to provide a T flip-flop as shown in Fig 17.

T flip-flop is a single input version of JK flip-flop, in which the inputs J and K are connected together, as shown, and is provided as a single input labelled as T. The operation is straight forward and easy, and summarized in truth-table given in Fig. 17(b), while characteristic equation is derived in Fig 17(c). When the clock is absent, the flip-flop is disable as usual and previously latched output is maintained at output. When the clock is present and T = 0, even though flip-flop is enabled the output does not switch its state. It happens so because for T = 0 we get J = K = 0 and thus next state is same as present state. Thus if either CLK = 0 or T = 0, state does not change next state is always same as present state.

When T = 1 during CLK = 1, it causes J = K = 1 and as earlier discussed it will toggle the output state. Thus when input T is HIGH, flip-flop toggles its output on the arrival of clock, and for this reason input T is called as the **Toggle Input**. Essentially the T flip-flop also, suffer from race around condition, (when input is HIGH) and thus causing multiple transition at output due to same reasons given in example.

Race Around Condition and Solution

Whenever the width of the trigger pulse is greater than the propagation time of the flipflop, then flip-flop continues to toggle 1-0-1-0 until the pulse turns 0. When the pulse turns 0, unpredictable output may result i.e. we don‘t know in what state the output is whether 0 or 1. This is called race around condition. In level-triggered flip-flop circuits, the circuits is always active when the clock signal is high, and consequently unpredictable output may result. For example, during this active clock period, the output of a T-FF may toggle continuously. The output at the end of the active period is therefore unpredictable. To overcome this problem, *edge triggered* circuits can be used whose output is determined by the edge, instead of the level, of the clock signal, for example, the rising (or trailing) edge.

Another way to resolve the problem is the Master-Slave circuit shown in Fig 18.

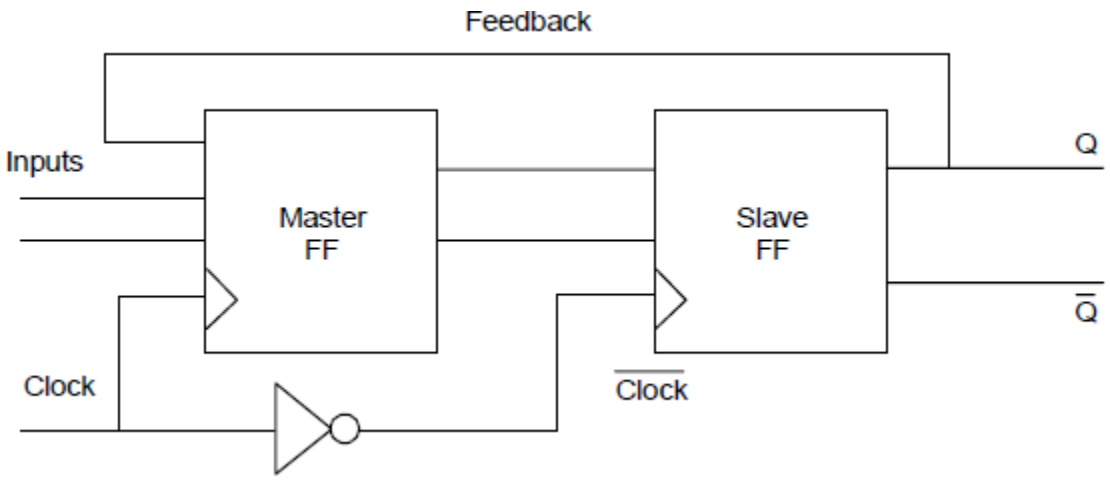


Fig 18: Master slave circuit

The operation of a Master-Slave FF has two phases as shown in Fig

- During the high period of the clock, the master FF is active, taking both inputs and feedback from the slave FF. The slave FF is de-activated during this period by the negation of the clock so that the new output of the master FF won‘t effect it.
- During the low period of the clock, the master FF is deactivated while the slave FF is active. The output of the master FF can now trigger the slave FF to properly set its output. However, this output will not effect the master FF through the feedback as it is not active.

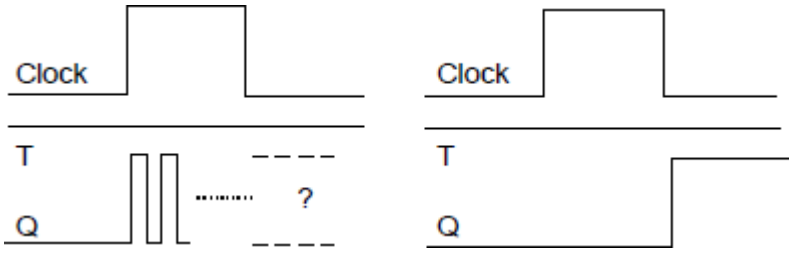


Fig 19: Master slave operation

It is seen that the trailing edge of the clock signal will trigger the change of the output of the Master-Slave FF. The logic diagram for a basic master-slave S-R flip-flop is shown in Fig 20.

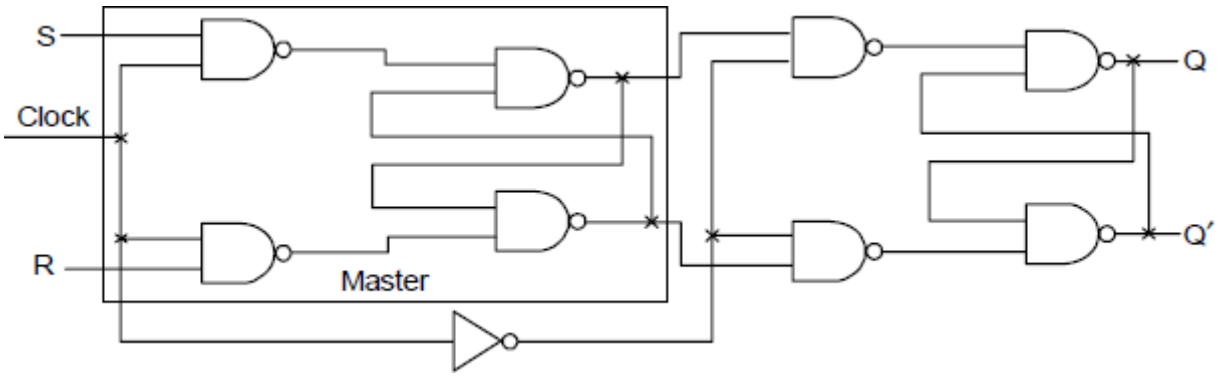


Fig 20: S-R master slave flip-flop

Flip-flops are generally used for storing binary information. One bit of information can be written into a flip-flop, and later read out from it. If a master-slave FF is used, both read and write operations can take place during the same clock cycle under the control of two control signals **read** and **write** as shown in Fig 21.

• During the first half of clock cycle: **clock = read = write = 1,** the old content in slave-FF is read out, while the new content is being written into master-FF at the same time.,

• During the second half of clock cycle : **clock = read = write = 0,** the new content in master-FF is written into slave-FF

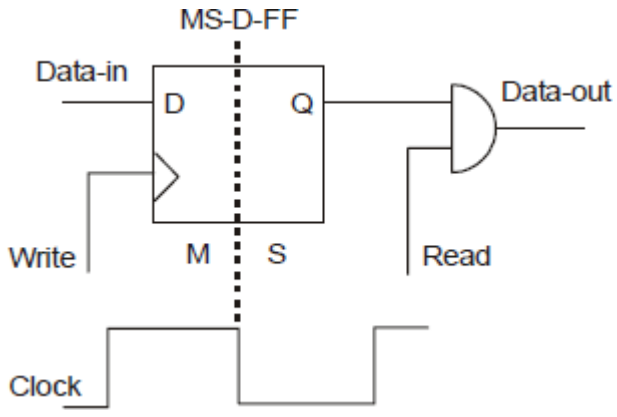


Fig 21: S-R master slave flip-flop Read and Write operation

Operating Characteristics of Flip-flops

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

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

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

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

Maximum Clock Frequency—is the highest rate that a flip-flop can be reliably triggered.

Power Dissipation—is the total power consumption of the device.

