U-3

- Review of MOS Transistor theory, second order effects, MOS Models, Scaling of MOSFETs, Operation of NMOS, PMOS, CMOS, BiCMOS, VTC, Noise Margins, power dissipation, speed area

U-4 - Basics of VLSI design, implementation of logic functions, rise time/fall time/delay time considerations, fan-in, fan-out, CMOS logic Structures: Domino logic, NP zipper Logic, CVSL, DVSL.

U-5 - FMS Design: State Machines, Mealey & Moore machines, state diagrams, state table reduction techniques for state tables, transition tables, design of sequential circuits using FSMs, VHDL coding for FSMs.

U-5

- FMS - finite state Machines :- Finite state Machines are widely used in modelling of application behaviour (control theory), design of hardware digital systems, software engineering, compilers, Network protocols & ~~linear~~ computational linguistics.

1. Mealy Machine :- In a Mealy Machine, the outputs are a function of the present state and the value of the inputs. Accordingly the outputs may change asynchronously in response to any change in the inputs.

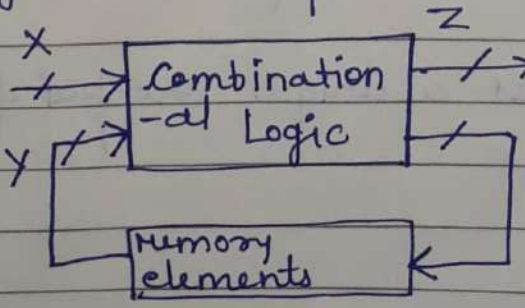
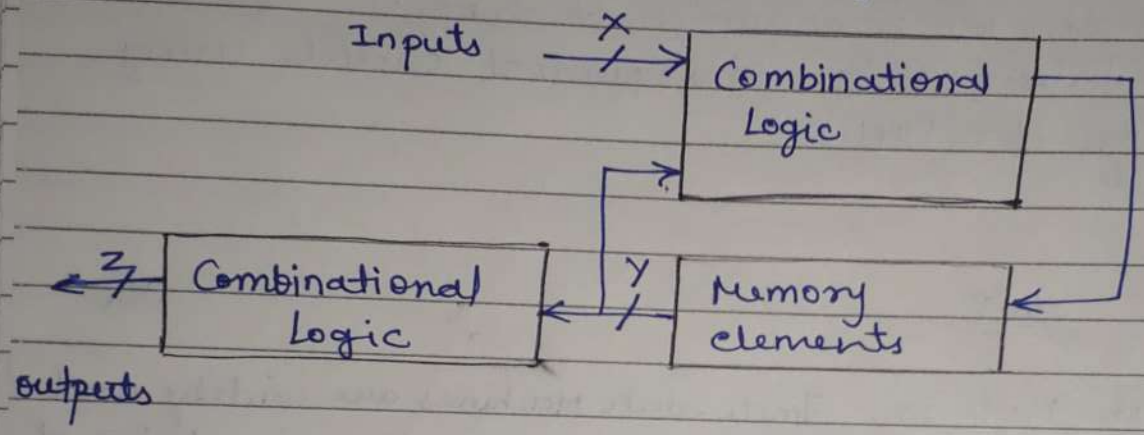


Fig :- Mearly type Machines

2) Moore Machine :- In a Moore Machine the outputs depend only on the present state.

- A combinational logic block maps the inputs & the current state into the neccassary flip-flop units to store the appropriate next state just like Mealy Machine.
- However, the outputs are computed by a combinational logic block whose inputs are only the flip-flop states outputs
- The outputs change synchronously with the state transition triggered by the active clock edge.

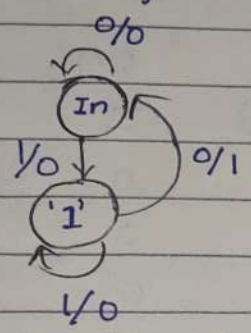


- Comparisions of the two Machine Types :-

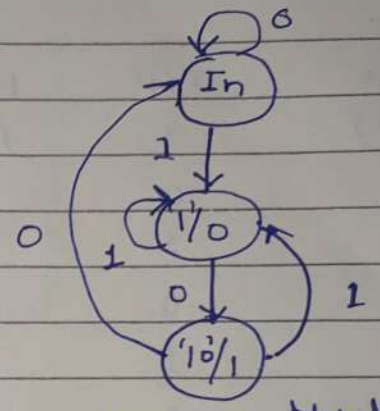
- Consider a finite state Machine that checks for a pattern of '10' & asserts logic high when it is detected.

