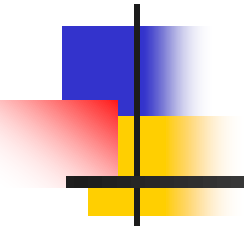
DIGITAL LOGIC DESIGN



UNIT-5

Finite State Machines

CONTENTS

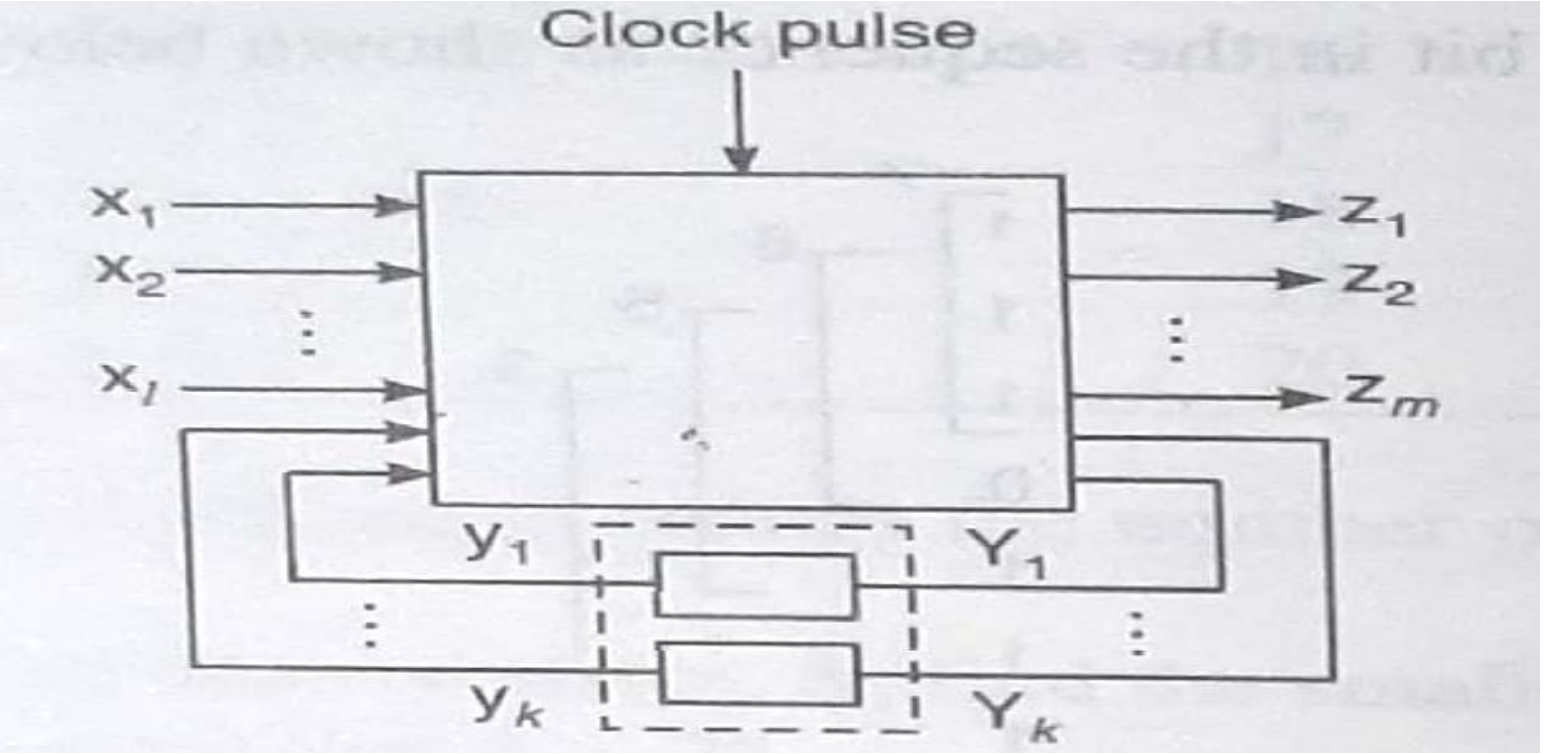
Analysis and Design of Synchronous Sequential Circuits:

- Moore and Mealy machine models, State Equations, State Table, State diagram, State reduction & assignment.
- Synthesis of synchronous sequential circuits- serial binary adder, sequence detector, and binary counter.
- Partition technique for completely specified sequential machines.

Definition of FSM

- An FSM is an abstract model describing the synchronous sequential machine and its spatial counter part, the iterative networks.
- The behavior of an FSM is described as a sequence of events that occurs at discrete instants of time t=1,2,3,….etc. That is, at time instant “ t ”, z(t) values depends on x(t) and past inputs of FSM.

FINITE STATE MACHINES



- Like this FSM might have infinite variety of possible histories, which would need an infinite capacity for storing them.

- Since in practice, it is impossible to implement machines which have infinite storage capabilities. Hence the concentration is only on those machines whose past histories can affect their future behavior in only number of ways. That is the FSM contains finite number of classes of input histories which are called as Internal States of a machine.
- Hence every FSM contains finite number of memory devices that store the information regarding the past input history.

Capabilities and Limitations of FSMs

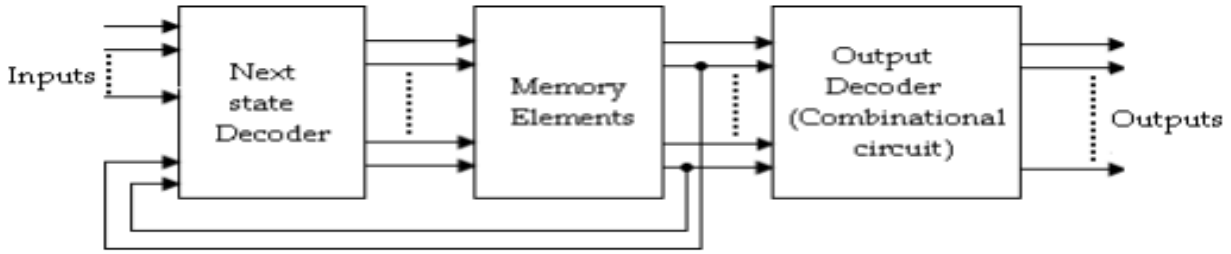
- Periodic Sequence of finite states.
- No Infinite Sequence
- Limited Memory

Representation of FSMs

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

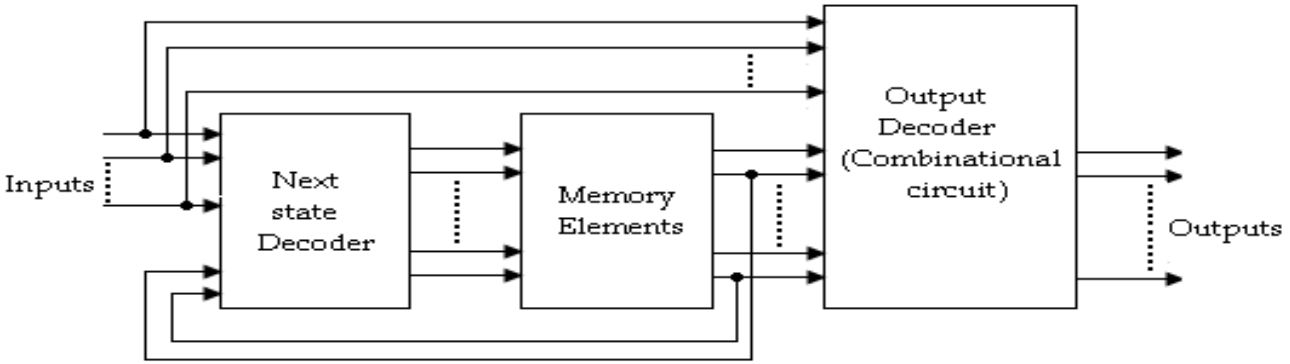
- Moore model: The output depends only on the present state of the Flip-Flops.
- Mealy model: The output depends on both the present state of the Flip-Flops and on the inputs.

Moore model: In the Moore model, the outputs are the functions 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 not 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.

