

6**Introduction to Sequential Circuits and Flip-Flops****6.1 : Introduction**

Q.1 Draw and explain the block diagram of sequential circuit.
Ans. : Fig. Q.1.1 shows the block diagram of sequential circuit/Finite State Machine (FSM).

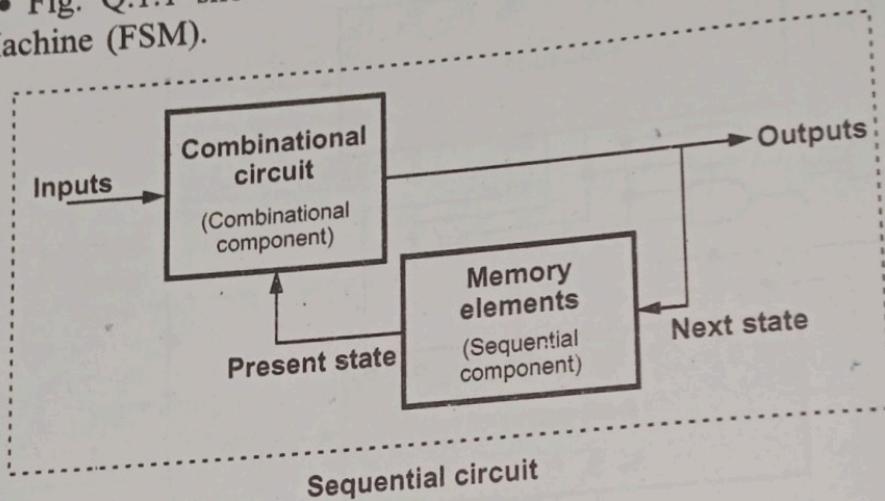


Fig. Q.1.1 Block diagram of sequential circuit / FSM

- Memory elements are connected to the combinational circuit as a feedback path.
- The information stored in the memory elements at any given time defines the **present state** of the sequential circuit.
- The present state and the external inputs determine the outputs and the **next state** of the sequential circuit.
- Thus we can specify the sequential circuit by a time sequence of external inputs, internal states (present states and next states), and outputs.
- The counters and registers are the common examples of sequential circuits.

Q.2 Differentiate between combinational logic circuits and sequential logic circuits.

☞ [SPPU : June-22, Marks 9]

Ans. :

Sr. No.	Combinational circuits	Sequential circuits
1.	In combinational circuits, the output variables are at all times dependent on the combination of input variables.	In sequential circuits, the output variables depend not only on the present input variables but they also depend upon the past history of these input variables.
2.	Memory unit is not required in combinational circuits.	Memory unit is required to store the past history of input variables in the sequential circuit.
3.	Combinational circuits are faster in speed because the delay between input and output is due to propagation delay of gates.	Sequential circuits are slower than the combinational circuits.
4.	Combinational circuits are easy to design.	Sequential circuits are comparatively harder to design.
5.	Parallel adder is a combinational circuit.	Serial adder is a sequential circuit.

Q.3 Name the two storage elements.

Ans. : 1. Latches 2. Flip-Flop.

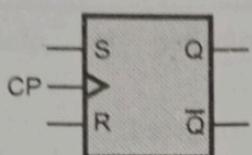
Q.4 What is flip-flop ?

Ans. : Flip-flop is a sequential circuit driven by clock input either by positive edge or negative edge. It is a binary storage device capable of storing one bit of information.

Q.5 What is level triggering and edge triggering ?

Ans. : **Level Triggering** : In the level triggering, the output state is allowed to change according to input(s) when active level (either positive or negative) is maintained at the enable input. There are two types of level triggered latches :

- **Positive level triggered** : The output of flip-flop responds to the input changes only when its enable input is 1 (HIGH).



(a) Logic symbol

CP	S	R	Q_n	Q_{n+1}	State
↑	0	0	0	0	No Change(NC)
↑	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	Indeterminate
↑	1	1	1	X	
0	X	X	0	0	No Change(NC)
0	X	X	1	1	

(b) Truth table for positive edge clocked SR flip-flop

SR	Q_n	0	1
00	0	0	1
01	0	0	
11	X	X	
10	1	1	

(c) Characteristic equation

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

Fig. Q.7.2

Case 4 : If $S = R = 1$ and the clock pulse is applied, the state of the flip-flop is undefined and therefore is indicated as indeterminate in the fourth row of the truth table.

Q.8 Draw and explain the working of D flip - flop. Give truth table and characteristic equation.

OR Draw a positive edge triggered D flip - flop using NAND gates and explain its function.

Ans. : • In SR Flip-Flop, when both inputs are same the output either does not change or it is invalid (Inputs \rightarrow 00, no change and inputs \rightarrow 11, invalid).

- These input conditions can be avoided by making them complement of each other. This modified SR flip-flop is known as D flip-flop.
- The D input goes directly to the S input, and its complement is applied to the R input. Due to these connections, only two input conditions exists, either $S = 0$ and $R = 1$ or $S = 1$ and $R = 0$.

Truth table : • The truth table for D flip-flop consider only these two conditions and it is as shown in the Fig. Q.8.1 (b).

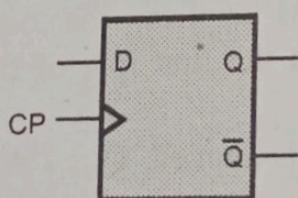


Fig. Q.8.1 (a) Logic symbol

CP	D	Q_{n+1}
↑	0	0
↑	1	1
0	X	Q_n

Fig. Q.8.1 (b) Truth table of D flip-flop

Q_{n+1} function follows D input at the positive going edges of the clock pulses. Hence the characteristic equation for D flip-flop is $Q_{n+1} = D$.

Q.9 Draw the logic diagram and give the characteristic table of JK flip-flop.

OR Explain the operation of JK flip-flop.

Ans.: Fig. Q.9.1 shows the logic diagram, symbol and truth table of positive edge triggered JK flip-flop.

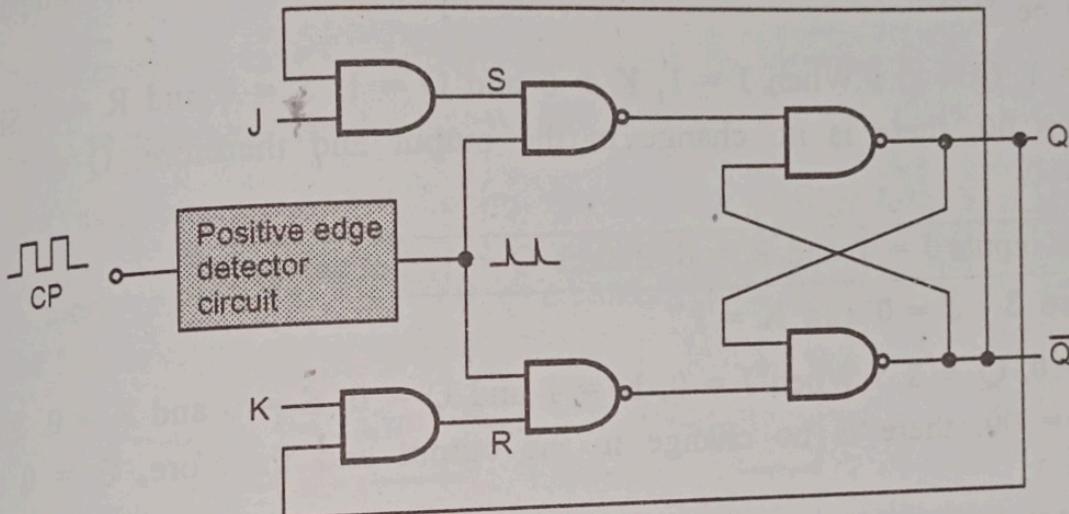


Fig. Q.9.1 (a) Clocked JK flip-flop

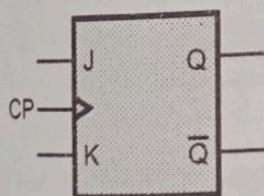


Fig. Q.9.1 (b) Logic symbol

Q_n	J	K	Q_{n+1}		J	K	Q_{n+1}
0	0	0	0	=	0	0	Q_n
0	0	1	0		0	1	0
0	1	0	1		1	0	1
0	1	1	1		1	1	$\overline{Q_n}$
1	0	0	0				
1	0	1	1				
1	1	0	1				
1	1	1	0				

Fig. Q.9.1 (c) Truth table

$Q_n \backslash JK$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

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

Fig. Q.9.1 (d) Characteristics equation

Operation of JK flip-flop**Case 1 : $J = K = 0$**

When $J = K = 0$, $S = R = 0$ and according to truth table of SR flip-flop there is no change in the output.

When inputs $J = K = 0$, output does not change.

Case 2 : $J = 1$ and $K = 0$

$Q = 0, \bar{Q} = 1$: When $J = 1, K = 0$ and $Q = 0, S = 1$ and $R = 0$. According to truth table of SR flip-flop it is set state and the output Q will be 1.

$Q = 1, \bar{Q} = 0$: When $J = 1, K = 0$ and $Q = 1, S = 0$ and $R = 0$. Since $SR = 00$, there is no change in the output and therefore, $Q = 1$ and $\bar{Q} = 0$.

The inputs $J = 1$ and $K = 0$, makes $Q = 1$, i.e. set state.

Case 3 : $J = 0$ and $K = 1$

$Q = 0, \bar{Q} = 1$: When $J = 0, K = 1$ and $Q = 0, S = 0$ and $R = 0$. Since $SR = 00$, there is no change in the output and therefore, $Q = 0$ and $\bar{Q} = 1$.

$Q = 1, \bar{Q} = 0$: When $J = 0, K = 1$ and $Q = 1, S = 0$ and $R = 1$. According to truth table of SR flip-flop it is a reset state and the output Q will be 0.

The inputs $J = 0$ and $K = 1$, makes $Q = 0$, i.e., reset state.

Case 4 : $J = K = 1$

$Q = 0, \bar{Q} = 1$: When $J = K = 1$ and $Q = 0, S = 1$ and $R = 0$. According to truth table of SR flip-flop it is a set state and the output Q will be 1.

$Q = 1, \bar{Q} = 0$: When $J = K = 1$ and $Q = 1, S = 0$ and $R = 1$. According to truth table of SR flip-flop it is a reset state and the output Q will be 0.

The input $J = K = 1$, toggles the flip-flop output.

Q.10 Discuss the race around condition and its solution.

[SPPU : Dec.-15, May-15, Marks 6]

OR What is race around condition ? Explain in brief.

Ans. • In JK flip-flop, when $J = K = 1$, the output toggles (output changes either from 0 to 1 or from 1 to 0).

- Consider that initially $Q = 0$ and $J = K = 1$. After a time interval Δt equal to the propagation delay through two NAND gates in series, the output will change to $Q = 1$ and after another time interval of Δt the output will change back to $Q = 0$. This toggling will continue until the flip-flop is enabled and $J = K = 1$. At the end of clock pulse the flip-flop is disabled and the value of Q is uncertain. This situation is referred to as the **race-around condition**. This is illustrated in Fig. Q.10.1.

- This condition exists when $t_p \geq \Delta t$. Thus by keeping $t_p < \Delta t$ we can avoid race around condition.
- We can keep $t_p < \Delta t$ by keeping the duration of edge less than Δt .
- A more practical method for overcoming this difficulty is the use of the Master-Slave (MS) configuration.

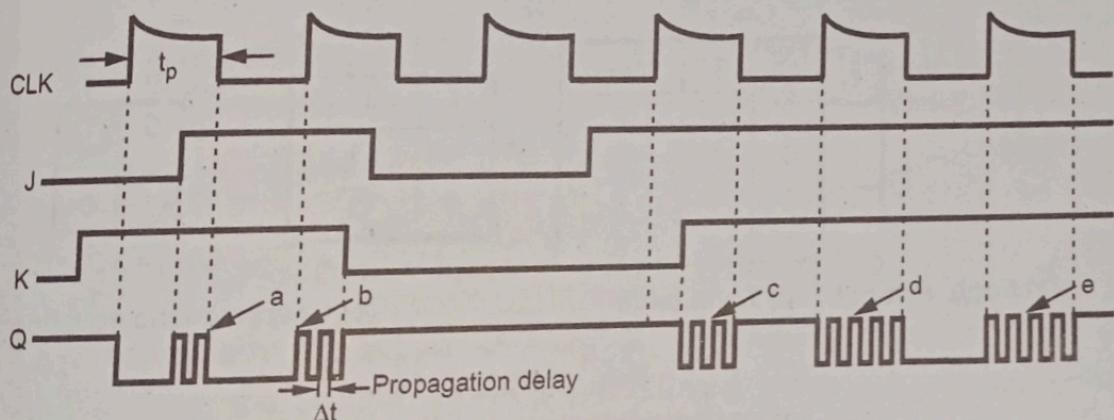


Fig. Q.10.1 Input and output waveforms for clocked JK flip-flop

Q.11 Explain the race condition in context of SR/RS flip-flop.

Ans. : In SR/RS flip-flop, when both inputs are logic ($S = R = 1$) and the clock pulse is applied, the state of the flip-flop is undefined. In this case, both the outputs Q and \bar{Q} try to become 1. This violates the rule that the outputs (Q and \bar{Q}) of the flip-flop are complement of each other. Such condition is known as the race condition in context of SR/RS flip-flop.

Q.12 Explain the operation of T flip - flop.

Ans. • T flip-flop is also known as 'Toggle flip-flop'.

- As shown in the Fig. Q.12.1, the T flip-flop is obtained from a JK flip-flop by connecting both inputs, J and K together.

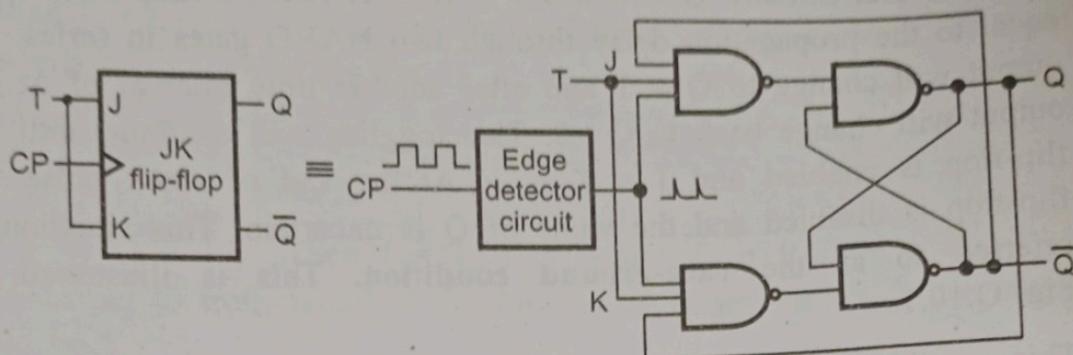


Fig. Q.12.1 T flip-flop using NAND gates

- When $T = 0$, $J = K = 0$ and hence there is no change in the output.
When $T = 1$, $J = K = 1$ and hence output toggles.
- The Fig. Q.12.2 shows logic symbol, truth table and the characteristic equation for T flip-flop.

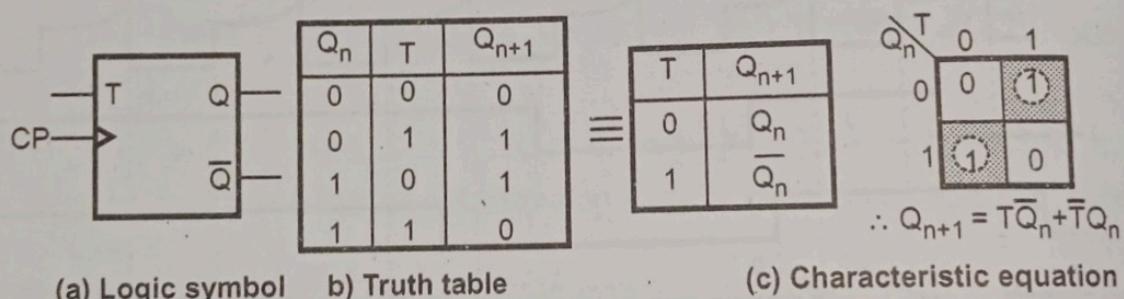


Fig. Q.12.2

6.3 : Preset and Clear

Q.13 What are preset and clear ?

Ans. : When power is turned ON, the state of the flip-flop is uncertain. It may come to set ($Q = 1$) or reset ($Q = 0$) state. In many applications, it is necessary to initially set or reset the flip-flop. Such initial state of flip-flop can be accomplished by using the direct or asynchronous inputs of the flip-flop. These inputs are : Preset (\bar{P}) and Clear (\bar{C}). They can be applied at any time between clock pulses and are not in synchronism with the clock.

6.4 : Master and Slave Flip Flop

Q.14 Explain working of master - slave JK flip-flop with necessary logic diagram, state equation and state diagram.

 [GTU : Winter-17, Marks 7]

OR Explain MS J-K flip-flop.

 [GTU : May-14, Dec.-13, May-12, Winter-17, Marks 4]

Ans. : Fig. Q.14.1 shows the master-slave JK flip-flop. Positive clock pulses are applied to first flip-flop and inverted (negative) clock pulses are applied to second flip-flop.