- The value stated on the arrows for Mealy Machine is of the form Z_i/X_i where Z_i represents input values & X_i represents output value.
- A Moore machine produces a unique output for every state irorespective of inputs.

- Accordingly the state diagram of the Moore Machine associates the O/P with the state in the form state-notation / output-value.
- The state transition arrows of Moore machine are labelled with the input value that triggers such transition.
- Since a Mealy Machine associates outputs with transitions, an output sequence can be generated in fewer states using Mealy machine as compared to Moore machine. The



Mealy Machine



Moore Machine

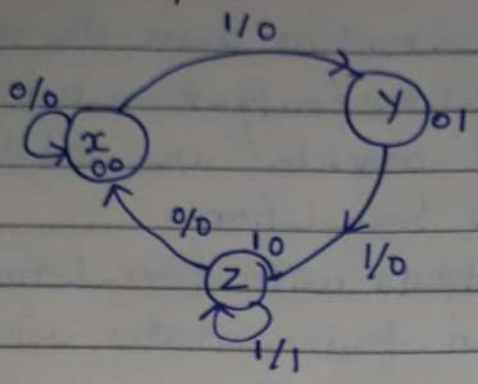
Mealy & Moore state diagrams for '10' sequence detector

- Timing diagram :- To analyze Mealey & Moore machine timings, consider the following problem. A state-machine outputs '1' if the input is '1' for three consecutive clocks.

- State diagram :- The graphical representation of truth table is known as state diagram.

Mealy Machine

Ex - 1.



state diagram.

Q// 111 → three consecutive 1

state table :-

Present state		I/P	Next state		O/P
A(t)	B(t)	x	A(t+1)	B(t+1)	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	0	1

- this state is in don't care → 11.

- output of Mealy Machine depend on current state & input.

use k-Map.