Specification of FSM

- The relationship among Inputs (I/P), Present States (PS), Outputs (O/P) and Next States (NS) is done by:

STATE DIAGRAM

or

STATE TABLE

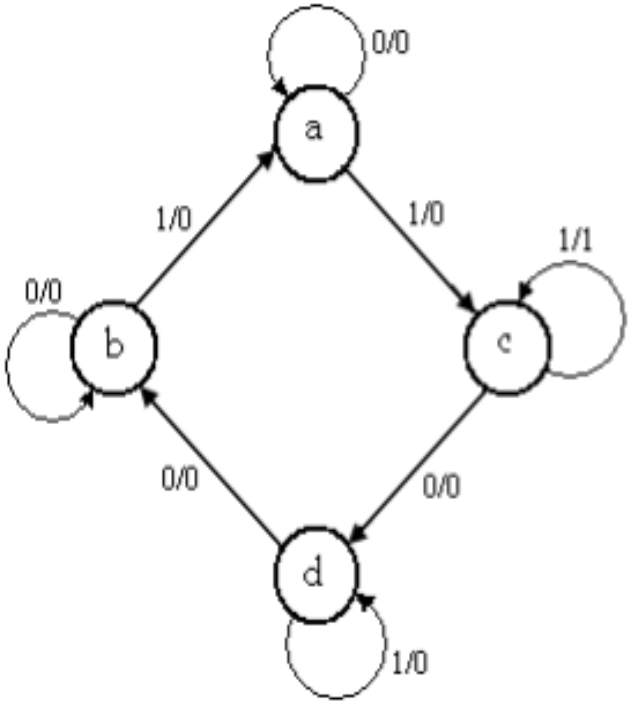
and

STATE EQUATION

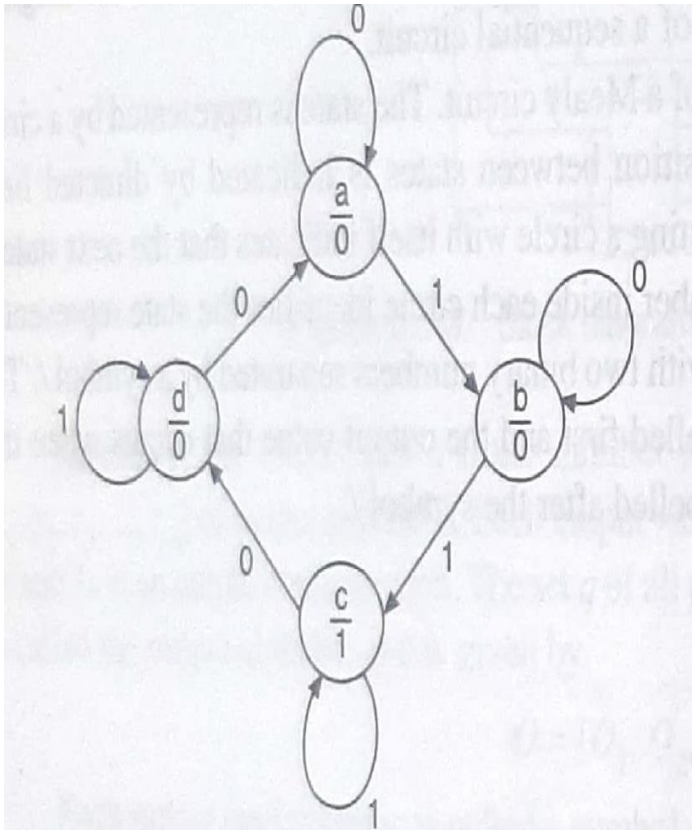
STATE DIAGRAM

- It is a pictorial representation of the 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 the input.

EXAMPLES OF STATE DIAGRAM



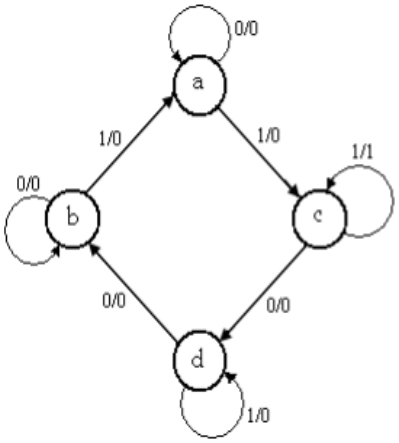
State diagram for Mealy circuit



Sate diagram for Moore circuit

STATE TABLE

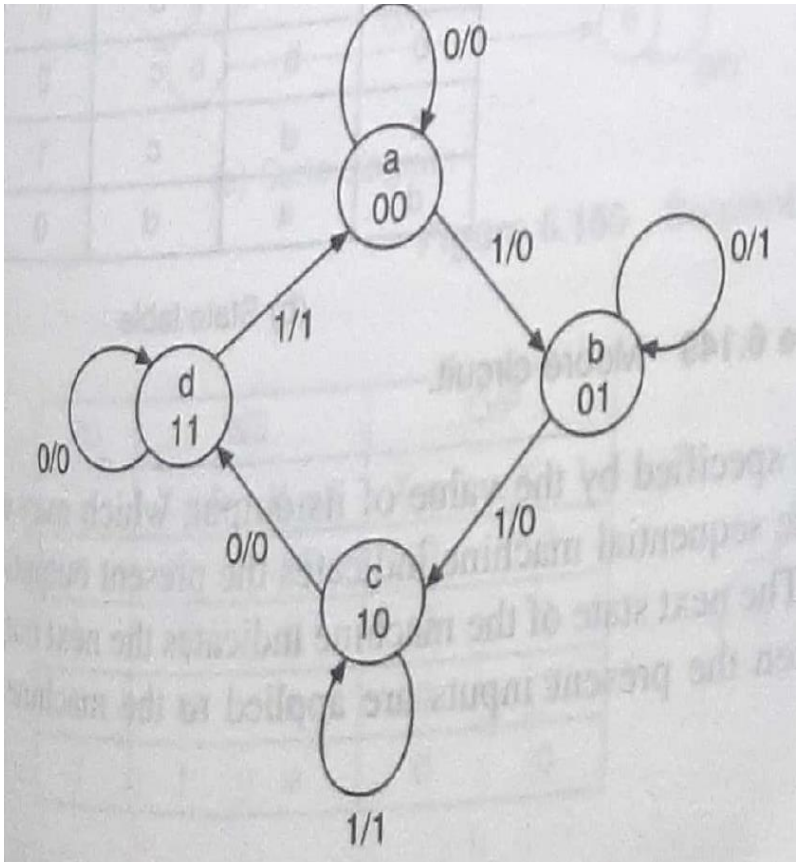
- It represents relationship between input, output and Flip-Flop states. It consists of three sections labeled present state, next state and output. The present state designates the state of Flip-Flops before the occurrence of a clock pulse, and the output section gives the values of the output variables during the present state. Both the next state and output sections have two columns representing for two possible input conditions: X= 0 and X=1.



State diagram for Mealy circuit

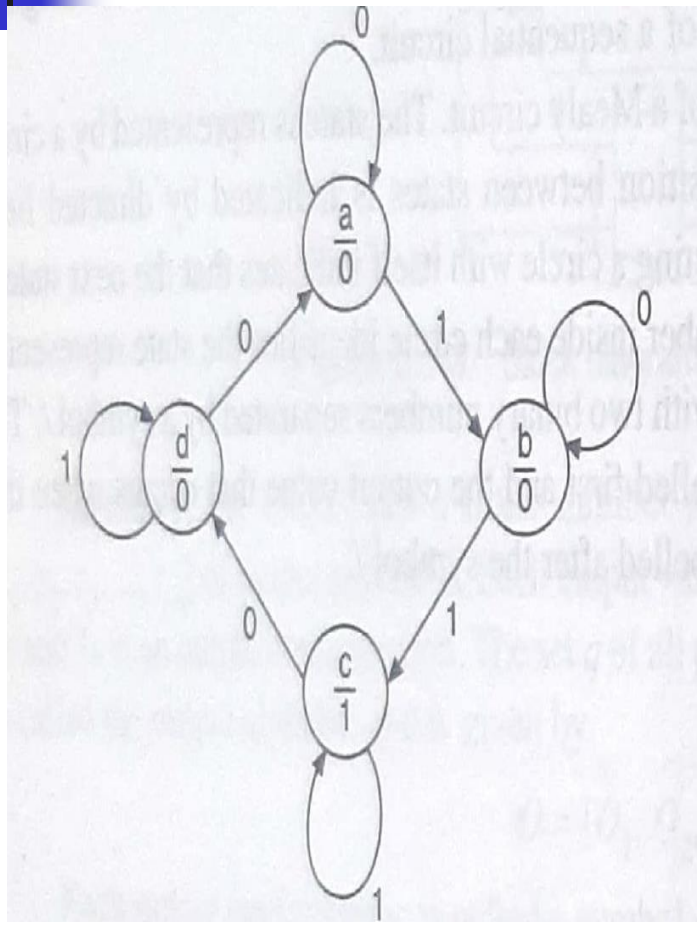
Present state	Next state		Output	
	X= 0	X= 1	X= 0	X= 1
			Y	Y
a	a	c	0	0
b	b	a	0	0
c	d	c	0	1
d	b	d	0	0

STATE TABLE



PS	NS, O/P	
	Input X	
	X = 0	X = 1
a	a, 0	b, 0
b	b, 1	c, 0
c	d, 0	c, 1
d	d, 0	a, 1