When $CK = 1$, the first flip-flop is enabled and the outputs Q_M and Preset

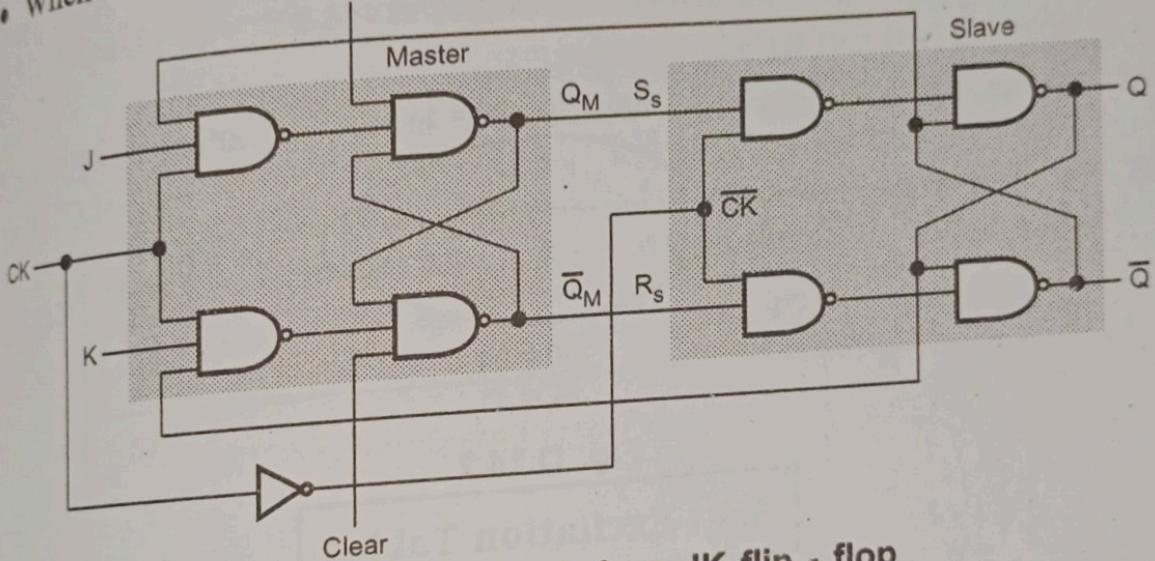


Fig. Q.14.1 Master - slave JK flip - flop

\bar{Q}_M responds to the inputs of J and K according to the Table Q.14.1. At this time, the second flip-flop is inhibited because its clock is low, $\bar{CK} = 0$.

- When CK goes Low ($\bar{CK} = 1$), the first flip-flop is inhibited and second flip-flop is enabled. At this time, the output of second flip-flop (Q and \bar{Q}) follow the outputs Q_M and \bar{Q}_M , respectively.
- Since the second flip-flop follows the first one, it is referred to as the slave and the first one as the master.
- In master-slave JK flip-flop state change occurs when flip-flop goes through both positive transition (first half) of clock and negative transition of the clock (second half). Thus, race-around condition does not exist in the master-slave JK flip-flop.

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

 \equiv

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Table Q.14.1 Truth table

Characteristics equation : $Q_{n+1} = \overline{Q}_n J + Q_n \overline{K} = J\overline{Q}_n + \overline{K}Q_n$

State Diagram :

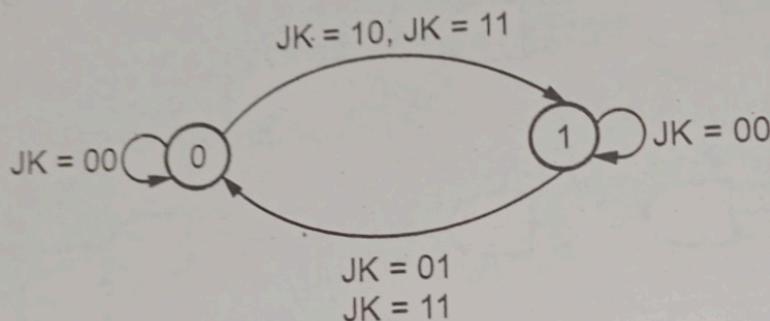


Fig. Q.14.2

6.5 : Excitation Tables

Q.15 What do you mean by excitation table ? Give excitation tables of SR, JK, D and T flip-flops ?

Ans. : • The table that gives the required inputs of the flip-flop for a given change of state is known as an excitation table of the flip-flop.

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

SR excitation table
Table Q.15.1

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

JK excitation table
Table Q.15.2

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

D excitation table

Table Q.15.3

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

Table Q.15.4

6.6 : Conversion from One FF to Another

Q.16 Explain how SR-FF is converted into D FF.

[SPPU : May-12, Marks 2]

Ans.: The excitation table for above conversion is as shown in Table Q.16.1.

Input	Present state		Next state		Flip-flop inputs	
	D	Q_n		Q_{n+1}	S	R
0		0		0	0	X
0		1		0	0	1
1		0		1	1	0
1		1		1	X	0

Table Q.16.1

K-map simplification

Logic diagram

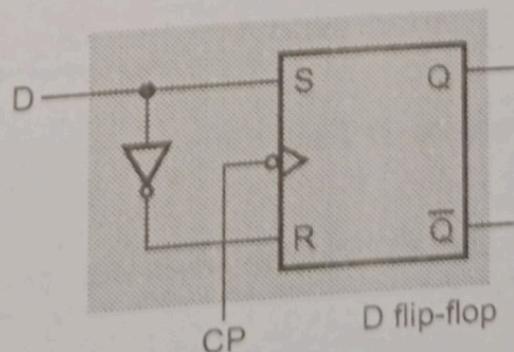
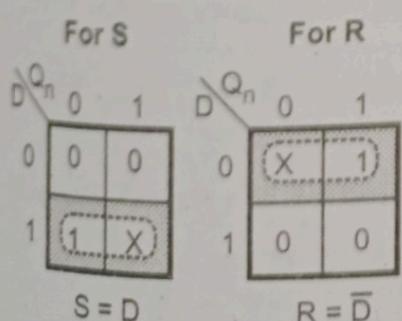


Fig. Q.16.1

Fig. Q.16.2 SR to D flip-flop conversion

Q.17 Explain how SR-FF is converted to JK FF.

[SPPU : May-12, 15, Marks 4]

Ans. : The excitation table for above conversion is as shown in Table Q.17.1.

Inputs		Present state Q_n	Next state Q_{n+1}	Flip-flop inputs	
J	K			S	R
0	0	0	0	0	X
	0	1	1	X	0
	1	0	0	0	X
1	1	1	0	0	1
	0	0	1	1	0
	1	0	1	X	0
1	0	1	1	1	0
	1	1	0	1	0
1	1	1	0	0	1

Table Q.17.1

K-map simplification

For S	
J	K Q_n
0	0 X 0 0
1	1 X 0 1

$S = J \bar{Q}_n$

For R	
J	K Q_n
0	X 0 1 1
1	0 0 1 0

$R = K Q_n$

Logic diagram

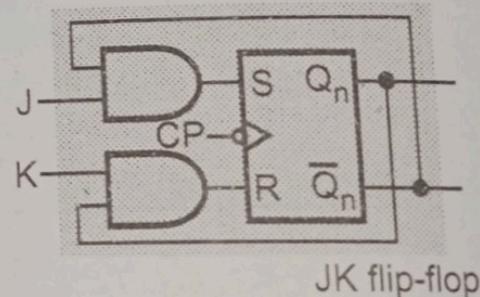


Fig. Q.17.1

Fig. Q.17.2 SR to JK
flip-flop conversion

Q.18 Explain how JK FF is converted to D FF.

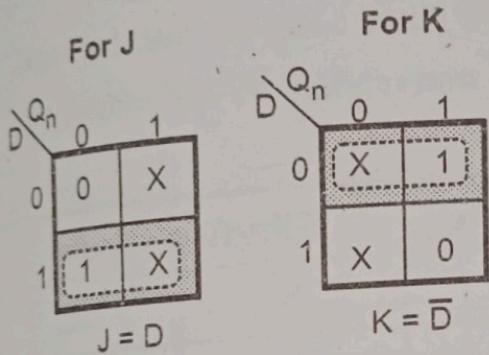
[SPPU : May-16, Marks 3]

Ans. : The excitation table for above conversion is as shown in the Table Q.18.1.

Input	Present state	Next state	Flip-flop inputs	
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

Table Q.18.1

K-map simplification



Logic diagram

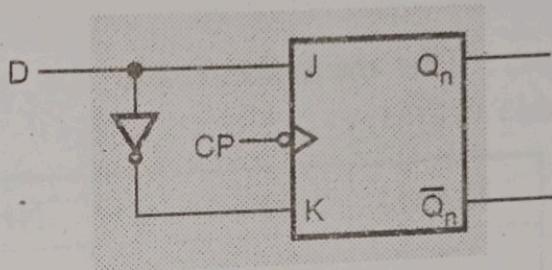


Fig. Q.18.2 JK to D flip-flop conversion

Fig. Q.18.1

Q.19 Explain how D FF is converted to T FF.

[SPPU : June-22, Marks 9]

Ans. : The excitation table for above conversion is as shown in the Table Q.19.1.

Input	Present state	Next state	Flip-flop input
T	Q_n	Q_{n+1}	D
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Table Q.19.1

K-map simplification

		For D	
		0	1
0		0	1
1		1	0

Fig. Q.19.1

$$D = \bar{T}Q_n + T\bar{Q}_n = T \oplus Q_n$$

Q.20 Explain how to convert SR flip-flop to T flip-flop ?

[SPPU : May-06, Dec.-16, Marks 3]

Ans. : The excitation table for above conversion is as shown in the Table Q.20.1.

Input	Present state	Next state	Flip-flop inputs	
			S	R
T	Q_n	Q_{n+1}		
0	0	0	0	X
0	1	1	X	0
1	0	1	1	0
1	1	0	0	1

Table Q.20.1

K-map simplification

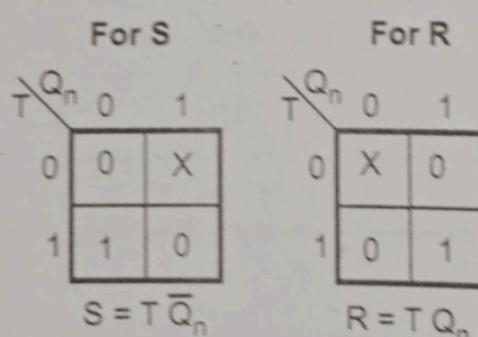


Fig. Q.20.1

Logic diagram

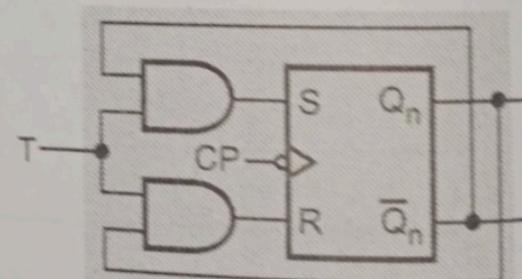


Fig. Q.20.2 SR to T flip-flop conversion

Q.21 Explain how to convert JK flip-flop to T flip-flop ?

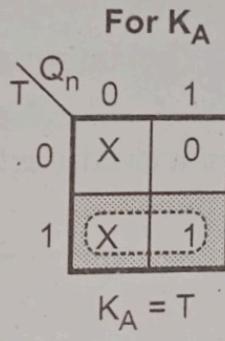
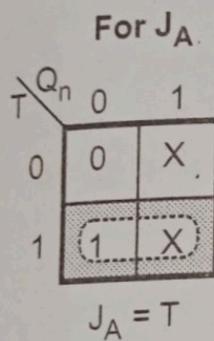
[SPPU : May-06,16,17 Marks 3]

Ans. : The excitation table for above conversion is as shown in Table Q.21.1.

Input	Present state	Next state	Flip-flop inputs	
T	Q_n	Q_{n+1}	J_A	K_A
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

Table Q.21.1

K-map simplification



Logic diagram

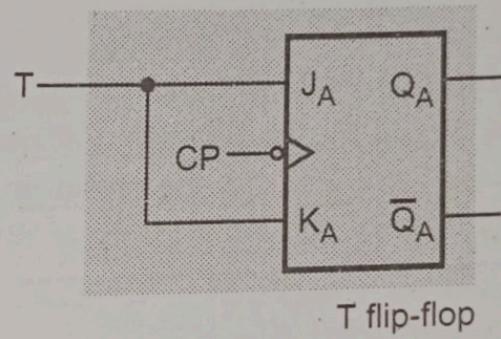


Fig. Q.21.1 JK to T flip-flop conversion

Fig. Q.21.1

Q.22 Write the Excitation Table of S-R flip-flop. Prepare the Truth Table for the following circuit and Determine the type of flip-flop.

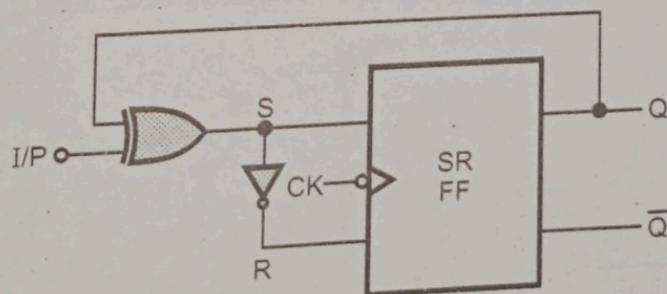


Fig.Q.22.1

[6]

Ans. : Refer Table Q.15.1

I/P	Q_n	S	R	Q_{n+1}
0	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	1	0

} When input is 0 output does not change

} When input is 1 output toggles

∴ Type of flip-flop is T

Truth Table

∴ Type of flip-flop is T.

6.7 : Study of 7474 and 7476 Flip-Flop ICs

Q.23 Write a short note on IC 7474.

Ans. : IC 7474 contains two independent positive-edge-triggered D flip-flops with complementary outputs. The information on the D input is accepted by the flip-flops on the positive going edge of the clock pulse. A low logic level on the preset or clear inputs will set or reset the outputs regardless of the logic levels of the other inputs.

Inputs			Outputs		
PR	CLR	CLK	D	Q	\bar{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	\bar{Q}_0

*This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (high) level.

Table Q.23.1 Function table

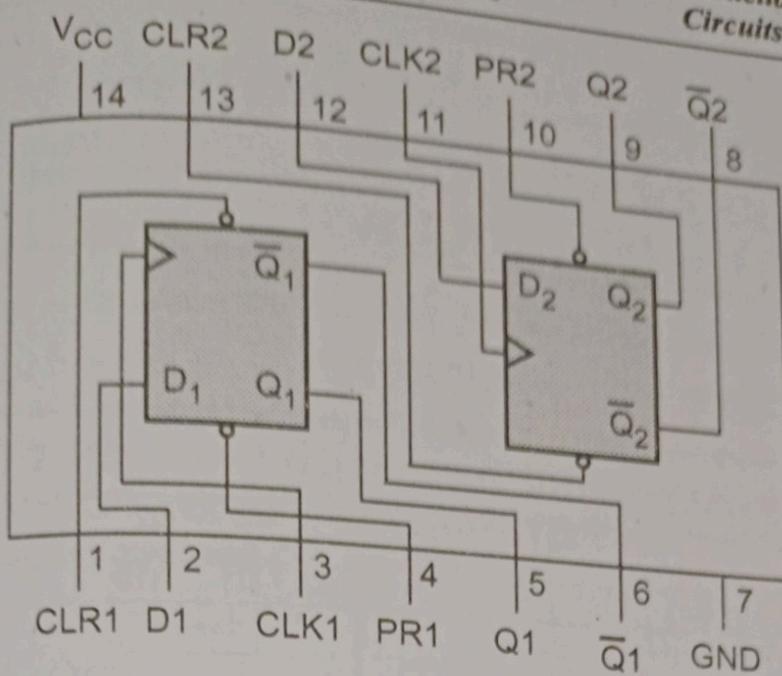


Fig. Q.23.1 Connection diagram

Q.24 Write a short note on IC 7476.

Ans. : IC 7476 contains two independent positive pulse triggered J-K flip-flops with complementary outputs. The J and K data is processed by the flip-flops after a complete clock pulse. While the clock is low the slave is isolated from the master. On the positive transition of the clock, the data from the J and K inputs is transferred to the master. While the clock is high the J and K inputs are disabled. On the negative transition of the clock, the data from the master is transferred to the slave. The logic state of J and K inputs must be allowed to change while the clock is high. The data is transferred to the outputs on the falling edge of the clock pulse. A low logic level on the preset or clear inputs will set or reset the outputs regardless of the logic levels of the other inputs.

Inputs					Outputs		
PR	CLR	CLK	J	K	Q	\bar{Q}	
L	H	X	X	X	H	L	
H	L	X	X	X	L	H	
L	L	X	X	X	H*	H*	
H	H		L	L	Q ₀	\bar{Q}_0	

H	H		H	L	H	L	
H	H		L	H	L	H	
H	H		H	H		Toggle	

*This configuration is nonstable; that is, it will not persist when the preset and/or clear inputs return to their inactive (high) level.

Table Q.24.1 Function table

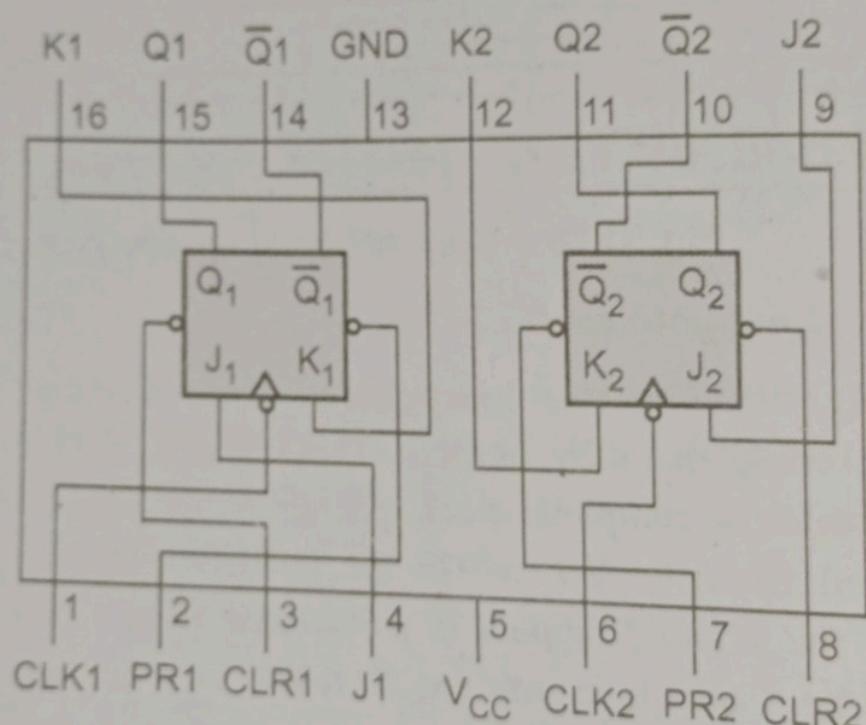


Fig. Q.24.1 Connection diagram

END... ↗

7

Applications of Flip-Flops-Registers

7.1 : Buffer Register and Shift Register

Q.1 What is buffer register ?

