

## Chapta-11

PAGE NO:  
DATE: / /

(19)

# Design of Synchronous and Asynchronous Sequential Circuits

A design problem starts with word description which is first converted to

- State transition diagram  
or

- Algorithmic State Machine

Later we can prepare state synthesis table.

For flip-flop based implementation, excitation

tables are used to generate design equations

through k-maps.

For ROM based implementation, excitation table

is not required but FF used as memory elements.

There are two different approaches of state machine design called Moore Model & Mealy Model.

Moore Model - output are generated solely

from secondary outputs for memory values

In Mealy Model - primary

inputs combine with memory elements  
to generate circuit output.

Design of Synchronous Sequential Circuit :-

### 11.1 Model Selection

Moore model - output depends only on  
present state

Mealy model - output is derived from  
present state as well as input.

Advantage of Mealy Model :-

- Requires less number of states thereby less  
hardware

- Output generated one clock cycle earlier

## Disadvantage of Mealy Model

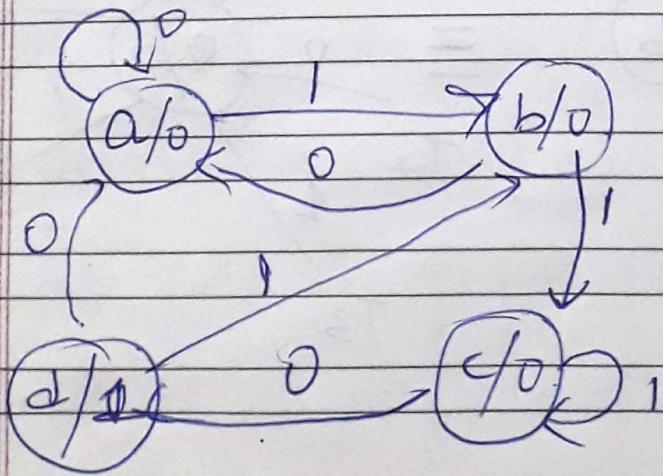
- Input transients, glitches are directly conveyed to output.

This disadvantage is not there in Moore Model as output remains stable over entire clock period.

We can also convert from one model to another.

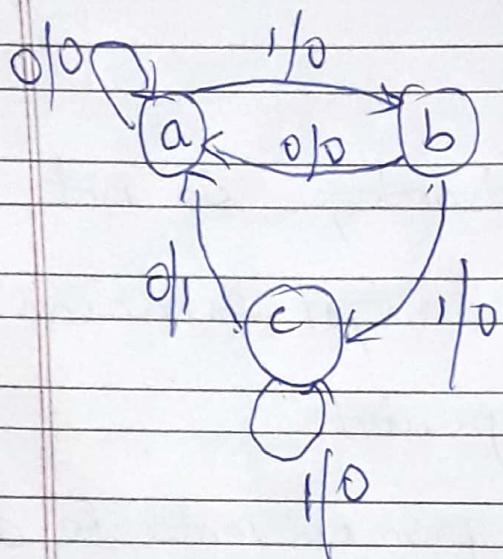
Example:-

Problem: Design a sequence detector that requires '011' using Moore Model state transition diagram.

