CS 8351 – DIGITAL PRINCIPLES AND SYSTEM DESIGN

UNIT – III : SYNCHRONOUS SEQUENTIAL CIRCUITS

CLASSIFICATION OF SYNCHRONOUS SEQUENTIAL CIRCUIT:

In synchronous or clocked sequential circuits, clocked Flip-Flops are used as memory elements, which change their individual states in synchronism with the periodic clock signal. Therefore, the change in states of Flip-Flop and change in state of the entire circuits occur at the transition of the clock signal.

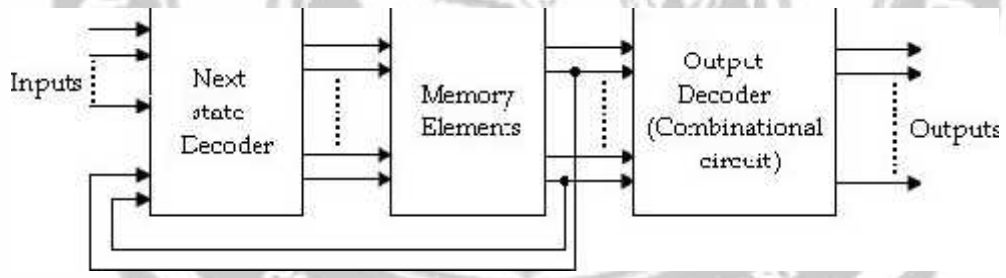
The synchronous or clocked sequential networks are represented by two models.

Moore model: The output depends only on the present state of the Flip-Flops.

Mealy model: The output depends on both the present state of the Flip-Flops and on the inputs.

Moore model:

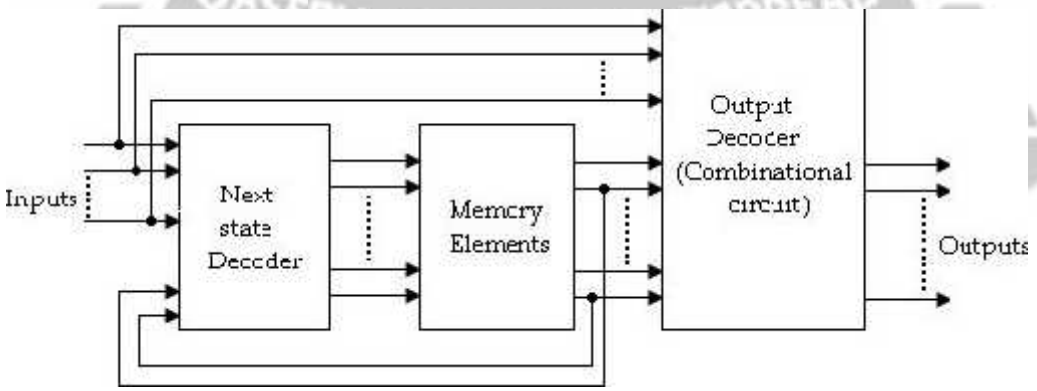
In the Moore model, the outputs are a function of the present state of the Flip- Flops only. The output depends only on present state of Flip-Flops, it appears only after the clock pulse is applied, i.e., it varies in synchronism with the clock input.



Moore model

Mealy model:

In the Mealy model, the outputs are functions of both the present state of the Flip-Flops and inputs.



Mealy model

Difference between Moore and Mealy model

Sl.No	Moore model	Mealy model
1	Its output is a function of present state only.	Its output is a function of present state as well as present input.
2	Input changes does no affect the output.	Input changes may affect the output of the circuit.
3	It requires more number of states for implementing same function.	It requires less number of states for implementing same function.

ANALYSIS OF SYNCHRONOUS SEQUENTIAL CIRCUIT:

The behavior of a sequential circuit is determined from the inputs, outputs and the state of its Flip- Flops. The outputs and the next state are both a function of the inputs and the present state. The analysis of a sequential circuit consists of obtaining a table or diagram from the time sequence of inputs, outputs and internal states.

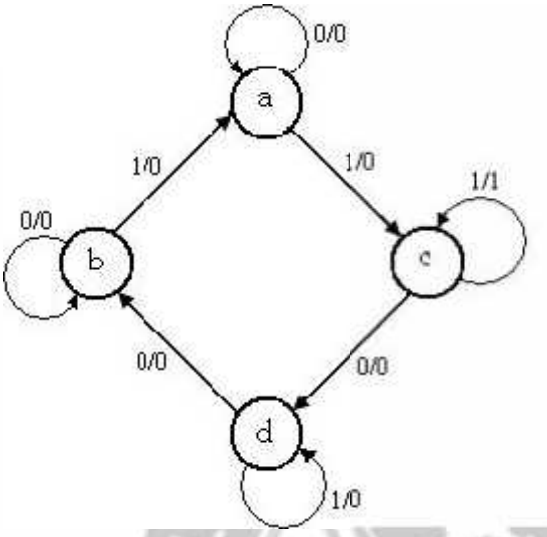
Before going to see the analysis and design examples, we first understand the state diagram, state table.

State Diagram

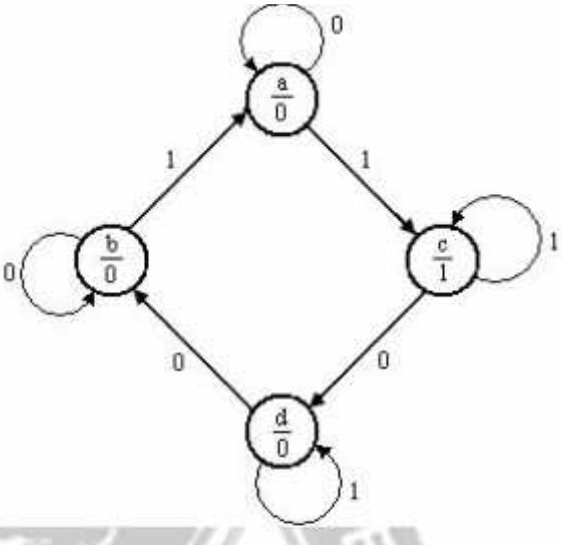
State diagram is a pictorial representation of a behavior of a sequential circuit.

- In the state diagram, a state is represented by a circle and the transition between states is indicated by directed lines connecting the circles.
- A directed line connecting a circle with circle with itself indicates that next state is same as present state.
- The binary number inside each circle identifies the state represented by the circle.
- The directed lines are labeled with two binary numbers separated by a symbol '/'. The input value that causes the state transition is labeled first and the output value during the present state is labeled after the symbol '/'.

In case of Moore circuit, the directed lines are labeled with only one binary number representing the state of the input that causes the state transition. The output state is indicated within the circle, below the present state because output state depends only on present state and not on t e input.



State diagram for Mealy circuit



State diagram for Moore circuit

State Table

State table represents relationship between input, output and Flip-Flop states.

It consists of three sections labeled present state, next state and output.

- The present state designates the state of Flip-Flops before the occurrence of a clock pulse, and the output section gives the values of the output variables during the present state.
- Both the next state and output sections have two columns representing two possible input conditions: X= 0 and X=1.

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

- In case of Moore circuit, the output section has only one column since output does not depend on input.

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

State Equation

It is an algebraic expression that specifies the condition for a Flip-Flop state transition.

The Flip-Flops may be of any type and the logic diagram may or may not include combinational circuit gates.

ANALYSIS PROCEDURE

The synchronous sequential circuit analysis is summarizes as given below:

1. Assign a state variable to each Flip-Flop in the synchronous sequential circuit.
2. Write the excitation input functions for each Flip-Flop and also write the Moore/ Mealy output equations.
3. Substitute the excitation input functions into the bistable equations for the Flip-Flops to obtain the next state output equations.
4. Obtain the state table and reduced form of the state table.
5. Draw the state diagram by using the second form of the state table.

Analysis of Mealy Model

1.A sequential circuit has two JK Flip-Flops A and B, one input (x) and one output (y). the Flip-Flop input functions are,

$J_A = B + x$ $J_B = A' + x'$

$K_A = 1$ $K_B = 1$

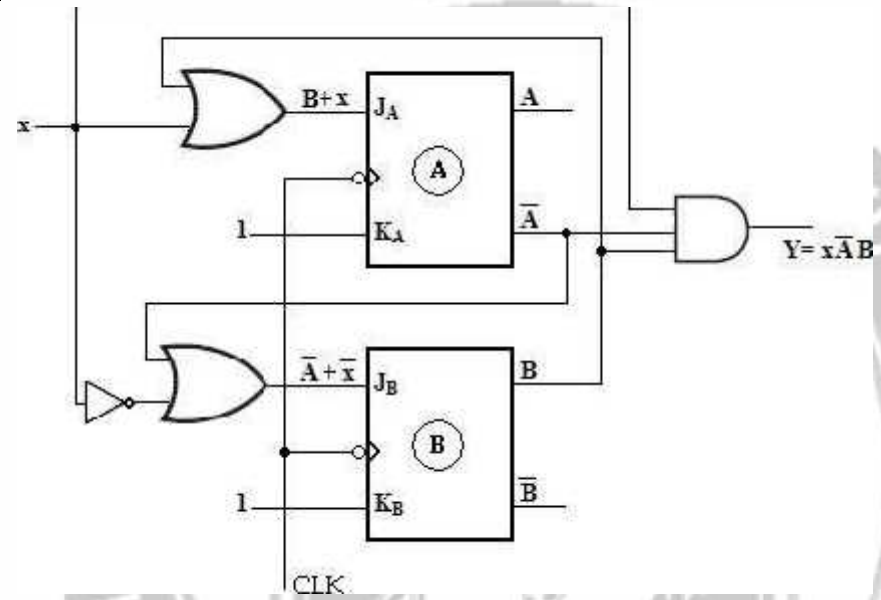
and the circuit output function, **Y= xA'B**.

a) Draw the logic diagram of the Mealy circuit,

b) Tabulate the state table,

c) Draw the state diagram.

Soln:



State table:

To obtain the next-state values of a sequential circuit with JK Flip-Flops, use the JK Flip-Flop characteristics table.

