

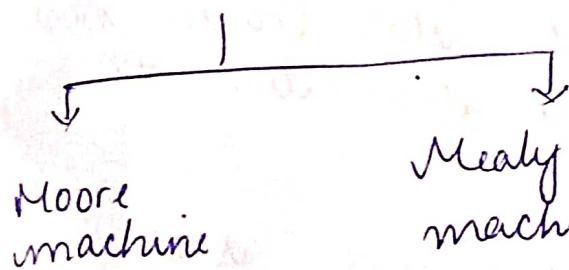
UNIT -5

Finite State machine (F.S Model)

- An abstract model describing synchronous sequential machine
- Its behaviour is described as a sequence of events that occurs at discrete intervals of time.
- Also called finite Automata (FA)
- Broadly divided into 2 types

F.A with output \oplus

Finite automata without O/P



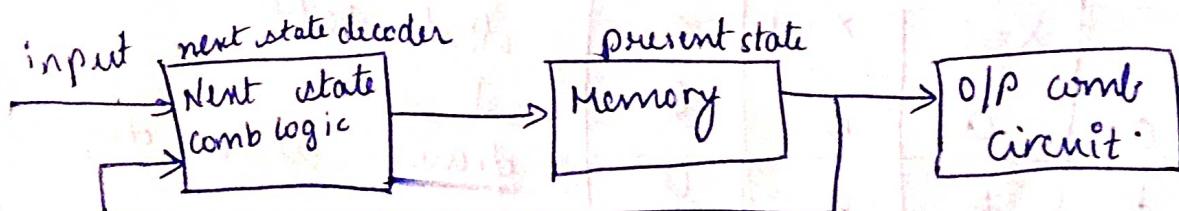
[They differ by the way o/p is generated]

Moore machine: [Moore state machine].

- Output is the function of present state only.

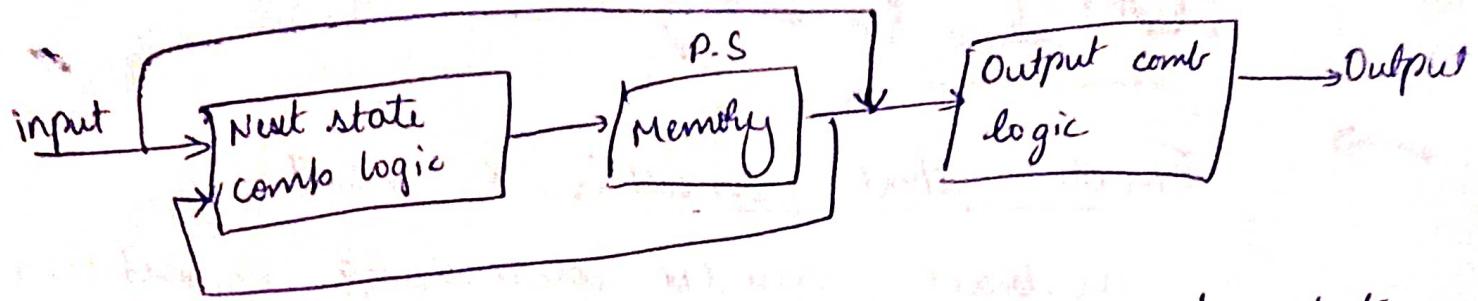
Mealy state machine

Output is the function of present state & input



O/P is func of P.S so \rightarrow

moore.



O/P is function of input also \rightarrow mealy state machine

Moore Machine

State Diagram

- pictorial representation of relationships b/w present state & input, next state, output of sequential circuit.
- circle/ vertex represent state (called node)
- lines b/w vertices are transitions or states
- binary number on circle represents state
- number on directed lines indicate the input.
- number inside state represents o/p.

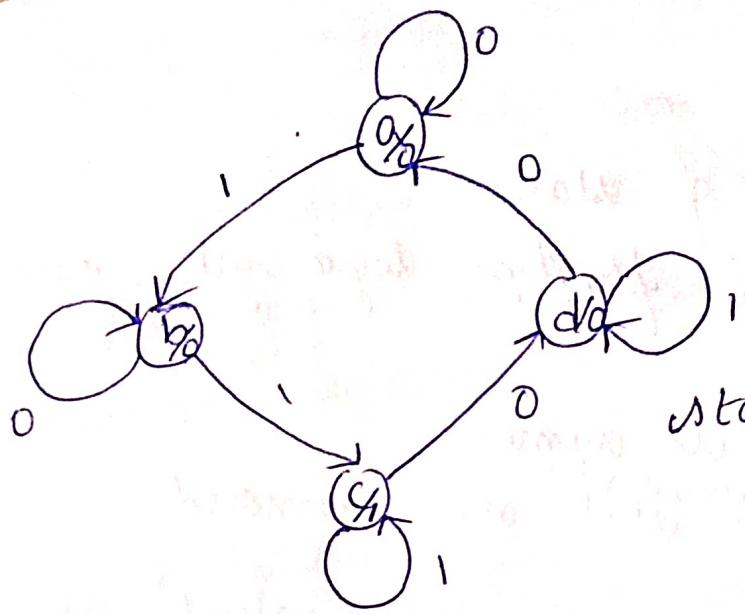
PS	NS	
	Input x.	NS
	x=0	x=1
a	a	b
b	b	c
c	d	c
d	a	d