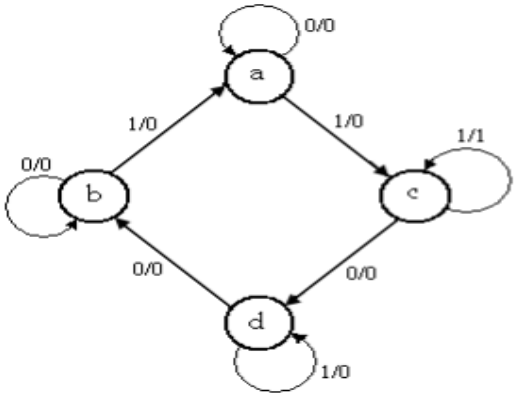
STATE TABLE



PS	NS / Input X		O/P
	X = 0	X = 1	
a	a	b	0
b	b	c	0
c	d	c	1
d	a	d	0

STATE TABLE

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



State diagram for Mealy circuit

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

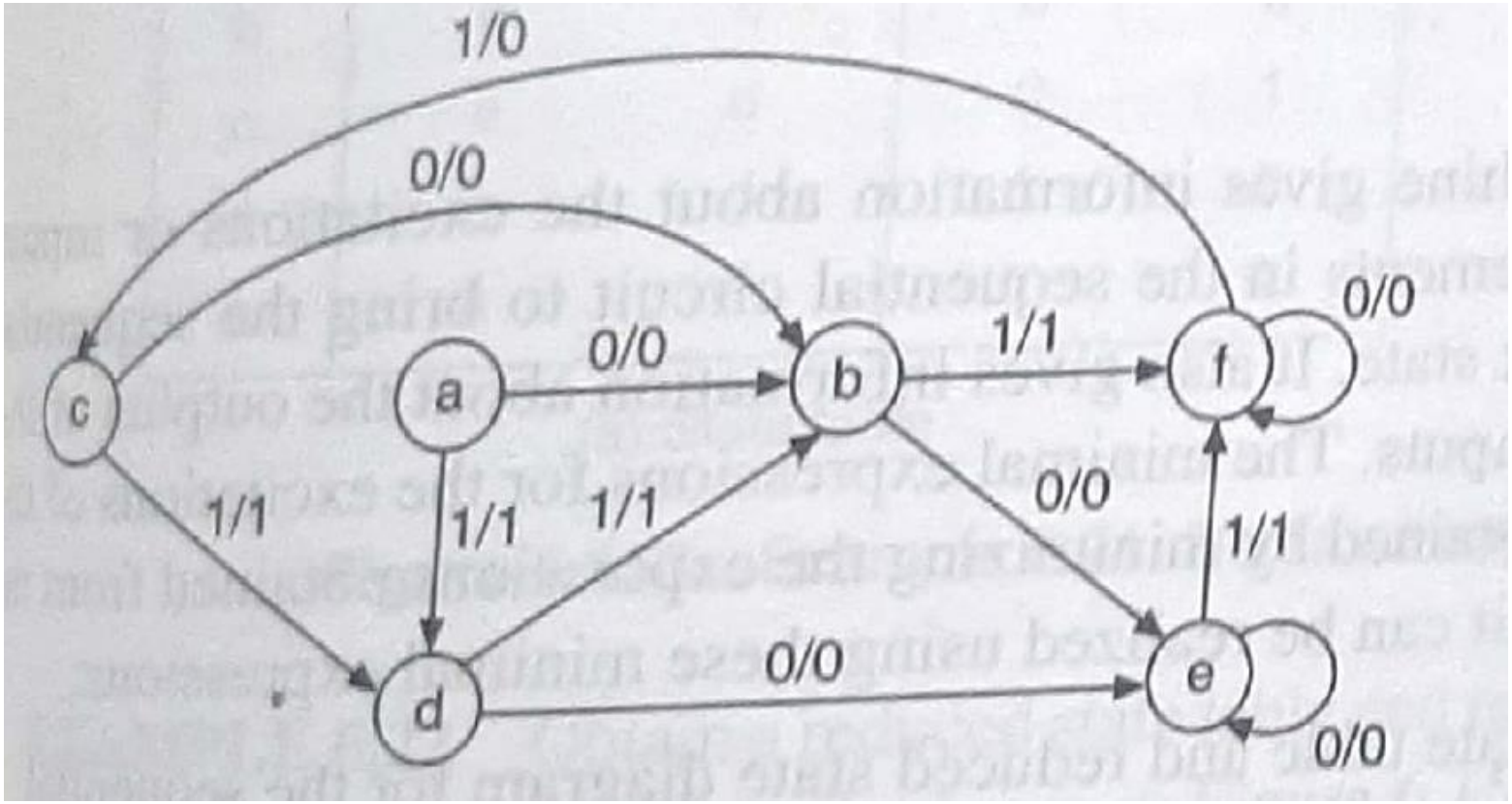
Successor: A's 1 successor is c and A's 0's successor is A(used in partition technique)

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.

State Reduction: If some states have same Next State and output then this state becomes a redundant state. This state can be removed. This process is known as State Reduction

State Assignment: In this process each state is assigned a binary value

STATE REDUCTION



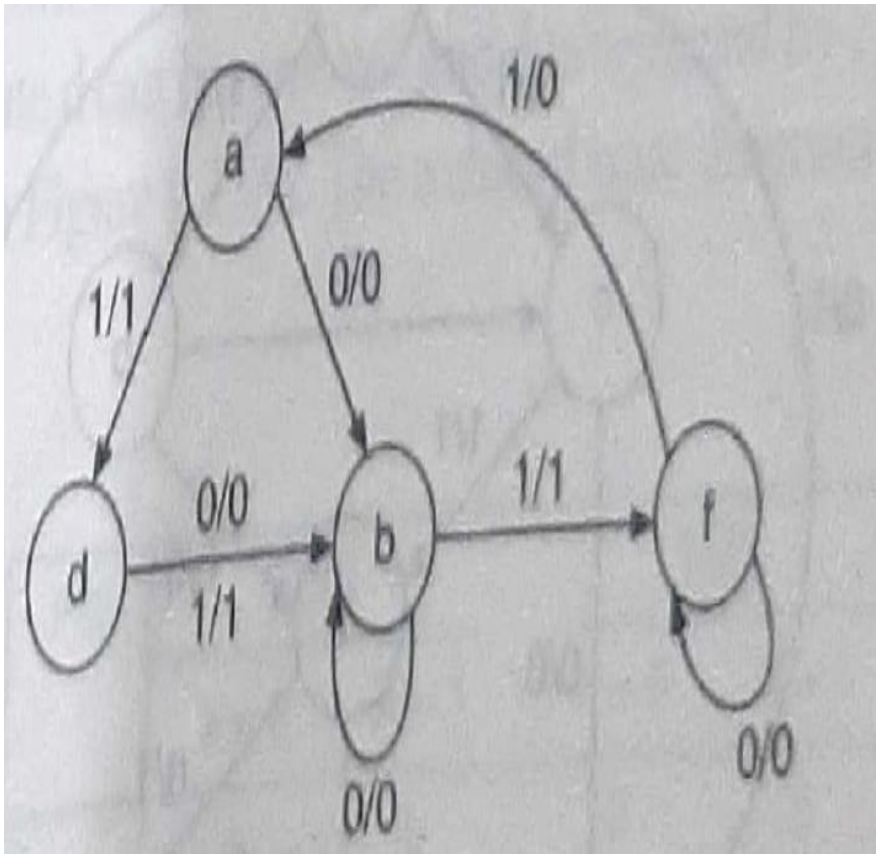
STATE REDUCTION

PS	NS		O/P	
	X = 0	X = 1	X = 0	X = 1
a	b	d	0	1
b	e	f	0	1
c	b	d	0	1
d	e	b	0	1
e	e	f	0	1
f	f	c	0	0

PS	NS		O/P	
	X = 0	X = 1	X = 0	X = 1
a	b	d	0	1
b	b	f	0	1
d	b	b	0	1
f	f	a	0	0

STATE REDUCTION

PS	NS		O/P	
	X = 0	X = 1	X = 0	X = 1
a	b	d	0	1
b	b	f	0	1
d	b	b	0	1
f	f	a	0	0



PARTITION TECHNIQUE

- This also helps in reducing no of States
- Example:

PS	NS, Z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

1. States having the same output under all input conditions can be grouped as

$$P_1 = (A, C, E)(B, D, F)$$

2. The 0- and 1-successors of (A, C, E), i.e. (E, E, C) and (D, B, F) are in the same block of P_1. 0-successors of (B, D, F), i.e. (F, F, B) are also in the same block of P_1. So, no partitioning is required, but 1-successors of (B, D, F), i.e. (D, B, C) are in different blocks of P_1. So, partition (B, D, F) into (B, D) and (F).

$$\therefore \quad P_2 = (A, C, E)(B, D)(F)$$

3. 0-successors of (A, C, E), i.e. (E, E, C) and the 0- and 1-successors of (B, D), i.e. (F, F) and (D, B) are in same blocks of P_2. So, no partitioning is required. The 1-successors of (A, C, E), i.e. (D, B, F) are in different blocks of P_2. So, partition (A, C, E) into (A, C) and (E).

$$\therefore \quad P_3 = (A, C)(E)(B, D)(F)$$

4. The 0- and 1-successors of (A, C), and (B, D) i.e. (E, E), (D, B) and (F, F), (D, B) are in the same blocks of P_3. So, no partitioning is required.

$$\therefore \quad P_4 = (A, C)(E)(B, D)(F)$$

Thus, equivalent states are

$$A = C \text{ and } B = D$$

So, states C and D are redundant and can be removed. C and D can be replaced by A and B respectively in the rest of the table. The resultant minimized state table is as shown in Table

PS	NS, Z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

PARTITION TECHNIQUE

Reduced State Table:

PS	NS, Z	
	X = 0	X = 1
A	E, 0	B, 1
B	F, 0	B, 0
E	A, 0	F, 1
F	B, 0	A, 0

PS	NS, Z	
	X = 0	X = 1
A	E, 0	D, 1
B	F, 0	D, 0
C	E, 0	B, 1
D	F, 0	B, 0
E	C, 0	F, 1
F	B, 0	C, 0

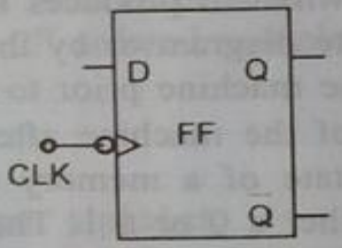
STATE DIAGRAM OF D FLIP-FLOP:

Circuit change From	To	Required input
Q(t)	Q(t+1)	D(t)
0	0	0
0	1	1
1	0	0
1	1	1

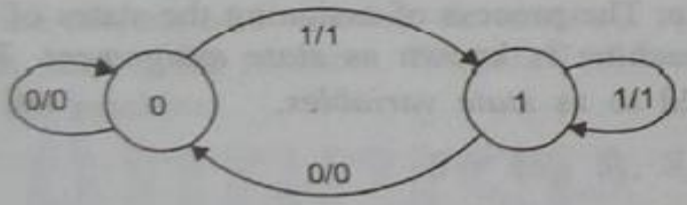
(a) Excitation requirements

Present state (PS)	Input to FF	Next state (NS)
Q(t)	D(t)	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

(b) Excitation table



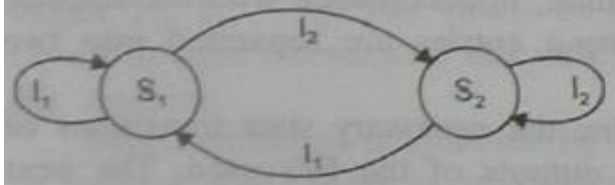
(c) Logic symbol



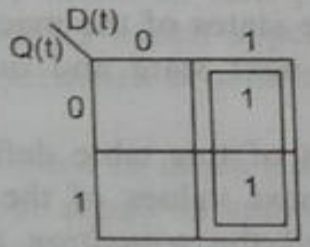
(d) State diagram

PS	N/S, O/P	
	D = 0	D = 1
0	0, 0	1, 1
1	0, 0	1, 1

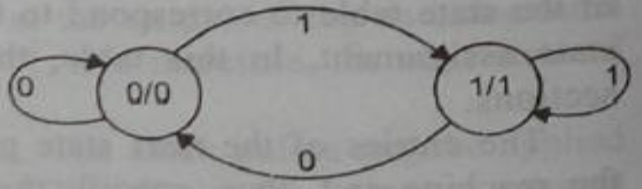
(e) State table



(f) General state diagram



(g) K-map for Q(t+1)



(h) Moore model

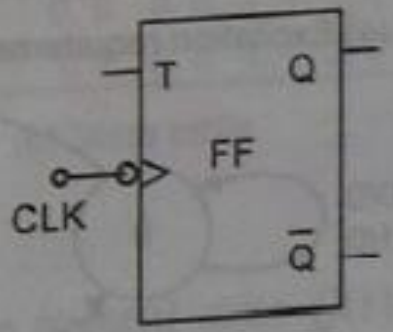
STATE DIAGRAM OF T FLIP-FLOP:

Circuit change From	To	Required input
Q(t)	Q(t+1)	T(t)
0	0	0
0	1	1
1	0	1
1	1	0

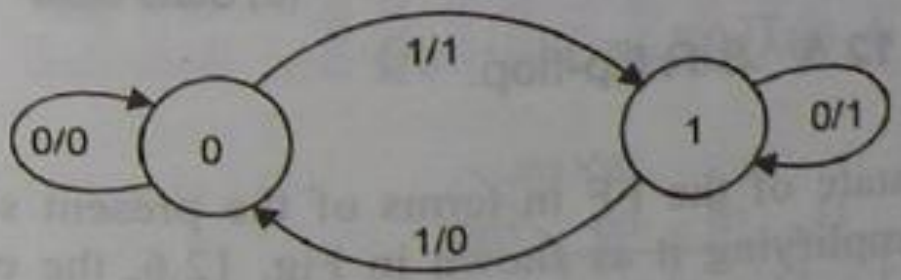
(a) Excitation requirements

Present state (PS)	Input to FF	Next state (NS)
Q(t)	T(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

(b) Excitation table



(c) Logic symbol



(d) State diagram

PS	NS, O/P	
	T = 0	T = 1
0	0, 0	1, 1
1	1, 1	0, 0

(e) State table

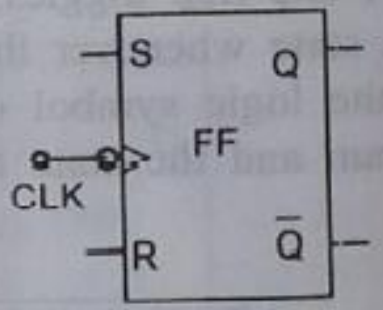
STATE DIAGRAM OF SR FLIP-FLOP:

Circuit change From Q(t)	To Q(t+1)	Required inputs S(t)	R(t)
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

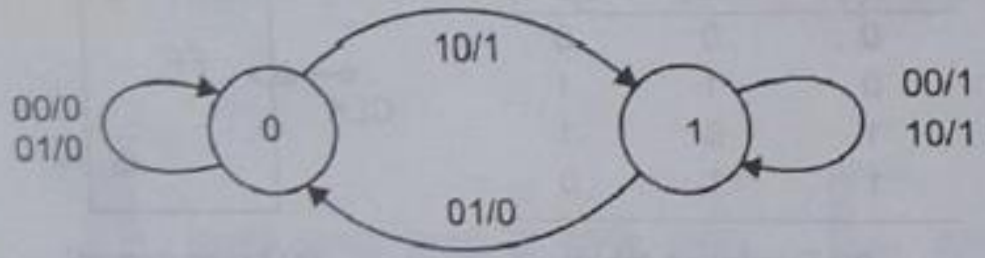
(a) Excitation requirements

Present state (PS) Q(t)	Inputs to FF S(t)	R(t)	Next state (NS) Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	1

(b) Excitation table



(c) Logic symbol



(d) State diagram

PS	NS, O/P S R = 0 0	S R = 0 1	S R = 1 0
0	0, 0	0, 0	1, 1
1	1, 1	0, 0	1, 1

(e) State table

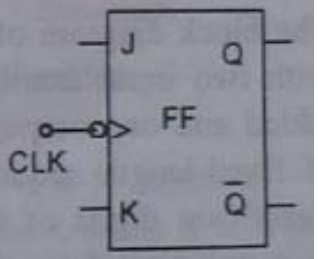
STATE DIAGRAM OF JK FLIP-FLOP:

Circuit change From	To	Required inputs	
Q(t)	Q(t+1)	J(t)	K(t)
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

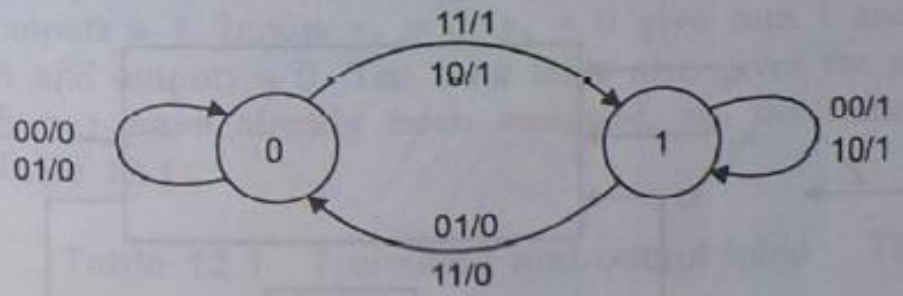
(a) Excitation requirements

Present state (PS)	Inputs to FF		Next state (NS)
Q(t)	J(t)	K(t)	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(b) Excitation table



(c) Logic symbol

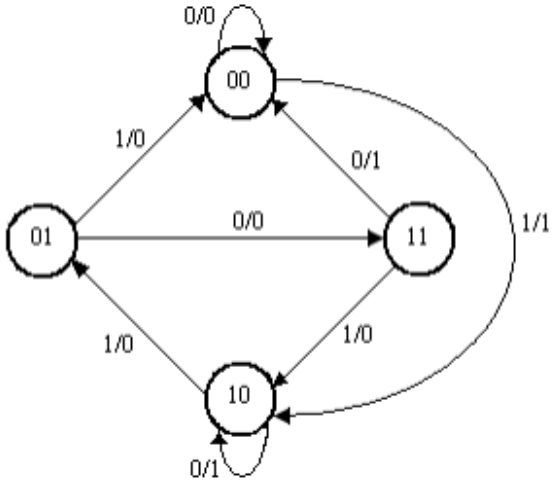


(d) State diagram

PS	NS, O/P							
	J	K	J	K	J	K	J	K
	0	0	0	1	1	0	1	1
0	0, 0		0, 0		1, 1		1, 1	
1	1, 1		0, 0		1, 1		0, 0	

(e) State table

DESIGN A CIRCUIT FOR A GIVEN STATE DIAGRAM USING D FLIP-FLOP:



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 Table

DESIGN A CIRCUIT FOR A GIVEN STATE DIAGRAM USING 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:

For Flip-flop A

A \ BX	00	01	11	10
0	0	(1)	0	(1)
1	(1)	0	(1)	0

D_A= A'B'X + A'BX' + ABX + AB'X'
= A⊕ (B ⊕ x)

For Flip-flop B

A \ BX	00	01	11	10
0	0	0	0	(1)
1	0	(1)	0	0

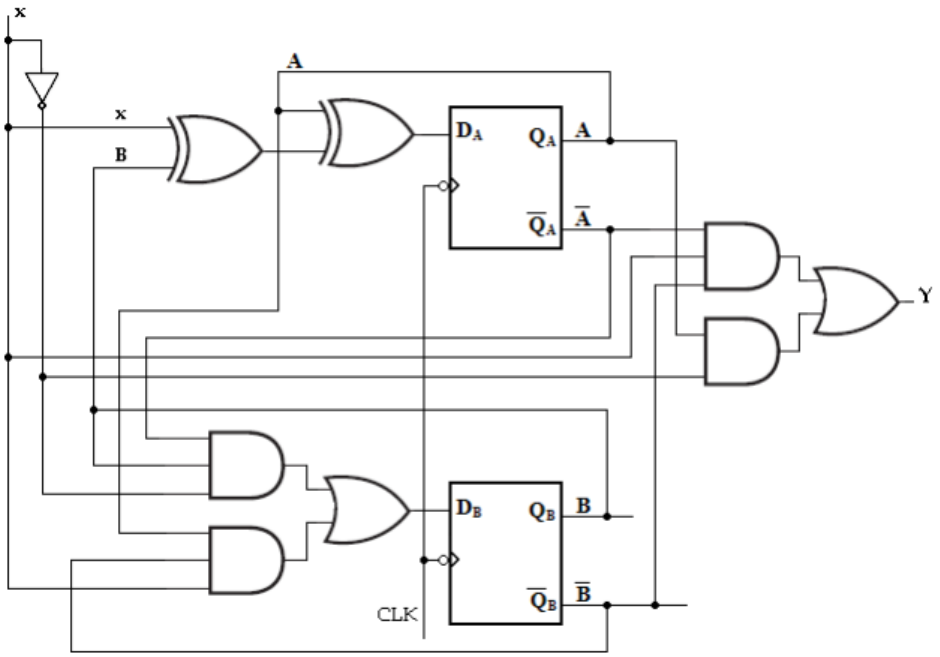
D_B= A'BX' + AB'X

For Output

A \ BX	00	01	11	10
0	0	(1)	0	0
1	(1)	0	0	(1)

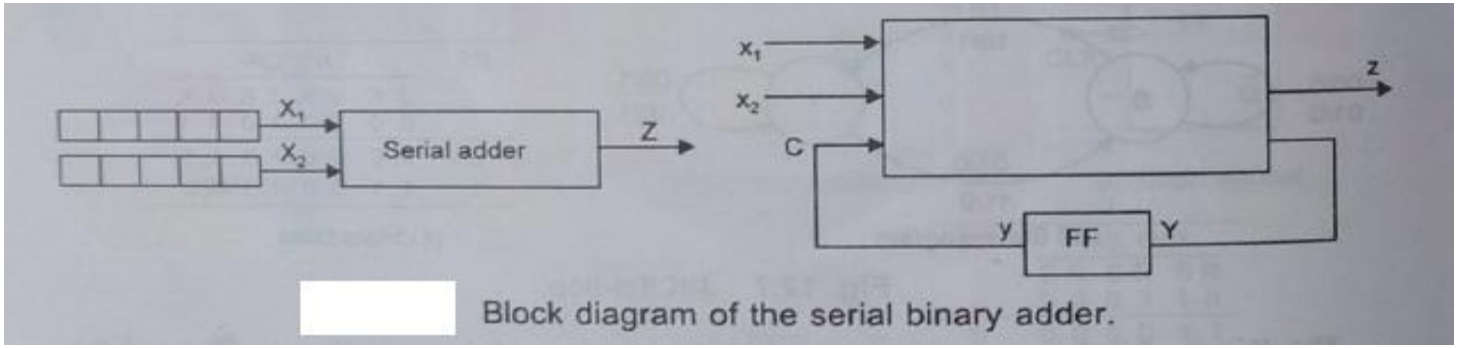
Y = A'B'X + AX'

DESIGN A CIRCUIT FOR A GIVEN STATE DIAGRAM USING D FLIP-FLOP:

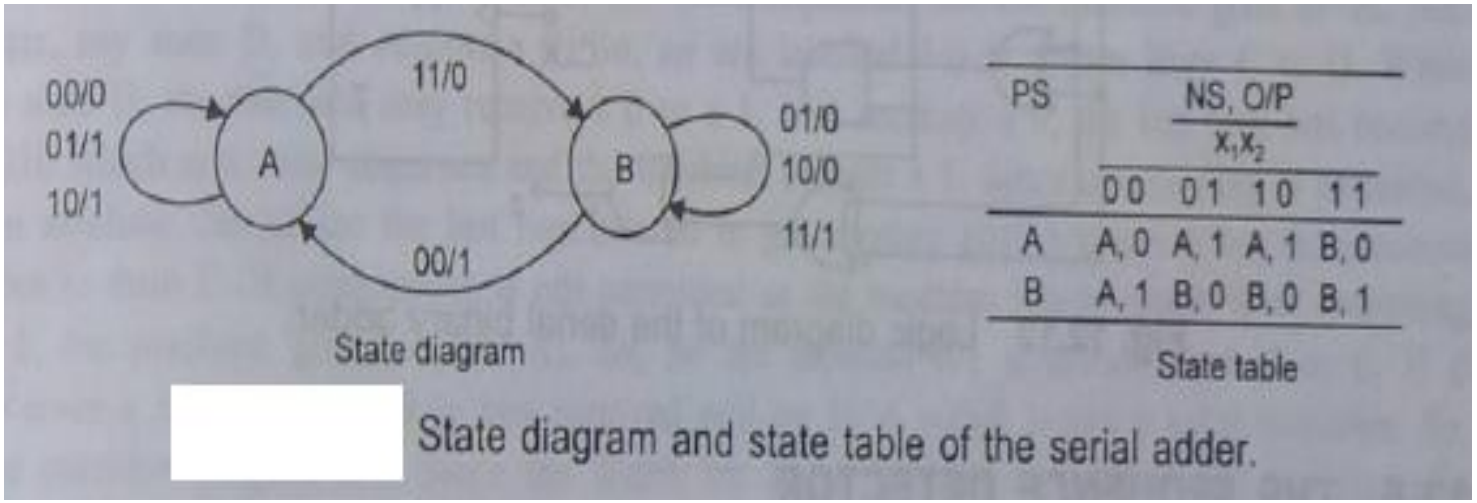


Logic diagram of given sequential circuit using D Flip-Flop

DESIGN A SERIAL BINARY ADDER FOR MEALY TYPE



Block diagram of the serial binary adder.



State diagram

PS	NS, O/P X_1X_2 = 00	01	10	11
A	A, 0	A, 1	A, 1	B, 0
B	A, 1	B, 0	B, 0	B, 1

State table

State diagram and state table of the serial adder.

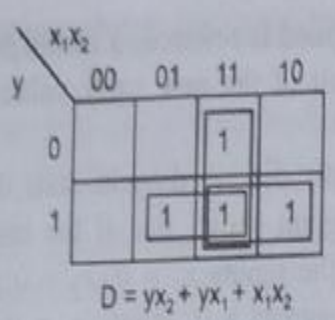
DESIGN A SERIAL BINARY ADDER FOR MEALY TYPE

Transition and output table

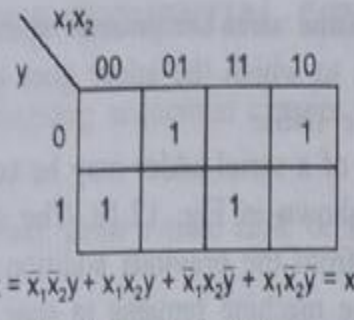
PS	NS x_1x_2				O/P x_1x_2			
	00	01	10	11	00	01	10	11
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

Excitation table

PS	I/P		NS	I/P to FF	O/P
y	x_1	x_2	y	D	z
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