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

Flip-Flop Applications

- Frequency Division
- Parallel Data Storage

FLIP-FLOP EXCITATION TABLE

The characteristic table is useful during the analysis of sequential circuits when the value of flip-flop inputs are known and we want to find the value of the flip-flop output Q after the rising edge of the clock signal. As with any other truth table, we can use the map method to derive the characteristic equation for each flip-flop.

During the design process we usually know the transition from present state to the next state and wish to find the flip-flop input conditions that will cause the required transition. For this reason

we will need a table that lists the required inputs for a given change of state. Such a list is called the *excitation table.*

There are four possible transitions from present state to the next state. The required input conditions are derived from the information available in the characteristic table.

The symbol X in the table represents a –don‘t care‖ condition, that is, it does not matter whether the input is 1 or 0.

The different types of flip flops (RS, JK, D, T) can also be described by their excitation, table as shown in Fig. The left side shows the desired transition from Q*n* to Q*n*+1, the right side gives the triggering signals of various types of FFs needed for the transitions.

Desired transition		*Triggering signal needed*					
Q_n	Q_{n+1}	*S*	*R*	*J*	*K*	*D*	*T*
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	0	1
1	1	x	0	x	0	1	0

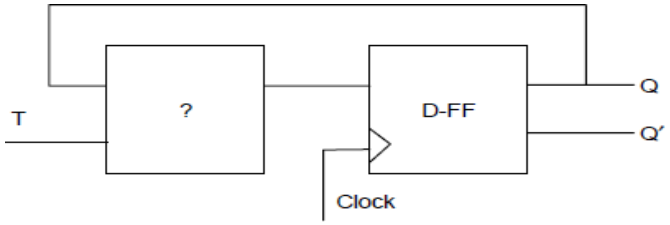
Excitation table

FLIP-FLOP CONVERSIONS

This section shows how to convert a given type A FF to a desired type B FF using some conversion logic.

The key here is to use the excitation table of Fig. which shows the necessary triggering signal (S, R, J, K, D and T) for a desired flip flop state transition Q*n* →Q*n*+1 is reproduced here.

Example 1. *Convert a D-FF to a T-FF:*

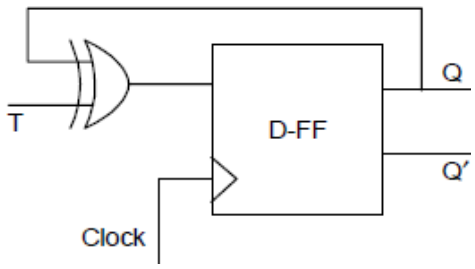


We need to design the circuit to generate the triggering signal D as a function of T and Q : D $= f$ (T, Q)

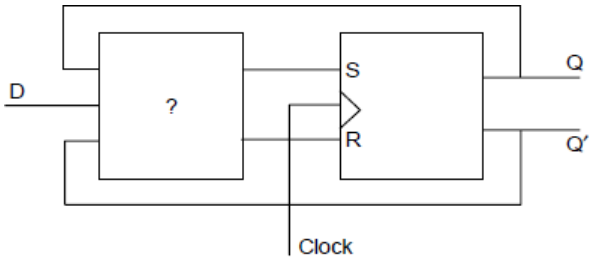
Consider the excitation table:

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

Treating D as a function of T and current FF state Q Q*n* we have D = T′Q + TQ = T⊕Q



Example 2. *Convert a RS-FF to a D-FF:*

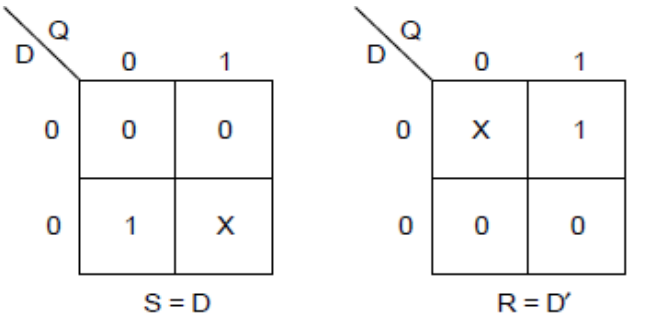


We need to design the circuit to generate the triggering signals S and R as functions of D and Q. Consider the excitation table:

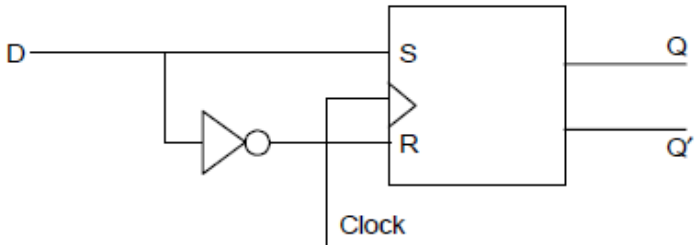
Q_n	Q_{n+1}	D	S	R
0	0	0	0	X
0	1	1	1	0
1	0	0	0	1
1	1	1	X	0

The desired signal S and R can be obtained as functions of T and current FF state Q

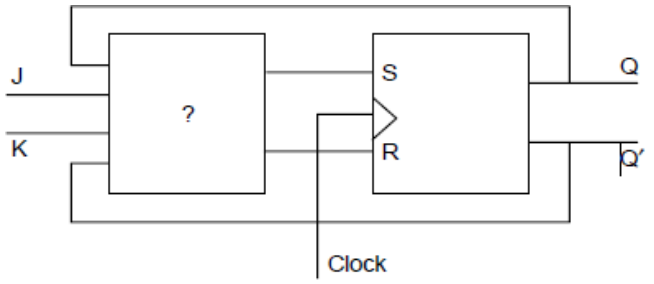
from the Karnaugh maps:



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



Example 3. *Convert a RS-FF to a JK-FF.*



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

Q_n	Q_{n+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

The desired signals S and R as function J, K and current FF state Q can be obtained from the Karnaugh maps:

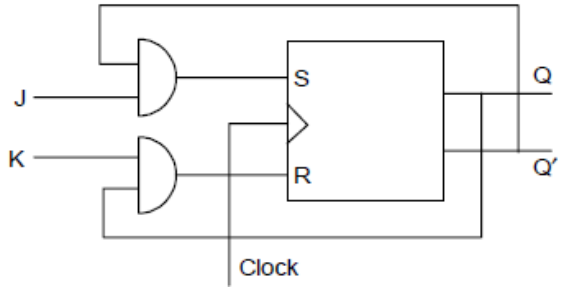
K \ QJ	00	01	11	10
0	0	1	X	X
1	1	1	0	0

S = Q′J

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

R = QK

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



DESIGN OF RIPPLE COUNTER

A register that goes through a prescribed sequence of states upon the application of input pulses is called a counter. The input pulses may be clock pulses, or they may originate from some external source and may occur at a fixed interval of time or at random. The sequence of states may follow the binary number sequence or any other sequence of states. A counter that follows the binary number sequence is called a binary counter. An n -bit binary counter consists of n flip-flops and can count in binary from 0 through 2n - 1.

Counters are available in two categories: ripple counters and synchronous counters.

In a ripple counter, a flip-flop output transition serves as a source for triggering other flip-flops. In other words, the C input of some or all flip-flops are triggered, not by the common clock pulses, but rather by the transition that occurs in other flip-flop outputs. In a synchronous counter, the C inputs of all flip-flops receive the common clock.

Binary Ripple Counter

A ripple counter is an asynchronous counter where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop. Asynchronous counters are also called ripple-counters because of the way the clock pulse ripples it way through the flip-flops. The MOD of the ripple counter or asynchronous counter is 2n if n flip-flops are used. For a 4-bit counter, the range of the count is 0000 to 1111 (24-1). A counter may count up or count down or count up and down depending on the input control. The count sequence usually repeats itself. When counting up, the count sequence goes from 0000, 0001, 0010, ... 1110 , 1111 , 0000, 0001, ... etc. When counting down the count sequence goes in the opposite manner: 1111, 1110, ... 0010, 0001, 0000, 1111, 1110, ... etc.

The complement of the count sequence counts in reverse direction. If the uncomplemented output counts up, the complemented output counts down. If the uncomplemented output counts down, the complemented output counts up. There are many ways to implement the ripple counter depending on the characteristics of the flip flops used and the requirements of the count sequence.