x \ AB	00	01	11	10
0	0	0	X	0
1	0	1	X	1

$A(t+1) = x.B + xA$

$\Rightarrow x(A+B)$

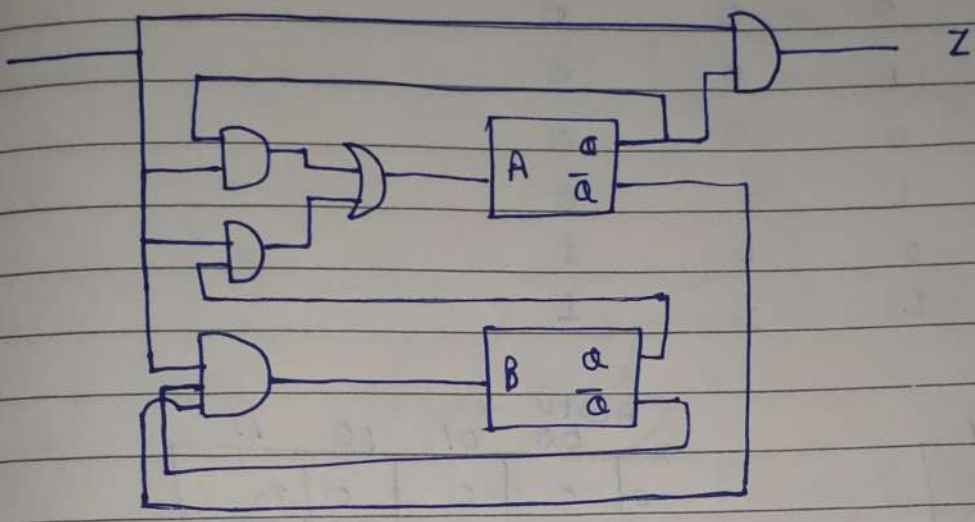
x \ AB	00	01	11	10
0	0	0	X	0
1	1	0	X	0

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

A' → expres

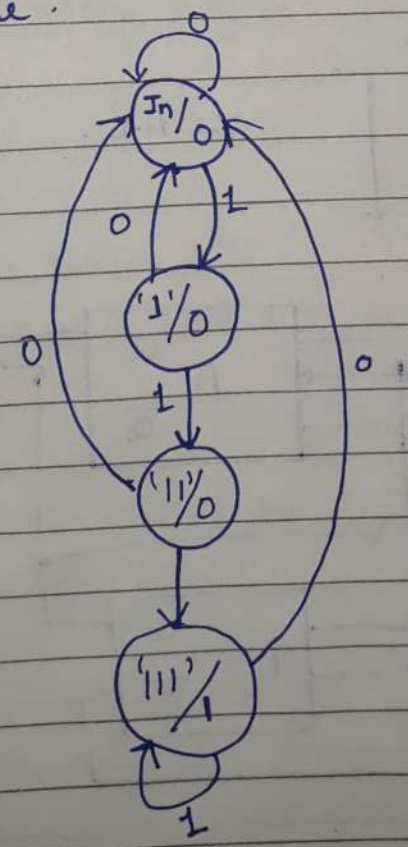
x \ AB	00	01	11	10
0	0	0	X	0
1	0	0	X	1

$z = x \cdot A$



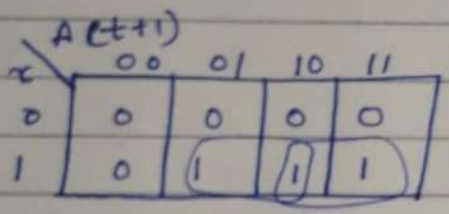
Q.2 Moore Machine :- it only depends on current or present state only.

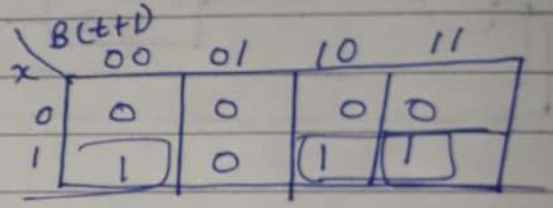
- 111 consecutive one.



Moore Machine state diagram

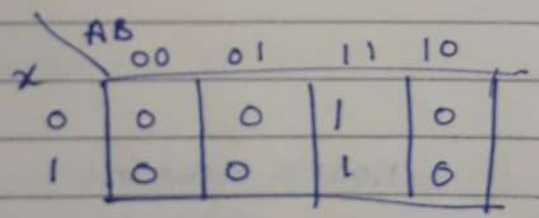
A	B	x	Next State A(t+1)	Next State B(t+1)	z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1



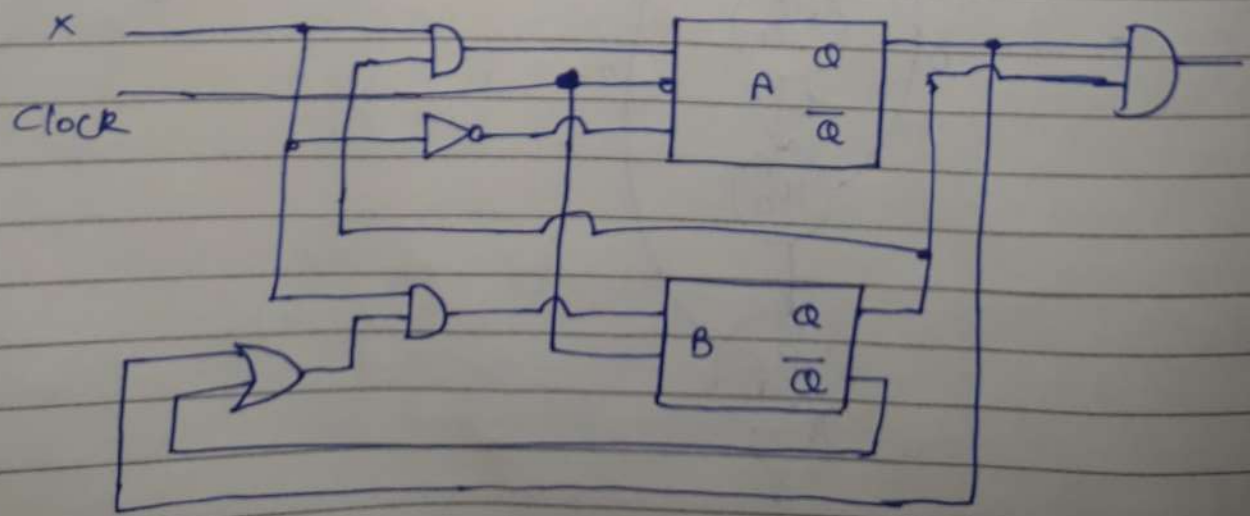


$A(t+1) \Rightarrow x.(A+B)$

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



$z = A.B$.