state table

↓
draw state diagram

we can denote

a	00
b	01
c	10
d	11

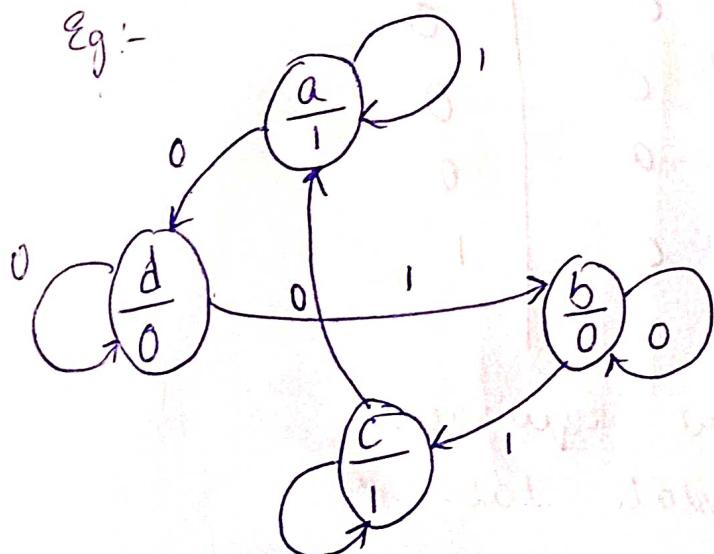


state diagram

State table

- even though the info of state diagram is convenient we need to translate the state diagram to state table to implement information.
- both S.T & SD contain same info.

Eg:-



Draw state table

PS	Input X		NS
	$x = 0$	$x = 1$	
a	d	a	1
b	b	c	0
c	a	c	1

State Reduction

- Reducing of no. of states
- Reduces no. of flipflops, logic gates, cost of final circuit.
- Output will be same.
- Only same ^(equivalent) states are removed.
- Eg: If 2 states are equivalent without altering input & output.

Eg:-

PS	NS		Output
	$x=0$	$x=1$	
a	d	c	0
b	d	c	0
c	d	a	0
d	d	c	1

a & b states are equivalent
 \therefore reduced state table is

PS	NS		Output
	$x=0$	$x=1$	
a	d	c	0
c	d	a	0
d	d	c	1

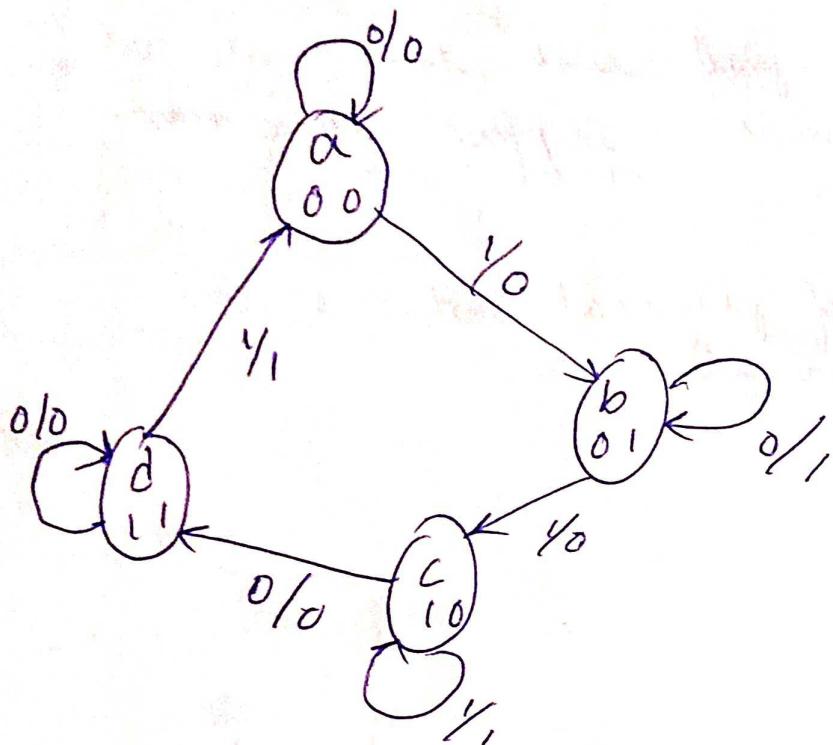
State assignment

- process of assigning states of a sequential machine
- binary values to machine → state assignment
- the values are assigned such that it is possible to implement flipflop using min. logic gates.
- Output values of physical devices are called state variables.

Rules

Mealy

State diagram & State table



a is i's successor of b, d
b is i's successor of a.
d is o's successor of c, d.

state table

PS	NS		O/p	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	1
b	b	c	1	0
c	d	c	0	1
d	d	a	0	1

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	f	e	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	e	f	0	1
d	e	b	0	1
f	f	c	0	0

Memory Elements

① D flip flop

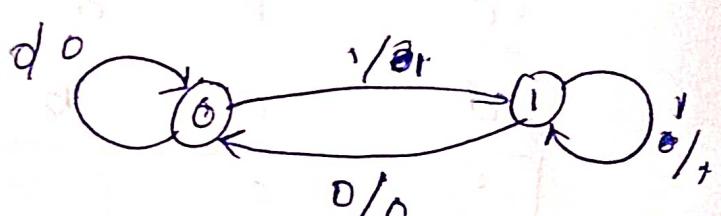


② Excitation table

PS Inp NS.

