UNIT - 3

SYNCHRONOUS SEQUENTIAL CIRCUITS

3.1 SEQUENTIAL LOGIC CIRCUITS

The output of a combinational logic depends on the input levels, whereas the output of a sequential logic depends on stored levels and also the input levels. A sequential circuit consists of a combinational circuit and a memory element.

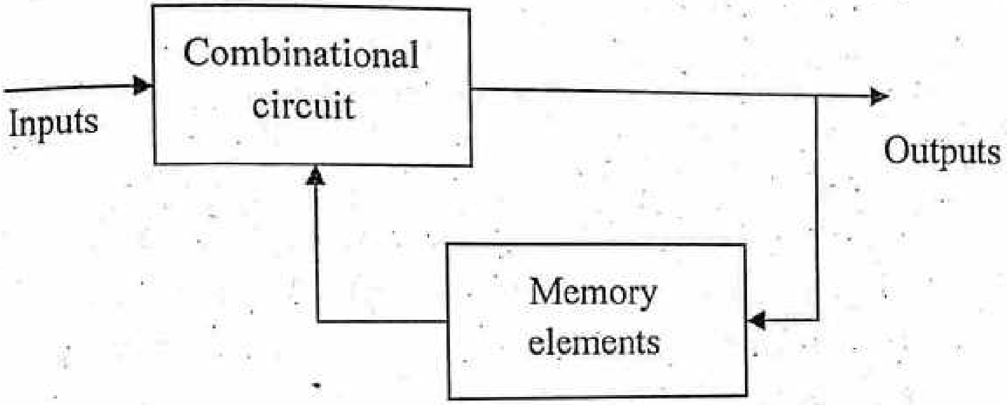


Figure 3.1 Block diagram of sequential Logic circuit

The information stored in the memory elements at any given time defines the present state of the sequential circuits. The present state and the inputs determine the next state and the outputs of the sequential circuit. Thus we can specify the sequential circuit by a time sequence of external inputs, present states, next states and outputs

SI.No	Combinational circuits	Sequential circuits
1.	The output depends on input only.	The output depends on present inputs and past outputs.
2.	Memory elements are not required.	Memory elements are required to store the past outputs.
3.	Combinational circuits are easy to design since it contains only gates	Sequential circuits are harder to design.
4.	Combinational circuits are faster in speed.	Sequential circuits are slower than the combinational circuits
5.	Examples of combinational circuits are parallel adder, half adders etc	Examples of sequential circuits are counters, shift registers etc.

Table 3.1 Difference between combinational and sequential circuits

There are two types of sequential circuits.

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

Asynchronous sequential circuits

In asynchronous sequential circuit the outputs depend upon the order in which its input variables change and can be affected at any instant of time.

Synchronous sequential circuits

In synchronous sequential circuit the outputs depend upon the order in which its input variables change and can be affected at discrete instants of time (based on the activation of clock signals).

Sl.No	Synchronous sequential circuits	Asynchronous Sequential circuits
1.	In synchronous circuits, the change in input can affect memory element based upon the activation of clock signal.	In asynchronous circuits the change in input can affect memory element at any instant of time.
2.	The speed of operation depends on clock.	Asynchronous circuits can operate faster than synchronous circuits.
3.	In synchronous circuits, memory elements are clocked Flip-flops.	In asynchronous circuits, memory elements are either unclocked Flip-flops or time delay elements.
4.	Synchronous circuits are easier to design.	Asynchronous circuits are difficult to design.

Table 3.2 Difference between synchronous and asynchronous sequential circuits

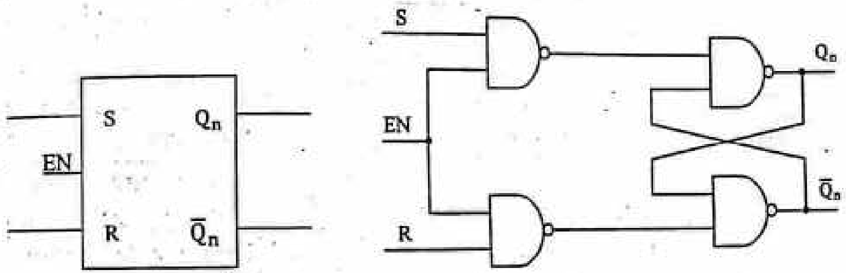
3.2 LATCHES

It is a sequential circuit that checks all of its inputs continuously and changes its outputs at any time. Many times enable signal is provided with the latch. When enable signal is active output changes occur as input changes.

But when enable signal is not activated input changes do not affect the output. Latch is a bistable element that has two stable states logic '0' and logic '1'.Here we can see.

1. SR latch
2. D latch
3. JK latch
4. T latch

3.2.1 SR Latch



(a) Symbol (b) Logic diagram

Figure 3.2 Logic symbol and logic diagram of SR latch

EN	S	R	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	

S	R	Q_{n+1}	State
0	0	Q_n	No change
0	1	0	Reset
1	0	1	Set
1	1	X	Indeterminate

Table 3.3 Truth table of SR Latch

When the enable signal is active and S = 0; R = 0 the state of the latch remains unchanged.

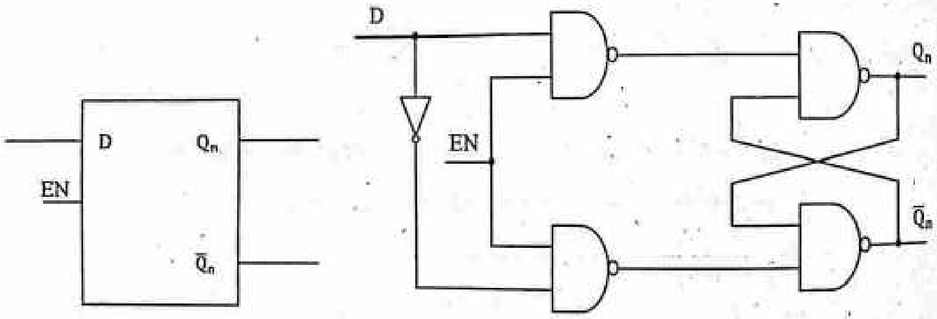
When the enable signal is active and S = 0; R = 1 the output of SR latch is in RESET condition ($Q_n = 0$).

When the enable signal is active and S = 1; R = 0 the output of SR latch is in SET condition. ($Q_n = 1$).

When the enable signal is active and S = 1 ; R = 1 the output is unpredictable. This is called indeterminate condition. When the enable signal is not activated, there will be no change in output. The characteristic equation of SR Latch is same as that of SR Flip-flop, given by

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

3.2.2 D Latch



(a) Symbol

(b) Logic diagram

Figure 3.3 Logic symbol and logic diagram of D-latch

EN	D	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
1	0	0	0	Reset
1	0	1	0	
1	1	0	1	Set
1	1	1	1	

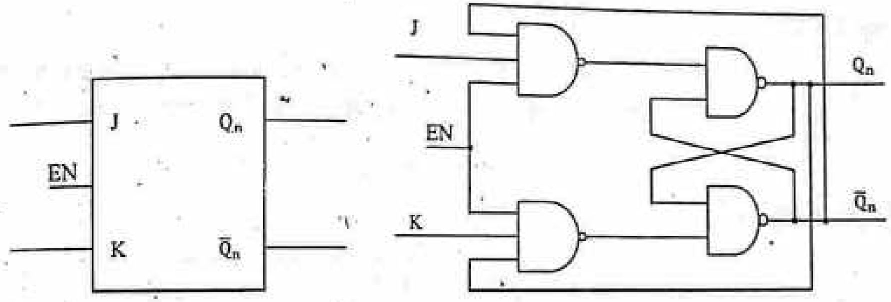
D	Q_{n+1}	State
0	0	Reset
1	1	Set

Table 3.4 Truth table of D-latch

When the Enable signal is active and D = 0, the output of D latch is in RESET condition ($Q_n = 0$).When the Enable signal is active and D = 1, the output of D latch is in SET condition ($Q_n = 1$). When the enable signal is not activated, there will be no change in output. The characteristic equation of D latch is same as that of D Flip-flop.

$$Q_{n+1} = D$$

3.2.3 JK Latch



(a) Symbol

(b) Logic diagram

Figure 3.4 Logic symbol and logic diagram of JK Latch

EN	J	K	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	1	Toggles
1	1	1	1	0	

J	K	Q_{n+1}	State
0	0	Q_n	No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}_n$	Toggles

Table 3.5 Truth table of JK Latch

When the enable signal is active and $J = 0; K = 0$, the output of JK latch remains unchanged.

When the enable signal is active and $J = 0; K = 1$, the output of JK latch is in Reset condition($Q_n = 0$).

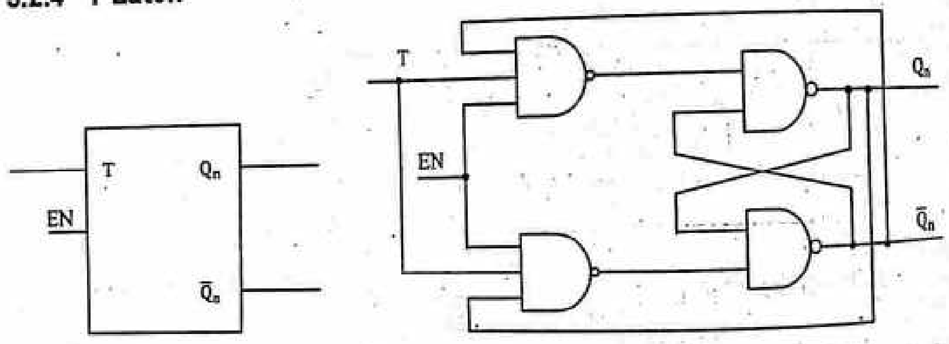
When the enable signal is active and $J = 1; K = 0$, the output of JK latch is in Set condition($Q_n = 1$).

When the enable signal is active and $J = 1; K = 1$, the state of the JK latch is complemented.

When the enable signal is not activated, there will be no change in output. The characteristic equation of JK latch is same as that of JK Flip-flop.

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

3.2.4 T Latch



(a) Symbol (b) Logic diagram

Figure 3.5 Logic symbol and logic diagram of T latch.

EN	T	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
1	0	0	0	No change
1	0	1	1	
1	1	0	1	Toggles
1	1	1	0	

T	Q_{n+1}	State
0	Q_n	No change
1	$\overline{Q}_n$	Toggles

Table 3.6 Truth table of T-latch

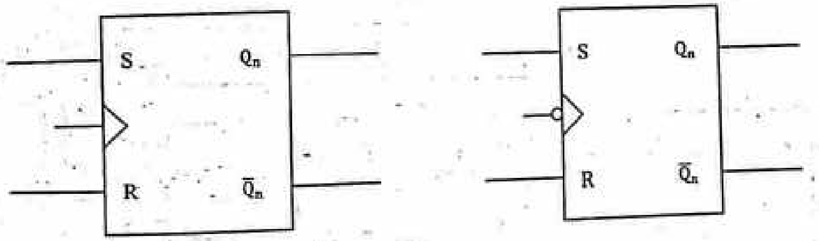
When the Enable signal is active and $T = 0$, the output of T latch remains unchanged. When the Enable signal is active and $T = 1$, the state of T latch is complemented. When the enable signal is not activated, there will be no change in output. The characteristic equation of T latch is same as that of T Flip-flop.

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

3.3 FLIP-FLOPS

It is a sequential device that normally samples its inputs and changes its output only at times determined by clocking signal. Flip-flops are bistable elements. The main difference between latches and Flip-flops is in the method used for changing their states.

3.3.1 SR Flip-flop



(a) Positive edge triggered (b) Negative edge triggered

Figure 3.6 Logic symbol of SR Flip-flop

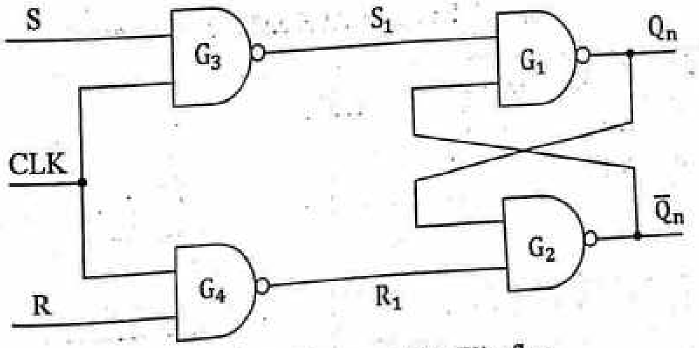


Figure 3.7 Logic diagram of SR Flip-flop

CLK	S	R	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
↑	0	0	0	0	No change
↑	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	

Table 3.7 Truth table for positive edge triggered

S	R	Q_{n+1}	State
0	0	Q_n	No change
0	1	0	Reset
1	0	1	Set
1	1	X	Indeterminate

Table 3.8 Truth table

Case 1: When S = 0 and R = 0

When S = 0, R = 0 and CLK is positive edge triggered S_1 becomes logic '1' and R_1 becomes logic '1'. Initially assume, $Q_n = 1$ and $\overline{Q}_n = 0$. So the inputs of NAND gate G_2 are $Q_n = 1$ and R_1=1. Hence the output of G_2 remains logic '0'. Similarly the inputs of G_1 are S_1=1 and $\overline{Q}_n = 0$. Hence the output of G_1 remains logic '1'. This shows that when S and R both are low the output state does not change (the output does not change even though we initially assume $Q_n = 0$ and $\overline{Q}_n = 1$).

Case 2: When S=0 and R=1

When S = 0, R = 1 and CLK is positive edge triggered S_1 becomes logic '1' and R_1 becomes logic '0'. Initially assume, $Q_n = 1$ and $\overline{Q}_n = 0$. So the inputs of NAND gate G_2 are $Q_n = 1$ and R_1=0. Hence the output of G_2 becomes logic '1'. Similarly the inputs of G_1 are S_1=1 and $\overline{Q}_n = 1$. Hence the output of G_1 resets to logic '0'. Hence if S = 0 and R = 1 the output will be $Q_n = 0$ (the output is in Reset condition even though we initially assume $Q_n = 0$ and $\overline{Q}_n = 1$).

Case 3: When S=1 and R=0

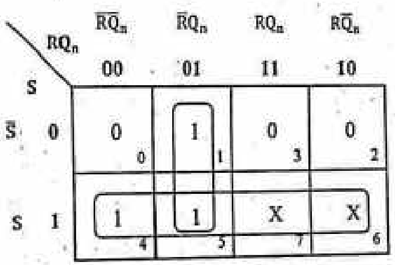
When S = 1, R = 0 and CLK is positive edge triggered S_1 becomes logic '0' and R_1 becomes logic '1'. Initially assume, $Q_n = 1$ and $\overline{Q}_n = 0$. So the inputs of NAND gate G_2 are $Q_n = 1$ and R_1=1. Hence the output of G_2 becomes logic '0'. Similarly the inputs of G_1 are S_1=0 and $\overline{Q}_n = 0$. Hence the output of G_1 sets to logic '1'. Hence if S = 1 and R = 0 the output will be $Q_n = 1$ (the output is in Set condition even though we initially assume $Q_n = 0$ and $\overline{Q}_n = 1$).

Case 4: When S = 1 and R = 1

When S = 1, R = 1 and CLK is positive edge triggered S_1 becomes logic '0' and R_1 becomes logic '0'. Initially assume, $Q_n = 1$ and $\overline{Q}_n = 0$. So the inputs of NAND gate G_2 are $Q_n = 1$ and R_1=0. Hence the output of G_2 becomes logic '1'. Similarly the inputs of G_1 are S_1=0 and $\overline{Q}_n = 1$. Hence the output of G_1 also become logic '1'. Here the outputs Q_n and $\overline{Q}_n$ are logic '1'. This input combination must be avoided. This is called indeterminate condition.

K-map for Q_{n+1}

The characteristic equation of SR Flip-flop can be obtained from the above truth table shown in table 3.7



S \ RQ_n	$\overline{R}\overline{Q}_n$ 00	$\overline{R}Q_n$ 01	RQ_n 11	$R\overline{Q}_n$ 10
$\overline{S}$ 0	0 (0)	1 (1)	0 (3)	0 (2)
S 1	1 (4)	1 (5)	X (7)	X (6)

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

Therefore the characteristic equation of SR Flip-flop is $Q_{n+1} = S + \overline{R}Q_n$

CLK	S	R	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
↓	0	0	0	0	No change
↓	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	

Table 3.9 Truth table for negative edge triggered

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

Table 3.10 Excitation table

0-to-0 Transition:

If the present state of the Flip-flop is '0' and the next state is to remain '0'. This happen when the inputs must be either S = 0, R = 0 or S = 0, R = 1. Thus S has to be at '0', but R can be either level. Thus if S=0 and R= 'X', the output will remain in same state '0'.

0-to-1 Transition:

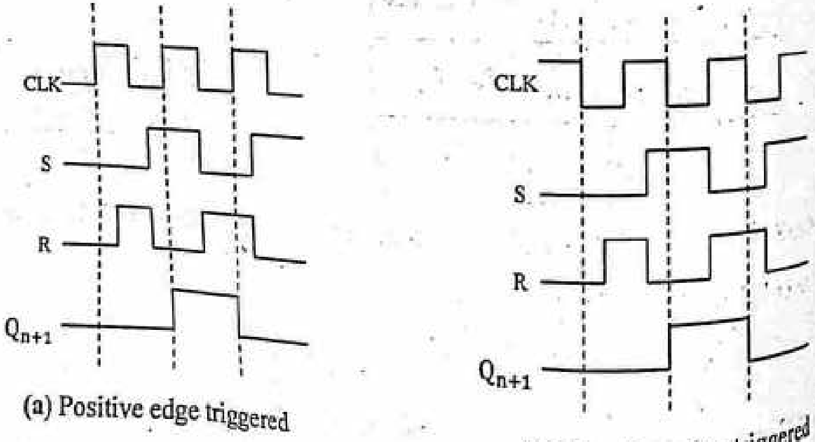
If the present state of the Flip-flop is '0' and the next state is to change '1'. This happen only when the inputs S = 1, R = 0. Thus S has to be at '1' and R has to be '0' for this transition to occur.

1-to-0 Transition:

If the present state of the Flip-flop is '1' and the next state is to change '0'. This happen only when the inputs S = 0, R = 1. Thus S has to be at '0' and R has to be '1' for this transition to occur.

1-to-1 Transition:

If the present state of the Flip-flop is '1' and the next state is to remain '1'. This happen when the inputs must be either S = 1, R = 0 or S = 0, R = 0. Thus R has to be at '0', but S can be either level. Thus if S= 'X' and R= 0, the output will remain in same state '1'.

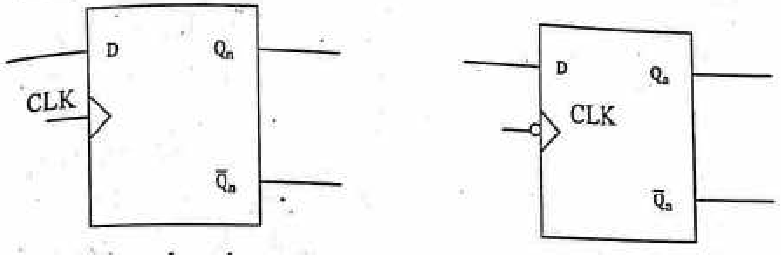


(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.8 Timing diagram of SR Flip-flop

3.3.2 D-Flip-flop



(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.9 Logic symbol of D Flip-flop

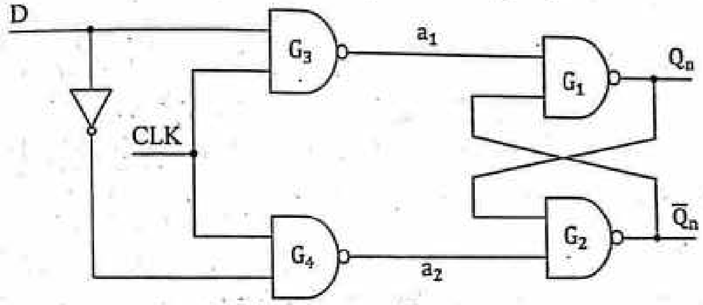


Figure 3.10 Logic diagram of D Flip-flop.

CLK	D	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
↑	0	0	0	Reset
↑	0	1	0	
↑	1	0	1	Set
↑	1	1	1	

Table 3.11 Truth table of positive edge triggered

D	Q_{n+1}	State
0	0	Reset
1	1	Set

Table 3.12 Truth table

Case 1: When D=0

When D = 0 and CLK is positive edge triggered a_1 becomes logic '1' and a_2 becomes logic '0'. Assume initially $Q_n = 1$ and $\overline{Q}_n = 0$. So the inputs of G_2 are $Q_n = 1$ and $a_2 = 0$, therefore the output of G_2 is $\overline{Q}_n = 1$. The inputs of G_1 are $\overline{Q}_n = 1$ and $a_1 = 1$, therefore the output of $G_1$is $Q_n = 0$. Even if we assume initially $Q_n = 0$ and $\overline{Q}_n = 1$ and CLK is positive edge triggered, the output obtained is $Q_n = 0$. Therefore the output Q_n is in reset condition if D = 0.

Case 2: When D=1

When D = 1 and CLK is positive edge triggered a_1 becomes logic '0' and a_2 becomes logic '1'. Assume initially $Q_n = 1$ and $\overline{Q}_n = 0$. So the inputs of G_2 are $Q_n = 1$ and $a_2 = 1$, therefore the output of G_2 is $\overline{Q}_n = 0$. The inputs of G_1 are $\overline{Q}_n = 0$ and $a_1 = 0$, therefore the output of G_1 is $Q_n = 1$. Even if we assume initially $Q_n = 0$ and $\overline{Q}_n = 1$ and CLK is positive edge triggered, the output obtained is $Q_n = 1$. Therefore the output Q_n is in set condition if D = 1.

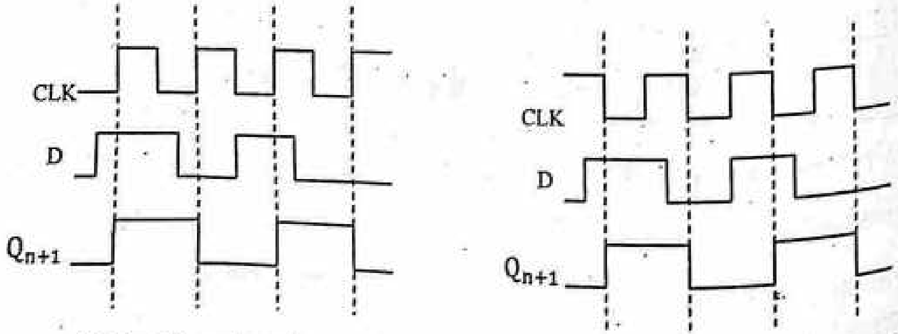
CLK	D	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
↓	0	0	0	Reset
↓	0	1	0	
↓	1	0	1	Set
↓	1	1	1	

Table 3.13 Truth table for negative edge triggered

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

Table 3.14 Excitation table

In D Flip-flop, the next state is always equal to the D input and it is independent of the present state. Therefore D must be '0' if Q_{n+1} has to be at '0' and '1' if Q_{n+1} has to be at '1', independent of the value of Q_n.



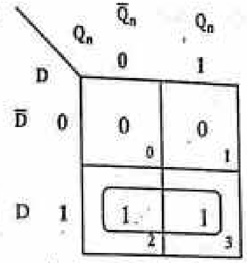
(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.11 Timing diagram of D Flip-flop

K-map for Q_{n+1}

The characteristic equation of D Flip-flop can be obtained from the above truth table shown in table 3.13



$Q_{n+1} = D$

Therefore the characteristic equation for D,Flip-flop is $Q_{n+1} = D$

3.3.3 JK Flip-flop

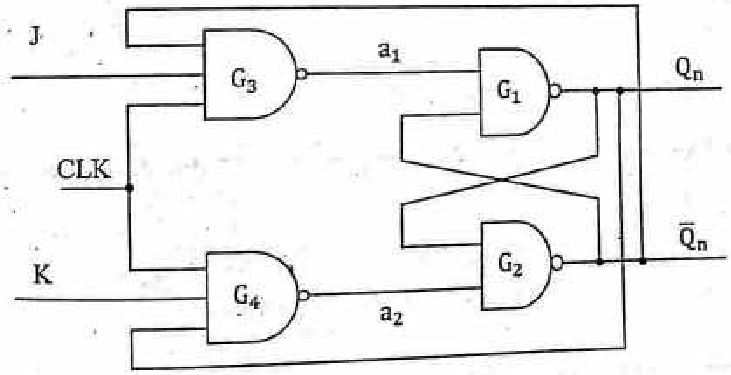
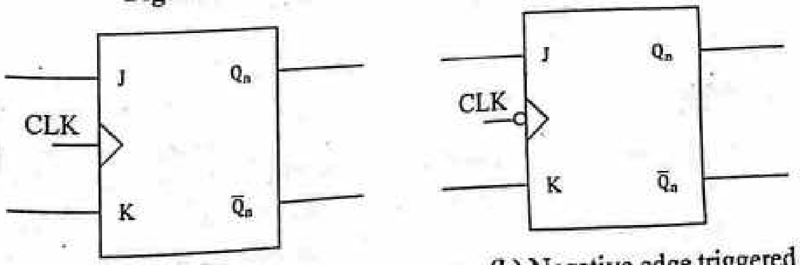


Figure 3.12 Logic diagram of JK Flip-flop



(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.13 Logic symbol of JK Flip-flop

CLK	J	K	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
↑	0	0	0	0	No change
↑	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	Toggles
↑	1	1	1	0	

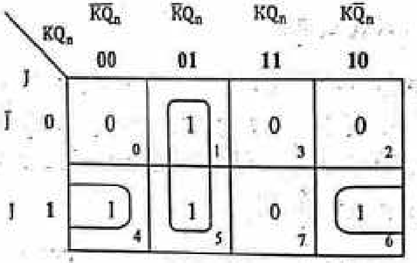
Table 3.15 Truth table of positive edge triggered

J	K	Q_{n+1}	State
0	0	Q_n	No change
0	1	0	Reset
1	0	1	Set
1	1	$\bar{Q}_n$	Toggles

Table 3.16 Truth table

The characteristic equation of JK Flip-flop can be obtained from the above truth table shown in table 3.15

K-map for Q_{n+1}



J \ KQ_n	$\bar{K}\bar{Q}_n$ 00	$\bar{K}Q_n$ 01	KQ_n 11	$K\bar{Q}_n$ 10
$\bar{J}$ 0	0	1	0	0
J 1	1	1	0	1

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

Therefore the characteristic equation of JK Flip-flop is

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

Case 1: When J = 0 and K = 0

Initially assume $Q_n = 1$ and $\bar{Q}_n = 0$. When $J = 0$, $K = 0$ and CLK is positive edge triggered, the inputs of NAND gate G_3 are $J = 0$ and $\bar{Q}_n = 0$. Hence G_3 output is logic high ($a_1 = 1$). The inputs of NAND gate G_4 are $K = 0$ and $Q_n = 1$. Hence G_4 output is also logic high ($a_2 = 1$).

The inputs of NAND gate G_2 are $a_2 = 1$ and $Q_n = 1$, so the output $\bar{Q}_n$ remains logic '0'. The input of NAND gate G_1 are $a_1 = 1$ and $\bar{Q}_n = 0$, so the output Q_n remains logic '1'.

This shows that when J and K both are low, the output state does not change even if we initially assume $Q_n = 0$ and $\bar{Q}_n = 1$.

Case 2: When J = 0 and K = 1

Initially assume $Q_n = 1$ and $\bar{Q}_n = 0$. When $J = 0$, $K = 1$ and CLK is positive edge triggered, the inputs of NAND gate G_3 are $J = 0$ and $\bar{Q}_n = 0$. Hence G_3 output is logic high ($a_1 = 1$). The inputs of NAND gate G_4 are $K = 1$ and $Q_n = 1$. Hence G_4 output is logic low ($a_2 = 0$).

The inputs of NAND gate G_2 are $a_2 = 0$ and $Q_n = 1$, so the output $\bar{Q}_n$ changes to logic '1'. The input of NAND gate G_1 are $a_1 = 1$ and $\bar{Q}_n = 1$, so the output Q_n changes to logic '0'.

Therefore the output Q_{n+1} is in Reset condition if $J = 0$ and $K = 1$, even if we initially assume $Q_n = 0$ and $\bar{Q}_n = 1$.

Case 3: When J = 1 and K = 0

Initially assume $Q_n = 1$ and $\bar{Q}_n = 0$. When $J = 1$, $K = 0$ and CLK is positive edge triggered, the inputs of NAND gate G_3 are $J = 1$ and $\bar{Q}_n = 0$. Hence G_3 output is logic high ($a_1 = 1$). The inputs of NAND gate G_4 are $K = 0$ and $Q_n = 1$. Hence G_4 output is also logic high ($a_2 = 1$).

The inputs of NAND gate G_2 are $a_2 = 1$ and $Q_n = 1$, so the output $\bar{Q}_n$ remains at logic '0'. The input of NAND gate G_1 are $a_1 = 1$ and $\bar{Q}_n = 0$, so the output Q_n sets to logic '1'.

Therefore the output Q_{n+1} is in set condition if $J = 1$ and $K = 0$, even if we initially assume $Q_n = 0$ and $\bar{Q}_n = 1$.

Case 4: When J = 1 and K = 1

Initially assume $Q_n = 1$ and $\bar{Q}_n = 0$. When $J = 1$, $K = 1$ and CLK is positive edge triggered, the inputs of NAND gate G_3 are $J = 1$ and $\bar{Q}_n = 0$. Hence G_3 output is logic high ($a_1 = 1$). The inputs of NAND gate G_4 are $K = 1$ and $Q_n = 1$. Hence G_4 output is logic low ($a_2 = 0$).

The inputs of NAND gate G_2 are $a_2 = 0$ and $Q_n = 1$, so the output $\overline{Q}_n$ gets changed to '1'. The input of NAND gate G_1 are $a_1 = 1$ and $\overline{Q}_n = 1$, so the output Q_n is changed to logic '0'.

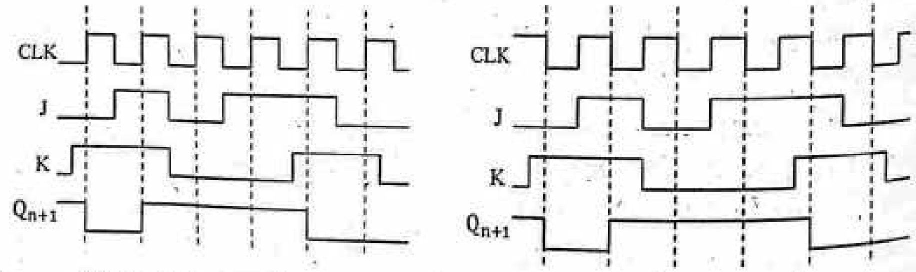
Therefore the output Q_{n+1} is the complement form of Q_n if $J = 1$ and $K = 0$, even if we initially assume $Q_n = 0$ and $\overline{Q}_n = 1$.

CLK	J	K	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
↓	0	0	0	0	No change
↓	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	Toggles
↓	1	1	1	0	

Table 3.17 Truth table for negative edge triggered

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

Table 3.18 Excitation table



(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.14 Timing diagram of JK Flip-flop

0-to-0 Transition:

If the present state of the Flip-flop is '0' and the next state is to remain '0'. This happen when the inputs must be either $J = 0, K = 0$ or $J = 0, K = 1$. Thus J has to be at '0', but K can be either level. Thus if J=0 and K= 'X' the output will remain in same state '0'.

0-to-1 Transition:

If the present state of the Flip-flop is '0' and the next state is to change '1'. This happen when the inputs must be either $J = 1, K = 0$ or $J = 1, K = 1$. Thus J has to be at '1', but K can be either level. Thus if J=1 and K= 'X' this transition will occur.

1-to-0 Transition:

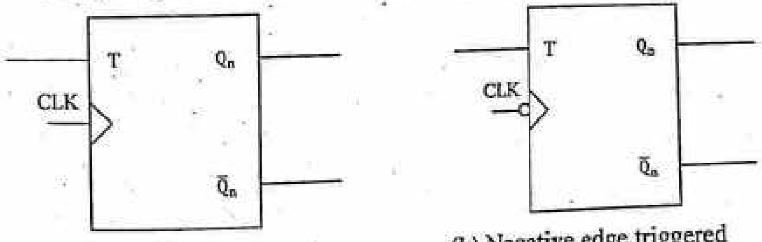
If the present state of the Flip-flop is '1' and the next state is to change '0'. This happen when the inputs must be either $J = 0, K = 1$ or $J = 1, K = 1$. Thus K has to be at '1', but J can be either level. Thus if J= 'X' and K= 1 this transition will occur.

1-to-1 Transition:

If the present state of the Flip-flop is '1' and the next state is to remain '1'. This happen when the inputs must be either $J = 0, K = 0$ or $J = 1, K = 0$. Thus K has to be at '0', but J can be either level. Thus if J= 'X' and K= 0 the next state output will remain in same state '1'.

3.3.4 T-Flip-flop

T Flip-flop is also known as Toggle Flip-flop.



(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.15 Logic symbol of T Flip-flop

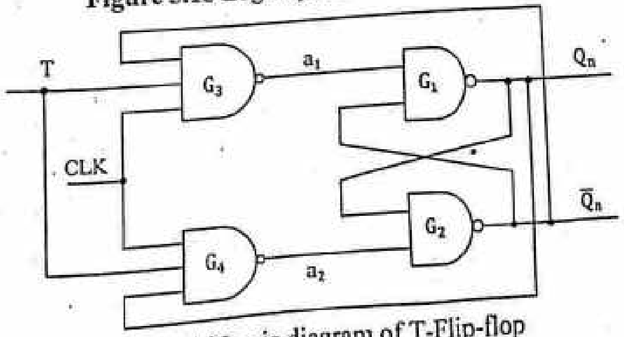


Figure 3.16 Logic diagram of T-Flip-flop

CLK	T	Q_n	Q_{n+1}	Stnte
0	X	0	0	No change
0	X	1	1	
↑	0	0	0	No change
↑	0	1	1	
↑	1	0	1	Toggles
↑	1	1	0	

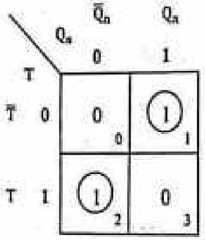
Table 3.19 Truth table of positive edge triggered

T	Q_{n+1}	State
0	Q_n	No change
1	$\overline{Q}_n$	Toggles

Table 3.20 Truth table

The characteristic equation of T Flip-flop can be obtained from the above truth table shown in table 3.19

K-map for Q_{n+1}



The characteristic equation of T Flip-flop is $Q_{n+1} = T\overline{Q}_n + \overline{T}Q_n$

Case 1: When T=0

Assume initially $Q_n = 1$ and $\overline{Q}_n = 0$. When $T = 0$ and CLK is positive edge triggered, the inputs of NAND gate G_3 are $T = 0$ and $\overline{Q}_n = 0$, hence G_3 output is logic high ($a_1 = 1$). The inputs of NAND gate G_4 are $T = 0$ and $Q_n = 1$, hence G_4 output is also logic high ($a_2 = 1$). The inputs of NAND gate G_2 are $a_2 = 1$ and $Q_n = 1$, hence the output $\overline{Q}_n$ remains logic '0'. The inputs of NAND gate G_1 are $a_1 = 1$ and $\overline{Q}_n = 0$. So the output Q_n remains logic '1'. The output Q_{n+1} does not change if T=0, even if we initially assume $Q_n = 0$ and $\overline{Q}_n = 1$.

Case 2: When T=1

Assume initially $Q_n = 1$ and $\overline{Q}_n = 0$. When $T = 1$ and CLK is positive edge triggered, the inputs of NAND gate G_3 are $T = 1$ and $\overline{Q}_n = 0$, hence G_3 output is logic high ($a_1 = 1$). The inputs of NAND gate G_4 are $T = 1$ and $Q_n = 1$, hence G_4

output is logic low ($a_2 = 0$). The inputs of NAND gate G_2 are $a_2 = 0$ and $Q_n = 1$, hence the output $\overline{Q}_n$ changes to logic '1'. The inputs of NAND gate G_1 are $a_1 = 1$ and $\overline{Q}_n = 1$. So the output Q_n changes to logic '0'. The output Q_{n+1} toggles if T=1, even if we initially assume $Q_n = 0$ and $\overline{Q}_n = 1$.

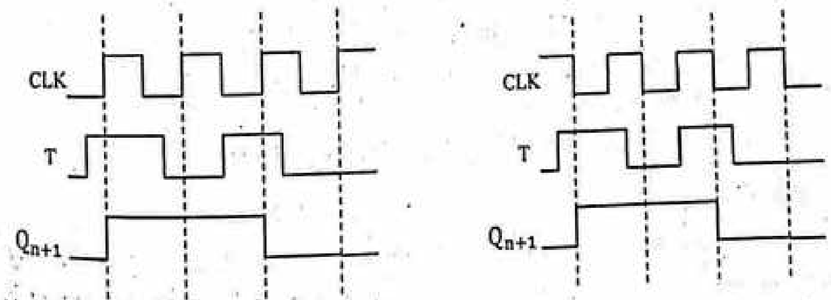
CLK	T	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
↓	0	0	0	No change
↓	0	1	1	
↓	1	0	1	Toggles
↓	1	1	0	

Table 3.21 Truth table for negative edge triggered

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

Table 3.22 Excitation table

In T Flip-flop, the input T must be 0, for the state of the Flip-flops to be unchanged. When T=1, the state of the Flip-flop is complemented.