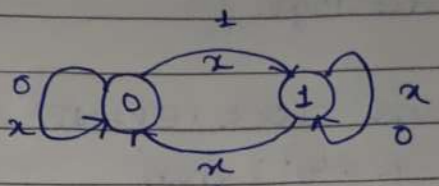
- Draw the FSM state diagram for J-K flip flop & write VHDL code for it.

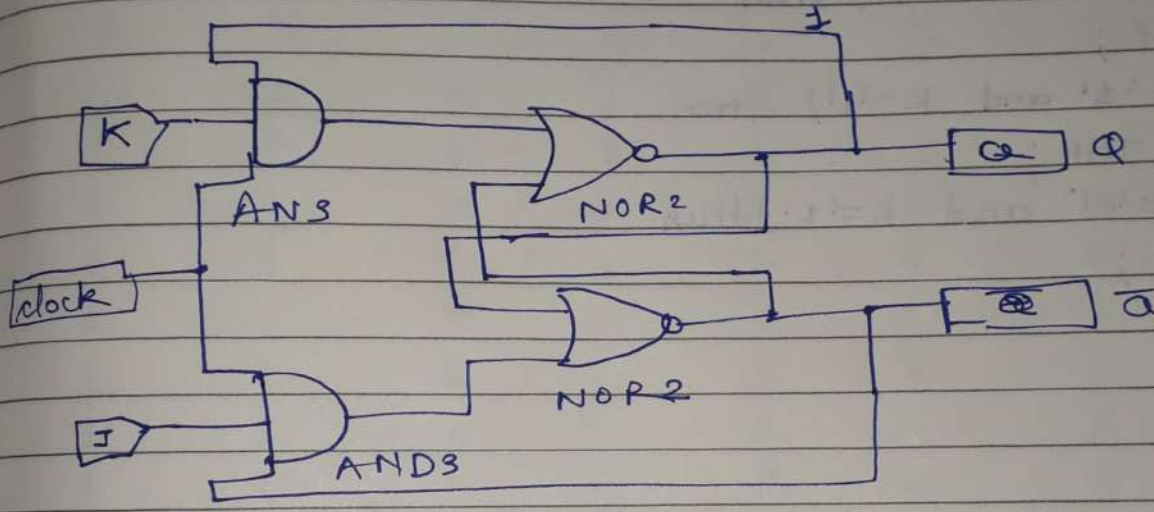
- J-K flip flop

T.T.

Q	J	K	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

state-diagram →





- VHDL code for JK-flip flop :-

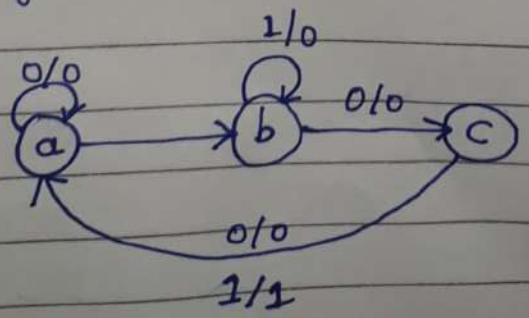
```
library ieee ;
use ieee. std_logic_1164.all;
use ieee. std_logic_arith.all;
use ieee. std_logic_unsigned.all;

entity JK_ff is
PORT (J, K, clock : in std_logic ;
Q, QB : out std_logic);
end JK_ff;
```

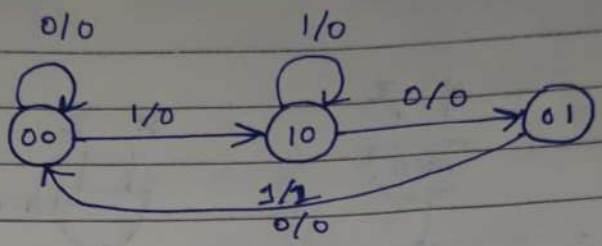
```
Architectural behavioral of JK_ff is
begin
PROCESS (CLOCK)
Variable TMP : std_logic ;
begin
if (CLOCK = '1' and CLOCK'EVENT) then
if (J='0' and K='0') then
TMP := TMP ;
else if (J='1' and K='1') then
TMP := not TMP ;
else if (J='0' and K='1') then
TMP := '0' ;
else
TMP := '1'
end if ;
end if ;
Q <= TMP ;
Q <= not TMP ;
end Procen ;
end behavioral
```