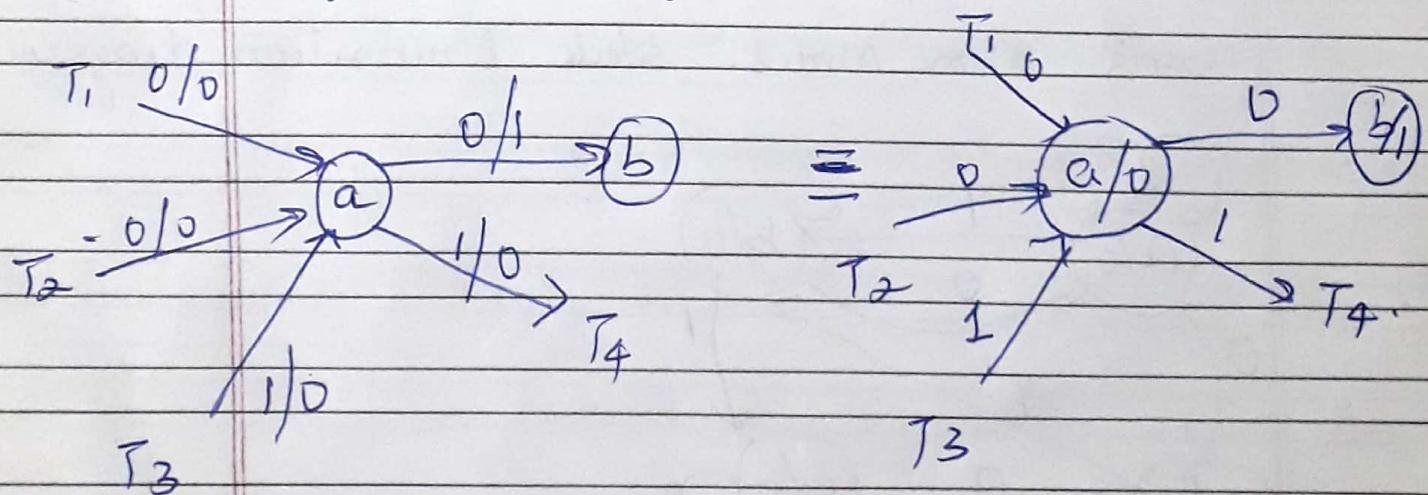


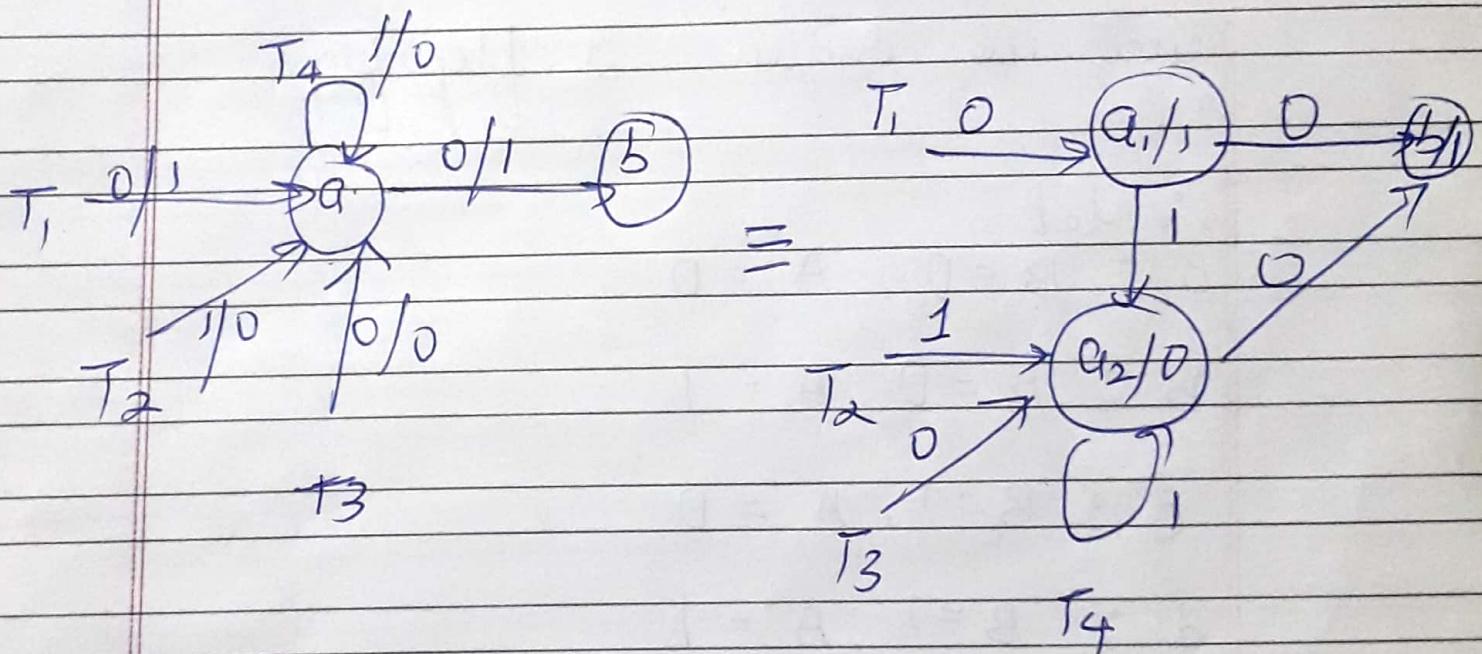
