# 3 Sequential Circuit

## 3.1 BASICS OF SEQUENTIAL CIRCUITS

In a sequential circuit, output at any instant is dependent on the present input as well as the previous state of the circuit.

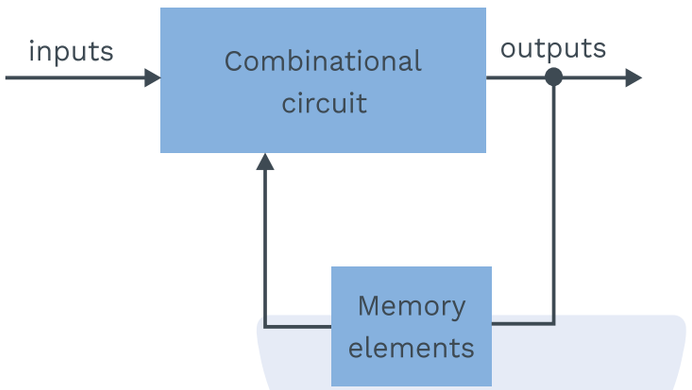



**Fig. 3.1 Block Diagram of a Sequential Circuit**

| Combinational circuits | Sequential circuits |
|---|---|
| **1)** The output at an instant depends only on the present input. | **1)** Output at an instant of time is dependent on the present input as well as the previous state of the circuit. |
| **2)** Memory units are not required in combinational circuits. | **2)** Memory units are required to store the past history of the system. |
| **3)** Combinational circuits are faster than sequential circuit as delay exists between the input and the output, which is due to the propagation delay of the logic gates. | **3)** Sequential circuits are slower than combinational circuits. |

**Table 1.1 Comparison Between Combinational and Sequential Circuit**

**Synchronous sequential circuit:**

- In synchronous circuits, clocked flip-flops are the memory elements.
- The change in input signals affects memory elements upon activation of a clock signal.
- Operating speed of the clock is directly proportional to the time required by the single memory element in the circuit.

**Asynchronous sequential circuit:**

- Memory elements of the asynchronous circuits are either unclocked Flip-flops (external clock is not given, the output of flip-flops is given as clock to next flip-flop) or time delay units.
- In asynchronous circuits, any changes in the input signals may affect memory elements at any point of time.

## 3.2 FLIP-FLOP

**Latch:**

**Definition**

The term Latch is used for certain flip-flops. It refers to non-clocked flip-flops because these flip-flops 'Latch on' to '1' or '0' immediately upon receiving the inputs pulse called SET or RESET. They are not dependent on clock signal for their operation. Clocked flip-flops are latches which respond to the inputs and latch on to 1 or 0 only when they are enabled.
On the other hand a flip-flop is a sequential device that normally samples it's inputs and changes it's outputs only at times determined by clock pulses.

**S-R Flip-flop:**

Figure 3.2 a) shows a logic diagram of edge triggered S-R flip-flop and Figure 3.2 b) shows a logic symbol of SR flip-flop. The characteristic table and excitation table of S-R flip-flop are shown in Figure 3.2 c) and Figure 3.2 d), respectively.

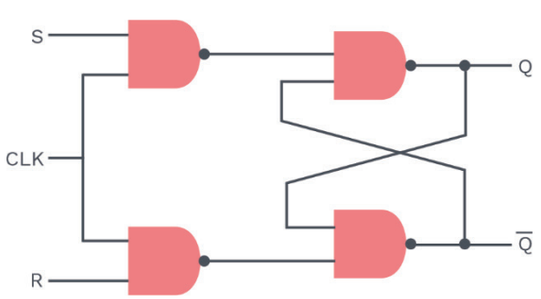



Fig. 3.2 a) Logic Diagram

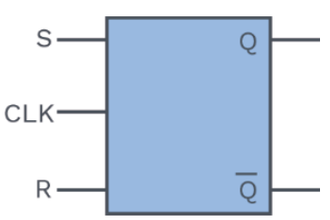



Fig. 3.2 b) Logic Symbol

| CLK | S | R | $Q_N$ | $Q_{N+1}$ | |
|---|---|---|---|---|---|
| ↑ | 0 | 0 | 0 | 0 | Latch Mode |
| ↑ | 0 | 0 | 1 | 1 | |
| ↑ | 0 | 1 | 0 | 0 | Reset |
| ↑ | 0 | 1 | 1 | 0 | |
| ↑ | 1 | 0 | 0 | 1 | SET |
| ↑ | 1 | 0 | 1 | 1 | |
| ↑ | 1 | 1 | 0 | x | Invalid |
| ↑ | 1 | 1 | 1 | x | |

**Table 1.2 c) Characteristic Table**

| Q | $Q_{N+1}$ | S | R |
|---|---|---|---|
| 0 | 0 | 0 | x |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | x | 0 |

**Table 1.3 d) Excitation Table**

**Characteristic equation:** $Q_{n+1} = S + Q_n R'$

**J-K flip-flop:**

J-K flip-flop is similar to the S-R flip-flop; the only difference is that it does not have any invalid state like in S-R flip-flop.

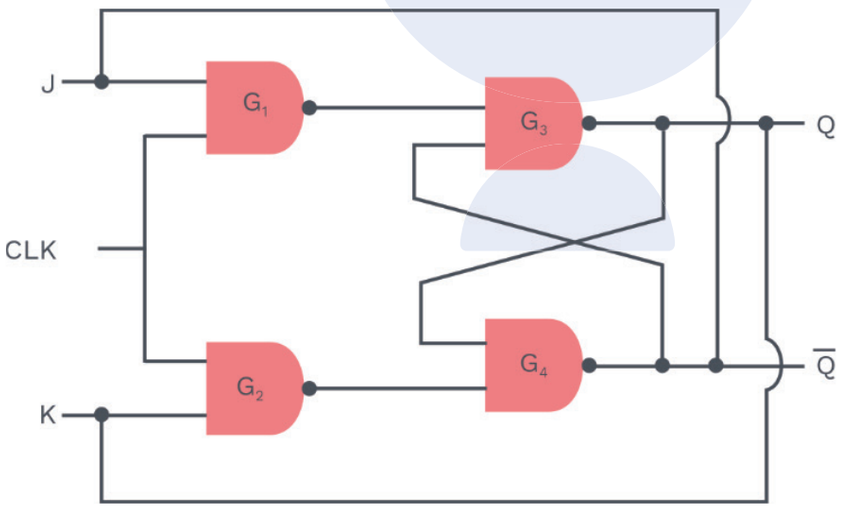



**Fig. 3.3 Logic Diagram**

**Function table:**

| J | K | $Q_{N+1}$ | |
|---|---|---|---|
| 0 | 0 | $Q_N$ | → Latch/Hold |
| 0 | 1 | 0 | → Reset |
| 1 | 0 | 1 | → Set |
| 1 | 1 | $\overline{Q}_N$ | → Toggle |

**Table 1.2 Functional Table of JK Flip-Flop**

**Characteristic table:**

| J | K | $Q_N$ | $Q_{N+1}$ | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Latch |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | Reset |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | Set |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 1 | Toggle |
| 1 | 1 | 1 | 0 | |

**Table 1.3 Characteristic Table of JK Flip-Flop**

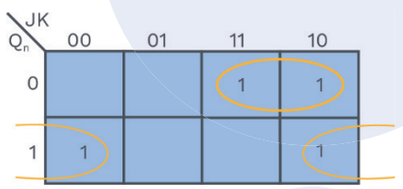



**Fig. 3.4 K-map Simplification of JK Flip-flop**

2 Prime implicants
2 essential prime implicants

$$Q_{n+1} = J\overline{Q}_n + \overline{K}\,Q_n \longrightarrow \text{Characteristic Equation}$$

$Q_{n+1}$ ⟶ Next state

$Q_n$ ⟶ Present state

**Excitation table:**

| $Q_n$ | $Q_{n+1}$ | J | K |
|---|---|---|---|
| 0 | 0 | 0 | $\phi$ |
| 0 | 1 | 1 | $\phi$ |
| 1 | 0 | $\phi$ | 1 |
| 1 | 1 | $\phi$ | 0 |

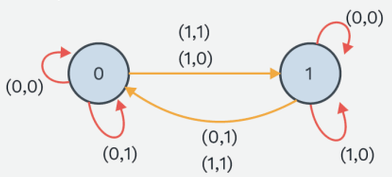
**Table 1.4 Excitation Table of JK Flip-Flop**

$\phi \rightarrow$ Don't care.

**Grey Matter Alert!**

As Flip–Flop is 1 bit memory, it can either store 0 or 1

Note: (0, 1) implies J = 0, K = 1




**State Diagram of J–K Flip–Flop**

**D-Flip-flop:**

The edge-triggered D flip-flop has one input. D-flip-flop can be obtained from a J-K flip-flop by placing an inverter between the J and K terminal.

