

UNIT - 4Algorithmic state Machines & Programmable Logic Devices.

930 2003 poib MEA 2003 poib stofz 301 307 (1)

0. f039301 of 301203

- * What is the Finite state Machine?

→ FSM:

A synchronous sequential circuit is also called as Finite state Machine (FSM), if it has finite number of states.

- Finite state Machine is an abstract mathematical model of a sequential logic function.

- It has finite inputs, outputs & number of states.

- * There are two types of fsm's

1) Mealy state Machine

2) Moore state Machine

- * ASM chart:-

ASM =

The Algorithmic state Machine (Asm) is method for designing fsm.

- It is used to represent diagrams of digital integrated circuits.

- It is like state diagram.

* Advantages of ASM:

- 1) For large state diagrams, ASM diagrams are easier to interpret.
- 2) Conditions for a proper state diagram are automatically satisfied.
- 3) ASM diagrams are easily converted to other forms.
- 4) Used for implementing the hardware design of the digital system.

* ASM charts & Applications:

- It is similar to conventional flowchart but interpreted differently.
- It is a graphical view.
- ASM chart is basically flowchart for H/W algorithm.
- It considers timing relationship.
- It is suitable for describes the sequence of events.
- Used in general control networks in any digital systems.
- Used in the design control unit of computer.

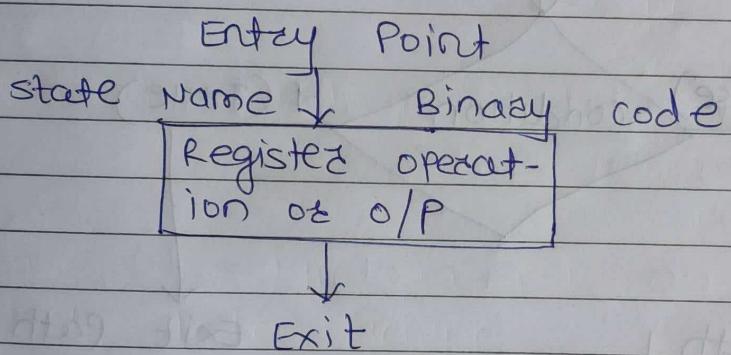
* Components of ASM chart :
(chart Notations)

Q. Explain the basic components of Asm chart ?

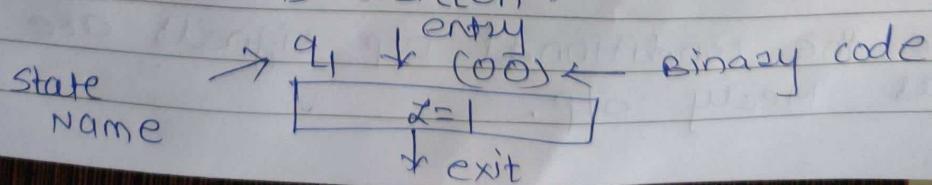
→ Three basic elements.

- 1) state Box
- 2) Decision Box
- 3) conditional Box .

① state Box :-

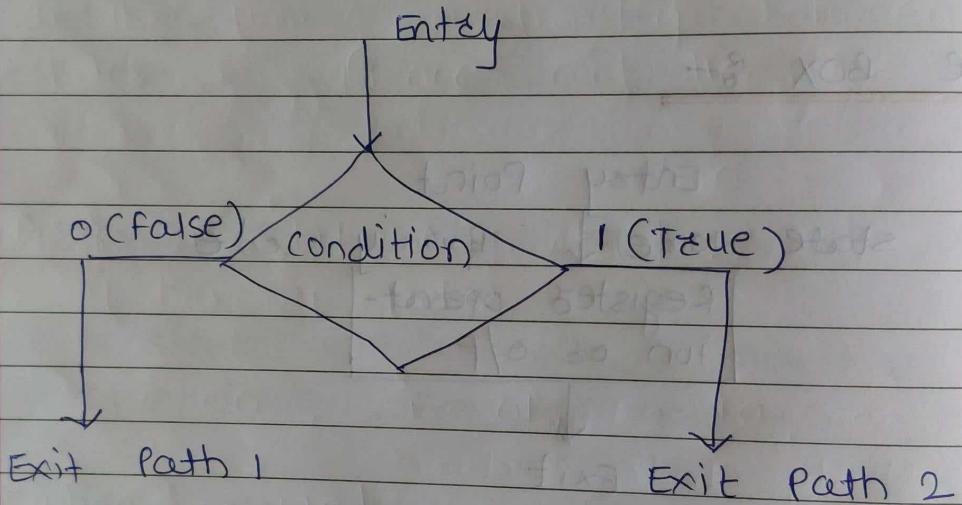


- State Box is used for indicating the state of the controller in the control sequence.
- The shape of state Box is rectangular.
- For each state there are one entry & exit Point.
- On the eight side of the box binary code of each state is written.



2) Decision Box -

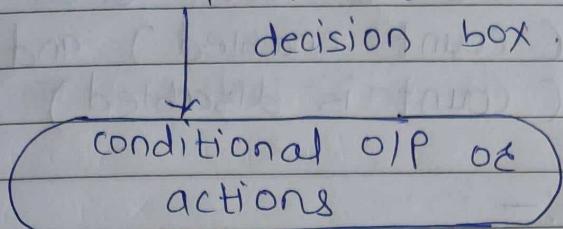
- the decision box is diamond shaped box with two or more exit paths.
- one exit path is taken when the condition is true, otherwise other is taken when the condition is false.
- It indicates by $(1) \rightarrow \text{True}$, $(0) \rightarrow \text{false}$.