Ans. : Fig. Q.1.1 shows the simplest register constructed with four D flip-flops. This register is also called **buffer register**. Each D flip-flop is triggered with a common negative edge clock pulse. The input bits set up the flip-flops for loading. Therefore, when the first negative clock edge arrives, the stored binary information becomes,

$$Q_A Q_B Q_C Q_D = ABCD$$

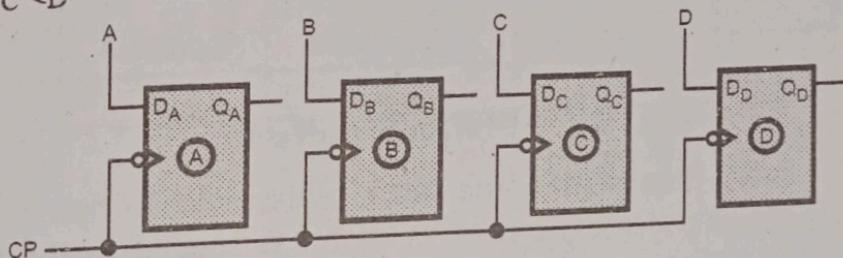


Fig. Q.1.1 Buffer register

In this register, four D flip-flops are used. So it can store 4-bit binary information. Thus the number of flip-flop stages in a register determines its total storage capacity.

☞ [SPPU : June-22, Marks 4]

Q.2 What are shift registers ?

Ans. : • A group of flip-flops can be used to store a word, which is called **register**.

- The binary information (data) in a register can be moved from stage to stage within the register or into or out of the register upon application of clock pulses. Such registers are called 'shift registers'.

Q.3 Explain the operational types of shift register.

☞ [SPPU : June-22, Marks 5]

Ans. : Fig. Q.3.1 gives the symbolical representation of the different types of data movement in shift register operations.

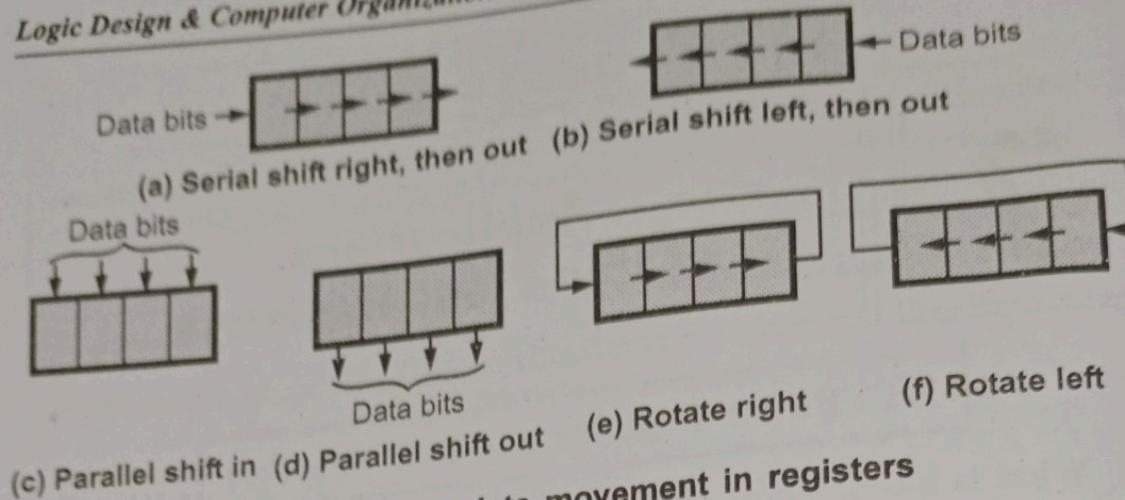


Fig. Q.3.1 Basic data movement in registers

Q.4 Explain the operation of SISO shift register.

☞ [SPPU : Dec.-14, Marks 6]

Ans. : Fig. Q.4.1 shows serial-in serial-out shift-left register.

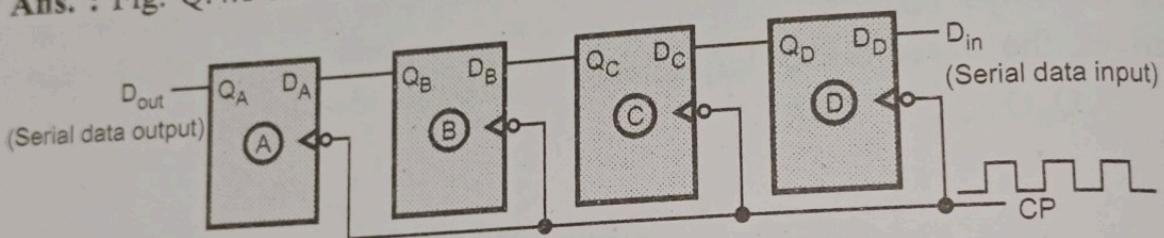


Fig. Q.4.1 Shift-left register

In this shift register, data within the shift register is shifted left one bit position at each clock pulse. The data input bit is loaded in the right most flip-flop.

Fig. Q.4.2 shows serial-in serial-out shift-right register.

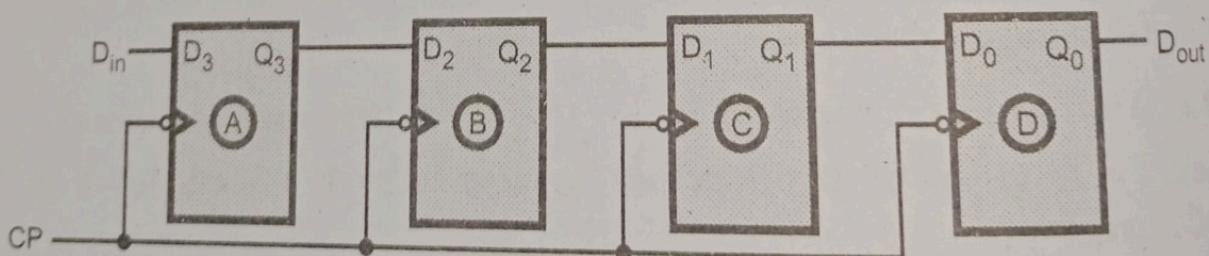


Fig. Q.4.2 Shift-right register

In this shift register, data within the shift register is shifted right one bit position at each clock pulse. The data input bit is loaded in the left most flip-flop.

Q.5 Explain the operation of SIPO shift register.

☞ [SPPU : Dec.-12, Marks 5]

Ans. : • In SIPO, the data bits are entered serially into the register but the output is taken in parallel.

- Once the data are stored, each bit appears on its respective output line and all bits are available simultaneously as shown in Fig. Q.5.1.

CP	Q_3	Q_2	Q_1	Q_0
-	NC	NC	NC	NC

Table Q.5.1 Truth table

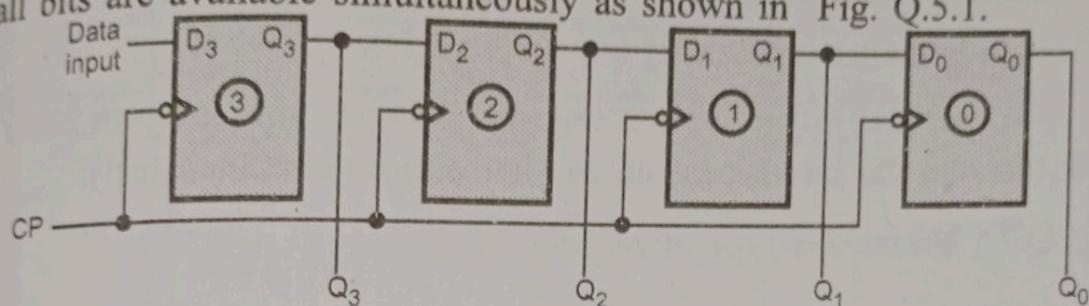


Fig. Q.5.1 A Serial In Parallel Out (SIPO) shift register

Q.6 Explain the operation of PISO shift register.

[SPPU : May-10, 14, Marks 6]

Ans. : • Fig. Q.6.1 illustrates a four-bit parallel in serial out register.

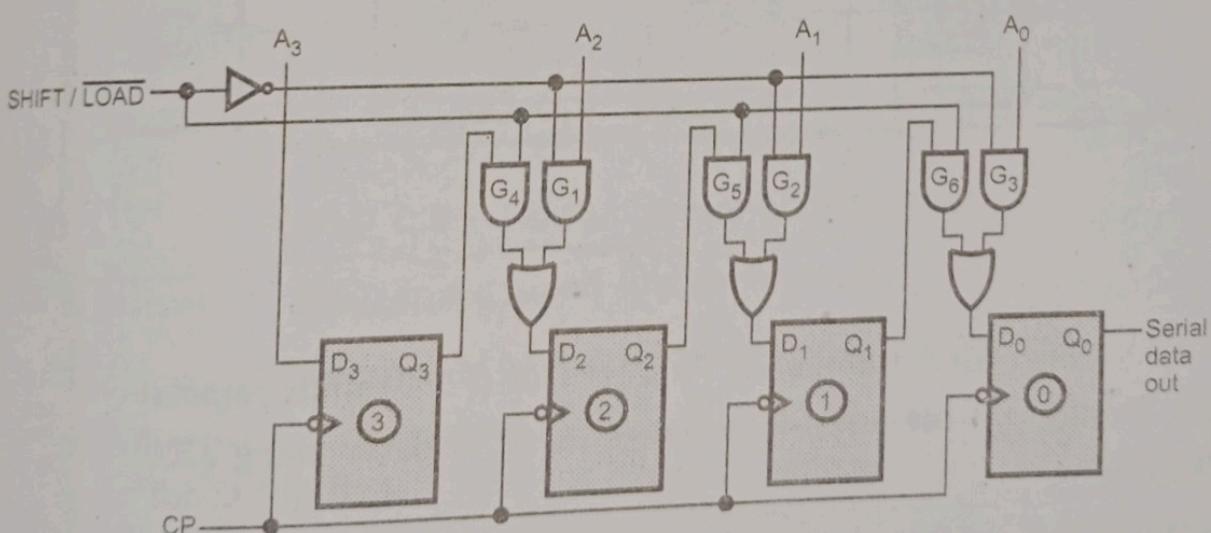


Fig. Q.6.1 Parallel In Serial Out (PISO) shift register

- There are four input lines A_3, A_2, A_1, A_0 for entering data in parallel into the register.
- SHIFT/LOAD is the control input which allows shift or loading data operation of the register.
- When SHIFT/LOAD is low, gates G_1, G_2, G_3 are enabled, allowing each input data bit to be applied to D input of its respective flip-flop.
- When a clock pulse is applied, the flip-flops with $D = 1$ will SET and those with $D = 0$ will RESET.

- All four bits are stored simultaneously.
- When SHIFT/LOAD is high, gates G_1, G_2, G_3 are disabled and gates G_4, G_5, G_6 are enabled. This allows the data bits to shift right from one stage to the next.
- The OR gates at the D-inputs of the flip-flops allow either the parallel data entry operation or shift operation, depending on which AND gates are enabled by the level on the SHIFT/LOAD input.

Q.7 Explain parallel in parallel out shift register.

Ans. : • In 'parallel in parallel out register', there is simultaneous entry of all data bits and the bits appear on parallel outputs simultaneously.

- Fig. Q.7.1 shows this type of register.

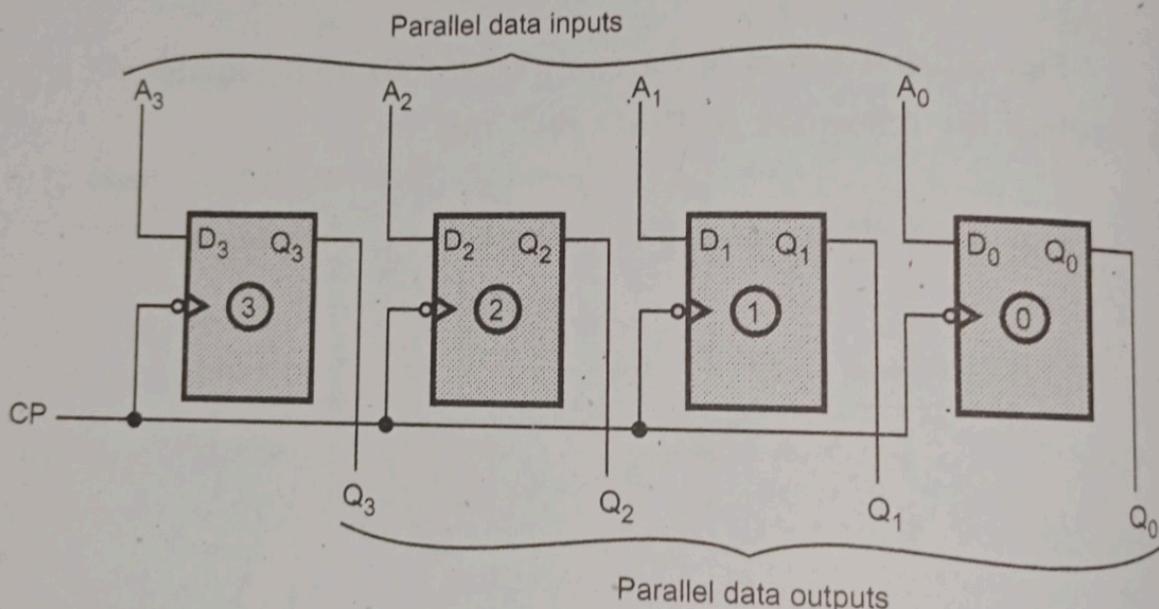


Fig. Q.7.1 Parallel In Parallel Out (PIPO) shift register

Q.8 Design a 4-bit serial-in-serial-out shift register using JK flip-flops.

Ans.: In Q.4 we have seen 4-bit serial-in serial-out shift registers using D Flip-Flops. By replacing D flip-flop using equivalent JK flip-flops we can implement SISO shift register using JK flip-flops. By giving complement inputs to JK flip-flops we can use it as a D flip-flop.

Q.9 Draw and explain the working of universal shift register.

Ans. : • If the register has both shifts (right shift and left shift) and parallel load capabilities, it is referred to as **universal shift register**.

• The Fig. Q.9.1 shows the 4-bit universal shift register.

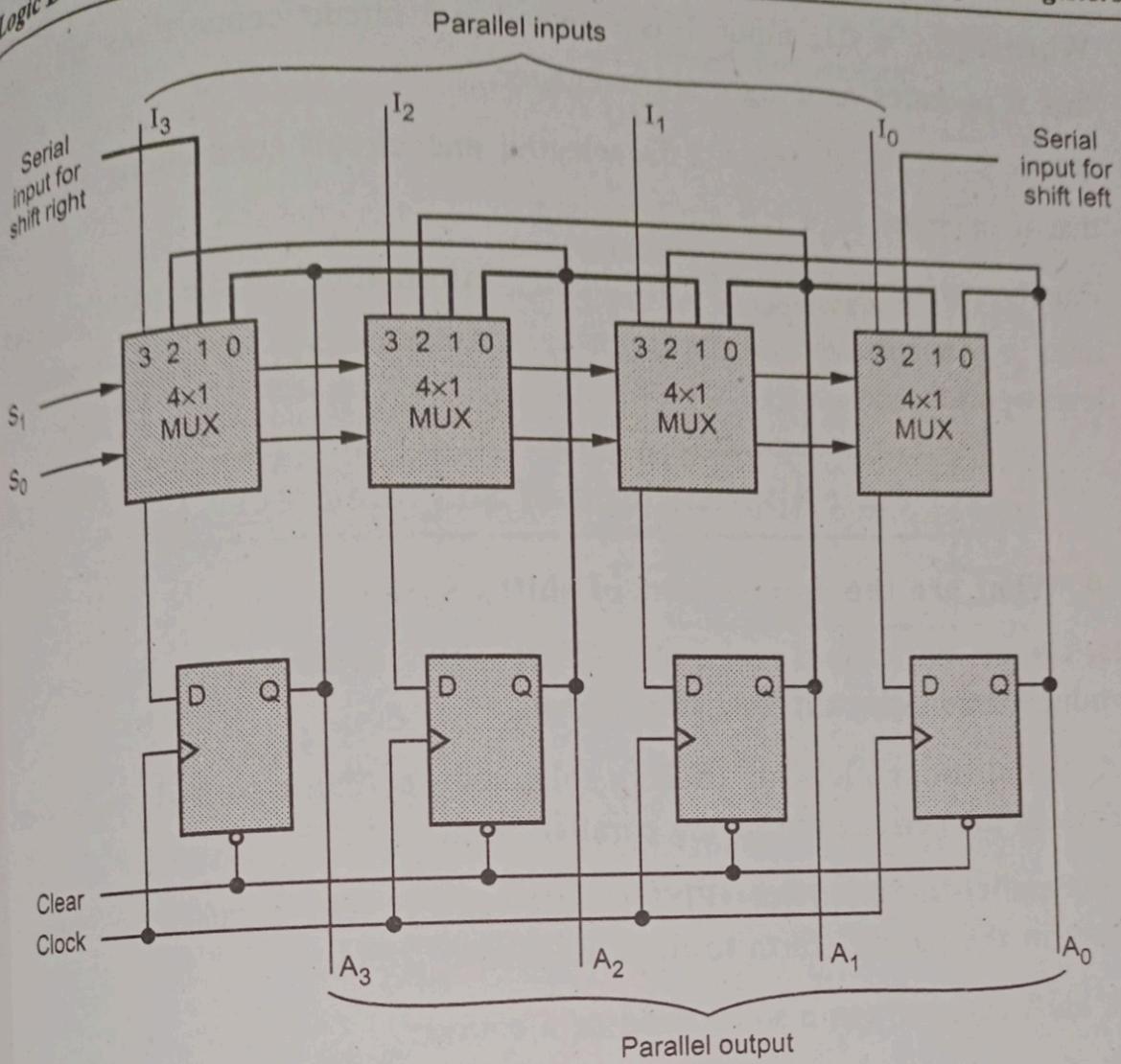


Fig. Q.9.1 4-bit universal shift register

- It consists of four flip-flops and four multiplexers.
- The four multiplexers have two common selection inputs S_1 and S_0 , and they select appropriate input for D flip-flop.
- The Table Q.9.1 shows the register operation depending on the selection inputs of multiplexers.
- When $S_1S_0 = 00$, input 0 is selected and the present value of the register is applied to the D inputs of the flip-flops. This results no change in the register value.