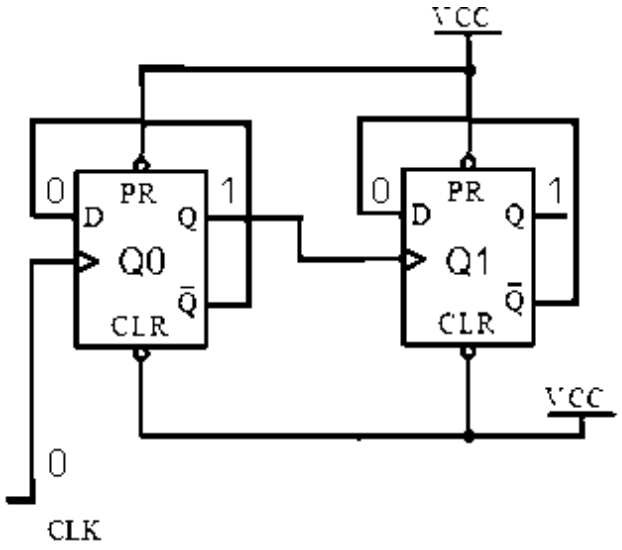
· Clock Trigger: Positive edged or Negative edged

· JK or D flip-flops

· Count Direction: Up, Down, or Up/Down

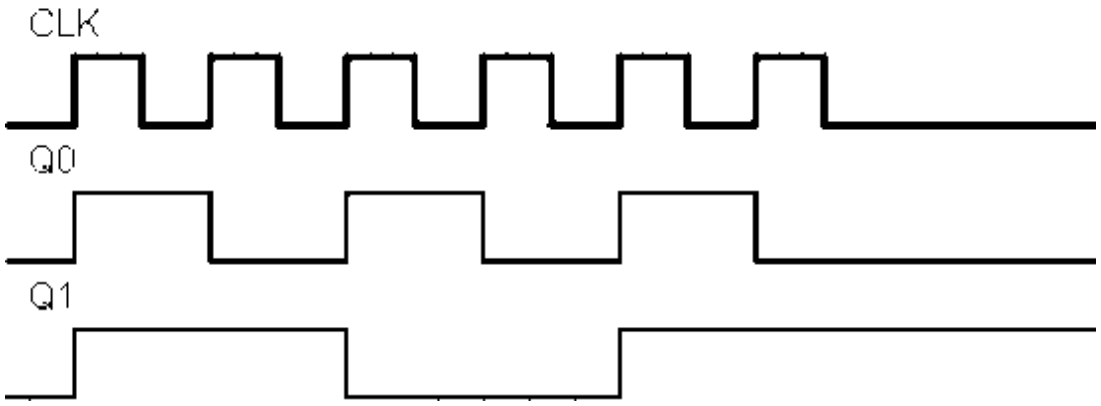
Asynchronous counters are slower than synchronous counters because of the delay in the transmission of the pulses from flip-flop to flip-flop. With a synchronous circuit, all the bits in the count change synchronously with the assertion of the clock. Examples of synchronous counters are the Ring and Johnson counter.

It can be implemented using D-type flip-flops or JK-type flip-flops. The circuit below uses 2 D flip-flops to implement a divide-by-4 ripple counter (2n = 22 = 4). It counts down.



TRUTH TABLE

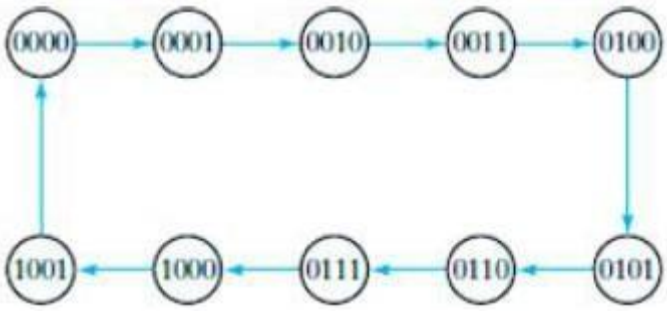
COUNT	Q1	Q0
3	1	1
2	1	0
1	0	1
0	0	0



- Click on CLK (Red) switch and observe the changes in the outputs of the flip flops. The CLK switch is a momentary switch (similar to a door bell switch - normally off).
- PR and CLR are both connected to VCC (set to 1)
- The D flip flop clock has a rising edge CLK input. For example Q0 behaves as follows
    - The D input value just before the CLK rising edge is noted (Q00).
    - When CLK rising edge occurs, Q0 is assigned the previously noted D value (Q00).
    - Thus, whenever a rising edge appears at the CLK of the D flip flop, the output Q changes state (or toggles).
- The MOD or number of unique states of this 2 flip flop ripple counter is 4 (22).
- Simulate and Breadboard the Ripple Counter circuit.
- A Truncated Ripple Counter is used if a MOD of less than 2n is required. For example, if you want to change the sequence from 3,2,1,0,3,2,1,0 ... to 3,2,0,3,2,0 ...

BCD Ripple Counter

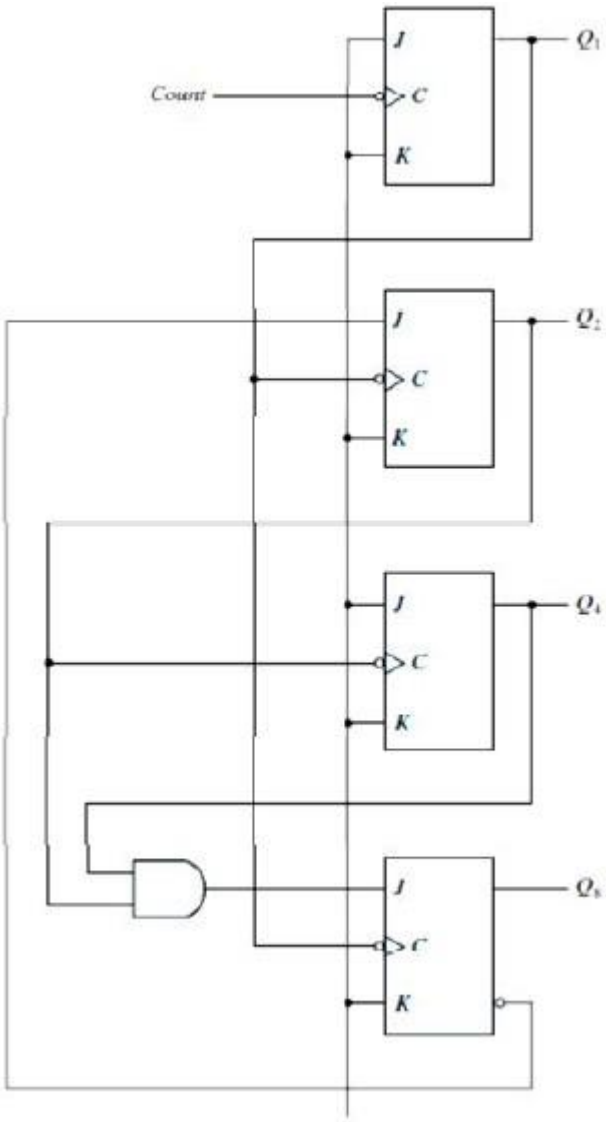
A decimal counter follows a sequence of 10 states and returns to 0 after the count of 9. Such a counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits. The sequence of states in a decimal counter is dictated by the binary code used to represent a decimal digit. If BCD is used, the sequence of states is as shown in the state diagram of Fig1. A decimal counter is similar to a binary counter, except that the state after 1001 (the code for decimal digit 9) is 0000 (the code for decimal digit 0).



The logic diagram of a BCD ripple counter using *JK* flip-flops is shown in Fig. below. 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. Note that the output of *Q*1 is applied to the *C*

inputs of both $Q2$ and $Q8$ and the output of $Q2$ is applied to the C input of $Q4$. The J and K inputs are connected either to a permanent 1 signal or to outputs of other flip-flops.