(a) Positive edge triggered

(b) Negative edge triggered

Figure 3.17 Timing diagram of T Flip-flop

3.3.5 Master slave SR Flip-flop

A master slave SR Flip-flop consists of two SR Flip-flops and one inverter. One flip-flop serves as a master and the other as a slave. Both the Flip-flops are positive edge triggered. Since the inverter is connected at the clock input of the slave Flip-flop the inverter forces the slave Flip-flop to trigger at the negative edge.

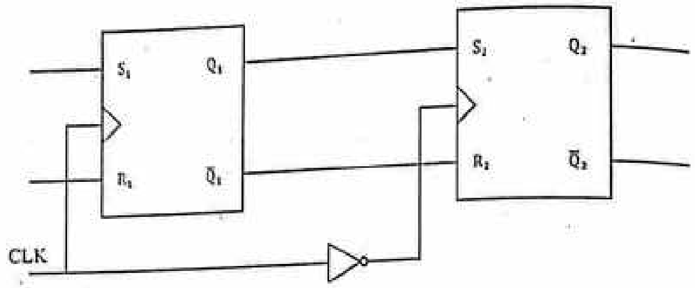


Figure 3.18 Block diagram of Master slave SR Flip-flop

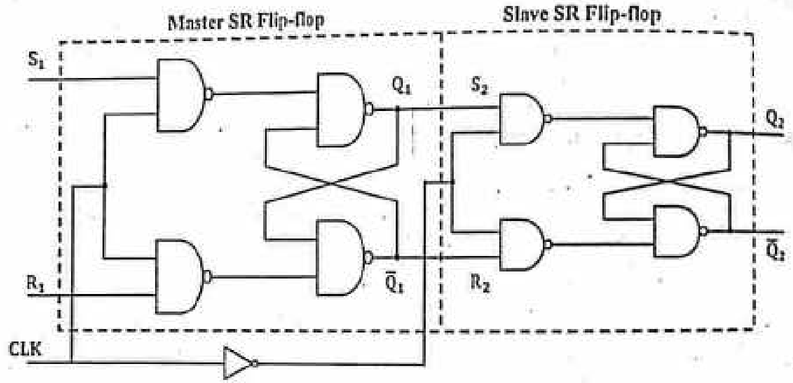


Figure 3.19 Logic diagram of master slave SR Flip-flop

The output state of the master Flip-flop is determined by the S_1 and R_1 inputs at the positive clock pulse. The output of master is fed as the input to the slave Flip-flop at S_2 and R_2. The slave Flip-flop uses this inputs S_2 and R_2 at the negative clock pulse to determine the output state Q_2 and $\overline{Q}_2$.

When $S_1 = 0$ and $R_1 = 0$, during the positive clock edge, the output state of master SR is same as that of past output and slave SR Flip-flop then copies the output of master on the negative clock.

When $S_1 = 0$ and $R_1 = 1$, during the positive clock edge, the master resets ($Q_1 = 0$ and $\overline{Q}_1 = 1$). Hence the slave input is $S_2 = 0$ and $R_2 = 1$. During the negative clock, the output of slave also resets ($Q_2 = 0$ and $\overline{Q}_2 = 1$).

When $S_1 = 1$ and $R_1 = 0$, during the positive clock edge, the master sets ($Q_1 = 1$ and $\overline{Q}_1 = 0$). Hence the slave input is $S_2 = 1$ and $R_2 = 0$. During the negative clock, the output of slave also sets ($Q_2 = 1$ and $\overline{Q}_2 = 0$).

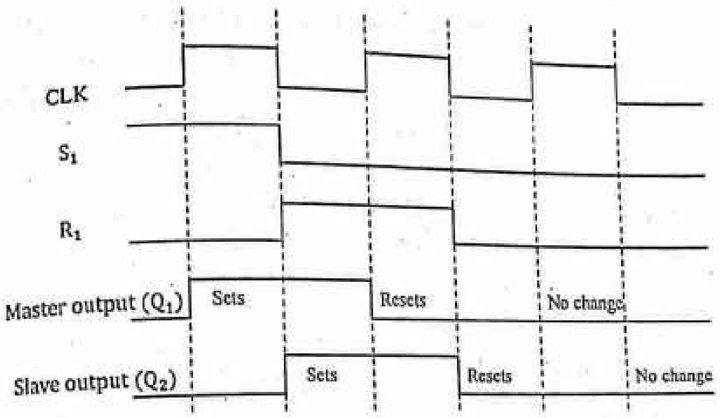


Figure 3.20 Timing diagram of master slave SR Flip-flop

3.3.6 Master slave JK Flip-flop

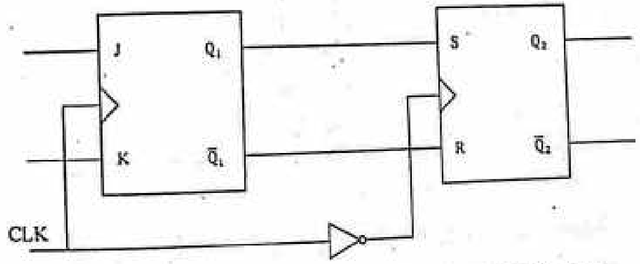


Figure 3.21 Block diagram of Master slave JK Flip-flop

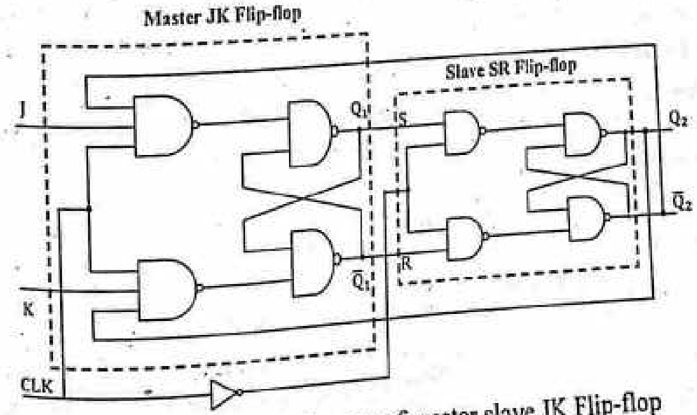


Figure 3.22 Logic diagram of master slave JK Flip-flop

The master slave JK Flip-flop consists of one clocked JK Flip-flop, one clocked SR Flip-flop and one inverter. The JK Flip-flop serves as a master and the SR Flip-flop serve as a slave. Both the Flip-flops are positive edge triggered. Since the inverter is connected at the clock input of the slave Flip-flop, the inverter forces the slave Flip-flop to trigger at the negative edge.

The output of master is fed as the input to the slave Flip-flop at S and R. The slave Flip-flop uses the inputs S and R at the negative clock pulse to determine the output state Q_2 and $\overline{Q}_2$.

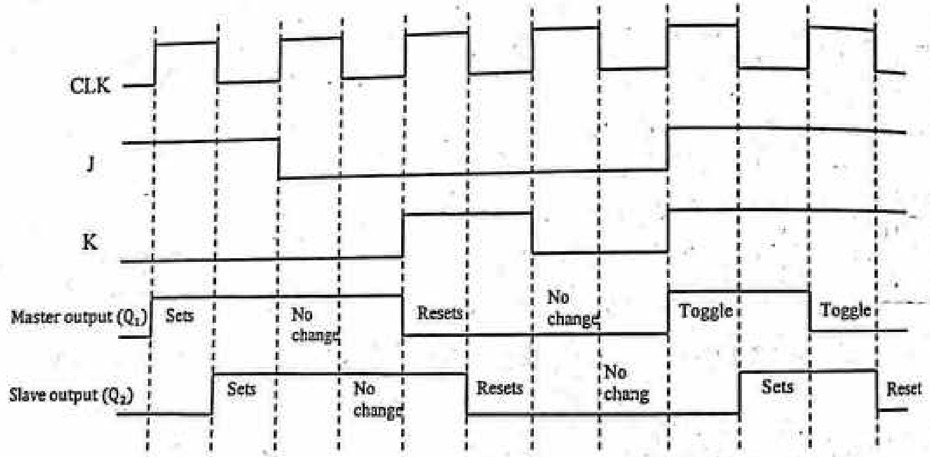


Figure 3.23 Timing diagram of master slave JK Flip-flop

When J=0 and K=0, during the positive clock, the output state of JK is same as that of past output and slave then copies the output of master on the negative clock.

When J=0 and K=1, during the positive clock, the output of master resets ($Q_1 = 0$ and $\overline{Q}_1 = 1$). Hence the slave input is S=0 and R=1. During the negative clock, the output of slave also resets. Hence if J=0 and K=1, the output of master slave JK is in reset condition.

When J=1 and K=0, during the positive clock, the output of master sets ($Q_1 = 1$ and $\overline{Q}_1 = 0$). Hence the slave input is S=1 and R=0. During the negative clock, the output of slave also sets. Hence if J=1 and K=0, the output of master slave JK is in set condition.

When J=1 and K=1, during the positive clock, the output of master toggles and slave then copies the output of master on the negative clock.

3.3.7 Characteristic table, application table and characteristic equation of Flip-flops

SR flip-flop

CLK	S	R	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0.	1	Set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	

Table 3.23 Characteristic table

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

Table 3.24 Excitation table or Application table

The characteristic equation of SR Flip-flop is $Q_{n+1} = S + \overline{R}Q_n$

D flip-flop

CLK	D	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
1	0	0	0	Reset
1	0	1	0.	
1	1	0	1	Set
1	1	1	1	

Table 3.25 Characteristic table

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

Table 3.26 Excitation table or Application table

The characteristic equation for D Flip-flop is $Q_{n+1} = D$

JK flip-flop

CLK	J	K	Q_n	Q_{n+1}	State
0	X	X	0	0	No change
0	X	X	1	1	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	1	Toggles
1	1	1	1	0	

Table 3.27 Characteristic table

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

Table 3.28 Excitation table or Application table

The characteristic equation of JK Flip-flop is $Q_{n+1} = J\overline{Q}_n + \overline{K}Q_n$

T flip-flop

CLK	T	Q_n	Q_{n+1}	State
0	X	0	0	No change
0	X	1	1	
1	0	0	0	No change
1	0	1	1	
1	1	0	1	Toggles
1	1	1	0	

Table 3.29 Characteristic table

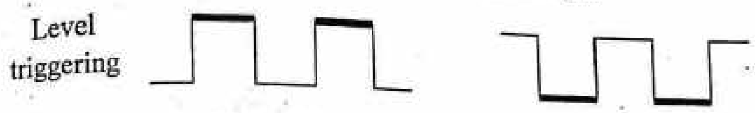
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.30 Excitation table or Application table

The characteristic equation of T Flip-flop is $Q_{n+1} = T\overline{Q}_n + \overline{T}Q_n$

3.3.8 Level triggering

We have seen SR, D, JK and T latches with enable input. Latches are controlled by enable signal, and they are level triggered, either positive level triggered or negative level triggered. The output is free to change according to the input values, when active level is maintained at the enable input.

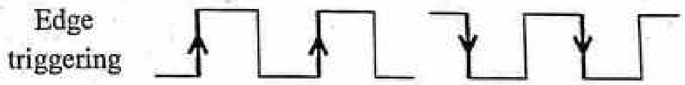


(a) Positive level (b) Negative level

Figure 3.24

3.3.9 Edge triggering

Flip-flops are edge triggered instead of level triggered.



(a) Positive edge (b) Negative edge

Figure 3.25

3.4 REALIZATION OF ONE FLIP-FLOP USING OTHER FLIP-FLOP

It is possible to convert one Flip-flop into another Flip-flop with some additional gates or using some extra connection.

3.4.1 Design of D Flip-flop using SR Flip-flop D → SR

Step 1: Write the possible binary combination for D Flip-flop input (D) and present state (Q_n).

Step 2: By using the characteristic equation of D Flip-flop, find next state (Q_{n+1}).

$$Q_{n+1} = D$$

Step 3: By using the excitation table of SR Flip-flop, find the values of S and R from Q_n and Q_{n+1}.

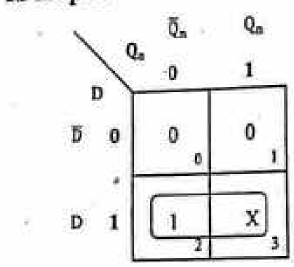
Step 4: Derive the SR Flip-flop input equations by using K-map.

Step 5: Draw the logic diagram.

D Flip-flop input	Present state	Next state	SR 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

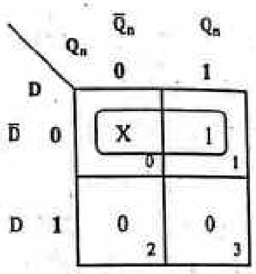
Table 3.31

K-map for S



$S = D$

K-map for R



$R = \bar{D}$

Logic diagram

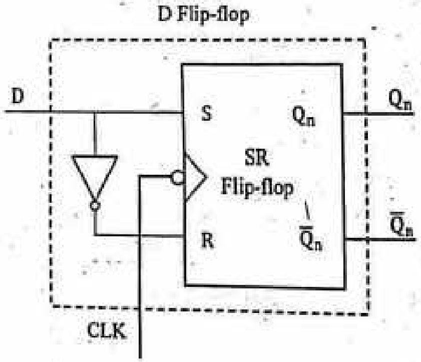


Figure 3.26 D Flip-flop using SR Flip-flop

3.4.2 Design of D Flip-flop using T Flip-flop

Step 1: Write the possible binary combination for D Flip-flop input (D) and present state (Q_n).

Step 2: By using the characteristic equation of D Flip-flop, find next state (Q_{n+1}).

$$Q_{n+1} = D$$

Step 3: By using the excitation table of T Flip-flop, find the values of T from Q_n and Q_{n+1}.

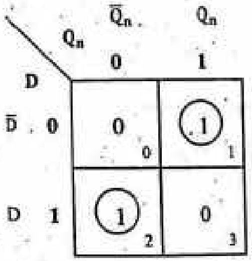
Step 4: Derive the T Flip-flop input equations by using K-map.

Step 5: Draw the logic diagram.

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

Table 3.32

K-map for T



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

$$T = D \oplus Q_n$$

Logic diagram

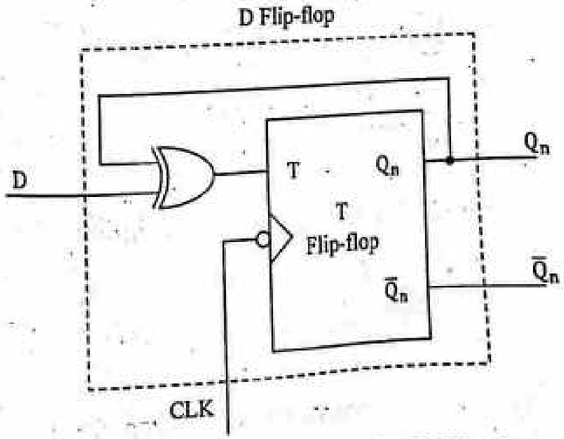


Figure 3.27 D Flip-flop using T Flip-flop

3.4.3 Design of JK Flip-flop using SR Flip-flop

Step 1: Write the possible binary combination for JK Flip-flop inputs (J and K) and present state (Q_n).

Step 2: By using the characteristic equation of JK Flip-flop, find the next state (Q_{n+1}).

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

Step 3: By using the excitation table of SR Flip-flop, find the values of S and R from Q_n and Q_{n+1}.

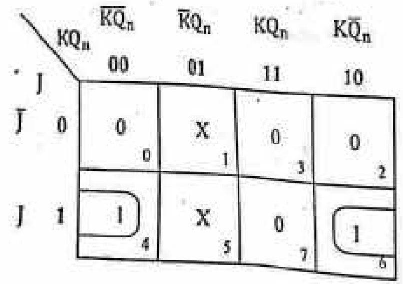
JK Flip-flop input		Present state	Next state	SR Flip-flop inputs	
J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Table 3.33

Step 4: Derive the SR Flip-flop input equations by using K-map.

Step 5: Draw the logic diagram.

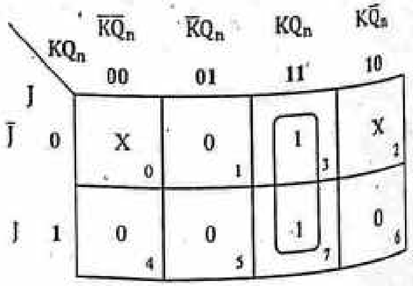
K-map for S



J \ KQ_n	$\overline{K}\overline{Q}_n$ 00	$\overline{K}Q_n$ 01	KQ_n 11	$K\overline{Q}_n$ 10
$\overline{J}$ 0	0 (0)	X (1)	0 (3)	0 (2)
J 1	1 (4)	X (5)	0 (7)	1 (6)

$S = J\overline{Q}_n$

K-map for R



J \ KQ_n	$\overline{K}\overline{Q}_n$ 00	$\overline{K}Q_n$ 01	KQ_n 11	$K\overline{Q}_n$ 10
$\overline{J}$ 0	X (0)	0 (1)	1 (3)	X (2)
J 1	0 (4)	0 (5)	1 (7)	0 (6)

$R = KQ_n$

Logic diagram

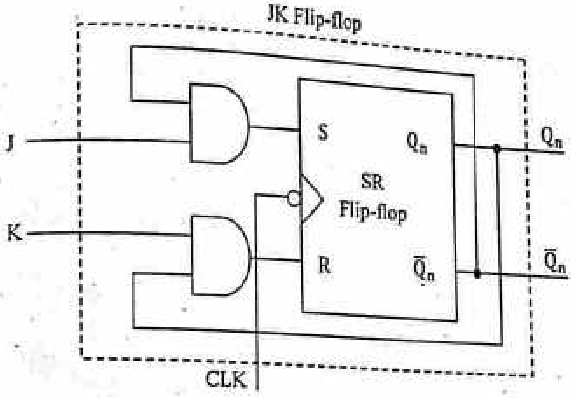


Figure 3.28 JK Flip-flop using SR Flip-flop

3.4.4 Design of T Flip-flop using SR Flip-flop

Step 1: Write the possible binary combination for T Flip-flop input (T) and present state (Q_n).

Step 2: By using the characteristic equation of T Flip-flop, find the next state (Q_{n+1}).

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

Step 3: By using the excitation table of SR Flip-flop, find the values of S and R from Q_n and Q_{n+1}.

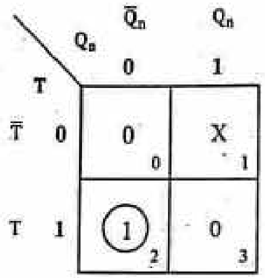
Step 4: Derive the SR Flip-flop input equations by using K-map.

Step 5: Draw the logic diagram.

T Flip-flop input	Present state	Next state	SR Flip-flop inputs	
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

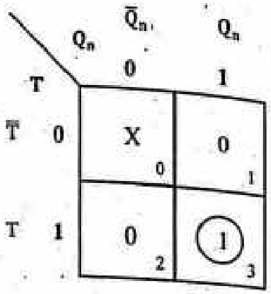
Table 3.34

K-map for S



$S = T\overline{Q}_n$

K-map for R



$R = TQ_n$

Logic diagram

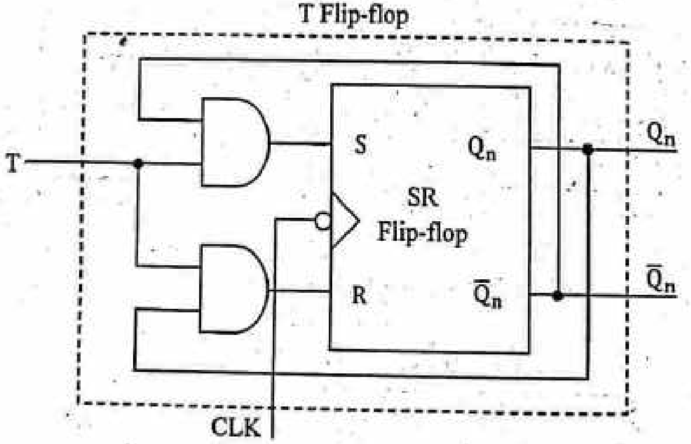


Figure 3.29 T Flip-flop using SR Flip-flop

3.4.5 Design of T Flip-flop using JK Flip-flop

Step 1: Write the possible binary combination for T Flip-flop input (T) and present state (Q_n).

Step 2: By using the characteristic equation of T Flip-flop, find the next state (Q_{n+1}).

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

Step 3: By using the excitation table of JK Flip-flop, find the values of J and K from Q_n and Q_{n+1}.

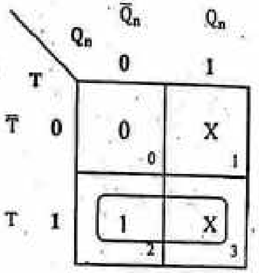
Step 4: Derive the JK Flip-flop input equations (J and K) by using K-map.

Step 5: Draw the logic diagram.

T Flip-flop input	Present state	Next state	JK Flip-flop inputs	
T	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

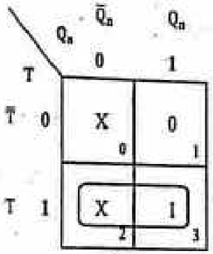
Table 3.35

K-map for J



J = T

K-map for K



K = T

Logic diagram

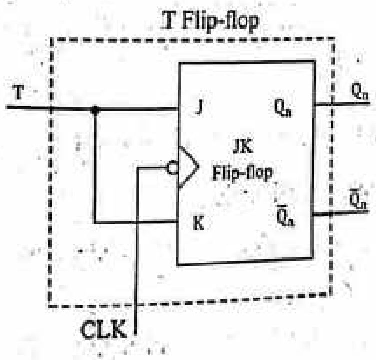


Figure 3.30 T Flip-flop using JK Flip-flop

3.4.6 Design of T Flip-flop using D Flip-flop

Step 1: Write the possible binary combination for T Flip-flop input (T) and present state (Q_n).

Step 2: By using the characteristic equation of T Flip-flop, find the next state (Q_{n+1}).

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

Step 3: By using the excitation table of D Flip-flop, find the values of D from Q_n and Q_{n+1}.

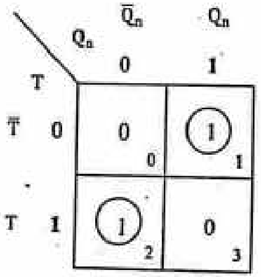
Step 4: Derive the D Flip-flop input equations (D) by using K-map.

Step 5: Draw the logic diagram.

T Flip-flop input	Present state	Next state	D 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 3.36

K-map for D



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

Logic diagram

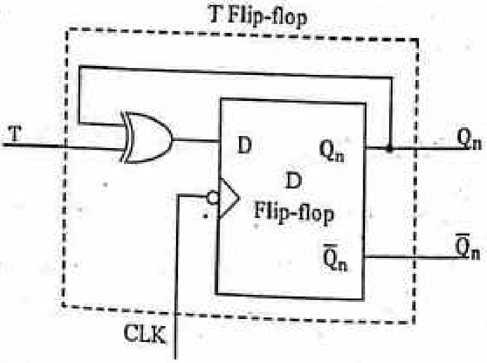


Figure 3.31 T Flip-flop using D Flip-flop

3.4.7 Design of D Flip-flop using JK Flip-flop

Step 1: Write the possible binary combination for D Flip-flop input (D) and present state (Q_n).

Step 2: By using the characteristic equation of D Flip-flop, find the next state (Q_{n+1}).

$$Q_{n+1} = D$$

Step 3: By using the excitation table of JK Flip-flop, find the values of J and K from Q_n and Q_{n+1}.

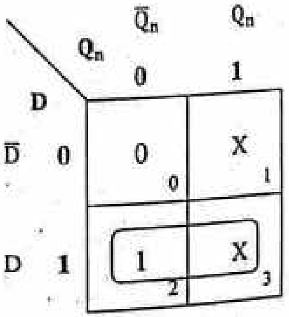
Step 4: Derive the JK Flip-flop input equations (J and K) by using K-map.

Step 5: Draw the logic diagram.

D Flip-flop input	Present state	Next state	JK 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

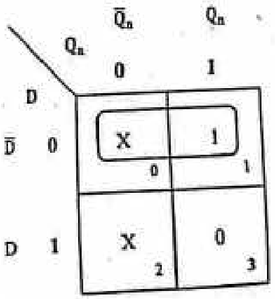
Table 3.37

K-map for J



$J = D$

K-map for K



$K = \overline{D}$

Logic diagram

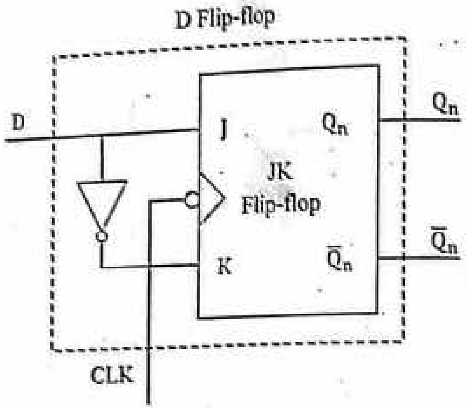


Figure 3.32 D Flip-flop using JK Flip-flop

3.4.8 Design of JK Flip-flop using T Flip-flop

Step 1: Write the possible binary combination for JK Flip-flop inputs (J and K) and present state (Q_n).

Step 2: By using the characteristic equation of JK Flip-flop, find the next state (Q_{n+1}).

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

Step 3: By using the excitation table of T Flip-flop, find the values of T from Q_n and Q_{n+1}.

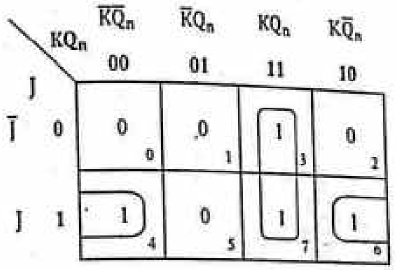
JK Flip-flop input		Present state	Next state	T Flip-flop inputs
J	K	Q_n	Q_{n+1}	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

Table 3.38

Step 4: Derive the T Flip-flop input equations by using K-map.

Step 5: Draw the logic diagram.

K-map for T



J \ KQ_n	$\overline{KQ}_n$ 00	$\overline{K}Q_n$ 01	KQ_n 11	$K\overline{Q}_n$ 10
$\overline{J}$ 0	0 (0)	0 (1)	1 (3)	0 (2)
J 1	1 (4)	0 (5)	1 (7)	1 (6)

$$T = J\overline{Q}_n + KQ_n$$

Logic diagram

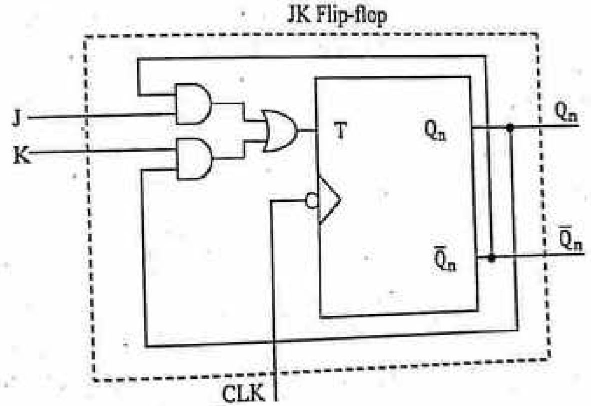


Figure 3.33 JK Flip-flop using T Flip-flop

3.5 SERIAL ADDER/SUBTRACTOR

Serial adder can add numbers stored in the right shift register A and B serially. The full adder is used to perform bit by bit addition and a D-flip flop is used to store the carry output generated after addition. This carry output stored in D-flip flop is used as carry input for the next addition. Initially the D-flip flop is cleared and addition starts with the LSB bit of register A and B. The full adder adds the LSB bit of A and B and provides the output sum and carry. After each clock pulse data present in the right shift registers are shifted right by 1-bit and also the previous carry is fed as input 'C' to the full adder.

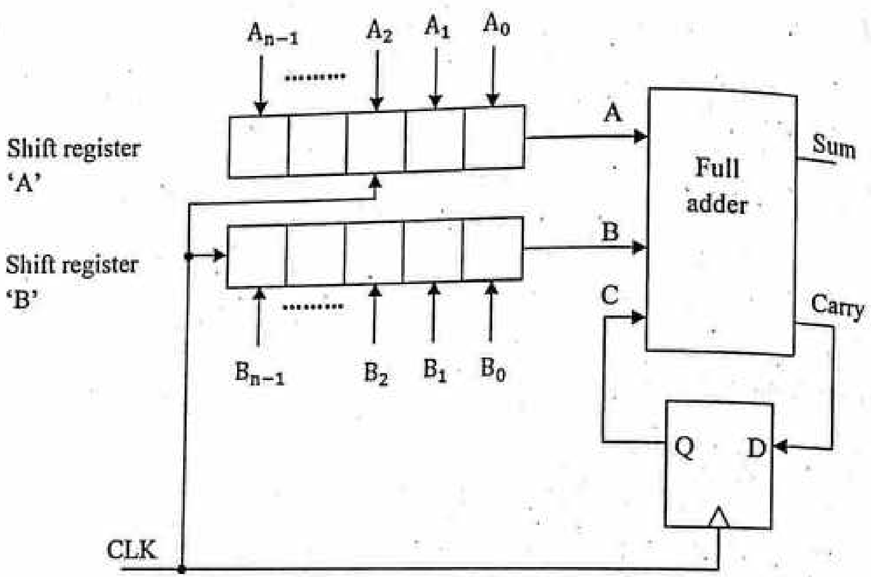
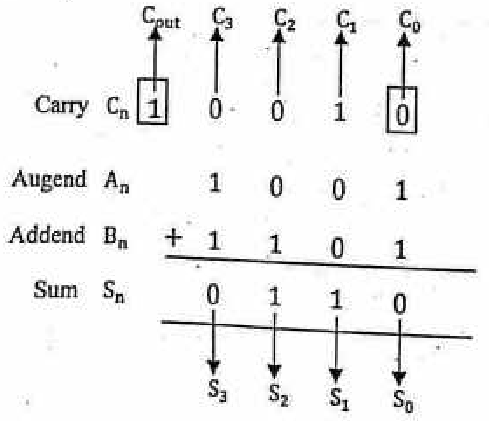


Figure 3.34 Block diagram of serial adder

The full adder provides the output sum and carries on receipt of each clock pulse (CLK) until the two numbers stored initially in shift register A and B have been added and the resulting sum has been clocked back into the shift register A. We can implement serial subtractor by replacing full subtractor instead of full adder.

Example 3.1 : Design a serial binary adder using delay flip-flop.

Solution:



Consider the binary numbers A=1001 and B=1101. Here $C_0 = 0$. The 4 bit adder performs the addition as shown above. The Transition table of a serial binary adder is given in table 3.39.

Input		Present State	Next state	Flip-flop inputs	Output
A_n	B_n	C_n	Carry (C_{n+1})	D	Sum (S_n)
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1

Table 3.39 Transition table

Let the state '0' be represented by 'a' and state '1' be represented by 'b'. The state table for a serial adder is shown in table 3.40.

Present State	Next state C_{n+1}				Output S_n			
C_n	A_nB_n=00	A_nB_n=01	A_nB_n=10	A_nB_n=11	A_nB_n=00	A_nB_n=01	A_nB_n=10	A_nB_n=11
a	a	a	a	b	0	1	1	0
b	a	b	b	b	1	0	0	1

Table 3.40 State table

The state diagram of a serial adder can be drawn as

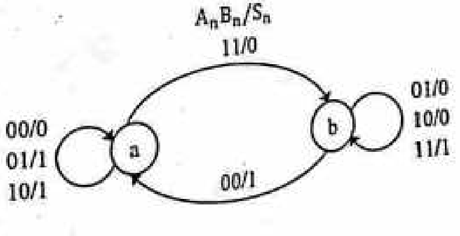


Figure 3.35 State Diagram

K-map for D

$A_n \backslash B_nC_n$	$\bar{B}_n\bar{C}_n$ 00	$\bar{B}_nC_n$ 01	B_nC_n 11	$B_n\bar{C}_n$ 10
$\bar{A}_n$ 0	0_{0}	0_{1}	1_{3}	0_{2}
A_n 1	0_{4}	1_{5}	1_{7}	1_{6}

K-map for S_n

$A_n \backslash B_nC_n$	$\bar{B}_n\bar{C}_n$ 00	$\bar{B}_nC_n$ 01	B_nC_n 11	$B_n\bar{C}_n$ 10
$\bar{A}_n$ 0	0_{0}	1_{1}	0_{3}	1_{2}
A_n 1	1_{4}	0_{5}	1_{7}	0_{6}

$$D = A_nB_n + A_nC_n + B_nC_n$$

$$\begin{aligned}\text{Sum } S_n &= \bar{A}_n\bar{B}_nC_n + \bar{A}_nB_n\bar{C}_n + A_n\bar{B}_n\bar{C}_n + A_nB_nC_n\\ &= \bar{A}_n(\bar{B}_nC_n + B_n\bar{C}_n) + A_n(\bar{B}_n\bar{C}_n + B_nC_n)\\ &= \bar{A}_n(B_n \oplus C_n) + A_n(B_n \odot C_n)\\ &= \bar{A}_n(B_n \oplus C_n) + A_n\overline{(B_n \oplus C_n)}\\ &= A_n \oplus (B_n \oplus C_n) = A_n \oplus B_n \oplus C_n\end{aligned}$$

Logic diagram

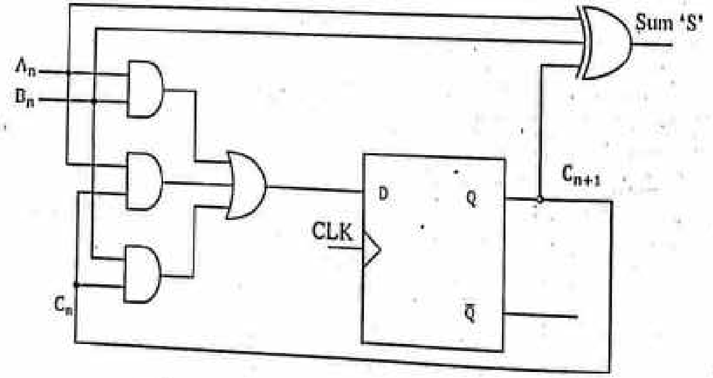


Figure 3.36 Logic diagram of serial adder

3.6 COUNTERS

A counter is a register (group of Flip-flop) capable of counting the number of clock pulse arriving at its clock input.

Counters are classified into two types.

i. Asynchronous (Ripple) counter
ii. Synchronous counter

3.6.1 Asynchronous (Ripple or Serial) Counter

In asynchronous (ripple) counter the external clock signal is connected to the clock input of first stage Flip-flop. The clock input of the second, third and fourth stage Flip-flops are triggered by its previous stage Flip-flop output Q or $\bar{Q}$.

Asynchronous counter is further classified into two types.

i. Asynchronous up-counter (Ripple up-counter or serial up-counter)
ii. Asynchronous down-counter (Ripple down-counter or serial down-counter)

3.6.1.1 Asynchronous up-counter (Ripple or serial up-counter)

The output of up-counter is incremented by one for each clock transition. A 4 bit asynchronous up-counter consists of 4 JK Flip-flops. The external clock signal is connected to the clock input of the first Flip-flop. The clock inputs of the remaining Flip-flops are triggered by the Q output of the previous stage.

We know that in JK Flip-flop, if J=1, K=1 and clock is triggered the past output will be complemented.

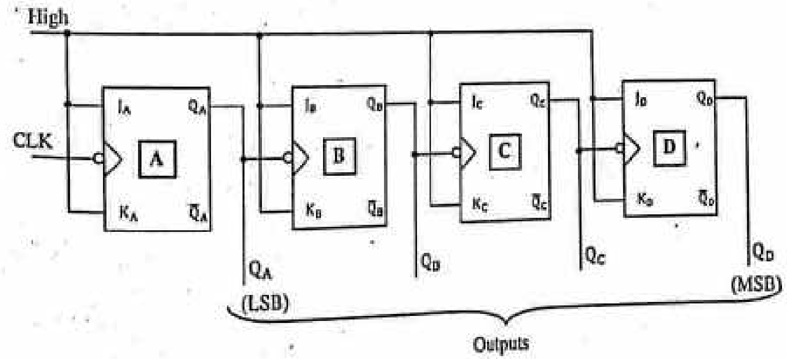


Figure 3.37 4-bit Asynchronous Up-counter

Initially, the register is cleared, $Q_DQ_CQ_BQ_A = 0000$. During the first clock pulse, Flip-flop A triggers, therefore $Q_A = 1$, $Q_B = Q_C = Q_D = 0$

$$Q_DQ_CQ_BQ_A = 0001$$

At the second clock pulse Flip-flop A triggers, therefore Q_A changes from 1 to 0, which triggers Flip-flop B, therefore $Q_B = 1$, $Q_A = Q_C = Q_D = 0$

$$Q_DQ_CQ_BQ_A = 0010$$

At the third clock pulse Flip-flop A triggers, therefore Q_A changes from 0 to 1,this never triggers Flip-flop B because 0 to 1 transition gives a positive edge triggering, but here the Flip-flops are triggered only at negative edge (1 to 0 transition) therefore $Q_A = Q_B = 1$, $Q_C = Q_D = 0$

$$Q_DQ_CQ_BQ_A = 0011$$

At the fourth clock pulse Flip-flop A triggers, therefore Q_A changes from 1 to 0, this triggers Flip-flop B therefore Q_B changes from 1 to 0. The change in Q_B from 1 to 0 triggers C Flip-flop, therefore Q_C changes from 0 to 1. Therefore $Q_A = Q_B = Q_D = 0$, $Q_C = 1$

$$Q_DQ_CQ_BQ_A = 0100$$

The same procedure repeats until the counter counts up to '1111'.