Mode control		Register operation
S_1	S_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Table Q.9.1 Mode control and register operation

- When $S_1S_0 = 01$, input 1 is selected and circuit connections are such that it operates as a right shift register.
- When $S_1S_0 = 10$, input 2 is selected and circuit connections are such that it operates as a left shift register.
- Finally, when $S_1S_0 = 11$, the binary information on the parallel input lines is transferred into the register simultaneously and it is a parallel load operation.

7.2 : Applications of Shift Registers

Q.10 What are the applications of shift register ?

Ans. : • A Serial-In-Serial-Out (SISO) shift register can be used to introduce time delay Δt in digital signals.

- A Serial-In-Parallel-Out (SIPO) shift register can be used to convert data in the serial form to the parallel form.
- A Parallel-In-Serial-Out (PISO) shift register can be used to convert data in the parallel form to the serial form.
- A shift register can also be used as a counter.
- Shift register is a pseudo-random binary sequence generator.
- The shift register can be used to generate a particular bit pattern repetitively.
- The shift register can be used to detect the desired sequence.

7.3 : Sequence Detector using Shift Register

Q.11 Write a note on IC 74194.

Ans. : We know that a register may operate in any of the modes, like, SISO, SIPO, PISO, PIPO or bidirectional. If a register can be operated in all the five possible ways, it is known as Universal Shift Register. The IC 74194 is a 4-bit universal shift register. Fig. Q.11.1 shows the pin configuration of IC 74194.

As shown in the Fig. Q.11.1, 74194 has 4 parallel data inputs ($D_0 - D_3$), and S_0 and S_1 are the control inputs.

Operating Modes

S ₁	S ₀	Operation	Description
Inputs			
0	0	Hold	Do nothing.
0	1	Shift right	Serial data is entered at the shift-right serial input, DSR.
1	0	Shift left	Serial data is entered at the shift-left serial input, DSL.
1	1	Parallel load	Data appearing on D ₀ -D ₃ inputs is transferred to the Q ₀ -Q ₃ outputs, respectively.

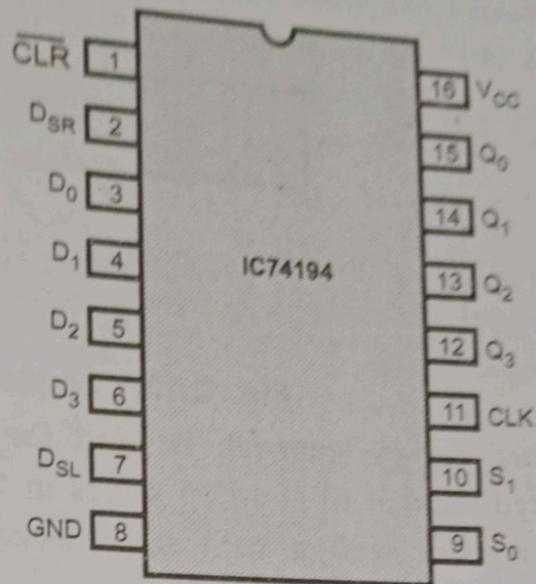


Fig. Q.11.1 Pin configuration

Table Q.11.1 Operating modes of 74194

Q.12 Explain the design process of sequence generator using shift register.

Ans. : The simplest way of designing sequence generator using shift register is to take shift register of n-bits where n is equal to the length of sequence. Then load the bit sequence in the shift register by parallel load operation and apply the clock signal. The Fig. Q.12.1 illustrates the operation of such a sequence generator. Here, the sequence to be generated is 11001.

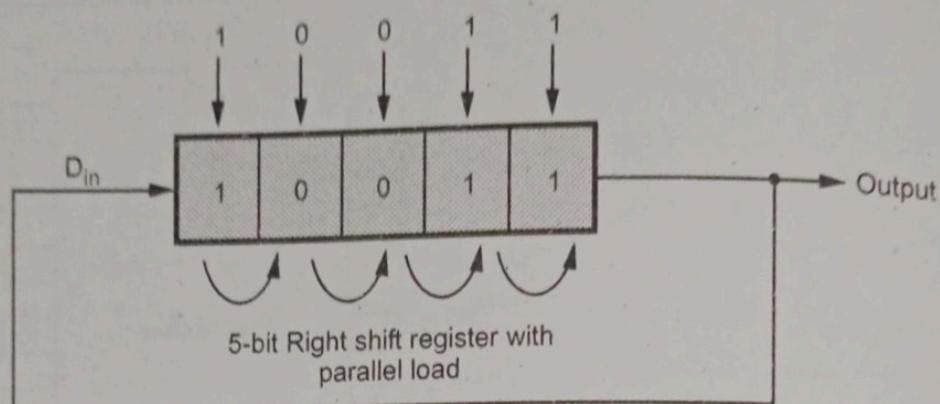


Fig. Q.12.1 Sequence generator

Another design approach is used for sequence generator to reduce the required number of flip-flop stages in the shift register. In this approach a shift register with a next state decoder and preset logic is used. The Fig. Q.12.1 shows the block diagram of this approach. Here, the output of the next state decoder is a function of Q_A, Q_B, \dots, Q_n and it is used to determine the D_{in} input for the shift right register. Initially, start button is pressed to activate parallel load operation. This loads initial value in the shift register. Then at each clock pulse shift register contents are shifted right by 1 bit position. The next state decoder circuit decodes the output and generates D_{in} input for the shift register such that output Q_A generates the desired sequence. After completion of one complete sequence, the register is again loaded with initial value to start the next train of sequence.

After completion of one complete sequence, the register is again loaded with initial value to start the next train of sequence.

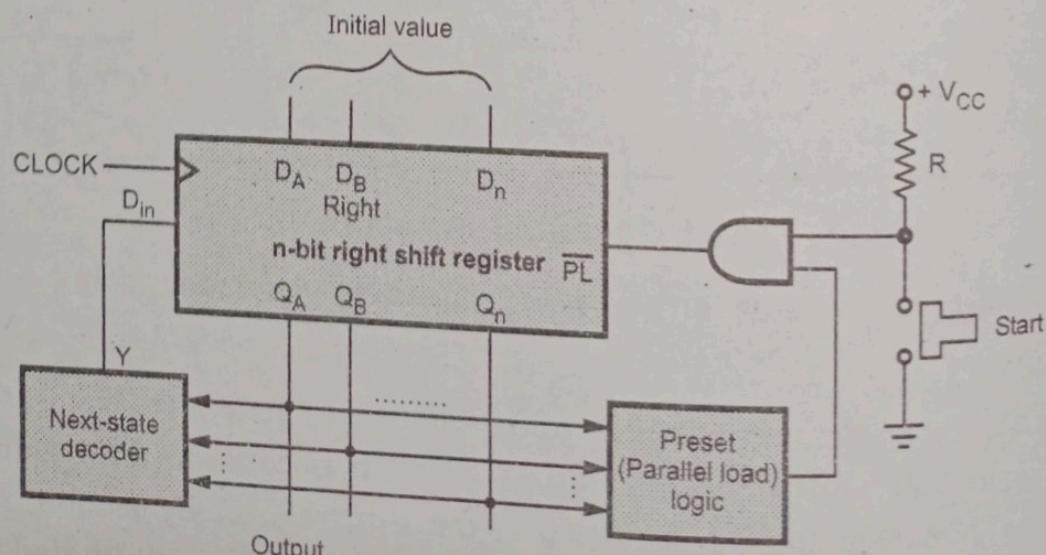


Fig. Q.12.2 Sequence generator using shift register

Q.13 Design a sequence generator to generate the sequence 1101011 by shift register method.

[May-10, 15, Marks 8]

Ans. : In this approach, the minimum number of flip-flops n, required to generate a sequence of length N is given by

$$N \leq 2^n - 1$$

In this example, $N = 7$, therefore, the minimum value of n, which may generate this sequence is 3. However, it is not guaranteed to lead to a solution. Let us try with 3 flip-flops. The Table Q.13.1 shows sequence generation with three flip-flops.

As shown in the Table Q.13.2, the state 6 is repeated. This means that $n = 3$ is not sufficient. Let us try with 4 flip-flops. The table shows sequence generation with four flip-flops.

CP	Flip-flop outputs			D_{in}	States
	Q_A	Q_B	Q_C		
1	1	0	0	1	4
2	1	1	0	0	6
3	0	1	1	1	3
4	1	0	1	0	5
5	0	1	0	1	2
6	1	0	1	-	5
-	-	-	-	-	-

State is repeated →

Table Q.13.1

CP	Flip-flop outputs				D_{in}	\overline{Preset}	States
	Q_A	Q_B	Q_C	Q_D			
Initial value	1	0	0	0	1	1	8
2	1	1	0	0	0	1	12
3	0	1	1	0	1	1	6
Sequence	4	1	0	1	1	0	11
5	0	1	0	1	1	1	5
6	1	0	1	0	1	1	10
7	1	1	0	1	1	1	13
Preset flip-flop	1	1	1	1	X	0	14
	1	1	0	0	0	1	8

Table Q.13.2

As shown in the Table Q.13.2, states are not repeated. After completion of one complete sequence, register is preset to value 1000 to start the next train of sequence. Fig. Q.13.1 (b) shows the logic diagram.

K-map simplification for D_{in}

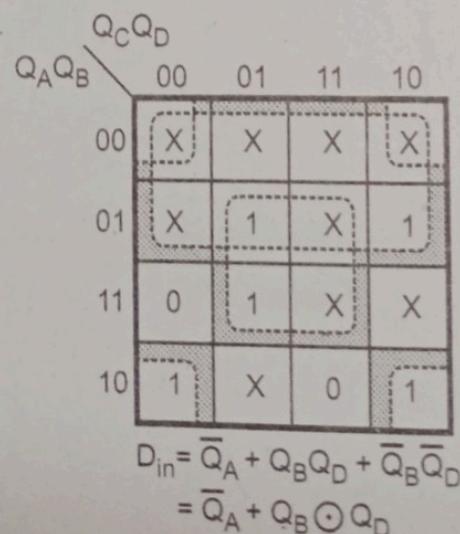


Fig. Q.13.1 (a)

Logic diagram

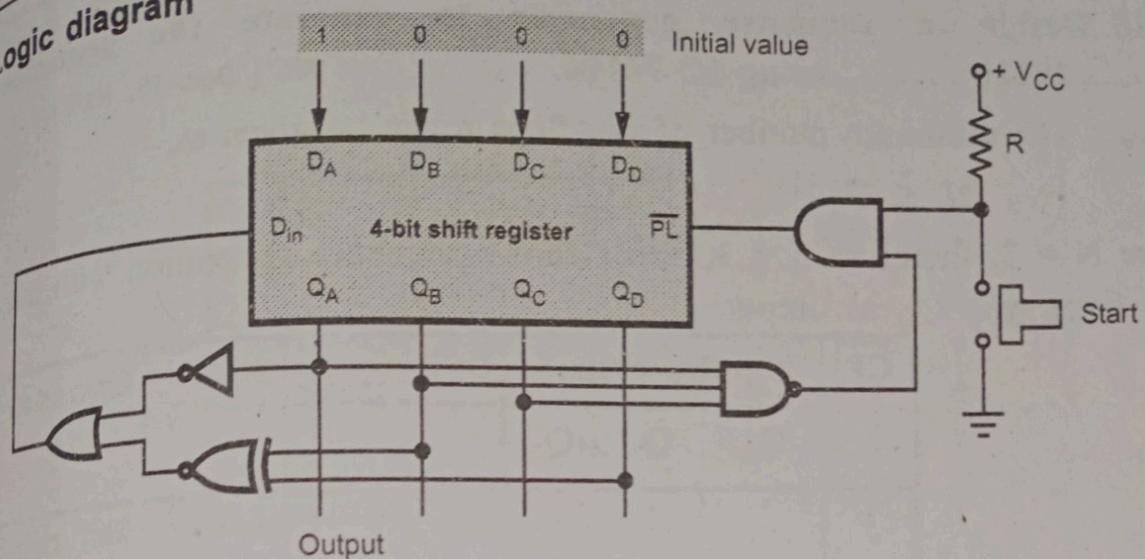


Fig. Q.13.1 (b)

Q14 Design a sequence generator using shift register and decoder circuit to generate the sequence1101011.....

[May-15, Dec.-16, Marks 6]

Ans. : Referring Table Q.13.2 we have,

Here, the combinational logic to derive D_{in} is implemented using decoder. Same decoder is used to generate preset signal.

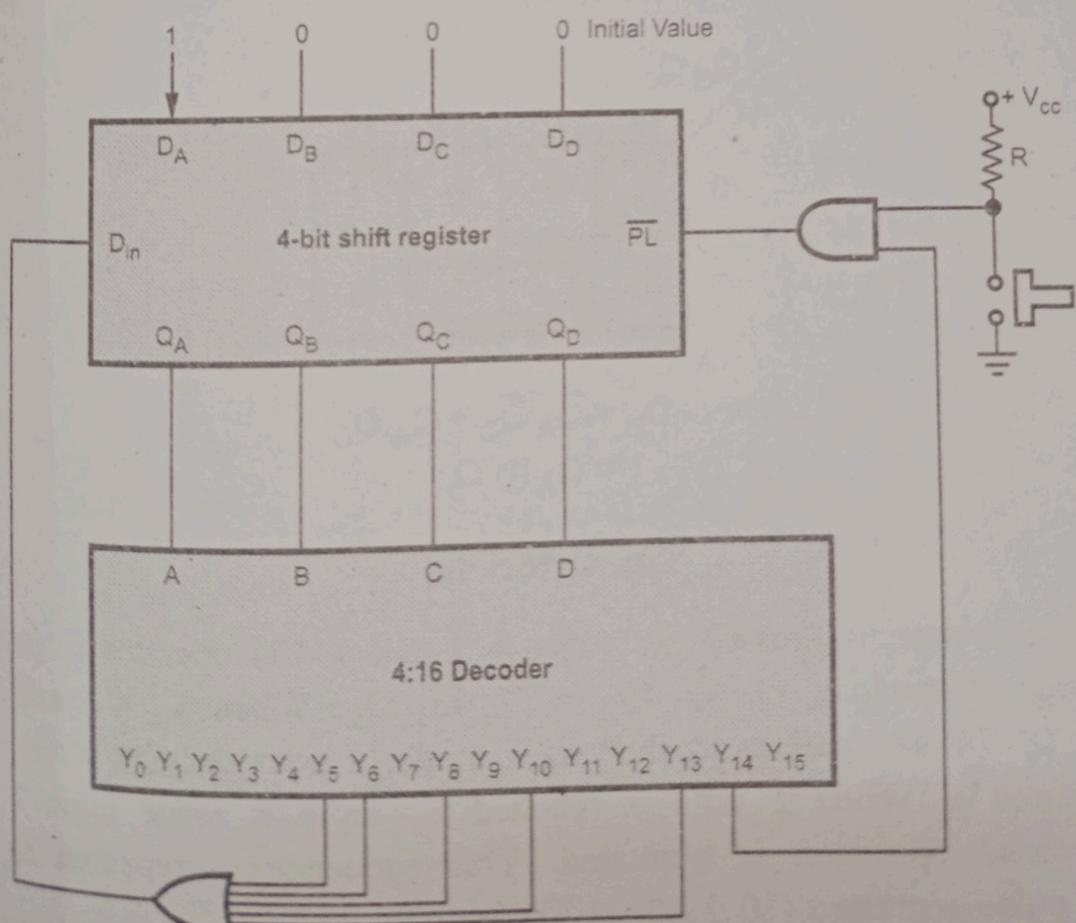


Fig. Q.14.1

Q.15 Design a sequence generator to generate the sequence10110using IC 74194. [Dec.-15, Marks 7]

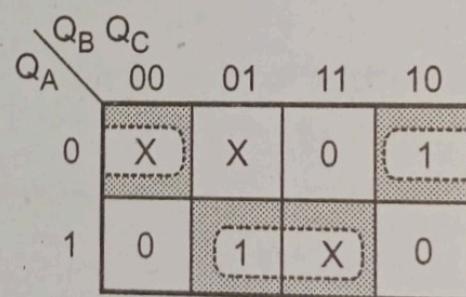
Ans. : The minimum number of flip-flops n can be given as,

$$N \leq 2^{n-1}$$

Here $N = 3$, therefore $n = 3$. Table shows sequence generation with three flip flops with Q_0 as output.

CP	Flip-flop outputs			DSR	S_1	S_0	States	Operation	
	Q_0	Q_1	Q_2						
Sequence	1	1	0	0	0	0	1	4	Shift right
	2	0	1	0	1	0	1	2	Shift right
	3	1	0	1	1	0	1	5	Shift right
	4	1	1	0	0	0	1	6	Shift right
	5	0	1	1	0	0	1	3	Shift right
	6	0	0	1	X	1	1	1	Parallel load
Preset flip-flop	7	1	0	0	0	0	1	4	Shift right

K-map simplification for D_{in}



$$\begin{aligned} D_{in} &= \overline{Q}_A \overline{Q}_C + Q_A Q_C \\ &= Q_A \odot Q_A \end{aligned}$$

Fig. Q.15.1 (a)

When start button is pressed $S_0 = S_1 = 1$ and 74194 operates in parallel load mode. Therefore, $Q_0 Q_1 Q_2 = D_0 D_1 D_2$. When $S_0 = 1$ and $S_1 = 0$, IC 74194 operates in shift right mode; it goes through states 4, 2, 5, 6, 3 and 1. When 1 state is reached S_1 input is made logic 1 and parallel load operation is activated. This sequence is repeated to get desired pulse sequence (10 11 0) at Q_0 output.