A ripple counter is an asynchronous sequential circuit. Signals that affect the flip-flop transition depend on the way they change from 1 to 0. The operation of the counter can be explained by a list of conditions for flip-flop transitions. These conditions are derived from the logic diagram and from knowledge of how a *JK* flip-flop operates. Remember that when the C input goes from 1 to 0, the flip-flop is set if $J = 1$, is cleared if $K = 1$, is complemented if $J = K = 1$, and is left unchanged if $J = K = 0$.



SYNCHRONOUS COUNTERS

Synchronous counters are different from ripple counters in that clock pulses are applied to the inputs of all flip-flops. A common clock triggers all flip-flops simultaneously, rather than one at a time in succession as in a ripple counter. The decision whether a flip-flop is to be complemented is determined from the values of the data inputs, such as T or J and K at the time of the clock edge. If T = 0 or J = K = 0, the flip-flop does not change state. If T = 1 or J = K = 1, the flip-flop complements.

Binary Counter

The design of a synchronous binary counter is so simple that there is no need to go through a sequential logic design process. In a synchronous binary counter, the flip-flop in the least significant position is complemented with every pulse. A flip-flop in any other position is complemented when all the bits in the lower significant positions are equal to 1. For example, if the present state of a four-bit counter is A3A2A1A0 = 0011, the next count is 0100. A0 is always complemented. A1 is complemented because the present state of A0 = 1. A2 is complemented because the present state of A1A0 = 11. However, A3 is not complemented, because the present state of A2A1A0 = 011, which does not give an all-1‘s condition.

Synchronous binary counters have a regular pattern and can be constructed with complementing flip-flops and gates. The regular pattern can be seen from the four-bit counter depicted in Fig. below. The C inputs of all flip-flops are connected to a common clock. The counter is enabled by Count_enable. If the enable input is 0, all J and K inputs are equal to 0 and the clock does not change the state of the counter. The first stage, A0, has its J and K equal to 1 if the counter is enabled. The other J and K inputs are equal to 1 if all previous least significant stages are equal to 1 and the count is enabled.

The chain of AND gates generates the required logic for the J and K inputs in each stage. The counter can be extended to any number of stages, with each stage having an additional flip-flop and an AND gate that gives an output of 1 if all previous flip-flop outputs are 1.

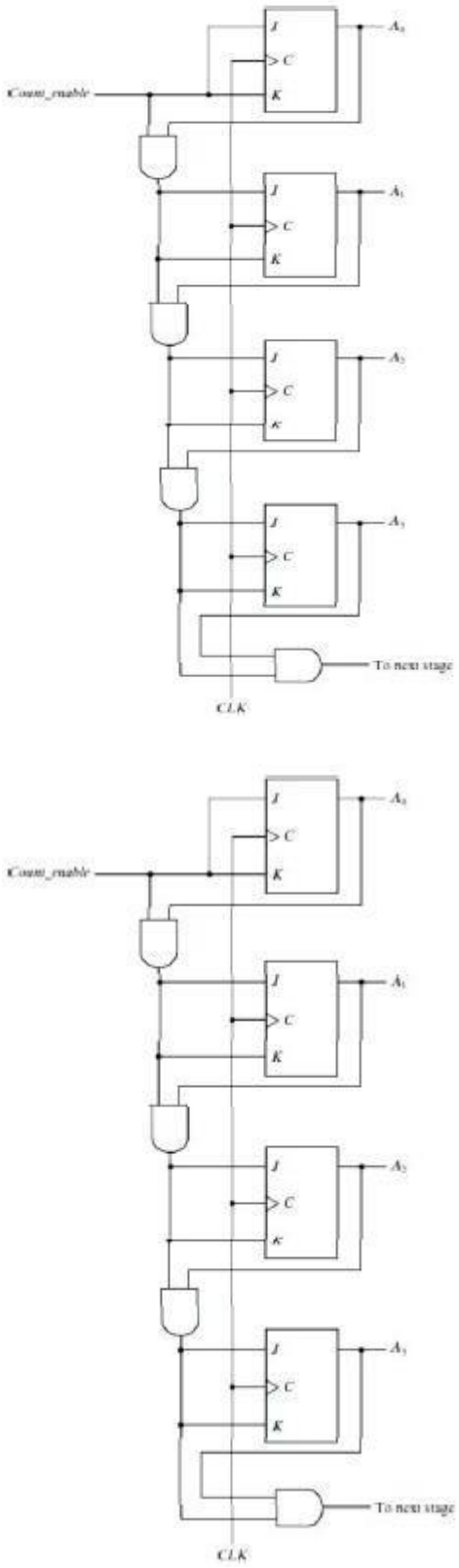


Fig 26: Four-bit Synchronous binary Counter

Binary Counter

A synchronous countdown binary counter goes through the binary states in reverse order, from 1111 down to 0000 and back to 1111 to repeat the count. It is possible to design a countdown counter in the usual manner, but the result is predictable by inspection of the downward binary count. The bit in the least significant position is complemented with each pulse. A bit in any other position is complemented if all lower significant bits are equal to 0. For example, the next state after the present state of 0100 is 0011. The least significant bit is always complemented.

The second significant bit is complemented because the first bit is 0. The third significant bit is complemented because the first two bits are equal to 0. But the fourth bit does not change, because not all lower significant bits are equal to 0.

A countdown binary counter can be constructed as shown in Fig. below, except that the inputs to the AND gates must come from the complemented outputs, instead of the normal outputs, of the previous flip-flops. The two operations can be combined in one circuit to form a counter capable of counting either up or down. The circuit of an up–down binary counter using T flip-flops is shown in Fig. It has an up control input and a down control input. When the up input is 1, the circuit counts up, since the T inputs receive their signals from the values of the previous normal outputs of the flip-flops. When the down input is 1 and the up input is 0, the circuit counts down, since the complemented outputs of the previous flip-flops are applied to the T inputs. When the up and down inputs are both 0, the circuit does not change state and remains in the same count.

When the up and down inputs are both 1, the circuit counts up. This set of conditions ensures that only one operation is performed at any given time. Note that the up input has priority over the down input.

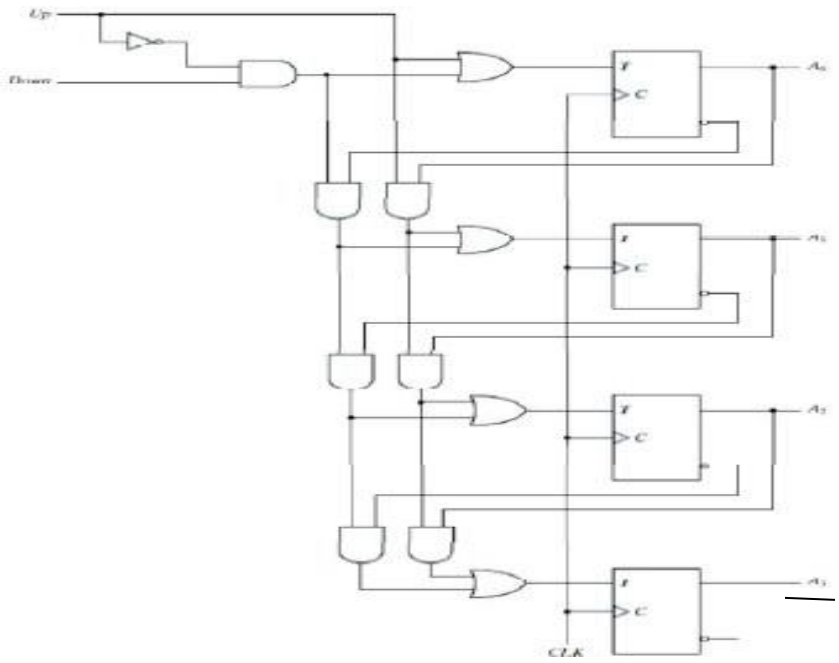


Fig 27: Four-bit up-down binary counter

Ring Counter

A ring counter is a Shift Register (a cascade connection of flip-flops) with the output of the last flip flop connected to the input of the first. It is initialized such that only one of the flip flop output is 1 while the remainder is 0. The 1 bit is circulated so the state repeats every n clock cycles if n flip-flops are used. The "MOD" or "MODULUS" of a counter is the number of unique states. The MOD of the n flip flop ring counter is n. It can be implemented using D-type flip-flops (or JK-type flip-flops).