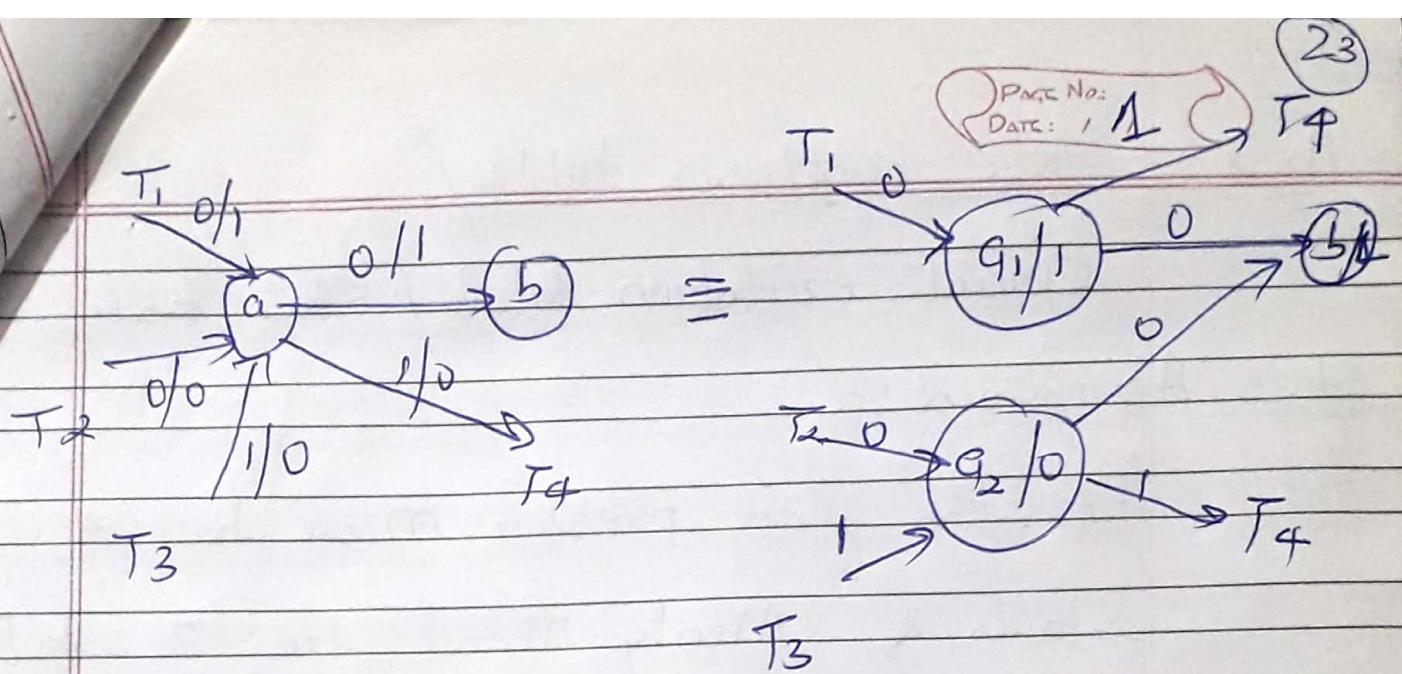
Example 2: Design STD using Mealy model