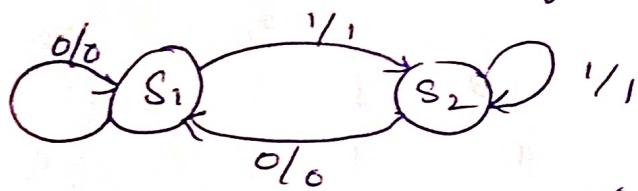
Q _n	D	Q _{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

③ State diagram.



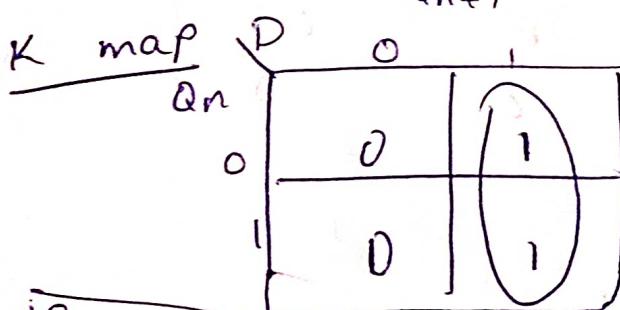
(mealy)

General state diagram



state table.

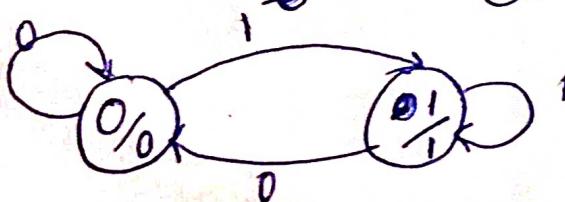
PS	NS	Out
$x=0$	$x=1$	$x=0$ $x=1$
S ₁	S ₁	0 1
S ₂	S ₁	0 1



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

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

moore



Partition technique

	NS	
PS	$x=0$	$x=1$
A	E, 0	<u>D, 1</u>
B	F, 0	D, 0
C	E, 0	<u>B, 1</u>
D	F, 0	B, 0
E	C, 0	<u>F, 1</u>
F	B, 0	C, 0

$$P_1 \rightarrow \text{outputs} \\ (ACE) (BDF)$$

$$P_2 = \begin{cases} x=0 & \rightarrow \text{no partition in ACE, BDF} \\ x=1 & \rightarrow \text{no } " " " " \text{, partition in BDF} \end{cases}$$

$$P_2 = (ACE) (BD)(F)$$

$$P_3 = \begin{cases} x=0 & \rightarrow \text{no par.} \\ x=1 & \rightarrow (AC)(E) \end{cases}$$

$$\begin{array}{c} A \downarrow C \downarrow E \\ E \downarrow E \downarrow C \\ B \downarrow D \downarrow F \\ F \downarrow F \downarrow B \end{array} \checkmark$$

$$\begin{array}{c} B \downarrow D \downarrow F \\ F \downarrow F \downarrow B \end{array} \checkmark$$

$$\begin{array}{c} A \downarrow C \downarrow E \\ \downarrow \downarrow \downarrow \\ D \downarrow B \downarrow F \end{array} \checkmark$$

belong to same set.

$$\begin{array}{c} B \downarrow O \downarrow F \\ \downarrow \downarrow \downarrow \\ D \downarrow B \downarrow C \end{array}$$

$$x=0 \quad |$$

$$\begin{array}{c} ACE \\ \downarrow \downarrow \downarrow \\ B \downarrow F \\ \swarrow \searrow \\ 1 \quad 1 \end{array}$$

$$\begin{array}{c} BD \\ \downarrow \downarrow \\ B \downarrow B \end{array} \checkmark$$

$$\begin{array}{c} F \\ \downarrow \\ C \end{array} \checkmark$$

$$\begin{array}{c} x=0 \\ ACE \quad \checkmark \quad BD \quad \checkmark \quad F \quad \checkmark \\ EEC \quad \checkmark \quad FF \quad \checkmark \quad B \quad \checkmark \\ \downarrow \\ \text{no par.} \end{array}$$

$$P_3 = (AC)(E)(BD)(F)$$

$$P_4 = \begin{cases} x=0 \\ x=1 \end{cases}$$

$x=0$

AC ✓ E ✓ BD ✓ F
EE E FF B

$\overbrace{x=1}$

AC ✓ E ✓ BD ✓ F
DB F OB C

$\therefore \underline{\text{no partition}}$

$\therefore \text{finally } P_3 = (A \cup C)(E)(BD)(F)$
 $\Rightarrow A = C \quad B = D$

~~Table becomes~~

PS	NS	
	$x=0$	$x=1$
A	E, 0	D, 1
B	F, 0	D, 0
E	C, 0	F, 1
F	B, 0	C, 0

Synthesis of Synchronous Sequential Circuits

→ Overall Steps

① Word statement

② State diagram

③ State table

④ Reduced standard form of state table.

⑤ State assignment & transition table.

⑥ Choose type of flipflop and form the excitation table.