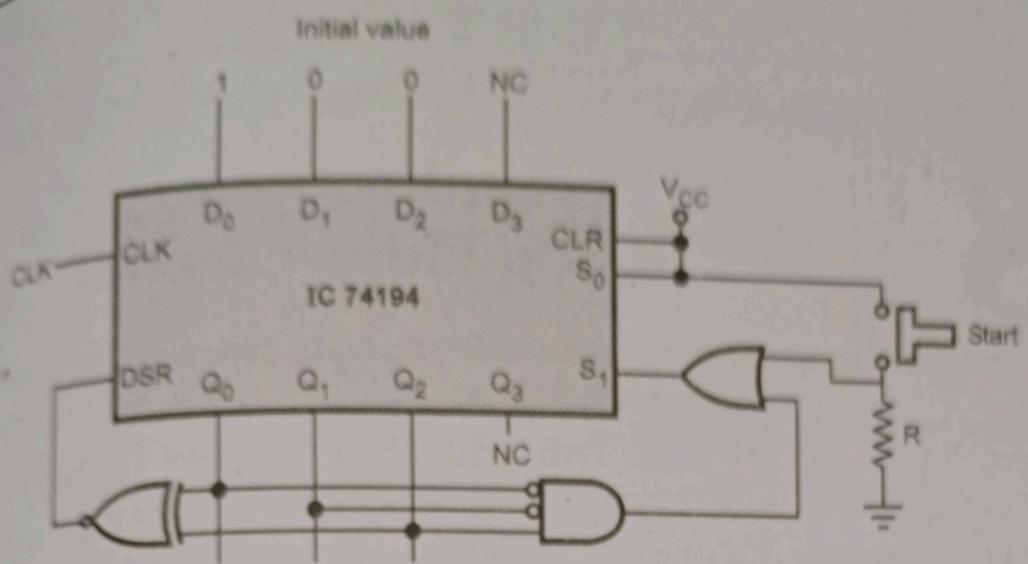


Fig. Q.15.1 (b)

Example for Practice

Q.16 Design sequence generator to generate the sequence 1011 using shift register IC 74194. [May-17, Marks 6]

END... ↗

8**Applications
of Flip-Flops-Counter****8.1 : Introduction**

Q.1 What is counter ? Give the difference between synchronous and asynchronous counters.

[SPPU : May-07, Dec.-07, Marks 2]

Ans. : • A counter is a register capable of counting the number of clock pulses arriving at its clock input.

- There are two types of counters : synchronous counter and asynchronous counter.

Sr. No.	Asynchronous counters	Synchronous counters
1.	In this type of counter flip-flops are connected in such a way that output of first flip-flop drives the clock for the next flip-flop.	In this type there is no connection between output of first flip-flop and clock input of the next flip-flop.
2.	All the flip-flops are not clocked simultaneously.	All the flip-flops are clocked simultaneously.
3.	Logic circuit is very simple even for more number of states.	Design involves complex logic circuit as number of states increases.
4.	Main drawback of these counters is their low speed as the clock is propagated through number of flip-flops before it reaches last flip-flop.	As clock is simultaneously given to all flip-flops there is no problem of propagation delay. Hence they are high speed counters and are preferred when number of flip-flops increases in the given design.

Table Q.1.1 Synchronous Vs asynchronous counters

8.2 : Asynchronous (Ripple Counters)

a.2 Draw and explain the working of 2-bit asynchronous binary counter.

Ans. : • Fig. Q.2.1 (a) shows 2-bit asynchronous counter using JK flip-flops.

- The clock signal is connected to the clock input of only first stage flip-flop.

- The clock input of the second stage flip-flop is triggered by the Q_A output of the first stage.

- Because of the inherent propagation delay time through a flip-flop, a transition of the input clock pulse and a transition of the Q_A output of first stage can never occur at exactly the same time. Therefore, the two flip-flops are never simultaneously triggered, which results in asynchronous counter operation.

- Fig. Q.2.1 (b) shows the timing diagram for two-bit asynchronous counter. It illustrates the changes in the state of the flip-flop outputs in response to the clock.

- J and K input of JK flip-flops are tied to logic HIGH hence output will toggle for each negative edge of the clock input.

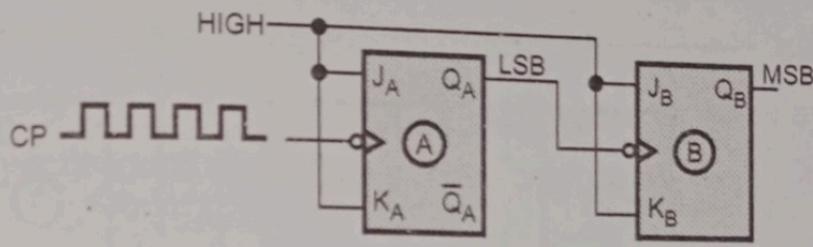


Fig. Q.2.1 (a) A two-bit asynchronous binary counter

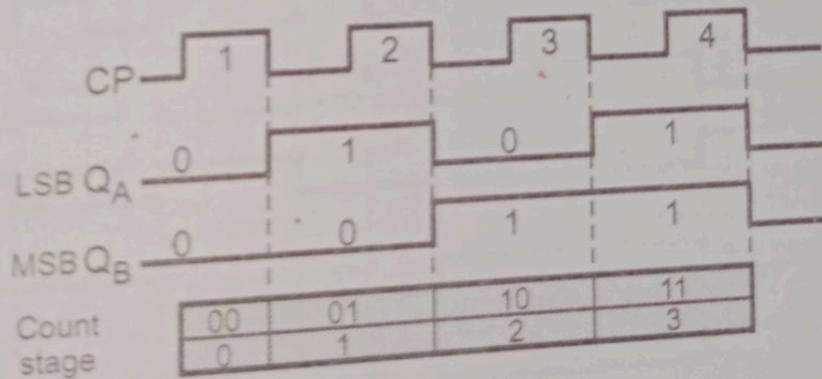


Fig. Q.2.1 (b) Timing diagram for the counter of Fig. Q.2.1 (a)

Q.3 Explain the working of 4-bit asynchronous down counter.

OR Draw diagram of a 4-bit binary ripple counter using flip - flops that trigger on negative edge transition. Also draw a timing diagram of the counter.

Ans. : • The Fig. Q.3.1 shows the 4-bit asynchronous down counter using JK flip-flops.

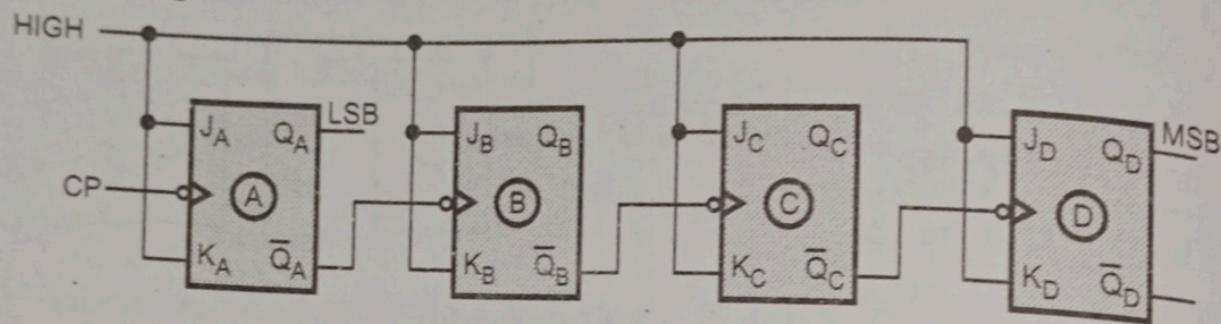


Fig. Q.3.1 4-bit asynchronous down counter

- The clock signal is connected to the clock input of only first flip-flop.
- The clock input of the remaining flip-flops is triggered by the \bar{Q}_A output of the previous stage instead of Q_A output of the previous stage.

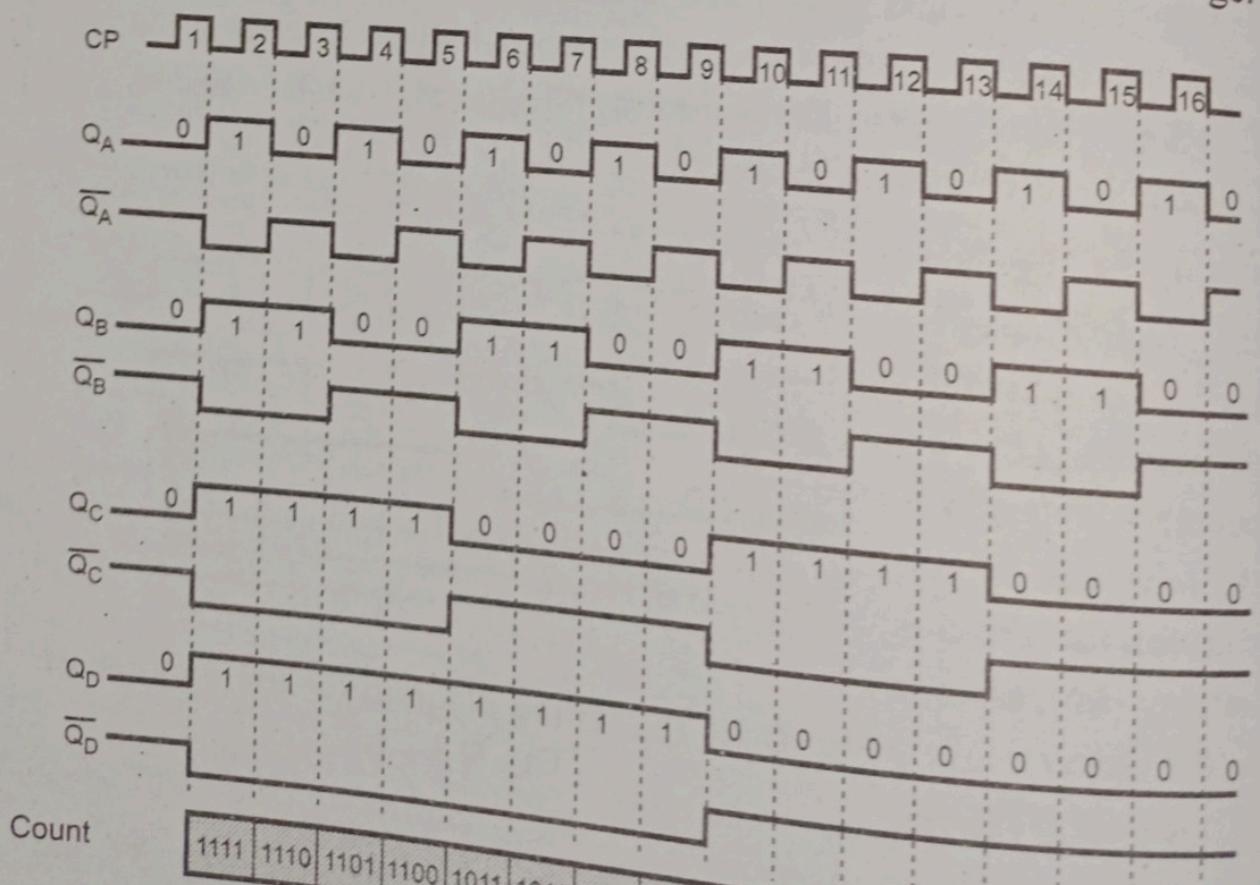


Fig. Q.3.2 Timing diagram of 4-bit asynchronous down counter

The Fig. Q.3.2 shows the timing diagram for 4-bit asynchronous down counter. It illustrates the changes in the state of the flip-flop outputs in response to the clock.

The J and K inputs of JK flip-flops are tied to logic HIGH hence output will toggle for each negative edge of the clock input.

Q.4 Design a 3-bit asynchronous up-down counter.

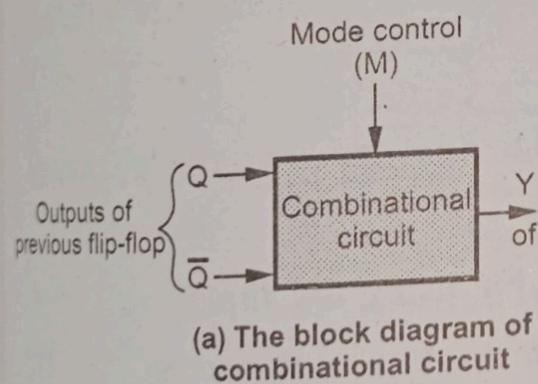
OR Design a 3-bit binary up/down counter. Draw its timing diagram.

[SPPU : Dec.-04, 12, Marks 8]

Ans. : To form an asynchronous up/down counter one control input say M is necessary to control the operation of the up/down counter.

When $M = 0$, the counter will count up and when $M = 1$, the counter will count down. To achieve this the M input should be used to control whether the normal flip-flop output (Q) or the inverted flip-flop output (\bar{Q}) is fed to drive the clock signal of the successive stage flip-flop, as shown in Fig. Q.4.1 (a).

The truth table is shown in Fig. Q.4.1 (b).



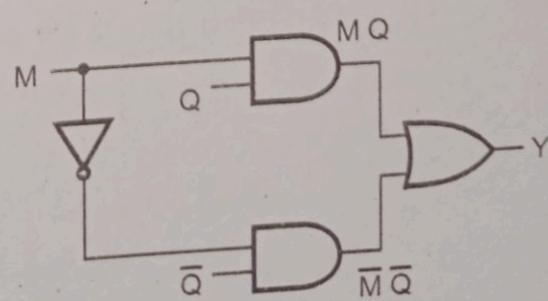
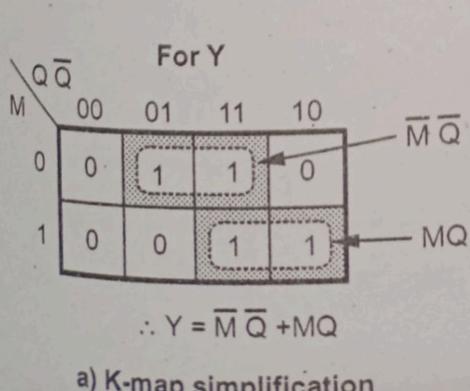
Inputs			Output Y
M	Q	\bar{Q}	
M = 0	0	0	0
	0	1	1
	1	0	0
	1	1	1
M = 1	1	0	0
	1	0	0
	1	1	1
	1	1	1

(b) Truth table

$\left. \begin{array}{l} Y = \bar{Q} \\ \text{for down counting} \end{array} \right\}$

$\left. \begin{array}{l} Y = Q \\ \text{for up counting} \end{array} \right\}$

Fig. Q.4.1



b) Logic diagram

Fig. Q.4.2

- Fig. Q.4.3 shows the 3-bit up/down counter that will count from 000 up to 111 when the mode control input M is 1 and from 111 down to 000 when mode control input M is 0.

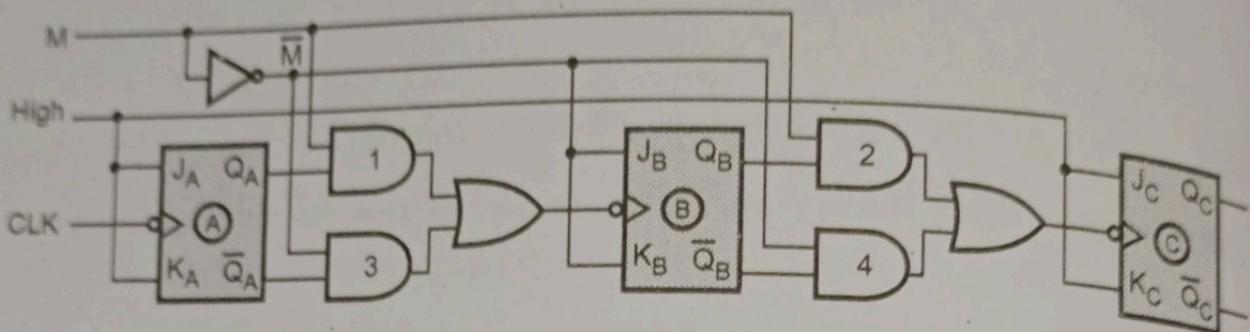


Fig. Q.4.3 3-bit asynchronous up/down counter

- A logic 1 on M enables AND gates 1 and 2 and disables AND gates 3 and 4. This allows the Q_A and Q_B outputs to drive the clock inputs of their respective next stages. So that counter will count up.
- When M is logic 0, AND gates 1 and 2 are disabled and AND gate 3 and 4 are enabled. This allows the \bar{Q}_A and \bar{Q}_B outputs to drive the clock inputs of their respective next stages so that counter will count down.
- Fig. Q.4.4 shows the timing diagram for 3-bit up/down ripple counter. (See Fig. Q.4.4 on next page page.)

Q.5 Design BCD (mod-10) ripple counter using JK flip-flop.

Ans. : Step 1 : Determine the number of flip-flops needed. The BCD counter goes through states 0-9, i.e. total 10 states. Thus, $N = 10$ and for $2^n \geq N$, we need $n = 4$, i.e. 4 flip-flops required.