Q. Design a mealy State Machine for a sequence detector that will detect a serial input sequence of 010.

- State diagram :-

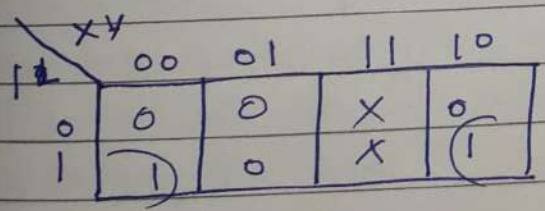


- code assignment :-

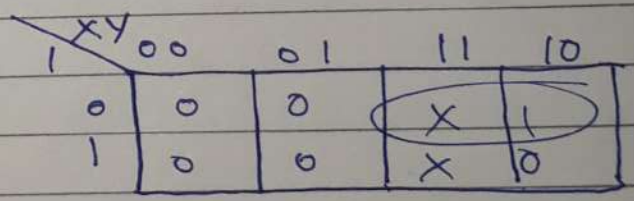


- state table :-

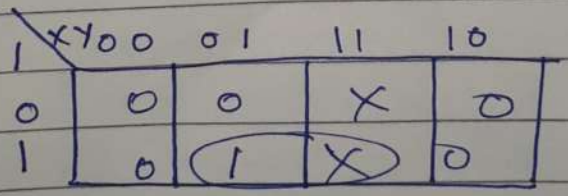
Present states		i/p	Next states		Flip Flop Excitations		o/p
X	Y		X'	Y'	Dx	Dy	
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X



$D_x = \bar{Y} \cdot 1$

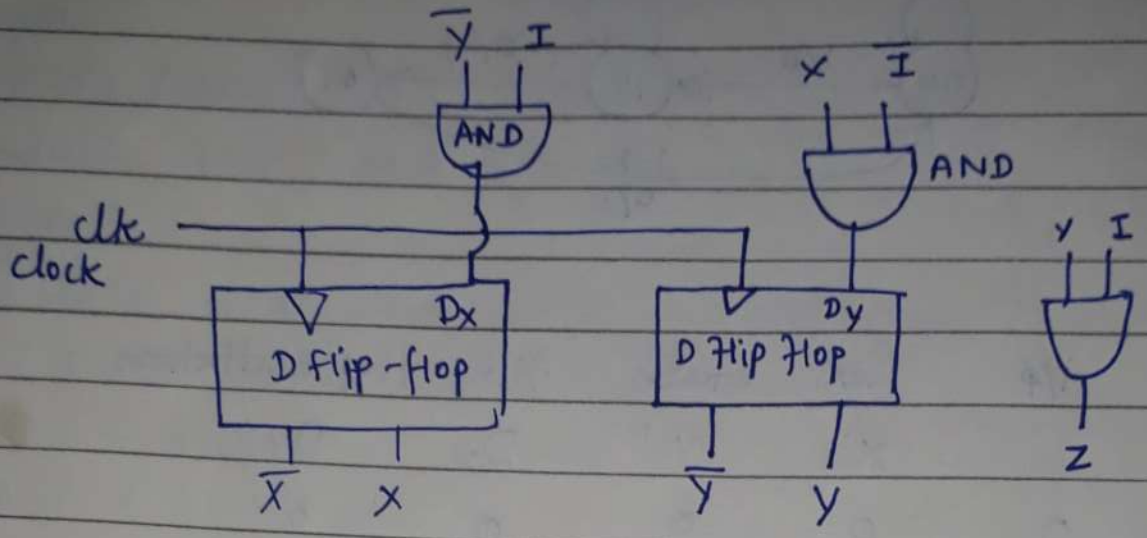


$D_y = X \cdot \bar{1}$



$Z = Y \cdot 1$

Circuits :-


$\bar{Y}$ I
AND
X $\bar{I}$
AND
clk
clock
Dx
D flip-flop
Dy
D flip flop
Y I
Z
$\bar{X}$ X
$\bar{Y}$ Y