CLK	Outputs			
	Q_D	Q_C	Q_B	Q_A
-	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table 3.41 Truth table for 4-bit asynchronous up-counter

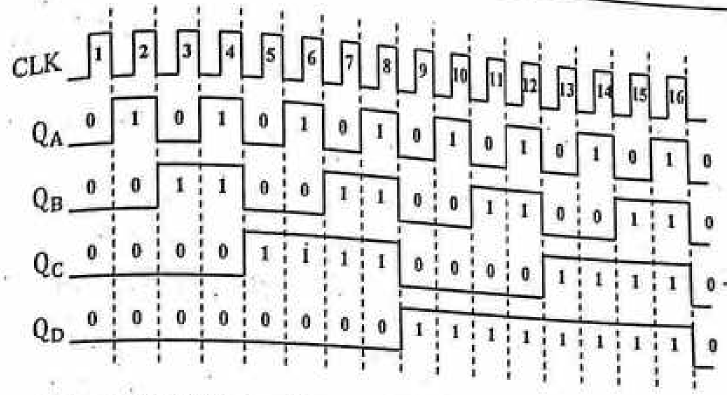


Figure 3.38 Timing diagram of 4-bit asynchronous up-counter.

3.6.1.2 Asynchronous down-counter (Ripple or serial down counter)

The output of down-counter is decremented by one for each clock transition. A 4 bit asynchronous down-counter consists of 4 JK Flip-flops. The external clock signal is connected to the clock input of the first Flip-flop.

The clock inputs of the remaining Flip-flops are triggered by the $\overline{Q}$ output of the previous stage. We know that, in JK Flip-flop, if J=1, K=1 and clock is triggered the past output will be complemented.

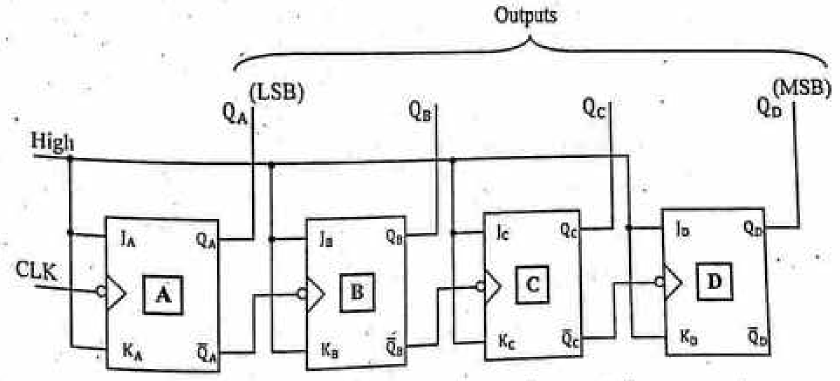


Figure 3.39 Logic diagram of 4-bit asynchronous down-counter

Initially, the register is cleared, $Q_DQ_CQ_BQ_A = 0000$

During the first clock pulse Flip-flop A triggers, therefore Q_A changes from 0 to1 also $\overline{Q}_A$ changes from 1 to 0. This triggers B Flip-flop, therefore Q_B changes from 0 to 1, also $\overline{Q}_B$ changes from 1 to 0 which triggers C Flip-flop.

Hence Q_C changes from 0 to1 and $\overline{Q}_C$ changes from 1 to 0, which further triggers, Flip-flop D.

$$Q_DQ_CQ_BQ_A = 1111$$

$$\overline{Q}_D\overline{Q}_C\overline{Q}_B\overline{Q}_A = 0000$$

During the second clock pulse Flip-flop A triggers, therefore Q_A changes from 1 to 0 also $\overline{Q}_A$ changes from 0 to 1 which never triggers B Flip-flop. Therefore C and D Flip-flop are not triggered.

$$Q_DQ_CQ_BQ_A = 1110$$

The same procedure repeats until the counter decrements up to '0000'.

CLK	Outputs			
	Q_D	Q_C	Q_B	Q_A
-	0	0	0	0
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	0	1	1	1
10	0	1	1	0
11	0	1	0	1
12	0	1	0	0
13	0	0	1	1
14	0	0	1	0
15	0	0	0	1
16	0	0	0	0

Table 3.42 Truth table for 4-bit asynchronous down-counter

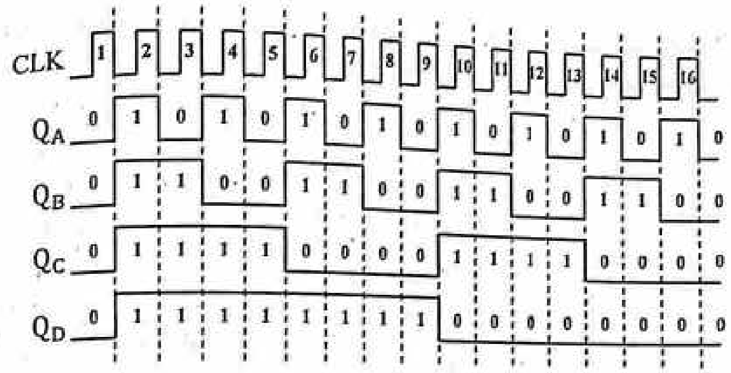


Figure 3.40 Timing diagram of 4-bit asynchronous down-counter.

3.6.1.3 Asynchronous Up/Down counter

The up-down counter has the capability of counting upwards as well as downwards. It is also called multimode counter. In a asynchronous up-counter, each flip-flop is triggered by the normal output Q of the proceding flip-flop. In a asynchronous down-counter, each flip-flop is triggered by the complement output $\overline{Q}$ of the proceding flip-flop. In both the counters, the first flip-flop is triggered by the clock input.

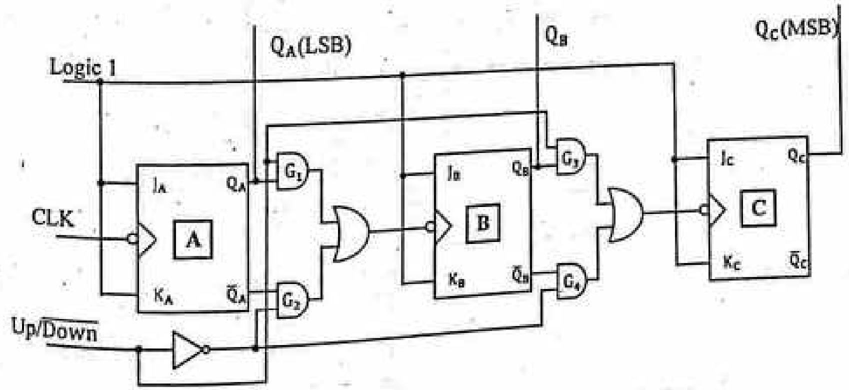


Figure 3.41 3-bit asynchronous up/down-counter

If $Up/\overline{Down} = 1$, the 3-bit asynchronous up/down-counter will perform up-counting. It will count from 000 to 111. If $Up/\overline{Down} = 1$ gates G_2 and G_4 are disabled and gates G_1 and G_3 are enabled. So that the circuit behaves as an up-counter circuit.

If Up/$\overline{\text{Down}}$ = 0, the 3-bit asynchronous up/down-counter will perform down-counting. It will count from 111 to 000. If Up/$\overline{\text{Down}}$ = 0 gates G_2 and G_4 are enabled and gates G_1 and G_3 are disabled. So that the circuit behaves as a down-counter circuit.

Up/$\overline{\text{Down}}$ = 1

Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Up/$\overline{\text{Down}}$ = 0

Table 3.43 Truth table for 3-Bit asynchronous Up/Down-counter

3.6.2 Synchronous counters

In synchronous counter each Flip-flop is triggered at the same time. Synchronous counter is further classified into two types.

i. Synchronous up-counter
ii. Synchronous down-counter

3.6.2.1 Synchronous up-counter

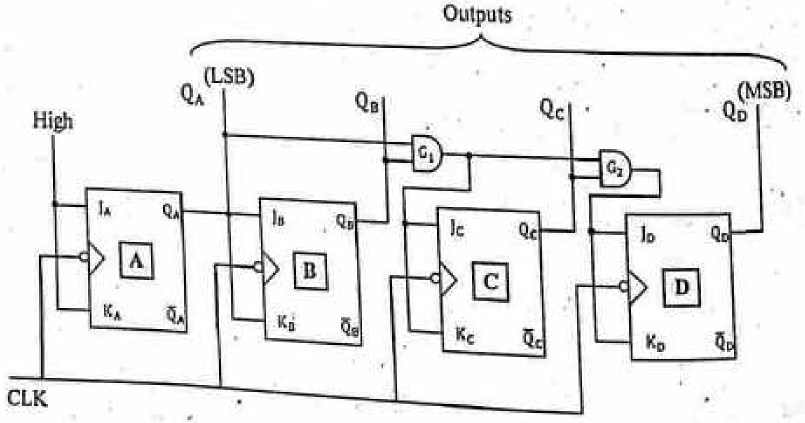


Figure 3.42 Logic diagram of 4-bit Synchronous up-counter

In JK Flip-flop, if J=0, K=0 and clock is triggered, the output never changes. If J=1, K=1 and clock is triggered, the past output will be complemented. Initially the register is cleared $Q_DQ_CQ_BQ_A = 0000$

During the first clock pulse, $J_A = K_A = 1$, Q_A becomes 1, Q_B, Q_C, Q_D remains 0

$Q_DQ_CQ_BQ_A = 0001$

During the second clock pulse, $J_A = K_A = 1, Q_A = 0$

$J_B = K_B = 1, Q_B = 1$, Q_C, Q_D remains 0

$Q_DQ_CQ_BQ_A = 0010$

During the third clock pulse, $J_A = K_A = 1, Q_A = 1$

$J_B = K_B = 0, Q_B = 1$, Q_C, Q_D remains 0

$Q_DQ_CQ_BQ_A = 0011$

During the fourth clock pulse, $J_A = K_A = 1, Q_A = 0$

$J_B = K_B = 1, Q_B = 0$

$J_C = K_C = 1, Q_C = 1$

Q_D remains 0

$Q_DQ_CQ_BQ_A = 0100$

The same procedure repeats until the counter counts up to '1111'.

CLK	Outputs			
	Q_D	Q_C	Q_B	Q_A
-	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table 3.44 Truth table for 4-bit synchronous up-counter

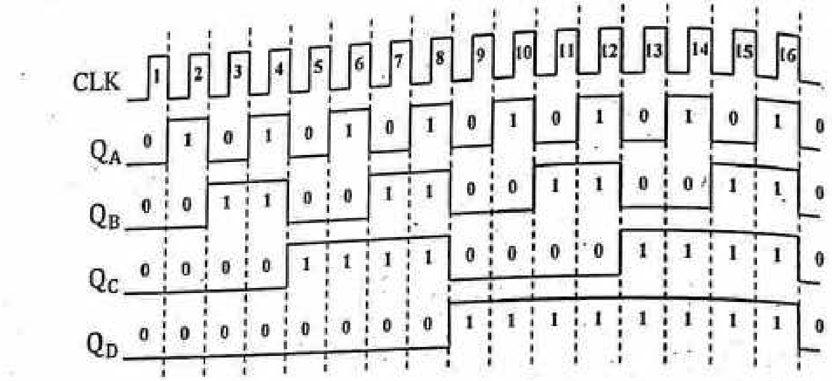


Figure 3.43 Timing diagram of 4-bit synchronous up-counter

3.6.2.2 Synchronous down-counter

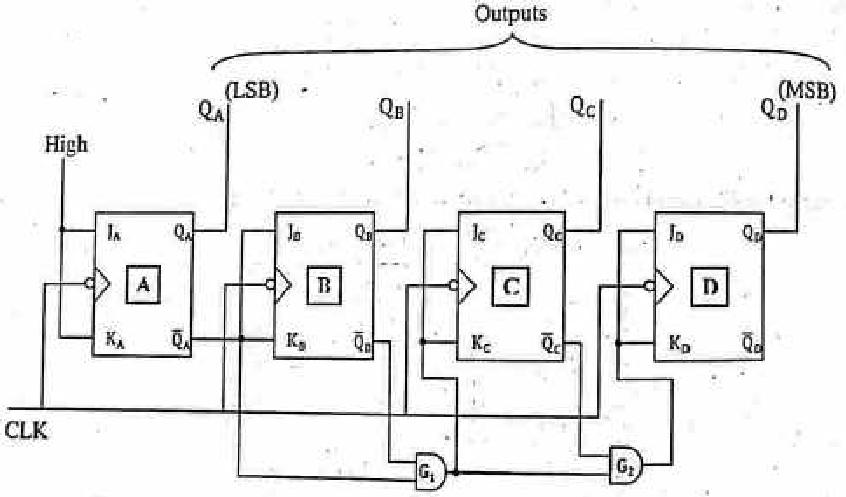


Figure 3.44 Logic diagram of 4-bit synchronous down-counter

In JK Flip-flop, if J=0, K=0 and clock is triggered, the output never changes. If J=1, K=1 and clock is triggered, the past output will be complemented

Initially the register is cleared

$$Q_DQ_CQ_BQ_A = 0000$$
$$\bar{Q}_D\bar{Q}_C\bar{Q}_B\bar{Q}_A = 1111$$

When the first clock edge triggers

$$J_A = K_A = 1, Q_A = 1$$
$$J_B = K_B = 1, Q_B = 1$$
$$J_C = K_C = 1, Q_C = 1$$

$$J_D = K_D = 1, Q_D = 1$$
$$Q_DQ_CQ_BQ_A = 1111$$
$$\bar{Q}_D\bar{Q}_C\bar{Q}_B\bar{Q}_A = 0000$$

When the second clock edge triggers

$$J_A = K_A = 1, Q_A = 0$$
$$J_B = K_B = 0, Q_B = 1$$
$$J_C = K_C = 0, Q_C = 1$$
$$J_D = K_D = 0, Q_D = 1$$
$$Q_DQ_CQ_BQ_A = 1110$$
$$\bar{Q}_D\bar{Q}_C\bar{Q}_B\bar{Q}_A = 0001$$

When the third clock edge triggers

$$J_A = K_A = 1, Q_A = 1$$
$$J_B = K_B = 1, Q_B = 0$$
$$J_C = K_C = 0, Q_C = 1$$
$$J_D = K_D = 0, Q_D = 1$$
$$Q_DQ_CQ_BQ_A = 1101$$

The process repeats until the counter down-counts up to '0000'

CLK	Outputs			
	Q_D	Q_C	Q_B	Q_A
-	0	0	0	0
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	0	1	1	1
10	0	1	1	0
11	0	1	0	1
12	0	1	0	0
13	0	0	1	1
14	0	0	1	0
15	0	0	0	1
16	0	0	0	0

Table 3.45 Truth table of 4-bit synchronous down-counter

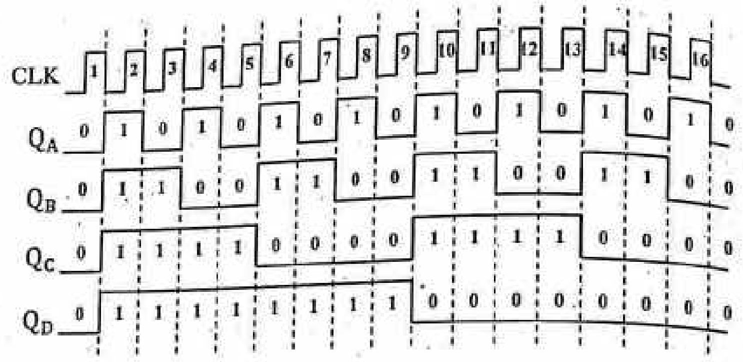


Figure 3.45 Timing diagram of 4-bit synchronous down-counter

3.6.2.3 Synchronous Up/Down counter

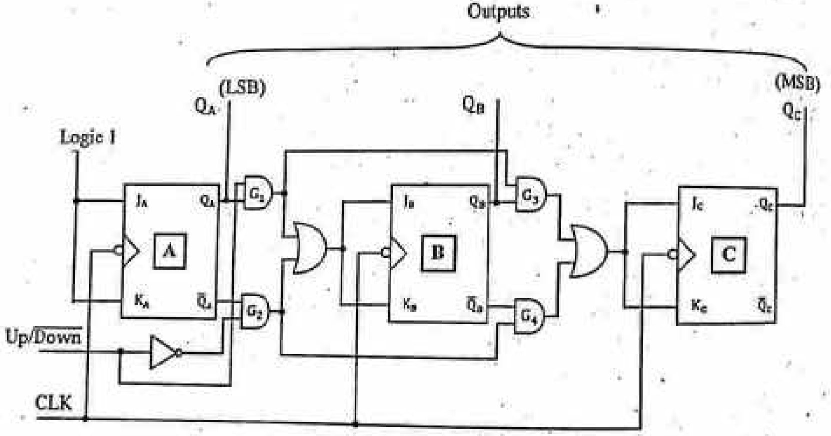


Figure 3.46 3-bit synchronous up/down-counter

In synchronous up-counter the Q_A output is given to J_B, K_B and $Q_A \cdot Q_B$ is given to J_C, K_C. But in synchronous down-counter $\overline{Q}_A$ output is given to J_B, K_B and $\overline{Q}_A \cdot \overline{Q}_B$ is given to J_C, K_C.

A control input (Up/$\overline{Down}$) is used to select the mode of operation.

If Up/$\overline{Down}$ = 1, the 3-bit synchronous up/down-counter will perform up-counting. It will count from 000 to 111. If Up/$\overline{Down}$ = 1 gates G_2 and G_4 are disabled and gates G_1 and G_3 are enabled. So that the circuit behaves as an up-counter circuit.

If Up/$\overline{Down}$ = 0, the 3-bit synchronous up/down-counter will perform down-counting. It will count from 111 to 000. If Up/$\overline{Down}$ = 0 gates G_2 and G_4 are enabled and gates G_1 and G_3 are disabled. So that the circuit behaves as a down-counter circuit.

Up/$\overline{Down}$ = 1

Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Up/$\overline{Down}$ = 0

Table 3.46 Truth table for 3-Bit synchronous Up/Down-counter

3.6.3 Programmable counter

Programmable counter is also known as presettable counters. A 3-bit up counter start from 000 while a 3 bit down counter starts from 111. This is accomplished by applying a momentary pulse to all the flip-flops. A counter can also be made to start in any desired state through the use of appropriate logic circuitry. Counters that have the capability to start counting from any desired states are called presettable or programmable counters.

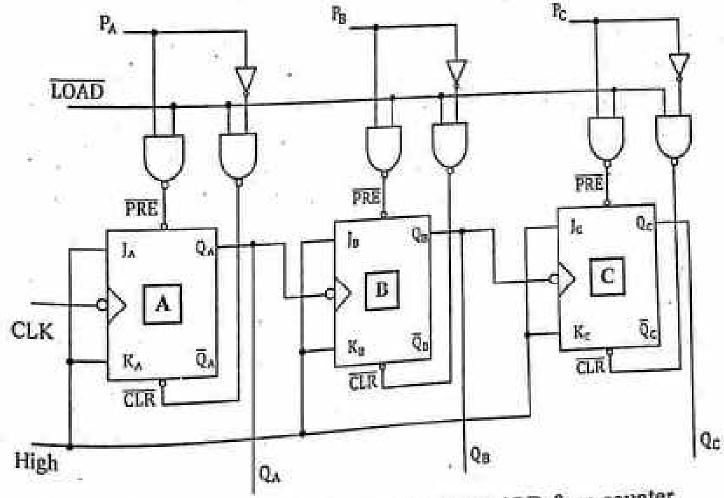


Figure 3.47 Presettable (Programmable) MOD-8 up counter.

A presettable MOD-8 ripple up-counter is shown in figure 3.47. In this counter, the desired starting state is entered using PRESET and CLEAR inputs. When $\overline{LOAD} = 0$, the desired count is determined by the inputs P_A, P_B and P_C whose values are transferred into the counter flip-flops. When $\overline{LOAD} = 1$, the NAND gates are disabled and the counter is free to count input clock pulses starting from the newly entered count that has been loaded in the flip-flops.

3.7 ANALYSIS OF SYNCHRONOUS SEQUENTIAL CIRCUITS

The behavior of a clocked sequential circuit can be found out from the inputs, outputs and state of its Flip-flops. The steps to analyze the synchronous sequential circuit are,

Step 1: Determine the Flip-flop input equation, and the output equation from the circuit. If there is no output, find the Flip-flop input equation alone.

Step 2: Plot the transition table.

The table should contain,

a. Present state
b. Inputs
c. Next state
d. Flip-flop inputs
e. Outputs

Step 3: Plot the state table

The state table should contain,

a. Present state
b. Next state
c. Outputs

If there is no output in the circuit, the transition table (Step 2) and the state table (Step 3) does not contain the outputs.

Step 4: Draw the state diagram

(transition equation)

state and

Transition table

The transition table should contain present state, inputs, next state and outputs. The next state is the state of the flip-flop after the application of the clock pulse. The present state is the state of the Flip-flop before the application of the clock pulse.

State table

The present state and the next states present in the transition table are replaced by an alphanumeric symbol. The resulting table is known as state table. The state table should contain present state, next state and outputs.

State diagram

State diagram is the graphical representation of state table. In state diagram, a state is represented by a circle, and the transitions between the states are indicated by directed lines connecting the circles. The input value during the present state is marked first and the number after the slash gives the output during the present state with the given input.

Example 3.2: Construct the transition table, state table and state diagram for the sequential circuit shown in figure 3.48.

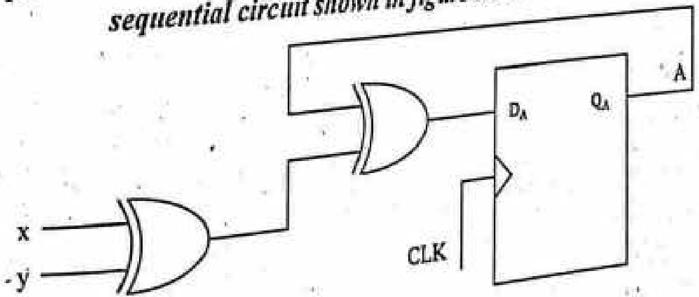


Figure 3.48 Logic diagram

Solution :

In the circuit shown in figure 3.48, A is the present state. Therefore assume, let A^+ be the next state. Here D_A is the Flip-flop input, (x, y) are the input to the circuit. There is no output in the given circuit.

Step 1: Determine the Flip-flop input equation and the output equations from the circuit.

equation: $D_A = (x \oplus y) \oplus A$

Step 2: Plot the transition table.

First write the possible binary combination for present state and inputs. Then by substituting the value of x, y and A in Flip-flop input equation, find D_A. Then write the next state A^+ by using the excitation table of D Flip-flop shown in table 3.47.

D_A	A^+
0	0
1	1

Table 3.47 Excitation table of D Flip-flop

Present state	Inputs		Next state	Flip-flop inputs
A	x	y	A^+	D_A
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Table 3.48 Transition table

Step 3: Plot the state table

Here the present state and next state is of 1 bit therefore assume, state a=0 and state b=1. Also input 'xy' have four possibilities xy=00, xy=01, xy=10 and xy=11

Present state	Next state A^+			
	xy=00	xy=01	xy=10	xy=11
a	a	b	b	a
b	b	a	a	b

Table 3.49 State table

Step 4: Draw the state diagram

From the state table, it is clear that the present states are a and b, the next state depend on input x and y.

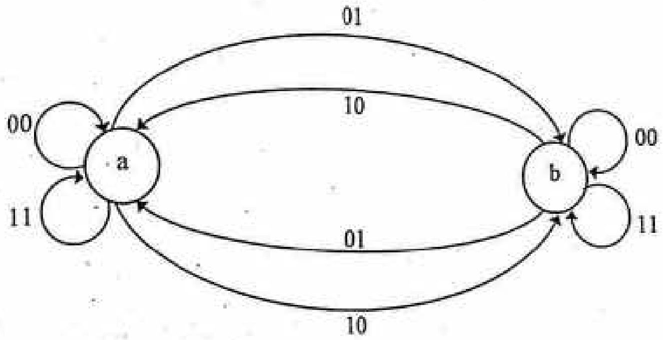


Figure 3.49 State diagram

Example 3.3: *Construct the transition table, state table and state diagram for the moore model sequential circuit shown in figure 3.50.*

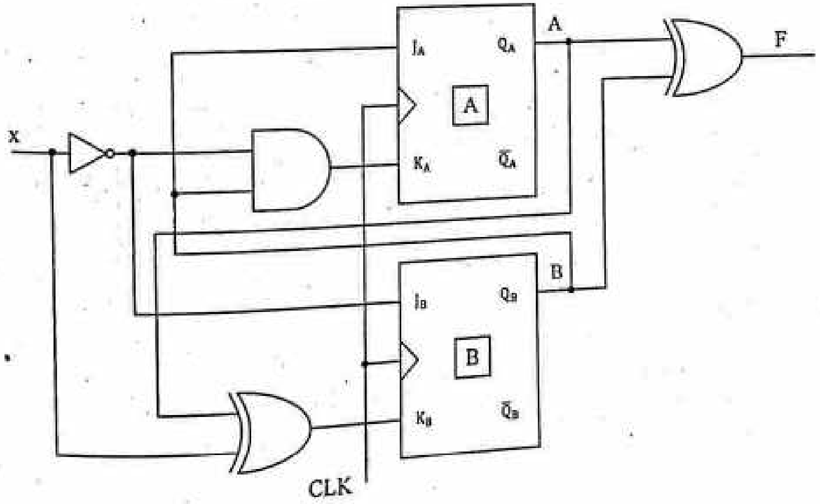


Figure 3.50 Logic diagram

Solution:

In the circuit shown in figure 3.50, A and B are the present states. Let A^+ and B^+ are the next states, 'x' is the input and 'F' is the output.

Step 1: Determine the Flip-flop input equation and the output equation from the circuit.

Flip-flop input equations:

$J_A = B$; $K_A = B\bar{x}$

$J_B = \bar{x}$; $K_B = x \oplus A$

Output equation: $F = A \oplus B$

Step 2: Plot the transition table

First write the possible binary combination for present state and inputs. Then by substituting the value of A, B and x in Flip-flop input equation and output equation, find J_A, K_A, J_B, K_B and F. Since the value of J_A, K_A, J_B and K_B are known, write the next states A^+ and B^+ from the excitation table of JK Flip-flop.

J_A	K_A	A^+
0	0	A
0	1	0
1	0	1
1	1	$\bar{A}$

Table 3.50 Excitation table of JK Flip-flop

Present state		Input	Next state		Flip-flop inputs				Output
A	B	x	A^+	B^+	J_A	K_A	J_B	K_B	F
0	0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0
0	1	0	1	1	1	1	1	0	1
0	1	1	1	0	1	0	0	1	1
1	0	0	1	1	0	0	1	1	1
1	0	1	1	0	0	0	0	0	1
1	1	0	0	0	1	1	1	1	0
1	1	1	1	1	1	0	0	0	0

Table 3.51 Transition table

Step 3: Plot the state table.

Here the present state and the next state is of 2 bits, therefore assume State a=0l; State b=01; State c=10 and State d=11. Here the input x have two possibilities x=0 and x=1.

Present state	Next state		Output F	
	x=0	x=1	x=0	x=1
a	b	a	0	0
b	d	c	1	1
c	d	c	1	1
d	a	d	0	0

Table 3.52 State table

Step 4: Draw the state diagram

From the state table shown in table 3.52, the states are a, b, c and d. The next state depends on input x. The input value during the present state is marked first and the number after the slash gives the output.

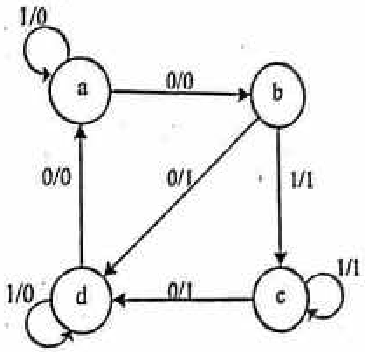


Figure 3.51 State diagram

Example 3.4: ***Construct the transition table, state table and state diagram. For the Mealy model sequential circuit shown in figure 3.52.***

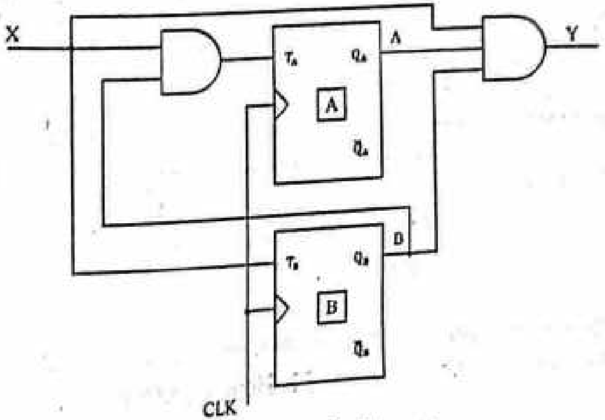


Figure 3.52 Logic diagram

Solution:

In the circuit shown in figure 3.52, A and B are the present states, A^+ and B^+ are the next states, 'X' is the input and 'Y' is the output.

Step1: Determine the Flip-flop input equations and the output equation.

Flip-flop input equation

$T_A = BX$; $T_B = X$

Output equation:

$Y = ABX$

Step 2: Plot the transition table.

First write the possible binary combination for present state and inputs. Then substituting the value of A, B and X in Flip-flop input equations and output equations, find T_A, T_B and Y. Then write the next states A^+ and B^+ from the excitation table of T Flip-flop, since the value of T_A, T_B are known.

T_A	A^+
0	A
1	$\overline{A}$

Table 3.53 Excitation table of T Flip-flop

Present state		Input	Next state		Flip-flop inputs		Output
A	B	X	A^+	B^+	T_A	T_B	Y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0
1	0	0	1	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	0	0	0
1	1	1	0	0	1	1	1

Table 3.54 Transition table

Step 3: Plot the state table.

Here the present state and the next state is of 2 bits, therefore assume State a=00; State b=01; State c=10 and State d=11. Here the input X have two possibilities X=0 and X=1.

Present state	Next state		Output F	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	b	c	0	0
c	c	d	0	0
d	d	a	0	1

Table 3.55 State table

Step 4: Draw the state diagram

From the state table shown in table 3.55, the states are a, b, c and d. The next state depends on input X. The input value is marked before the slash and the output is marked after the slash. The state diagram is shown in figure 3.53.

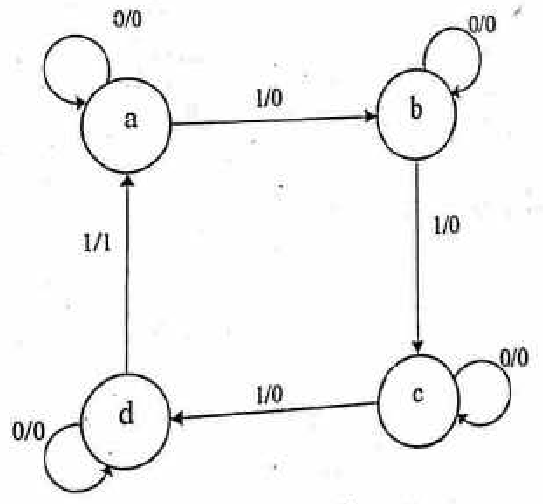


Figure 3.53 State diagram

3.8 DESIGN OF SYNCHRONOUS COUNTERS

The procedure for designing synchronous sequential circuit is given below.

1. From the given specification, draw the state diagram.
2. Plot the state table.
3. Reduce the number of states if possible.
4. Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.
5. Derive the Flip-flop input equations and output equations by using K-map.
6. Draw the logic diagram.

3.8.1 State diagram

State diagram is the graphical representation of state table. In state diagram, a state is represented by a circle, and the transitions between the states are indicated by directed lines connecting the circles. The input value during the present state is marked first and the number after the slash gives the output during the present state with the given input. Figure 3.54 shows a state diagram with states a,b,c,d and e.

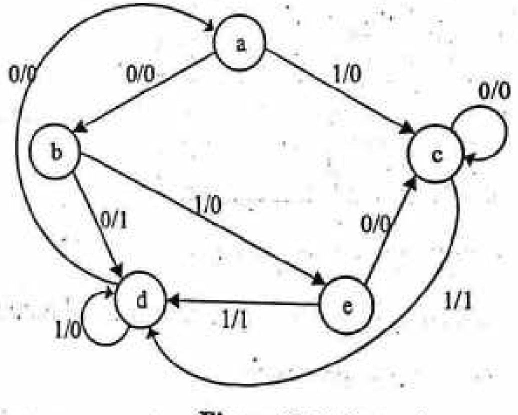


Figure 3.54 State diagram

3.8.2 State table

In the state diagram shown in figure 3.54

(i) The state 'a' has next states 'b' and 'c',
If the input is x=0, the next state is 'b' and the output is 0.
If the input is x=1, the next state is 'c' and the output is 0.

(ii) The state 'b' has next states 'd' and 'e',
If the input is x=0, the next state is 'd' and the output is 1.
If the input is x=1, the next state is 'e' and the output is 0.

(iii) The state 'c' has next states 'c' and 'd',
If the input is x=0, the next state is 'c' and the output is 0.
If the input is x=1, the next state is 'd' and the output is 1.

(iv) The state 'd' has next states 'd' and 'a',
If the input is x=0, the next state is 'a' and the output is 0.
If the input is x=1, the next state is 'd' and the output is 0.

(v) The state 'e' has next states 'd' and 'c',
If the input is x=0, the next state is 'c' and the output is 0.
If the input is x=1, the next state is 'd' and the output is 1.

Present state	Next state		Output	
	x=0	x=1	x=0	x=1
a	b	c	0	0
b	d	e	1	0
c	c	d	0	1
d	a	d	0	0
e	c	d	0	1

Equivalent states

Table 3.56 State table

The state table should contain present state, next state and outputs.

3.8.3 State Minimization or state reduction

The state reduction technique avoids the introduction of redundant states. The reduction in redundant states reduces the number of Flip-flops and logic gates which reduces the cost and circuit complexity of the circuit.

State reduction algorithm is concerned with the procedures for reducing the number of states in the state table. When two states in the state table are equivalent by producing the same next state and same output then one of the states in the state table can be removed without altering the input output relationship.

Two states are said to be equivalent if the two states produces the same next state and same output for each input. When two states are equivalent, one of them can be removed without altering the input-output relationship. In the above state table shown in table 3.56, two states 'c' and 'e' in the present state are equivalent. Both 'c' and 'e' states (present state) are having the same next states and same outputs. Let 'Z' be the output.

'c' and 'e' has next state 'c', if the input is x=0

'c' and 'e' has next state 'd', if the input is x=1

'c' and 'e' has output 0, if the input is x=0

'c' and 'e' has output 1, if the input is x=1

Since states 'c' and 'e' are equivalent we can remove one of these two states. Here the state 'e' is removed, replacing 'e' by 'c'

Present state	Next state		Output Z	
	x=0	x=1	x=0	x=1
a	b	c	0	0
b	d	($\cancel{e}$) c	1	0
c	c	d	0	1
d	a	d	0	0

Table 3.57 Reduced state table or minimized state table

The reduced state diagram has only 4 states. Thus the 5 states have been reduced to 4 states.

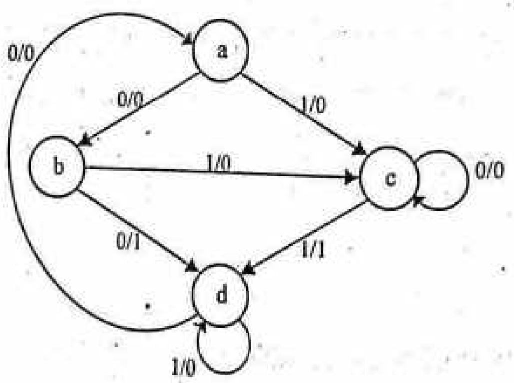


Figure 3.55 Reduced or minimized state diagram

3.8.4 State assignment

In order to design a sequential circuit, it is necessary to assign binary values to the state. For a circuit with 'm' states, the codes must contain 'n' bits, where $2^n \geq m$. For example, with three bits we can assign codes to maximum of 8 states.