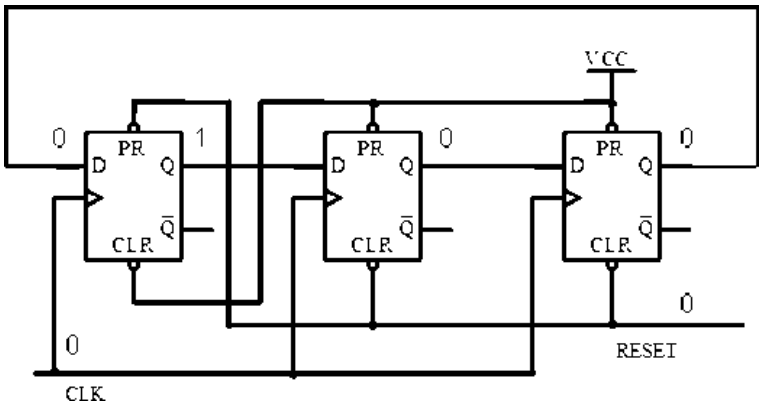


Fig 28: Ring Counter

Notes:

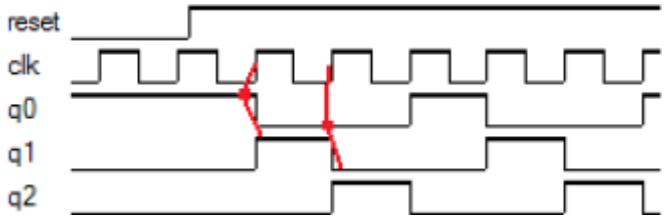
· Enable the flips flops by clicking on the RESET (Green) switch. The RESET switch is a on/off switch (similar to a room light switch)

· Click on CLK (Red) switch and observe the changes in the outputs of the flip flops. The CLK switch is a momentary switch (similar to a door bell switch - normally off).

· The D flip flop clock has a rising edge CLK input. For example Q1 behaves as follows:

-The D input value just before the CLK rising edge is noted (Q0).

-When CLK rising edge occurs, Q1 is assigned the previously noted D value (Q0).



The MOD or number of unique states of this 3 flip flop ring counter is 3.

Truth Table

State	Q0	Q1	Q2
0	1	0	0
1	0	1	0
2	0	0	1

Johnson Counter

A Johnson counter is a modified ring counter, where the inverted output from the last flip flop is connected to the input to the first. The register cycles through a sequence of bit-patterns. The MOD of the Johnson counter is 2n if n flip-flops are used. The main advantage of the Johnson

counter is that it only needs half the number of flip-flops compared to the standard ring counter for the same MOD.

It can be implemented using D-type flip-flops (or JK-type flip-flops).

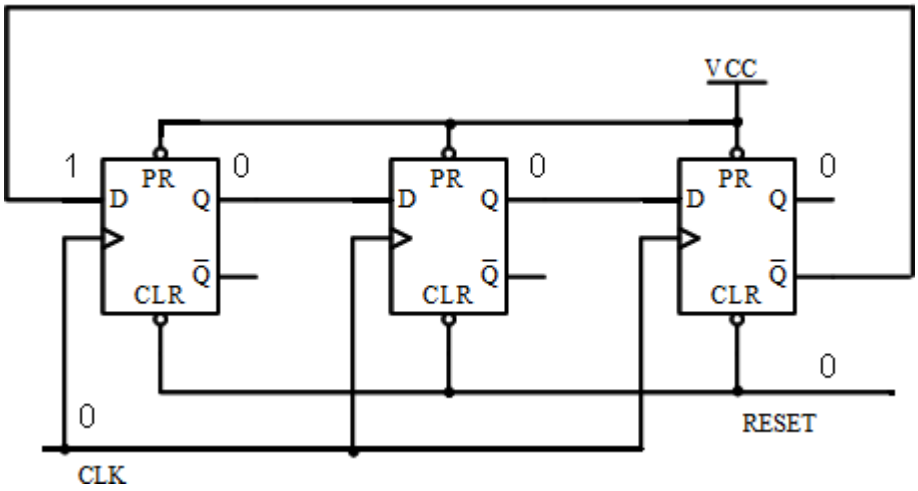
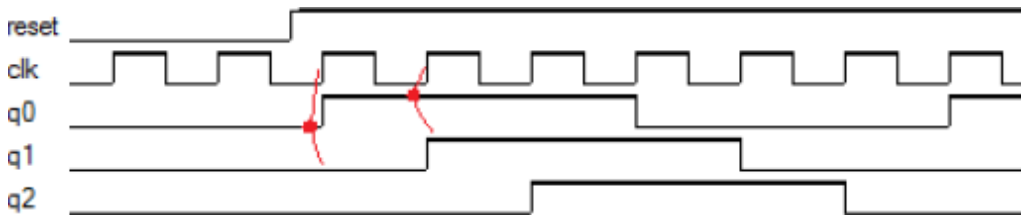


Fig 29: Johnson Counter

Notes:

- Enable the flips flops by clicking on the RESET (Green) switch. The RESET switch is a on/off switch (similar to a room light switch)
- Click on CLK (Red) switch and observe the changes in the outputs of the flip flops. The
- CLK switch is a momentary switch (similar to a door bell switch - normally off).
    - The D flip flop clock has a rising edge CLK input. For example Q1 behaves as follows:
- The D input value just before the CLK rising edge is noted (Q0).
- When CLK rising edge occurs, Q1 is assigned the previously noted D value (Q0).



The MOD or number of unique states of this 3 flip flop Johnson counter is 6.

Truth Table

State	Q0	Q1	Q2
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1

DESIGN OF REGISTER:

A clocked sequential circuit consists of a group of flip-flops and combinational gates. The flip-flops are essential because, in their absence, the circuit reduces to a purely combinational circuit (provided that there is no feedback among the gates). A circuit with flip-flops is considered a sequential circuit even in the absence of combinational gates. Circuits that include

flip-flops are usually classified by the function they perform rather than by the name of the sequential circuit. Two such circuits are registers and counters.

A register is a group of flip-flops, each one of which shares a common clock and is capable of storing one bit of information. An n -bit register consists of a group of n flip-flops capable of storing n bits of binary information. In addition to the flip-flops, a register may have combinational gates that perform certain data-processing tasks. In its broadest definition, a register consists of a group of flip-flops together with gates that affect their operation. The

flip-flops hold the binary information, and the gates determine how the information is transferred into the register.

A counter is essentially a register that goes through a predetermined sequence of binary states. The gates in the counter are connected in such a way as to produce the prescribed sequence of states. Although counters are a special type of register, it is common to differentiate them by giving them a different name.

Various types of registers are available commercially. The simplest register is one that consists of only flip-flops, without any gates. A register constructed with four D -type flip-flops to form a four-bit data storage register is shown in figure below. The common clock input triggers all flip-flops on the positive edge of each pulse, and the binary data available at the four inputs are transferred into the register. The value of (I 3 , I 2 , I 1 , I 0) immediately before the clock edge determines the value of (A 3 , A 2 , A 1 , A 0) after the clock edge. The four outputs can be sampled at any time to obtain the binary information stored in the register.

The input Clear_b goes to the active-low R (reset) input of all four flip-flops. When this input goes to 0, all flip-flops are reset asynchronously. The Clear_b input is useful for clearing the register to all 0‘s prior to its clocked operation. The R inputs must be maintained at logic 1 (i.e., de-asserted) during normal clocked operation. Note that, depending on the flip-flop, either Clear, Clear_b, reset, or reset_b can be used to indicate the transfer of the register to an all 0‘s state.

SHIFT REGISTERS:

A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction, is called a *shift register*. The logical configuration of a shift register consists of a chain of flip-flops in cascade, with the output of one flip-flop connected to the input of the next flip-flop. All flip-flops receive common clock pulses, which activate the shift of data from one stage to the next. The simplest possible shift register is one that uses only flip-flops, as shown in Fig.