**Function table:**

| D | $Q_{n+1}$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

**Table 1.5 Functional Table of D Flip-Flop**

**Characteristic table:**

| D | $Q_n$ | $Q_{n+1}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Table 1.6 Characteristic Table of D Flip-Flop**

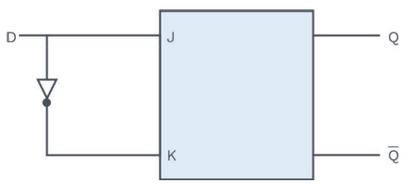



**Fig. 3.5 Representation of D Flip-Flop Using JK Flip-Flop**

**Note:**

From the characteristic table

$Q_{n+1} = D\overline{Q}_n + D\overline{Q}_n$

$Q_{n+1} = D(Q_n + \overline{Q}_n) = D$ = Characteristic Equation

**Excitation table:**

| $Q_n$ | $Q_{n+1}$ | D |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Table 1.7 Excitation Table of D Flip-Flop**

**State diagram:**

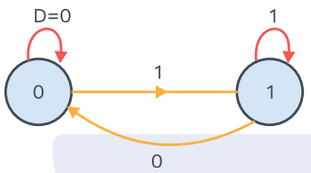


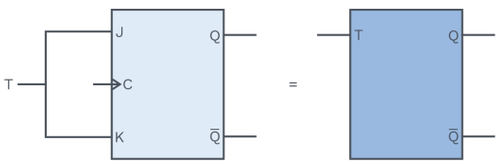

**Fig. 3.6 State Diagram of D Flip-Flop**

**T flip-flop:**

A T flip-flop has single control-input, labelled T for Toggle. When T is '1', the flip-flop toggles on every clock pulse; when T is 0, the flip-flop remains in the same state.

**Grey Matter Alert!**

It is easy to convert a J-K flip-flop to the functional equivalence of a T flip-flop by just connecting J and K together and labelling common connection as 'T'.




(a) T Flip–flop from J–K flip flop

(b) Logic Symbol

| C | T | $Q_N$ | $Q_{N+1}$ | State |
|---|---|---|---|---|
| ↑ | 0 | 0 | 0 | No change |
| ↑ | 0 | 1 | 1 | |
| ↑ | 1 | 0 | 1 | Toggle |
| ↑ | 1 | 1 | 0 | |
| 0 | x | 0 | 0 | No change |
| 0 | x | 1 | 1 | |

(c) Truth table

**Fig. 3.7 Edge Triggered T Flip-Flop**

**Function table:**

| T | $Q_{n+1}$ |
|---|---|
| 0 | $Q_n$ |
| 1 | $\overline{Q_n}$ |

**Table 1.9 Functional Table of T Flip-Flop**

**Characteristic table:**

| T $Q_n$ | $Q_{n+1}$ |
|---|---|
| 0 0 | 0 |
| 0 1 | 1 |
| 1 0 | 1 |
| 1 1 | 0 |

**Table 1.10 Characteristic Table of T Flip-Flop**

**Characteristic equation:**

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

$$\boxed{Q_{n+1} = T \oplus Q_n}$$

**Excitation table:**

| $Q_n \rightarrow Q_{n+1}$ | T |
|---|---|
| 0 0 | 0 |
| 0 1 | 1 |
| 1 0 | 1 |
| 1 1 | 0 |

**Table 1.11 Excitation Table of T Flip-Flop**

**State diagram:**

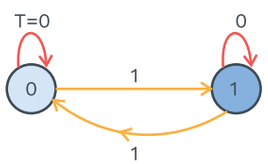



**Fig. 3.8 State Diagram of T Flip-Flop**

**Triggering:**

**Definition**

The momentary change in control of the latch or flip-flop to switch it from one state to the other is called a trigger, and the transition it causes is said to trigger the flip-flop. The process of applying the control of applying the control signal to change the state of Flip-flop is called triggering.

**There are two types of triggering:**

**i)** Level triggering

**ii)** Edge triggering

In level triggering, flip-flop changes its state when the clock is at logic 1 level.

In edge triggering, the input signals affect the Flip-flop if they are present at the positive going or negative going edge of the clock pulse.

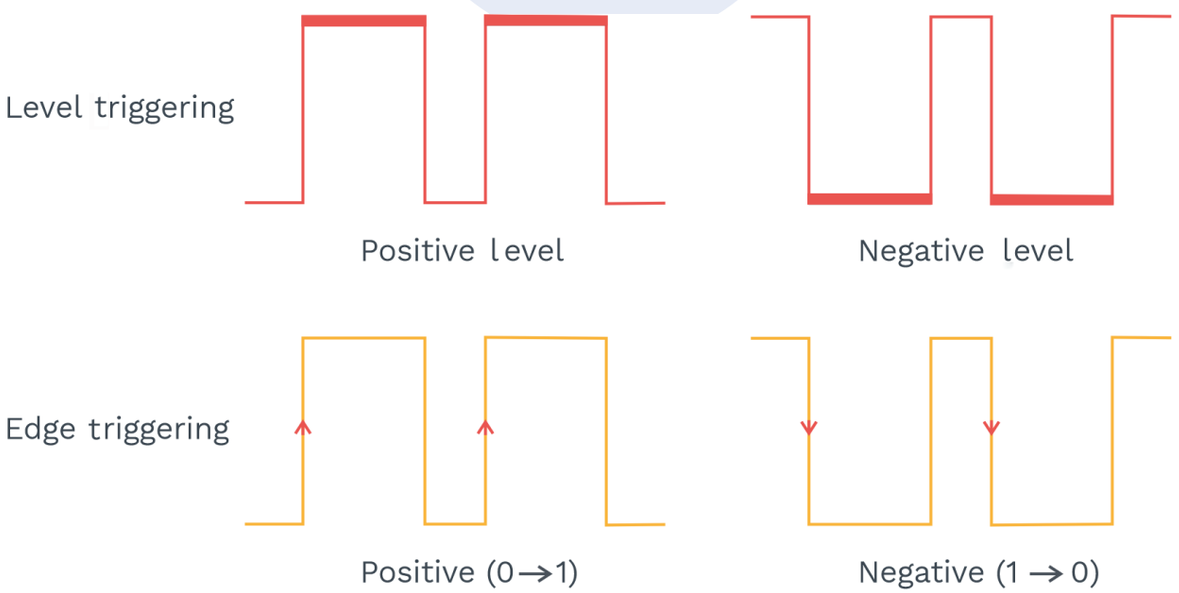



**Fig. 3.9 Waveform of Clock Pulse**

**Race around condition:**

For the J-K flip-flop shown in Figure 3.5, consider the assignment of excitations J=K=1. If the clock pulse is high for too long, the output of the flip-flop will keep fluctuating between '0' and '1', and at the end of the clock, its state will become uncertain. This is known as a race around the condition.

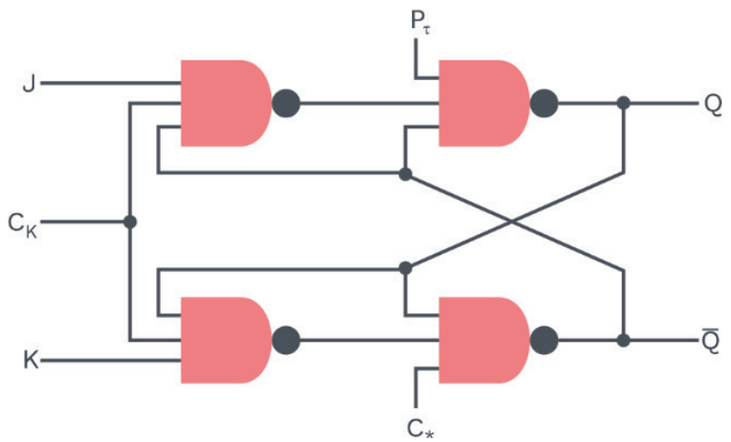



**Fig.3.10 Level Triggering Flip-Flop**

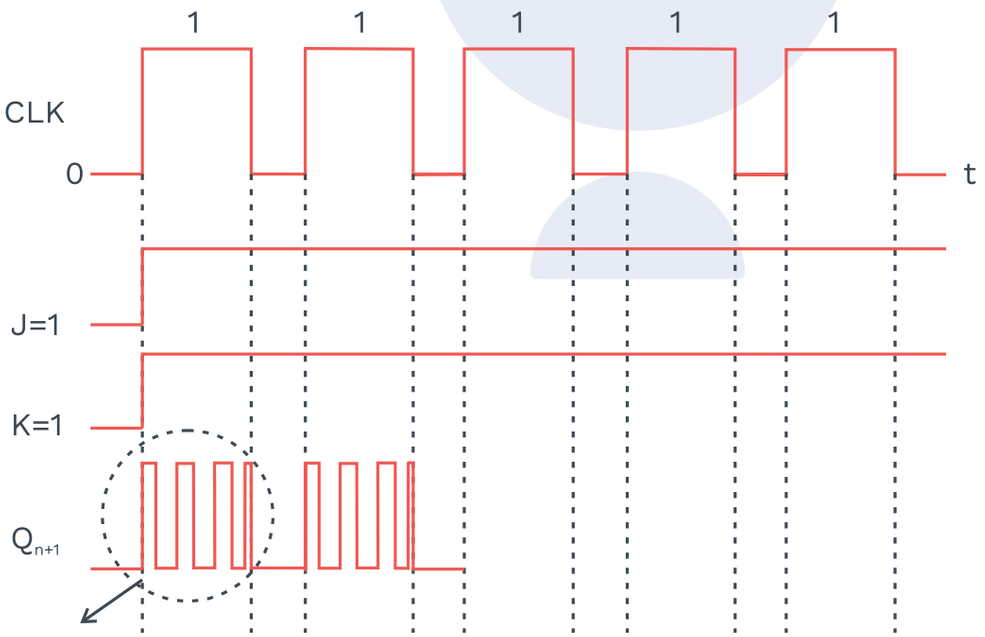



**Fig. 3.11 Waveform of JK Flip-Flop**

**Conditions to avoid racing:**

**i)** Edge triggering JK flip-flop

**ii)** Master slave JK flip-flop

**Master-slave S-R flip-flop:**

The truth table is the same as that of the edge-triggered S-R flip-flop, except for the way it is clocked.