Step 2 : Type of flip-flops to be used : JK

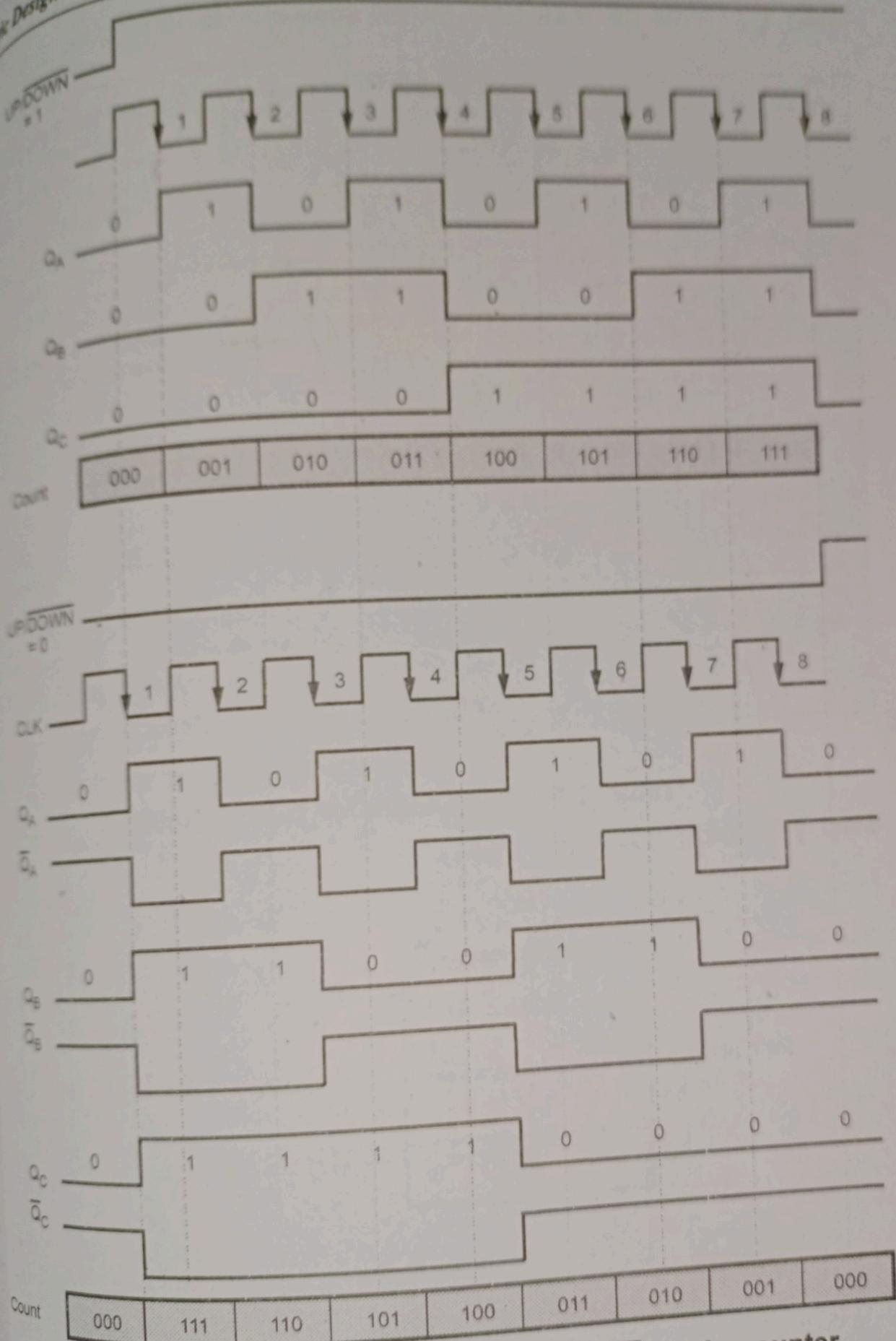


Fig. Q.4.4 Timing diagram for 3-bit UP/ DOWN ripple counter

Step 3 : Write the truth table for the counter

CLK	D	C	B	A	Output of reset logic Y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
-	1	0	1	0	0
-	1	0	1	1	0
-	1	1	0	0	0
-	1	1	0	1	0
-	1	1	1	0	0
-	1	1	1	1	0

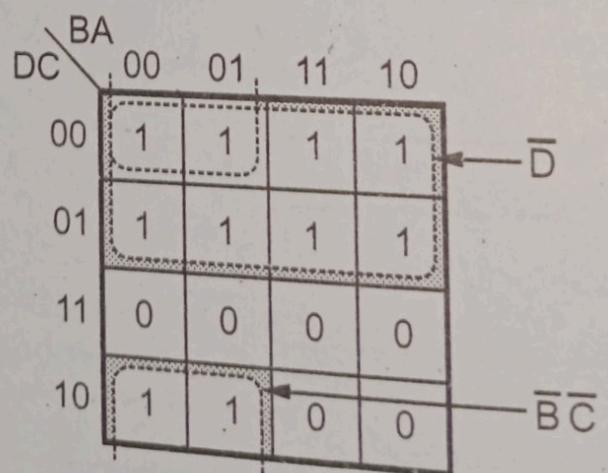
Valid states

Invalid states

Note : The reset input (CLR) of each Flip-Flop is active-low input. By making CLR input of all Flip-Flops logic 0, we can reset the counter. Thus reset logic is designed such a way that for invalid states, $Y = 0$ and counter resets.

Table Q.5.1 Truth table for BCD counter

Step 4 : Derive reset logic



$$\therefore Y = \overline{D} + \overline{B}\overline{C}$$

Fig. Q.5.1

Step 5 : Draw logic diagram.

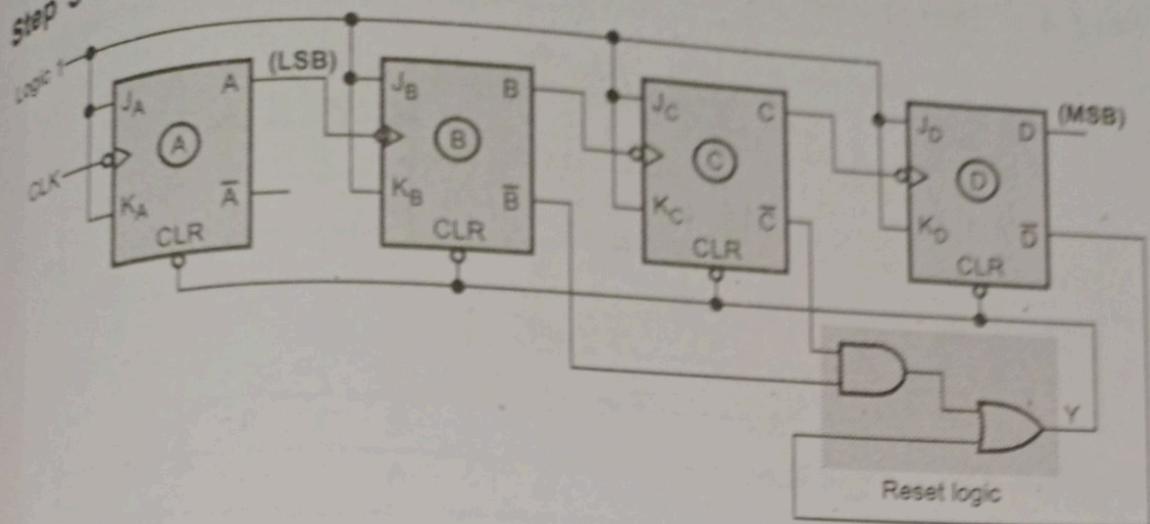


Fig. Q.5.2 Logic diagram of BCD ripple counter

Q.6 Design mod 6 ripple counter using T flip-flops.

Ans. : Step 1 : Determine the number of flip-flop required. Here, counter goes through 0 - 5 states, i.e., total 6 states. Thus $N = 6$ and for $2^n \geq N$ we need $n = 3$, i.e. 3 flip-flops.

Step 2 : Type of flip-flops to be used : T

Step 3 : Write the truth table for counter

CLK	C	B	A	Output of reset logic Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
-	1	1	0	0
-	1	1	1	0

Valid states Invalid states

Fig. Q.6.1

Step 4 : Derive reset logic

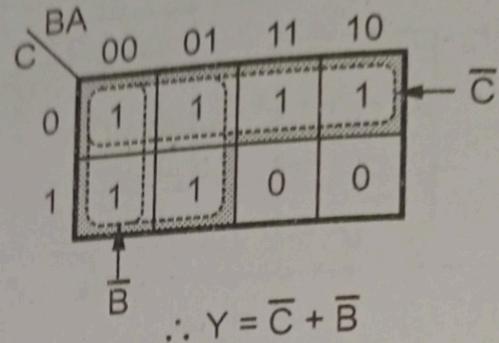


Fig. Q.6.2

Step 5 : Draw logic diagram

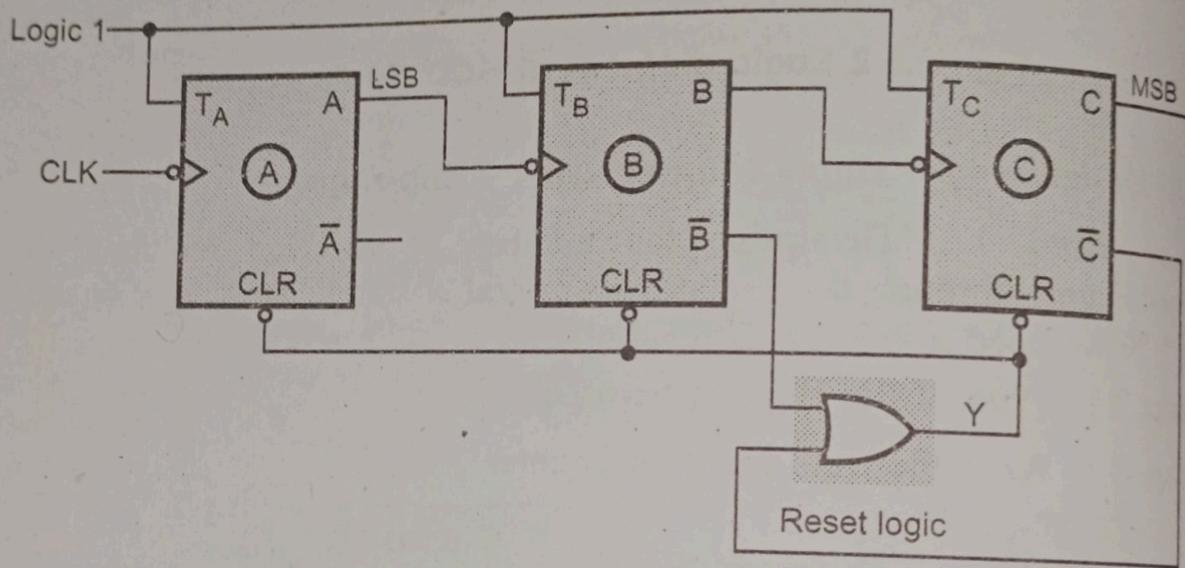


Fig. Q.6.3

Q.7 Design ripple counter for state diagram shown.

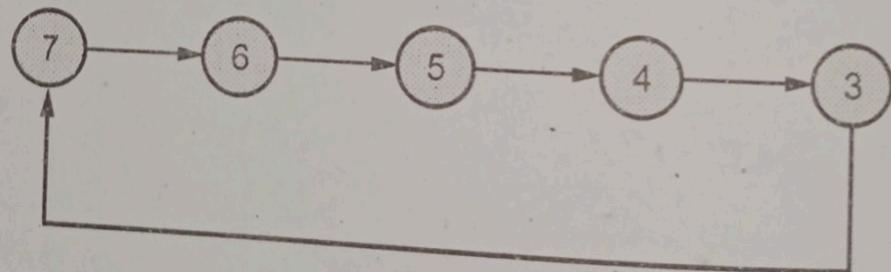


Fig. Q.7.1

Ans. : Step 1 : Determine the number of flip-flops needed.
We know that $2^n \geq N$. Here, $N = 8 \therefore n = 3$ i.e. 3 flip-flops needed.

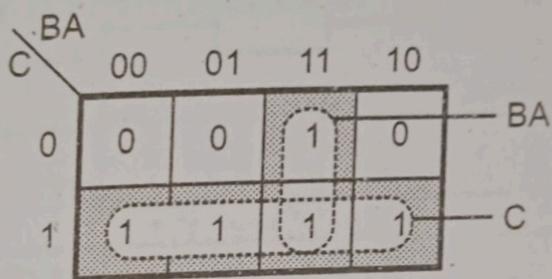
Step 2 : Type of flip-flops to be used : JK

Step 3 : Write truth table for counter.

CLK	C	B	A	Output of reset logic Y
0	1	1	1	1
1	1	1	0	1
2	1	0	1	1
3	1	0	0	1
4	0	1	1	1
5	0	1	0	0
6	0	0	1	0
7	0	0	0	0

Table Q.7.1 Truth table

Step 4 : Derive preset logic. Since it is a down counter we need to derive preset logic instead of reset logic.



$$Y = BA + C$$

Fig. Q.7.2

Step 5 : Draw logic diagram.

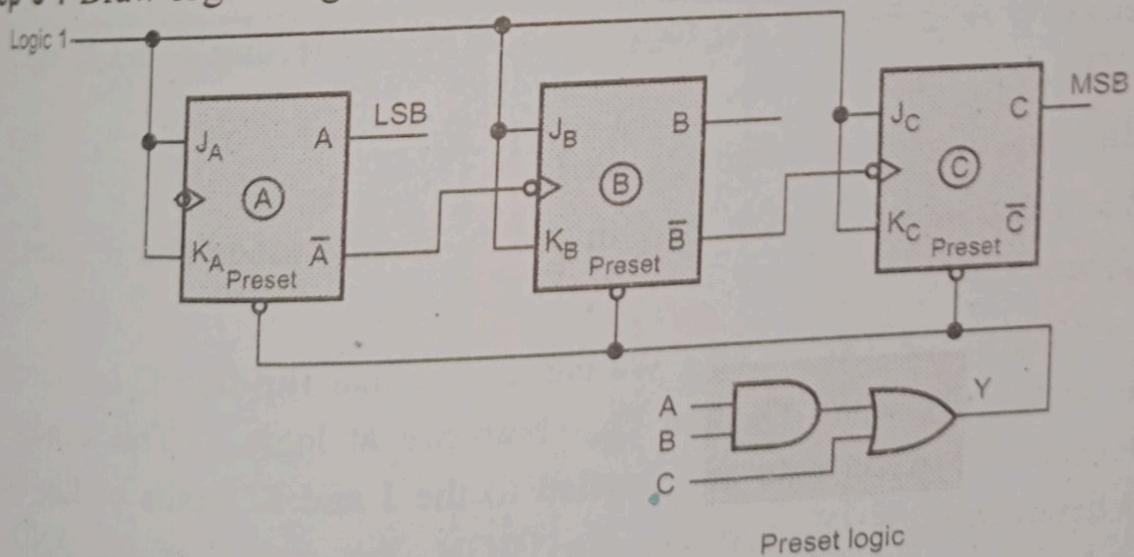


Fig. Q.7.3

Since it is a down counter, the clock of the next flip-flop is given by the \bar{Q} output of the previous flip-flop.

8.3 : Synchronous Counters

Q.8 Draw and explain the working of 3-bit synchronous counter.

Ans. : Fig. Q.8.1 (a) shows 3-bit synchronous binary counter and its timing diagram.

- The state sequence for this counter is shown in Table Q.8.1. (Refer Table Q.8.1 on next page)

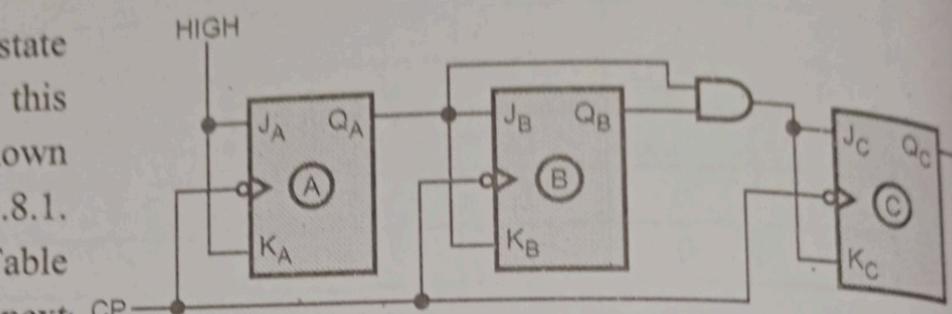


Fig. Q.8.1 (a) A three-bit synchronous binary counter

- Looking at Fig. Q.8.1(b), we can see that Q_A changes on each clock pulse as we progress from its original state to its final state and then back to its original state.

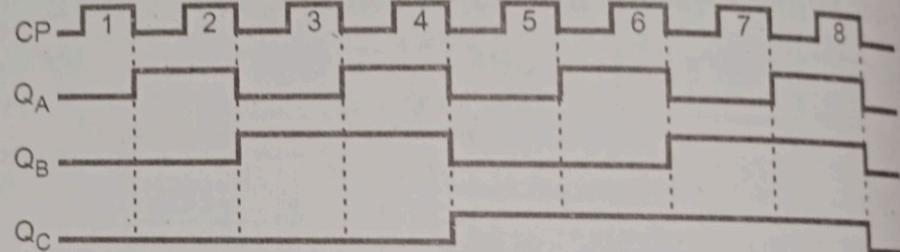


Fig. Q.8.1 (b) Timing diagram for 3-bit synchronous binary counter

- Flip-flop A is held in the toggle mode by connecting J and K inputs to HIGH.
- Flip-flop B toggles, when Q_A is 1.
- When Q_A is 0, flip-flop B is in the no-change mode and remains in its present state.
- Looking at the Table Q.8.1 we can notice that flip-flop C has to change its state only when Q_B and Q_A both are at logic 1. This condition is detected by AND gate and applied to the J and K inputs of flip-flop C. Whenever both Q_A and Q_B are HIGH, the output of the AND gate makes the J and K inputs of flip-flop C HIGH and flip-flop C toggles on the following clock pulse. At all other times, the J and K inputs of flip-flop C are held LOW by the AND gate output and flip-flop does not change state.

CP	Q _C	Q _B	Q _A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Table Q.8.1 State sequence for 3-bit binary counter

Design and implement 3 bit synchronous counter using JK FF.

[SPPU : May-12, Marks 8]

Ans. : Design of 3-bit synchronous binary up-counter :

Step 1 : Number of flip-flops : 3-bit counter so we require 3 flip-flops.