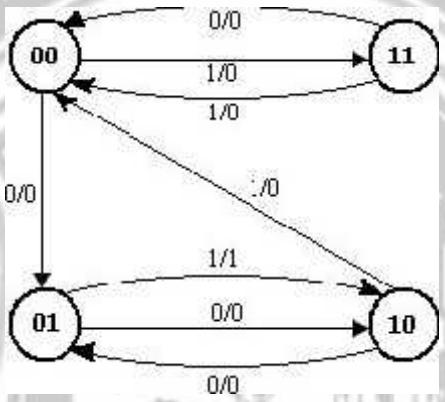
Present state		Input	Flip-Flop Inputs				Next state		Output
A	B	x	J_A= B+ x	K_A= 1	J_B= A'+ x'	K_B= 1	A(t+1)	B(t+1)	Y= xA'B
0	0	0	0	1	1	1	0	1	0
0	0	1	1	1	1	1	1	1	0
0	1	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	1	1	0	1	0
1	0	1	1	1	0	1	0	0	0
1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	1	0	0	0

Present state		Next state				Output	
		x= 0		x= 1		x= 0	x= 1
A	B	A	B	A	B	y	y

0	0	0	1	1	1	0	0
0	1	1	0	1	0	0	1
1	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0

Second form of state table

State Diagram:



State Diagram

2. A sequential circuit with two 'D' Flip-Flops A and B, one input (x) and one output (y). the Flip-Flop input functions are:

   D_A= Ax+ Bx

   D_B= A'x and the circuit output function is,

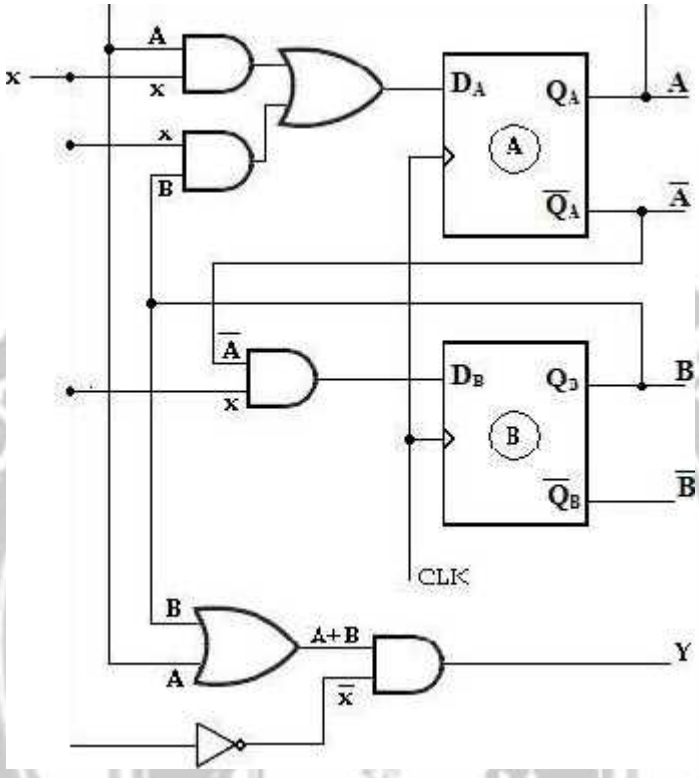
   Y= (A+ B) x'.

   (a) Draw the logic diagram of the circuit,

   (b) Tabulate the state table,

   (c) Draw the state diagram.

Soln:



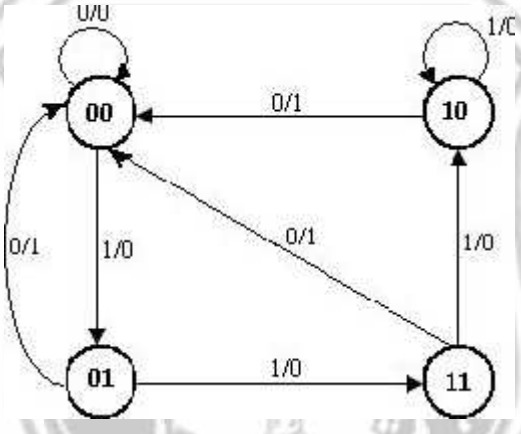
State Table:

Present state		Input	Flip-Flop Inputs		Next state		Output
A	**B**	**x**	**D_A= Ax+Bx**	**D_B= A'x**	**A(t+1)**	**B(t+1)**	**Y= (A+B)x'**
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0

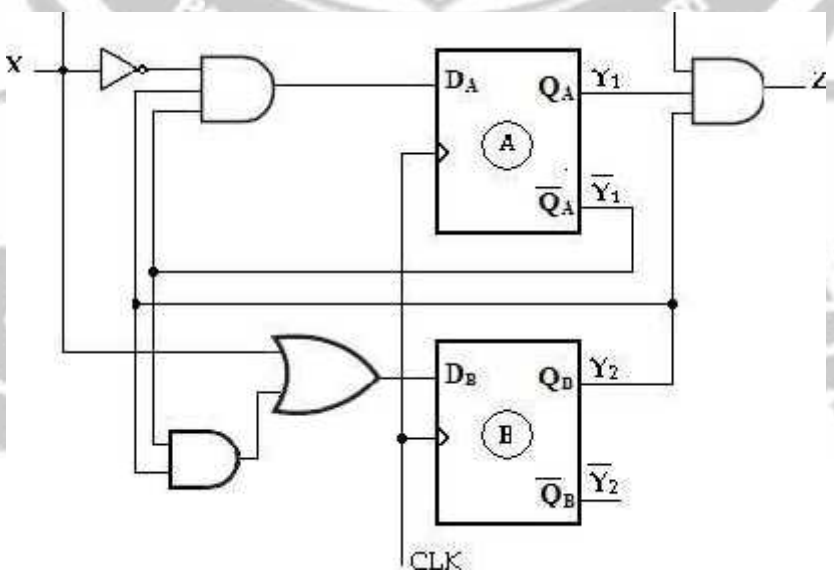
Present state		Next state				Output	
		x = 0		x= 1		x= 0	x= 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Second form of state table

State Diagram:



3. Analyze the synchronous Mealy machine and obtain its state diagram.



Soln:

The given synchronous Mealy machine consists of two D Flip-Flops, one inputs and one output. The Flip-Flop input functions are,

$D_A = Y_1'Y_2X'$

$D_B = X + Y_1'Y_2$

The circuit output function is, $Z = Y_1Y_2X$

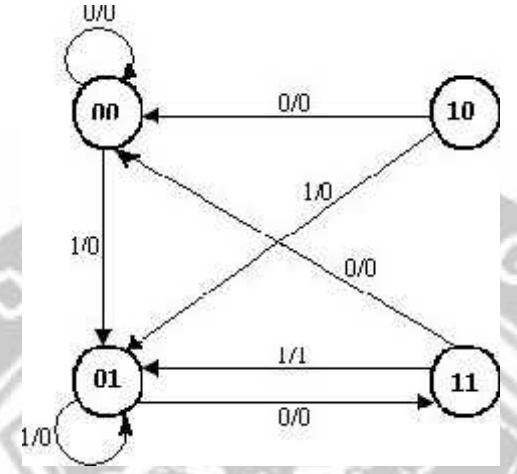
State Table:

Present state		Input	Flip-Flop Inputs		Next state		Output
Y_1	Y_2	X	$D_A = Y_1'Y_2X'$	$D_B = X + Y_1'Y_2$	Y_1 (t+1)	Y_2 (t+1)	$Z = Y_1Y_2X$
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	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Present state		Next state				Output	
		X= 0		X = 1		X= 0	X= 1
Y_1	Y_2	Y_1	Y_2	Y_1	Y_2	Z	Z
0	0	0	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	0
1	1	0	0	0	1	0	1

Second form of state table

State Diagram:

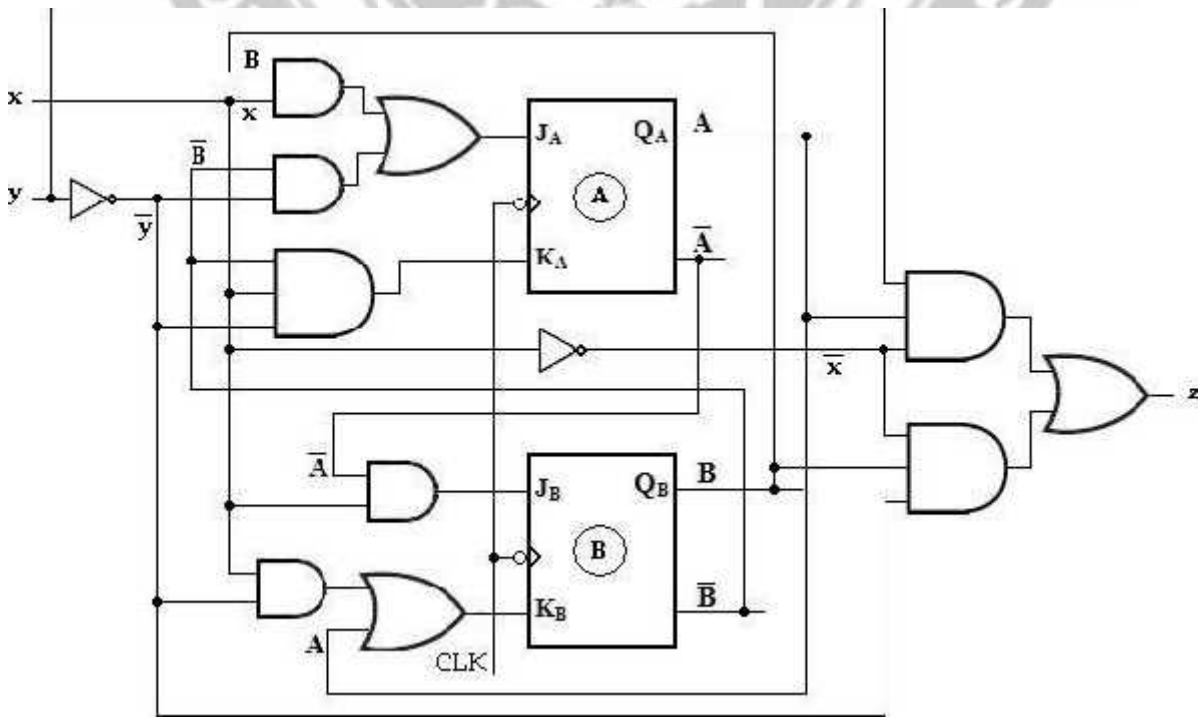


4. A sequential circuit has two JK Flop-Flops A and B, two inputs x and y and one output z. The Flip-Flop input equation and circuit output equations are

   $J_A = Bx + B'y'$ $K_A = B'xy'$ $J_B = A'x$ $K_B = A + xy'$, $z = Ax'y' + Bx'y'$

   (a) Draw the logic diagram of the circuit (b) Tabulate the state table. (c) Derive the state equation.