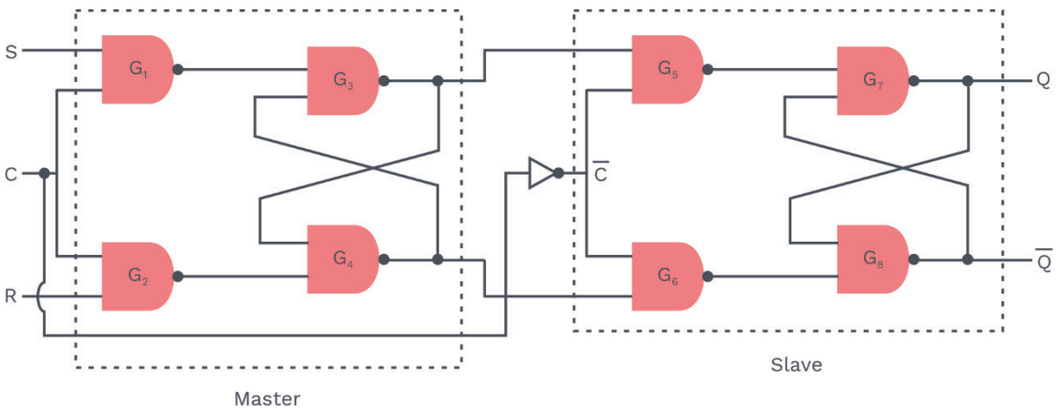



**Fig. 3.12 Logic Diagram of Master-Slave S-R Flip-Flop**

| Inputs | | | Output | Comments |
|---|---|---|---|---|
| S | R | CLK | $Q_N$ | |
| 0 | 0 | ⎍ | $Q_{n-1}$ | Latch |
| 0 | 1 | ⎍ | 0 | Reset |
| 1 | 0 | ⎍ | 1 | Set |
| 1 | 1 | ⎍ | x | Invalid |

**Table 1.12 Truth Table of Master-Slave S-R Flip-Flop**

**Conversion between flip-flops:**

**1)** The characteristic table of the target flip-flop is obtained.

**2)** The next state in the characteristic table (obtained in step 1) is replaced using the excitation table of the given flip-flop.

**3)** The expression for the i/p of the given flip-flop is then realised.

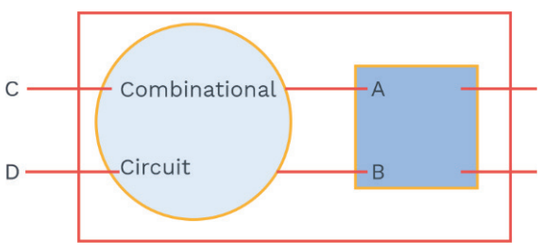



**Fig. 3.13**

## SOLVED EXAMPLES

**Q1** **Convert S–R flip-flop to J–K flip-flop.**

**Sol:** Step 1: Characteristic table of J-K flip-flop.

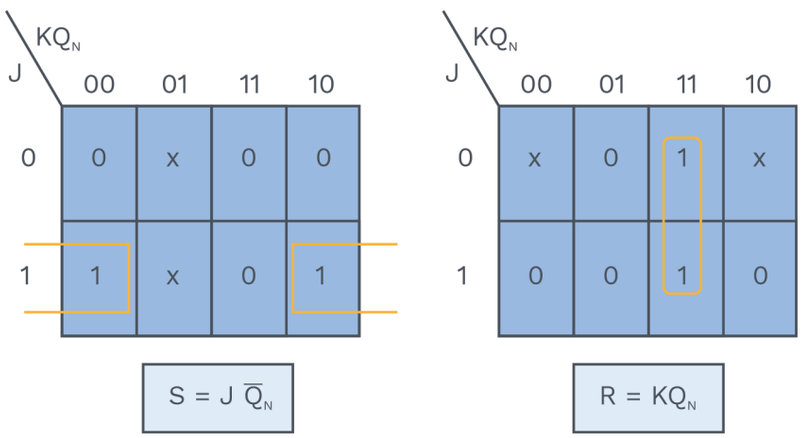
| J | K | $Q_N$ | $Q_{N+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Step 2: Replace the next state using the excitation table of SR flip-flop

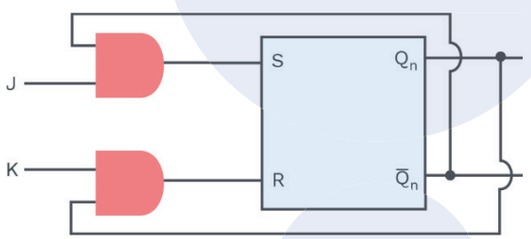
| J | K | $Q_N$ | $Q_{N+1}$ | S | R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | x |
| 0 | 0 | 1 | 1 | x | 0 |
| 0 | 1 | 0 | 0 | 0 | x |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | x | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

where ‘x’ is don’t care

Step 3: The expression for the input of a given flip-flop is obtained and realised.




**Realisation**

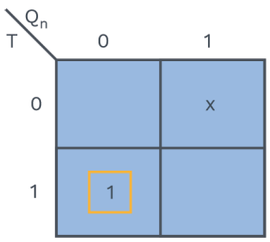



**Q2** **Convert SR flip-flop to T flip-flop.**

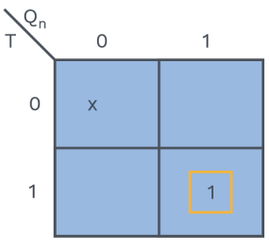
**Sol:** Following is the characteristic table of T flip-flop and excitation table of S-R flip-flop.

| T | $Q_n$ | $Q_{n+1}$ | S | R |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | x |
| 0 | 1 | 1 | x | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

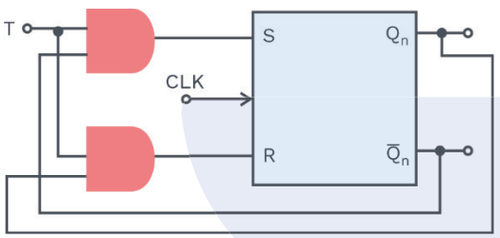
Step 3:




$S = T\,\overline{Q}_N$



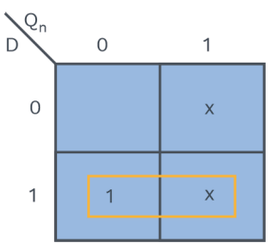

$R = T\,Q_N$



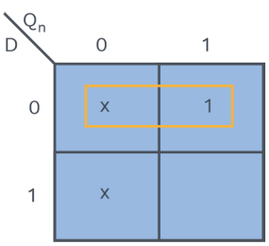

**Q3** **Convert J–K flip-flop to D flip-flop.**

**Sol:** Let's construct the characteristic table of D flip-flop and replace next state with the excitation table of J-K flip-flop.