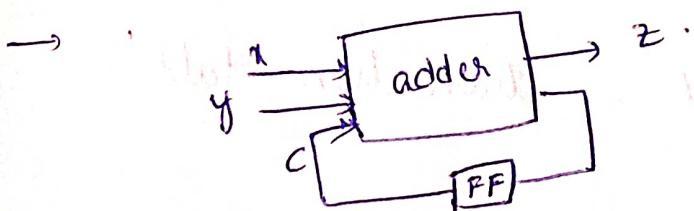
⑦ Kmaps & minimal expression

⑧ Realization of expression

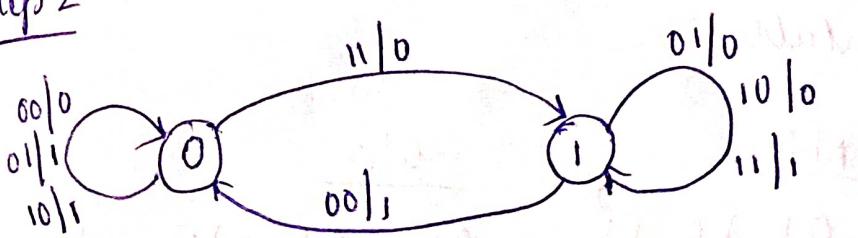
⑨ Serial Binary adder -

(Mealy)

Step 1



Step 2



states 0, 1
↓
correspond to carry.

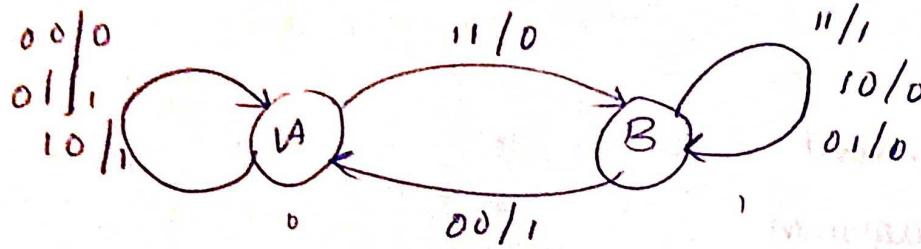
$$c=0 \quad x=0 \quad y=0 \rightarrow s=0 \quad c=0$$

$$c=0 \quad x=1 \quad y=0 \rightarrow s=1 \quad c=0$$

$$c=0 \quad x=1 \quad y=1 \rightarrow s=1 \quad c=0$$

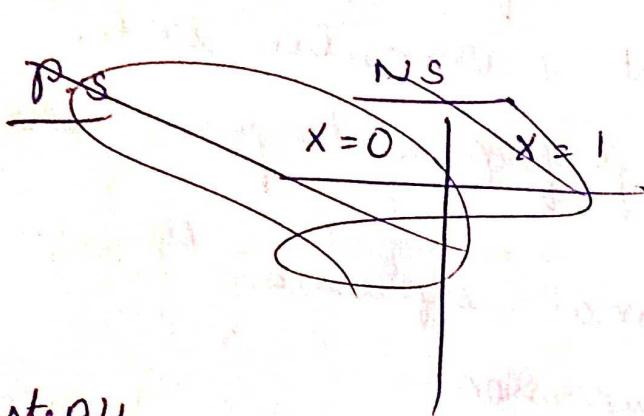
$$c=1 \quad x=0 \quad y=0 \rightarrow s=0 \quad c=1$$

Step 3



Step 3

state table



PS	NS (xy)			
	00	01	10	11
A	A, 0	A, 1	A, 1	B, 0
B	A, 1	B, 0	B, 0	B, 1

Step 4

This is already in reduced form, no change

Step 5

state assignment & transition table.

let A be 0

B be 1.

transition table

PS	NS				O/P			
	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

choose a ft

step 6 * D flip flop \longrightarrow

excitation table \longrightarrow

\downarrow

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

<u>ps</u>	<u>$I \mid P$</u>	<u>N_s</u>	<u>D</u>	<u>Q_{n+1}</u>
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