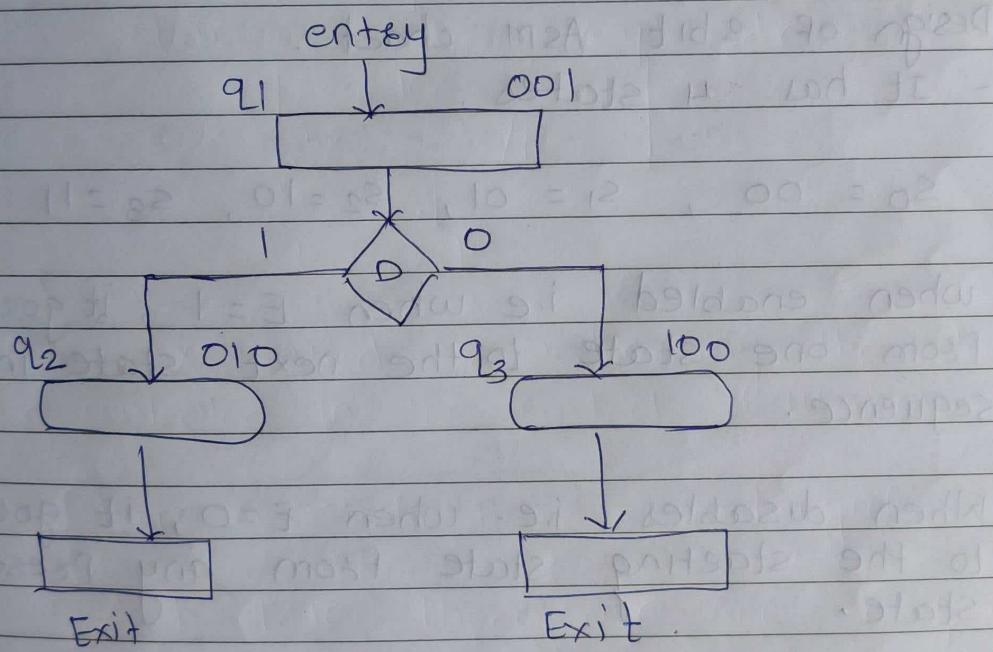
3) Conditional Box:-

- It is oval shaped.
- The o/p signals that are of Mealy type
- These o/p depend not only on the state but also the inputs to the fsm.
- i.e. the conditional o/p signals are known as Mealy o/p.

- condition O/P must follow a decision box from exit path of decision box.



* Example of ASM chart:



~~IMP.~~

Q.1 What is an ASM chart? Design the ASM chart for a 2-bit binary counter having one enable line E such that when:
 $E=1$ (count enabled) and
 $E=0$ (count is disabled)

→ ASM chart:-

is an alternative for describing the behaviour of FSM.

It describes the sequence of events as well as the timing relationship between the states of sequence controlled.

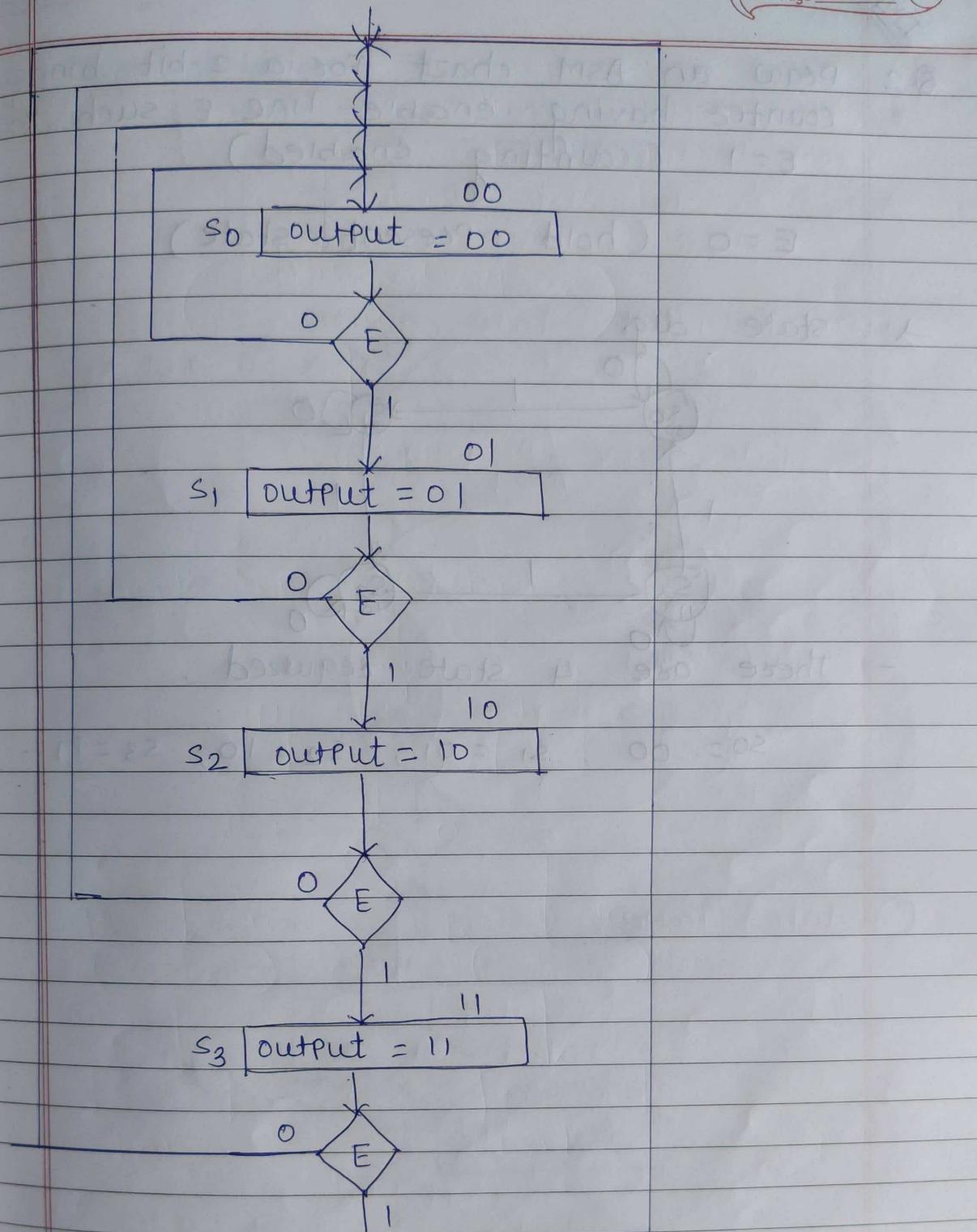
Design of 2bit ASM chart.

- It has 4 states.

$$S_0 = 00, S_1 = 01, S_2 = 10, S_3 = 11$$

when enabled i.e. when $E=1$, it goes from one state to the next state in sequence.

When disables i.e. when $E=0$, it goes to the starting state from any present state.