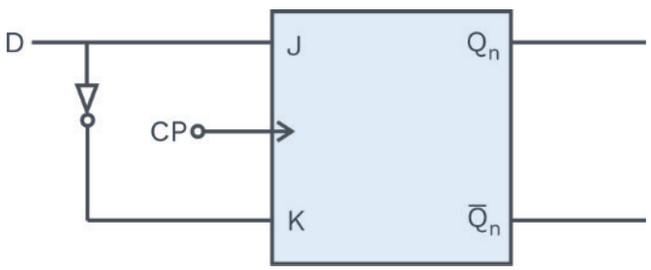
| D | $Q_N$ | $Q_{N+1}$ | J | K |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | x |
| 0 | 1 | 0 | x | 1 |
| 1 | 0 | 1 | 1 | x |
| 1 | 1 | 1 | x | 0 |




$J = D$

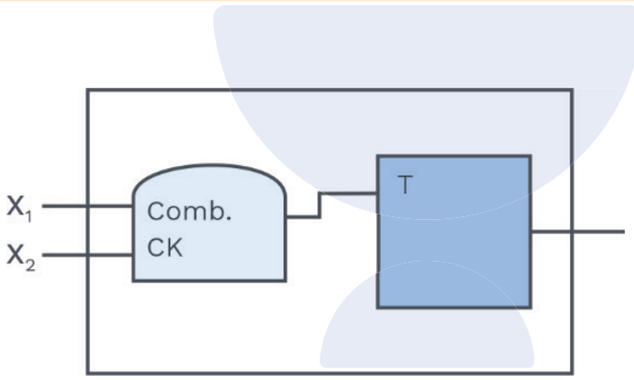



$K = \overline{D}$




**Q4** **A new Flip-flop $x_1$, $x_2$ has characteristic equation $Q_n = \overline{x}_1 \overline{Q} + \overline{x}_2 Q$. Realise it using T–flip-flop.**

**Sol:**




**Characteristic equation:**

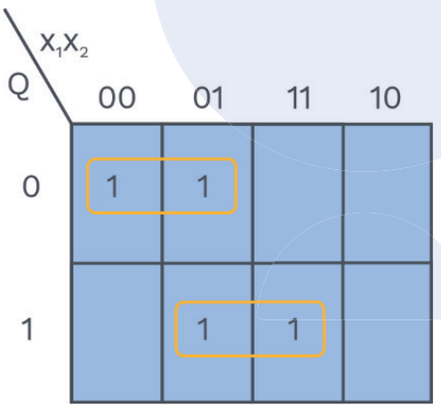
$$Q_n = \overline{x_1}\,\overline{Q} + \overline{x_2}\,Q$$

Excitation table for T flip-flop.

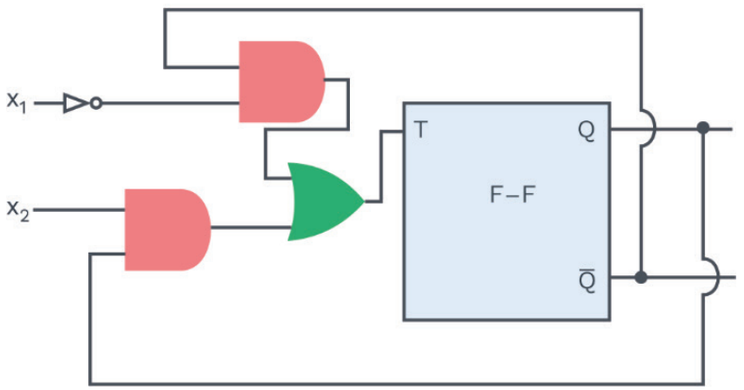
| Q | $Q_N$ | T |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

From the characteristic equation of new flipflop and excitation table of the given flip-flop

| $x_1$ | $x_2$ | Q | $Q_N$ | T |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

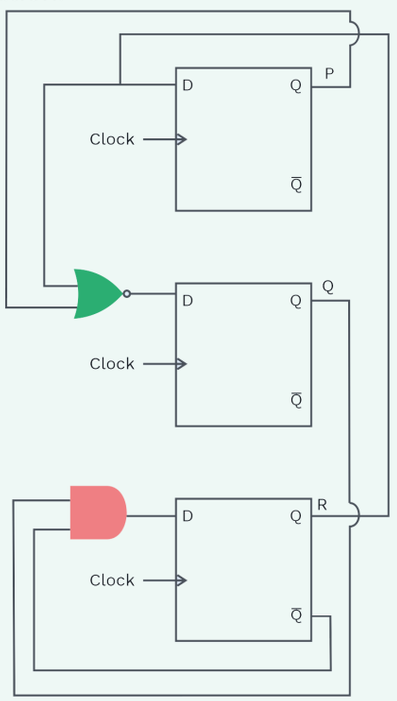



$$T = \overline{x_1}\,\overline{Q} + x_2\,Q$$



**Previous Years' Question**

Consider the following circuit involving three D–type Flip-flops used in a certain type of counter configuration. **(GATE-2011)**




If at some instance prior to the occurrence of the clock edge, P, Q and R have value 0, 1 and 0 respectively, what shall be the value of PQR after the clock edge?

**a)** 000 **b)** 001
**c)** 010 **d)** 011

**Sol: d)**

## 3.3 REGISTERS

A flip-flop can only store 1 bit of data, either '0' or '1'; it is also called single bit register. The more is the number of bits to be stored, the more number of flip-flops are used. A register is a group of flip-flops that is used to store digital data. The storage capacity of the register depends on the size(number of bits) of the digital data it can store. The process of setting and resetting registers is known as loading. Loading can be serial or parallel.

**Serial in serial out (SISO) shift register:**

SISO shift registers accept data as input serially, i.e. one bit at a time and also output digital data serially. The logic diagram of 4 bit serial-in,

serial-out shift register is shown in figure 3.6. With four stages, i.e. four flip-flops, the register can store up to 4 bits of data. Serial data is applied at the $D_1$ input of the first flip-flop. The $Q_1$ output of the first flip-flop is connected to the $D_2$ input of the second flip-flop and the $Q_2$ output of the second flip-flop is connected to the $D_3$ input of the third flip-flop and the same happens for the last flip-flop. The $Q_4$ terminal of the last flip-flop outputs the data.

**Grey Matter Alert!**

When serial data is transferred into a register, each new bit is clocked into the first flip-flop at the postive-going edge of each clock pulse. The bit that was previously stored by the first flip-flop is transferred to the second flip-flop and so on. The bit that was stored by the last flip-flop is shifted out.

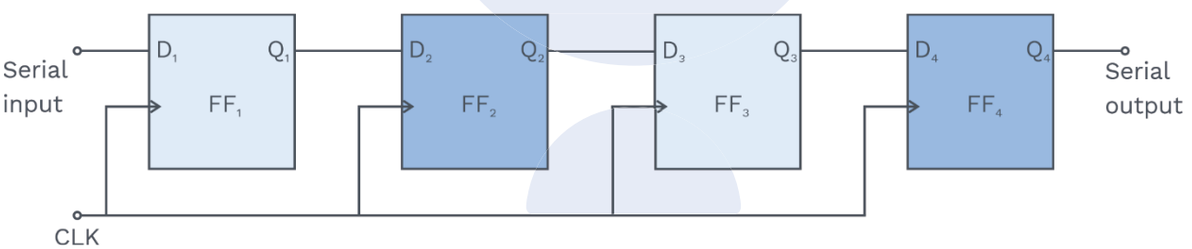



**Fig. 3.14 4-Bit Serial-In, Serial-Out Shift-Register**

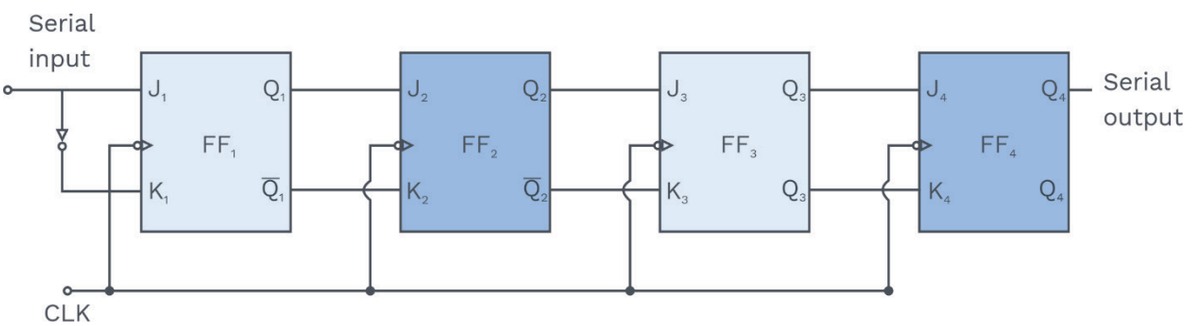



**Fig. 3.15 A 4 Bit Serial-In, Serial-Out Shift-Register Using JK FF**

**Serial-in, parallel-out shift register:**

Figure 3.8 shows the diagram and the logic symbol of a 4-bit serial-in, parallel-out, shift register. In SIPO registers, data are fed into the registers serially and register shifts output the data in parallel form.

Once the data bits are stored, all bits are available simultaneously.

**Grey Matter Alert!**

The serial–IN, Parallel–OUT shift register can be used as a serial–IN, serial–OUT shift register if the output is taken from the Q terminal of the last Flip-flop.