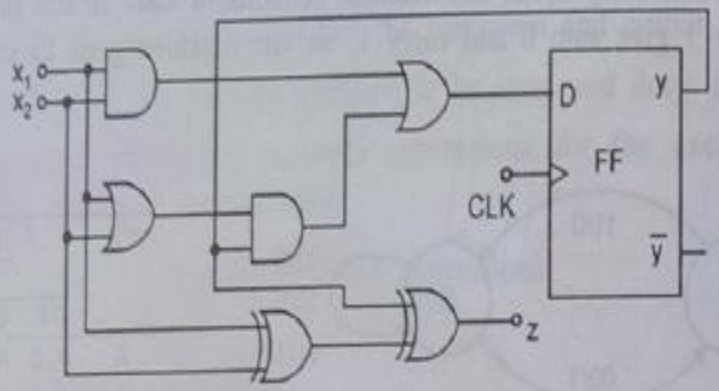


$D = yx_2 + yx_1 + x_1x_2$



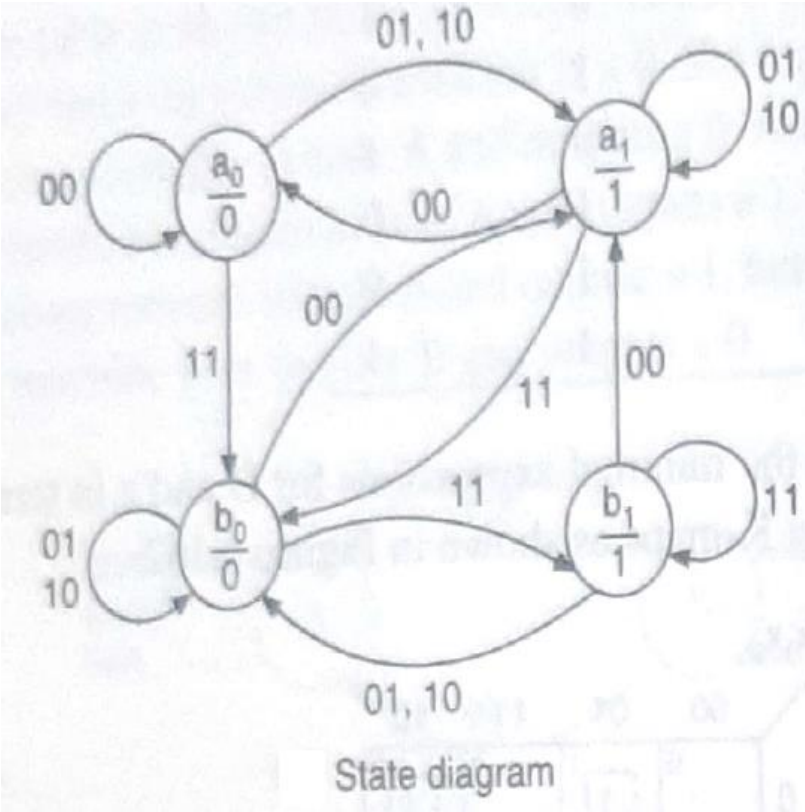
$z = \bar{x}_1\bar{x}_2y + x_1x_2y + \bar{x}_1x_2\bar{y} + x_1\bar{x}_2\bar{y} = x_1 \oplus x_2 \oplus y$

K-maps for the serial adder.



Logic diagram of the serial binary adder.

DESIGN A SERIAL BINARY ADDER FOR MOORE TYPE



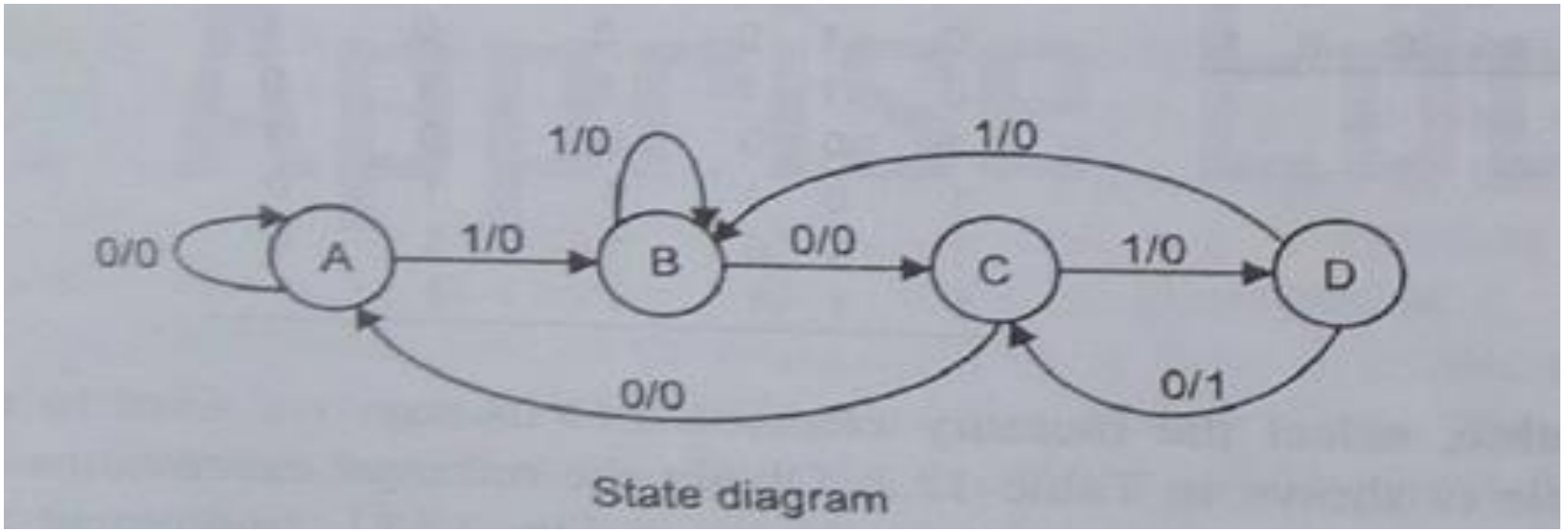
State diagram

PS	NS / Input				O/P (sum)
	y = 00	01	10	11	
a_0	a_0	a_1	a_1	b_0	0
a_1	a_0	a_1	a_1	b_0	1
b_0	a_1	b_0	b_0	b_1	0
b_1	a_1	b_0	b_0	b_1	1

State table

DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1010 FOR MEALY TYPE

- If the input Sequence is 01101010 then the output is 00000101
- First draw the state diagram to detect the Sequence 1010(consider overlapping).
- When the Sequence is completed the output will be 1.
- First State table, Excitation table, Boolean Expressions, Logic diagram



State diagram

DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1010 FOR MEALY TYPE

PS	NS, z	
	x = 0	x = 1
A	A, 0	B, 0
B	C, 0	B, 0
C	A, 0	D, 0
D	C, 1	B, 0

State table

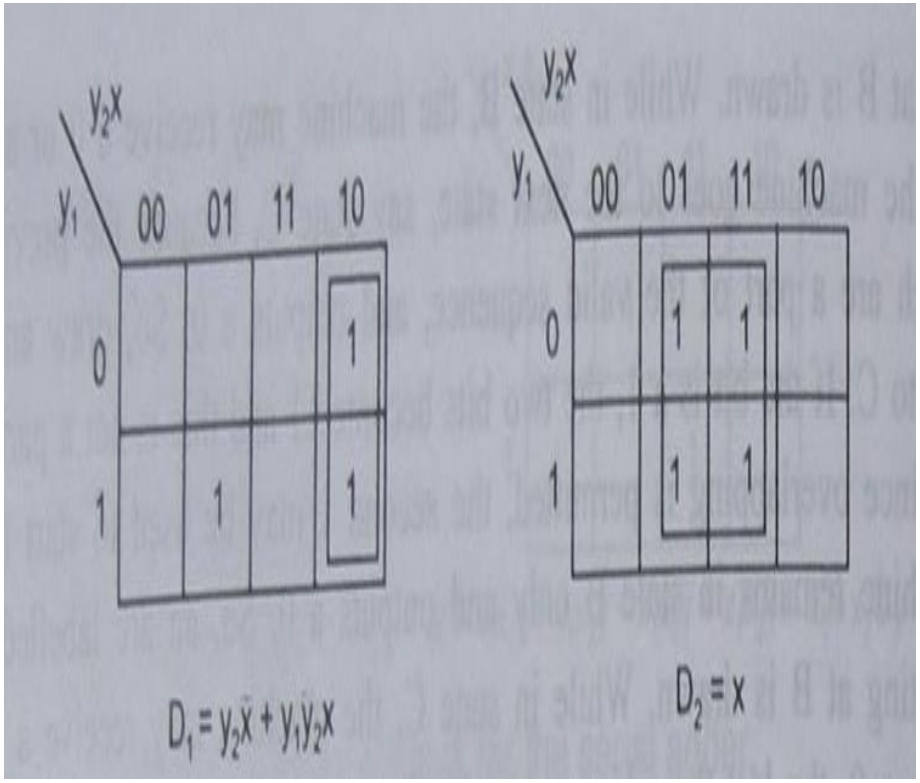
Transition and output table

PS	NS		O/P	
	x = 0	x = 1	x = 0	x = 1
A→00	00	01	0	0
B→01	10	01	0	0
C→10	00	11	0	0
D→11	10	01	1	0

DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1010 FOR MEALY TYPE

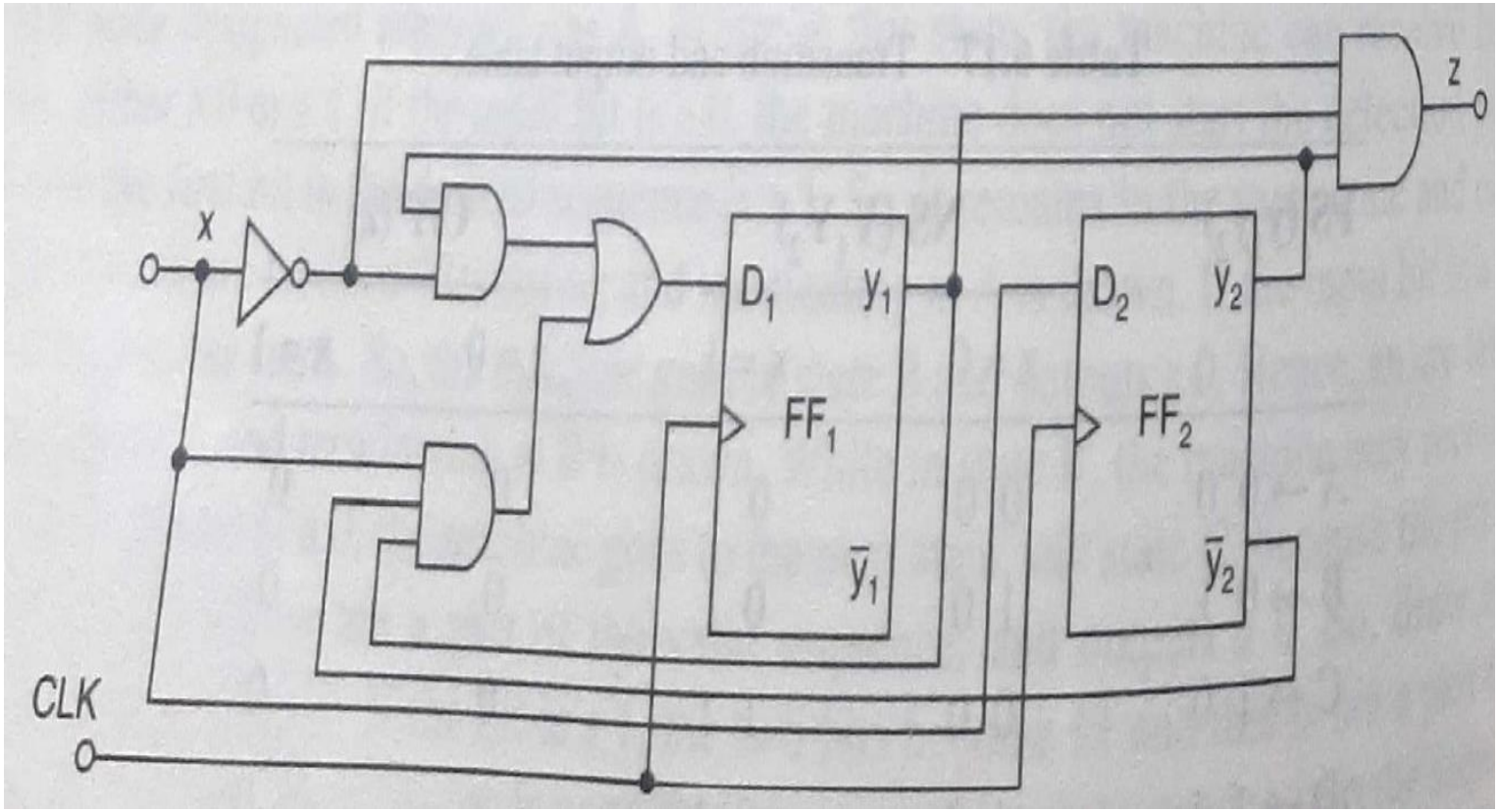
Excitation table

PS y_1	PS y_2	I/P x	NS Y_1	NS Y_2	I/P to FFs D_1	I/P to FFs D_2	O/P 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	1
1	1	1	0	1	0	1	0