(ASM chart)

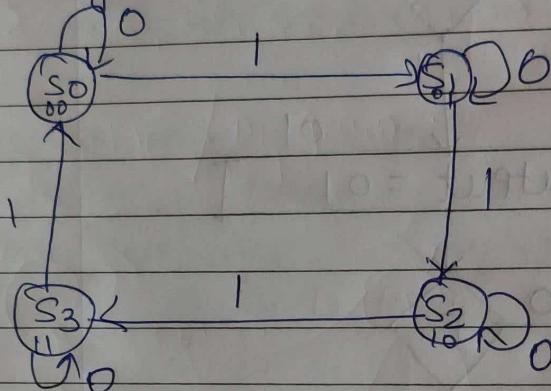
IMP

Q2

Draw an ASM chart for a 2-bit binary counter having enable line E such that
 $E=1$ (counting enabled)

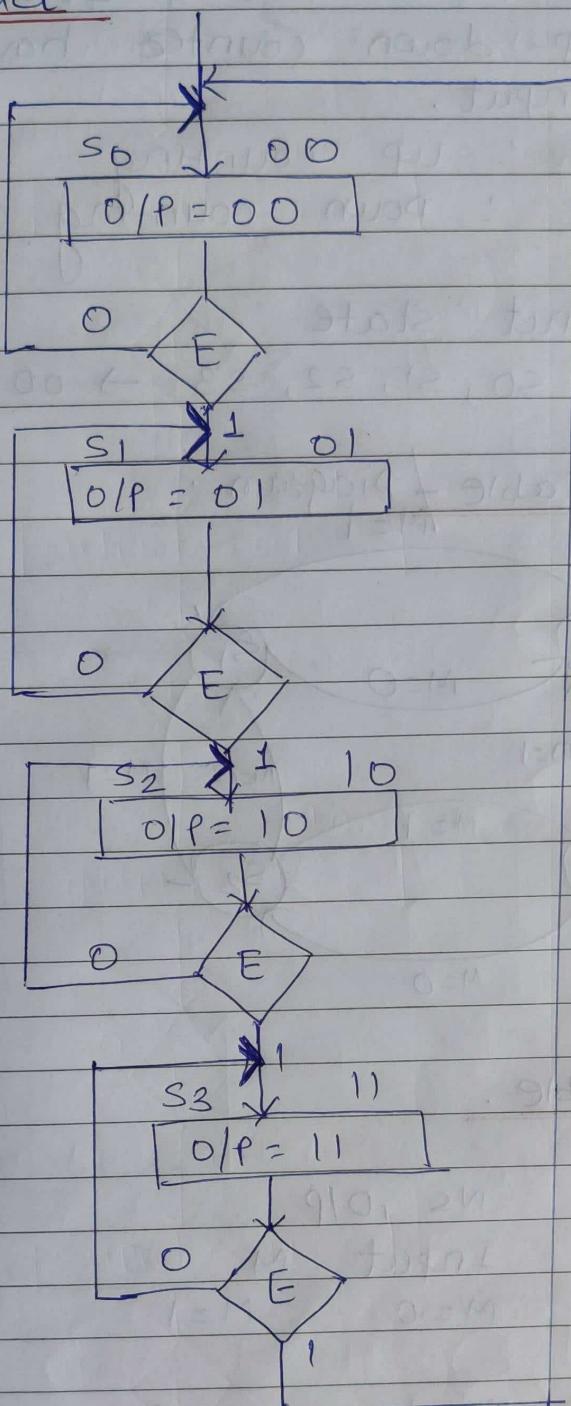
$E=0$ (hold present state)

→ state dig.



- These are 4 state required.

$$S_0 = 00, S_1 = 01, S_2 = 10, S_3 = 11$$

ASM chart -

IMP

Q.3)

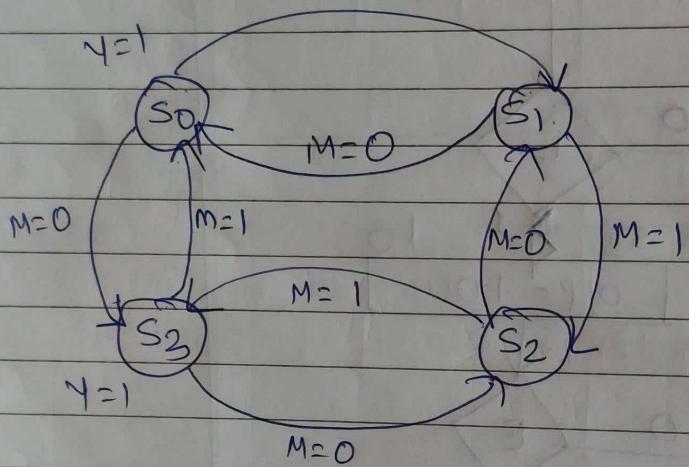
Draw an ASM chart & state Table for 2-bit up-down counter having mode control input.

$M = 1$: up counting
 $M = 0$: down counting.

→ ① 4 input state

i.e. $s_0, s_1, s_2, s_3 \rightarrow 00 \rightarrow 01 \rightarrow 10 \rightarrow 11$

② State Table / Diagram



③ state Table.