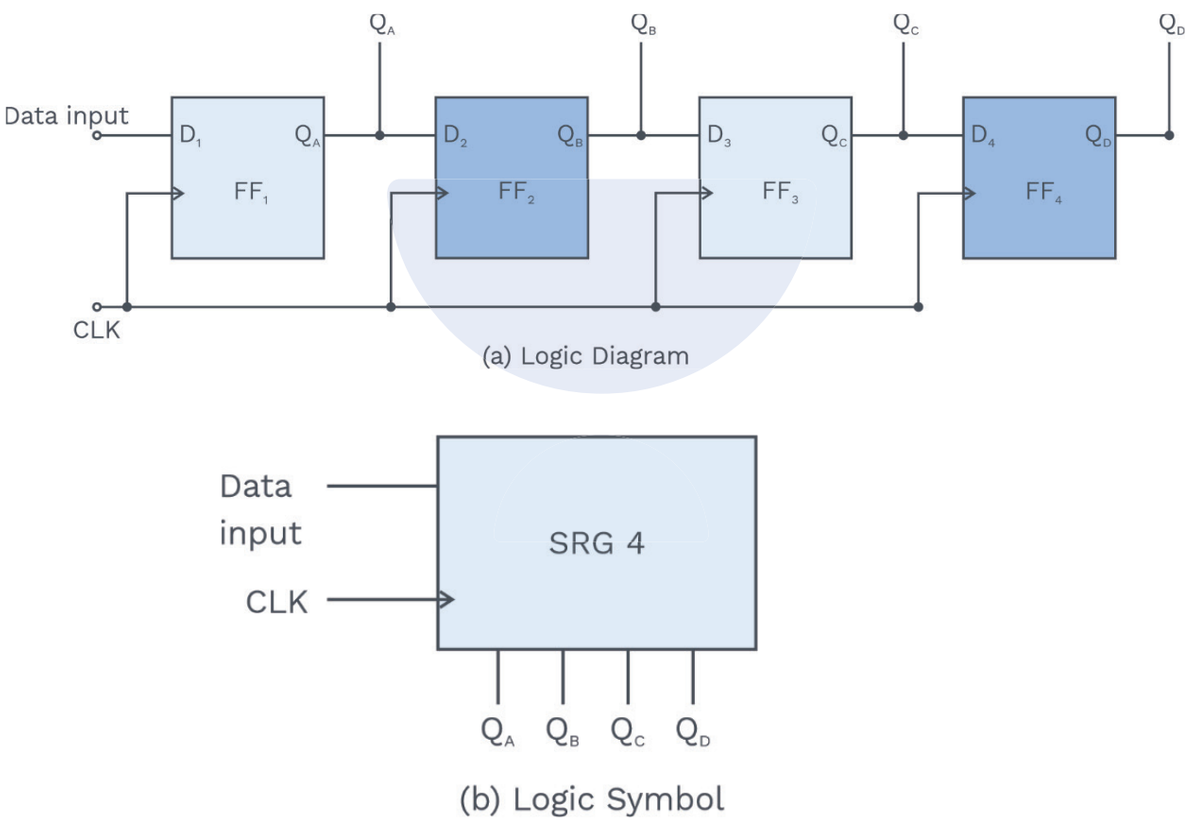



**Fig. 3.16 A 4 Bit Serial-IN Parallel-OUT Shift Register**

**Parallel IN parallel out (PIPO):**

In PIPO, shift register input and output of the data is in a parallel way.

Figure 3.9 shows a 4-bit parallel-in, parallel-out shift register using D flip-flops. When clock pulse is applied, and data is applied to the 'D' i/p terminals of the flip-flops, the D i/ps are shifted in the output terminal of the flip-flops.

The stored data is outputted instantaneously in parallel form.

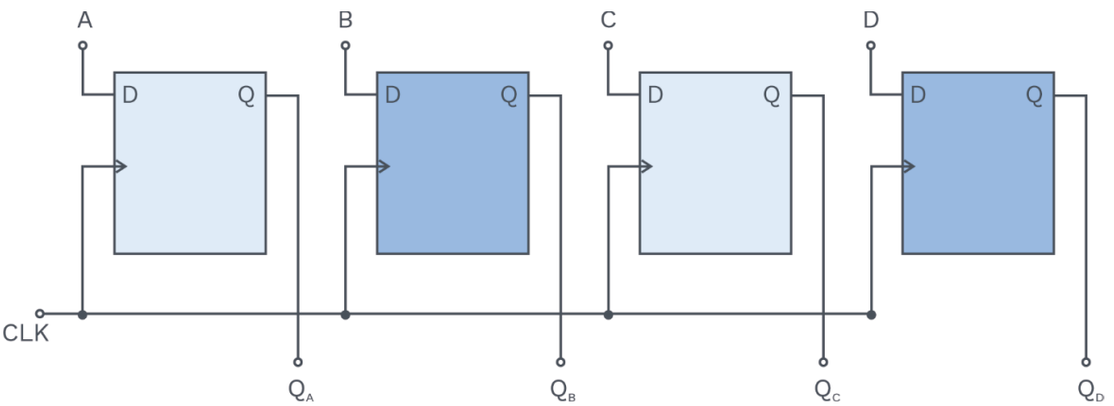
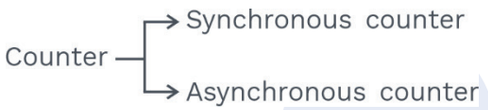



**Fig. 3.16 PIPO Shift Register**

## 3.4 COUNTER

**Introduction:**

A digital counter is a group of Flip-flops that changes its state in response to the clock applied to the counter. The flip-flops are interconnected in such a way that their combined state at any time is the binary equivalent of the total number of pulses that have occurred till that moment.




Another name for the asynchronous counter is ripple counter.

**Grey Matter Alert!**

In ripple counter, the flip-flops within the counter are not made to change the states exactly at the same time. This is because the flip-flops within the counter are not triggered simultaneously. An asynchronous counter uses T flip-flops usually to perform a counting function.

In the asynchronous counter, the output of the first stage acts as a clock to the second stage. The output of the second stage acts as a clock to 3rd stage and so on.

Synchronous counters are clocked such that each of the flip-flops in the counter is triggered at the same time.

**Grey Matter Alert!**

Synchronous counters are faster than asynchronous counters because the propagation delay involved is less.

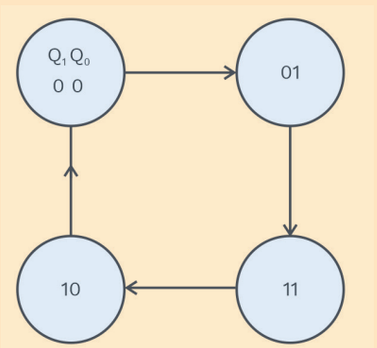
**Counter design:**

**1)** From the state diagram, get the state table.

**2)** Identify the flip-flop to be used and replace the concerned next state using the excitation table of the associated flip-flop.

**3)** Get the expressions for inputs and realise them.

**Q5** **Design a synchronous for the following state diagram using T – FFs**




**Sol:** Number of states = 4

Minimum number of flip-flops required = $\lceil \log_2 4 \rceil = 2$

**State table:**

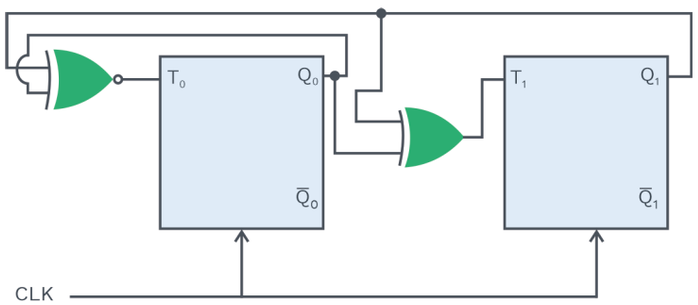
| $Q_1$ | $Q_0$ | $Q_{1N}$ | $Q_{0N}$ | $T_1$ | $T_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |

$$T_1 = \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$$

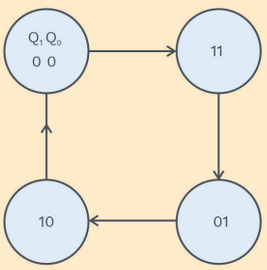
$$\boxed{T_1 = Q_1 \oplus Q_0}$$

$$T_0 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0$$

$$\boxed{T_0 = Q_1 \odot Q_0}$$




**Q6** **Consider the following state diagram which is to be designed using T-FF for MSB (Most Significant Bit) and an unknown flip-flop 'XY' for LSB (Least Significant Bit). The behaviour of XY flip-flop is shown below. Design a synchronous counter.**




| X | Y | $Q_N$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | Q |
| 1 | 0 | $\bar{Q}$ |
| 1 | 1 | 1 |

**Sol:** Characteristic table for XY flip-flop from the given function table

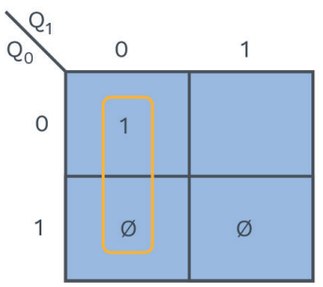
| X | Y | Q | $Q_N$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