Here the reduced state diagram contains 4 states. So the binary code must contain 2 bits, so assign a=00; b=01; c=10 and d=11.

Since each state is represented by two bits, the circuit for the reduced state diagram must contain two Flip-flops. If the state is assigned without state reduction states a, b, c, d and e are assigned with 3 bit binary code which requires 3 Flip-flops for the circuit design. Thus the state reduction reduces 3 Flip-flops to 2 Flip-flops for the given state diagram shown in figure 3.56.

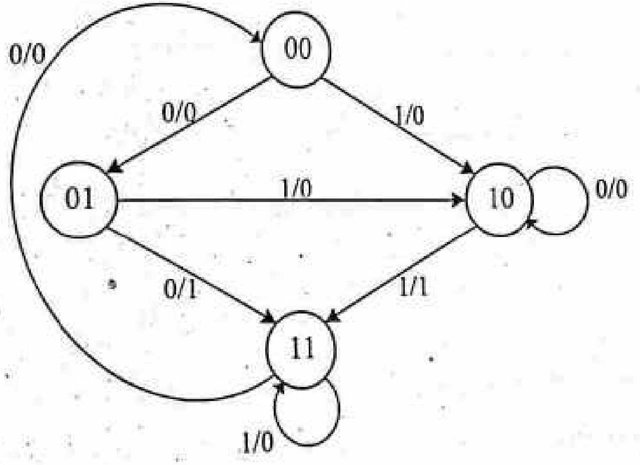


Figure 3.56 State diagram with assigned binary values

Present state AB	Next state A^+B^+		Output Z	
	x=0	x=1	x=0	x=1
00	01	10	0	0
01	11	10	1	0
10	10	11	0	1
11	00	11	0	0

Table 3.58 Reduced state table with assigned binary codes

3.8.5 Excitation table

By using the excitation table of any flip-flop (here we have D flip-flop), determine the excitation(transition) table for the state table shown in table 3.60.

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

Table 3.59 Excitation table of D flip-flop

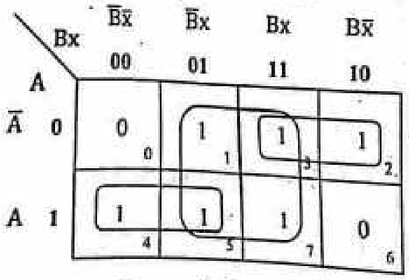
Present state		Input	Next state		Flip-flop inputs		Output
A	B	x	A^+	B^+	D_A	D_A	Z
0	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
0	1	0	1	1	1	1	1
0	1	1	1	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	1	1	1	1	1
1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	0

Table 3.60 Excitation table (Transition table)

3.8.6 Excitation maps

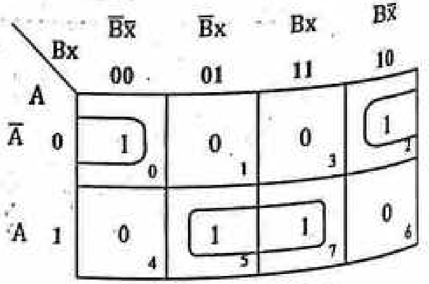
The Flip-flop input equations and output equations can be estimated by using Excitation map. Here the value of D_A, D_B and Z are estimated as follows.

Excitation map for D_A



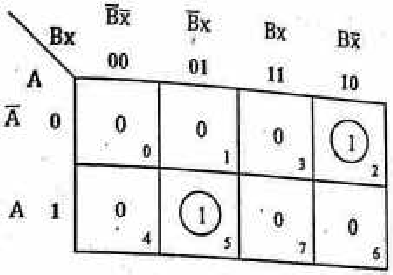
$$D_A = A\bar{B} + x + \bar{A}B$$

Excitation map for D_B



$$D_B = Ax + \bar{A}\bar{x}$$

Excitation map for output 'Z'



$$Z = A\bar{B}x + \bar{A}B\bar{x}$$

3.8.7 Circuit implementation

Implement the sequential circuit using logic gates and D flip-flop as shown in figure 3.57.

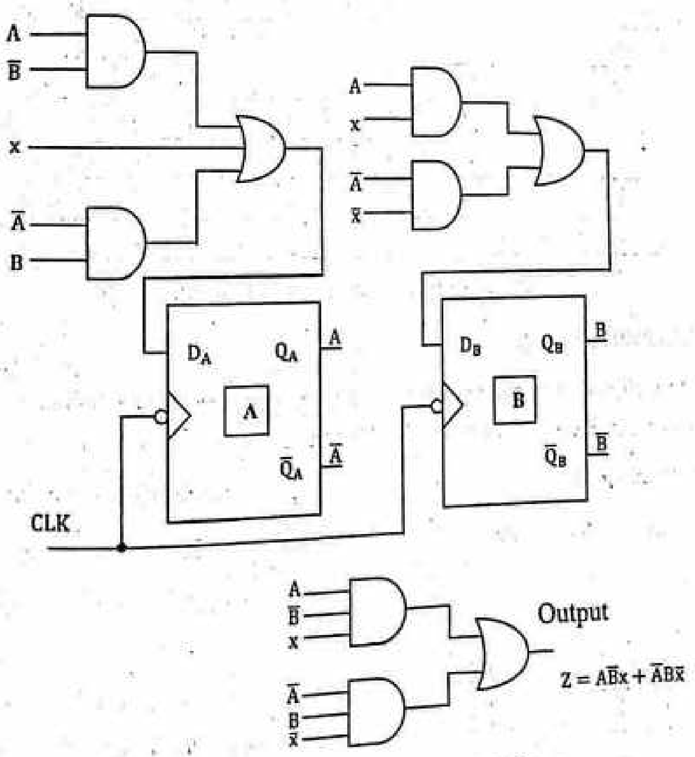


Figure 3.57 Circuit Implementation

3.9 GENERAL MODELS IN SEQUENTIAL CIRCUITS

There are two models in sequential circuits. They are.

1. Mealy model.
2. Moore model.

3.9.1 Mealy model

In the mealy model, the output is a function of both the present state and input.

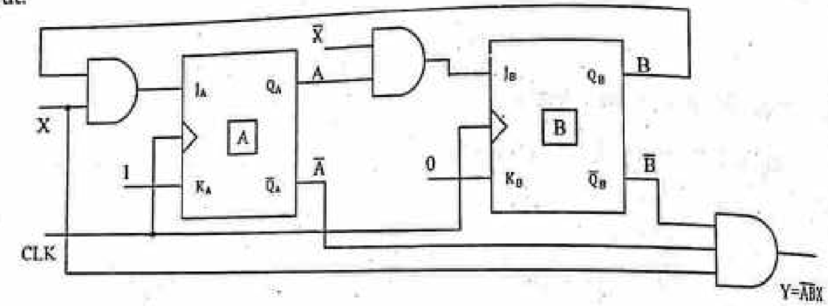


Figure 3.58 Example of Mealy model

In the mealy model shown in figure 3.58, the output is given by

$$Y = \overline{AB}X$$

Here the output Y is a function of present states A, B and input X.

3.9.2 Moore model

In the Moore model, the output is a function of the present state only.

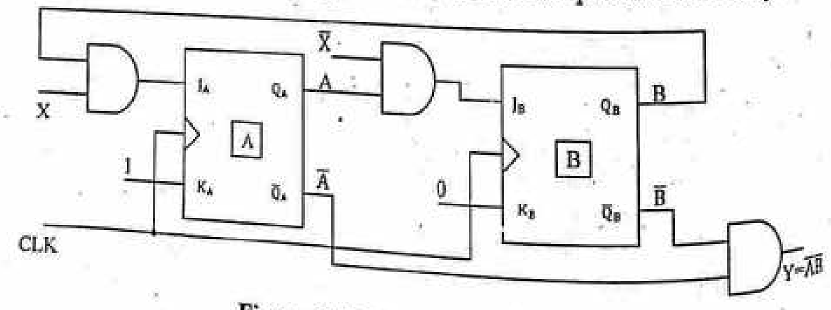


Figure 3.59 Example of Moore model

In the Moore model shown in figure 3.59, the output is given by

$$Y = \overline{AB}$$

Here, the output Y is a function of present states A and B only

Difference between mealy and Moore model sequential circuit.

SI.No	Mealy models	Moore model
1.	Its output is a function of present state and present input.	Its output depends on present state only.
2.	Input changes may affect the output.	Input change does not affect the output.
3.	It requires less number of states for implementing the same function.	It requires more number of states for implementing the same function.

Table 3.61 Difference between Mealy and Moore model

3.9.3 Design Examples

Example 3.5: ***Design a clocked sequential circuit using D Flip-flop for the state diagram shown in figure 3,60.***

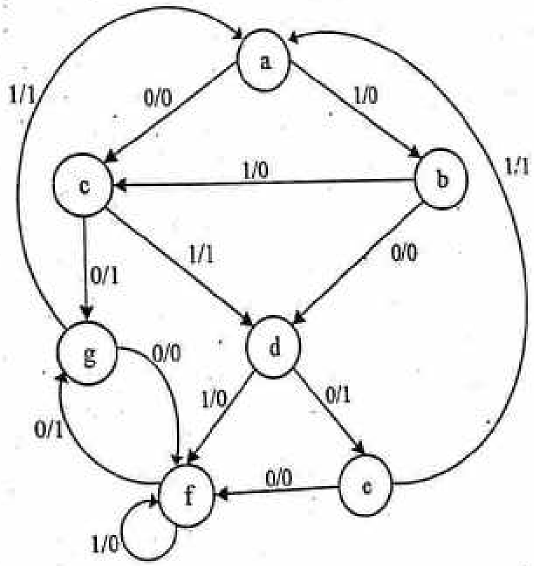


Figure 3.60 State diagram

Solution :

Let 'x' be the input and 'Z' be the output

Step 1: Plot the state table

Present state	Next state		Output	
	x=0	x=1	x=0	x=1
a	c	b	0	0
b	d	c	0	0
c	g	d	1	1
d	e	f	1	0
e	f	a	0	1
f	g	f	1	0
g	f	a	0	1

Equivalent states

Table 3.62 State table

Step 2: Reduce the number of states if possible

Here the state 'e' and state 'g' are equivalent, having same next states and outputs. Here the state 'g' is removed, also replace 'g' by 'e'.

Present state	Next state		Output	
	x=0	x=1	x=0	x=1
a	c	b	0	0
b	d	c	0	0
c	(~~g~~) e	d	1	1
d	e	f	1	0
e	f	a	0	1
f	(~~g~~) e	f	1	0

Equivalent states

Table 3.63 State table with state 'g' reduced

Here the state 'd' and state 'f' are equivalent, having same next states and outputs. Here the state 'f' is removed. Also replace 'f' by 'd'.

Present state	Next state		Output	
	x=0	x=1	x=0	x=1
a	c	b	0	0
b	d	c	0	0
c	e	d	1	1
d	e	(~~f~~) d	1	0
e	(~~f~~) d	a	0	1

Table 3.64 Reduced state table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 5 states, the number of bits required to assign each state must contain 3 bit ($2^3 \geq 5$). Therefore assign a = 000, b = 001 c = 010, d = 011 and e = 100. Use D Flip-flop excitation table to find the values of D_A, D_B and D_C.

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

Table 3.65 Excitation table of D Flip-flop

Present state			Input	Next state			Flip-flop inputs			Output
A	B	C	x	A^+	B^+	C^+	D_A	D_B	D_C	Z
0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	1	0	1	1	0
0	0	1	1	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1
0	1	0	1	0	1	1	0	1	1	1
0	1	1	0	1	0	0	1	0	0	1
0	1	1	1	0	1	1	0	1	1	0
1	0	0	0	0	1	1	0	1	1	0
1	0	0	1	0	0	0	0	0	0	1

Table 3.66 Transition table

Step 4: Derive the Flip-flop input equations and output equations using K-map.

The Flip-flop input equations and output equations must be functions of present state and inputs (assume don't care for unused states).

K-map for D_A

AB \ Cx	$\bar{C}\bar{x}$ 00	$\bar{C}x$ 01	Cx 11	$C\bar{x}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	1_{4}	0_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	0_{9}	X_{11}	X_{10}

$$D_A = B\bar{x}$$

K-map for D_B

AB \ Cx	$\bar{C}\bar{x}$ 00	$\bar{C}x$ 01	Cx 11	$C\bar{x}$ 10
$\bar{A}\bar{B}$ 00	1_{0}	0_{1}	1_{3}	1_{2}
$\bar{A}B$ 01	0_{4}	1_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	1_{8}	0_{9}	X_{11}	X_{10}

$$D_B = Bx + Cx + \bar{B}\bar{x}$$

K-map for D_C

AB \ Cx	$\bar{C}\bar{x}$ 00	$\bar{C}x$ 01	Cx 11	$C\bar{x}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	1_{1}	0_{3}	1_{2}
$\bar{A}B$ 01	0_{4}	1_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	1_{8}	0_{9}	X_{11}	X_{10}

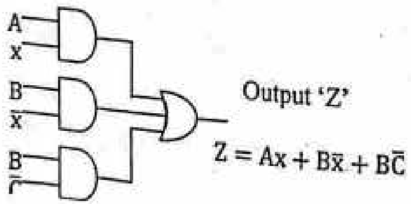
$$D_C = A\bar{C}\bar{x} + Bx + \bar{A}\bar{C}x + \bar{B}C\bar{x}$$

K-map for Z

AB \ Cx	$\bar{C}\bar{x}$ 00	$\bar{C}x$ 01	Cx 11	$C\bar{x}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	1_{4}	1_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$Z = Ax + B\bar{x} + B\bar{C}$$

Logic diagram



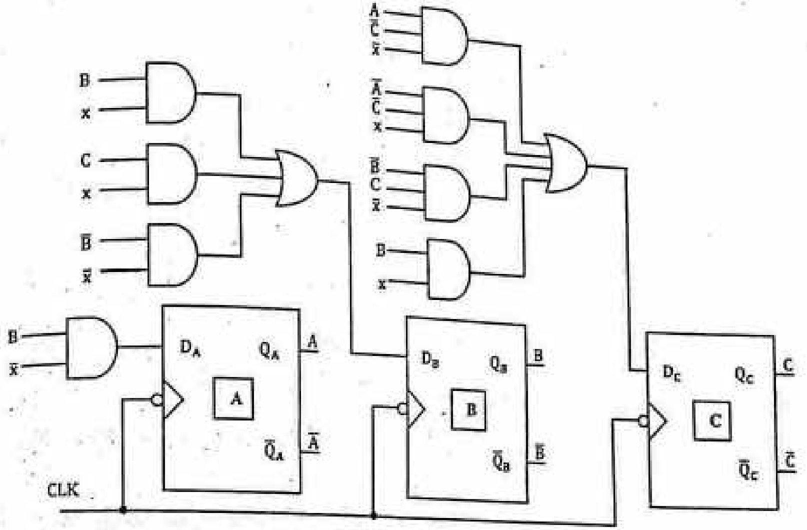


Figure 3.61 Logic diagram

Example 3.6: ***Design a sequential circuit that has 3 flip-flops A, B, C with one input X and one output Y. The circuit is to be designed by treating the unused states as don't care conditions. Use JK flip-flops in the design. State diagram of the circuit is as given below.***

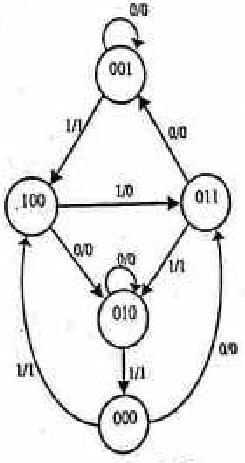


Figure 3.62

Solution:

Let X be the input and Y be the output. Let a = 000 ; b = 001 ; c = 010 ; d = 011 ; e = 100

Step 1: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	d	e	0	1
b	b	e	0	1
c	c	a	0	1
d	b	c	0	1
e	c	d	0	0

Table 3.67 State table

Step 2: Reduce the number of states if possible

Here state reduction is not possible because no two states have same next states and outputs.

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop. Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C and K_C

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.68 Excitation table of JK flip-flop

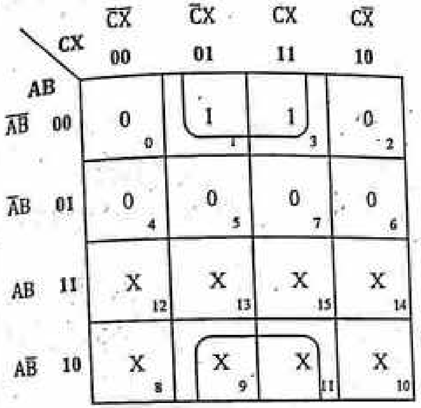
Present state			Input	Next state			Flip-flop inputs						Output
A	B	C	X	A^+	B^+	C^+	J_A	K_A	J_B	K_B	J_C	K_C	Y
0	0	0	0	0	1	1	0	X	1	X	1	X	0
0	0	0	1	1	0	0	1	X	0	X	0	X	1
0	0	1	0	0	0	1	0	X	0	X	X	0	0
0	0	1	1	1	0	0	1	X	0	X	X	1	1
0	1	0	0	0	1	0	0	X	X	0	0	X	0
0	1	0	1	0	0	0	0	X	X	1	0	X	1
0	1	1	0	0	0	1	0	X	X	1	X	0	0
0	1	1	1	0	1	0	0	X	X	0	X	1	1
1	0	0	0	0	1	0	X	1	1	X	0	X	0
1	0	0	1	0	1	1	X	1	1	X	1	X	0

Table 3.69 Transition table

Step 4: Derive the Flip-flop input equations and output equations using K-map

The Flip-flop input equations and output equations must be a function of present state A, B, C and input X (assume don't cares for unused states)

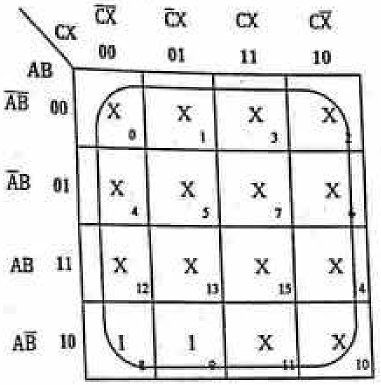
K-map for J_A



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	1	1	0
$\bar{A}B$ 01	0	0	0	0
AB 11	X	X	X	X
$A\bar{B}$ 10	X	X	X	X

$J_A = \bar{B}X$

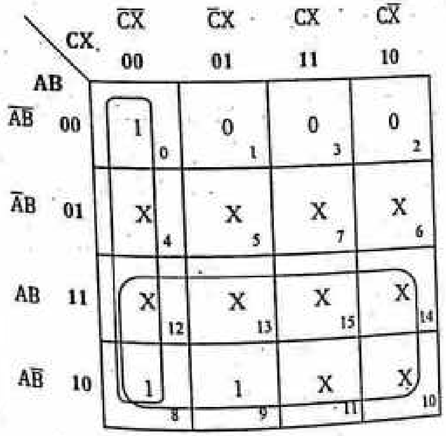
K-map for K_A



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X	X	X	X
$\bar{A}B$ 01	X	X	X	X
AB 11	X	X	X	X
$A\bar{B}$ 10	1	1	X	X

$K_A = 1$

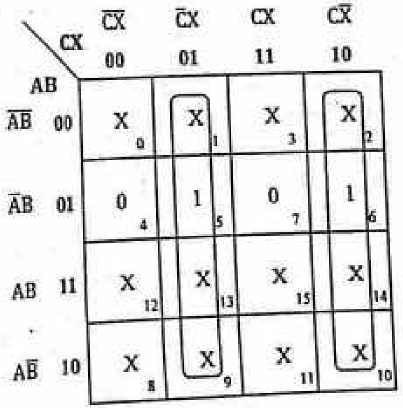
K-map for J_B



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	1	0	0	0
$\bar{A}B$ 01	X	X	X	X
AB 11	X	X	X	X
$A\bar{B}$ 10	1	1	X	X

$J_B = A + \bar{C}\bar{X}$

K-map for K_B



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X	X	X	X
$\bar{A}B$ 01	0	1	0	1
AB 11	X	X	X	X
$A\bar{B}$ 10	X	X	X	X

$K_B = \bar{C}X + C\bar{X}$

K-map for J_C

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	1_{0}	0_{1}	X_{3}	X_{2}
$\overline{A}B$ 01	0_{4}	0_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$J_C = AX + \overline{ABX}$$

K-map for K_C

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X_{0}	X_{1}	1_{3}	0_{2}
$\overline{A}B$ 01	X_{4}	X_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$$K_C = X$$

K-map for Y

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	1_{1}	1_{3}	0_{2}
$\overline{A}B$ 01	0_{4}	1_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	0_{9}	X_{11}	X_{10}

$$Y = \overline{A}X$$

Step 5: Draw the logic diagram

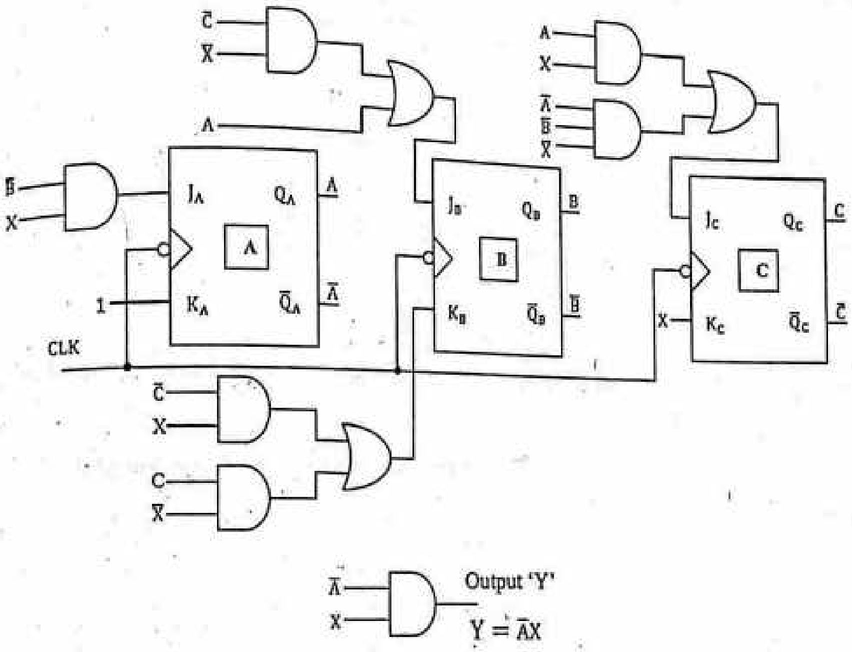


Figure 3.63 Logic diagram

Example 3.7: ***Design a synchronous sequential circuit using JK Flip-flop :o generate the following sequence and repeat 0,1,2,4,5,6.***

Solution:

Step1: Draw the state diagram.

Let $a=0_{10}=000$; $b=1_{10}=001$; $c=2_{10}=010$; $d=4_{10}=100$; $e=5_{10}=101$; $f=6_{10}=110$

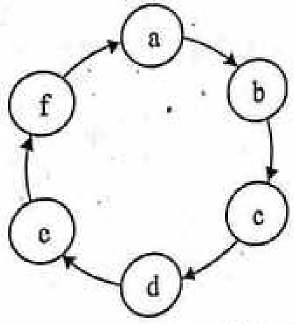


Figure 3.64 State diagram

Step 2: Plot the state table.

Present state	Next state
a	b
b	c
c	d
d	e
e	f
f	a

Table 3.70 State table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C and K_C

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.71 Excitation table of JK Flip-flop

Present state			Next state			Flip-flop inputs					
A	B	C	A^+	B^+	C^+	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	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

Table 3.72 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

Assume don't cares for unused states.

K-map for J_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 0	0 1	X 3	1 2
A 1	X 4	X 5	X 7	X 6

$J_A = B$

K-map for K_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X 0	X 1	X 3	X 2
A 1	0 4	0 5	X 7	1 6

$K_A = B$

K-map for J_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 0	1 1	X 3	X 2
A 1	0 4	1 5	X 7	X 6

$J_B = C$

K-map for K_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X 0	X 1	X 3	1 2
A 1	X 4	X 5	X 7	1 6

$K_B = 1$

K-map for J_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1 0	X 1	X 3	0 2
A 1	1 4	X 5	X 7	0 6

$J_C = \overline{B}$

K-map for K_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X 0	1 1	X 3	X 2
A 1	X 4	1 5	X 7	X 6

$K_C = 1$

Logic diagram

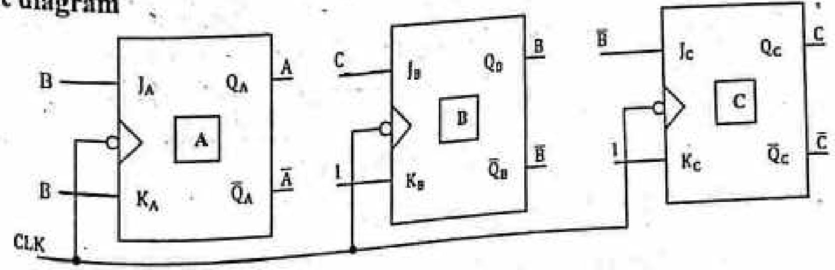


Figure 3.65 Logic diagram

Example 3.8: Design a 3-bit Synchronous counter using JK flip-flops.

Solution:

Step 1: Draw the logic diagram

Let a=000; b=001; c=010; d=011; e=100; f=101; g=110; h=111

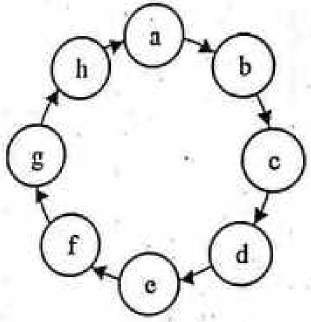


Figure 3.66 State diagram

Step 2: Plot the state table

Present state	Next state
a	b
b	c
c	d
d	e
e	f
f	g
g	h
h	a

Table 3.73 State table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C and K_C.

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.74 Excitation table of JK Flip-flop

Present state			Next state			Flip-flop inputs					
A	B	C	A^+	B^+	C^+	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	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	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	0	0	X	1	X	1	X	1

Table 3.75 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

K-map for J_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 ${}_0$	0 ${}_1$	1 ${}_3$	0 ${}_2$
A 1	X ${}_4$	X ${}_5$	X ${}_7$	X ${}_6$

$J_A = BC$

K-map for K_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X ${}_0$	X ${}_1$	X ${}_3$	X ${}_2$
A 1	0 ${}_4$	0 ${}_5$	1 ${}_7$	0 ${}_6$

$K_A = BC$

K-map for J_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 ${}_0$	1 ${}_1$	X ${}_3$	X ${}_2$
A 1	0 ${}_4$	1 ${}_5$	X ${}_7$	X ${}_6$

$J_B = C$

K-map for K_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X ${}_0$	X ${}_1$	1 ${}_3$	0 ${}_2$
A 1	X ${}_4$	X ${}_5$	1 ${}_7$	0 ${}_6$

$K_B = C$

K-map for J_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1	X	X	1
A 1	1	X	X	1

$J_C = 1$

K-map for K_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X	1	1	X
A 1	X	1	1	X

$K_C = 1$

Logic diagram

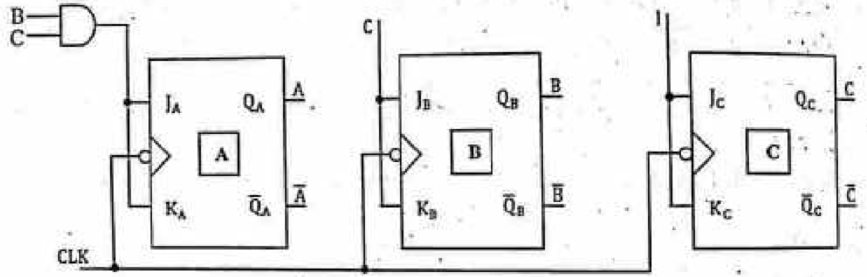


Figure 3.67 Logic diagram

Example 3.9 : Design a counter to count the sequence 0,1,2,4,5,6 using SR-FF.

Solution:

Step 1: Draw the state diagram.

Let a=0_{10}=000; b=1_{10}=001; c=2_{10}=010; d=4_{10}=100; e=5_{10}=101; f=6_{10}=110

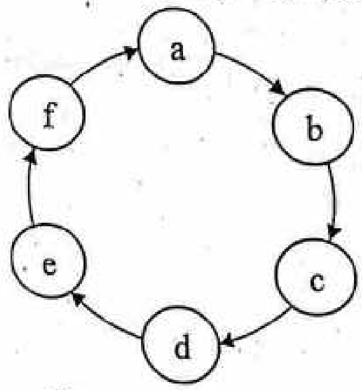


Figure 3.68 State diagram

Step 2: Plot the state table.

Present state	Next state
a	b
b	c
c	d
d	c
e	f
f	a

Table 3.76 State table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Use the following SR Flip-flop excitation table to find the value of S_A, R_A, S_B, R_B, S_C and R_C.

A	A^+	S_A	R_A
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Table 3.77 Excitation table of SR Flip-flop

Present state			Next state			Flip-flop inputs					
A	B	C	A^+	B^+	C^+	S_A	R_A	S_B	R_B	S_C	R_C
0	0	0	0	0	1	0	X	0	X	1	0
0	0	1	0	1	0	0	X	1	0	0	1
0	1	0	1	0	0	1	0	0	1	0	X
1	0	0	1	0	1	X	0	0	X	1	0
1	0	1	1	1	0	X	0	1	0	0	1
1	1	0	0	0	0	0	1	0	1	0	X

Table 3.78 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

Assume don't cares for unused states.

K-map for S_A

A \ BC	$\overline{BC}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0	0	X	1
A 1	X	X	X	0

$S_A = \overline{A}B$

K-map for R_A

A \ BC	$\overline{BC}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X	X	X	0
A 1	0	0	X	1

$R_A = AB$

K-map for S_B

A \ BC	$\overline{BC}$ 00	$\overline{B}C$ 01	$\overline{BC}$ 11	$B\overline{C}$ 10
$\overline{A}$ 0	0	1	X	0
A 1	0	1	X	0

$S_B = C$

K-map for R_B

A \ BC	$\overline{BC}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	X	0	X	1
A 1	X	0	X	1

$R_B = B$

K-map for S_C

A \ BC	$\overline{BC}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1	0	X	0
A 1	1	0	X	0

$S_C = \overline{BC}$

K-map for R_C

A \ BC	$\overline{BC}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0	1	X	X
A 1	0	1	X	X

$R_C = C$

Logic diagram

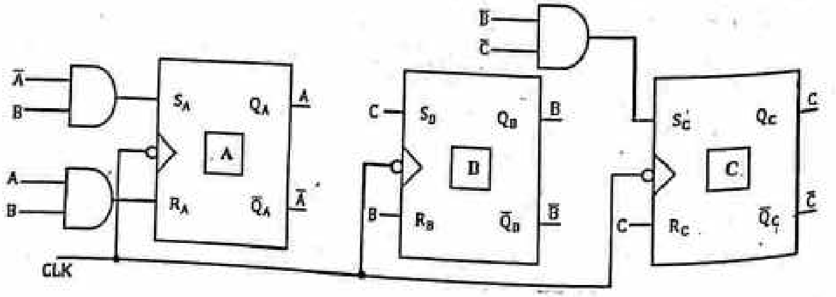


Figure 3.69 Logic diagram

*Example 3.10 : **Design a shift register using JK Flip-flops.***

Solution:

Let X be the control input. If X=0 shift left operation is performed. If X=1 Shift right operation is performed. The transition table of 3-bit shift register using JK Flip-flop is shown in table 3.80.

Use JK Flip-flop excitation table to find the values of J_A, K_A, J_B, K_B, J_C and K_C.

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.79 Excitation table of JK Flip-flop

Present state			Input	Next state			Flip-flop inputs					
A	B	C	X	A^+	B^+	C^+	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	0	0	0	X	0	X	0	X
0	0	0	1	0	0	0	0	X	0	X	0	X
0	0	1	0	0	1	0	0	X	1	X	X	1
0	0	1	1	0	0	0	0	X	0	X	X	1
0	1	0	0	1	0	0	1	X	X	1	0	X
0	1	0	1	0	0	1	0	X	X	1	1	X
0	1	1	0	1	1	0	1	X	X	0	X	1
0	1	1	1	0	0	1	0	X	X	1	X	0
1	0	0	0	0	0	0	X	1	0	X	0	X
1	0	0	1	0	1	0	X	1	1	X	0	X
1	0	1	0	0	1	0	X	1	1	X	X	1
1	0	1	1	0	1	0	X	1	1	X	X	1
1	1	0	0	1	0	0	X	0	X	1	0	X
1	1	0	1	0	1	1	X	1	X	0	1	X
1	1	1	0	1	1	0	X	0	X	0	X	1
1	1	1	1	0	1	1	X	1	X	0	X	0

Table 3.80 Transition table

Then derive the Flip-flop input equations using K-map.

K-map for J_A

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0	0	0	0
$\overline{A}B$ 01	1	0	0	1
AB 11	X	X	X	X
$A\overline{B}$ 10	X	X	X	X

$$J_A = B\overline{X}$$

K-map for K_A

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X	X	X	X
$\overline{A}B$ 01	X	X	X	X
AB 11	0	1	1	0
$A\overline{B}$ 10	1	1	1	1

$$K_A = \overline{B} + X$$

K-map for J_B

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0	0	0	1
$\overline{A}B$ 01	X	X	X	X
AB 11	X	X	X	X
$A\overline{B}$ 10	0	1	1	1

$$J_B = AX + C\overline{X}$$

K-map for K_B

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X	X	X	X
$\overline{A}B$ 01	1	1	1	0
AB 11	1	0	0	0
$A\overline{B}$ 10	X	X	X	X

$$K_B = \overline{C}\overline{X} + \overline{A}X$$

K-map for J_C

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0	0	X	X
$\overline{A}B$ 01	0	1	X	X
AB 11	0	1	X	X
$A\overline{B}$ 10	0	0	X	X

$$J_C = BX$$

K-map for K_C

AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X	X	1	1
$\overline{A}B$ 01	X	X	0	1
AB 11	X	X	0	1
$A\overline{B}$ 10	X	X	1	1

$$K_C = \overline{B} + \overline{X}$$

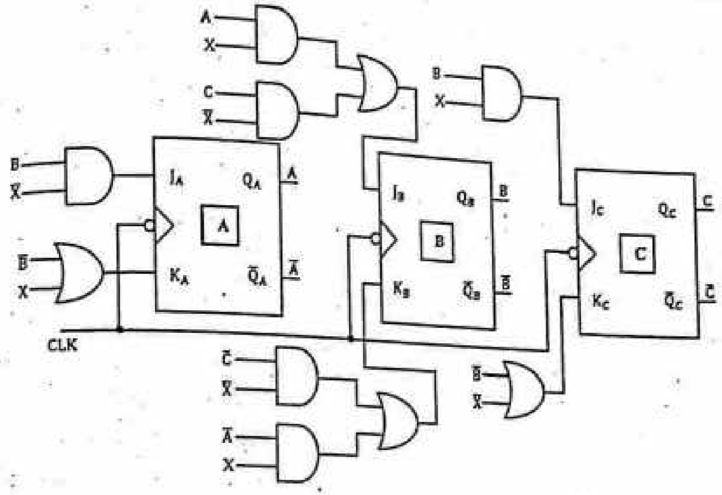


Figure 3.70 Logic diagram

Example 3.11: A sequential circuit has one input and one output. The state diagram is shown in figure 3.71. Design the sequential circuit with a RS Flip-flop.

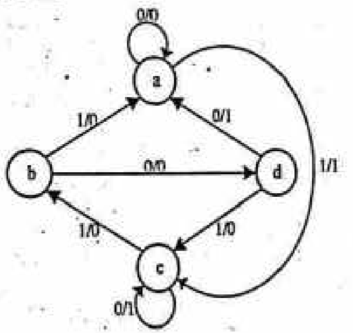


Figure 3.71 State diagram

Solution : Let 'X' be the input and 'Z' be the output.

Step 1: Plot the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	c	0	1
b	d	a	0	0
c	c	b	1	0
d	a	c	1	0

Table 3.81 State table

Step 2: Reduce the number of states if possible

Here state reduction is not possible because no two states have same next states and outputs.

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop. Since there are 4 states the number of bits required to assign each state must contain 2 bit ($2^2 \geq 4$). Therefore assign $a = 00$, $b = 01$, $c = 10$ and $d = 11$. Use the following SR Flip-flop excitation table to find the value of S_A, R_A, S_B and R_B.

A	A^+	S_A	R_A
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Table 3.82 Excitation table of SR Flip-flop

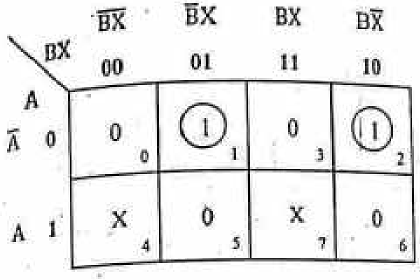
Present state		Input	Next state		Flip-flop inputs				Output
A	B	X	A^+	B^+	S_A	R_A	S_B	R_B	Z
0	0	0	0	0	0	X	0	X	0
0	0	1	1	0	1	0	0	X	1
0	1	0	1	1	1	0	X	0	0
0	1	1	0	0	0	X	0	1	0
1	0	0	1	0	X	0	0	X	1
1	0	1	0	1	0	1	1	0	0
1	1	0	0	0	0	1	0	1	1
1	1	1	1	0	X	0	0	1	0

Table 3.83 Transition table

Step 4: Derive the Flip-flop input equations and output equations using K-map.