\rightarrow step 7

Kmaps \bullet

<u>ps</u>	<u>$x \bar{y}$</u>	<u>y</u>	<u>D</u>
0	00 01 11 10		
1	0 1 (1) (1) 1		

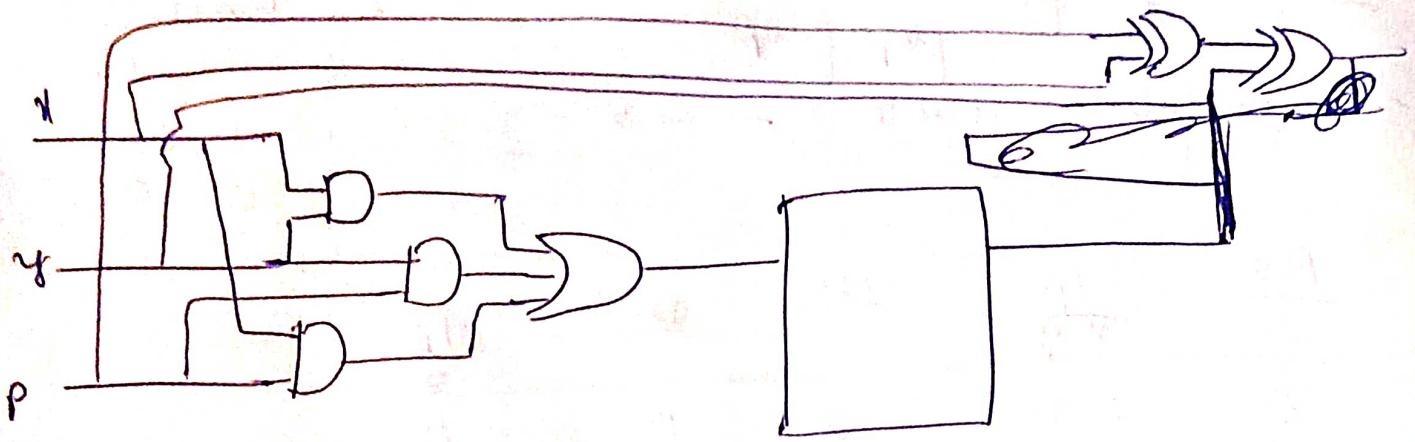
<u>P</u>	<u>$x \bar{y}$</u>	<u>y</u>	<u>Z</u>
0	00 01 11 10		
1	0 1 0 1 0		

$$D = xy + Py + xP.$$

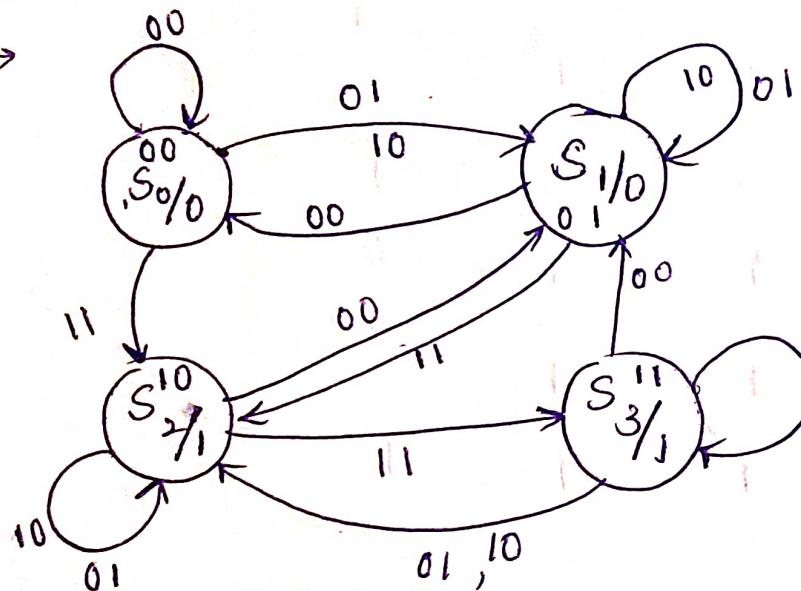
$$Z = P \oplus x \oplus y.$$

\therefore Circuit becomes

step 8



Moore



$$\begin{array}{r} 11 \text{ (D0)} \\ \hline 0 \\ S_3 \rightarrow 10 \end{array}$$

$$1 \ 00$$

$$10$$

$$01$$

$$0$$

$$10$$

$$11$$

$$00$$

$$11$$

$$00 \rightarrow S_0 \ C_0$$

$$01 \rightarrow S_1 \ C_0$$

$$10 \rightarrow S_1 \ C_0$$

$$11 \rightarrow S_0 \ C_1$$

state table

PS

NS

S₀

S₁

S₂

S₃

O/P.

$$\begin{array}{r} 10 \\ \hline 10 \\ 10 \end{array}$$

$$11$$

$$S_1 \ C_1$$

$$00$$

$$01$$

$$10$$

$$11$$

$$10$$

$$11$$

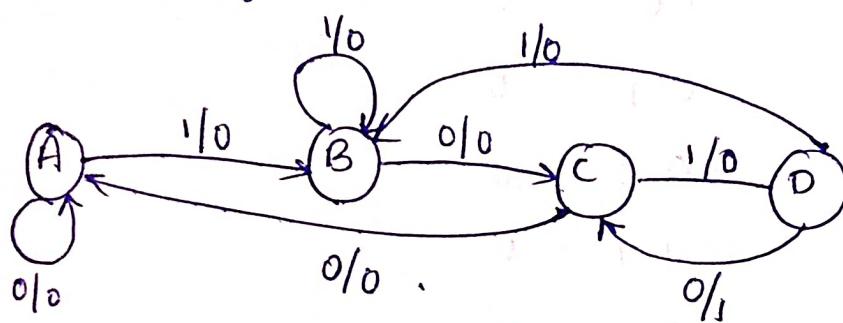
$$10$$

Sequence Detector

Mealy

① 1010 seq detector.

state diagram.



state table

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

→ * reduced S.T

→ State assignment & transition table.

$$A \rightarrow 00 \quad B \rightarrow 01 \quad C \rightarrow 10 \quad D \rightarrow 11$$