The output of a given flip-flop is connected to the D input of the flip-flop at its right. This shift register is unidirectional (left-to-right). Each clock pulse shifts the contents of the register one bit position to the right. The configuration does not support a left shift. The serial input determines what goes into the leftmost flip-flop during the shift. The serial output is taken from the output of the rightmost flip-flop.

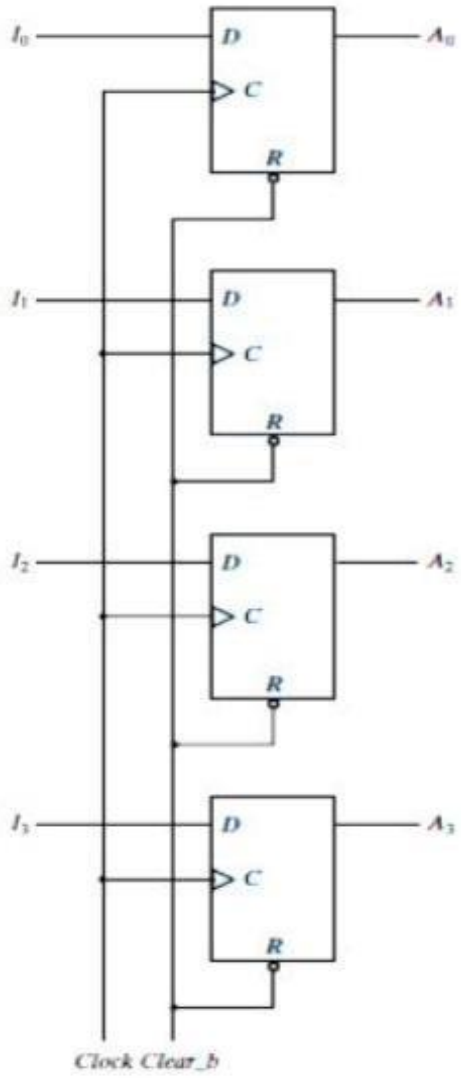


Fig 22: 4 –Bit Register

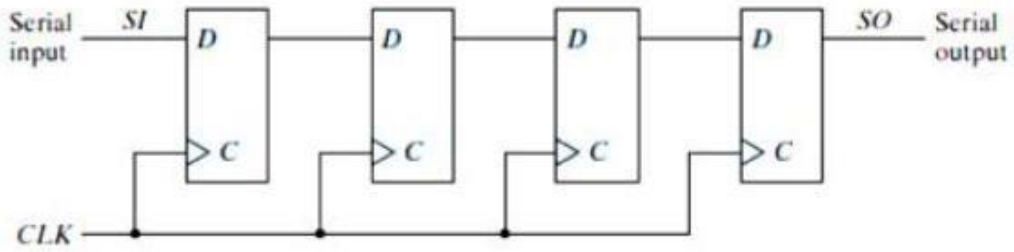


Fig 23: **Four bit Shift register**

Sometimes it is necessary to control the shift so that it occurs only with certain pulses, but not with others. As with the data register discussed in the previous section, the clock‘s signal can be suppressed by gating the clock signal to prevent the register from shifting. A preferred alternative in high speed circuits is to suppress the clock action, rather than gate the clock signal, by leaving the clock path unchanged, but recirculating the output of each register cell back through a two-channel mux whose output is connected to the input of the cell. When the clock action is not suppressed, the other channel of the mux provides a data path to the cell.

Shift registers have found considerable application in arithmatic operations. Since, moving a binary number one bit to the left is equivalent to multiplying the number by 2 and moving the number one bit position to the right amounts to dividing the number by 2. Thus, multiplications and divisions can be accomplished by shifting data bits. Shift registers find considerable application in generating a sequence of control pulses.

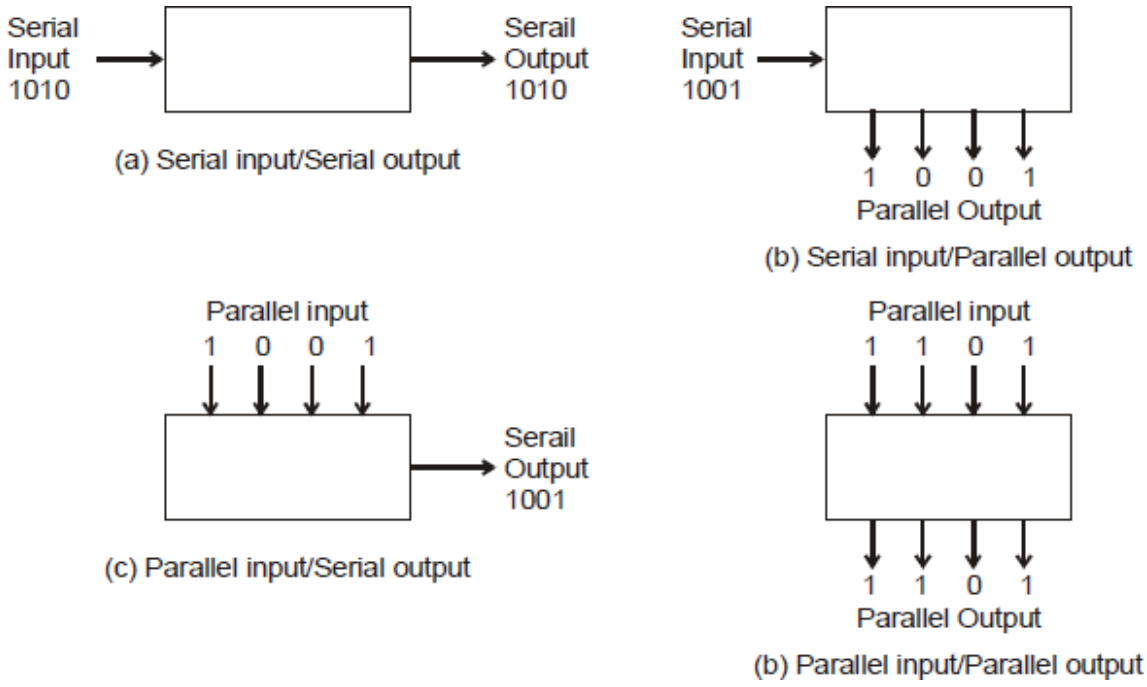


Fig 24: Data Transmission in Shift Register

Bidirectional Shift Registers (Universal Shift Register)

The registers discussed so far involved only right shift operations. Each right shift operation has the effect of successively dividing the binary number by two. If the operation is reversed (left shift), this has the effect of multiplying the number by two. With suitable gating arrangement a serial shift register can perform both operations.

A bi-directional, or reversible shift register is one in which the data can be shift either left or right. A four-bit bi-directional shift register using D-flip-flops is shown in Fig 25.

Here a set of NAND gates are configured as OR gates to select data inputs from the right or left adjacent bistables, as selected by the $\overline{\text{LEFT}}$_/RIGHT control line.

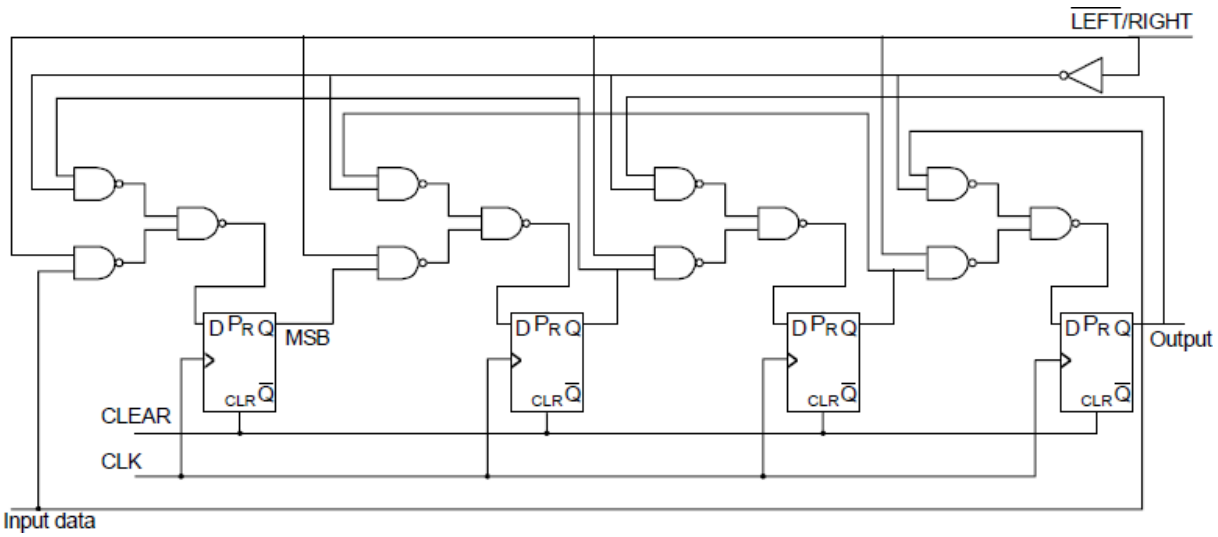


Fig 25: 4 Bit Universal Shift Register

APPLICATIONS OF SHIFT REGISTERS

Shift registers can be found in many applications. Here is a list of a few.