State diagram:



State table:

To obtain the next-state values of a sequential circuit with JK Flip-Flop, use the JK Flip-Flop characteristic table,

Present state		Input		Flip-Flop Inputs				Next state		Output
A	B	x	y	J_A= Bx+B'y'	K_A= B'xy'	J_B= A'x	K_B= A+xy'	A(t+1)	B(t+1)	z
0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1	1	1	0
0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	1	1	0
0	1	1	1	1	0	1	0	1	1	0
1	0	0	0	1	0	0	1	1	0	1
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	1	1	0	1	0	0	0
1	0	1	1	0	0	0	1	1	0	0
1	1	0	0	0	0	0	1	1	0	1
1	1	0	1	0	0	0	1	1	0	0
1	1	1	0	1	0	0	1	1	0	0
1	1	1	1	1	0	0	1	1	0	0

State Equation:

For A(t+1)

AB \ xy	00	01	11	10
00	1	0	0	1
01	0	0	1	1
11	1	1	1	1
10	1	1	1	0

$A(t+1) = Ax' + Ay - Bx + A'B'y'$

For B(t+1)

AB \ xy	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

$B(t+1) = A'x$

5. A sequential circuit has two JK Flip-Flop A and B. the Flip-Flop input functions are: J_A=

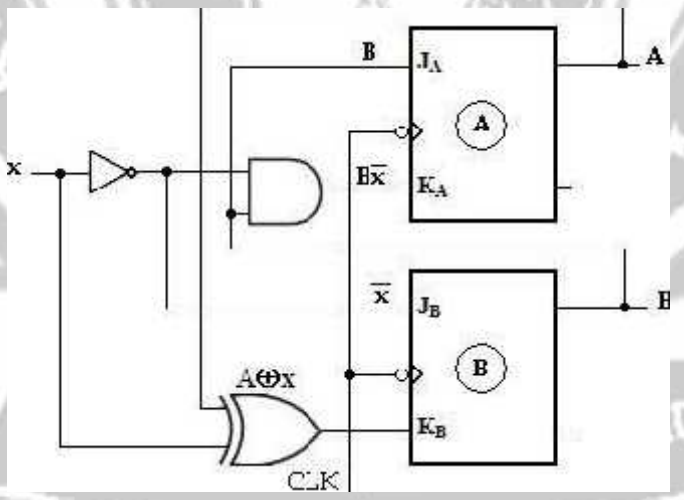
B J_B= x' K_A= Bx' K_B= A x.

(a) Draw the logic diagram of the circuit,

(b) Tabulate the state table,

(c) Draw the state diagram.

Logic diagram:



The output function is not given in the problem. The output of the Flip-Flops may be considered as the output of the circuit.

State table:

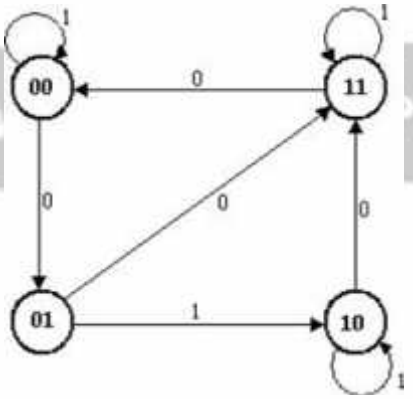
To obtain the next-state values of a sequential circuit with JK Flip-Flop, use the JK Flip-Flop characteristic table.

Present state		Input	Flip-Flop Inputs				Next state	
A	B	x	J_A= B	K_A= Bx'	J_B= x'	K_B= A x	A(t+1)	B(t+1)
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1

Present s ate		Next state			
		X= 0		X= 1	
A	B	A	B	A	B
0	0	0	1	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	0	0	1	1

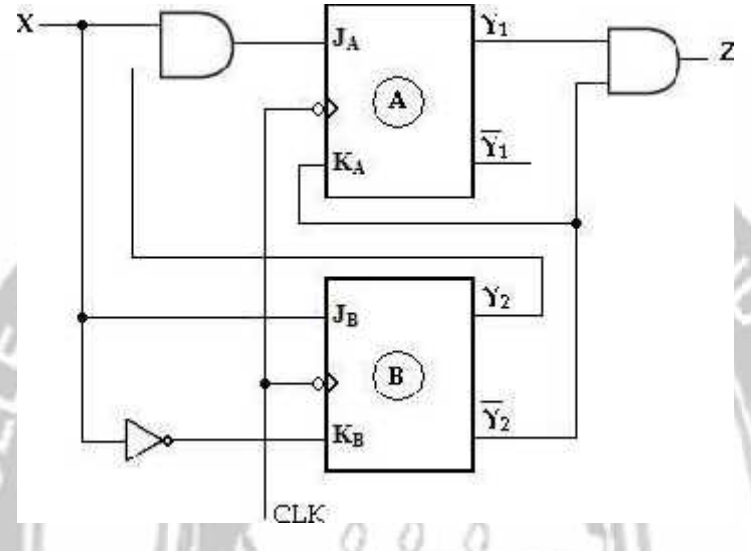
Second form of state table

State Diagram:



Analysis of Moore Model

6. Analyze the synchronous Moore circuit and obtain its state diagram.



Soln:

Using the assigned variable Y_1 and Y_2 for the two JK Flip-Flops, we can write the four excitation input equations and the Moore output equation as follows:

$J_A = Y_2X$; $K_A = Y_2'$

$J_B = X$; $K_B = X'$ and output function**, $Z = Y_1Y_2'$**

State table:

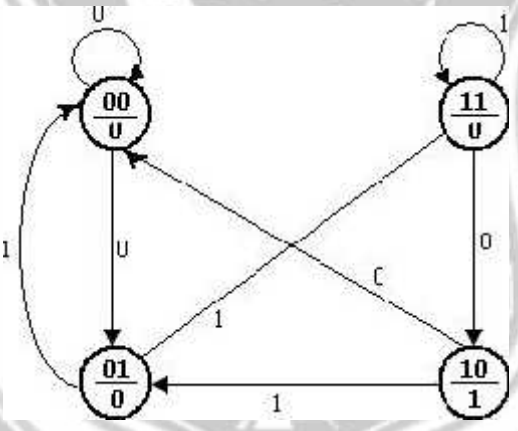
Present state		Input	Flip-Flop Inputs				Next state		Output
Y_1	Y_2	X	$J_A = Y_2X$	$K_A = Y_2'$	$J_B = X$	$K_B = X'$	Y_1 (t+1)	Y_2 (t+1)	$Z = Y_1Y_2'$
0	0	0	0	1	0	1	0	0	0
0	0	1	0	1	1	0	0	1	0
0	1	0	0	0	0	1	0	0	0
0	1	1	1	0	1	0	1	1	0
1	0	0	0	1	0	1	0	0	1
1	0	1	0	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0	0
1	1	1	1	0	1	0	1	1	0

Present state		Next state				Output
		X = 0		X= 1		Y
Y_1	Y_2	Y_1	Y_2	Y_1	Y_2	
0	0	0	0	0	1	0
0	1	0	0	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	0

Second form of state table **State**

Diagram:

Here the output depends on the present state only and is independent of the input. The two values inside each circle separated by a slash are for the present state and output.



7. A sequential circuit has two T Flip-Flop A and B. The Flip-Flop input functions are:

T_A= Bx **T_B= x**

y= AB

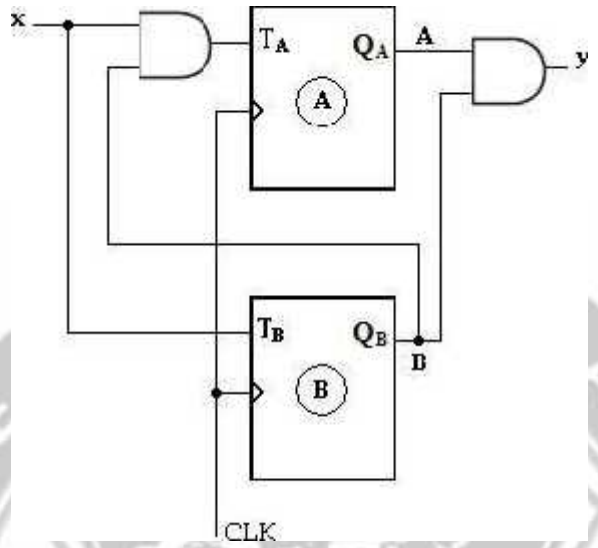
(a) Draw the logic diagram of the circuit,

(b) Tabulate the state table,

(c) Draw the state diagram.

Soln:

Logic diagram:



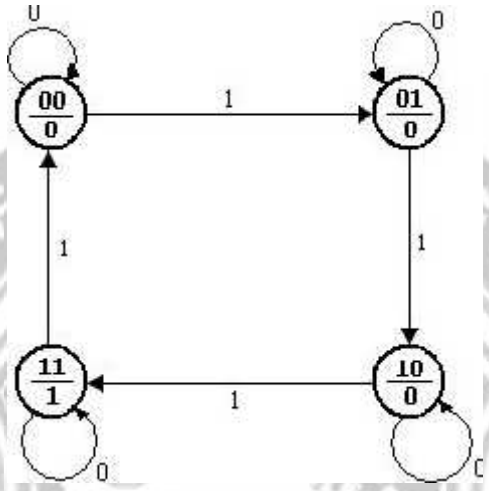
State table

Present state		Input	Flip-Flop Inputs		Next state		Output
A	B	x	T_A= Bx	T_B= x	A (t+1)	B (t+1)	y= AB
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	1

Present state		Next state				Output	
		x = 0		x= 1		x= 0	x= 1
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	1	1	0	0	0
1	0	1	0	1	1	0	0
1	1	1	1	0	0	1	1

Second form of state table

State Diagram:



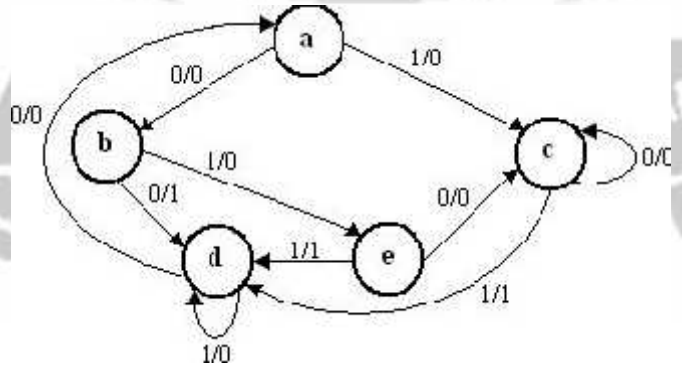
STATE REDUCTION/ MINIMIZATION

The state reduction is used to avoid the redundant states in the sequential circuits. The reduction in redundant states reduces the number of required Flip- Flops and logic gates, reducing the cost of the final circuit.

The two states are said to be redundant or equivalent, if every possible set of inputs generate exactly same output and same next state. When two states are equivalent, one of them can be removed without altering the input-output relationship.

Since 'n' Flip-Flops produced 2^n state, a reduction in the number of states may result in a reduction in the number of Flip-Flops.

The need for state reduction or state minimization is explained with one example.