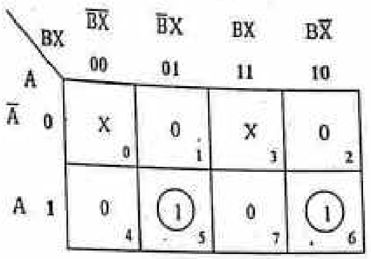
The Flip-flop input equations and output equations must be a function of present states A, B and input X.

K-map for S_A



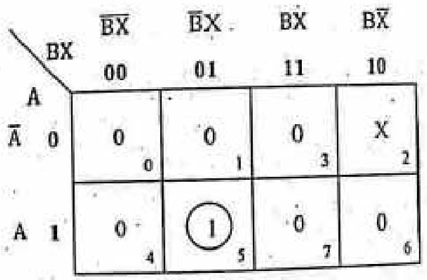
$$S_A = \overline{AB}X + \overline{A}B\overline{X}$$

K-map for R_A



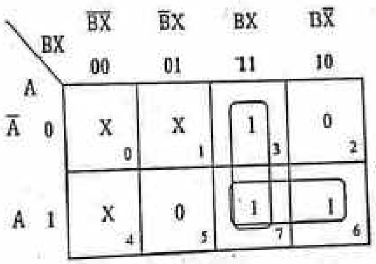
$$R_A = A\overline{B}X + AB\overline{X}$$

K-map for S_B



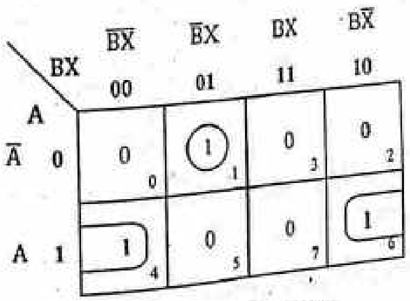
$$S_B = A\overline{B}X$$

K-map for R_B



$$R_B = AB + BX$$

K-map for Z



$$Z = A\overline{X} + \overline{AB}X$$

Step 5: Draw the logic diagram

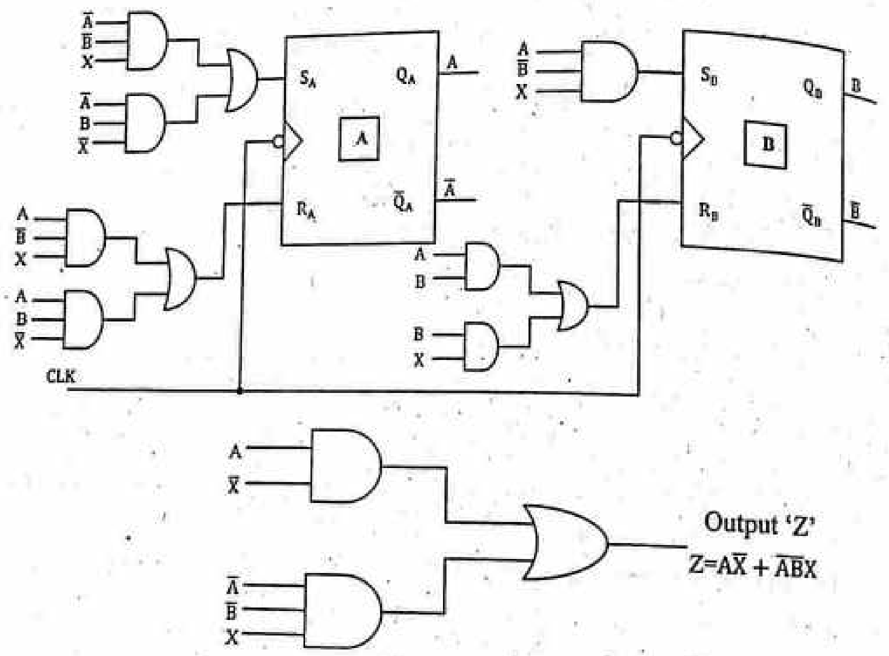


Figure 3.72 Logic diagram

Example 3.12:Design the sequential circuit for the state diagram shown in figure 3.73. Use JK Flip-flops.

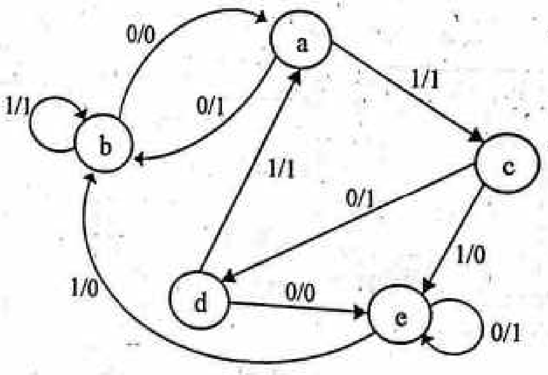


Figure 3.73 State diagram

Solution:

Let X be the input and Z be the output.

Step 1: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	b	c	1	1
b	a	b	0	1
c	d	e	1	0
d	e	a	0	1
e	e	b	1	0

Table 3.84 State table

Step 2: Reduce the number of states if possible

Here state reduction is not possible because no two states have same next states and outputs.

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop. Assign a = 000, b = 001, c = 010, d = 011 and e = 100

Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C and K_C

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.85 Excitation table of JK flip-flop

Present state			Input	Next state			Flip-flop inputs						Output
A	B	C	X	A^+	B^+	C^+	J_A	K_A	J_B	K_B	J_C	K_C	Z
0	0	0	0	0	0	1	0	X	0	X	1	X	1
0	0	0	1	0	1	0	0	X	1	X	0	X	1
0	0	1	0	0	0	0	0	X	0	X	X	1	0
0	0	1	1	0	0	1	0	X	0	X	X	0	1
0	1	0	0	0	1	1	0	X	X	0	1	X	1
0	1	0	1	1	0	0	1	X	X	1	0	X	0
0	1	1	0	1	0	0	1	X	X	1	X	1	0
0	1	1	1	0	0	0	0	X	X	1	X	1	1
1	0	0	0	1	0	0	X	0	0	X	0	X	1
1	0	0	1	0	0	1	X	1	0	X	1	X	0

Table 3.86 Transition table

Step 4: Derive the Flip-flop input equations and output equations using K-map

The Flip-flop input equations and output equations must be a function of present state A, B, C and input X (assume don't cares for unused states)

K-map for J_A

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	0_{4}	1_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$$J_A = B\bar{C}X + BC\bar{X}$$

K-map for K_A

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$K_A = X$$

K-map for J_B

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	1_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	0_{9}	X_{11}	X_{10}

$$J_B = \bar{A}\bar{C}X$$

K-map for K_B

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	0_{4}	1_{5}	1_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$$K_B = X + C$$

K-map for J_C

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	1_{0}	0_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	1_{4}	0_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$J_C = \bar{A}\bar{X} + AX$$

K-map for K_C

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	0_{3}	1_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	1_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$$K_C = \bar{X} + B$$

K-map for Z

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	1_{0}	1_{1}	1_{3}	0_{2}
$\bar{A}B$ 01	1_{4}	0_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	1_{8}	0_{9}	X_{11}	X_{10}

$$Z = \bar{C}\bar{X} + CX + \bar{A}\bar{B}X$$

Step 5: Draw the logic diagram

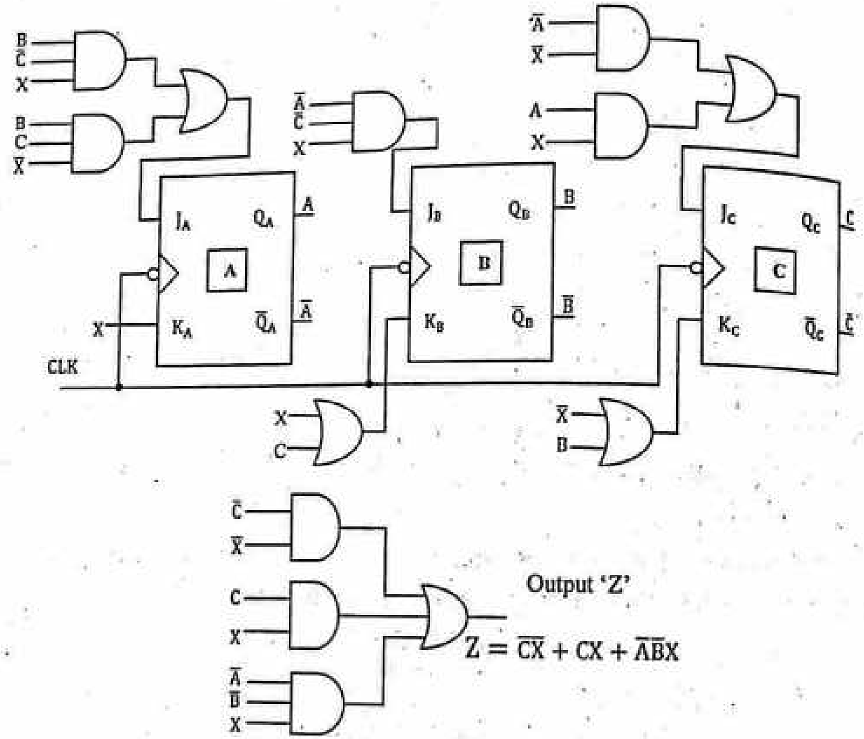


Figure 3.74 Logic diagram

Example 3.13: *Design a clocked sequential machine using T-Flip-flops for the following state diagram shown in figure 3.75. Use state reduction if possible. Also use straight binary state assignment.*

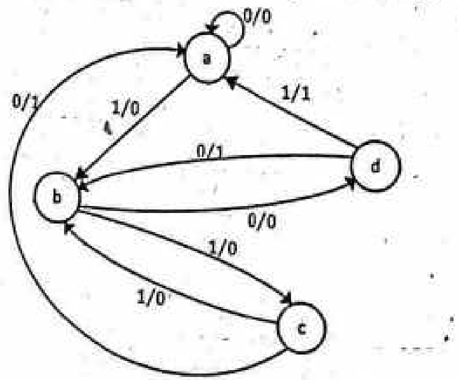


Figure 3.75 State diagram

Solution : Let X be the input and Z be the output

Step 1: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	d	c	0	0
c	a	b	1	0
d	b	a	1	1

Table 3.87 State table

Step 2: Reduce the number of states if possible

Here state reduction is not possible.

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 4 states, assign a = 00, b = 01, c = 10 and d = 11. Use the following T Flip-flop excitation table to find the value of T_A and T_B.

A	A^+	T_A
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.88 Excitation table of T Flip-flop

Present state		Input	Next state		Flip-flop inputs		Output
A	B	X	A^+	B^+	T_A	T_B	Y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	1	0	1	1	0
1	0	0	0	0	1	0	1
1	0	1	0	1	1	1	0
1	1	0	0	1	1	0	1
1	1	1	0	0	1	1	1

Table 3.89 Transition table

Step 4: Derive the Flip-flop input equations and output equations.

K-map for T_A

A \ BX	$\overline{B}\,\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0_{0}	0_{1}	1_{3}	1_{2}
A 1	1_{4}	1_{5}	1_{7}	1_{6}

$T_A = A + B$

K-map for T_B

A \ BX	$\overline{B}\,\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0_{0}	1_{1}	1_{3}	0_{2}
A 1	0_{4}	1_{5}	1_{7}	0_{6}

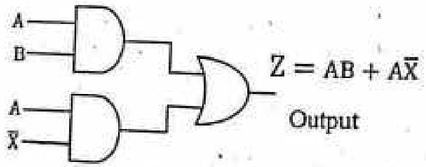
$T_B = X$

K-map for Z

A \ BX	$\overline{B}\,\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0_{0}	0_{1}	0_{3}	0_{2}
A 1	1_{4}	0_{5}	1_{7}	1_{6}

$Z = AB + A\overline{X}$

Step 5: Draw the logic diagram



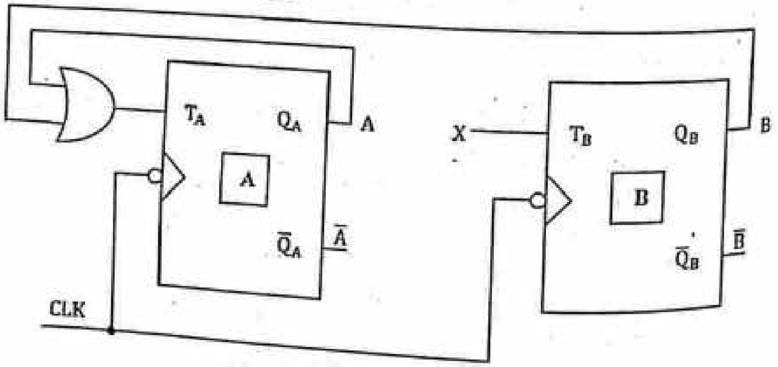


Figure 3.76 Logic diagram

Example 3.14: ***Design the sequential circuit specified by the state diagram shown in figure 3.77 using JK Flip-flops.***

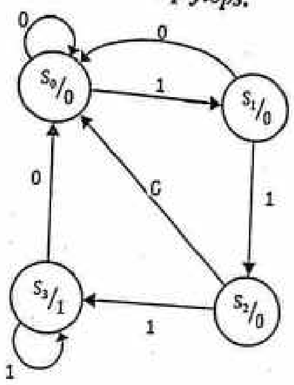


Figure 3.77 State diagram

Solution : Let X be the input and Z be the output.

Step 1: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
S_0	S0	S1	0	0
S_1	S0	S2	0	0
S_2	S0	S3	0	1
S_3	S0	S3	0	1

Equivalent states

Table 3.90 State table

Step 2: Reduce the number of states if possible

Here the states S_2 and S_3 are equivalent, having same next states and outputs. So remove the state S_3, also replace S_3 by S_2.

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
S_0	S_0	S_1	0	0
S_1	S_0	S_2	0	0
S_2	S_0	S_2	0	1

Table 3.91 Reduced state table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 3 states, assign $S_0 = 00$, $S_1 = 01$, $S_2 = 10$. Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B and K_B

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.92 Excitation table of JK Flip-flop

Present state		Input	Next state		Flip-flop inputs				Output
A	B	X	A^+	B^+	J_A	K_A	J_B	K_B	Z
0	0	0	0	0	0	X	0	X	0
0	0	1	0	1	0	X	1	X	0
0	1	0	0	0	0	X	X	1	0
0	1	1	1	0	1	X	X	1	0
1	0	0	0	0	X	1	0	X	0
1	0	1	1	0	X	0	0	X	1

Table 3.93 Transition table

Step 4: Derive the Flip-flop input equations and output equations using K-map.

The Flip-flop input equations and output equations must be a function of present states A, B and input X (assume don't cares for unused states)

K-map for J_A

A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0_{0}	0_{1}	1_{3}	0_{2}
A 1	X_{4}	X_{5}	X_{7}	X_{6}

$J_A = BX$

K-map for K_A

A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	X_{0}	X_{1}	X_{3}	X_{2}
A 1	1_{4}	0_{5}	X_{7}	X_{6}

$K_A = \overline{X}$

K-map for J_B

A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0_{0}	1_{1}	X_{3}	X_{2}
A 1	0_{4}	0_{5}	X_{7}	X_{6}

$J_B = \overline{A}X$

K-map for K_B

A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	X_{0}	X_{1}	1_{3}	1_{2}
A 1	X_{4}	X_{5}	X_{7}	X_{6}

$K_B = 1$

K-map for Z

A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0_{0}	0_{1}	0_{3}	0_{2}
A 1	0_{4}	1_{5}	X_{7}	X_{6}

$Z = AX$

Step 5: Draw the logic diagram

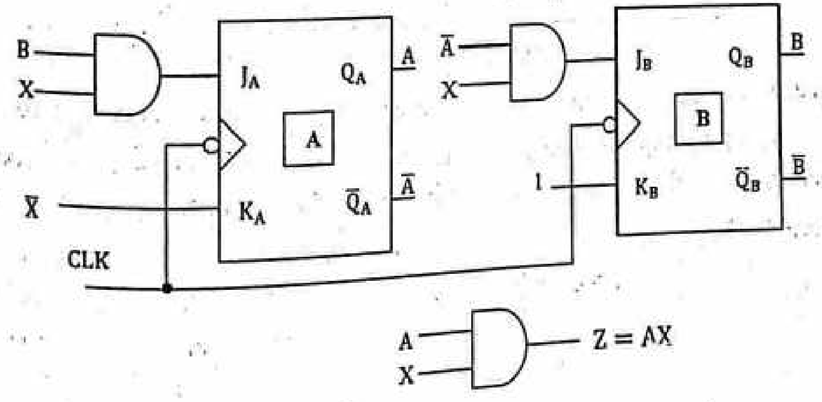


Figure 3.78 Logic diagram

Example 3.15: Design a sequential circuit using RS Flip-flops for the state table given below in table 3.94 using minimum number of Flip-flops.

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Table 3.94 State table

Solution: Given

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Equivalent states

Table 3.95 State reduction

Here the states 'e' and 'g' are equivalent having same next states and outputs. So remove the state 'g' and replace 'g' by 'e'.

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Equivalent states

Table 3.96 State table with state 'g' reduced

Also the states 'd' and 'f' are equivalent. So remove the state 'f' and also replace 'f' by 'd'. So the above state table given in table 3.96 can be further reduced as shown in table 3.97.

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Table 3.97 Reduced state table

Assign binary values to the states and plot the transition table by choosing the type of Flip-flop. Since there are 5 states, the number of bits required to assign each state must contain 3-bit. ($2^3 \geq 5$). Therefore assign a=000, b=001, c=010, d=011 and e=100.

Use the following SR Flip-flop excitation table to find the value of S_A, R_A, S_B, R_B, S_C and R_C.

A	A^+	S_A	R_A
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

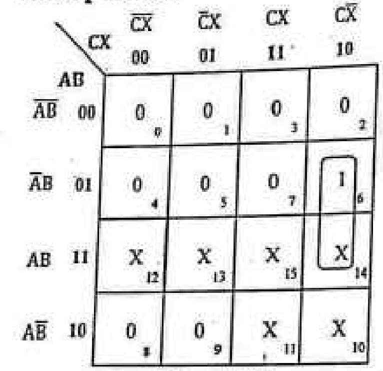
Table 3.98 Excitation table of SR Flip-flop

Present state			Input	Next state			Flip-flop inputs						Output
A	B	C	X	A^+	B^+	C^+	S_A	R_A	S_B	R_B	S_C	R_C	Z
0	0	0	0	0	0	0	0	X	0	X	0	X	0
0	0	0	1	0	0	1	0	X	0	X	1	0	0
0	0	1	0	0	1	0	0	X	1	0	0	1	0
0	0	1	1	0	1	1	0	X	1	0	X	0	0
0	1	0	0	0	0	0	0	X	0	1	0	X	0
0	1	0	1	0	1	1	0	X	X	0	1	0	0
0	1	1	0	1	0	0	1	0	0	1	0	1	0
0	1	1	1	0	1	1	0	X	X	0	X	0	1
1	0	0	0	0	0	0	0	1	0	X	0	X	0
1	0	0	1	0	1	1	0	1	1	0	1	0	1

Table 3.99 Transition table

Then derive the Flip-flop input equations and output equations using K-map.

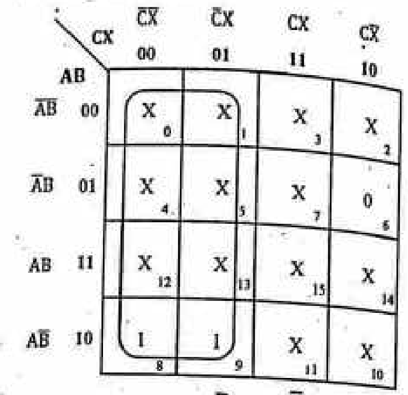
K-map for S_A



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\overline{A}B$ 01	0_{4}	0_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	0_{9}	X_{11}	X_{10}

$$S_A = BC\overline{X}$$

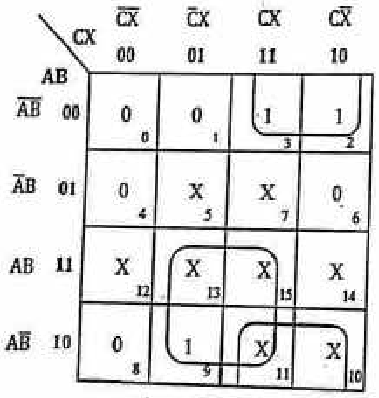
K-map for R_A



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\overline{A}B$ 01	X_{4}	X_{5}	X_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	1_{8}	1_{9}	X_{11}	X_{10}

$$R_A = \overline{C}$$

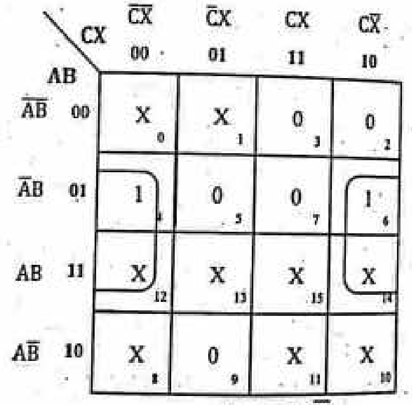
K-map for S_B



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	0_{1}	1_{3}	1_{2}
$\overline{A}B$ 01	0_{4}	X_{5}	X_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$S_B = \overline{B}C + AX$$

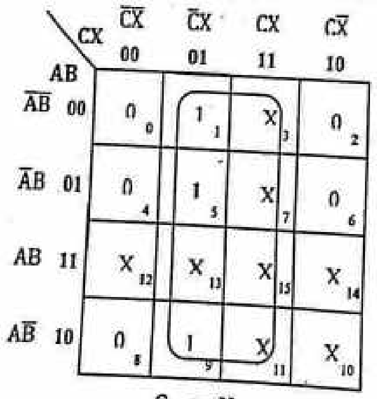
K-map for R_B



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X_{0}	X_{1}	0_{3}	0_{2}
$\overline{A}B$ 01	1_{4}	0_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	X_{8}	0_{9}	X_{11}	X_{10}

$$R_B = B\overline{X}$$

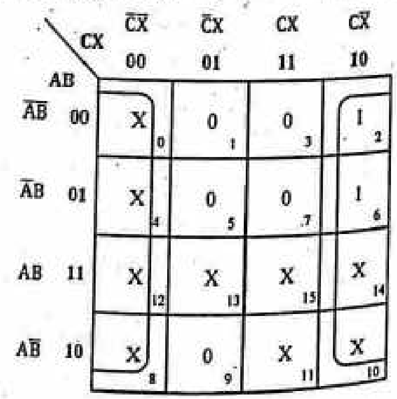
K-map for S_C



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	1_{1}	X_{3}	0_{2}
$\overline{A}B$ 01	0_{4}	1_{5}	X_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$S_C = X$$

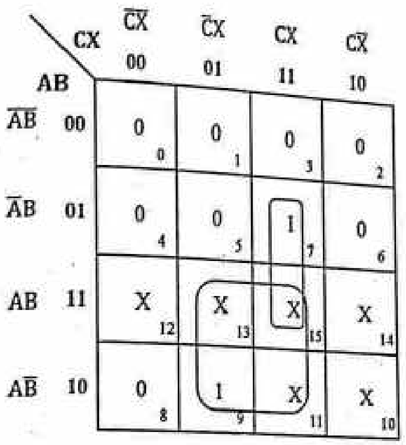
K-map for R_C



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X_{0}	0_{1}	0_{3}	1_{2}
$\overline{A}B$ 01	X_{4}	0_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	X_{8}	0_{9}	X_{11}	X_{10}

$$R_C = \overline{X}$$

K-map for Z



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\overline{A}B$ 01	0_{4}	0_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$Z = AX + BCX$$

The logic diagram is shown in figure 3.79.

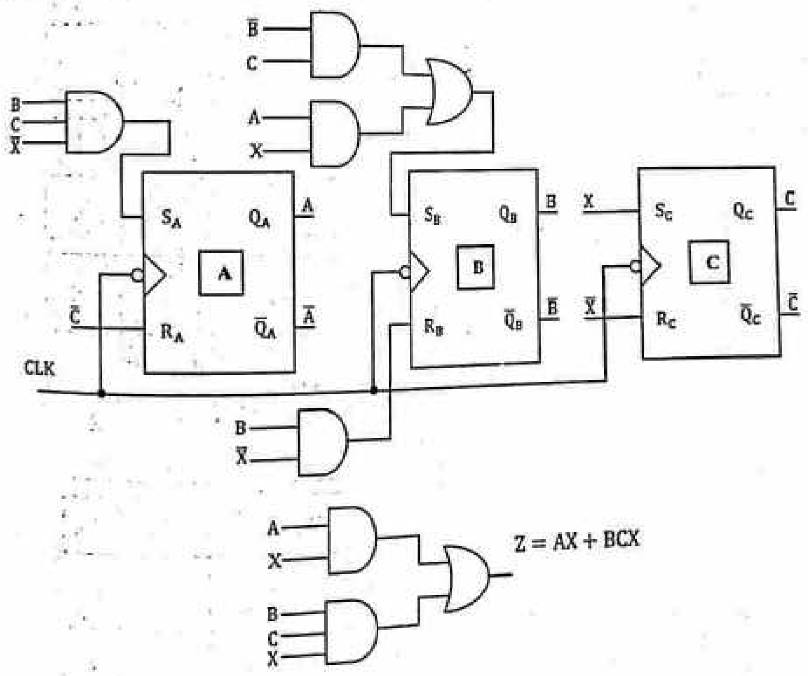


Figure 3.79 Logic diagram

Example 3.16 : Design the sequential circuit specified by the following state diagram using T flip-flops.

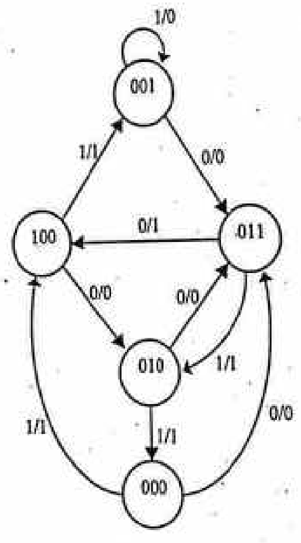


Figure 3.80

Solution:

Let X be the input and Z be the output. Let a = 000 ; b = 001 ; c = 010 ; d = 011 ; e = 100

Step 1: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	d	e	0	1
b	d	b	0	0
c	d	a	0	1
d	e	c	1	1
e	c	b	0	1

Table 3.100 State table

Step 2: Reduce the number of states if possible

Here state reduction is not possible because no two states have same next states and outputs.

Step 3: Plot the transition table using the excitation table of T Flip-flop.

A	A^+	T_A
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.101 Excitation table of T Flip-flop

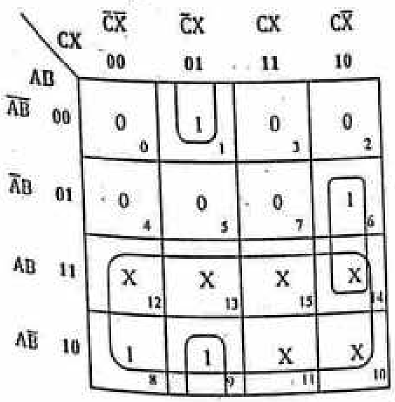
Present state			Input	Next state			Flip-flop inputs			Output
A	B	C	X	A^+	B^+	C^+	T_A	T_B	T_C	Z
0	0	0	0	0	1	1	0	1	1	0
0	0	0	1	1	0	0	1	0	0	1
0	0	1	0	0	1	1	0	1	0	0
0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	0	1	1	0	0	1	0
0	1	0	1	0	0	0	0	1	0	1
0	1	1	0	1	0	0	1	1	1	1
0	1	1	1	0	1	0	0	0	1	1
1	0	0	0	0	1	0	1	1	0	0
1	0	0	1	0	0	1	1	0	1	1

Table 3.102 Transition table

Step 4: Derive the Flip-flop input equations and output equations.

The Flip-flop input equations and output equations must be a function of present state A, B, C and input X (assume don't cares for unused states).

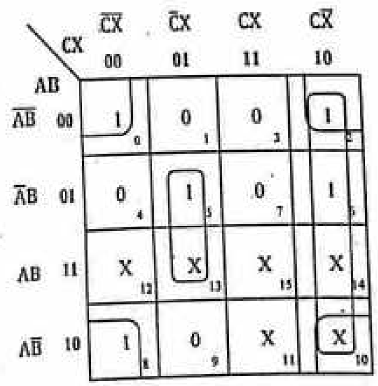
K-map for T_A



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	1	0	0
$\bar{A}B$ 01	0	0	0	1
AB 11	X	X	X	X
$A\bar{B}$ 10	1	1	X	X

$$T_A = A + \bar{B}\bar{C}X + BC\bar{X}$$

K-map for T_B



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	1	0	0	1
$\bar{A}B$ 01	0	1	0	1
AB 11	X	X	X	X
$A\bar{B}$ 10	1	0	X	X

$$T_B = C\bar{X} + \bar{B}\bar{X} + B\bar{C}X$$

K-map for T_C

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	1	0	0	0
$\bar{A}B$ 01	1	0	1	1
AB 11	X	X	X	X
$A\bar{B}$ 10	0	1	X	X

$$T_C = AX + BC + \overline{A}\overline{C}\overline{X}$$

K-map for Z

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	1	0	0
$\bar{A}B$ 01	0	1	1	1
AB 11	X	X	X	X
$A\bar{B}$ 10	0	1	X	X

$$Z = \bar{C}X + BC$$

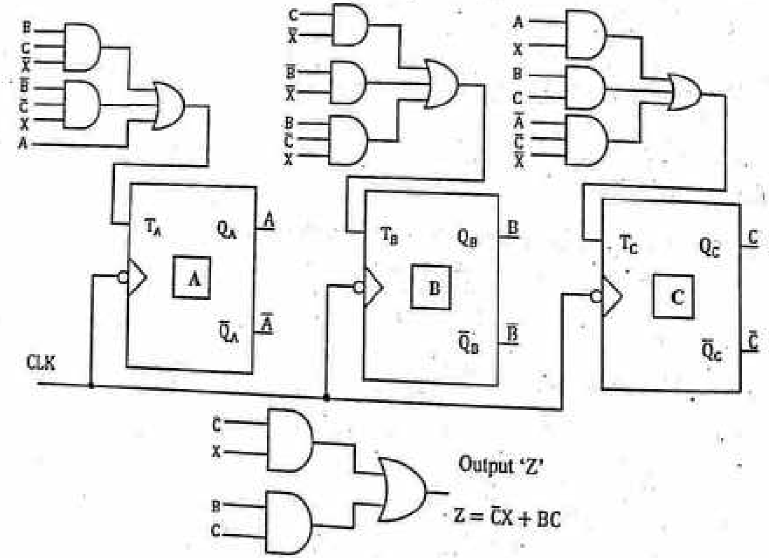


Figure 3.81 Logic Diagram

Example 3.17: Design a synchronous counter with states 0, 1, 2, 3, 0, 1, ... Using JK Flip-flop.

Solution:

Step 1: Draw the state diagram

The state diagram for the given problem can be drawn as shown in figure 3.82. Let 0, 1, 2, 3 be represented by states a, b, c and d respectively. Let $a = 0_{10} = 00_2$; $b = 1_{10} = 01_2$; $c = 2_{10} = 10_2$ and $d = 3_{10} = 11_2$

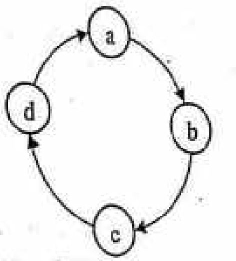


Figure 3.82 State diagram

Step 2: Plot the state table

Present state	Next state
a	b
b	c
c	d
d	a

Table 3.103 State table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Use the JK Flip-flop excitation table to find the value of J_A, K_A, J_B and K_B

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.104 Excitation table of JK flip-flop

Present state		Next state		Flip-flop inputs			
A	B	A^+	B^+	J_A	K_A	J_B	K_B
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

Table 3.105 Transition table

If there is no output and input in the state diagram, then the state table and transition table does not contain input and output terms.

Step 4: Derive the Flip-flop input equations using K-map.

K-map for J_A

A \ B	$\overline{B}$ 0	B 1
$\overline{A}$ 0	0_{0}	1_{1}
A 1	X_{2}	X_{3}

$J_A = B$

K-map for K_A

A \ B	$\overline{B}$ 0	B 1
$\overline{A}$ 0	X_{0}	X_{1}
A 1	0_{2}	1_{3}

$K_A = B$

K-map for J_B

A \ B	$\overline{B}$ 0	B 1
$\overline{A}$ 0	1_{0}	X_{1}
A 1	1_{2}	X_{3}

$J_B = 1$

K-map for K_B

A \ B	$\overline{B}$ 0	B 1
$\overline{A}$ 0	X_{0}	1_{1}
A 1	X_{2}	1_{3}

$K_B = 1$

Step 5: Draw the logic diagram

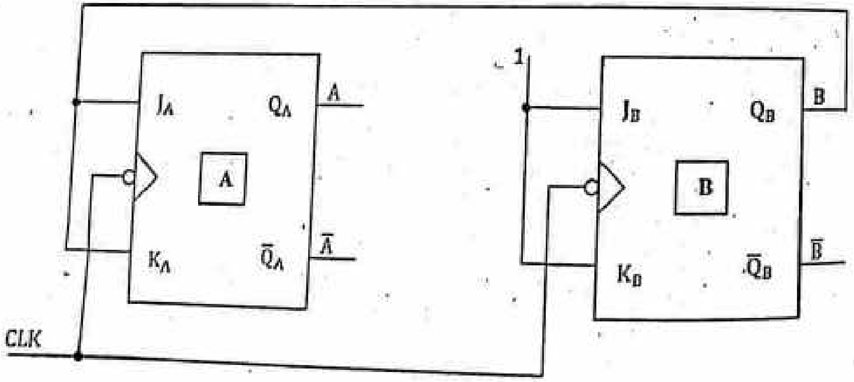


Figure 3.83 Logic diagram

Example 3.18: Using SR Flip-flop, design a parallel counter which counts the sequence 000, 111, 101, 110, 001, 010, 000.........

Solution:

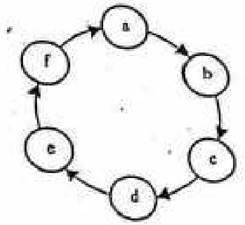


Figure 3.84 State diagram

Step 1: Draw the state diagram

Let a = 000; b = 111 ; c = 101 ; d = 110 ; e = 001 and f = 010

Step 2: Plot the state table

Present state	Next state
a	b
b	c
c	d
d	e
e	f
f	a

Table 3.106 State table

Step 3: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

If there is no output and input in the state diagram, then the state table and transition table does not contain input and output terms.

Use SR Flip-flop excitation table to find the value of S_A, R_A, S_B, R_B, S_C and R_C.

A	A^+	S_A	R_A
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Table 3.107 Excitation table of SR Flip-flop

Present state			Next state			Flip-flop inputs					
A	B	C	A^+	B^+	C^+	S_A	R_A	S_B	R_B	S_C	R_C
0	0	0	1	1	1	1	0	1	0	1	0
1	1	1	1	0	1	X	0	0	1	X	0
1	0	1	1	1	0	X	0	1	0	0	1
1	1	0	0	0	1	0	1	0	1	1	0
0	0	1	0	1	0	0	X	1	0	0	1
0	1	0	0	0	0	0	X	0	1	0	X

Table 3.108 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

Assume don't cares for unused states.

K-map for S_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1 0	0 1	X 3	0 2
A 1	X 4	X 5	X 7	0 6

$S_A = \overline{B}\overline{C}$

K-map for R_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 0	X 1	X 3	X 2
A 1	X 4	0 5	0 7	1 6

$R_A = B\overline{C}$

K-map for S_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1 0	1 1	X 3	0 2
A 1	X 4	1 5	0 7	0 6

$S_B = \overline{B}$

K-map for R_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 0	0 1	X 3	1 2
A 1	X 4	0 5	1 7	1 6

$R_B = B$

K-map for S_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1 0	0 1	X 3	0 2
A 1	X 4	0 5	X 7	1 6

$S_C = \overline{B}\overline{C} + AB$

K-map for R_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 0	1 1	X 3	X 2
A 1	X 4	1 5	0 7	0 6

$R_C = \overline{B}C$

Step 5: Draw the logic diagram

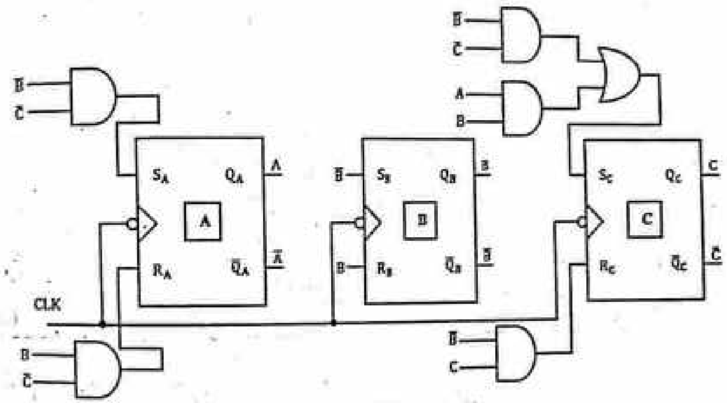


Figure 3.85 Logic diagram

Example 3.19: *Assume that there is a parking area in a shop whose capacity is 10. No more than 10 cars are allowed inside the parking area and the gate is closed as soon as the capacity is reached. There is a gate sensor to detect the entry of car which is to be synchronized with the clock pulse. Design and implement a suitable counter using JK flip flops. Also determine the number of flip flops to be used if the capacity is increased to 50.*

Step1: Draw the state diagram.