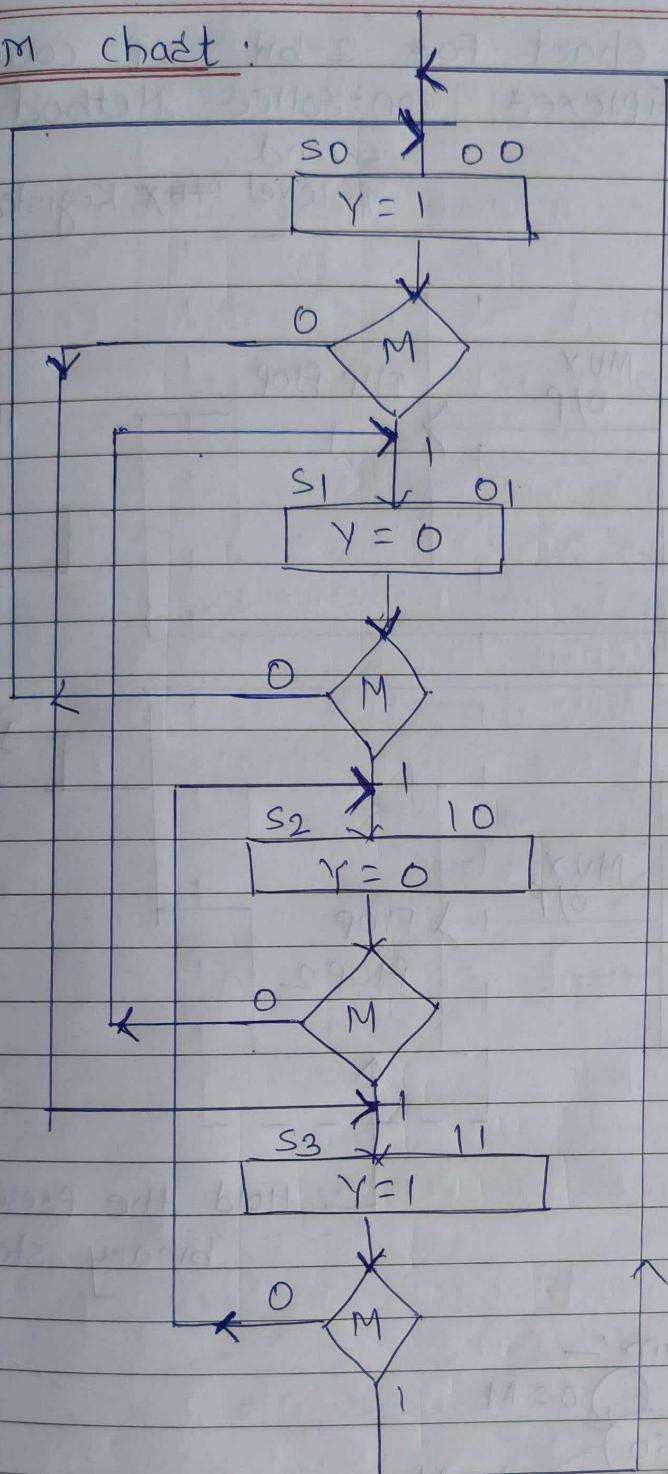
PS

NS, O/P

Input M

M=0 M=1

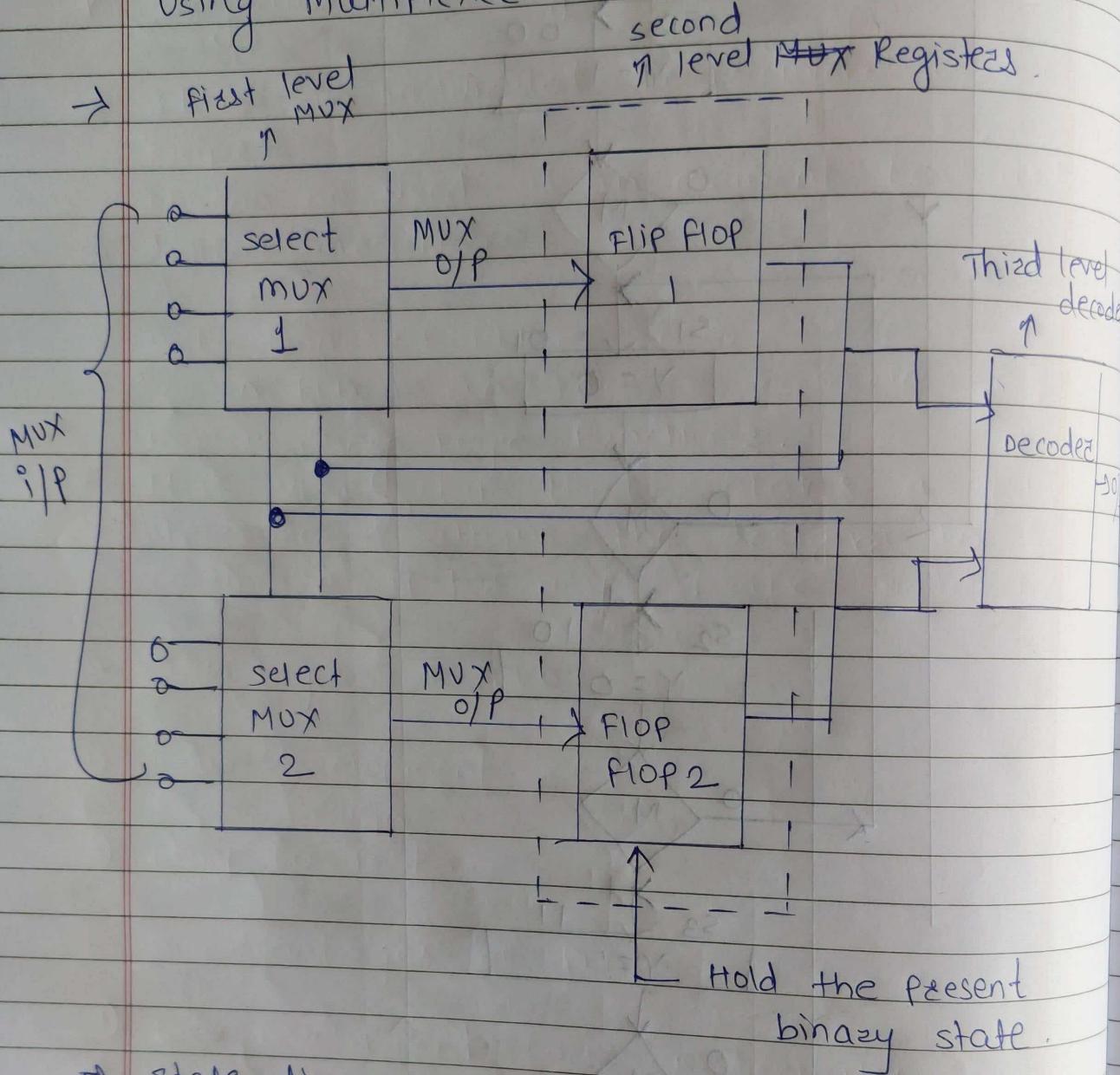
s_0	$s_3, 1$	s_1, f_0
s_1	$s_0, 1$	s_2, f_0
s_2	$s_1, 0$	s_3, f_1
s_3	$s_2, 0$	s_0, f_1

Q1) ASM chart :

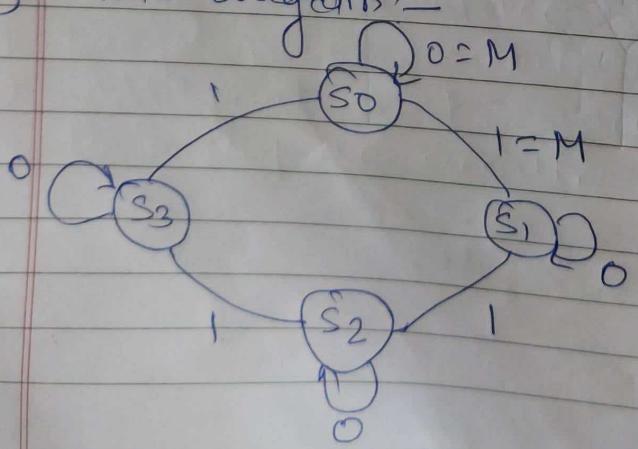
Q. END SEM
2022

Date _____
Page _____

Q.4 Draw ASM chart for 2-bit up counter using multiplexed controlled Method.