- To Produce Time Delay
- To Simplify Combinational Logic
- To Convert Serial Data to Parallel Data

Assignment-Cum-Tutorial Questions

A. Questions testing the remembering / understanding level of students

I) Objective Questions

1. A sequential logic circuit
   A) Must contain flip-flops B) may contain flip-flops
   C) does not contain flip-flops D) contain latches
2. A sequential circuit does not use clock pulses. It is
   A) an asynchronous sequential circuit B) a synchronous sequential circuit
   C) a counter D) a shift register
3. The basic memory element in a digital circuit
   A) consists of a NAND gate B) consists of a NOR gate
   C) is a flip-flop D) is a shift register
4. A flip-flop can store
   A) one bit of data B) two bits of data
   C) tree bits of data D) any number of bits of data
5. The characteristic equation of a J-K flip-flop is
   A) $Q_{n+1}=J\overline{Q_n}+\bar{K}Q_n$ B) $Q_{n+1}=JQ_n+K\overline{Q_n}$
   C) $Q_{n+1}=\bar{J}Q_n+\bar{K}Q_n$ D) $Q_{n+1}=\bar{J}Q_n+\overline{\bar{K}Q_n}$
6. The characteristic equation of a D flip-flop is
   A) $Q_{n+1}=D$ B) $Q_{n+1}=Q_n$
   C) $Q_{n+1}=1$ D) $Q_{n+1}=\overline{Q_n}$
7. A shift register using flip-flops is called a
   A) dynamic shift register B) flip-flop shift register
   C) static shift register D) buffer shift register
8. Dynamic shift register is made up of
   A) dynamic flip-flops B) MOS inverter
   C) MOS NAND gates D) CMOS inverter
9. The transparent latch is
   A) an S-R flip-flop B) a D flip-flop
   C) a T flip-flop D) a J-K flip flop
10. A universal register
    A) accepts serial input B) accepts parallel input
    C) gives serial and parallel D) is cable of all of the above

II) Descriptive Questions

1. Distinguish between combinational and sequential circuits.
2. Find the characteristic equation for JK flip-flop.
3. Convert a J-K flip-flop into T flip-flop

4. Convert an SR flip-flop into JK flip-flop
5. Discuss the applications of flip-flops
6. What is a universal shift register
7. Write the design steps of synchronous counters
8. Write the Excitation Tables for D, T, SR, JK Flip Flops
9. Explain the Operating Characteristics of Flip Flops
10. What are shift register counters? Draw Ring Counter and explain the operation with Truth Table.

B. Question testing the ability of students in applying the concepts.

I) Multiple Choice Questions

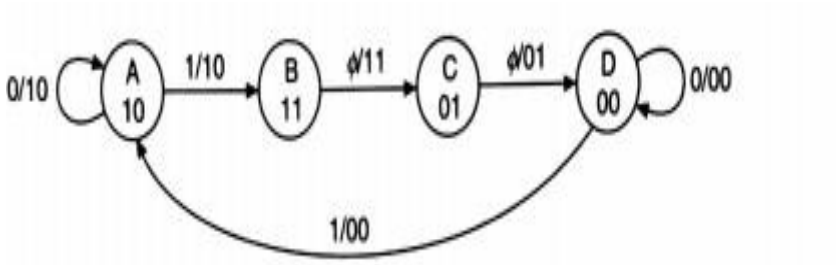
1. The output Q_n of a J-K flip-flop is 1. It changes to 0 when a clock pulse is applied. The inputs J_n and K_n are respectively
   A) 0 and X B) 1 and X C) X and 1 D) 0 and X
2. The output Q_n of a S-R flip-flop is 0.It changes to 1 when a clock pulse is applied. The inputs S_n and R_n are respectively
   A) X and 1 B) 0 and 1 C) X and 0 D) X and 1
3. The outputs Q and Q‘ of a master slave S-R flip-flop are connected to its R and S inputs respectively. Its output Q when clock pulses are applied will be
   A) permanently 0 B) permanently 1
   C) fixed 0 or 1 D) complementing with every clock pulse
4. A 4-bit binary ripple counter uses flip-flops with propagation delay time of 25ns each. The maximum possible time required for change of state will be
   A) 25ns B) 50ns C) 75ns D) 100ns
5. A 4-bit synchronous counter uses flip-flops with propagation delay time of 25ns each. The maximum possible time required for change of state will be
   A) 25ns B) 50ns C) 75ns D) 100ns
6. A mod-2 counter followed by a mod-5 counter is
   A) the same as a mod-5 counter followed by a mod-2 counter
   B) a decade counter C) a mod-7 counter D) none of above
7. A sequential circuit with ten states will have
   A) 10 flip-flops B) 5 flop-flops C) 4 flip-flops D) 0 flip-flops
8. The output frequency of a mod-16 counter, clocked from a 20 KHz clock input signal is
   A) 20 KHz B) 52 KHz C) 625 Hz D) 1250 Hz
9. The output frequency of a mod-12 counter is 6KHz. Its input frequency is
   A) 6 KHz B) 500 Hz C) 24 KHz D) 72 KHz
10. The minimum number of flop-flops required for a mod-12 ripple counter is
    A) 3 B) 4 C) 6 D) 12

II) Problems

1. Design a 3 bit synchronous up/down counter using JK flip-flop.
2. Design a mod 7 asynchronous counter using JK flip-flop.
3. Design a mod 12 synchronous counter using T flip-flop.
4. Design a BCD up/down counter using SR flip-flop.
5. Design a J-K counter that goes through states 2,4,5,7,2,4,…… Is the counter self-starting? Modify the circuit such that whenever it goes to any invalid state it comes back t state 2.
6. Convert D flip-flop to SR flip-flop.
7. Convert T flip-flop to JK flip-flop.

C. Questions testing the analyzing / evaluating ability of students

1. Design a circuit using S-R flip-flops that will function as per the state diagram shown below



2) Design a synchronous counter that counts as 000, 010, 101, 110, 000, 010 … Ensure that the un used states of 001, 011, 100 and 111 go to 000 on the next clock pulse. Use J-K flip-flops.

3) Find a modulo-6 gray code using k-map and design the corresponding counter.

4) The content of a 4 bit shift register is initially 1101. The register is shifted six times to the right, with the serial input being 101101 what is the content of the register after each shift?