PS	NS		O _{out}	
	$x=0$	$x=1$	$x=0$	$x=1$
00	00	01	0	0
01	10	01	0	0
10	00	11	0	0
11	10	01	1	0

ff f mutation table

$D \rightarrow Q_n Q_{n+1} D$

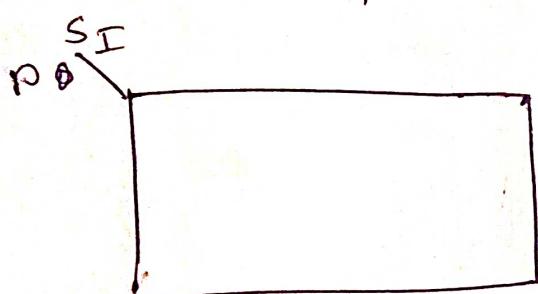
0	0	0
0	1	1
1	0	0

101
state

PS	In ₁	N _S	<u>D₁</u>	<u>O_P</u>
00	0	00	00	0
00	1	01	01	0
01	0	10	10	0
011	01	-	01	0
100	00	-	00	0
101	11	-	11	0
110	10	-	10	1
111	01	-	01	0

Kmaps

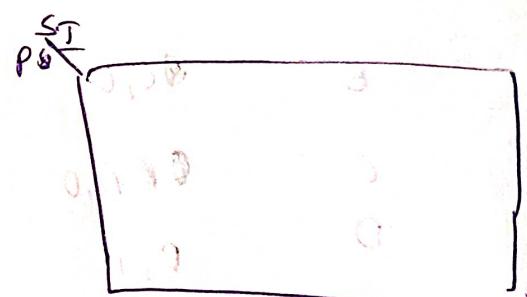
D₁



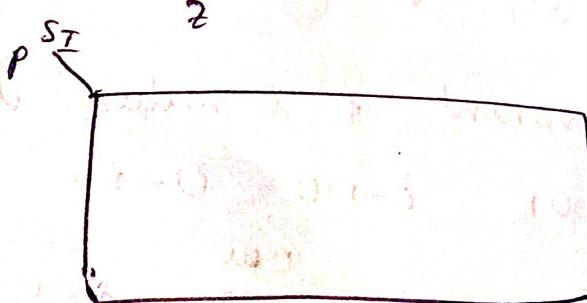
D₂

Q_n

A



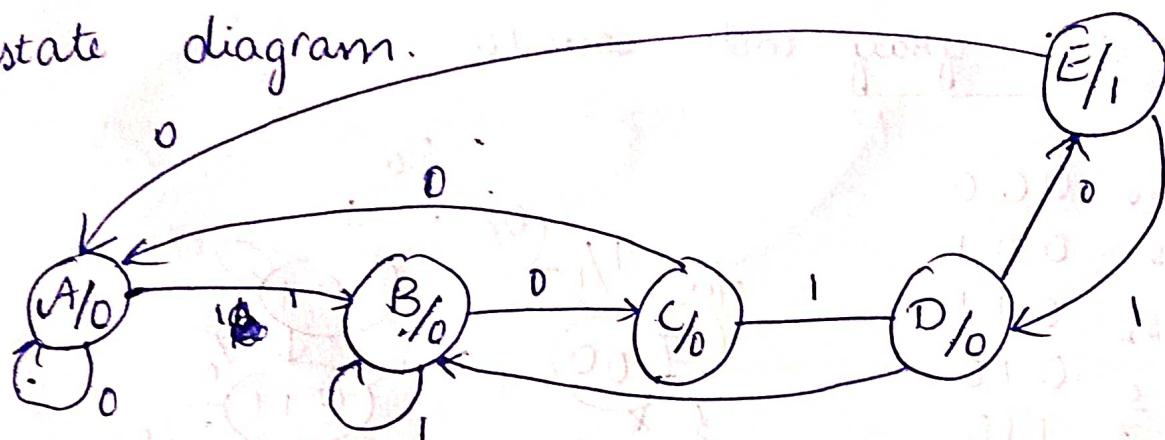
2



Moore type

1010

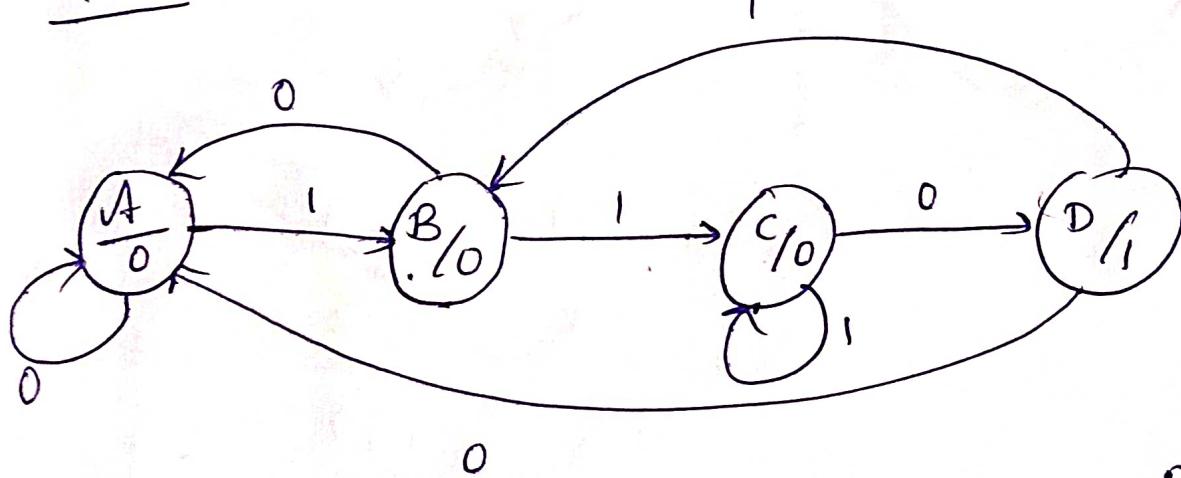
state diagram.