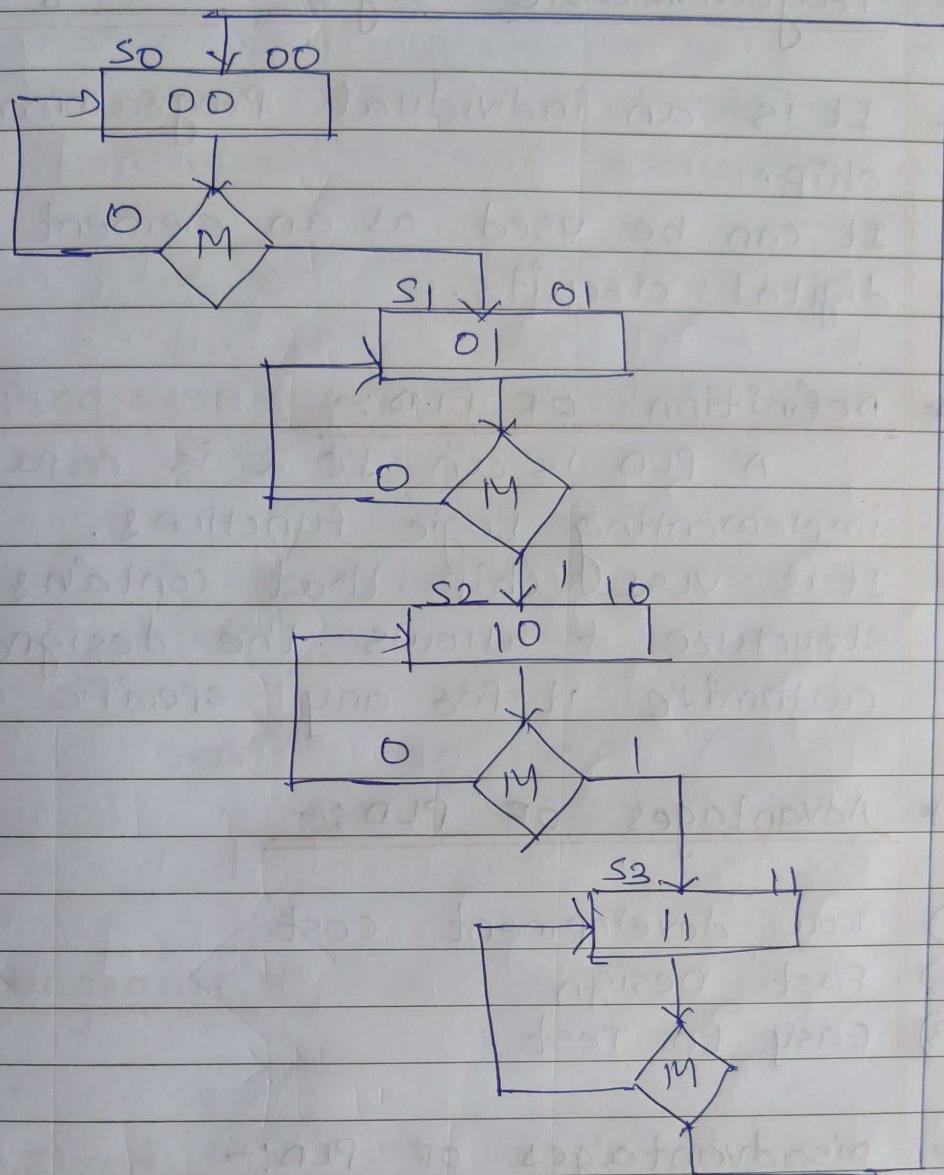


ii) State diagram:-



up counter takes
Place for
 $M=1$ & the counter
progresses from
states S_0, S_1, S_2, S_3

(3) ASM chart :-



* Programmable Logic Device :- (PLD)

- It is an individual Programmable electronic chip.
It can be used as an element to build digital circuits.

①
②
③

* Definition of PLD:-

- A PLD is an IC & is capable of implementing logic functions.
- It is VLSI chip that contains a regular structure & allows the designer to customize it for any specific application.

①

* Advantages of PLD:-

- 1) Low development cost
- 2) Fast design
- 3) Easy to test.

* Disadvantages of PLD:-

- 1) ~~too~~ development lack of security
- 2) Large board space
- 3) Large power requirements.
- 4) Additional cost, space, power requirements to modify the circuit.

* B

n in

- R
C

* Types of PLD:

- ① PROM (Programmable Read only Memory)
- ② PAL (Programmable Array Logic)
- ③ PLA (Programmable Logic Array)

① PROM (Programmable Read only Memory) (ROM)

- It has a fixed AND array (constructed on a decoder) & programmable connections for the output OR array.
- The input lines to the AND array are hard-wired & the output lines to the OR array.

* Block diagram of PROM :-

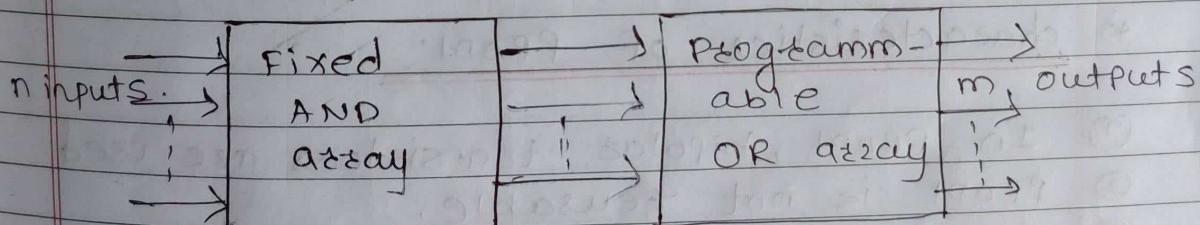


fig: PROM.