State diagram

Step 1: Determine the state table for given state diagram

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

State table

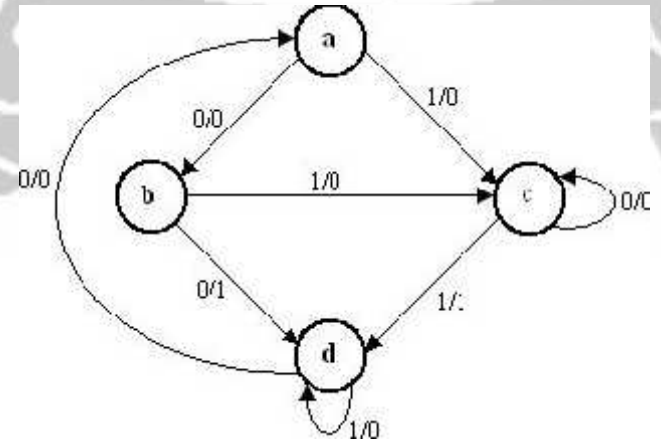
Step 2: Find equivalent states

From the above state table **c** and **e** generate exactly same next state and same output for every possible set of inputs. The state **c** and **e** go to next states **c** and **d** and have outputs 0 and 1 for x=0 and x=1 respectively. Therefore state **e** can be removed and replaced by **c**. The final reduced state table is shown below.

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

Reduced state table

The state diagram for the reduced table consists of only four states and is shown below.



Reduced state diagram

1.Reduce the number of states in the following state table and tabulate the reduced state table.

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

Soln: From the above state table **e** and **g** generate exactly same next state and same output for every possible set of inputs. The state **e** and **g** go to next states **a** and **f** and have outputs 0 and 1 for x=0 and x=1 respectively. Therefore state **g** can be removed and replaced by **e**.

The reduced state table-1 is shown below.

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

Reduced state table-1

Now states d and f are equivalent. Both states go to the same next state (e, f) and have same output (0, 1). Therefore one state can be removed; **f** is replaced by **d.**

The final reduced state table-2 is shown below.

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

Reduced state table-2 Thus 7 states are reduced into 5 states.

2.Determine a minimal state table equivalent furnished below

Present state	Next state	
	X= 0	X= 1
1	1, 0	1, 0
2	1, 1	6, 1
3	4, 0	5, 0
4	1, 1	7, 0
5	2, 0	3, 0
6	4, 0	5, 0
7	2, 0	3, 0

Soln:

Present state	Next state		Output	
	X= 0	X= 1	X= 0	X= 1
1	1	1	0	0
2	1	6	1	1
3	4	5	0	0
4	1	7	1	0
5	2	3	0	0
6	4	5	0	0
7	2	3	0	0

From the above state table, **5** and **7** generate exactly same next state and same output for every possible set of inputs. The state **5** and **7** go to next states **2** and **3** and have outputs 0 and 0 for x=0 and x=1 respectively. Therefore state **7** can be removed and replaced by **5**.

Similarly, **3** and **6** generate exactly same next state and same output for every possible set of inputs. The state **3** and **6** go to next states **4** and **5** and have outputs 0 and 0 for x=0 and x=1 respectively. Therefore state **6** can be removed and replaced by **3**.

The final reduced state table is shown below.

Present state	Next state		Output	
	X= 0	X= 1	X= 0	X= 1
1	1	1	0	0
2	1	3	1	1
3	4	5	0	0
4	1	5	1	0
5	2	3	0	0

Reduced state table Thus 7 states are reduced into 5 states.

3.Minimize the following state table.

Present state	Next state	
	X= 0	X= 1
A	D, 0	C, 1
B	E, 1	A, 1
C	H, 1	D, 1
D	D, 0	C, 1
E	B, 0	G, 1
F	H, 1	D, 1
G	A, 0	F, 1
H	C, 0	A, 1
I	G, 1	H,1

Soln:

Present state	Next state		Output	
	X= 0	X= 1	X= 0	X= 1
A	D	C	0	1
B	E	A	1	1
C	H	D	1	1
D	D	C	0	1
E	B	G	0	1
F	H	D	1	1
G	A	F	0	1
H	C	A	0	1
I	G	H	1	1

From the above state table, **A** and **D** generate exactly same next state and same output for every possible set of inputs. The state **A** and **D** go to next states **D** and **C** and have outputs 0 and 1 for x=0 and x=1 respectively. Therefore state **D** can be removed and replaced by **A**. Similarly, **C** and **F** generate exactly same next state and same output for every possible set of inputs. The state **C** and **F** go to next states **H** and **D** and have outputs 1 and 1 for x=0 and x=1 respectively. Therefore state **F** can be removed and replaced by **C**.

The reduced state table-1 is shown below.

Present state	Next state		Output	
	X= 0	X= 1	X= 0	X= 1
A	A	C	0	1
B	E	A	1	1
C	H	A	1	1
E	B	G	0	1
G	A	C	0	1
H	C	A	0	1
I	G	H	1	1

Reduced state table-1

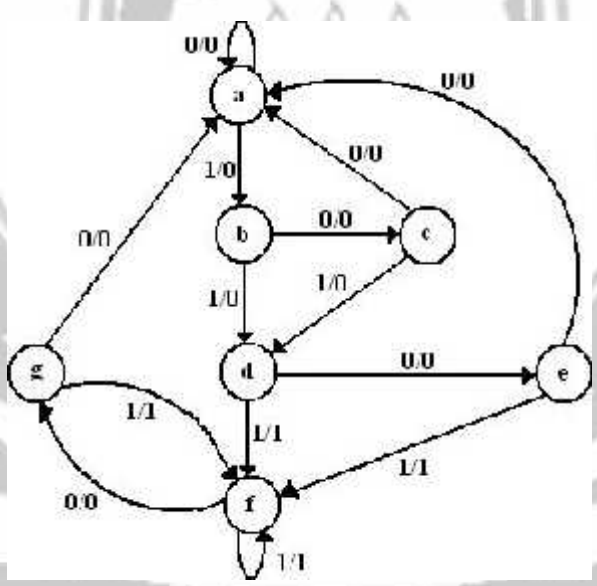
From the above reduced state table-1, **A** and **G** generate exactly same next state and same output for every possible set of inputs. The state **A** and **G** go to next states **A** and **C** and have outputs 0 and 1 for

x=0 and x=1 respectively. Therefore state **G** can be removed and replaced by **A**. The final reduced state table-2 is shown below.

Present state	Next state		Output	
	X= 0	X= 1	X= 0	X= 1
A	A	C	0	1
B	E	A	1	1
C	H	A	1	1
E	B	A	0	1
H	C	A	0	1
I	A	H	1	1

Reduced state table-2 Thus 9 states are reduced into 6 states.

4.Reduce the following state diagram.



Soln:

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

State table

From the above state table **e** and **g** generate exactly same next state and same output for every possible set of inputs. The state **e** and **g** go to next states **a** and **f** and have outputs 0 and 1 for x=0 and x=1 respectively. Therefore state **g** can be removed and replaced by **e**. The reduced state table-1 is shown below.

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

Reduced state table-1

Now states d and f are equivalent. Both states go to the same next state (e, f) and have same output (0, 1). Therefore one state can be removed; **f** is replaced by **d.**

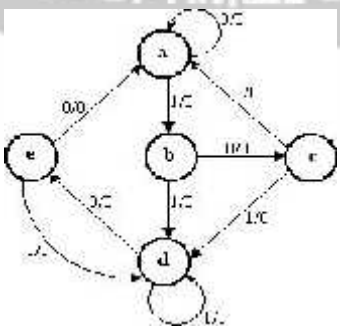
The final reduced state table-2 is shown below.

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

Reduced state table-2

Thus 7 states are reduced into 5 states.

The state diagram for the reduced state table-2 is,



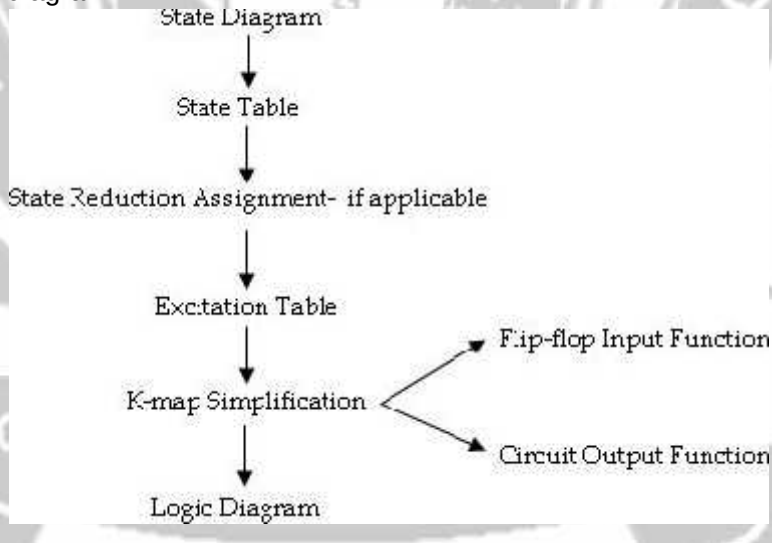
DESIGN OF SYNCHRONOUS SEQUENTIAL CIRCUITS:

A synchronous sequential circuit is made up of number of Flip-Flops and combinational gates. The design of circuit consists of choosing the Flip-Flops and then finding a combinational gate structure together with the Flip-Flops. The number of Flip-Flops is determined from the number of states needed in the circuit.

The combinational circuit is derived from the state table.

Design procedure:

1. The given problem is determined with a state diagram.
2. From the state diagram, obtain the state table.
3. The number of states may be reduced by state reduction methods (if applicable).
4. Assign binary values to each state (Binary Assignment) if the state table contains letter symbols.
5. Determine the number of Flip-Flops and assign a letter symbol (A, B, C,…) to each.
6. Choose the type of Flip-Flop (SR, JK, D, T) to be used.
7. From the state table, circuit excitation and output tables.
8. Using K-map or any other simplification method, derive the circuit output functions and the Flip- Flop input functions.
9. Draw the logic diagram.



The type of Flip-Flop to be used may be included in the design specifications or may depend what is available to the designer. Many digital systems are constructed with JK Flip-Flops because they are the most versatile available. The selection of inputs is given as follows.

Flip-Flop	Application
J K D T	General Applications Applications requiring transfer of data (Ex: Shift Registers) Application involving complementation (Ex: Binary Counters)

Excitation Tables:

Before going to the design examples for the clocked synchronous sequential circuits we revise Flip- Flop excitation tables.