From the state diagram:

| $Q_1$ | $Q_0$ | $Q_{1N}$ | $Q_{0N}$ | T | x | y |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | Ø |
| 1 | 1 | 0 | 1 | 1 | Ø | 1 |
| 0 | 1 | 1 | 0 | 1 | Ø | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | Ø |

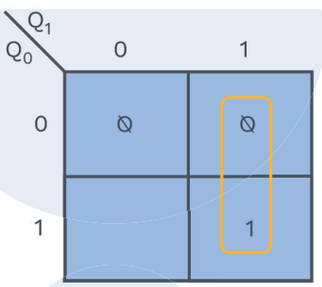
Ø ⟶ Don't care

for x :

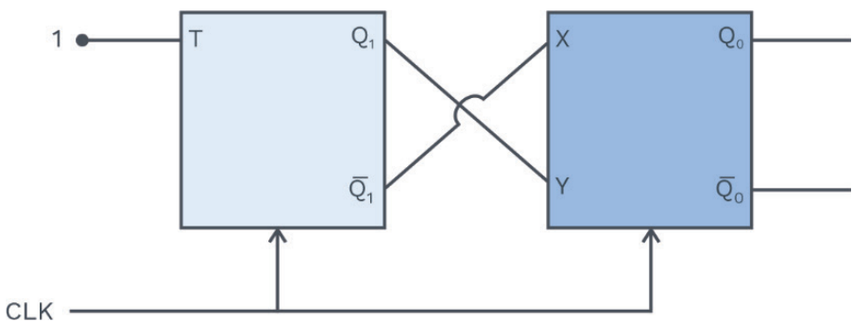



$x = \overline{Q}_1$

for y :



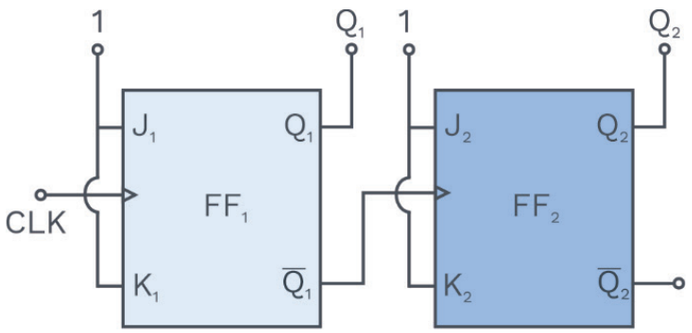

$Y = Q_1$




**Asynchronous counter:**

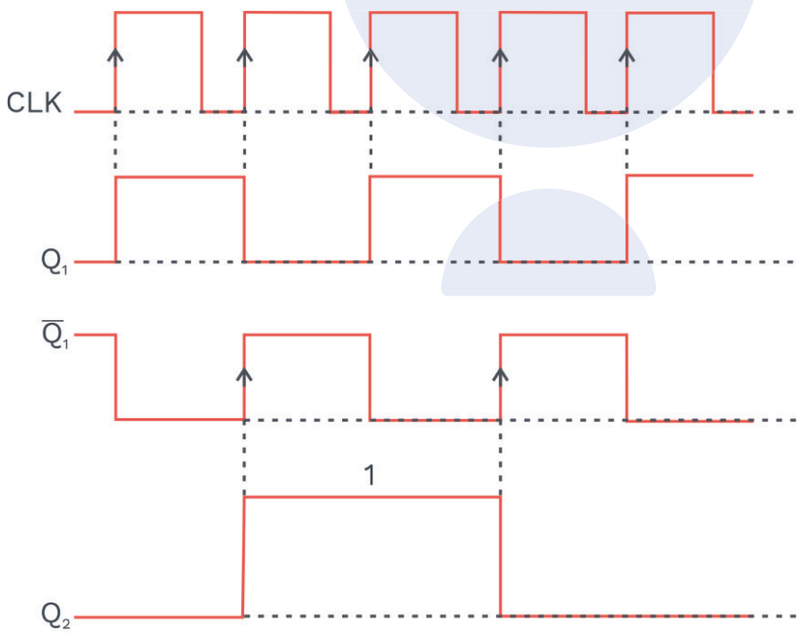
**Two-bit ripple up-counter using positive edge-triggered flip-flops:**

A 2-bit ripple up-counter, using (positive edge-triggered) J-K Flip-flops, are shown in figure 3.10. The $\overline{Q}_1$ output of the first flip-flop is connected to the clock of $FF_2$. The external clock signal is applied to the first flip-flop $FF_1$. The $FF_1$ toggles (changes its state) at the positive edge of each clock pulse, and $FF_2$ toggles when $\overline{Q}$ changes from '0' to '1'.




a) Logic Diagram

**Fig. 3.17 Asynchronous 2-bit up Counter**




b) Timing Diagram

**Fig. 3.18**

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

A 2-bit ripple down-counter using positive edge-triggered J-K FFs is shown in figure 3.11. The $Q_1$ output of the first flip-flop is connected to the clock of $FF_2$. The external clock pulse is directly applied to $FF_1$. The $FF_1$ toggles at the positive-going edge of each clock pulse.

The counting sequence is 00, 11, 10, 01, 00, 11, 10 ..... etc.

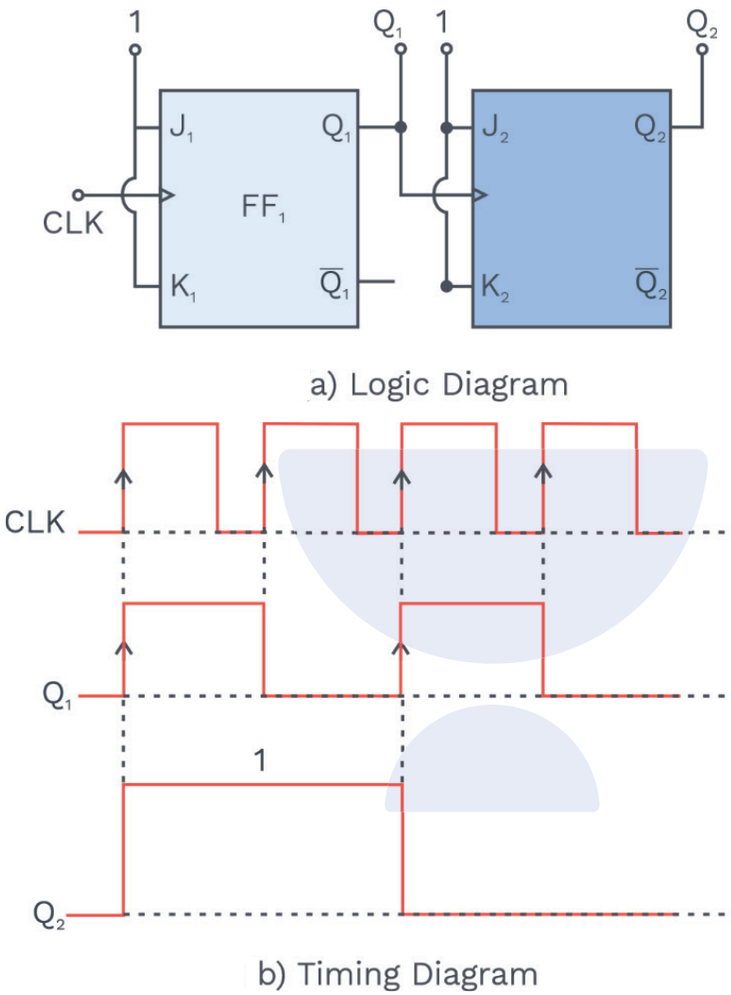



**Fig. 3.19 2-bit Ripple Up Counter Design**

**Mod-8 up-counter:**

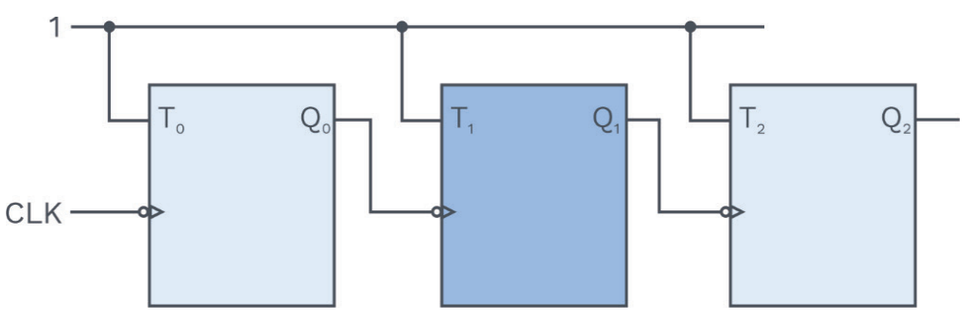



**Fig. 3.20 Mod-8 Up Counter**

All the flip-flops are negative edge triggered, i.e. changes in their state when the clock makes a transition from 1 to 0.