to detect 011'



Conversion from Mealy to Moore Model :-





If Moore to Mealy use the same  
above sequence (see Right side - to -  
left-side for conversion)

### 11.3 State Synthesis table /

Circuit excitation table / state table

State Assignment :-

For the given problem Moore has 4 states & Mealy Model has 3 states, hence we require 2 flip-flops.

∴ let

a :  $B = 0, A = 0$

b :  $B = 0, A = 1$

c :  $B = 1, A = 0$

d :  $B = 1, A = 1$

We can make any assignment.

There is no rule defined for the order,

but sometimes random order has

led to minimum hardware implementation.

## State Synthesis Table

Microw Model (Table 1)



Present State <u>B<sub>n</sub></u>	An	Present I/p <u>x<sub>n</sub></u>	Next State <u>B<sub>n+1</sub></u>	An <sub>n+1</sub>	Output <u>y<sub>n</sub></u>	J <sub>B</sub>	K <sub>B</sub>	J <sub>A</sub>	K <sub>A</sub>
0 0	0	0	0 0	0	0	0 X	0 X	0	X
0 0	1	0	0 1	0	0	0 X	1 X	1	X
0 1	0	0	0 0	0	0	0 X	X X	X	1
0 1	1	1	1 0	0	0	1 X	X X	1	1
1 0	0	1	1 1	0	0	X 0	1 X	1	X
1 0	1	1	1 0	0	0	X 0	0	0	X
1 1	0	0	0 0	1	1	X 1	1 X	1	1
1 1	1	0	0 1	1	1	X 1	1 X	1	0

(Table 2) State synthesis Table for Mealy mo.