Present State	Next State	Inputs	
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation table for SR Flip-Flop

Present State	Next State	Inputs	
Q_n	Q_{n+1}	J	K
0	0	0	x x
0	1	1 x	1
1	0	x	0
1	1		

Excitation table for JK Flip-Flop

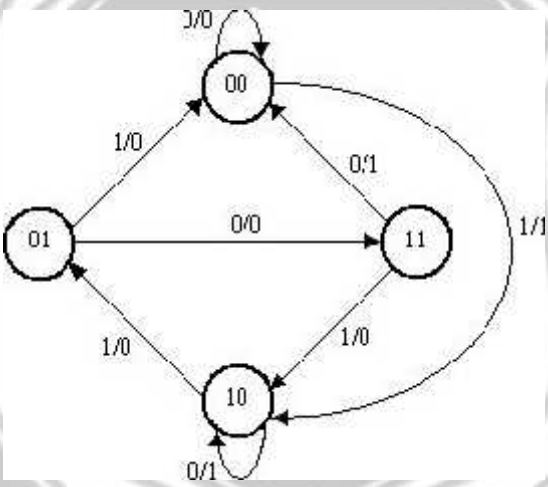
Present State	Next State	Input
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table for T Flip-Flop

Present State	Next State	Input
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation table for D Flip-Flop

1.A sequential circuit has one input and one output. The state diagram is shown below. Design the sequential circuit with a) D-Flip-Flops, b) T Flip-Flops, c) RS Flip-Flops and d) JK Flip-Flops.



Solution:

State Table:

The state table for the state diagram is,

Present state		Next state		Output	
		X= 0	X= 1	X= 0	X= 1
A	B	AB	AB	Y	Y
0	0	00	10	0	1
0	1	11	00	0	0
1	0	10	01	1	0
1	1	00	10	1	0

State reduction:

As seen from the state table there is no equivalent states. Therefore, no reduction in the state diagram.

The state table shows that circuit goes through four states, therefore we require 2 Flip-Flops (number of states= 2^m, where m= number of Flip-Flops). Since two Flip-Flops are required first is denoted as A and second is denoted as B.

i) Design using D Flip-Flops:

Excitation table:

Using the excitation table for T Flip-Flop, we can determine the excitation table for the given circuit as,

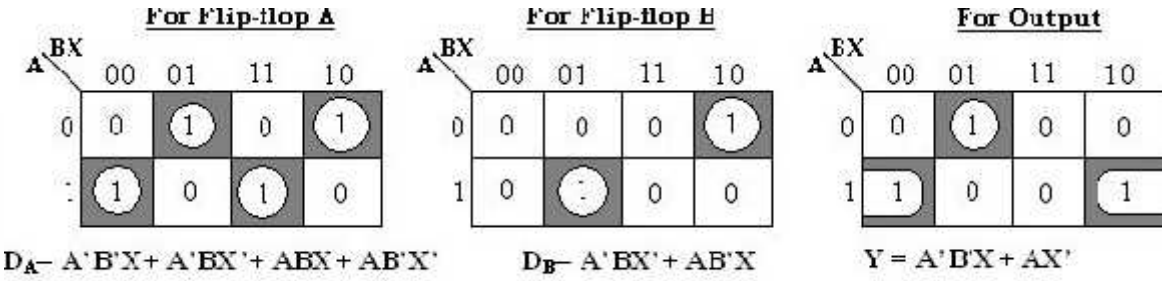
Present State	Next State	Input
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation table for D Flip-Flop

Present state		Input	Next state		Flip-Flop Inputs		Output
A	B	X	A	B	D_A	D_B	Y
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1
0	1	0	1	1	1	1	0
0	1	1	0	0	0	0	0
1	0	0	1	0	1	0	1
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0

Circuit excitation table

K-map Simplification:



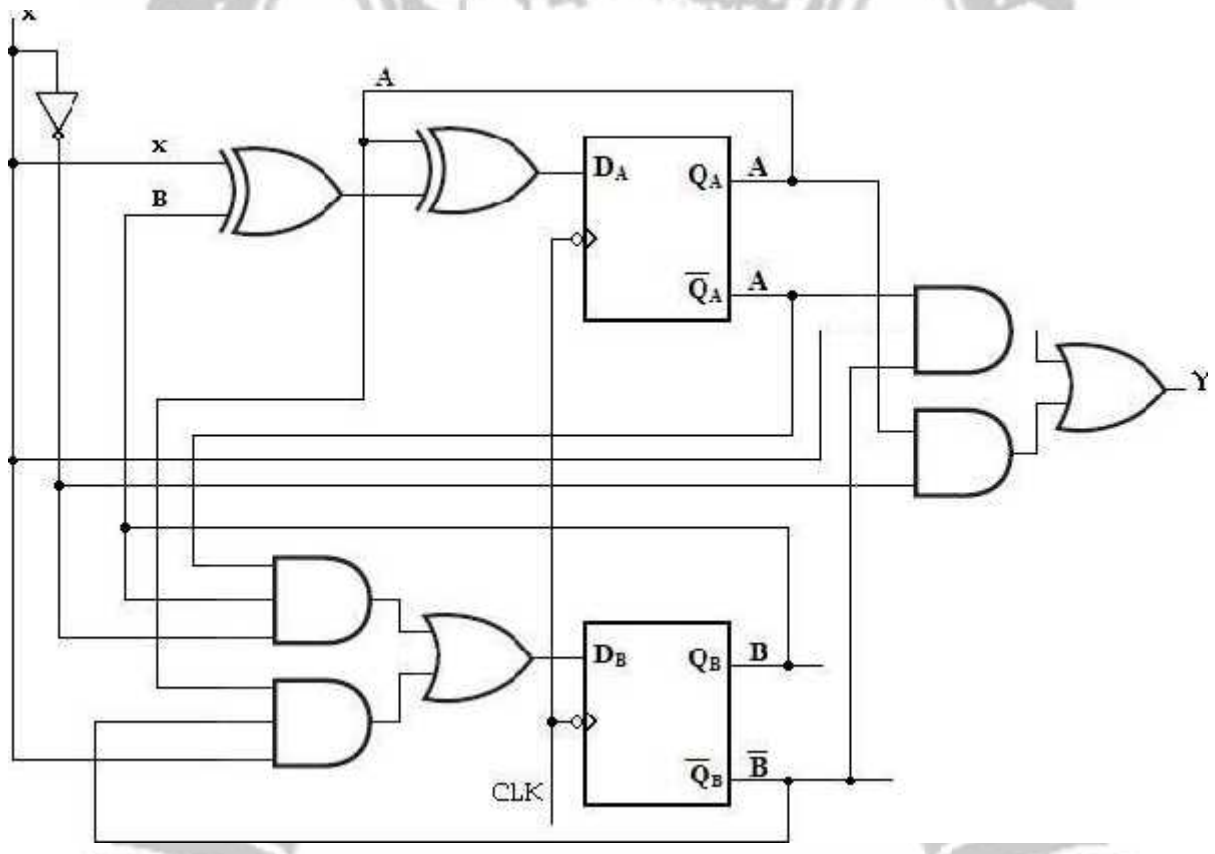
$D_A = A'B'X + A'BX' + ABX + AB'X'$

$= A \oplus (B \oplus x)$

$D_B = A'BX' + AB'X$

$Y = A'B'X + AX'$

With these Flip-Flop input functions and circuit output function we can draw the logic diagram as follows.



Logic diagram of given sequential circuit using D Flip-Flop

ii) Design using T Flip-Flops:

Using the excitation table for T Flip-Flop, we can determine the excitation table for the given circuit as,

Present State	Next State	Input
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

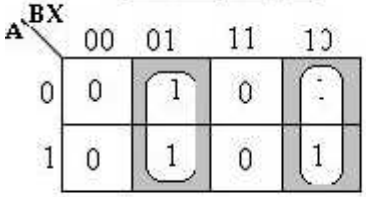
Excitation table for T Flip-Flop

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	1	0	1	0	1
0	1	0	1	1	1	0	0
0	1	1	0	0	0	1	0
1	0	0	1	0	0	0	1
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	1
1	1	1	1	0	0	1	0

Circuit excitation table

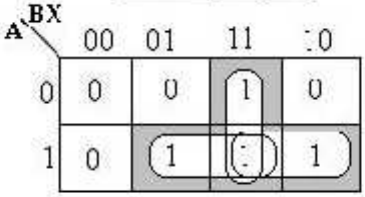
K-map Simplification:

For Flip-flop A



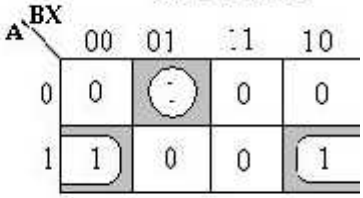
$T_A = B \oplus x$

For Flip-flop B



$T_B = AB + AX + BX$

For Output



$Y = A'BX + AX'$

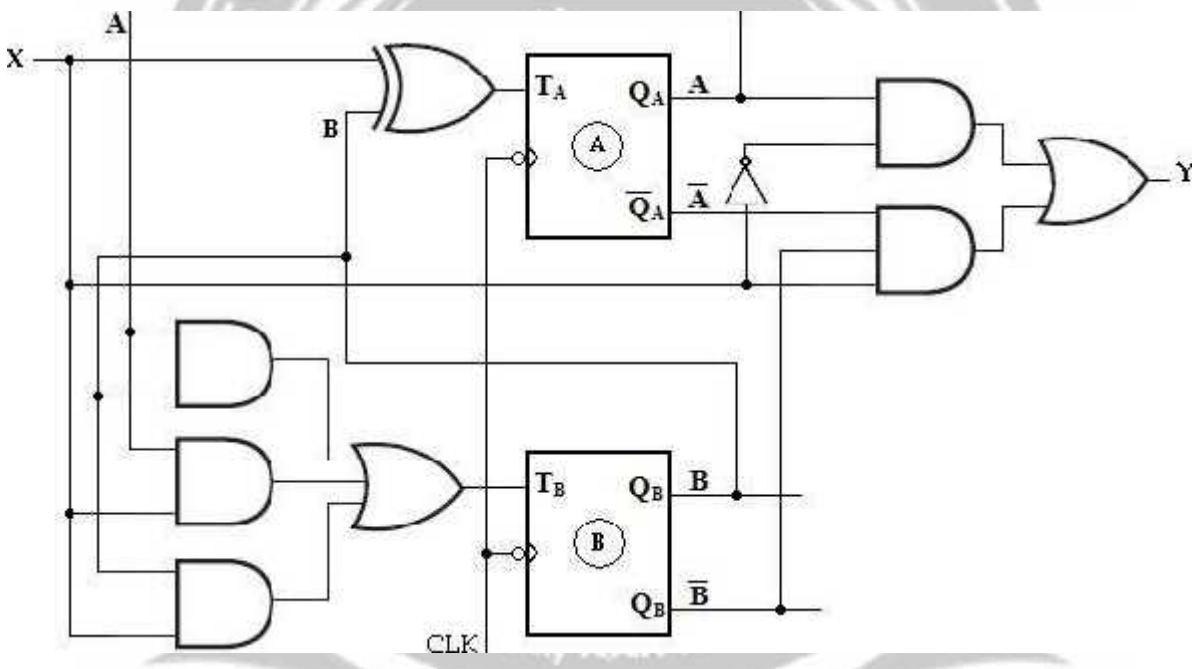
Therefore, input functions for,

T_A= B x and

T_B= AB+ AX+ BX

Circuit output function, **Y = XA'B'+ X'A**

With these Flip-Flop input functions and circuit output function we can draw the logic diagram as follows.