- ROM is a basic memory unit of any computer system.

- PROM is a type of ROM.
- ROM is a non-volatile memory, where the stored data cannot be erased even when the power is turned off.
- In PROM, where the data can be stored & the stored data can be changed by ~~EEPROM~~ programming the device.

* Features of PROM:-

- 1) used in digital electronic devices to store permanent data.
- 2) PROM are manufactured blank & depending on technology, can be programmed at final test or in system.

* Characteristics of PROM:-

- ① In PROM, bipolar transistors are used
- ② PROM is not reusable.
- ③ PROM is expensive.
- ④ storage endurance of PROM is high.

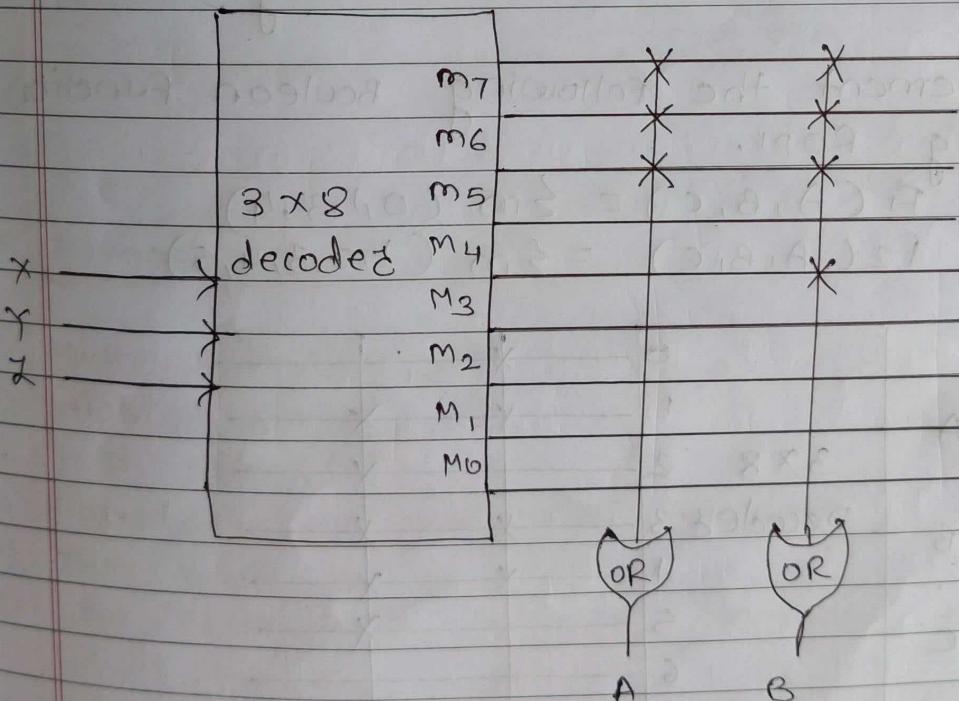
Q. Implement the following Boolean functions using PROM.

$$A(X, Y, Z) = \sum m(5, 6, 7)$$

$$B(X, Y, Z) = \sum m(3, 5, 6, 7)$$

→ Step 1 - 3 variables X, Y, Z . So, we require a 3 to 8 decoder & two programmable OR gates for producing two functions.

Step 2 →



Step 3 →

Here 3×8 decoded generates each min terms.

The two programmable OR gates have the access of all three min terms.

But, only the required min terms are propagated in order to produce the respective Boolean functions by each OR gate.

* Practice Examples

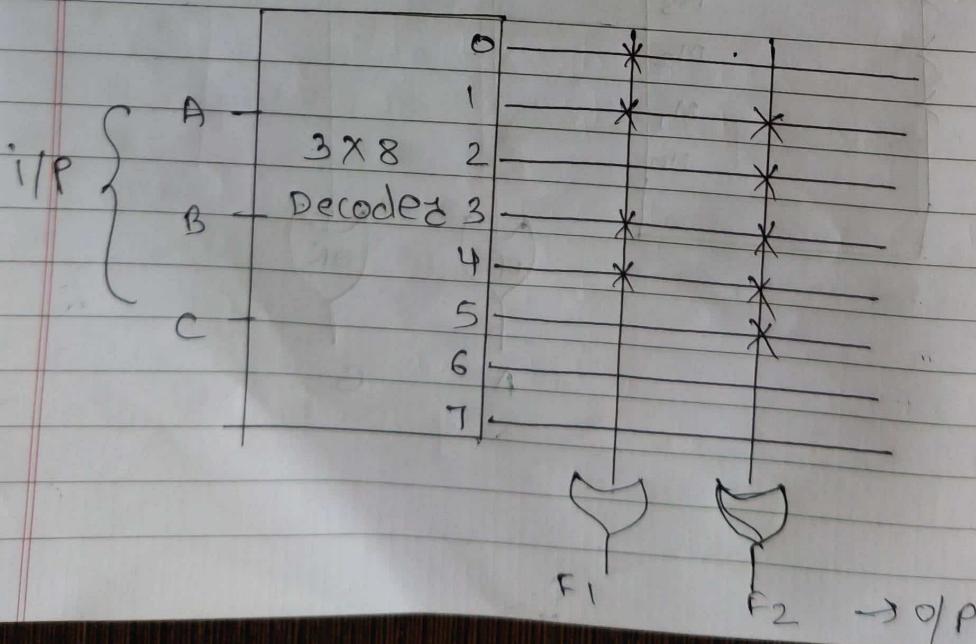
① Implement a full adder using ROM.

② Implement the following Boolean function using ROM.

$$F_1(A, B, C) = \sum m(0, 1, 3, 4)$$

$$F_2(A, B, C) = \sum m(1, 2, 3, 4, 5)$$

→



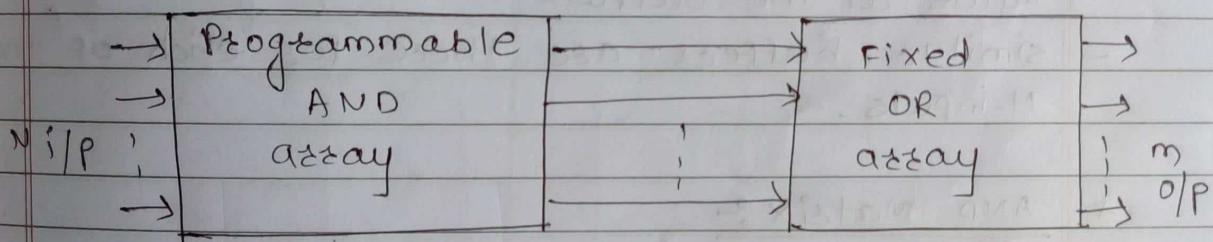
PAL

* Programmable Array Logic Device:- (PAL)

- The PAL is a Programmable logic device that has Programmable AND array & fixed OR array.