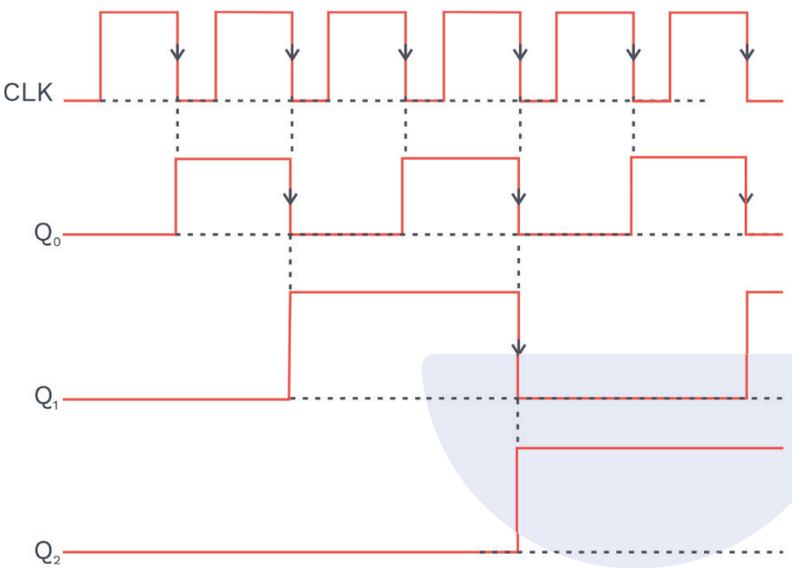



**Fig. 3.21 Timing Diagram of Mod-8 up Counter**

Time period of $Q_0 = 2 \times T_{CLK}$

Frequency, $f_{Q0} = \frac{1}{2} f_{CLK}$

Time period $Q_1 = 2 \times T_{Q0}$

$$f_{Q1} = \frac{1}{2} f_{Q0} = \frac{1}{4} f_{CLK}$$

Time period $Q_2 = 2 \times T_{Q1}$

$$f_{Q2} = \frac{1}{2} f_{Q1}$$

$$= \frac{1}{4} f_{Q0}$$

$$= \frac{1}{8} f_{CLK}$$

In the state Transition table,

$Q_{0N} = \overline{Q}_0$, for every clock

$Q_{1N} = \overline{Q}_1$ {When $Q_0 : 1 \rightarrow 0$}

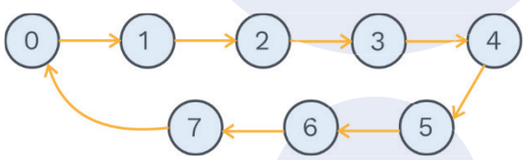
$Q_{2N} = \overline{Q}_2$ {When $Q_1 : 1 \rightarrow 0$}

**State transition table:**

| $Q_2$ | $Q_1$ | $Q_0$ | $Q_{2N}$ | $Q_{1N}$ | $Q_{0N}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

**Table 1.13 State Transition Table**

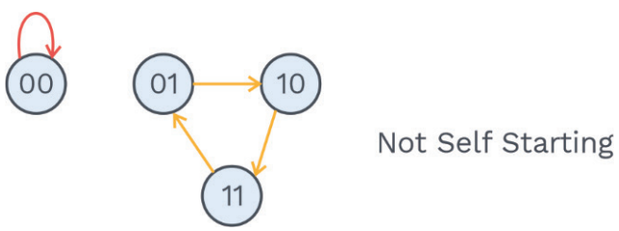
**State diagram:**



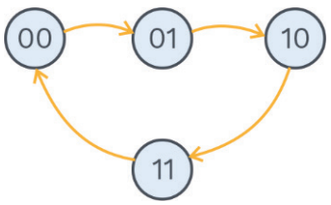

**Self starting and free running counters:**

- A counter is called self starting counter if the counter can enter counting loops irrespective of the starting state of the counter.

Example




- A counter is called to be free running counter if the counter contains all possible states in the counting loop.

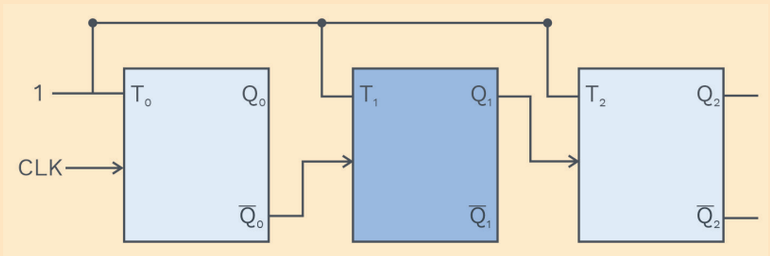



It is both self starting and free running.

**Grey Matter Alert!**

Every free running counter is self starting counter but every self starting counter is not necessarily be the free running counter.


**Q7**




**If the initial state of the counter is $Q_2\ Q_1\ Q_0$ = 101. What will be the state after 5 clock cycles?**

**Sol:** $Q_0 \rightarrow$ Present state

$Q_{0N} \rightarrow$Next state

$Q_{0N} = \overline{Q}_0$, for every clock

$Q_{1N} = \overline{Q}_1 \ \ \overline{Q}_0 : 0 \rightarrow 1$

$Q_0 : 1 \rightarrow 0$

$Q_{2N} : \overline{Q}_2, \quad Q_1 : 0 \rightarrow 1$

**State table:**

| $Q_2$ | $Q_1$ | $Q_0$ | $Q_{2N}$ | $Q_{1N}$ | $Q_{0N}$ |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |

**Table 1.14 State Table**

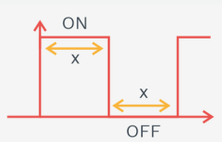
**Sol: 110**

**Cascading of ripple counters:**

Ripple counters can be cascaded to increase the modulus of the counter. A mod-M and a mod-N counter cascaded gives a mod-MN counter. While cascading, the most significant stage of the first counter, is connected to the toggling stage of the second counter. The order of cascading does not affect the frequency division; however, the duty cycle of the most significant output may depend on the order in which the counters are cascaded.

**Grey Matter Alert!**

A duty cycle is the function of one period in which a signal or system is active. It is commonly expressed in percentage.



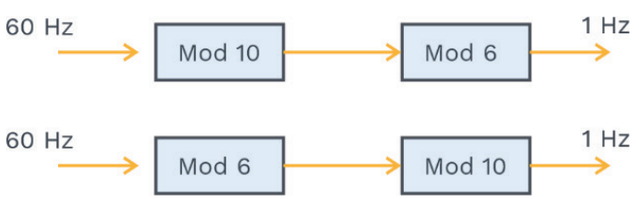




**Fig. 3.22 Examples of Cascaded Counters**

### Rack Your Brain

A binary ripple counter is required to count up to $(16, 584)_{10*}$ How many flip-flop are required? If the clock frequency is 7.940 MHz, what is the frequency at the output of the MSB?

**Synchronous counter:**
Synchronous counters are good in terms of speed and decoding, but they require more circuitry than that of asynchronous counters.

**Design of synchronous counters:**
Step 1: Find the required value of 'n' where 'n' is the number of the flip-flop. The smallest value of n is such that the number of states $N \leq 2^n$ and the desired counting sequence is to be found out.

Step 2: The state diagram is drawn, showing all the possible states.
Step 3: Choice of flip-flop and excitation table:
Select the type of flip-flops to be used and write the excitation table.

Step 4: Minimal expressions for excitations:
The minimal expression is obtained for the excitations of the flip-flop using the K-maps drawn.

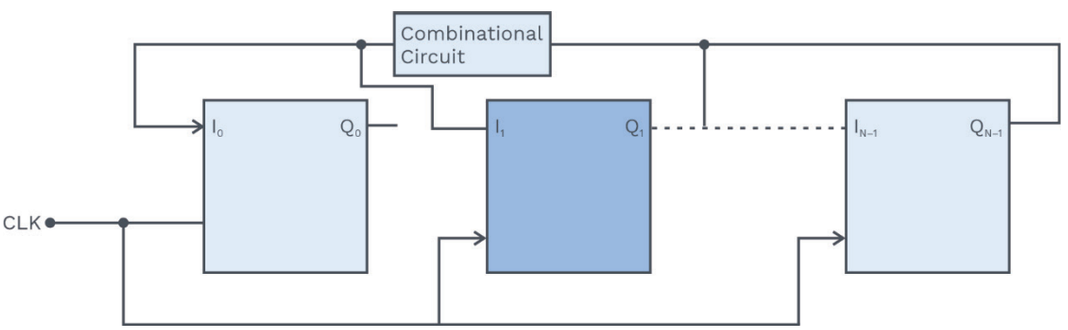



**Fig. 3.23 Synchronous Counter**

- Propagation delay in synchronous counter is:

$$T_{psyn} = T_{FF} + T_{combination}$$

$$T_{CLK} \geq T_{psyn}$$

- Propagation delay in asynchronous counter is:

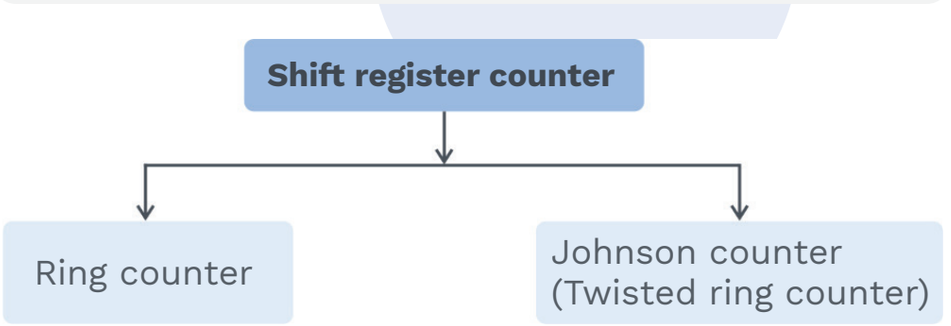
$$T_{psyn} = N \times T_{FF} + T_{combination}$$

$$T_{CLK} \geq T_{psyn}$$

**Shift register counters:**
The application of the shift register is that it can be arranged to form different types of counters.