Logic diagram of given sequential circuit using T Flip-Flop

iii)Design using SR Flip-Flops:

Using the excitation table for RS Flip-Flop, we can determine the excitation table for the given circuit as,

Present State	Next State	Inputs	
Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

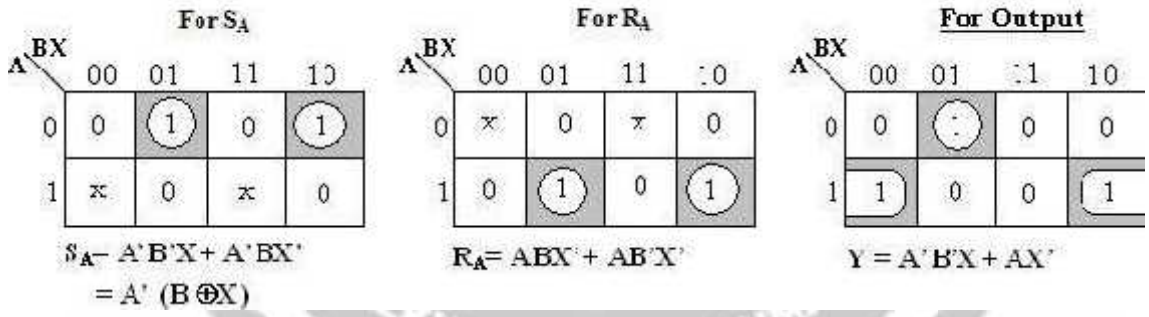
Excitation table for SR Flip-Flop

Present state		Input	Next state		lip-Flop Inputs				Output
A	B	X	A	B	S_A	R_A	S_B	R_B	Y
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

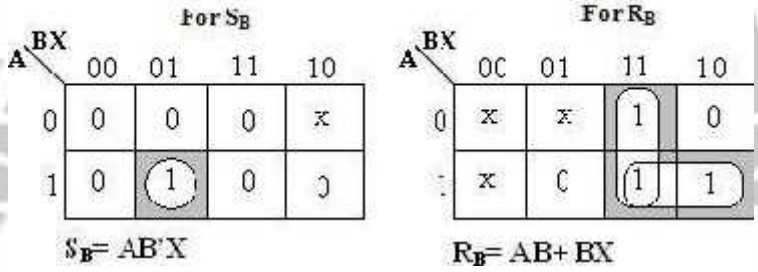
Circuit excitation table

K-map Simplification:

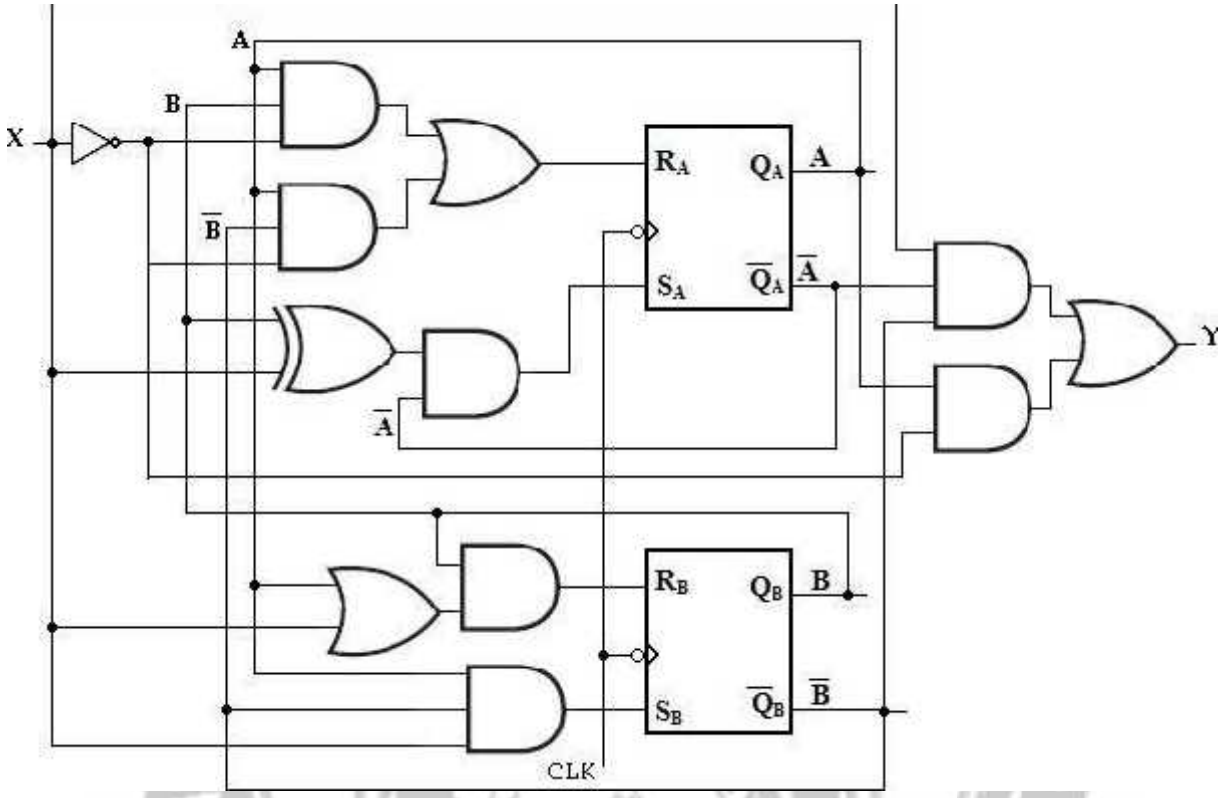
For Flip-flop A



For Flip-flop B



With these Flip-Flop input functions and circuit output function we can draw the logic diagram as follows.



iii) Design using JK Flip-Flops:

Using the excitation table for JK Flip-Flop, we can determine the excitation table for the given circuit as,

Present State	Next State	Inputs	
Q_n	Q_{n+1}	J	K
0	0	0 1	x x
0	1	x x	1
1	0		0
1	1		

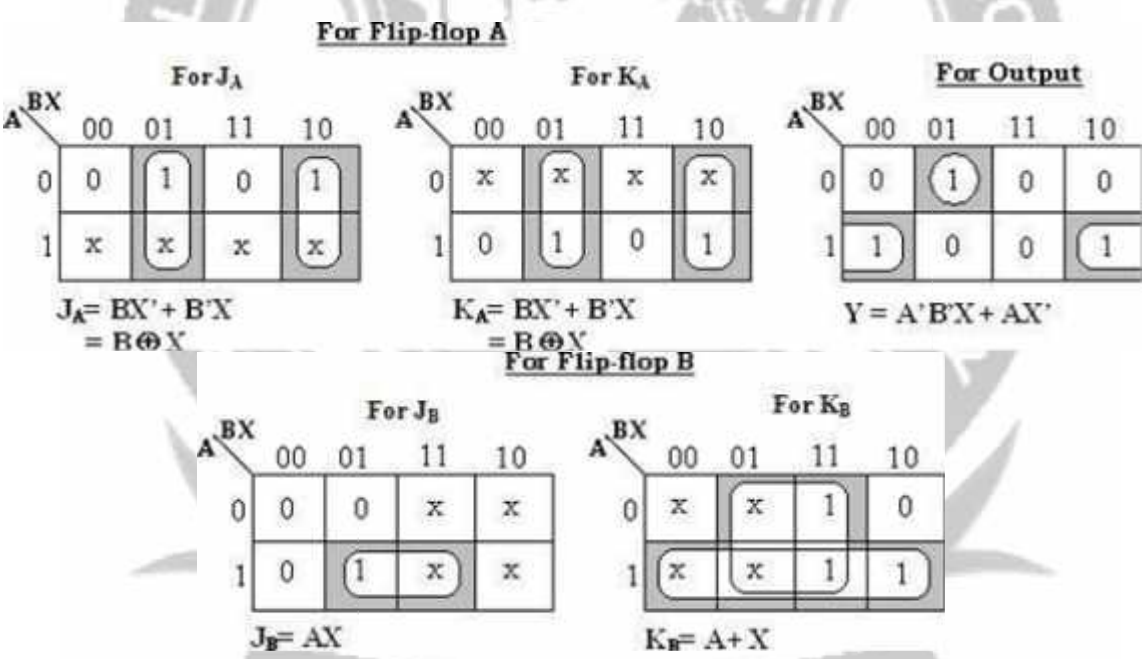
Excitation table for JK Flip-Flop

Present state		Input	Next state		Flip-Flop Inputs				Output
A	B	X	A	B	J_A	K_A	J_B	K_B	Y
0	0	0	0	0	0	x	0	x	0

0	0	1	1	0	1	x	0	x	1
0	1	0	1	1	1	x	x	0	0
0	1	1	0	0	0	x	x	1	0
1	0	0	1	0	x	0	0	x	1
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	1
1	1	1	1	0	x	0	x	1	0

Circuit excitation table

K-map Simplification:



The input functions for,

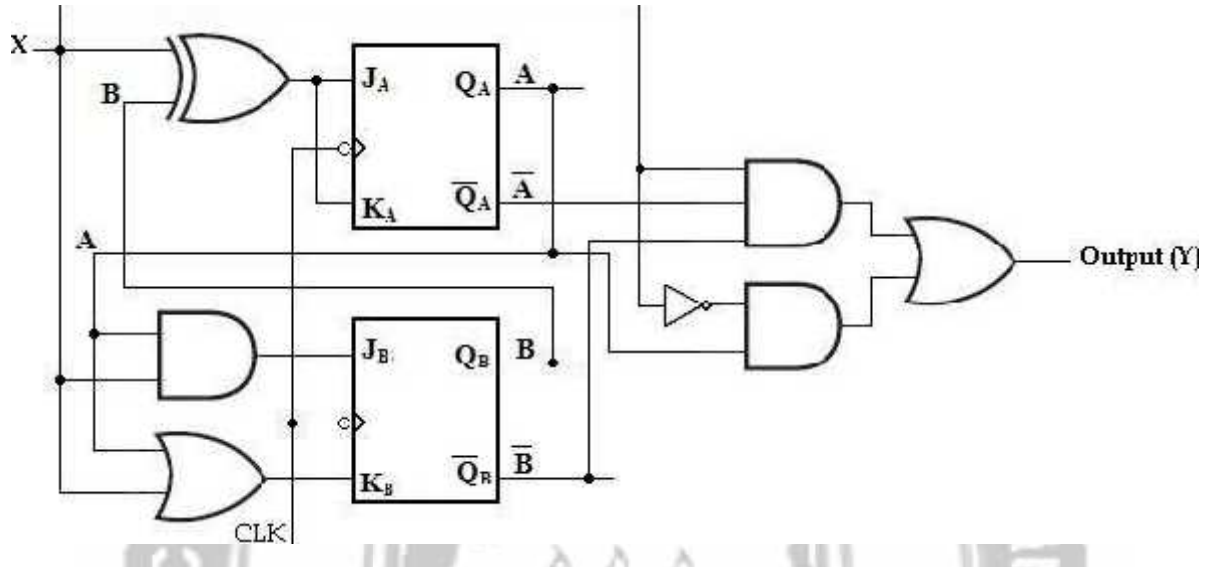
J_A= BX'+ B'X **J_B= AX**
= B

X

K_A= BX'+ B'X **K_B= A+ X**
= B X

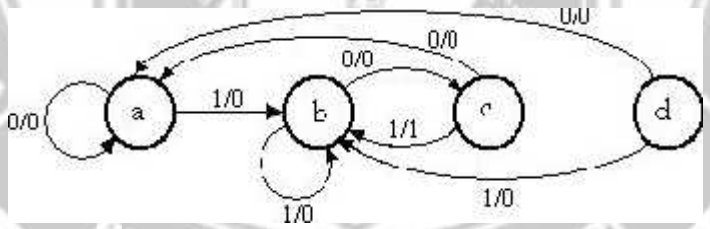
Circuit output function, **Y= AX'+ A'B'X**

With these Flip-Flop input functions and circuit output function we can draw the logic diagram as follows.



Logic diagram of given sequential circuit using JK Flip-Flop

2. Design a clocked sequential machine using JK Flip-Flops for the state diagram shown in the figure. Use state reduction if possible. Make proper state assignment.



Soln:

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
d	a	b	0	0

From the above state table **a** and **d** generate exactly same next state and same output for every possible set of inputs. The state **a** and **d** go to next states **a** and **b** and have outputs 0 and 0 for x=0 and x=1

respectively. Therefore state **d** can be removed and replaced by **a**. The final reduced state table is shown below.

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

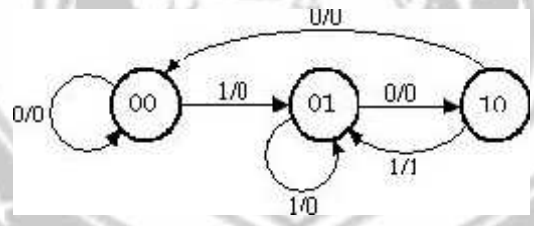
Reduced State table

Binary Assignment:

Now each state is assigned with binary values. Since there are three states, number of Flip-Flops required is two and 2 binary numbers are assigned to the states.

a= 00; b= 0; and c= 10

The reduced state diagram is drawn as,



Reduced State Diagram

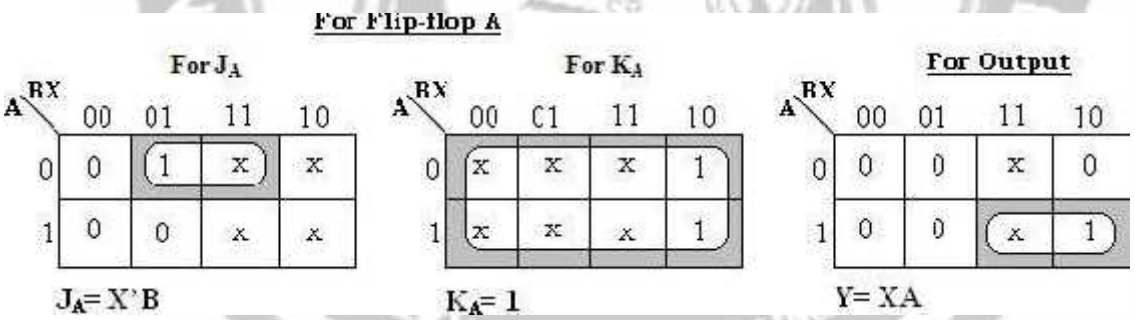
Excitation Table:

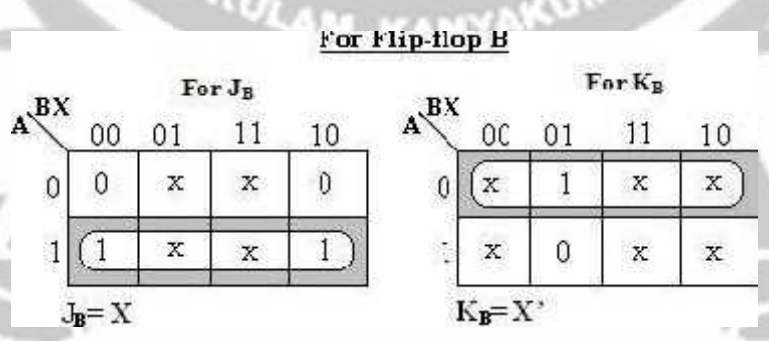
Present State	Next State	Inputs	
Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1		0

Excitation table for JK Flip-Flop

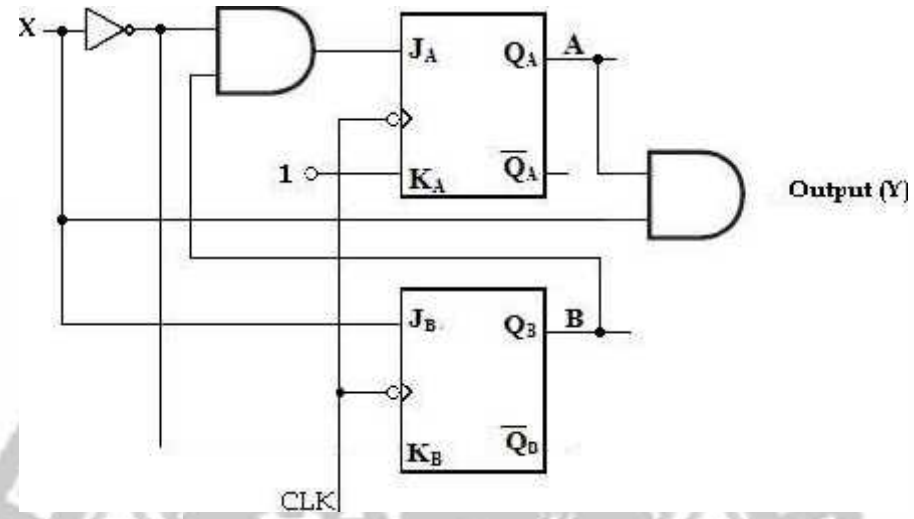
Input	Present state		Next state		lip-Flop Inputs				Output
X	A	B	A	B	J_A	K_A	J_B	K_B	Y
0	0	0	0	0	0	x	0	x	0
1	0	0	0	1	0	x	1	x	0
0	0	1	1	0	1	x	x	1	0
1	0	1	0	1	0	x	x	0	0
0	1	0	0	0	x	1	0	x	0
1	1	0	0	1	x	1	1	x	1
0	1	1	x	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x	x

K-map Simplification:

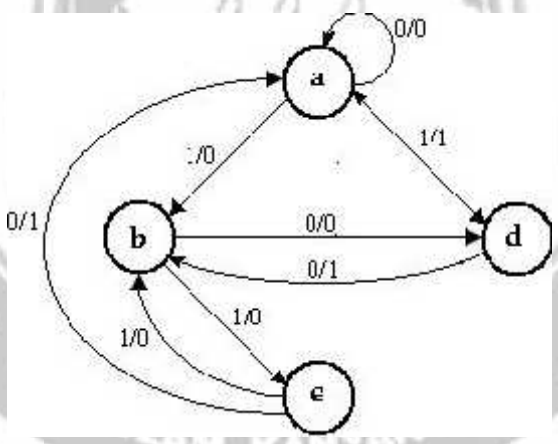




With these Flip-Flop input functions and circuit output function we can draw the logic diagram as follows.



3. Design a clocked sequential machine using T Flip-Flops for the following state diagram. Use state reduction if possible. Also use straight binary state assignment.



Soln:

State Table:

State table for the given state diagram is,

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

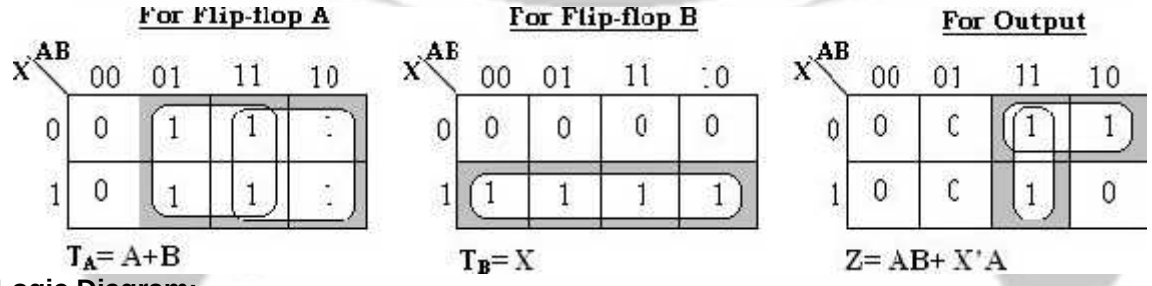
Even though a and c are having same next states for input X=0 and X=1, as the outputs are not same state reduction is not possible.