Present state		Present I/p	Next state		Present O/p	$J_B$	$K_B$	$J_A$	$K_A$
$B_n$	$A_n$	$X_n$	$B_{n+1}$	$A_{n+1}$	$Y_n$				
0	0	0	0	0	0	0	x	0	x
0	0	1	0	1	0	0	x	1	x
0	1	0	0	0	0	0	x	x	1
0	1	1	1	0	0	1	x	x	1
1	0	0	0	0	1	x	1	0	x
1	0	1	1	0	0	x	0	0	x

#### 11.4 Design Equations & Circuit Diagram

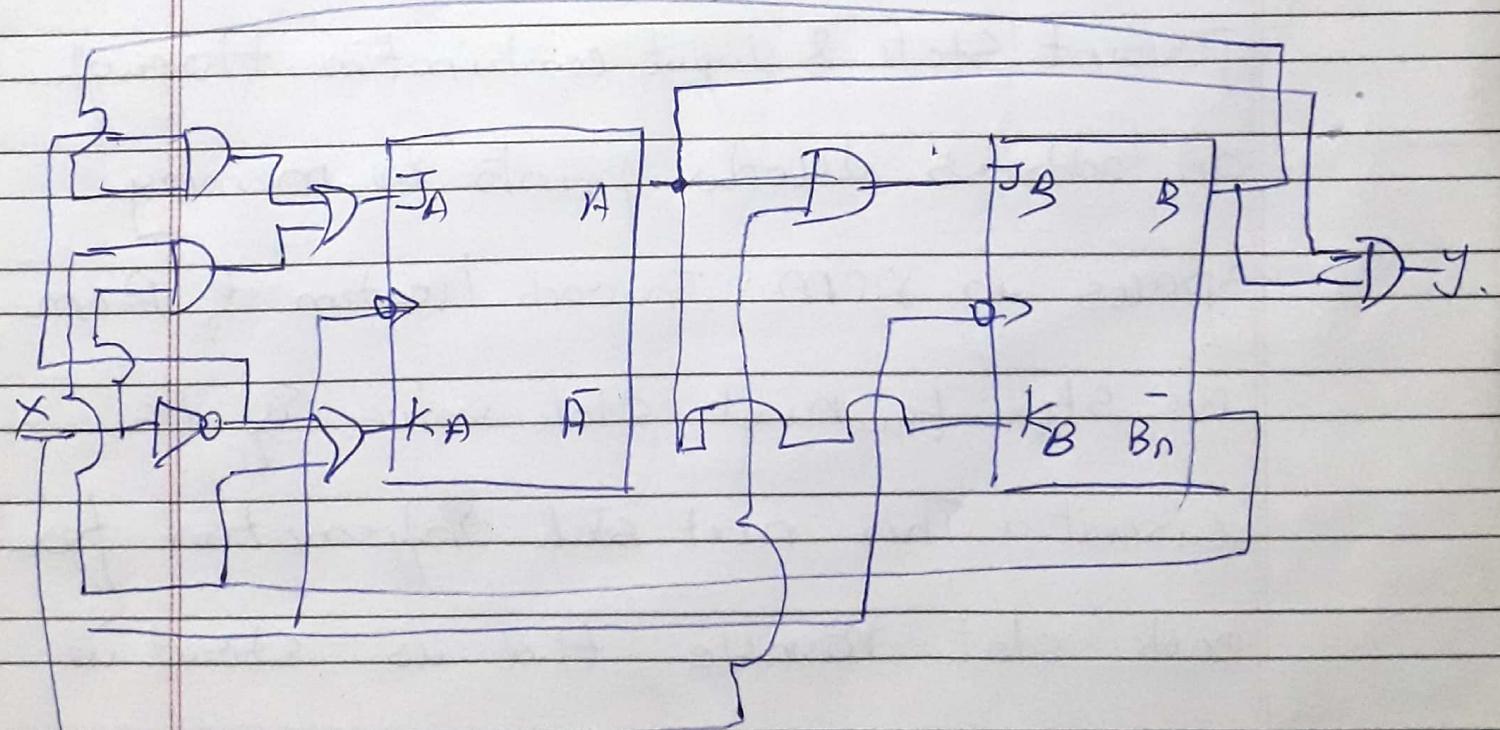
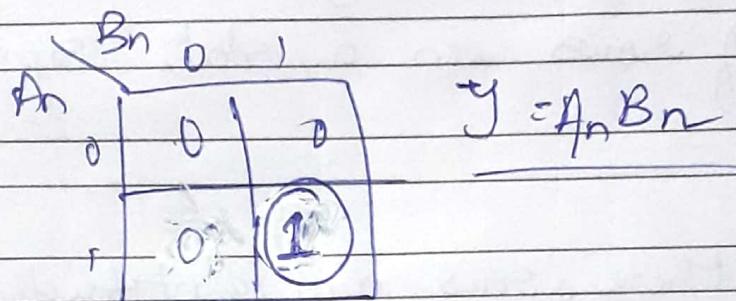
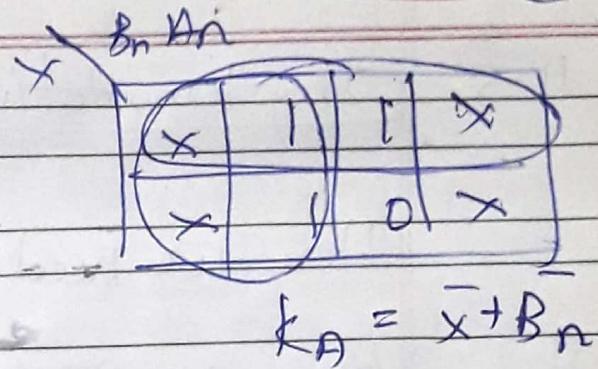
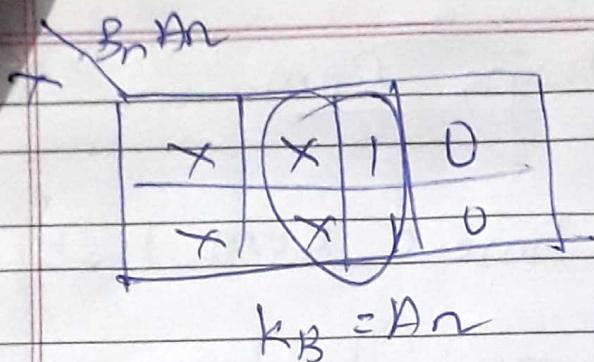
The Karnaugh map & sequence detector circuit diagram for table 1, is given below:-