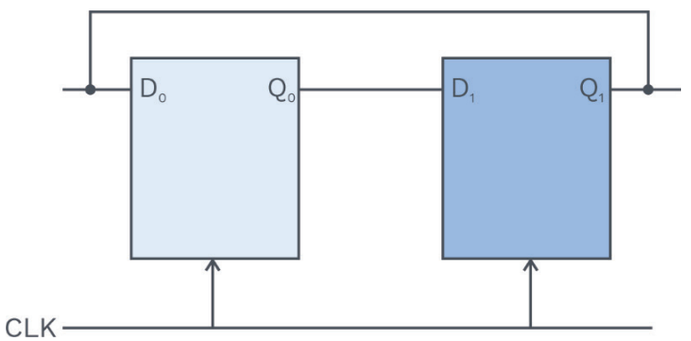
**Grey Matter Alert!**

Shift register counter are obtained from serial–in, serial–out shift register by providing feeback from the output of the last flip-flop to the input of the first flip-flop.



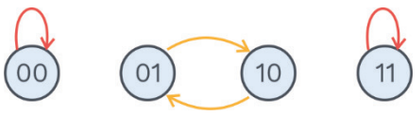

**Ring counter:**
The ring counter is the simplest shift register counter. The flip-flop is arranged the same as in a normal shift register, i.e. Q output of each stage is connected to the D input of the next stage, but the Q output of the last flip-flop is connected back to the input terminal of the first flip-flop.

**Mod 2 ring counter:**




**Mod 2 ring counter**

| $D_1$ | $D_0$ | $Q_1$ | $Q_0$ | $Q_{1N}$ | $Q_{0N}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |




**Note:**

Even though there are all four states, it is performing mod-2 count.

**Mod-3 Ring counter:**

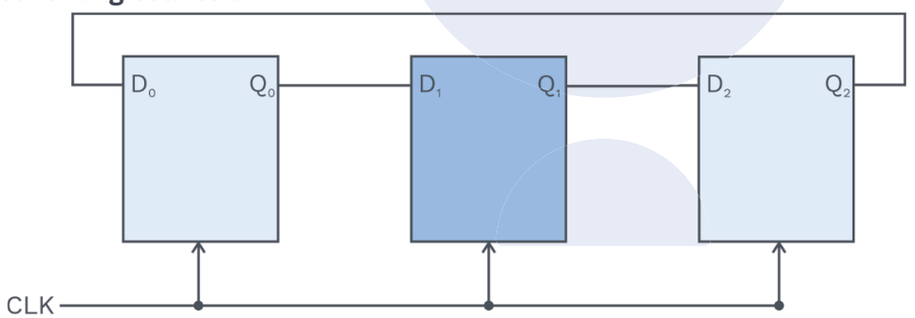
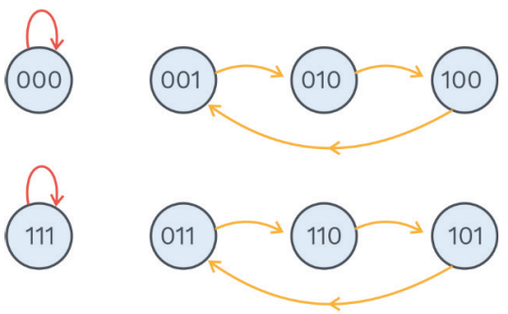



**Fig. 3.24 Mod 3 Ring Counter**

**State table:**

| $Q_2$ | $Q_1$ | $Q_0$ | $Q_{2N}$ | $Q_{1N}$ | $Q_{0N}$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |




$\therefore$ It performs mod–3 count

**Grey Matter Alert!**

The ring counter performs mod (N) count.
N $\rightarrow$ Number of FFs
Example Given mod-4 ring counter then, N = 4

**Previous Years' Question**

We want to design a synchronous counter that 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, SET-1)**

**Sol: 4**

**Twisted ring counter (johnson counter):**

This counter is designed from a SISO shift register by giving feedback from the inverted output of the last FF to the input of the first flip-flop. The output of every stage is connected to the input terminal of the next stage and so on. $\overline{Q}$ the output of the last stage is connected to the input terminal of the first stage.

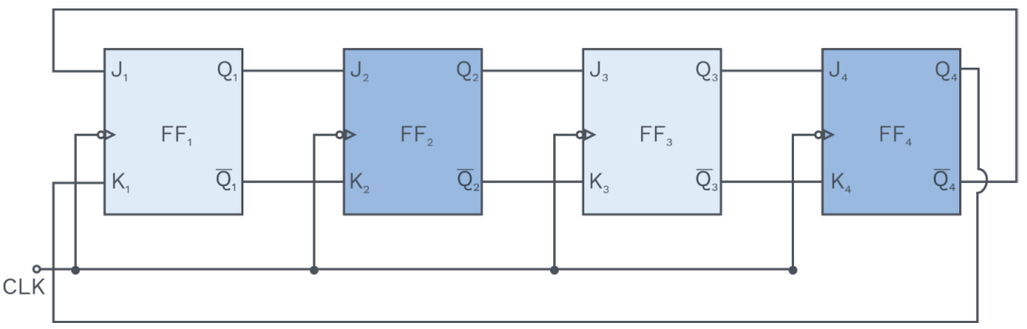



**Fig. 3.25 Logic Diagram of a 4-bit Johnson Counter Using J-K Flip-Flop**

**Mod 4 johnson counter:**

**Grey Matter Alert!**

Johnson counter performs mod (2N) count.
N → Number of FFs
Example given mod-4 ring counter then, 2N = 4 ⇒ N = 2

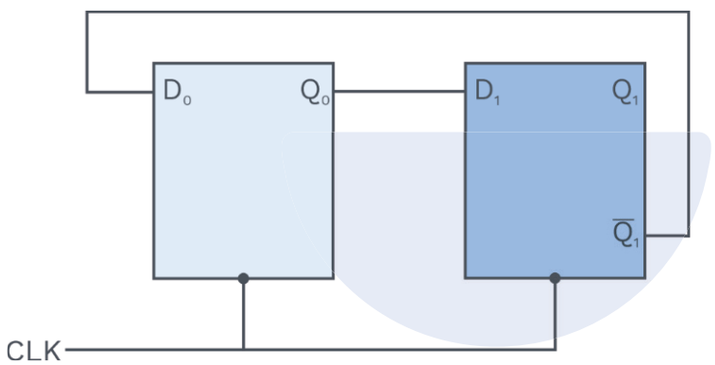


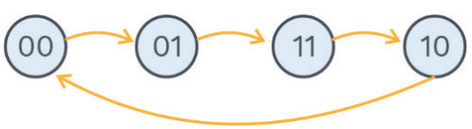

**Fig. 3.26**

**State table:**

| $Q_1$ | $Q_0$ | $Q_{1N}$ | $Q_{0N}$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

**Table 1.15**

**State diagram:**




∴ The counter performs mod-4 count.

## Chapter summary

- Sequential circuit → Output variables at any instant are dependent on the present state and past history of the system.
- Synchronous sequential circuit → Memory elements are clocked flip-flops
- Asynchronous sequential circuit → Memory elements are either unlocked flip—flops (external clock is not given, output of flip—flops is given as clock to next flip-flops) or time delay elements.
- S—R flip—flop → For S = 1, R = 1, it is a invalid input combination.
- J—K flip—flop → For J = 1, K = 1, flip-flop toggles.
  T—flip—flop → J = K = T
- Registers
  - → SISO
  - → SIPO
  - → PIPO
- Counter
  - → Synchronous counter
  - → Asynchronous counter
- Synchronous counter
  - → Ring counter ($1_O = Q_{N-1}$) [Where $Q_{N-1}$ is the output of the last flip-flop]
  - → Johnson counter ($1_O = \overline{Q}_{N-1}$)