11
100
1011

1010
10101
10100

110



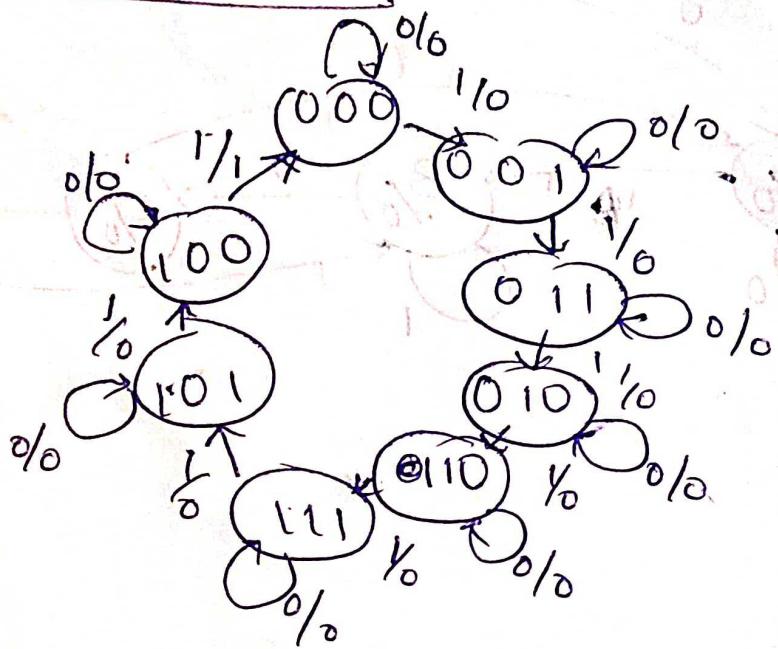
0
10
111

D
1101
1100
1101
1111

Counters

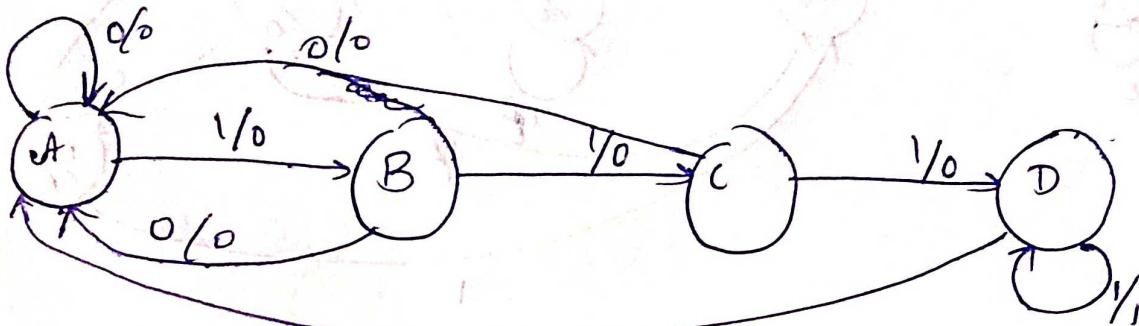
3 bit gray code counter.

$s_0 \quad 000$
 $s_1 \quad 001$
 $s_2 \quad 011$
 $s_3 \quad 010$
 $s_4 \quad 110$
 $s_5 \quad 111$
 $s_6 \quad 101$
 $s_7 \quad 100$