State Assignment:

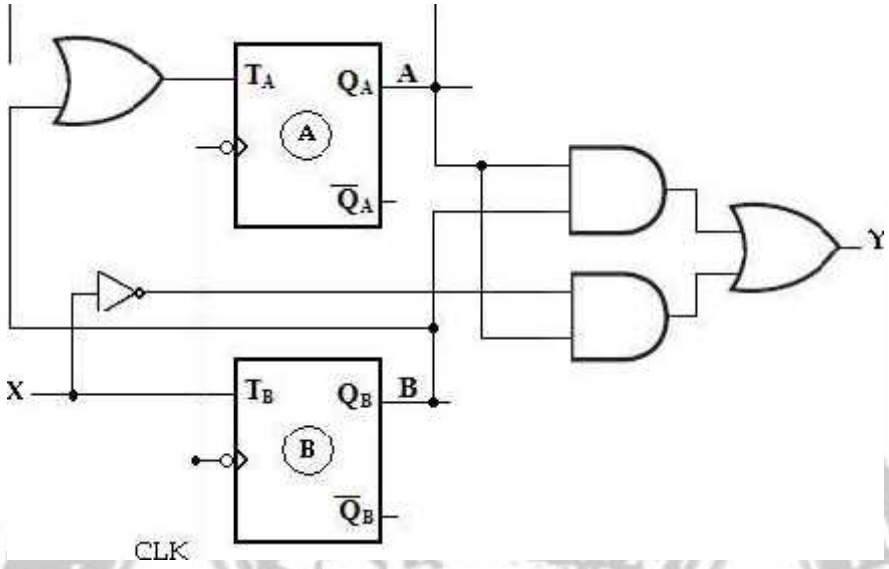
Use straight binary assignments as a= 00, b= 01, c= 10 and d= 11, the transition table is,

Input	Present state		Next state		Flip-Flop Inputs		Output
X	A	B	A	B	T_A	T_B	Y
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	1
0	1	1	0	1	1	0	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	1	0
1	1	0	0	1	1	1	0
1	1	1	0	0	1	1	1

K-map simplification:



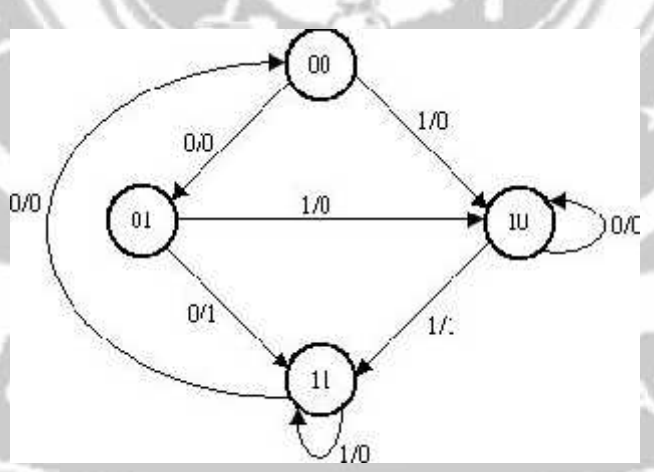
Logic Diagram:



STATE ASSIGNMENT:

In sequential circuits, the behavior of the circuit is defined in terms of its inputs, present states, next states and outputs. To generate desired next state at particular present state and inputs, it is necessary to have specific Flip-Flop inputs. These Flip-Flop inputs are described by a set of Boolean functions called Flip-Flop input functions.

To determine the Flip-Flop functions, it is necessary to represent states in the state diagram using binary values instead of alphabets. This procedure is known as ***state assignment***.



Reduced state diagram with binary states

Rules for state assignments

There are two basic rules for making state assignments.

Rule 1:

States having the **same** NEXT STATES for a given input condition should have assignments which can be grouped into logically adjacent cells in a K-map.

Rule 2:

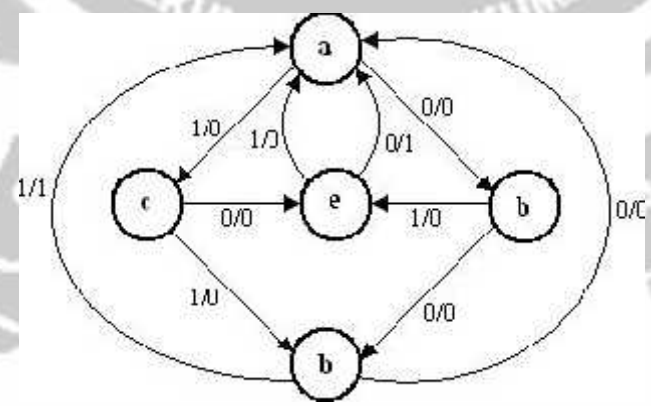
States that are the NEXT STATES of a single state should have assignment which can be grouped into logically adjacent cells in a K-map.

Present state	Next state		Output	
	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

State table with assignment states

State Assignment Problem:

1. Design a sequential circuit for a state diagram shown below. Use state assignment rules for assigning states and compare the required combinational circuit with random state assignment.

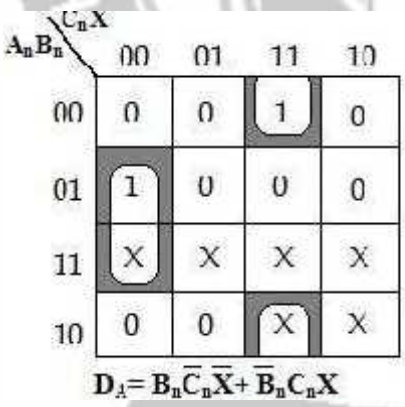


Using random state assignment we assign, a= 000, b= 001, c= 010, d= 011 and e= 100.

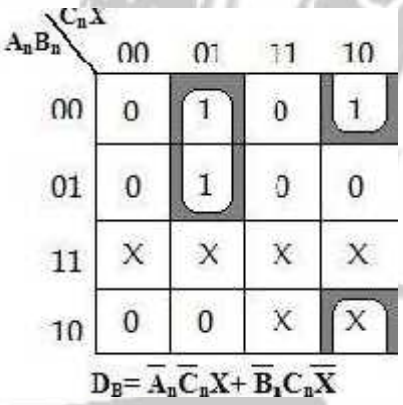
The excitation table with these assignments is given as,

Present state			Input	Next state			Output
A_n	B_n	C_n	X	A_{n+1}	B_{n+1}	C_{n+1}	Z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	1	1	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

K-map Simplification:



$D_A = B_n\overline{C}_n\overline{X} + \overline{B}_nC_nX$



$D_B = \overline{A}_n\overline{C}_nX + \overline{B}_1C_n\overline{X}$

A_nB_n \ C_nX	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	X	X	X	X
10	0	0	X	X

$D_C = \overline{A}_n\overline{B}_n\overline{X} + B_n\overline{C}_nX$

A_nB_n \ C_nX	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

$Z = B_nC_nX + A_n\overline{X}$

The random assignments require:

7 three input AND

functions 1 two input AND

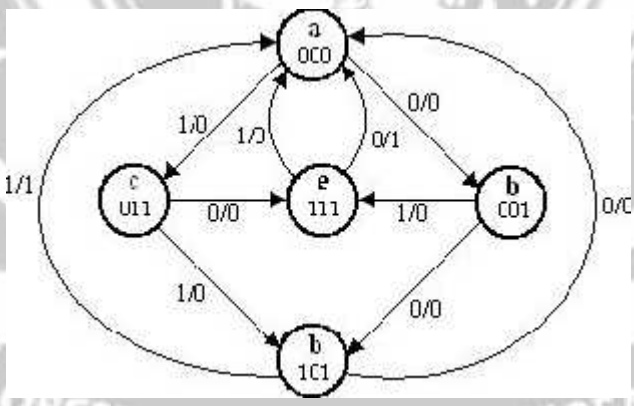
function

4 two input OR functions

---

12 gates with 31 inputs

Now, we will apply the state assignment rules and compare the results.



State diagram after applying Rules 1 and 2

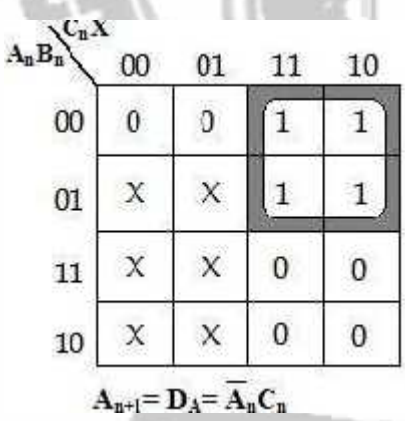
Rule 1 says that: e and d must be adjacent, and b and c must be adjacent.

Rule 2 says that: e and d must be adjacent, and b and c must be adjacent.

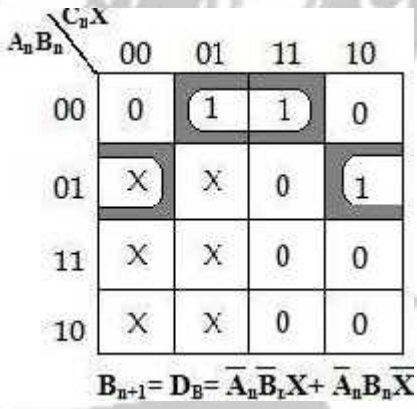
Applying Rule 1, Rule 2 to the state diagram we get the state assignment as,

Present state			Input	Next state			Output
A_n	B_n	C_n	X	A_{n+1}	B_{n+1}	C_{n+1}	Z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	1	0
0	0	1	0	1	0	1	0
0	0	1	1	1	1	1	0
0	1	0	0	x	x	x	x
0	1	0	1	x	x	x	x
0	1	1	0	1	1	1	0
0	1	1	1	1	0	1	0
1	0	0	0	x	x	x	x
1	0	0	1	x	x	x	x
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0

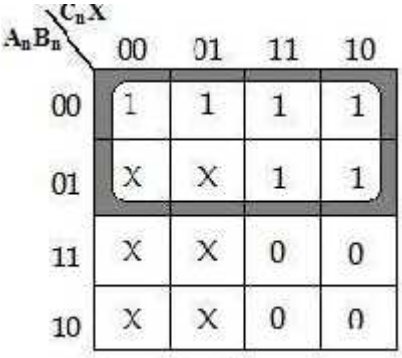
K-map Simplification:



$A_{n+1} = D_A = \overline{A}_n C_n$

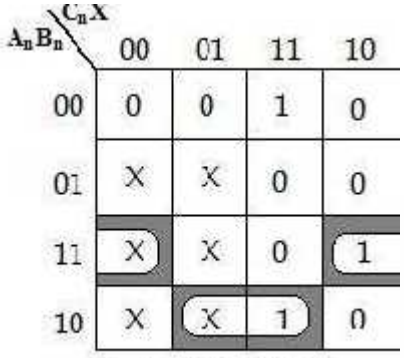


$B_{n+1} = D_B = \overline{A}_n \overline{B}_n X + \overline{A}_n B_n \overline{X}$



A_nB_n \ C_nX	00	01	11	10
00	1	1	1	1
01	X	X	1	1
11	X	X	0	0
10	X	X	0	0

$C_{n+1} = D_C = \overline{A}_n$



A_nB_n \ C_nX	00	01	11	10
00	0	0	1	0
01	X	X	0	0
11	X	X	0	1
10	X	X	1	0

$Z = A_nB_n\overline{X} + A_1\overline{B}_nX$

The state assignments using Rule 1 and 2 require:

4 three input AND functions

1 two input AND function

2 two input OR functions

---

7 gates with 18 inputs

Thus by simply applying Rules 1 and 2 good results have been achieved.