Because only the AND array is programmable, it is easier to use but not flexible as compared to PLA.

* Block diagram of PAL



* Advantages of PAL:-

- 1) Highly efficient
- 2) Highly secure
- 3) High reliability
- 4) They are more flexible to plan.
- 5) They need low power.
- 6) Low production cost as compared to PLA.

* Disadvantages of PAL:-

- 1) Speed is low
- 2) They are expensive.

* Input Buffer :-

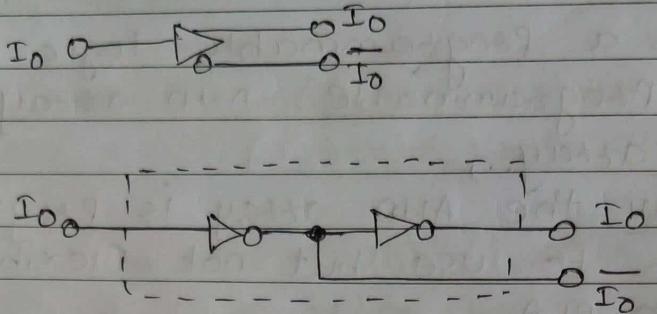


Fig: Input Buffer. [for one i/p]

- It produces inverted as well as non-inverted inputs at the outputs.
- similar buffers are there for each of the M-inputs.

* AND Matrix :-

- An AND matrix is used to form product terms
- It has n AND gates with outputs P_0 through P_{n-1}
- 2M inputs (I_0 through I_{M-1} & \bar{I}_0 through \bar{I}_{M-1}) for each AND gate.
- Each AND gate generates one product term which is given by

$$P = I_0 \cdot \bar{I}_0 \cdot I_1 \cdot \bar{I}_1 \cdots I_{M-1} \cdot \bar{I}_{M-1}$$

- * OR
- The for the
- OR means
- Output

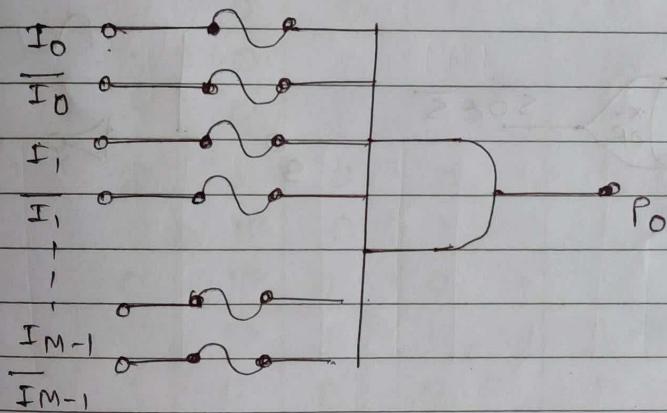
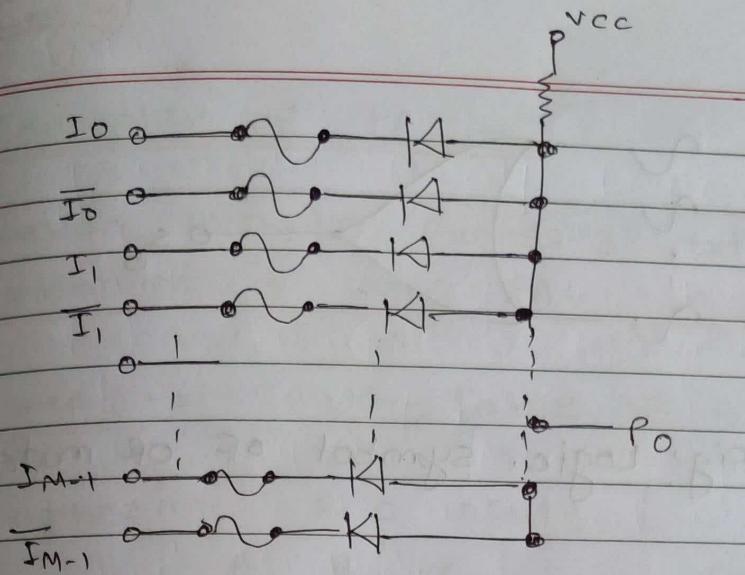


Fig: AND Matrix.

* OR Matrix :-

- The OR Matrix is used to produce the logical sum of the product term outputs of the AND matrix.
- OR matrix (gate) consist of parallel connected transistors with a common emitter load.
- Outputs are obtained at S_0 through S_{n-1}

$$\left. \begin{aligned} S_0 &= P_0 + P_1 + \dots + P_{n-1} \end{aligned} \right\}$$

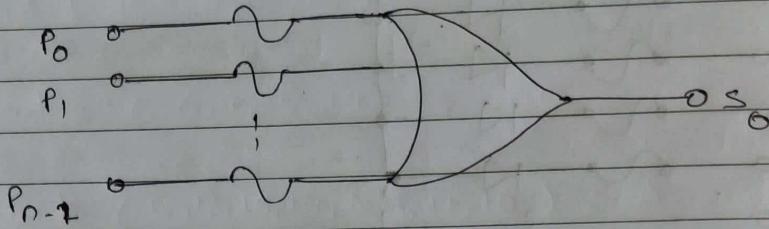
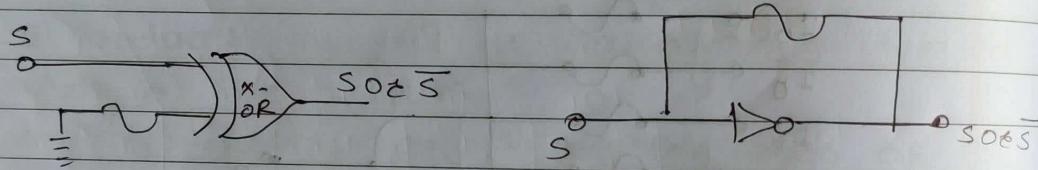


Fig: Logic symbol of OR matrix.