Step 2 : Types of flip-flops to be used JK

Decoder	Present state			Next state			Flip-flops inputs					
	C	B	A	C ₊₁	B ₊₁	A ₊₁	J _C	K _C	J _B	K _B	J _A	K _A
0	0	0	0	0	0	1	0	×	0	×	1	×
1	0	0	1	0	1	0	0	×	1	×	×	1
2	0	1	0	0	1	1	0	×	×	0	1	×
3	0	1	1	1	0	0	1	×	×	1	×	1
4	1	0	0	1	0	1	×	0	0	×	1	×
5	1	0	1	1	1	0	×	0	1	×	1	1
6	1	1	0	1	1	1	×	0	×	0	1	1
7	1	1	1	0	0	0	×	1	1	1	1	1

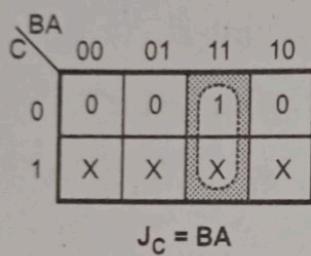
Table Q.9.1 Excitation table

Step 3 : Determine the excitation table for the counter.

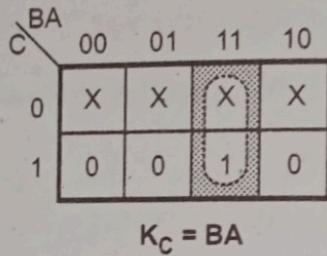
Q_n	Q_{n+1}	J	K
0	0	0	\times
0	1	1	\times
1	0	\times	1
1	1	\times	0

Table Q.9.2 Jk flip-flop excitation table

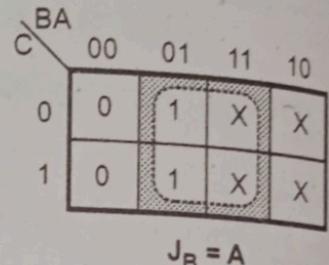
K-map simplification :



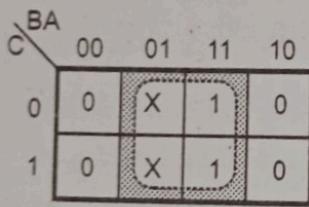
$$J_C = BA$$



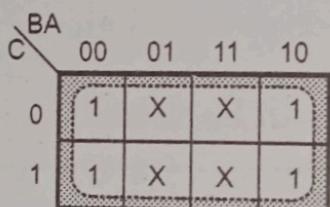
$$K_C = BA$$



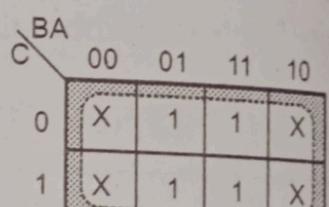
$$J_B = A$$



$$K_B = A$$



$$J_A = 1$$



$$K_A = 1$$

Fig. Q.9.1

Step 5 : Draw logic diagram.

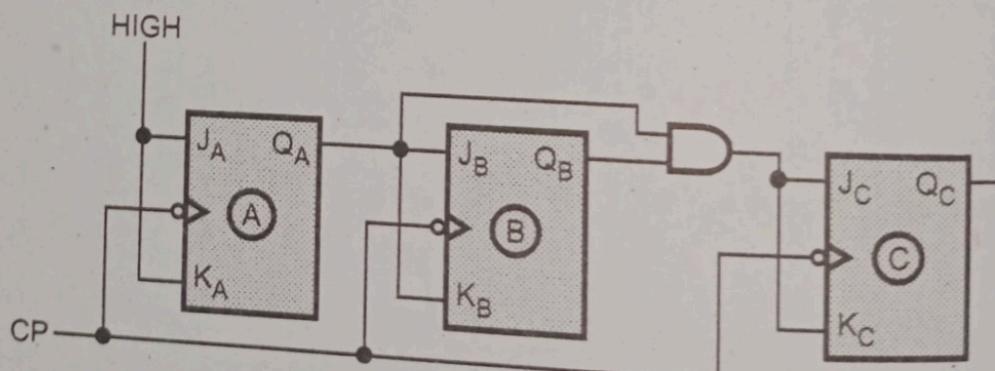


Fig. Q.9.2 A three-bit synchronous binary counter

Q.10 Design divide by 6 counter using T-flip-flops. Write state table and reduce the expression using K-map.

☞ [SPPU : Dec.-09, May-14, Marks 8]

Ans. :

Step 1 : Determine the number of flip-flops needed.

For designing mod 6 counter using the formula

$$2^n \geq N$$

Here

$$N = 6 \therefore n = 3 \quad \text{i.e. 3 flip-flops are required.}$$

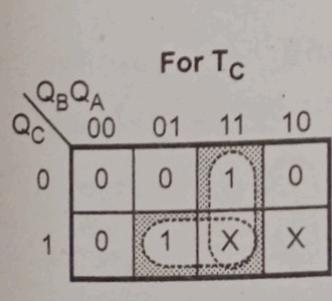
Step 2 : Type of flip-flops to be used : T**Step 3 :** Determine the excitation table for counter.

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

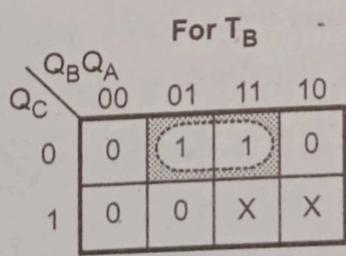
Table Q.10.1 Excitation table for T-flip-flop

CP	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_C	T_B	T_A
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	0	0	1	1	1
4	1	0	0	1	0	1	0	0	1
5	1	0	1	0	0	0	1	0	1

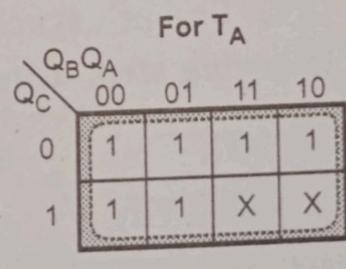
Table Q.10.2 Excitation table for counter

Step 4 : K-map simplification.

$$T_C = Q_C Q_A + Q_B Q_A$$



$$T_B = \overline{Q}_C Q_A$$



$$T_A = 1$$

Fig. Q.10.1

Step 5 : Draw the logic diagram.

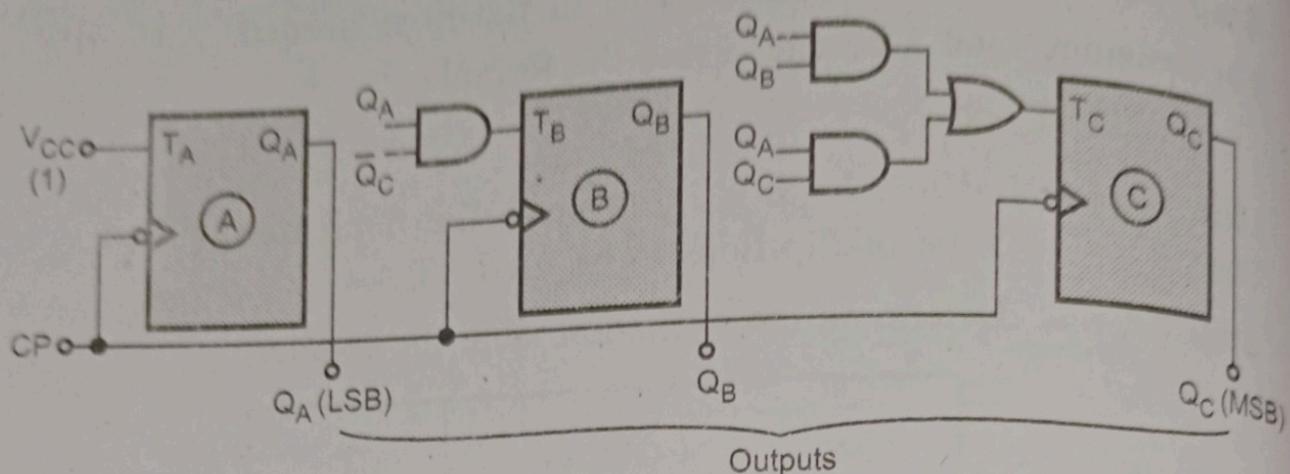


Fig. Q.10.2 Logic diagram

Q.11 What is lock-out condition and bushless circuit ?

Ans. : In a counter if the next state of some unused state is again an unused state and if by chance the counter happens to find itself in the unused states and never arrived at a used state then the counter is said to be in the lockout conditions. This is illustrated in the Fig. Q.11.1. The counter which never goes in lockout condition is called self starting counter.

The circuit that goes in lockout condition is called bushless circuit.

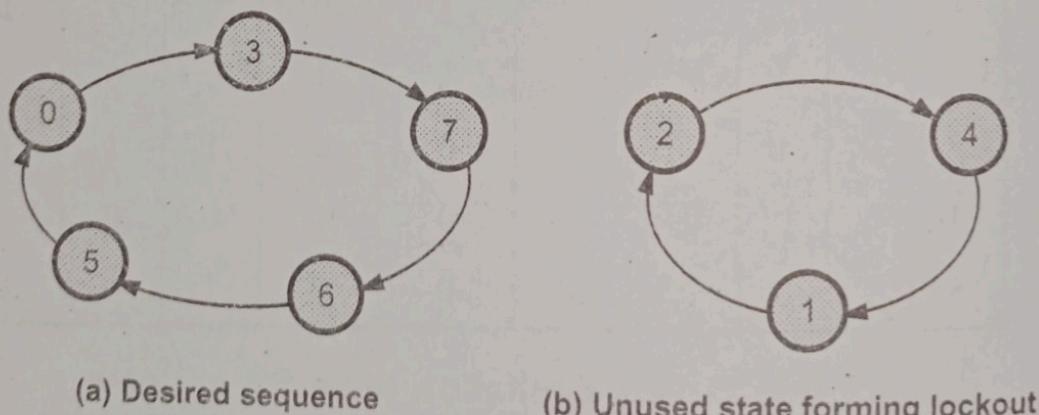
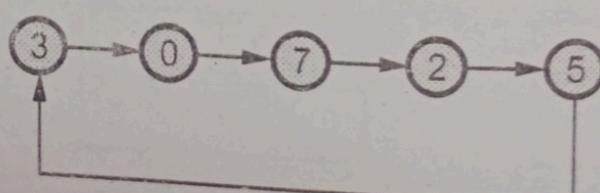


Fig. Q.11.1

Q.12 Design synchronous counter which will go through the following step, using JK flip-flop. (Avoid lock out condition.).



Ans. : The Fig. Q.12.1 shows the state diagram for the given counter. To avoid lock-out condition states 1, 4 and 6 are forced to enter into state 3.

Flip-flop excitation table is as shown below.

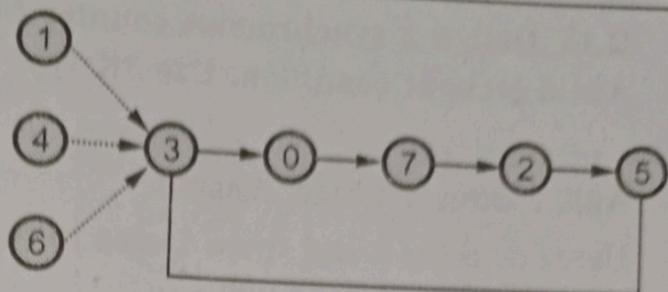


Fig. Q.12.1

Present state			Next state		
A	B	C	A_{+1}	B_{+1}	C_{+1}
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	0	1	0

Table Q.12.1

K-map simplification

For D_A

A	BC		
	00	01	11
0	1	0	1
1	0	0	0

$$D_A = \overline{A} \overline{C}$$

For D_B

A	BC		
	00	01	11
0	1	1	0
1	1	1	1

$$D_B = \overline{B} + A$$

For D_C

A	BC		
	00	01	11
0	1	1	0
1	1	1	1

$$D_C = \overline{B} + \overline{C}$$

Fig. Q.12.2

Logic diagram

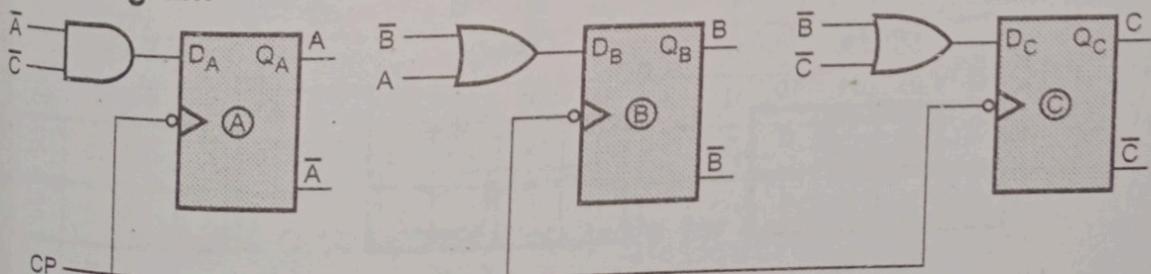


Fig. Q.12.3

Q.13 Design a synchronous counter for $4 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 1 \rightarrow 4 \dots$
 Avoid lockout condition. Use JK type design.

[SPPU : Dec.-15, Marks 6]

Ans. : Step 1 : State diagram

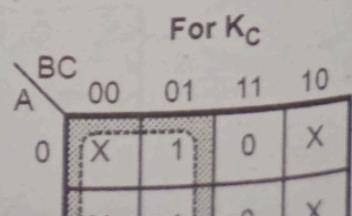
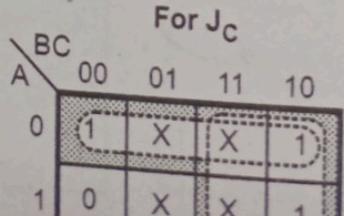
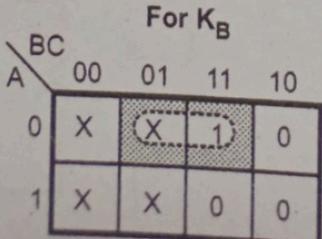
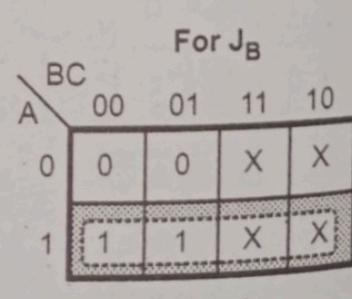
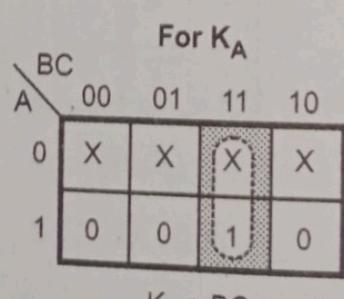
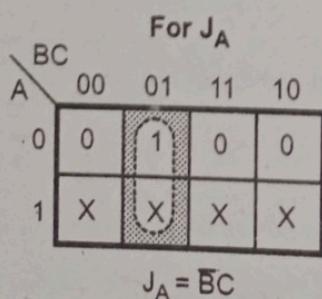
Here, states 5, 2 and 0 are forced to go into 6, 3 and 1 state, respectively to avoid lockout condition.

Step 2 : Excitation table

Present states			Next states			Flip-flop inputs					
A	B	C	A_{+1}	B_{+1}	C_{+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	1	0	0	1	X	0	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	1	1	X	1	X	0	X	0

Table Q.13.1

Step 3 : K-map simplification



Step 4 : Logic diagram

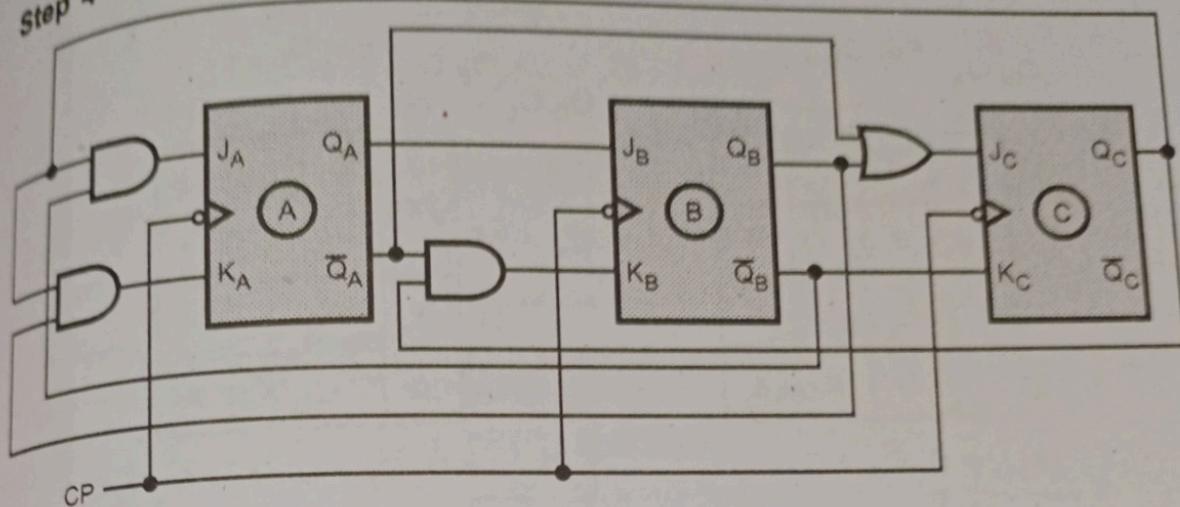


Fig. Q.13.2

Q.14 Design a synchronous decade counter using D flip-flop.

Ans. : The decade counter is a mod-10 counter. It has ten states : 0 - 9.

Step 1 : Determine the number of flip-flops needed.

We know that $2^n \geq N$. Here, $N = 10 \therefore n = 4$ i.e. 4 flip-flops needed.

Step 2 : Types of flip-flops to be used : D

Step 3 : Determine the excitation table for counter.