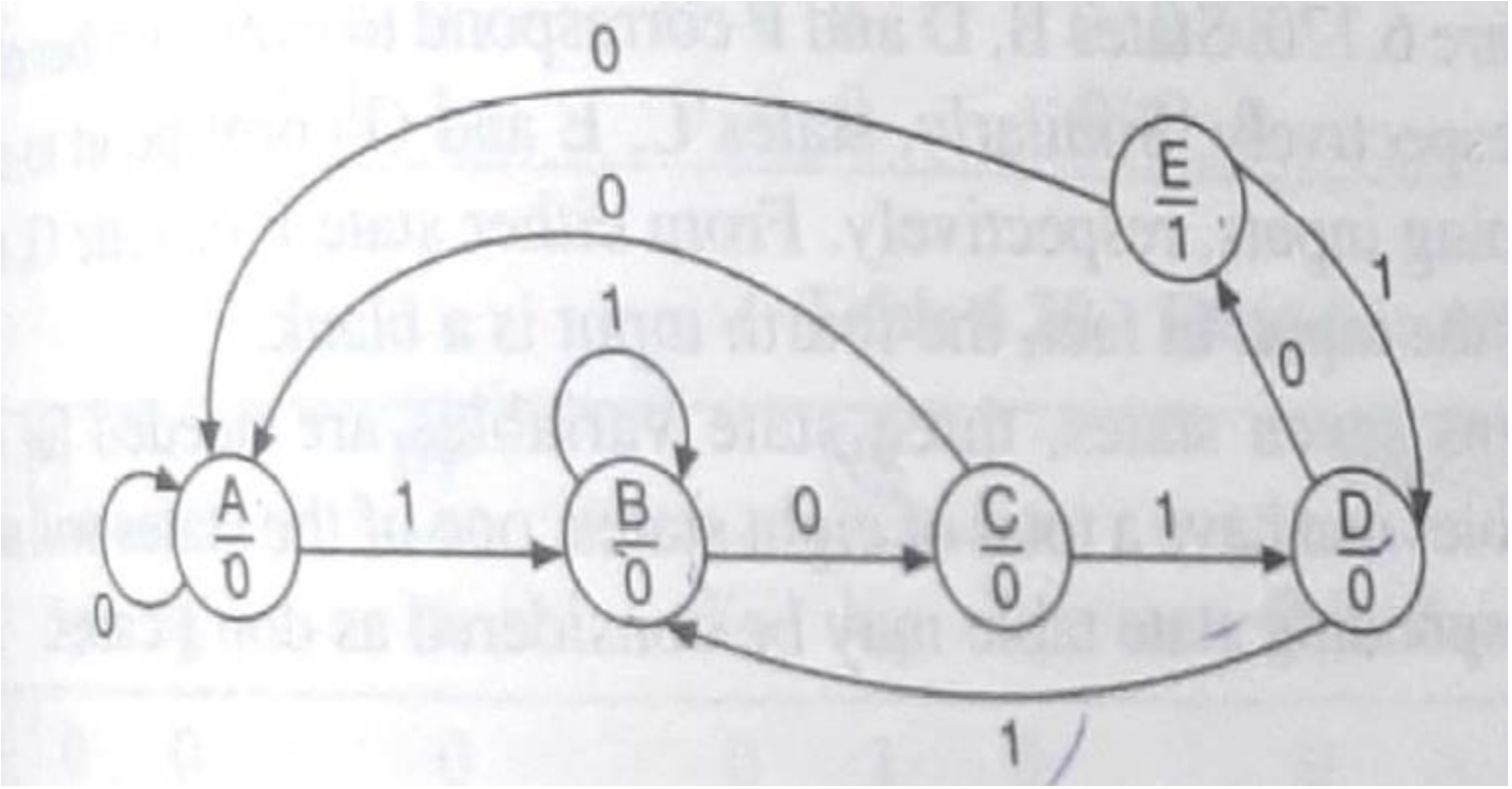
$Z = Y_1 Y_2 X'$

DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1010 FOR MEALY TYPE



Logic Diagram

DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1010 FOR MOORE TYPE

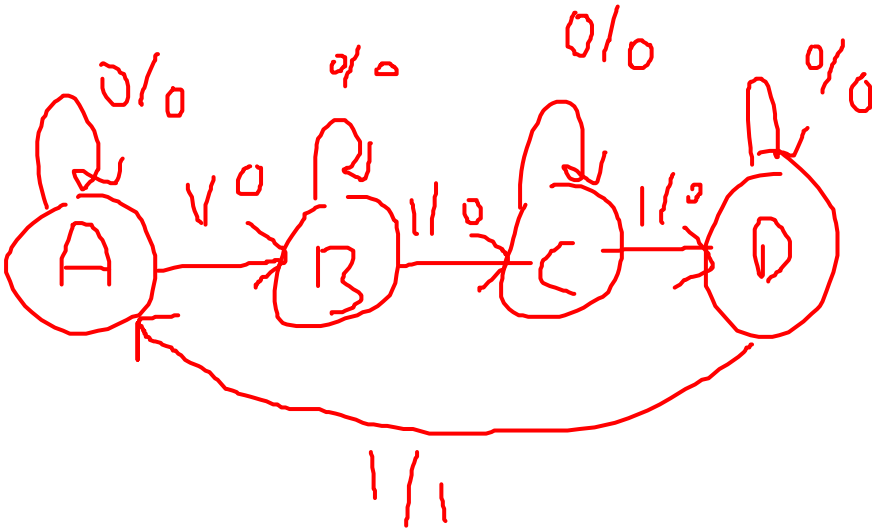


DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1010 FOR MOORE TYPE

PS	NS		O/P
	X = 0	X = 1	
A	A	B	0
B	C	B	0
C	A	D	0
D	E	B	0
E	A	D	1

State Table and the remaining Steps are same

DESIGN A SEQUENCE DETECTOR TO DETECT SEQUENCE 1111 FOR MEALY TYPE



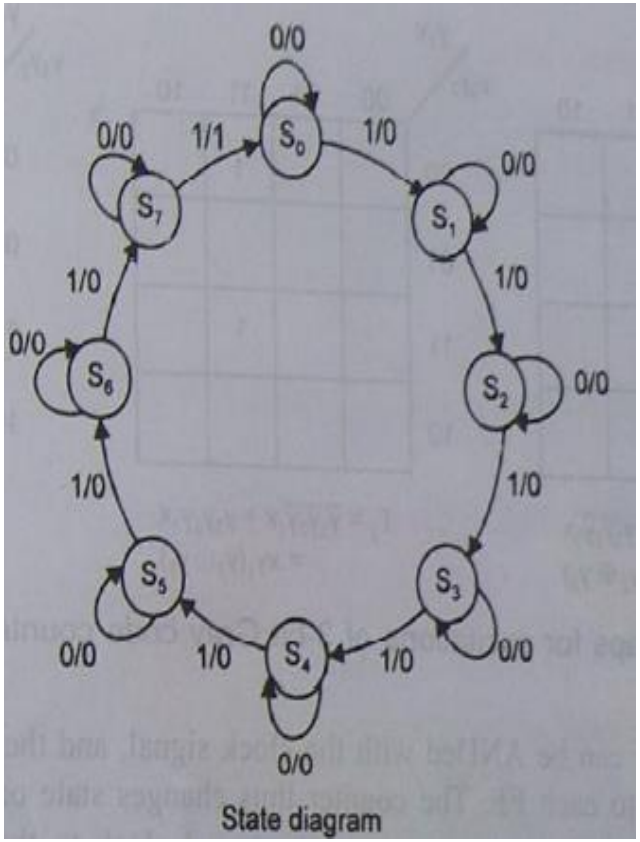
PS	NS,Z	
	A=0	A=1
A	A,0	B,0
B	B,0	C,0
C	C,0	D,0
D	D,0	A,1

State Diagram and State Table and the remaining Steps are same (Non Overlapping Sequence)

DESIGN A THREE BIT GRAY CODE COUNTER

State assignment	
S_0	0 0 0
S_1	0 0 1
S_2	0 1 1
S_3	0 1 0
S_4	1 1 0
S_5	1 1 1
S_6	1 0 1
S_7	1 0 0

State assignment table



State diagram

PS	NS, O/P	
	x = 0	x = 1
S_0	S_0, 0	S_1, 0
S_1	S_1, 0	S_2, 0
S_2	S_2, 0	S_3, 0
S_3	S_3, 0	S_4, 0
S_4	S_4, 0	S_5, 0
S_5	S_5, 0	S_6, 0
S_6	S_6, 0	S_7, 0
S_7	S_7, 0	S_0, 0

State table

DESIGN A THREE BIT GRAY CODE COUNTER

Transition and output table

PS	NS		O/P	
	x = 0	x = 1	x = 0	x = 1
000	000	001	0	0
001	001	011	0	0
011	011	010	0	0
010	010	110	0	0
110	110	111	0	0
111	111	101	0	0
101	101	100	0	0
100	100	000	0	1

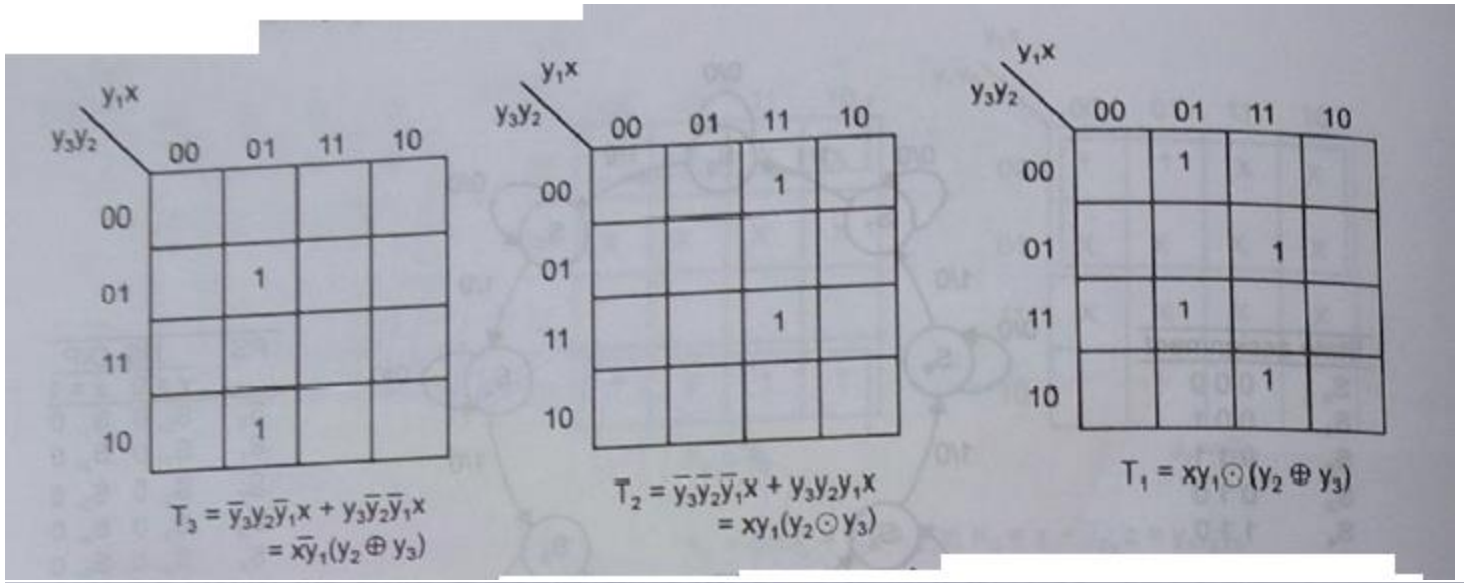
Excitation table

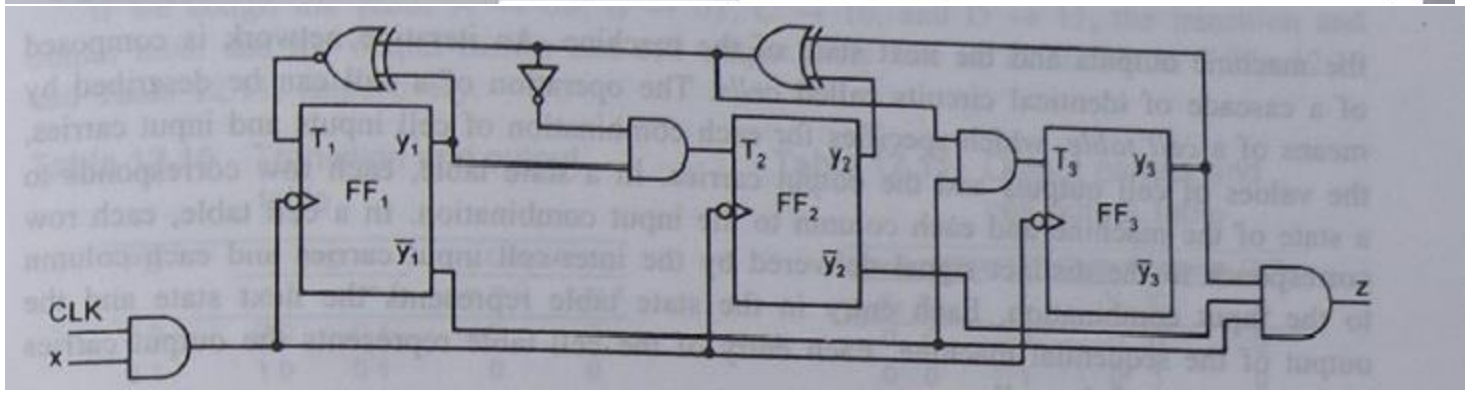
PS			I/P	NS			Inputs to FFs			O/P
y_3	y_2	y_1	x	Y_3	Y_2	Y_1	T_3	T_2	T_1	z
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0	0	0	0
0	0	1	1	0	1	1	0	1	0	0
0	1	1	0	0	1	1	0	0	0	0
0	1	1	1	0	1	0	0	0	1	0
0	1	0	0	1	1	0	0	0	0	0
0	1	0	1	1	1	0	1	0	0	0
1	1	0	0	1	1	0	0	0	0	0
1	1	0	1	1	1	1	0	0	1	0
1	1	1	0	1	1	1	0	0	0	0
1	1	1	1	1	0	1	0	1	0	0
1	0	1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	1

$$z = y_3 \bar{y}_2 \bar{y}_1 x$$

DESIGN A THREE BIT GRAY CODE COUNTER

K-Maps and Logic Diagram





References

- Kumar, A. Anand. *Switching Theory and Logic Design*. PHI Learning Pvt. Ltd., 2014.
- Kumar, A. Anand. *Fundamentals of Digital Circuits* PHI Learning Pvt. Ltd., 2014.
- Videos from NESCO Academy

THANK YOU