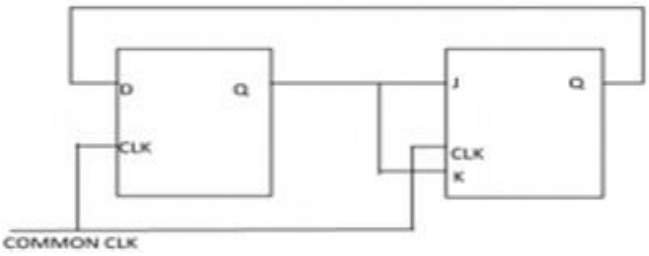
D. GATE/IES Questions

1. A synchronous counter counts the sequence 0-1-0-2-0-3 and then repeats. The minimum number of J-K flip-flops required to implement this counter is **GATE-2016**
   A) 1 B) 2 C) 4 D) 5

2. A positive edge-triggered D flip-flop is connected to a positive edge-triggered JK flip flop as follows. The Q output of the D flip-flop is connected to both the J and K inputs of the JK flip-flop, while the Q output of the JK flip-flop is connected to the input of the D flip-flop. Initially, the output of the D flip-flop is set to logic one and the output of the JK flip-flop is cleared. Which one of the following is the bit sequence (including the initial state) generated at the Q output of the JK flip-flop when the flip-flops are connected to a free-running common clock? Assume that J = K = 1 is the toggle mode and J = K = 0 is the state-holding

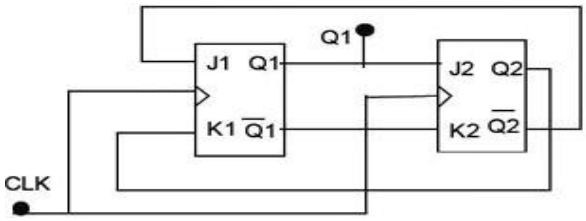
mode of the JK flip-flop. Both the flip-flops have non-zero propagation delays**.**

GATE-2015



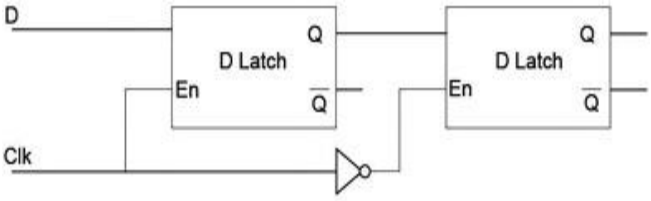
A) 0110110…
B) 0100100…
C) 011101110…
D) 011001100…

3. The outputs of the two flip-flops Q_1, Q_2 in the figure shown are initialized to 0, 0. The sequence generated at Q_1 upon application of clock signal is **GATE-2014**



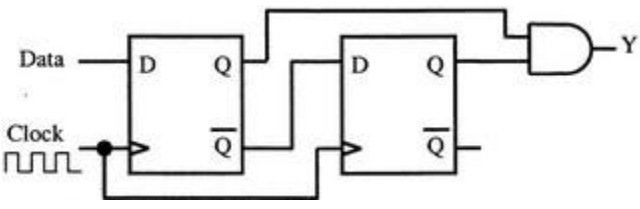
A) 01110…
B) 01010…
C) 00110…
D) 01100…

4. The circuit shown in the figure is a **GATE-2014**



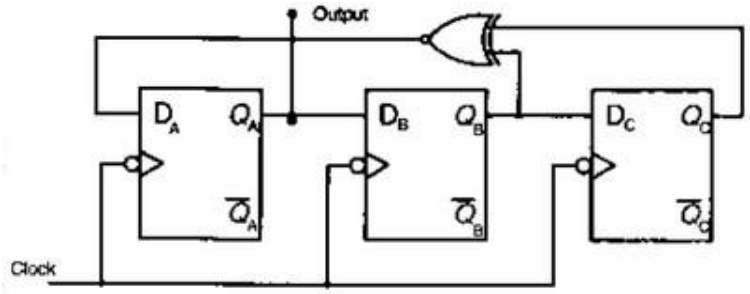
A) Toggle flip-flop
B) JK flip-flop
C) SR flip-flop
D) Master-Slave D flip-flop

5. When the output Y in the circuit below is _1' . It implies that data has **GATE-2011**



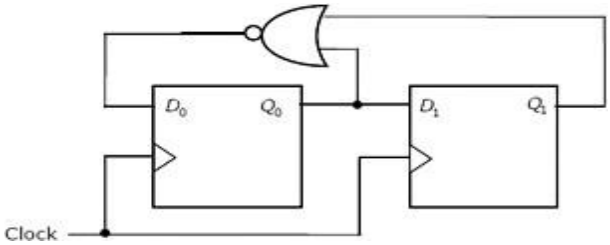
A) Changed from _0‘ to _1‘
B) Changed from _1‘ to _0‘
C) Changed in either direction
D) Not changed

6. Assuming that the all flip-flops are in reset condition initially, the count sequence observed at Q_A in the circuit shown is **GATE-2010**



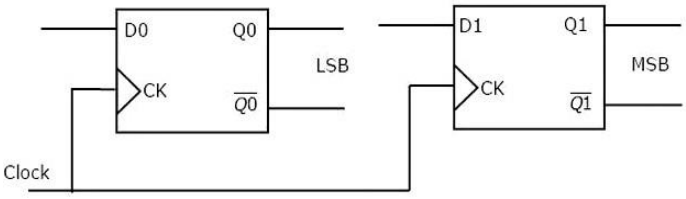
A) 0010111….
B) 0001011….
C) 0101111….
D) 0110100….

7. For the circuit shown, the counter state (Q_1Q_0) follows the sequences **GATE-2007**

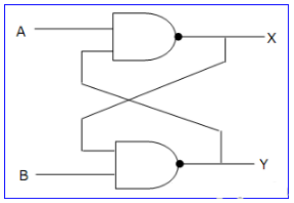


A) 00,01,10,11,00,……
B) 00,01,10,00,01,……
C) 00,01,11,00,01,……
D) 00,10,11,00,10,……

8. Two D flip-flops are to be connected as a synchronous counter as shown below, that goes through the following Q_1 Q_0 sequence 00→01→11→10→00→……
The inputs D_0 and D_1 respectively should be connected as **GATE-2006**

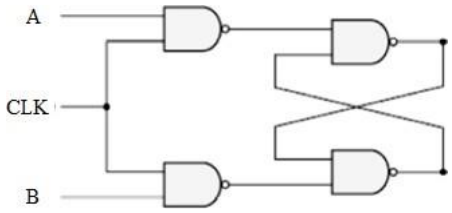


A) $\overline{Q_1}$ and Q_0 B) $\overline{Q_0}$ and Q_1 C) $\overline{Q_1}$ $Q_0$and Q_0 D) $\overline{Q_1}$ $Q_0$and Q_1 Q_0

9. The present output Q_n of an  Flop is logic _0‘. If J=1,then Q_{n+1} is
A) Can‘t determined **GATE-2005**

B) Will be logic‗1‘
C) Will be logic‗0‘
D) Will race around

10. Consider the given circuit. In the circuit race around condition will **GATE-2005**

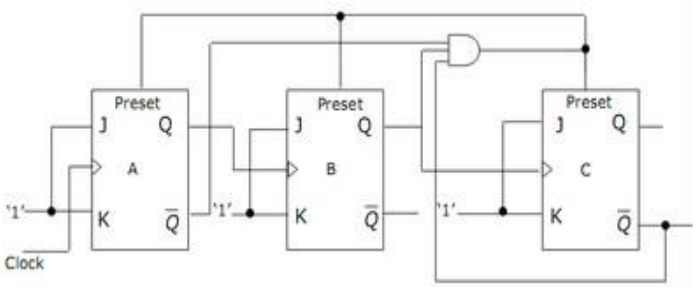


A) Does not occur
B) Occur when CLK=0
C) Occur when CLK=1 and A=B=1
D) Occur when CLK=1 and A=B=0

11. A Master-Slave flip-flop has the characteristic that **GATE -2004**

A) Change in the input immediately reflected in the output.
B) Change in the output occurs when the state of the master is affected.
C) Change in the output occurs when the state of the slave is affected.
D) Both the master and slave states are affected at the same time.

12. The ripple counter shown in the given figure is works as a **GATE -1999**



A) Mod-3 up counter
B) Mod-5 up counter
C) Mod-3 down counter
D) Mod-5 down counter

13. In the figure shown is A=1 and B=1, the input B is now replaced with a sequence 101010…., the output X and Y will be **IES-2005**

A) Fixed at 0 and 1 respectively
B) X = 1010... while Y = 0101....
C) X = 1010.... and Y = 1010....
D) Fixed at 1 and 0 respectively

14. A Master Slave flip flop has the characteristic that **IES-2001**

A) Change in input immediately reflected in the output
B) Change in the output occurs when the state of the Master is affected
C) Change in the output occurs when the state of the Slaver is affected
D) Both the master and the slave states are affected at the same time