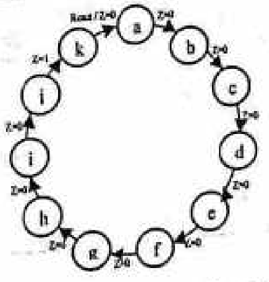


Figure 3.86 State diagram

Initially the counter is cleared, therefore $a = 0000$. The counter value gets incremented by '1' for each car entry. Since the capacity is ten, the counter counts upto '1010'. Therefore assign $k = 1010$. Inorder to close the gate make z=1.

Here, $a = 0000, b = 0001, c = 0010, d = 0011, e = 0100, f = 0101,$ $g = 0110, h = 0111, i = 1000, j = 1001, k = 1010$

Step 2: Plot the state table.

Present state	Next state	Output Z
a	b	0
b	c	0
c	d	0
d	e	0
e	f	0
f	g	0
g	h	0
h	i	0
i	j	0
j	k	1
k	a	0

Table 3.109 State table

Step 3: Assign binary values to the states and plot the transition table.

Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C, K_C, J_D and K_D

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.110 Excitation table of JK Flip-flop

Present state				Next state				Flip-flop inputs								Output
A	B	C	D	A^+	B^+	C^+	D^+	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D	Z
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X	0
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1	0
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X	0
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1	0
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X	0
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1	0
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X	0
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1	0
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X	0
1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1	1
1	0	1	0	0	0	0	0	X	1	0	X	X	1	0	X	0

Table 3.111 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

K-map for J_A

AB \ CD	$\overline{C}\overline{D}$ 00	$\overline{C}D$ 01	CD 11	$C\overline{D}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\overline{A}B$ 01	0_{4}	0_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$J_A = BCD$

K-map for K_A

AB \ CD	$\overline{C}\overline{D}$ 00	$\overline{C}D$ 01	CD 11	$C\overline{D}$ 10
$\overline{A}\overline{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\overline{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	0_{8}	0_{9}	X_{11}	1_{10}

$K_A = C$

K-map for J_B

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	1_{3}	0_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	0_{9}	X_{11}	0_{10}

$J_B = CD$

K-map for K_B

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	0_{4}	0_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$K_B = CD$

K-map for J_C

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	1_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	0_{4}	1_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$J_C = D$

K-map for K_C

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	1_{3}	0_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	1_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	1_{10}

$K_C = D + A$

K-map for J_D

AB \ CX	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1_{0}	X_{1}	X_{3}	1_{2}
$\bar{A}B$ 01	1_{4}	X_{5}	X_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	1_{8}	X_{9}	X_{11}	0_{10}

$J_D = \bar{C} + \bar{A}$

K-map for K_D

AB \ CX	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	1_{1}	1_{3}	X_{2}
$\bar{A}B$ 01	X_{4}	1_{5}	1_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	1_{9}	X_{11}	X_{10}

$K_D = 1$

K-map for Z

AB \ CX	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	0_{4}	0_{5}	0_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	1_{9}	X_{11}	0_{10}

$Z = AD$

Step 5: Draw the logic diagram.

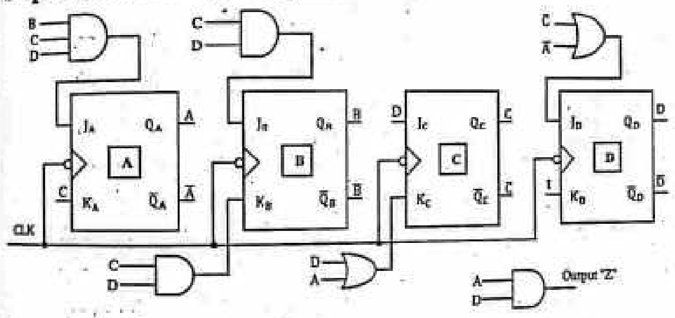


Figure 3.87: Logic diagram

The binary equivalent of 50 is 110010_2. Therefore 6 flip flops are required if the capacity is increased to 50.

***Example 3.20:* What is the aim of state reduction? Reduce the state diagram shown in figure 3.88**

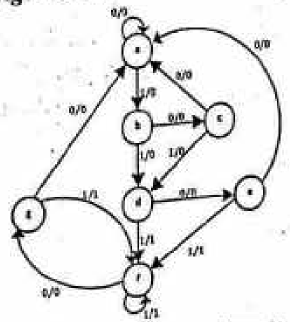


Figure 3.88 State diagram

Solution:

Aim of state reduction:

The state reduction technique avoids the introduction of redundant states. The reduction in redundant states reduces the number of Flip-flops and logic gates which reduces the cost and circuit complexity of the circuit.

Present state	Next state $x = 0$	Next state $x = 1$	Output $x = 0$	Output $x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Equivalent states

Table 3.112 State reduction

Here the states 'e' and 'g' are equivalent having same next states and outputs. So remove the state 'g' and replace 'g' by 'e'.

Present state	Next state $x = 0$	Next state $x = 1$	Output $x = 0$	Output $x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Equivalent states

Table 3.113 State table with state 'g' reduced

Also the states 'd' and 'f' are equivalent. So remove the state 'f' and also replace 'f' by 'd'. So the above state table shown in table 3.113 can be further reduced as shown in table 3.114.

Present state	Next state $x = 0$	Next state $x = 1$	Output $x = 0$	Output $x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Table 3.114 Reduced state table

From the reduced state table the state diagram can be drawn as shown in figure 3.89.

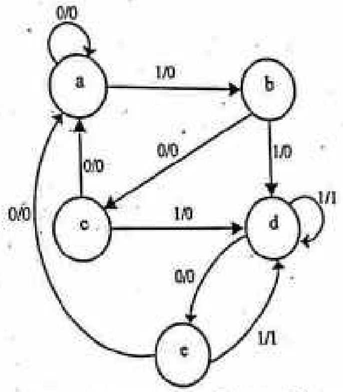
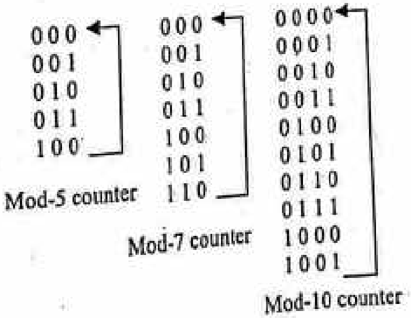


Figure 3.89 Reduced state diagram

3.10 MODULO-N-COUNTER

A Modulo-n-counter will count 'n' states. For example a mod-6 counter will count the sequence 000, 001, 010, 011, 100, 101 and then recycles to 000. Mod -6 counter skips 110 and 111 states and it goes through only six different states.

(e.g.)



Example 3.21: Design a MOD-5 counter using T Flip Flops.

Solution

Step1: Draw the state diagram.

Let a=000; b=001; c=010; d=011; e=100

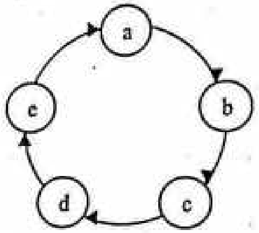


Figure 3.90 State diagram

Step 2: Plot the state table.

Present state	Next state
a	b
b	c
c	d
d	e
e	a

Table 3.115 State table

Step 3: Assign binary values to the states and plot the transition table by choosing T flip-flop excitation table to find the value of T_A, T_B and T_C

A	A^+	T_A
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.116 Excitation table of T Flip-flop

Present state			Next state			Flip-flop inputs		
A	B	C	A^+	B^+	C^+	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	0	0	0	1	0	0

Table 3.117 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

Assume don't cares for unused states.

K-map for T_A

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 $_{0}$	0 $_{1}$	1 $_{3}$	0 $_{2}$
A 1	1 $_{4}$	X $_{5}$	X $_{7}$	X $_{6}$

$T_A = BC + A$

K-map for T_B

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	0 $_{0}$	1 $_{1}$	1 $_{3}$	0 $_{2}$
A 1	0 $_{4}$	X $_{5}$	X $_{7}$	X $_{6}$

$T_B = C$

K-map for T_C

A \ BC	$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
$\overline{A}$ 0	1 $_{0}$	1 $_{1}$	1 $_{3}$	1 $_{2}$
A 1	0 $_{4}$	X $_{5}$	X $_{7}$	X $_{6}$

$T_C = \overline{A}$

Logic diagram

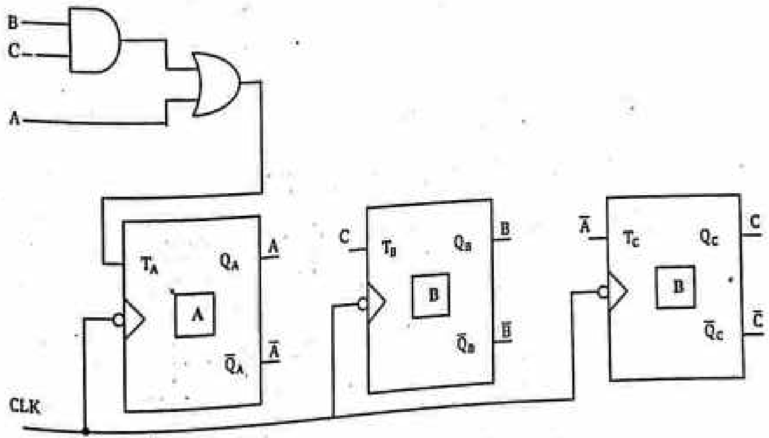


Figure 3.91 Logic diagram

3.11 REGISTERS

A Flip-flop can store 1-bit of data. So an n-bit register has a group of 'n' Flip-flops and is capable of storing n-bit of information. Group of flip-flops used to store a word is called register.

3.11.1 Shift Registers

The binary information in a register can be moved from one Flip-flop to another Flip-flop based on the activation of clock pulses. This give rise to a group of registers called shift registers.

Based on the mode of operation shift registers are classified into

i. Serial In-Serial Out shift registers

ii. Serial In-Parallel Out shift registers

iii. Parallel In-Serial Out shift registers

iv. Parallel In-Parallel Out shift registers

v. Bidirectional shift registers

vi. Universal shift registers

3.11.1.1 Serial In-Serial Out shift register (SISO shift register)

The Serial In-Serial Out shift registers can either shift the data's towards the left or towards the right. It has single input D_{in} and single output D_{out}.

Shift Left SISO register

Initially, the register is cleared, $Q_AQ_BQ_CQ_D = 0000$, therefore $D_{out} = 0$.

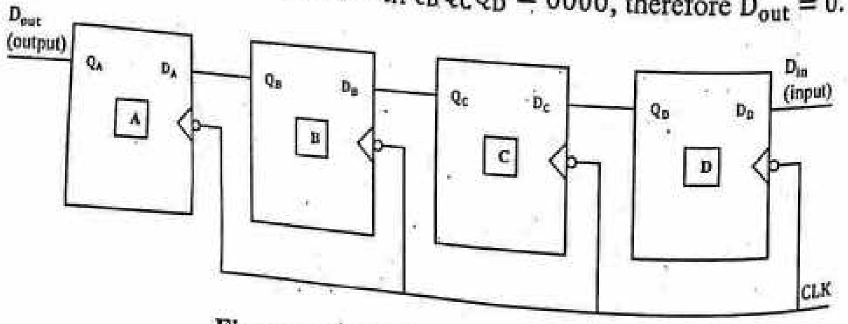


Figure 3.92 Shift left SISO register

Consider the data 1111 is applied serially to D_{in}

i. When the first negative clock edge hits, Q_D becomes '1' while Q_A, Q_B, Q_C remain '0'.

$Q_AQ_BQ_CQ_D = 0001$

$D_{out} = 0$

ii. When the second negative clock edge hits, Q_D and Q_C becomes '1', while Q_A, Q_B remain '0'.

$Q_AQ_BQ_CQ_D = 0011$

$D_{out} = 0$

iii. When the third negative clock edge hits, Q_D, Q_C, Q_B becomes '1', while Q_A remain '0'.

$Q_AQ_BQ_CQ_D = 0111$

$D_{out} = 0$

iv. When the fourth negative clock edge hits Q_D, Q_C, Q_B, Q_A becomes '1'.

$Q_AQ_BQ_CQ_D = 1111$

$D_{out} = 1$

Shift right SISO Register

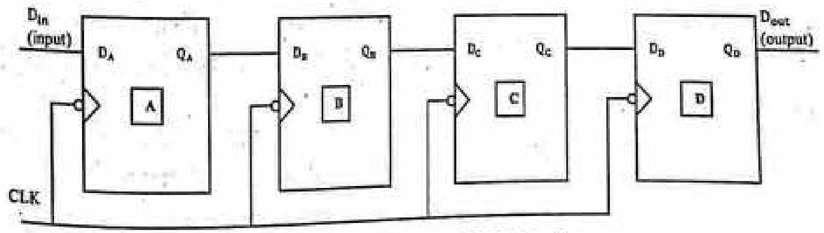


Figure 3.93 Shift right SISO Register

Initially, the register is cleared, $Q_AQ_BQ_CQ_D = 0000$, therefore $D_{out} = 0$.

Consider the data 1111 is applied serially to D_{in}

i. When the first negative clock edge hits, Q_A becomes '1' while Q_B, Q_C, Q_D remain '0'.

$Q_AQ_BQ_CQ_D = 1000$

$D_{out} = 0$

ii. When the second negative clock edge hits, Q_A and Q_B becomes '1', while Q_C, Q_D remain '0'.

$Q_AQ_BQ_CQ_D = 1100$

$D_{out} = 0$

iii. When the third negative clock edge hits, Q_A, Q_B Q_C becomes '1', while Q_D remain '0'.

$Q_AQ_BQ_CQ_D = 1110$

$D_{out} = 0$

iv. When the fourth negative clock edge hits, Q_A, Q_B, Q_C, Q_D becomes '1'.

$Q_AQ_BQ_CQ_D = 1111$

$D_{out} = 1$

Input D_{in}	CLK	Q_A	Q_B	Q_C	Q_D	Output D_{out}
1	1	1	0	0	0	0
1	2	1	1	0	0	0
1	3	1	1	1	0	0
1	4	1	1	1	1	1

Table 3.118 Truth table of Shift Right SISO register

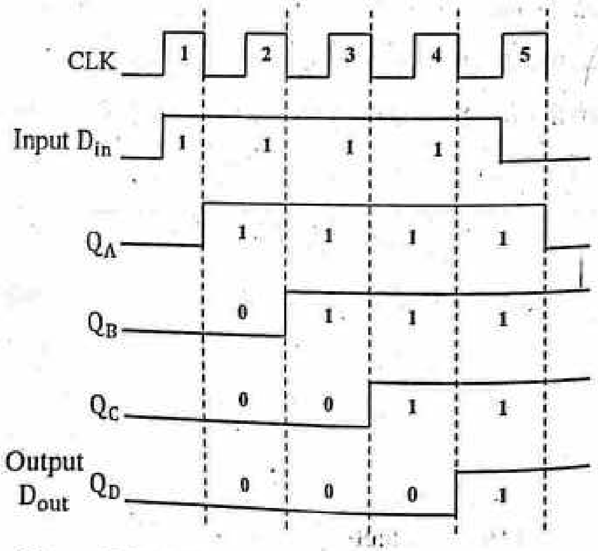


Figure 3.94: Waveform of SISO shift register

3.11.1.2 Serial In-Parallel Out shift register (SIPO shift register)

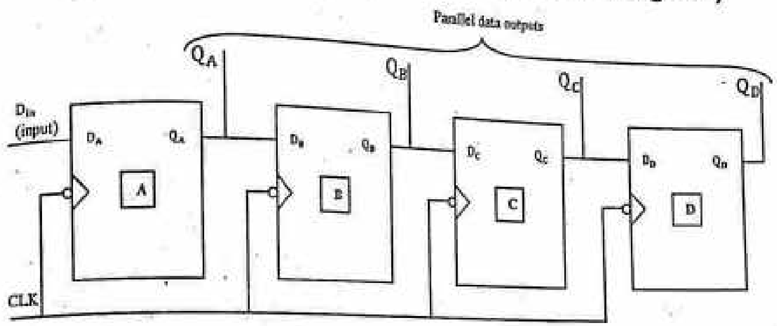


Figure 3.95 Serial In-Parallel out shift register

A 'n' bit serial in parallel out shift register contains a single input D_{in} , and 'n' number of outputs. Here, the input data is applied serially to D_{in}, and output is taken parallel at terminals Q_A, Q_B, Q_C and Q_D.

Initially, the register is cleared, $Q_AQ_BQ_CQ_D = 0000$. Consider the data 1111 is applied serially to D_{in}

i. When the first negative clock edge hits, $Q_AQ_BQ_CQ_D = 1000$

ii. When the second negative clock edge hits, $Q_AQ_BQ_CQ_D = 1100$

iii. When the third negative clock edge hits, $Q_AQ_BQ_CQ_D = 1110$

iv. When the fourth negative clock edge hits, $Q_AQ_BQ_CQ_D = 1111$

Input D_{in}	CLK	Output			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
1	2	1	1	0	0
1	3	1	1	1	0
1	4	1	1	1	1

Table 3.119 Truth table of Serial-In Parallel out shift register

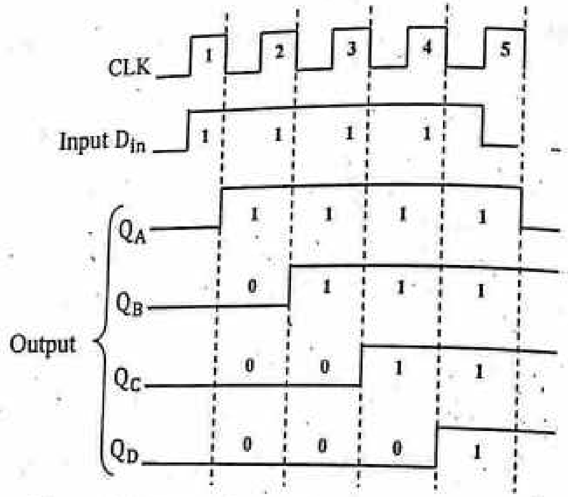


Figure 3.96 Waveform of SIPO Shift Register

3.11.1.3 Parallel In-Serial Out shift register (PISO shift register)

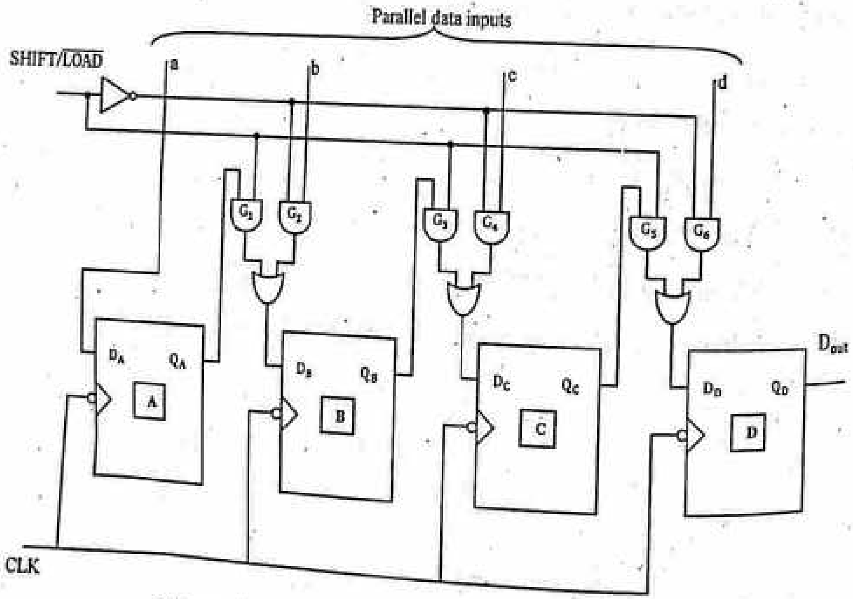


Figure 3.97 Parallel-In Serial Out shift register

A 4 bit parallel In-serial out shift register has 4 inputs a, b, c and d which are entered in parallel and one serial output D_{out}. The control signal used here is SHIFT/$\overline{\text{LOAD}}$ which is used to select shifting or loading data operation of the register.

If SHIFT/$\overline{\text{LOAD}}$ = 0, gates G_1, G_3, G_5 are disabled and gates G_2, G_4, G_6 are enabled which allows the data a, b, c, d to be at D_A, D_B, D_C, and D_D respectively.

When SHIFT/$\overline{\text{LOAD}}$ = 1, gates G_1, G_3, G_5 are enabled and gates G_2, G_4, G_6 are disabled. This allows the data bits to shift from one stage to the next as soon as the negative clock edge triggers.

SHIFT/$\overline{\text{LOAD}}$	Inputs				CLK	Output D_{out}
	a	b	c	d		
0	1	0	1	0	-	-
1	X	X	X	X	1	0
1	X	X	X	X	2	1
1	X	X	X	X	3	0
1	X	X	X	X	4	1

Table 3.120 Truth table of Parallel-In serial out shift register

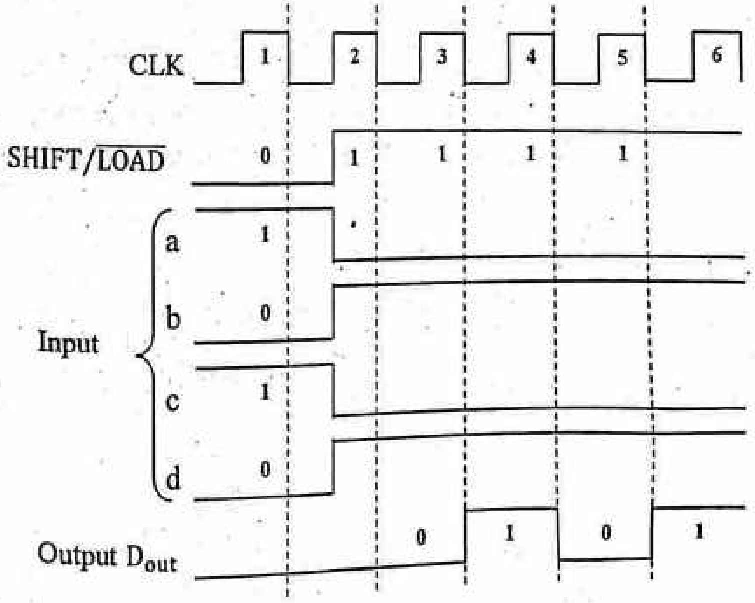


Figure 3.98 Waveform of PISO shift register

3.11.1.4 Parallel In-Parallel Out shift register (PIPO Shift register)

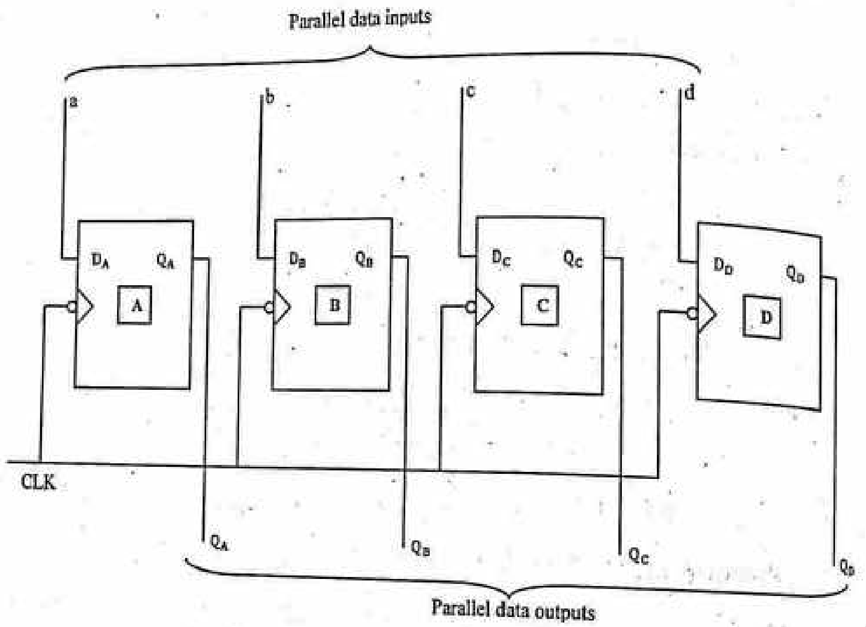


Figure 3.99 Parallel-In Parallel Out shift register

In parallel in parallel out shift register the inputs a, b, c and d are applied to D_A , D_B , D_C and D_D respectively. When the first negative clock edge hits $Q_AQ_BQ_CQ_D = abcd$

$$Q_A = a\ , Q_B = b\ , Q_C = c \text{ and } Q_D = d$$

Inputs				CLK	Outputs			
a	b	c	d		Q_A	Q_B	Q_C	Q_D
1	0	1	0	1	1	0	1	0
1	0	1	1	2	1	0	1	1
0	1	0	0	3	0	1	0	0
1	1	1	1	4	1	1	1	1

Table 3.121 Truth table of Parallel-In Parallel Out shift register

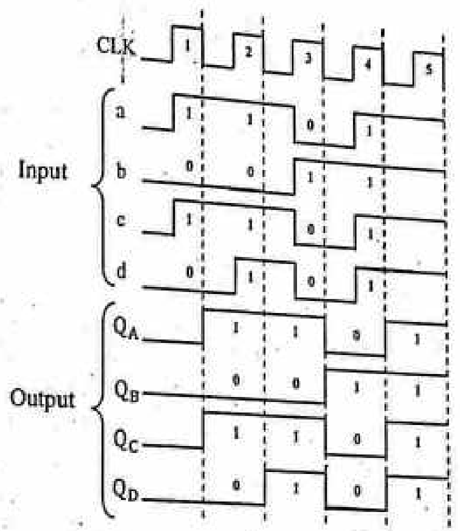


Figure 3.100 Waveform of PIPO shift register

3.11.1.5 Bidirectional shift register

Bidirectional shift register allows shifting of data either to the left or to the right side. Here, the control signal used is RIGHT/$\overline{\text{LEFT}}$ which allows the data shifting either towards right or towards left..

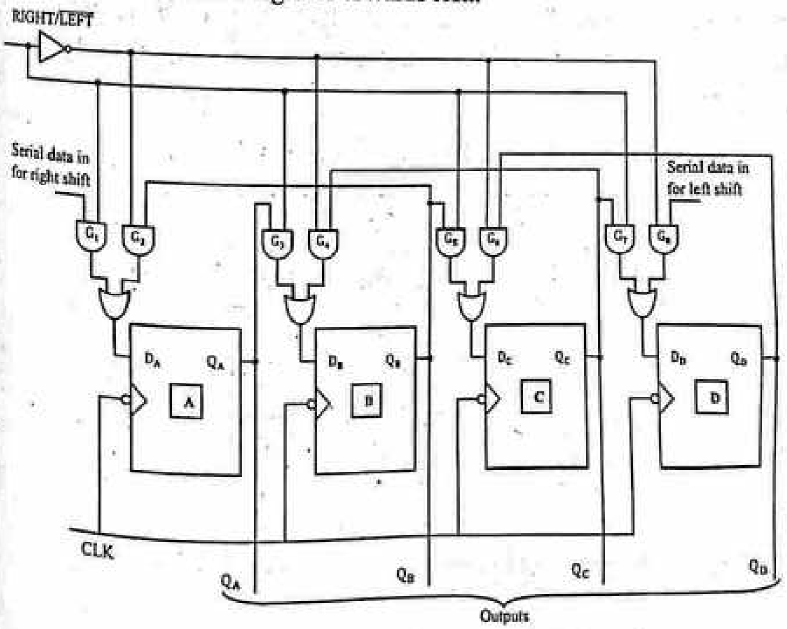


Figure 3.101 Bidirectional shift register

If RIGHT/$\overline{LEFT}$ = 1 gates G_1, G_3, G_5, G_7 are enabled and gates G_2, G_4, G_6, G_8 are disabled. When the negative clock edge gets triggered the data are shifted towards the right.

If RIGHT/$\overline{LEFT}$ = 0 gates G_1, G_3, G_5, G_7 are disabled and gates G_2, G_4, G_6, G_8 are enabled. When the negative clock edge gets triggered the data are shifted towards the left

3.11.1.6 Universal shift register

A shift register is referred to as universal shift register if it has both shifts (right shift and left shift) and parallel load capabilities. A 4 bit universal shift register consists of 4 Flip-flops and four 4 to1 multiplexers. The four multiplexers have two common selection lines $S_1$and S_0 and they select the mode of operation.

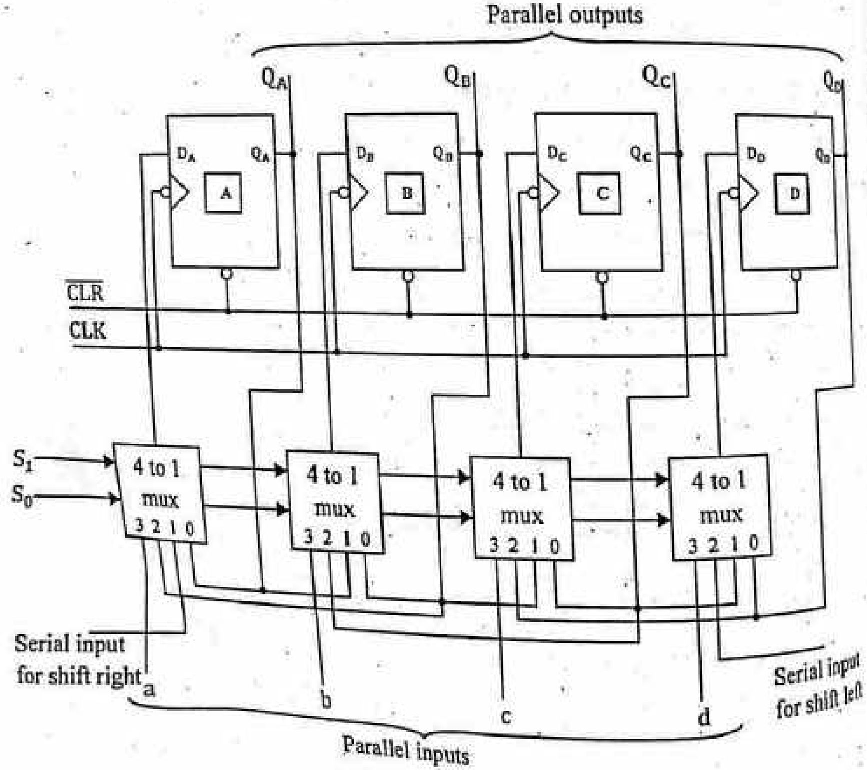


Figure 3.102 Universal shift register

When S_1S_0 = 00, the multiplexer select the input '0' and the present value of the register is applied to the D inputs of the Flip-flops. This results in no change in the register value.

When S_1S_0 = 01, the multiplexer select the input '1', the universal shift register operates as a shift right register.

When S_1S_0 = 10, the multiplexer select the input '2', the universal shift register operates as a shift left register.

When S_1S_0 = 11, the multiplexer select the input '3' the universal shift register loads the parallel inputs a, b, c and d simultaneously.

Mode selection		Operation
S_1	S_0	
0	0	No change
0	1	Right shift
1	0	Left shift
1	1	Parallel load

Table 3.122 Truth table of universal shift register

3.11.2 Shift register counters

Shift registers can be arranged to form several types of counters. All shift-register counters use feedback, whereby the output of the last flip-flop in the shift register is connected back to the first flip-flop input. Based on the type of feedback connection, the shift register counters are classified as

(i) Ring or standard ring counter

(ii) Twisted ring or Johnson or shift counter

3.11.2.1 Ring Counter (Standard ring counter)

In ring counter the output Q of each stage is connected to the D input of the next stage and the output of the last stage is fed back to the input of first stage. Figure 3.103 shows the logic diagram of a 5 bit ring counter. The $\overline{CLR}$ followed by $\overline{PRE}$ makes the output of first stage to '1' and remaining outputs to '0'. Thus Q_A = 1 and Q_B, Q_C, Q_D, Q_E are 0.

[Q_AQ_B Q_C Q_D Q_E=10000]

When the first clock pulse is triggered Q_B become '1' and remaining output are '0'.

$[Q_AQ_B\ Q_C\ Q_D\ Q_E=01000]$

When the second clock pulse is triggered Q_C become '1' and remaining outputs are '0'.

$[Q_AQ_B\ Q_C\ Q_D\ Q_E=00100]$.

When the third clock pulse is triggered Q_D become '1' and remaining outputs are '0'.

$[Q_AQ_B\ Q_C\ Q_D\ Q_E=00010]$.

When the fourth clock pulse is triggered Q_E become '1' and remaining outputs are '0'.

$[Q_AQ_B\ Q_C\ Q_D\ Q_E=00001]$.

When the fifth clock pulse is triggered Q_A become '1' and remaining outputs are '0'. $[Q_AQ_BQ_C\ Q_DQ_E=10000]$. The same process gets repeated for further clock pulse.

CLK	Q_A	Q_B	Q_C	Q_D	Q_E
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1

Table 3.123 Truth table of 5-bit ring counter

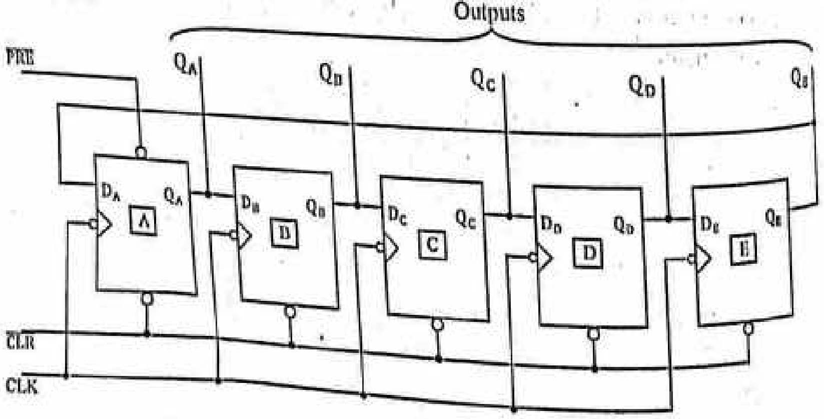


Figure 3.103 Logic diagram of 5-bit ring counter

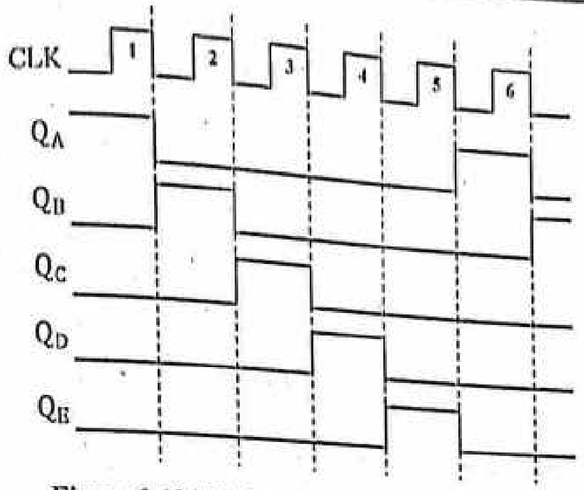


Figure 3.104 Timing diagram of 5 bit ring counter

3.11.2.2 Shift counter (Johnson or Twisted ring counter)

In a Johnson counter or twisted ring counter, the Q output of each stage of Flip-flop is connected to the D input of the next stage and the complement output ($\overline{Q}$) of the last stage is fed back to the input of first stage. In the circuit shown in figure 3.105, the last stage output $\overline{Q}_D$ is fed back to the first stage input D_A. Initially the register is cleared. Therefore $Q_AQ_BQ_C\ Q_D$=0000. Since $Q_D = 0, \overline{Q}_D = 1$. Therefore D_A=1. When the first clock pulse is triggered, Q_A becomes '1' and the remaining outputs are '0'. Hence $Q_AQ_BQ_C\ Q_D$=1000. When the second clock pulse is triggered, Q_A and Q_B becomes '1' and the remaining outputs are '0'. Hence $Q_AQ_BQ_C\ Q_D$=1100. The change of states is shown in table 3.124. After 8 states the same sequence gets repeated.