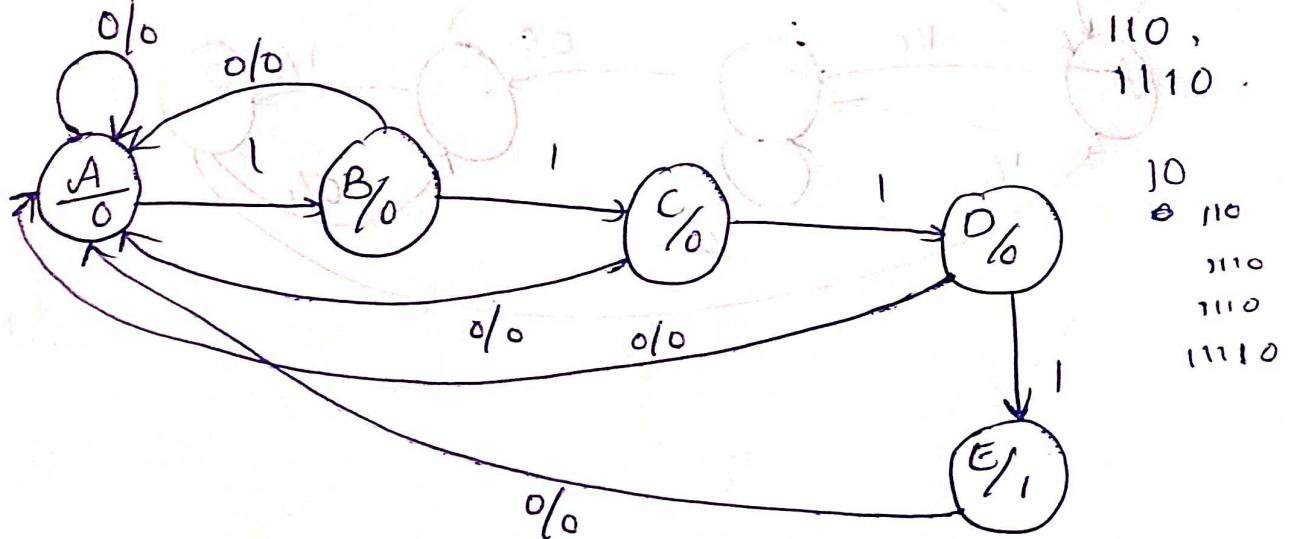


1111 sequence detector

mealy



moore

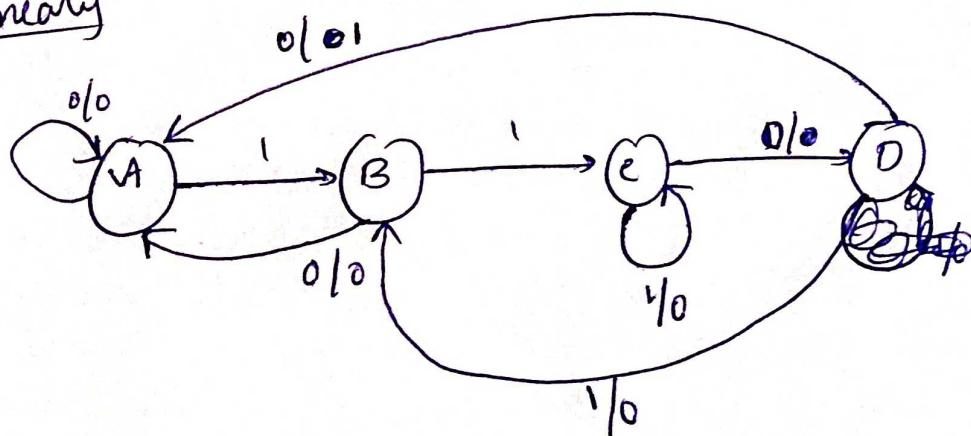


→ 1100, 1010, 1001

① 1100

~~1101~~

mealy



1110
1100
1100.

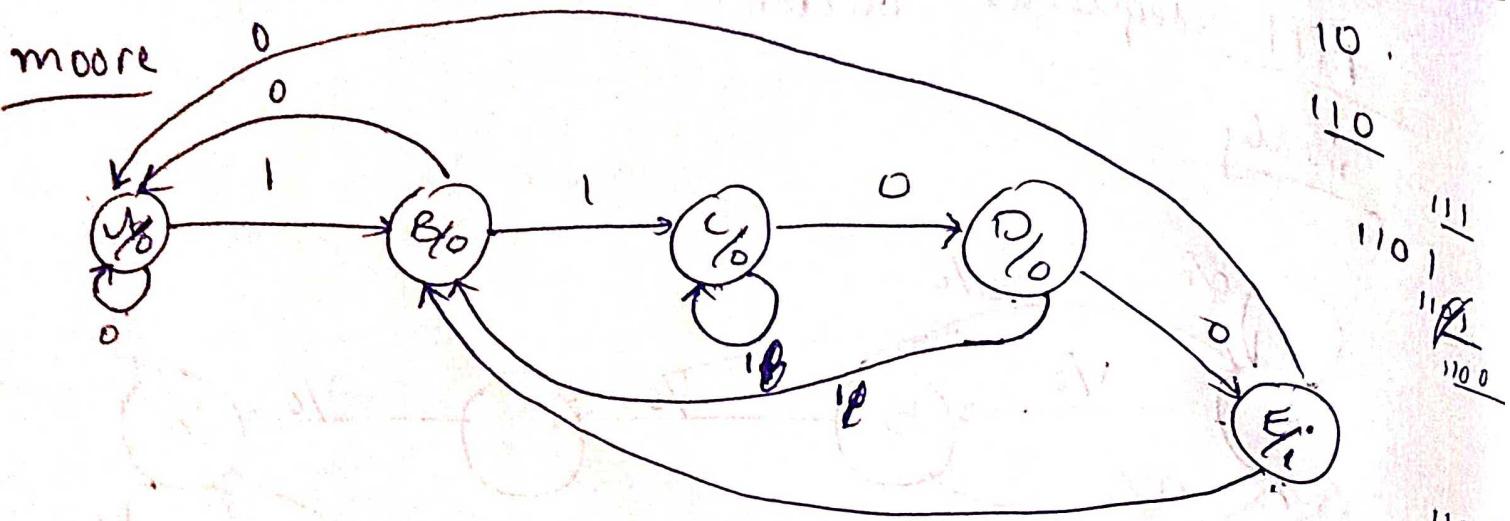
10000

10

1101

1101

1100



$\rightarrow \underline{1010}$ (mealy)