$B_n A_n$		00	01	11	10
0	0	0	x	x	
1	0	1	x	x	

$B_n A_n$		00	01	11	10
0	0	x	x	1	
1	x	1	x	0	

$$J_B = X A_n$$

$$J_A = \bar{X} B_n + X \bar{B}_n$$



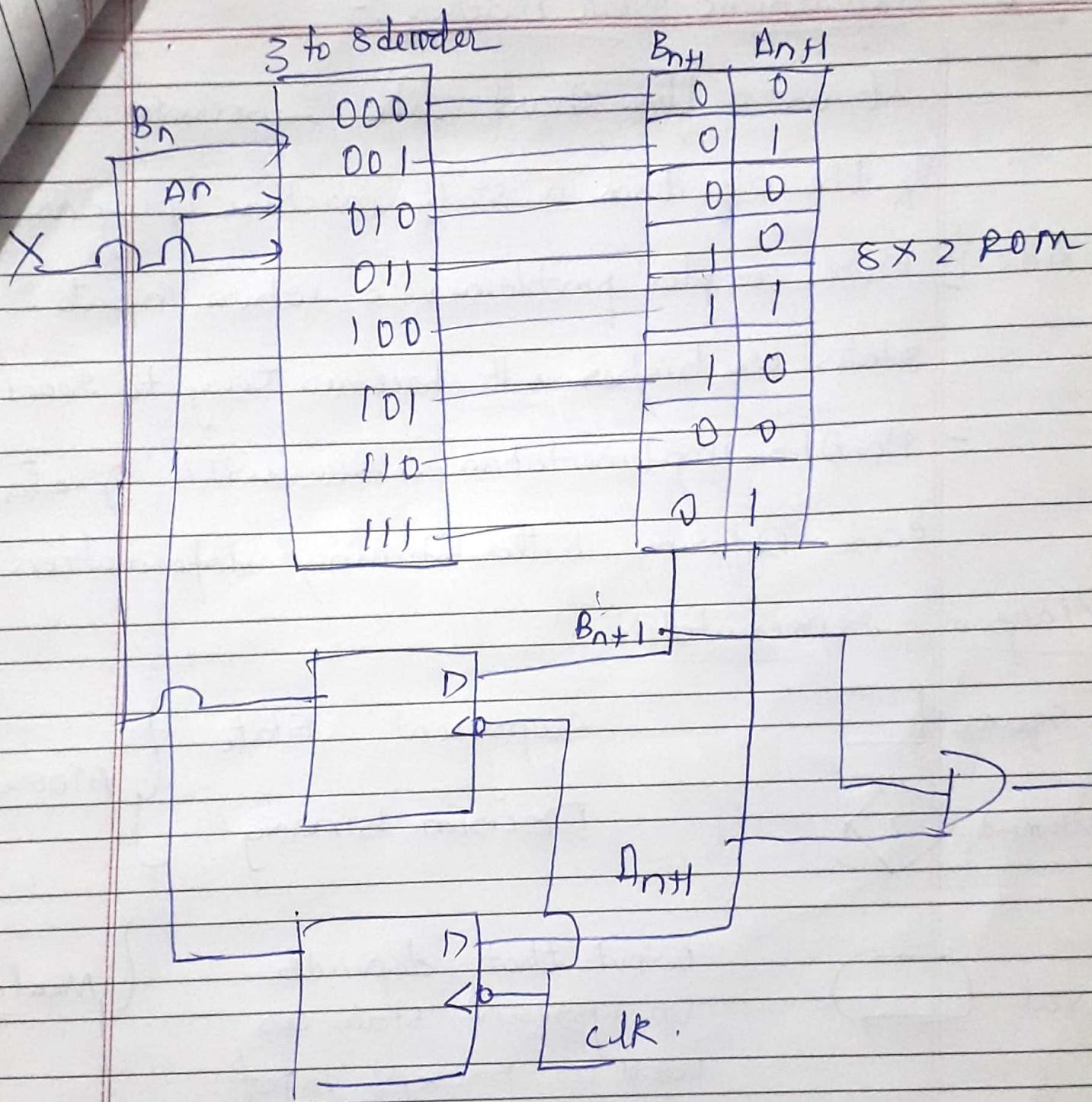
Refer TB Pg 400 for mealy model

11.5

## Implementation using ROM

Here we need to have a ROM that has as many memory locations as the number of rows in a state synthesis table.

Each of these rows are identified by present state & present input. This present state & input combination through an address decoder points to memory spaces in ROM. In each location of ROM we store the next state value of the circuit. This next state information for each state variable that is stored in ROM is fed to flip-flop inputs. At clock trigger they appear at the output of the flip-flop. This is the present state which is fed to decoders.



The ~~ROM~~ implementation for table 2

is given in Pg 402.

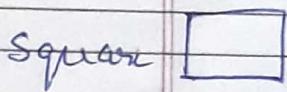
## 11. b Algorithmic State Machine :-

It is a flow chart like representation of the algorithm a state machine performs.

Adv :- More complex problem, i.e. where inputs & states are higher, it becomes easy to read.

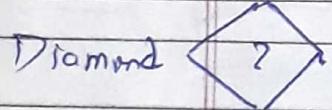
- Handles implementation issue with greater ease differing better timing information.

### Diagram representation

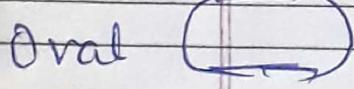


present state

} Moore

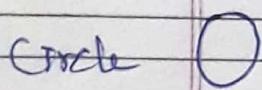


Decision binary



output that depends  
on present state as  
well as present input.