CLK	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 3.124 Truth table of 4-bit twisted ring counter or 4-bit Johnson counter

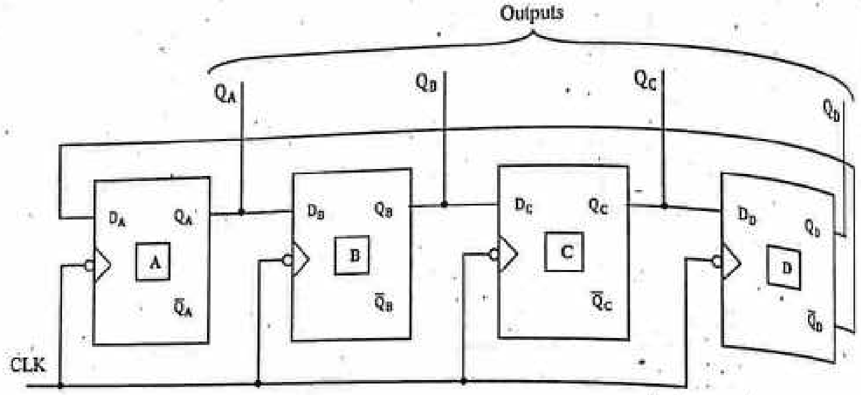


Figure 3.105 4-bit twisted ring or Johnson counter

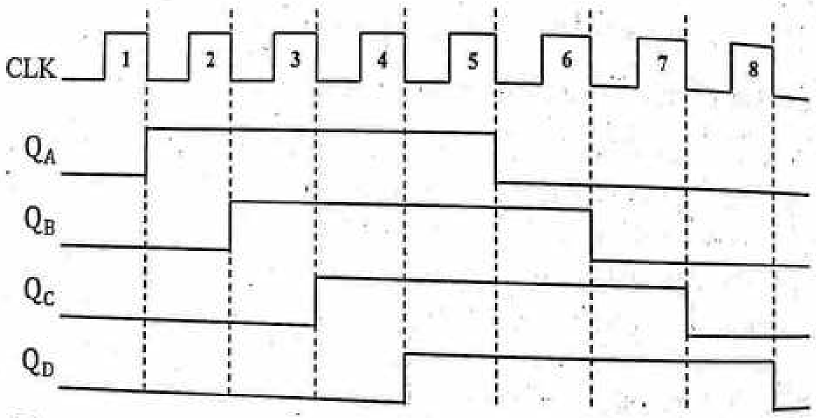


Figure 3.106 Timing diagram of 4-bit twisted ring or Johnson counter

Example 3.22: Design a 3 bit Johnson counter and explain its operation.

Solution:

In a Johnson counter, the Q output of each stage of Flip-flop is connected to the D input of the next stage and the complement output ($\overline{Q}$) of the last stage is fed back to the input of first stage. In the circuit shown in figure 3.107, the last stage output $\overline{Q}_C$ is fed back to the first stage input D_A. Initially the register is cleared. Therefore $Q_AQ_BQ_C$=000. Since $Q_C = 0, \overline{Q}_C = 1$. Therefore D_A=1. When the first clock pulse is triggered, Q_A becomes '1' and the remaining outputs are '0'. Hence $Q_AQ_BQ_C$ =100. When the second clock pulse is triggered, Q_A and Q_B becomes '1' and the remaining output $Q_C = 0$. Hence $Q_AQ_BQ_C$ =110. The change of states is shown in table 3.125. After 6 states the same sequence gets repeated.

CLK	Q_A	Q_B	Q_C
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1

Table 3.125 Truth table of 3-bit Johnson counter

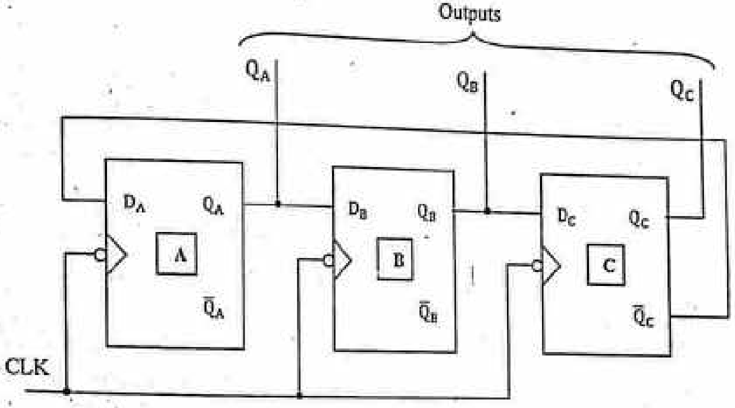


Figure 3.107 3-bit Johnson counter

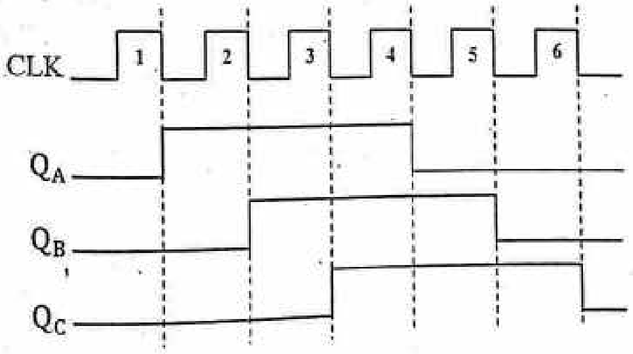


Figure 3.108 Timing diagram of 3-bit Johnson counter

3.12 DESIGN OF SEQUENCE DETECTOR

The sequence detector is a single input circuit that will accept a stream of bits and generate an output '1' whenever the particular sequence is detected.

The sequence detector is of two types.

i. Non-overlapping sequence detector.

ii. Overlapping sequence detector.

3.12.1 Non-overlapping sequence detector

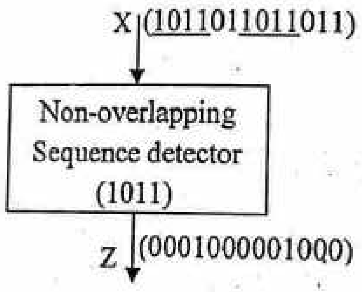


Figure 3.109

In the non-overlapping sequence detector shown in figure 3.109, whenever the sequence 1011 is detected it generates the output '1'. Then the circuit will go back to the initial state and wait for the next 1011 sequence.

Example 3.23: Design a sequence detector that produces an output '1' whenever the non-overlapping sequence 1011 is detected. Use D Flip-flop.

Solution:

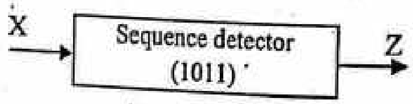


Figure 3.110

Step 1: Draw the state diagram

The state diagram of a sequence detector can be easily drawn by using the diagram shown in figure 3.111. Let 'a' be the initial state (a= 'null'), 'b' be the state for first bit '1' (b=1), 'c' be the state for first two bit '10' (c=10), 'd' be the state for first three bits '101' (d=101) and again 'a' be the state for all the four bits '1011' (a=1011). Then split each states into two, one for input X=0 and other for input X=1.

Then concatenate the state value and the input. Since the value of 'a' is null, the next state of 'a' is 0 for input X=0 and 1 for input X=1. The next state of b(1) is 10 and 11 for input X=0 and X=1 respectively. The next state of c(10) is 100 for input X=0 and 101 for input X=1. The next state of d(101) is 1010 for input X=0 and 1011 for input X=1.

Then assign the next states b=1, c=10, d=101 and a=1011. Also assign a=0, therefore the next state of 'a' is 'a' and 'b'. The next state of 'b' is 'c' and 11. Here discard the first bit of 11. The remaining bit is '1', which is already assigned b=1. Therefore the next state of 'b' is 'c' and 'b'. The next state of 'c' is 100 and 'd'. Here discard the first bit of 100. The remaining bit is 00. No state is assigned with 00. So discard the second bit also. So the remaining bit is 0. So assign a=0. Therefore the next state of 'c' is 'a' and 'd'.

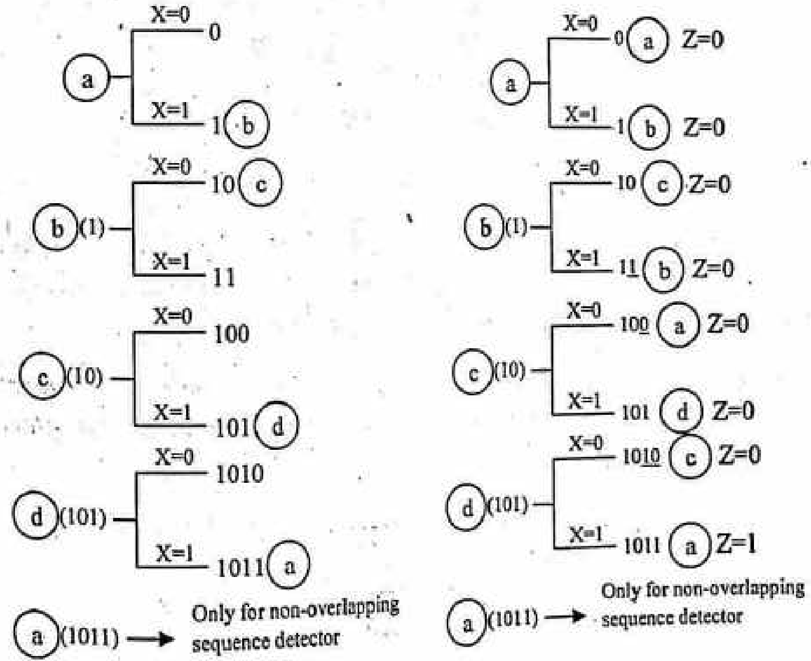


Figure 3.111

The next state of 'd' is 1010 and 'a'. Here discard the first bit of 1010. The remaining bits are 010. No state is assigned with 010. So discard the first two bits of 1010. So the remaining bits are 10. So assign c=10. Therefore the next states of 'd' is 'c' and 'a'. Also the sequence detector generates a output '1' only if it detects the sequence 1011. Therefore if the state is1011, then the output will be Z=1 else Z=0. By using the diagram shown in figure 3.111 the state diagram can be drawn as shown in figure 3.112.

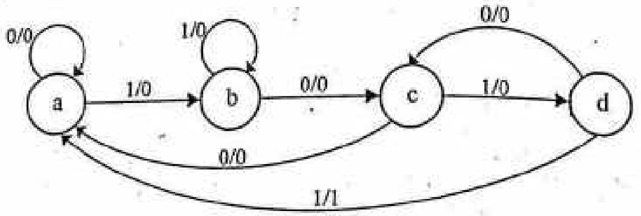


Figure 3.112 State diagram

Step 2: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	b	0	0
c	a	d	0	0
d	c	a	0	1

Table 3.126 State table

Step 3: Reduce the number of states if possible

Here state reduction is not possible

Step 4: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 4 states, assign a = 00, b = 01, c = 10, d = 11. Use the following D Flip-flop excitation table to find the value of D_A and D_B.

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

Table 3.127 Excitation table of D Flip-flop

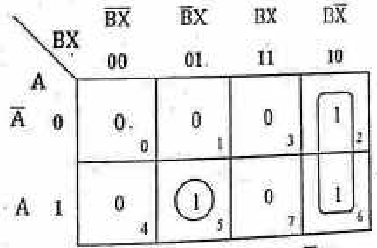
Present state		Input	Next state		Flip-flop inputs		Output
A	B	X	A^+	B^+	D_A	D_B	Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0
1	1	0	1	0	1	0	0
1	1	1	0	0	0	0	1

Table 3.128 Transition table

Step 5: Derive the Flip-flop input equations and output equations using K-map.

The Flip-flop input equations and output equations must be a function of present state A, B and input X.

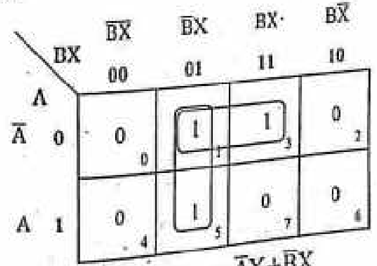
K-map for D_A



A \ BX	$\overline{BX}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0	0	0	1
A 1	0	1	0	1

$D_A = B\overline{X} + A\overline{B}X$

K-map for D_B



A \ BX	$\overline{BX}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0	1	1	0
A 1	0	1	0	0

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

K-map for Z

A \ BX	$\bar{B}\bar{X}$ 00	$\bar{B}X$ 01	BX 11	$B\bar{X}$ 10
$\bar{A}$ 0	0	0	0	0
A 1	0	0	1	0

$Z = ABX$

Step 6: Draw the logic diagram

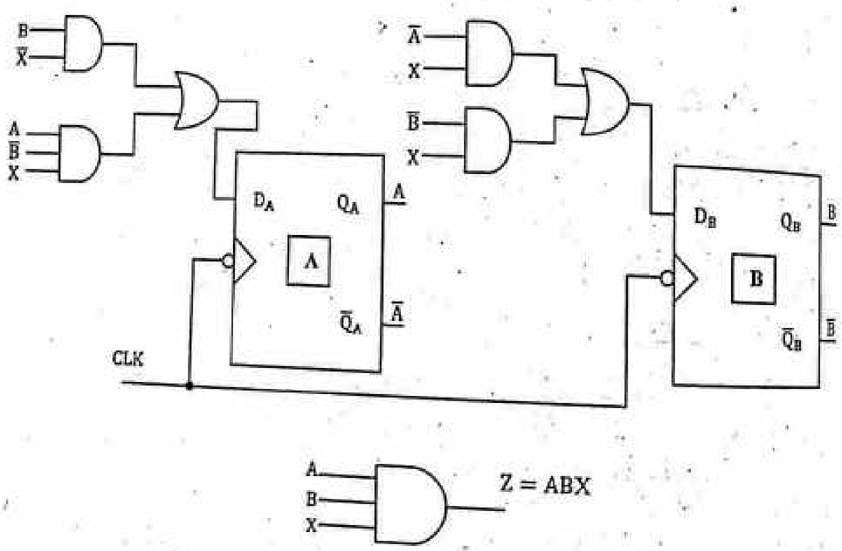


Figure 3.113 Logic diagram

Example 3.24: ***Design a sequence detector that produces an output '1' whenever the non-overlapping sequence 101101 is detected. Use SR Flip-flop.***

Solution:

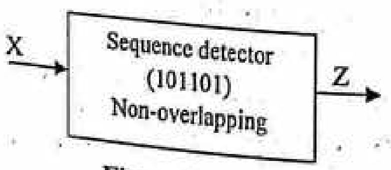


Figure 3.114

Step 1: Draw the state diagram

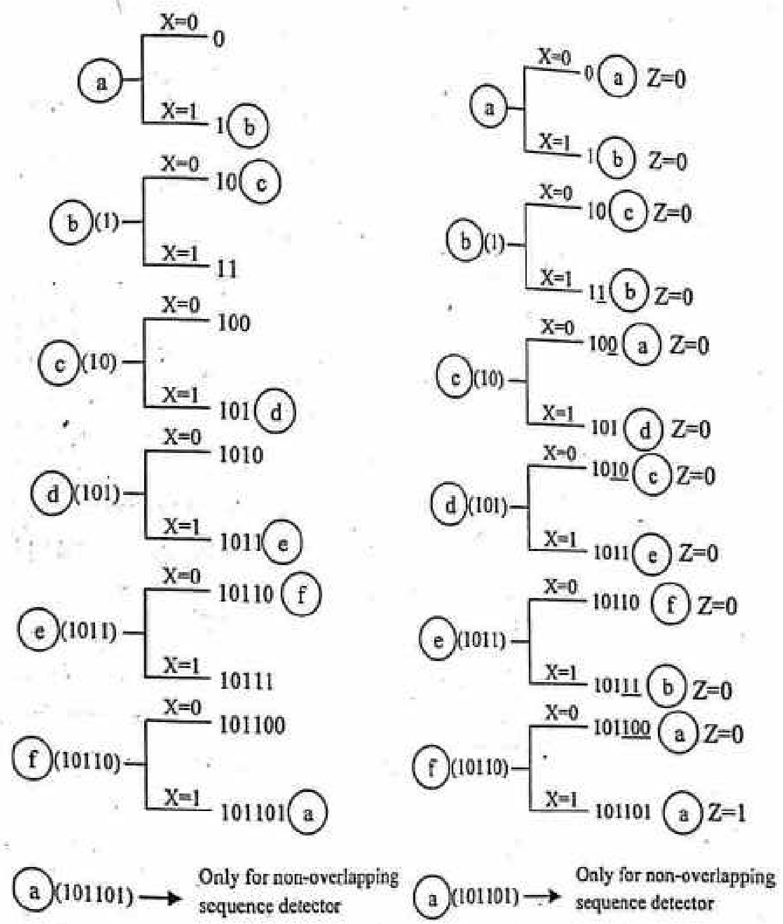


Figure 3.115

The state diagram of a sequence detector can be drawn by using the diagram shown in figure 3.115. Let 'a' be the initial state (a= 'null'), 'b' be the state for first bit '1' (b=1), 'c' be the state for first two bit '10' (c=10), 'd' be the state for first three bits '101' (d=101), 'e' be the state for first four bits '1011' (e=1011), 'f' be the state for first five bits '10110' (f=10110) and again 'a' be the state for all the six bits '101101' (a=101101). Then split each states into two, one for input X=0 and other for input X=1.

Then concatenate the state value and the input. Since the value of 'a' is null, the next state of 'a' is 0 for input X=0 and 1 for input X=1. The next state of b(1) is 10 and 11 for input X=0 and X=1 respectively. The next state of c(10) is 100 for input X=0 and 101 for input X=1. The next state of d(101) is 1010 for input X=0 and 1011 for input X=1. The next state of e(1011) is 10110 for input X=0 and 10111 for input X=1. The next state of f(10110) is 101100 for input X=0 and 101101 for input X=1. Then assign the next states b=1, c=10, d=101, e=1011, f=10110 and a=101101. Also assign a=0, therefore the next state of 'a' is 'a' and 'b'. The next state of 'b' is 'c' and 11. Here discard the first bit of 11. The remaining bit is '1', which is already assigned b=1. Therefore the next state of 'b' is 'c' and 'b'.

The next state of 'c' is 100 and 'd'. Here discard the first bit of 100. The remaining bit is 00. No state is assigned with 00. So discard the second bit also. So the remaining bit is 0. So assign a=0. Therefore the next state of 'c' are 'a' and 'd'. The next state of 'd' is 1010 and 'e'. Here discard the first bit of 1010. The remaining bits are 010. No state is assigned with 010. So discard the first two bits of 1010. So the remaining bits are 10. So assign c=10. Therefore the next states of 'd' are 'c' and 'e'.

The next state of 'e' are 'f' and 10111. Here discard the first bit of 10111. The remaining bits are 0111. No state is assigned with 0111. So discard the first two bits of 10111. So the remaining bits are 111. No state is assigned with 111. So discard the first three bits of 10111. The remaining bits are 11. We have already assigned 11 as state 'b', so assign 'b' for 10111. Therefore the next states of 'e' are 'f' and 'b'.

The next state of 'f' are 101100 and 'a'. Here discard the first bit of 101100. The remaining bits are 01100. No state is assigned with 01100. So discard the first two bits of 101100. So the remaining bits are 1100. No state is assigned with 1100. So discard the first three bits of 101100. The remaining bits are 100. We have already assigned 100 as state 'a', So assign 'a' for 101100. Therefore the next states of 'f' is 'a' for input X=0 and X=1. Also the sequence detector generates an output *'1' only if it* detects the sequence 101101. Therefore if the state is101101, then the *output will be* Z=1 else Z=0. By using the diagram shown in figure 3.115 the state *diagram can be drawn* as shown in figure 3.116.

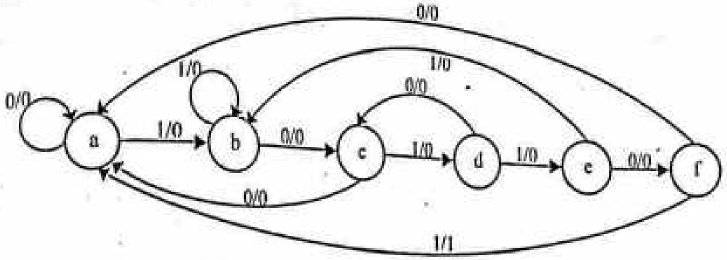


Figure 3.116 State diagram

Step 2: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	b	0	0
c	a	d	0	0
d	c	e	0	0
e	f	b	0	0
f	a	a	0	1

Table 3.129 State table

Step 3: Reduce the number of states if possible

Here state reduction is not possible

Step 4: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 6 states, assign $a = 000$, $b = 001$, $c = 010$, $d = 011$, $e = 100$ and $f = 101$. Use the following SR Flip-flop excitation table to find the S,R values .

A	A^+	S_A	R_A
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

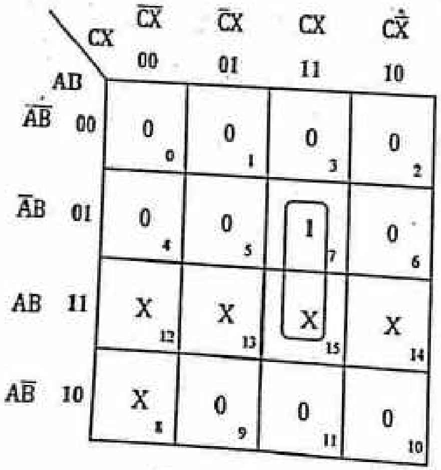
Table 3.130 Excitation table of SR Flip-flop

Present state			Input	Next state			Flip-flop inputs						Output
A	B	C	X	A^+	B^+	C^+	S_A	R_A	S_B	R_B	S_C	R_C	Z
0	0	0	0	0	0	0	0	X	0	X	0	X	0
0	0	0	1	0	0	1	0	X	0	X	1	0	0
0	0	1	0	0	1	0	0	X	1	0	0	1	0
0	0	1	1	0	0	1	0	X	0	X	X	0	0
0	1	0	0	0	0	0	0	X	0	1	0	X	0
0	1	0	1	0	1	1	0	X	X	0	1	0	0
0	1	1	0	0	1	0	0	X	X	0	0	1	0
0	1	1	1	1	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	1	X	0	0	X	1	0	0
1	0	0	1	0	0	1	0	1	0	X	1	0	0
1	0	1	0	0	0	0	0	1	0	X	0	1	0
1	0	1	1	0	0	0	0	1	0	X	0	1	1

Table 3.131 Transition table

Step 5: Derive the Flip-flop input equations and output equation using K-map

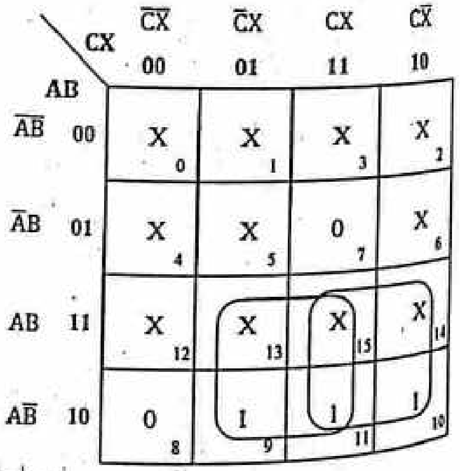
K-map for S_A



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	0	0	0
$\bar{A}B$ 01	0	0	1	0
AB 11	X	X	X	X
$A\bar{B}$ 10	X	0	0	0

$S_A = BCX$

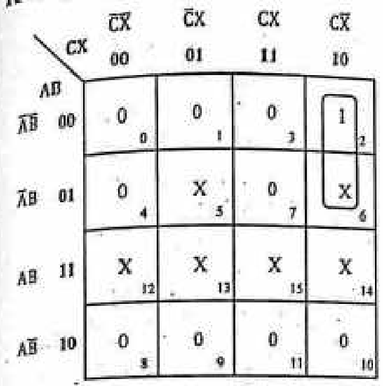
K-map for R_A



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X	X	X	X
$\bar{A}B$ 01	X	X	0	X
AB 11	X	X	X	X
$A\bar{B}$ 10	0	1	1	1

$R_A = AX + AC$

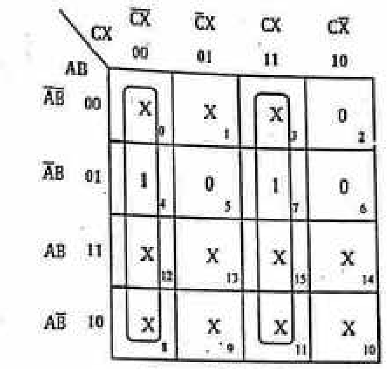
K-map for S_B



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	0	0	1
$\bar{A}B$ 01	0	X	0	X
AB 11	X	X	X	X
$A\bar{B}$ 10	0	0	0	0

$S_B = \bar{A}C\bar{X}$

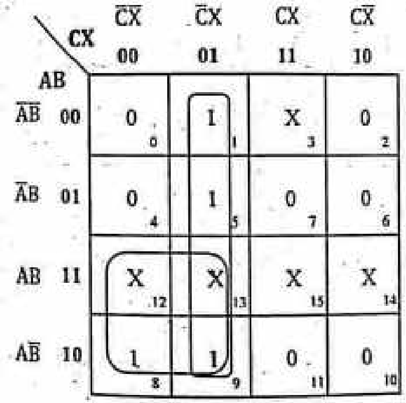
K-map for R_B



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X	X	X	0
$\bar{A}B$ 01	1	0	1	0
AB 11	X	X	X	X
$A\bar{B}$ 10	X	X	X	X

$R_B = \bar{C}\bar{X} + CX$

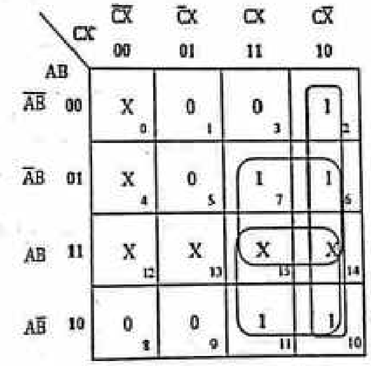
K-map for S_C



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	1	X	0
$\bar{A}B$ 01	0	1	0	0
AB 11	X	X	X	X
$A\bar{B}$ 10	1	1	0	0

$S_C = \bar{C}X + A\bar{C}$

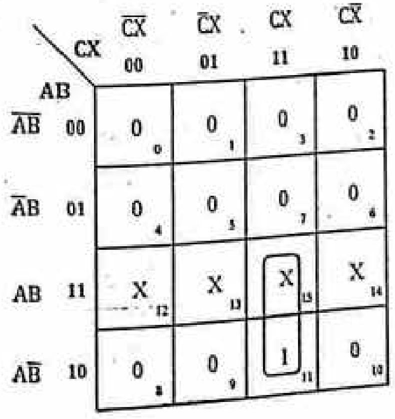
K-map for R_C



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X	0	0	1
$\bar{A}B$ 01	X	0	1	1
AB 11	X	X	X	X
$A\bar{B}$ 10	0	0	1	1

$R_C = C\bar{X} + BC + AC$

K-map for Z



AB CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0	0	0	0
$\bar{A}B$ 01	0	0	0	0
AB 11	X	X	X	X
$A\bar{B}$ 10	0	0	1	0

$Z = ACX$

Step 6: Draw the logic diagram

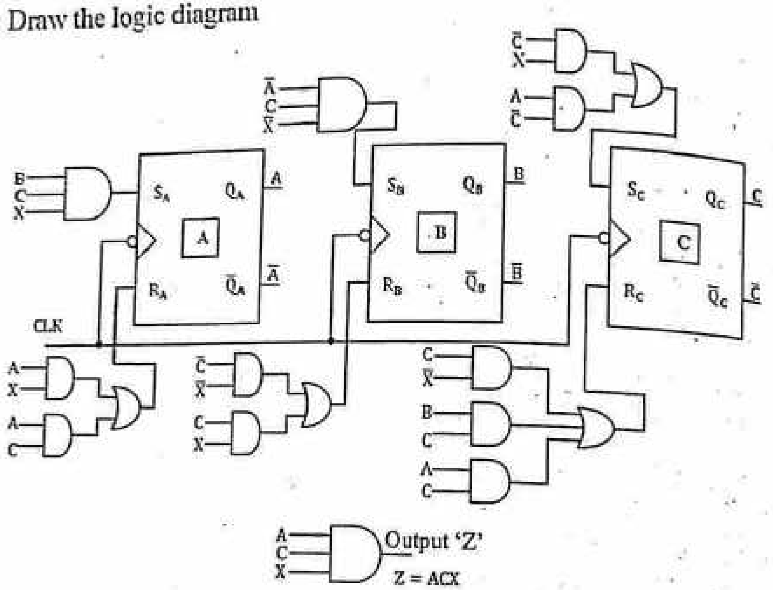


Figure 3.117 Logic diagram

3.12.2 Overlapping sequence detector

In overlapping sequence detector whenever the sequence 1011 is detected it generates the output '1', even though the sequence 1011 is overlapped.

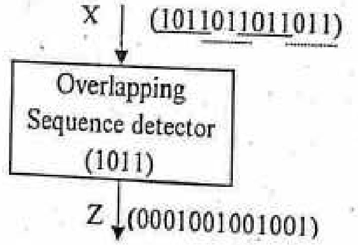


Figure 3.118

Example 3.25: Design a sequence detector that produces an output '1' whenever the sequence 101101 is detected. Use D Flip-flop.

Solution:

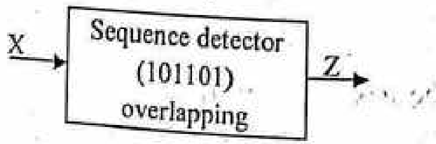


Figure 3.119

Step 1: Draw the state diagram

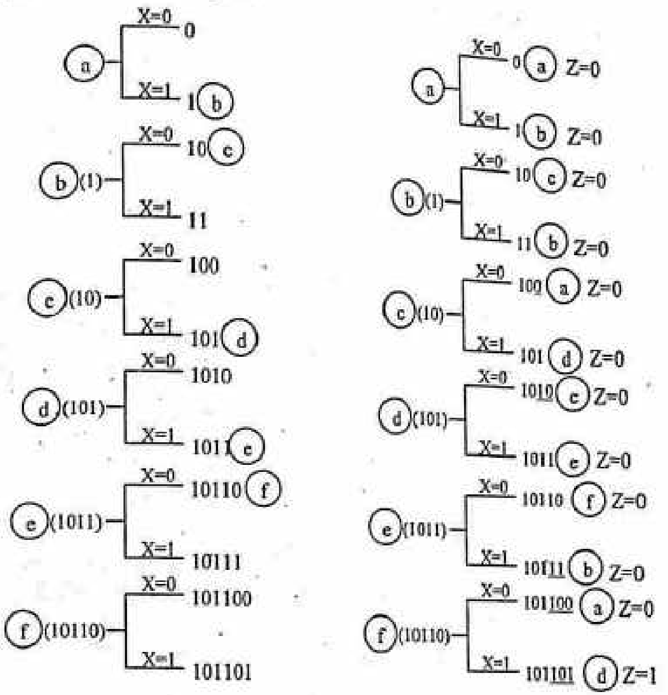


Figure 3.120

The state diagram of a sequence detector can be easily drawn by using the diagram shown in figure 3.120.

Let 'a' be the initial state (a= 'null'), 'b' be the state for first bit '1' (b=1), 'c' be the state for first two bit '10' (c=10), 'd' be the state for first three bits '101' (d=101), 'e' be the state for first four bits '1011' (e=1011) and 'f' be the state for first five bits '10110' (f=10110).

Then split each states into two, one for input X=0 and other for input X=1. Then concatenate the state value and the input. Since the value of 'a' is null, the next state of 'a' is 0 for input X=0 and 1 for input X=1. The next state of b(1) is 10 and 11 for input X=0 and X=1 respectively. The next state of c(10) is 100 for input X=0 and 101 for input X=1. The next state of d(101) is 1010 for input X=0 and 1011 for input X=1. The next state of e(1011) is 10110 for input X=0 and 10111 for input X=1. The next state of f(10110) is 101100 for input X=0 and 101101 for input X=1.

Then assign the next states b=1, c=10, d=101, e=1011 and f=10110. Also assign a=0, therefore the next state of 'a' is 'a' and 'b'. Follow the same procedure of non-overlapping sequence detector and find the next states of a, b, c, d, e and f. Here the next states of 'f' are 101100 and 101101. Here discard the first bit of 101100. The remaining bits are 01100. No state is assigned with 01100. So discard the first two bits of 101100. The remaining bits are 1100. No state is assigned with 1100. So discard the first three bits of 101100. The remaining bits are 100. We have already assigned 100 as 'a'. So assign 101100 as state 'a'.

Similarly find the state for 101101 by discarding the first three bits. The remaining bits are 101. We have already assigned 'd' as 101. So assign 101101 as state 'd'. Therefore the next states of 'f' are 'a' and 'd'.

Also the sequence detector generates an output '1' only if it detects the sequence 101101. Therefore if the state is101101, then the output will be Z=1 else Z=0. By using the diagram shown in figure 3.120 the state diagram can be drawn as shown in figure 3.121.

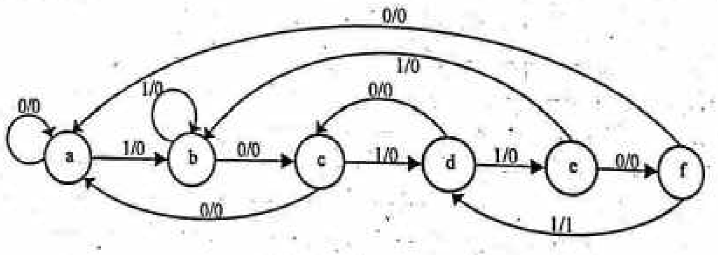


Figure 3.121 State diagram

Step 2: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	b	0	0
c	a	d	0	0
d	c	e	0	0
e	f	b	0	0
f	a	d	0	1

Table 3.132 State table

Step 3: Reduce the number of states if possible

Here state reduction is not possible

Step 4: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 6 states, assign $a = 000$, $b = 001$, $c = 010$, $d = 011$, $e = 100$ and $f = 101$. Use the following D Flip-flop excitation table to find the value of D_A, D_B and D_C.

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

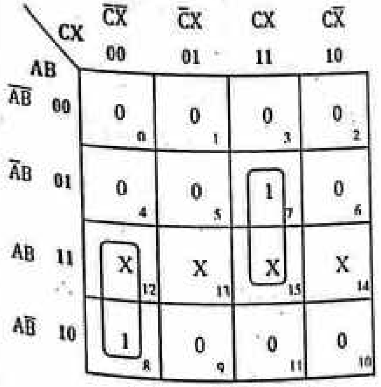
Table 3.133 Excitation table of D Flip-flop

Present state			Input	Next state			Flip-flop inputs			Output
A	B	C	X	A^+	B^+	C^+	D_A	D_B	D_C	Z
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	0	1	0	0
0	1	1	1	1	0	0	1	0	0	0
1	0	0	0	1	0	1	1	0	1	0
1	0	0	1	0	0	1	0	0	1	0
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	0	1	1	1

Table 3.134 Transition table

Step 5: Derive the Flip-flop input equations and output equation using K-map

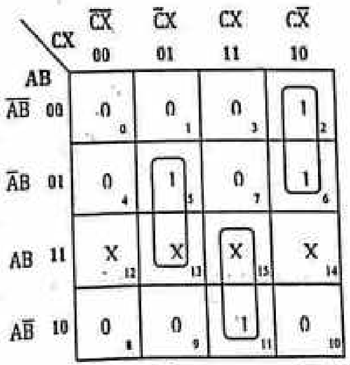
K-map for D_A



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0 (0)	0 (1)	0 (3)	0 (2)
$\bar{A}B$ 01	0 (4)	0 (5)	1 (7)	0 (6)
AB 11	X (12)	X (13)	X (15)	X (14)
$A\bar{B}$ 10	1 (8)	0 (9)	0 (11)	0 (10)

$D_A = A\bar{C}\bar{X} + BCX$

K-map for D_B



AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0 (0)	0 (1)	0 (3)	1 (2)
$\bar{A}B$ 01	0 (4)	1 (5)	0 (7)	1 (6)
AB 11	X (12)	X (13)	X (15)	X (14)
$A\bar{B}$ 10	0 (8)	0 (9)	1 (11)	0 (10)

$D_B = B\bar{C}X + ACX + \bar{A}C\bar{X}$

K-map for D_C

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0 $_{0}$	1 $_{1}$	1 $_{3}$	0 $_{2}$
$\bar{A}B$ 01	0 $_{4}$	1 $_{5}$	0 $_{7}$	0 $_{6}$
AB 11	X $_{12}$	X $_{13}$	X $_{15}$	X $_{14}$
$A\bar{B}$ 10	1 $_{8}$	1 $_{9}$	1 $_{11}$	0 $_{10}$

$$D_C = \bar{C}X + A\bar{C} + \bar{B}X$$

K-map for Z

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
$\bar{A}B$ 01	0 $_{4}$	0 $_{5}$	0 $_{7}$	0 $_{6}$
AB 11	X $_{12}$	X $_{13}$	X $_{15}$	X $_{14}$
$A\bar{B}$ 10	0 $_{8}$	0 $_{9}$	1 $_{11}$	0 $_{10}$

$$Z = ACX$$

Step 6: Draw the logic diagram

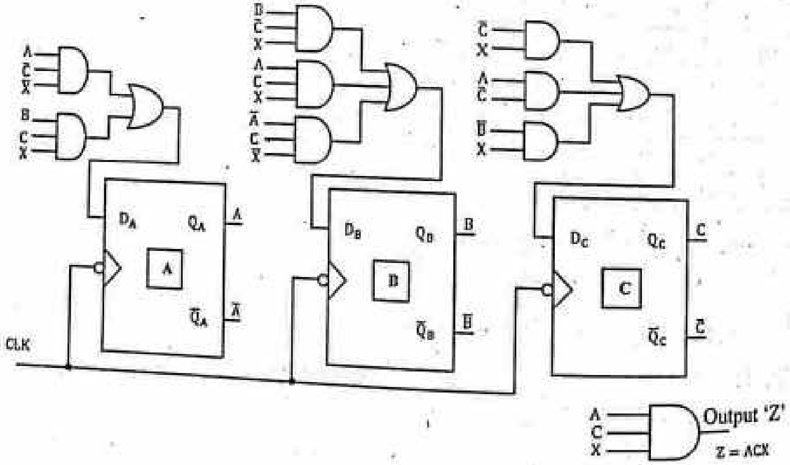


Figure 3.122 Logic diagram

Example 3.26: Design a sequence detector to detect the sequence 101011. Use JK Flip-flop.