Present state				Next state			
Q_D	Q_C	Q_B	Q_A	Q_{D+1}	Q_{C+1}	Q_{B+1}	Q_{A+1}
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

Table Q.14.1 Excitation table for counter

Step 4 : K-map simplification

		For $Q_D + 1$					
		Q_B	Q_A	00	01	11	10
Q_D	Q_C	00	0	0	0	0	
	01	0	0	1	0		
	11	X	X	X	X		
	10	1	0	X	X		

$$Q_{D+1} = Q_D \bar{Q}_A + Q_C Q_B Q_A$$

		For $Q_C + 1$					
		Q_B	Q_A	00	01	11	10
Q_D	Q_C	00	0	0	1	0	
	01	1	1	0	0	1	
	11	X	X	X	X	X	
	10	0	0	X	X	X	X

$$Q_{C+1} = Q_C \bar{Q}_B + Q_C \bar{Q}_A + \bar{Q}_C Q_B Q_A$$

		For $Q_B + 1$					
		Q_B	Q_A	00	01	11	10
Q_D	Q_C	00	0	1	0	1	
	01	0	1	0	0	1	
	11	X	X	X	X	X	
	10	0	0	X	X	X	

$$Q_{B+1} = \bar{Q}_D \bar{Q}_B Q_A + Q_B \bar{Q}_A$$

		For $Q_A + 1$					
		Q_B	Q_A	00	01	11	10
Q_D	Q_C	00	1	0	0	1	
	01	1	0	0	0	1	
	11	X	X	X	X	X	
	10	1	0	X	X	X	X

$$Q_{A+1} = \bar{Q}_A$$

Fig. Q.14.1

$$Q_{D+1} = Q_D \bar{Q}_A + Q_C Q_B Q_A$$

$$Q_{D+1} = Q_D \bar{Q}_A + Q_C Q_B Q_A$$

$$Q_{C+1} = Q_C \bar{Q}_B + Q_C \bar{Q}_A + \bar{Q}_C Q_B Q_A$$

$$Q_{B+1} = \bar{Q}_D \bar{Q}_B Q_A + Q_B \bar{Q}_A$$

$$Q_{A+1} = \bar{Q}_A$$

Step 5 : Draw the logic diagram.

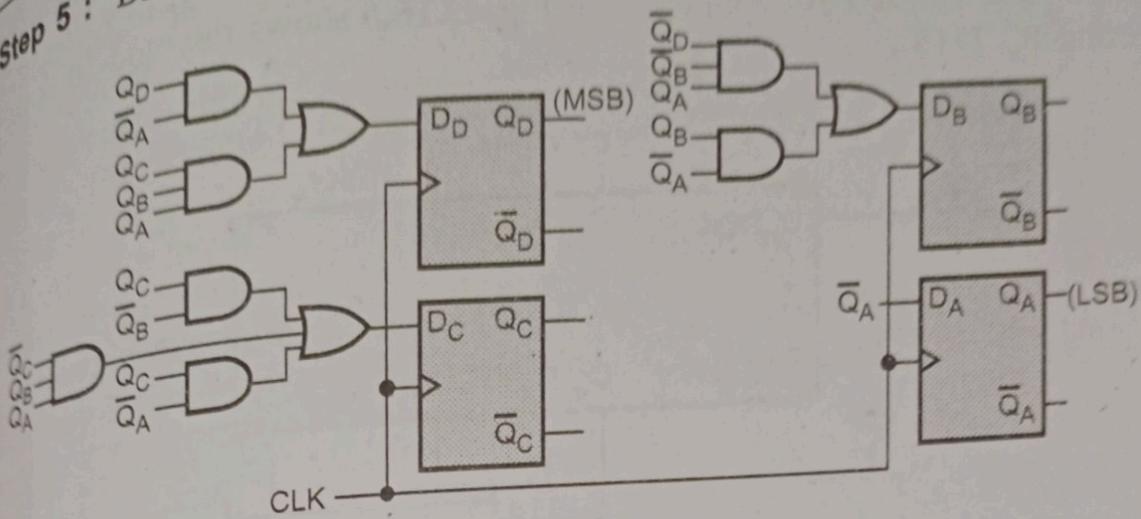


Fig. Q.14.2 Logic diagram

8.4 : Study of 7490 Modulus - n Counter IC

Q.15 Design a MOD-11 counter using IC 74191. Use RC signal.

[SPPU : May-06, Dec.-15, Marks 6]

Ans. : IC 74191 is a 4-bit counter. Thus it is MOD-16 counter. However, we require MOD-11 counter. The difference between 16 and 11 is 5. Hence 5 steps must be skipped from the full modulus sequence. This can be achieved by presetting counter to value 5. Each time when counter recycles it starts counting from 5 upto 16 on each full cycle. Therefore, each full cycle of the counter consists of 11 states.

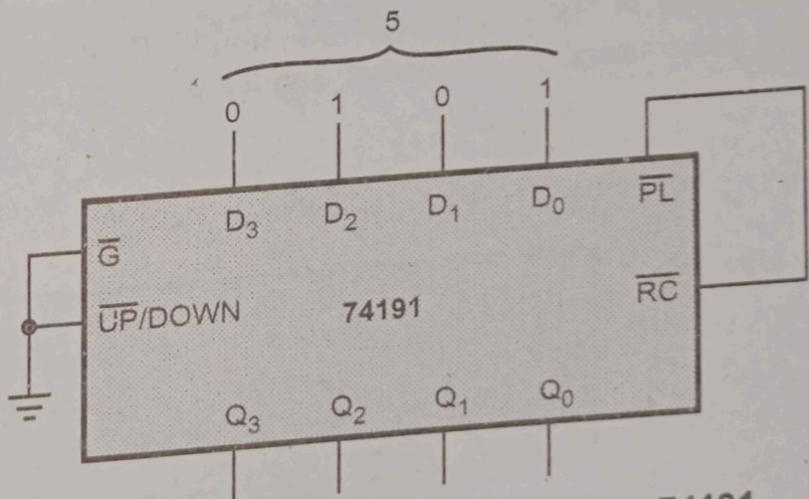


Fig. Q.15.1 MOD-11 counter using 74191

Q.16 How output frequency, f_{out} and clock frequency, f_{CLK} are related in case of binary counter, IC 74191 in up and down counting mode ? If f_{CLK} = 500 Hz and f_{out} = 50 Hz, design the programmable frequency divider using IC 74191 in up counting mode.

[SPPU : Dec.-05, Marks 6]

Ans. : The IC 74191 is a 4-bit binary counter, therefore $f_{out} = f_{CLK}/16$ in up and down counting mode. If $f_{CLK} = 500$ Hz and $f_{out} = 50$ Hz we need mod 10 (500/50) counter. The Fig. Q.16.1 shows the mod-10 counter using IC 74191.

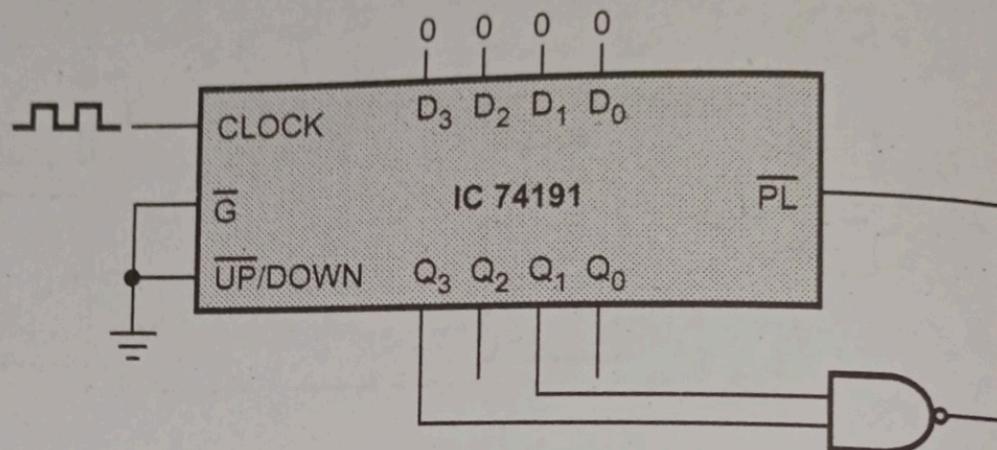


Fig. Q.16.1

Q.17 Explain how IC 74191 can also be used as programmable frequency divider.

[SPPU : Dec.-07, Marks 2]

Ans. : IC 74191 is a 4-bit binary counter. Thus it divides the input frequency by 16. However, we can design MOD-N counter using IC 74191. For MOD-N counter the output frequency will be $f_{out} = \frac{f_{in}}{N}$. Thus by changing N we can change the output frequency. The Fig. Q.17.1 shows the programmable frequency divider using IC 74191.

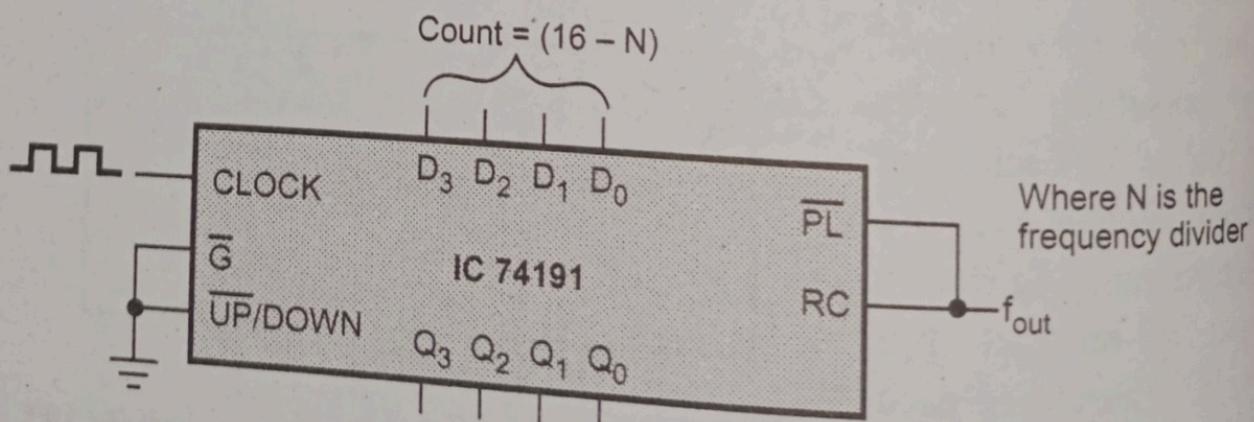


Fig. Q.17.1 Programmable frequency divider

Q.18 Design divide-by-2 for upcounting and divide by 5 for down-counting using frequency divider IC 74191.

[SPPU : May 08, Marks 6]

Ans. : Divide-by-2
for up counting :

Divide-by-2 is a mod-2 counter. Since, after preset above counter goes through 2 states 1110 and 1111, it is a mod-2 counter. Thus, above

circuit is a divide by 2 counter for up counting mode.

Divide-by-5 for down counting mode :

Q_3	Q_2	Q_1	Q_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Truth table

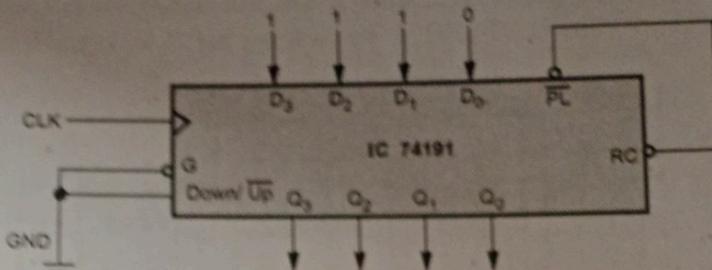
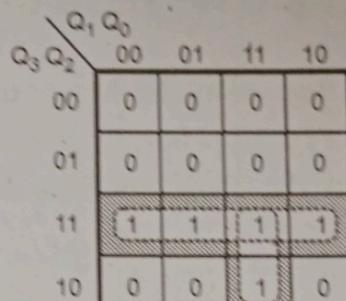


Fig. Q.18.1



$$Y = Q_3 Q_2 + Q_3 Q_1 Q_0$$

Fig. Q.18.2 K-map simplification

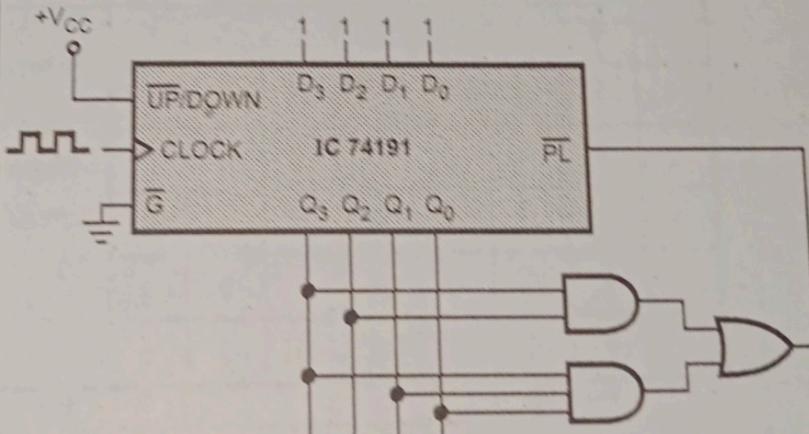


Fig. Q.18.3 Logic diagram

8.5 : Ring and Johnson Counters

Q.19 Draw a six stage ring counter and explain its operation. Mention about the use of presetting the counter.

[SPPU : Dec.-08, Dec.-15, May-17, Marks 6]

Ans. : The Fig. Q.19.1 shows the six stage ring counter. The counter is present to value $(000001)_2$ by setting bit 0 = 1 and remaining bits = 0.

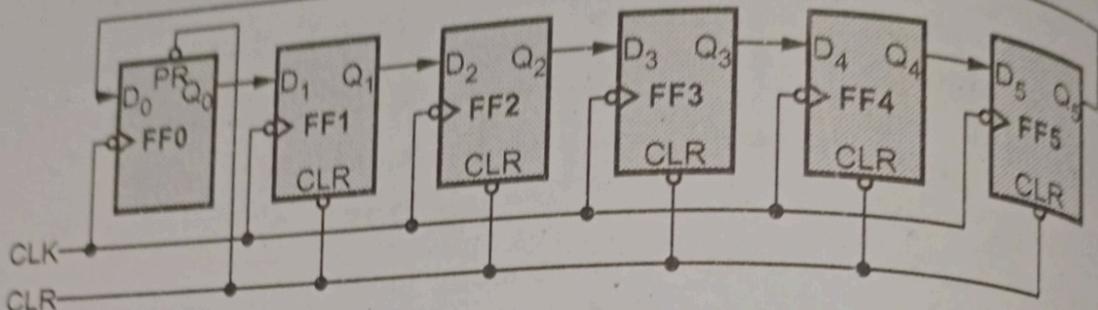


Fig. Q.19.1 Six stage ring counter

Operation : The Fig. Q.19.2 shows the operation of six-stage ring counter. On preset, FF0 (flip-flop 0) is set and FF1 to FF5 are reset. After each falling edge of the clock contents of ring counter are shifted 1 bit from LSB to MSB.

CLR	CLK	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Operation
U	X	1	0	0	0	0	0	Preset
1	↓	0	1	0	0	0	0	'1' Bit follows a circular path to form ring counter
1	↓	0	0	1	0	0	0	
1	↓	0	0	0	1	0	0	
1	↓	0	0	0	0	1	0	
1	↓	0	0	0	0	0	1	
1	↓	1	0	0	0	0	0	

Fig. Q.19.2 Illustrating operation of six-stage ring counter
Q.20 Draw and explain the operation of 4-bit Johnson counter.

[SPPU : Dec.-08, Dec.-16, Marks 3]

OR Draw 4-bit twisted ring counter using D flip-flop. Consider initially all flip-flop. Consider initially all flip-flops are cleared. What

will be the output after 5th clock pulse and prove that the modulus of this twisted ring counter is 8.

WSET [SPPU : May-16, Marks 6]

- Ans. • In a Johnson counter, the Q output of each stage of flip-flop is connected to the D input of the next stage.
 • The single exception is that the complement output of the last flip-flop is connected back to the D-input of the first flip-flop as shown in Fig. Q.20.1.

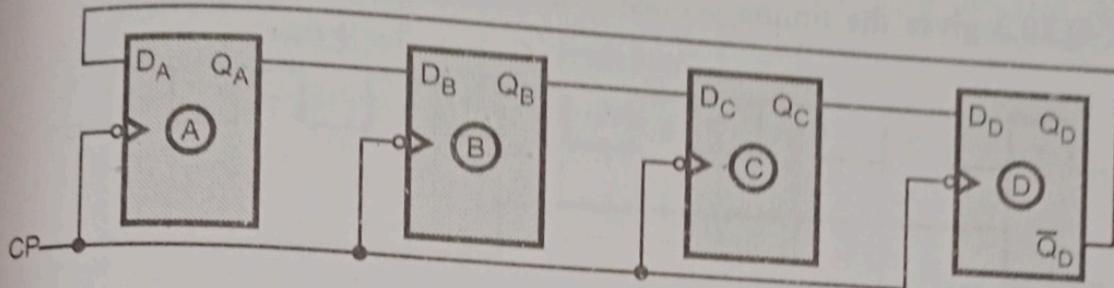


Fig. Q.20.1 Four-bit Johnson counter

Note : Johnson counter can be implemented with SR or JK flip-flops as well.

- As shown in Fig. Q.20.1 there is a feedback from the rightmost flip-flop complement output to the leftmost flip-flop input. This arrangement produces a unique sequence of states.
- Initially, the register (all flip-flops) is cleared. So all the outputs, Q_A , Q_B , Q_C , Q_D are zero.
- The output of last stage, Q_D is zero. Therefore complement output of last stage, \bar{Q}_D is one. This is connected back to the D input of first stage. So D_A is one.
- The first falling clock edge produces $Q_A = 1$ and $Q_B = 0$, $Q_C = 0$, $Q_D = 0$ since D_B , D_C , D_D are zero.

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

Table Q.20.1 Four-bit Johnson sequence