} Mealy

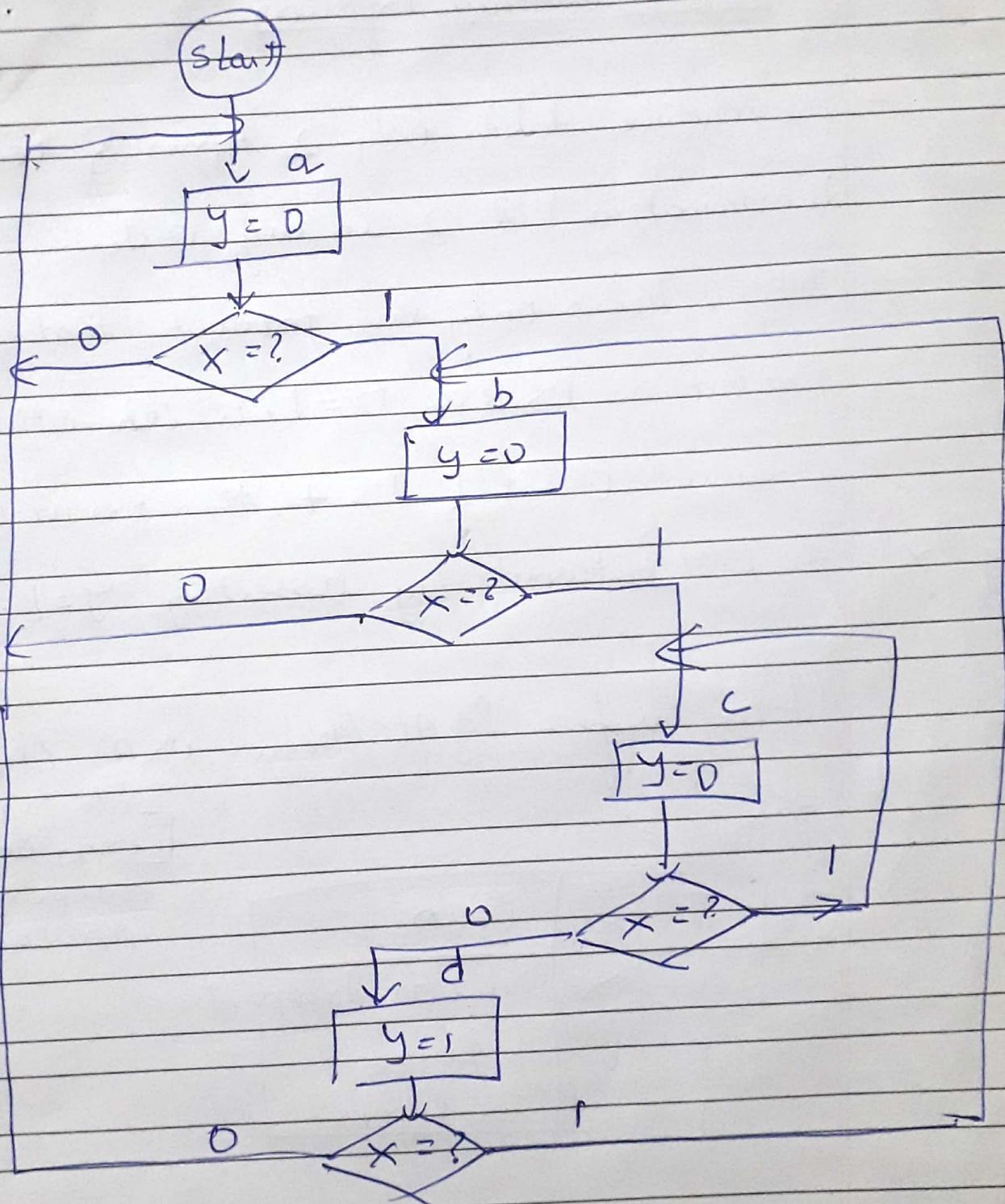


Start / Stop / connectors

↳ used by  
both Moore  
and Mealy

# ASM Chart : Moore Model

PAGE No.:  
DATE: / /



## Example 2: Vending Machine Problem

- The machine takes only 2 types of coins denomination 1 & 2 in any order.
- It delivers only one product that is priced at Rs 3  $\Rightarrow D=1$ , which otherwise remains 0. If Rs. 4, then product delivered by 'X'  $\in \{1\}$  returned by asserting  $Y=1$ .

Please refer Pg 405/406 for ASM chart 2

for

State Synthesis table  
Design eq. circuit  
Diagram & ROM  
on Pg 407-408]

I	J	Coin
0	X	No coin dropped
1	0	1 Rupee
1	1	Two Rupees

a  no coins

$X \rightarrow$  Product delivered  
 $Y \rightarrow$  Re. 1 returned.

b  Re. 1

c  Re. 2