Solution:

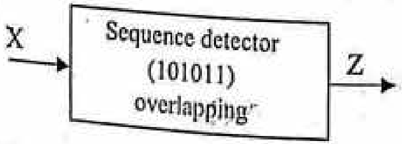
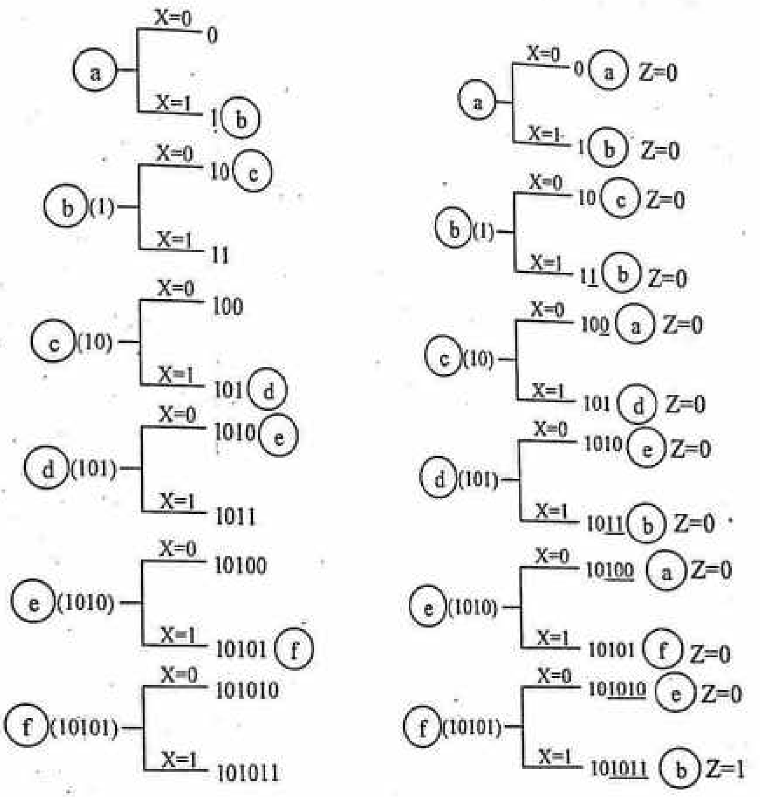


Figure 3.123

Step 1: Draw the state diagram



a X=0 0 X=1 1 b
b (1) X=0 10 c X=1 11
c (10) X=0 100 X=1 101 d
d (101) X=0 1010 e X=1 1011
e (1010) X=0 10100 X=1 10101 f
f (10101) X=0 101010 X=1 101011
a X=0 0 a Z=0 X=1 1 b Z=0
b (1) X=0 10 c Z=0 X=1 11 b Z=0
c (10) X=0 100 a Z=0 X=1 101 d Z=0
d (101) X=0 1010 e Z=0 X=1 1011 b Z=0
e (1010) X=0 10100 a Z=0 X=1 10101 f Z=0
f (10101) X=0 101010 e Z=0 X=1 101011 b Z=1

Figure 3.124

The state diagram of a sequence detector can be easily drawn by using the diagram shown in figure 3.124.

Let 'a' be the initial state (a= 'null'), 'b' be the state for first bit '1' (b=1), 'c' be the state for first two bit '10' (c=10), 'd' be the state for first three bits '101' (d=101), 'e' be the state for first four bits '1010' (e=1010) and 'f' be the state for first five bits '10101' (f=10101).

Then split each states into two, one for input X=0 and other for input X=1. Find the next states of a, b, c, d, e and f. Also the sequence detector generates an output '1' only if it detects the sequence 101011. Therefore if the state is 101011, then the output will be Z=1 else Z=0. By using the diagram shown in figure 3.124 the state diagram can be drawn as shown in figure 3.125.

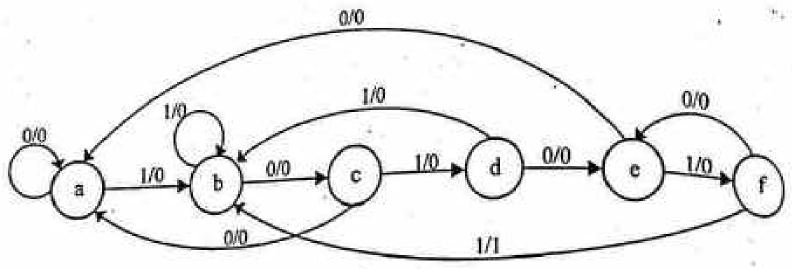


Figure 3.125 State diagram

Step 2: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	b	0	0
c	a	d	0	0
d	e	b	0	0
e	a	f	0	0
f	e	b	0	1

Table 3.135 State table

Step 3: Reduce the number of states if possible

Here state reduction is not possible

Step 4: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 6 states, assign $a = 000$, $b = 001$, $c = 010$, $d = 011$, $e = 100$ and $f = 101$. Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C and K_C

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

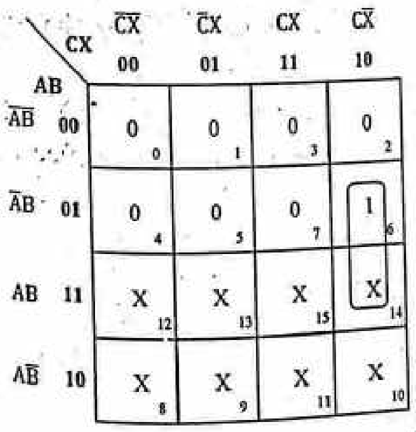
Table 3.136 Excitation table of JK Flip-flop

Present state			Input	Next state			Flip-flop inputs						Output
A	B	C	X	A^+	B^+	C^+	J_A	K_A	J_B	K_B	J_C	K_C	Z
0	0	0	0	0	0	0	0	X	0	X	0	X	0
0	0	0	1	0	0	1	0	X	0	X	1	X	0
0	0	1	0	0	1	0	0	X	1	X	X	1	0
0	0	1	1	0	0	1	0	X	0	X	X	0	0
0	1	0	0	0	0	0	0	X	X	1	0	X	0
0	1	0	1	0	1	1	0	X	X	0	1	X	0
0	1	1	0	1	0	0	1	X	X	1	X	1	0
0	1	1	1	0	0	1	0	X	X	1	X	0	0
1	0	0	0	0	0	0	X	1	0	X	0	X	0
1	0	0	1	1	0	1	X	0	0	X	1	X	0
1	0	1	0	1	0	0	X	0	0	X	X	1	0
1	0	1	1	0	0	1	X	1	0	X	X	0	1

Table 3.137 Transition table

Step 5: Derive the Flip-flop input equations and output equation using K-map

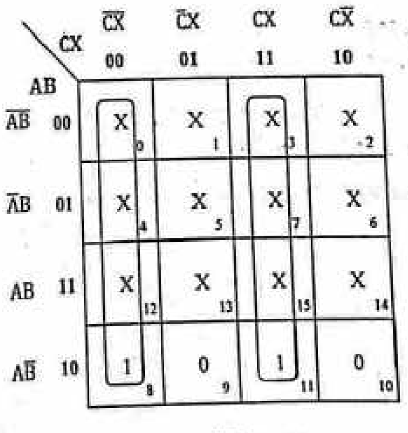
K-map for J_A



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\overline{A}B$ 01	0_{4}	0_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$J_A = BC\overline{X}$

K-map for K_A



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\overline{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\overline{B}$ 10	1_{8}	0_{9}	1_{11}	0_{10}

$K_A = \overline{C}\overline{X} + CX$

K-map for J_B

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	1_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	0_{9}	0_{11}	0_{10}

$$J_B = \bar{A}C\bar{X}$$

K-map for K_B

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	1_{4}	0_{5}	1_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$$K_B = \bar{X} + C$$

K-map for J_C

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	1_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	0_{4}	1_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	1_{9}	X_{11}	X_{10}

$$J_C = X$$

K-map for K_C

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	0_{3}	1_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	0_{7}	1_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	0_{11}	1_{10}

$$K_C = \bar{X}$$

K-map for Z

AB \ CX	$\bar{C}\bar{X}$ 00	$\bar{C}X$ 01	CX 11	$C\bar{X}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	0_{4}	0_{5}	0_{7}	0_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	0_{8}	0_{9}	1_{11}	0_{10}

$$Z = ACX$$

Step 6: Draw the logic diagram

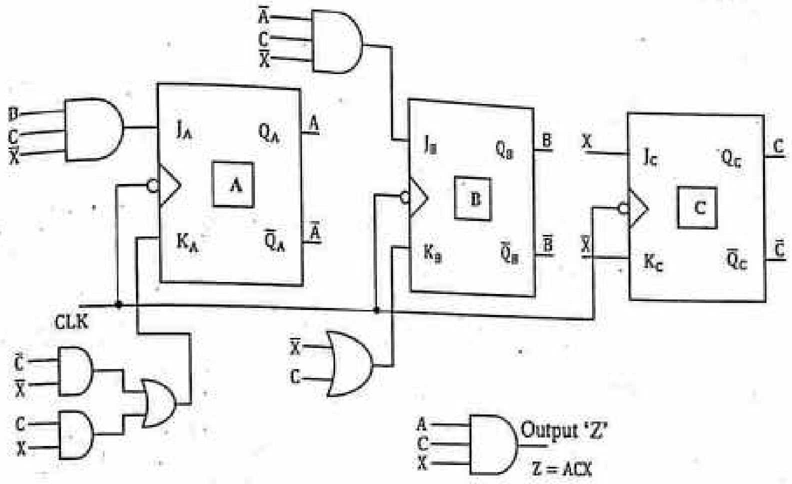


Figure 3.126 Logic diagram

Example 3.27: Design a sequence detector to detect a serial input sequence of 101.

Solution:

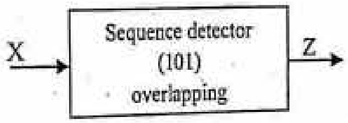


Figure 3.127

Step 1: Draw the state diagram

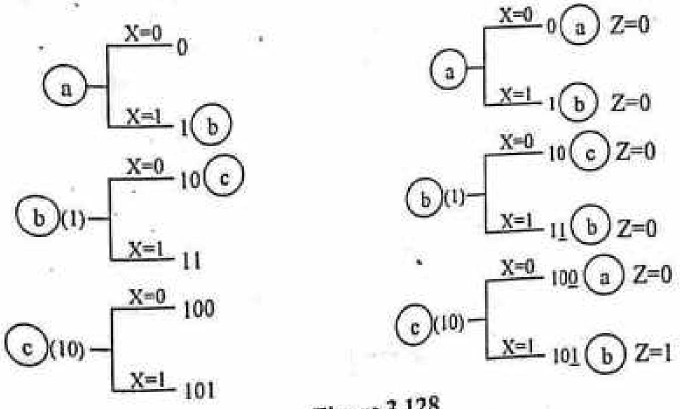


Figure 3.128

By using the diagram shown in figure 3.128 the state diagram can be drawn as shown in figure 3.129.

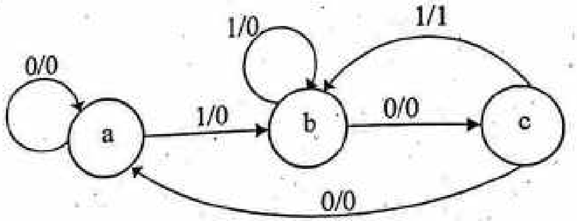


Figure 3.129 State diagram

Step 2: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	a	b	0	0
b	c	b	0	0
c	a	b	0	1

Table 3.138 State table

Step 3: Reduce the number of states if possible

Here state reduction is not possible

Step 4: Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.

Since there are 3 states, assign $a = 00, b = 01, c = 11$. Use the following D Flip-flop excitation table to find the value of D_A, D_B and D_C.

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

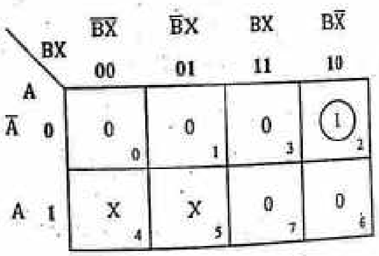
Table 3.139 Excitation table of D Flip-flop

Present state		Input	Next state		Flip-flop inputs		Output
A	B	X	A^+	B^+	D_A	D_B	Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Table 3.140 Transition table

Step 5: Derive the Flip-flop input equations and output equation using K-map

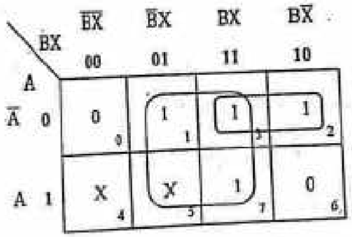
K-map for D_A



A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0	0	0	1
A 1	X	X	0	0

$D_A = \overline{A}B\overline{X}$

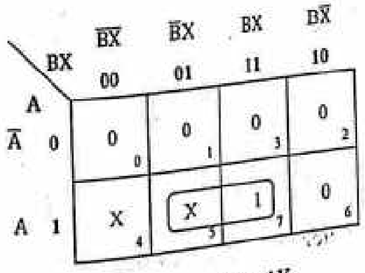
K-map for D_B



A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0	1	1	1
A 1	X	X	1	0

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

K-map for 'Z'



A \ BX	$\overline{B}\overline{X}$ 00	$\overline{B}X$ 01	BX 11	$B\overline{X}$ 10
$\overline{A}$ 0	0	0	0	0
A 1	X	X	1	0

$Z = AX$

Step 6: Draw the logic diagram

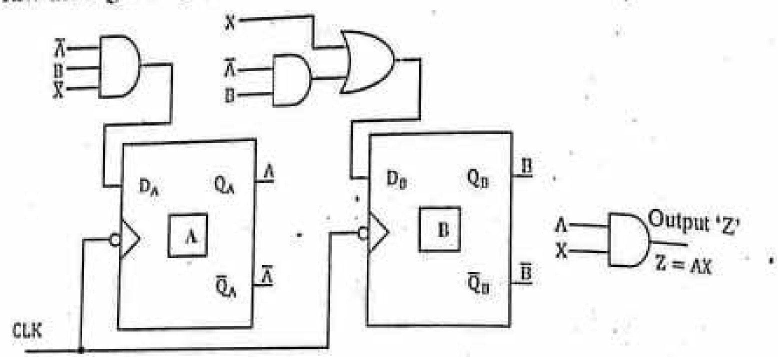


Figure 3.130 Logic diagram

Example 3.28: Design a sequence detector which detect the sequence "01110" using D flip flops(one bit overlapping).

Solution:

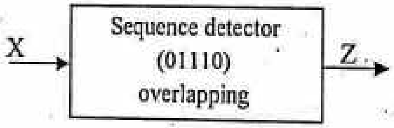


Figure 3.131

Step 1: Draw the state diagram

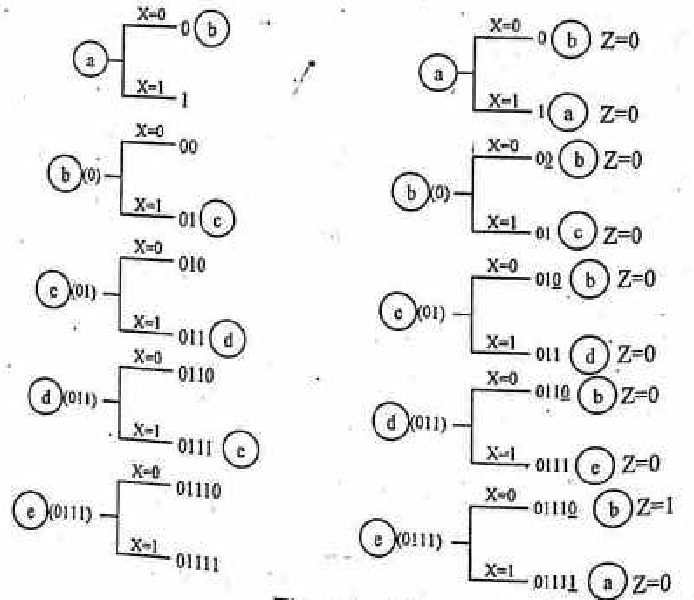


Figure 3.132

The state diagram of a sequence detector can be easily drawn by using the diagram shown in figure 3.132.

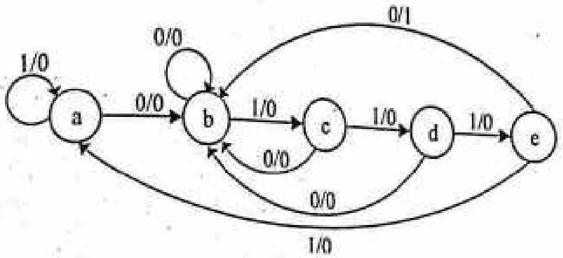


Figure 3.133 State diagram

Step 2: Draw the state table

Present state	Next state		Output	
	X=0	X=1	X=0	X=1
a	b	a	0	0
b	b	c	0	0
c	b	d	0	0
d	b	e	0	0
e	b	a	1	0

Table 3.141 State table

Step 3: Reduce the number of states if possible

Here state reduction is not possible

Step 4: Assign binary values to the states and plot the transition table using the excitation table of D Flip-flop.

Since there are 5 states, assign $a = 000$, $b = 001$, $c = 010$, $d = 011$ and $e = 100$. Use the following D Flip-flop excitation table to find the value of D_A, D_B and D_C.

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

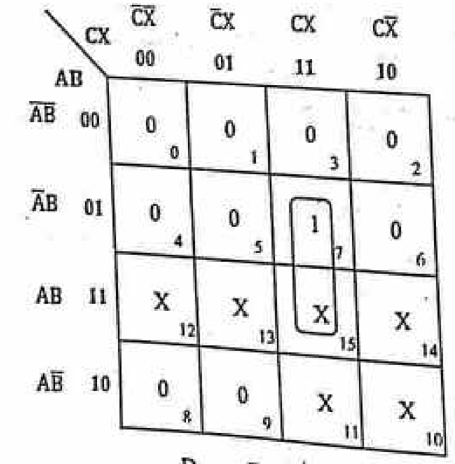
Table 3.142 Excitation table of D Flip-flop

Present state			Input	Next state			Flip-flop inputs			Output
A	B	C	X	A^+	B^+	C^+	D_A	D_B	D_C	Z
0	0	0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0	0	1	0
0	1	0	1	0	1	1	0	1	1	0
0	1	1	0	0	0	1	0	0	1	0
0	1	1	1	1	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	1	1
1	0	0	1	0	0	0	0	0	0	0

Table 3.143 Transition table

Step 5 & 6: Derive the Flip-flop input equations and output equation and draw the logic diagram.

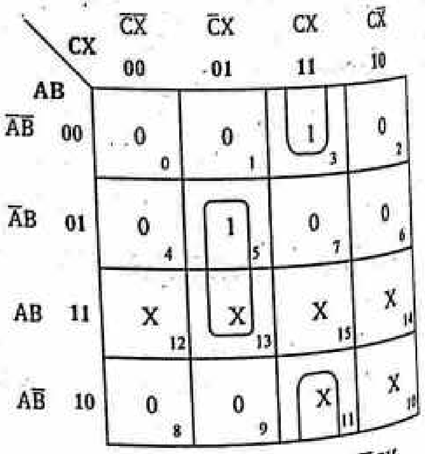
K-map for D_A



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0	0	0	0
$\overline{A}B$ 01	0	0	1	0
AB 11	X	X	X	X
$A\overline{B}$ 10	0	0	X	X

$$D_A = BCX$$

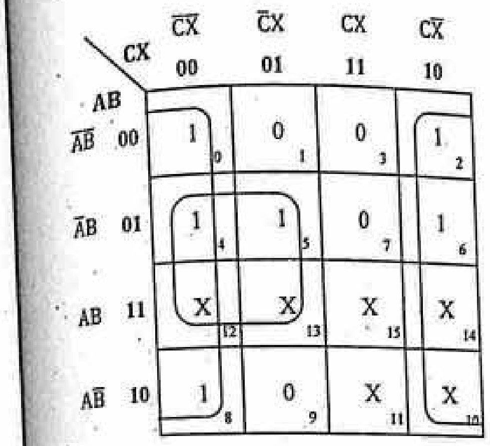
K-map for D_B



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0	0	1	0
$\overline{A}B$ 01	0	1	0	0
AB 11	X	X	X	X
$A\overline{B}$ 10	0	0	X	X

$$D_B = B\overline{C}X + \overline{B}CX$$

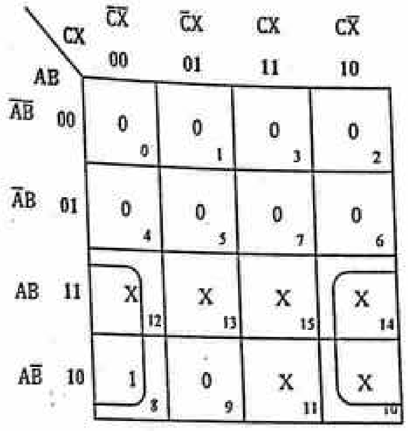
K-map for D_C



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	1	0	0	1
$\overline{A}B$ 01	1	1	0	1
AB 11	X	X	X	X
$A\overline{B}$ 10	1	0	X	X

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

K-map for Z



AB \ CX	$\overline{C}\overline{X}$ 00	$\overline{C}X$ 01	CX 11	$C\overline{X}$ 10
$\overline{A}\overline{B}$ 00	0	0	0	0
$\overline{A}B$ 01	0	0	0	0
AB 11	X	X	X	X
$A\overline{B}$ 10	1	0	X	X

$$Z = A\overline{X}$$

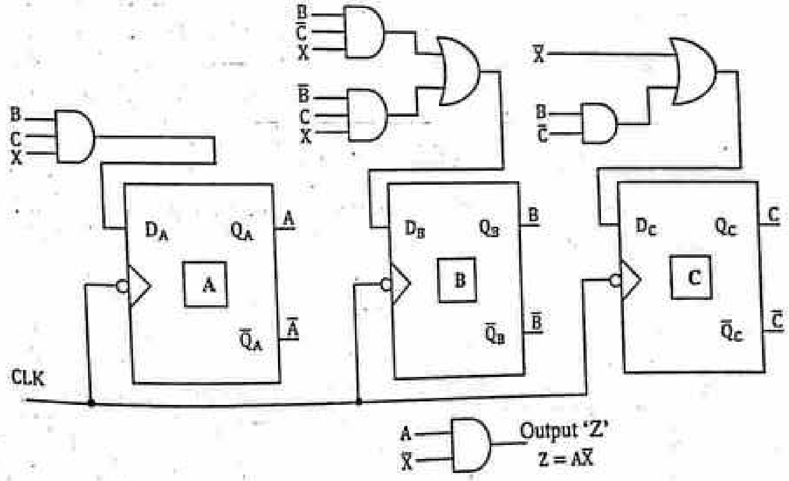


Figure 3.134 Logic diagram

3.13 SEQUENCE GENERATORS

A sequential circuit which generates a prescribed binary sequence with respect to clock is known as sequence generator.

Design procedure

The procedure for designing sequence generator is given below.

1. From the given sequence, draw the state diagram. The state diagram can be drawn as follows.
    a. Assign odd numbers to 1's in ascending order.
    b. Assign '0' and even numbers to 0's in ascending order.
    c. The number assigned to the corresponding bit represents its state.

For example, the state diagram for the sequence generator 110100 can be drawn as follows.

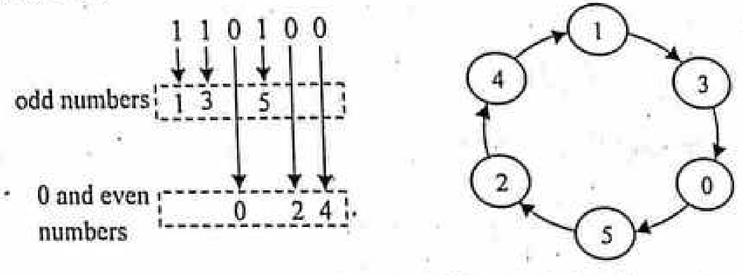


Figure 3.135 State diagram

2. Plot the state table.
3. Reduce the number of states if possible.
4. Assign binary values to the states and plot the transition table by choosing the type of Flip-flop.
5. Derive the Flip-flop input equations and output equations by using K-map.
6. Draw the logic diagram.

***Example 3.29:** Design a sequence generator using JK flip-flop to generate the sequence 1101011*

Solution:

Step 1: Draw the state diagram.

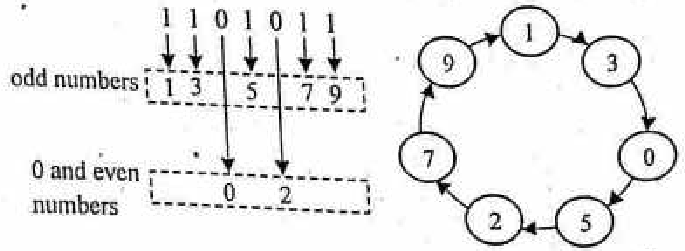


Figure 3.136 State diagram

Step 2: Plot the state table

Present state	Next state
1	3
3	0
0	5
5	2
2	7
7	9
9	1

Table 3.144 State table

Step 3: Assign binary values to the states and plot the transition table.

Use the following JK Flip-flop excitation table to find the value of J_A, K_A, J_B, K_B, J_C, K_C, J_D and K_D.

A	A^+	J_A	K_A
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table 3.145 Excitation table of JK Flip-flop

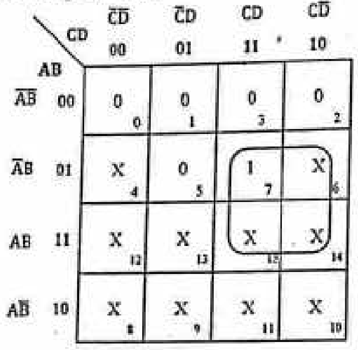
$1_{10} = 0001_2$; $3_{10} = 0011_2$; $0_{10} = 0000_2$; $5_{10} = 0101_2$; $2_{10} = 0010_2$; $7_{10} = 0111_2$; $9_{10} = 1001_2$

Present state				Next state				Flip-flop inputs							
A	B	C	D	A^+	B^+	C^+	D^+	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0	0	0	1	0	0	1	1	0	X	0	X	1	X	X	0
0	0	1	1	0	0	0	0	0	X	0	X	X	1	X	1
0	0	0	0	0	1	0	1	0	X	1	X	0	X	1	X
0	1	0	1	0	0	1	0	0	X	X	1	1	X	X	1
0	0	1	0	0	1	1	1	0	X	1	X	X	0	1	X
0	1	1	1	1	0	0	1	1	X	X	1	X	1	X	0
1	0	0	1	0	0	0	1	X	1	0	X	0	X	X	0

Table 3.146 Transition table

Step 4: Derive the Flip-flop input equations using K-map.

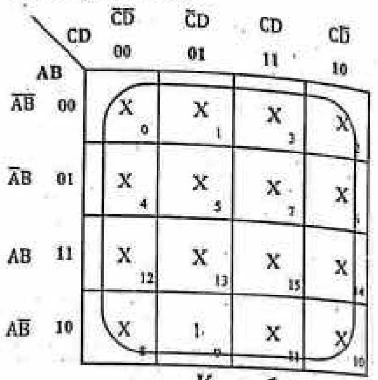
K-map for J_A



AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	0_{1}	0_{3}	0_{2}
$\bar{A}B$ 01	X_{4}	0_{5}	1_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$J_A = BC$

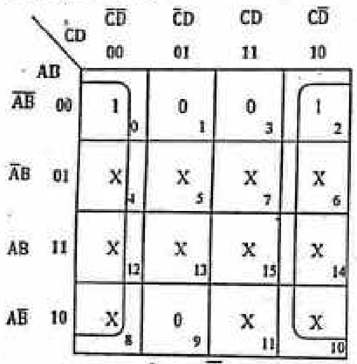
K-map for K_A



AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	1_{9}	X_{11}	X_{10}

$K_A = 1$

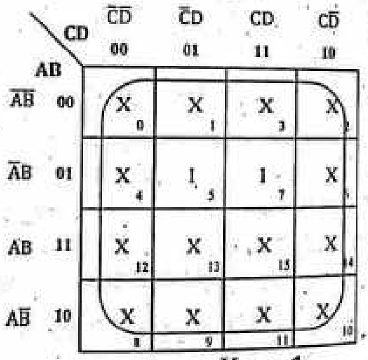
K-map for J_B



AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1_{0}	0_{1}	0_{3}	1_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	0_{9}	X_{11}	X_{10}

$J_B = \bar{D}$

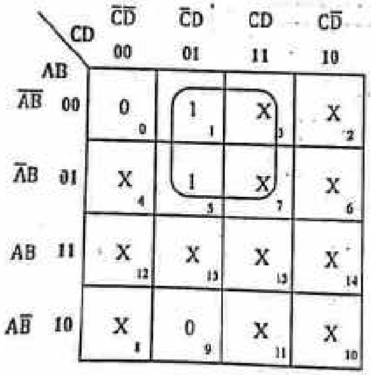
K-map for K_B



AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	X_{4}	1_{5}	1_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$K_B = 1$

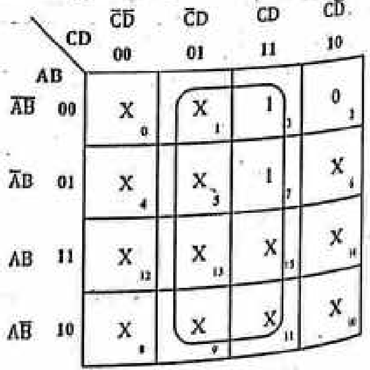
K-map for J_C



AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_{0}	1_{1}	X_{3}	X_{2}
$\bar{A}B$ 01	X_{4}	1_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	0_{9}	X_{11}	X_{10}

$J_C = \bar{A}D$

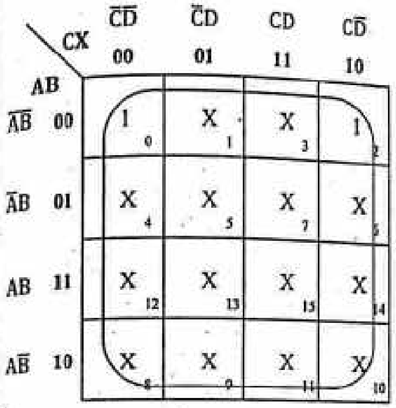
K-map for K_C



AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	X_{1}	1_{3}	0_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	1_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$K_C = D$

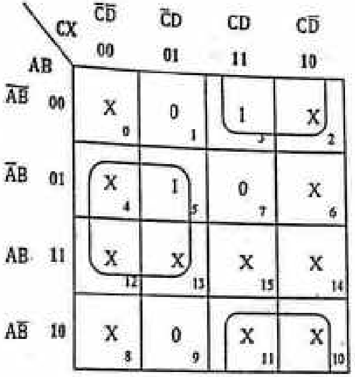
K-map for J_D



AB \ CX	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1_{0}	X_{1}	X_{3}	1_{2}
$\bar{A}B$ 01	X_{4}	X_{5}	X_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	X_{9}	X_{11}	X_{10}

$J_D = 1$

K-map for K_D



AB \ CX	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	X_{0}	0_{1}	1_{3}	X_{2}
$\bar{A}B$ 01	X_{4}	1_{5}	0_{7}	X_{6}
AB 11	X_{12}	X_{13}	X_{15}	X_{14}
$A\bar{B}$ 10	X_{8}	0_{9}	X_{11}	X_{10}

$K_D = B\bar{C} + \bar{B}C$

Step 5: Draw the logic diagram.

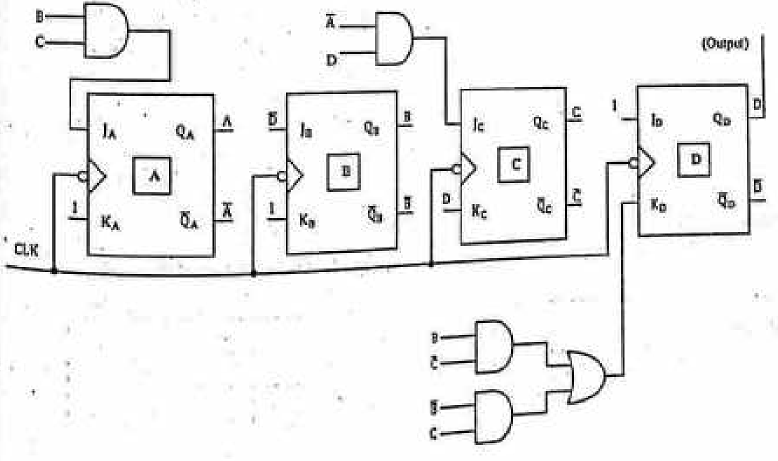
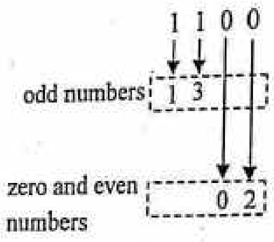


Figure 3.137: Logic diagram

Example 3.30 Design a sequence generator to generate the sequence 1100 using D-Flip flop

Solution:

Step 1: Draw the state diagram.



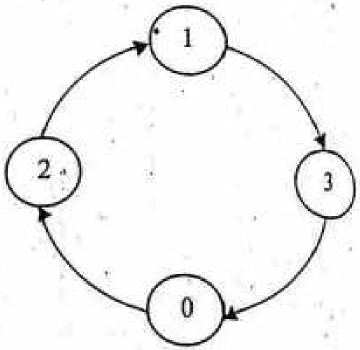


Figure 3.138 State diagram

Step 2: Plot the state table

Present state	Next state
1	3
3	0
0	2
2	1

Table 3.147 State table

Step 3: Assign binary values to the states and plot the transition table.

Use the following D Flip-flop excitation table to find the value of D_A and D_B

A	A^+	D_A
0	0	0
0	1	1
1	0	0
1	1	1

Table 3.148 Excitation table of D Flip-flop

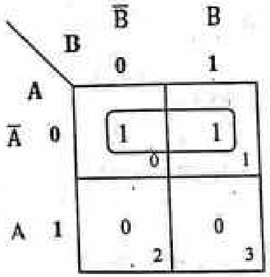
$1_{10} = 01_2$; $3_{10} = 11_2$; $0_{10} = 00_2$; $2_{10} = 10_2$

Present state		Next state		Flip-flop inputs	
A	B	A^+	B^+	D_A	D_B
0	1	1	1	1	1
1	1	0	0	0	0
0	0	1	0	1	0
1	0	0	1	0	1

Table 3.149 Excitation table

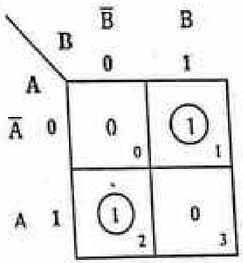
Step 4: Derive the flip-flop input equations using K-map.

K-map for D_A



$D_A = \overline{A}$

K-map for D_B



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

Step 5: Draw the logic diagram.

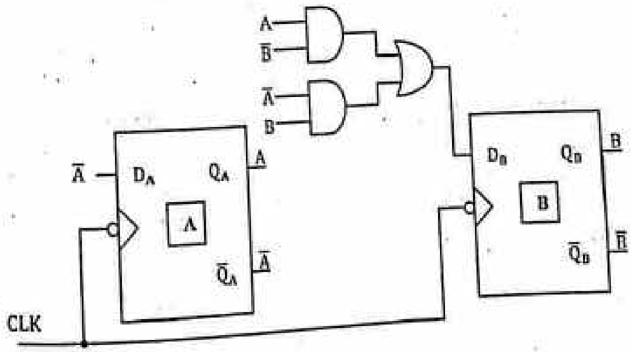


Figure 3.139 Logic diagram