* Invert / Non Invert matrix.



Examples of PAL:-

- ① Design BCD to Excess-3 code converted & implement it using PAL.

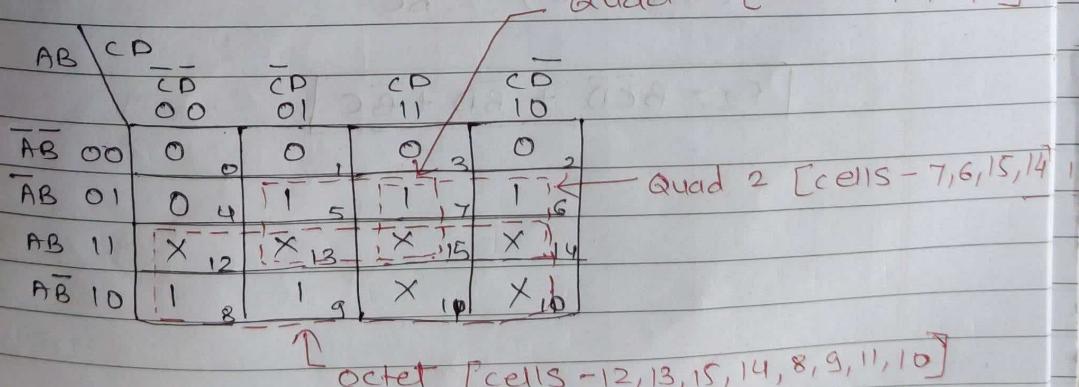
→ Step 1 → Truth Table of BCD to Excess-3

Number	BCD inputs				Excess-3 outputs				Outputs
	A	B	C	D	F ₁	F ₂	F ₃	f ₄	
0	0	0	0	0	0	0	1	1	1
1	0	0	0	1	0	1	0	0	0
2	0	0	1	0	0	1	0	0	1
3	0	0	1	1	0	1	1	0	0
4	0	1	0	0	0	1	0	1	1
5	0	1	0	1	1	0	0	0	0
6	0	1	1	0	1	0	1	0	1
7	0	1	1	1	1	0	1	0	0
8	1	0	0	0	1	0	1	1	1
9	1	0	0	1	1	1	0	0	0

Step 2 → K-map

K-map for F₁

quad 1 [cells - 5, 7, 13, 15]



* Examples of PAL:-

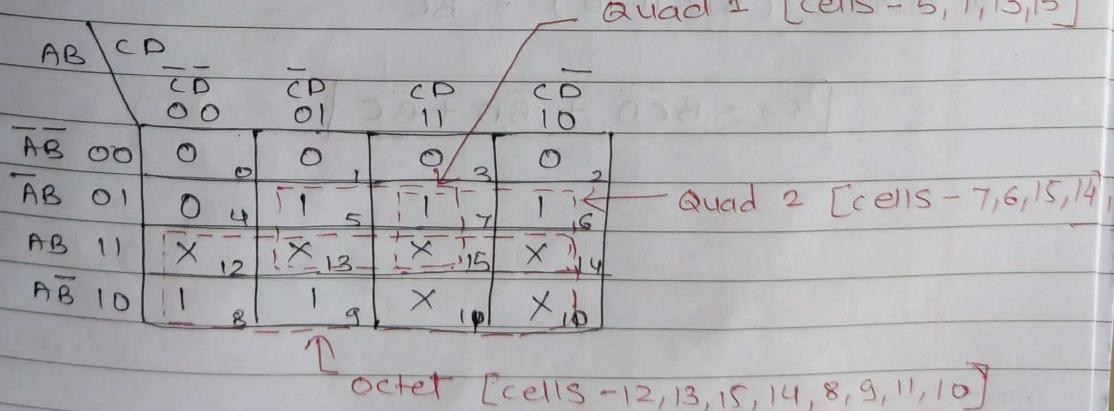
- ① Design BCD to Excess-3 code converted & implement it using PAL.

→ Step 1 → Truth Table of BCD to Excess-3

Number	BCD inputs				Excess-3 outputs			
	A	B	C	D	F ₁	F ₂	F ₃	F ₄
0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Step 2 → K-map

K-map for F₀ & F₁



① Quad 1 (5, 7, 13, 15) = BD

② Quad 2 (6, 7, 15, 14) = BC

③ Octet (12, 13, 15, 14, 8, 9, 11, 10) = A

$$F_1 = A + BD + BC$$

④ K-map for F_2 :-

		AB \ CD	Quad 1 (1, 3, 9, 11)				Quad 2 (3, 2, 11, 10)			
		00	01	11	101					
		00	0	1	1	1				
		01	X	0	0	0	1	0	1	0
		11	X	X	X	X	X	X	X	X
Pair (4, 12)		10	0	1	1	1	X	1	0	1
		11	X	X	X	X	X	X	X	X

① Pair 1 (4, 12) = $B\bar{C}\bar{D}$

② Quad 1 (1, 3, 9, 11) = $\bar{B}D$

③ Quad 2 (3, 2, 11, 10) = $\bar{B}C$

$$F_2 = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C$$

③ K-map for F_3 :-

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
	00	00	01	11	10
$\bar{A}\bar{B}$ 00	11	0	11	0	
$\bar{A}\bar{B}$ 01	11	0	11	0	
$A\bar{B}$ 11	X	X	1X1	X	
$A\bar{B}$ 10	11	0	1X1	X	

↳ Quad 1 (0, 4, 12, 8)

↳ Quad 2 (3, 7, 15, 11)

① Quad 1 (0, 4, 12, 8) = $\bar{C}\bar{D}$

② Quad 2 (3, 7, 15, 11) = CD

$$F_3 = \bar{C}\bar{D} + CD$$

④ K-map for F_4 :-

AB	CD	$\bar{A}\bar{B}$	$\bar{C}\bar{D}$	CD	$C\bar{D}$
	00	00	01	11	10
$\bar{A}\bar{B}$ 00	11	0	0	11	
$\bar{A}\bar{B}$ 01	11	0	0	11	
$A\bar{B}$ 11	X	X	X	1X	
$A\bar{B}$ 10	11	0	X	1X	

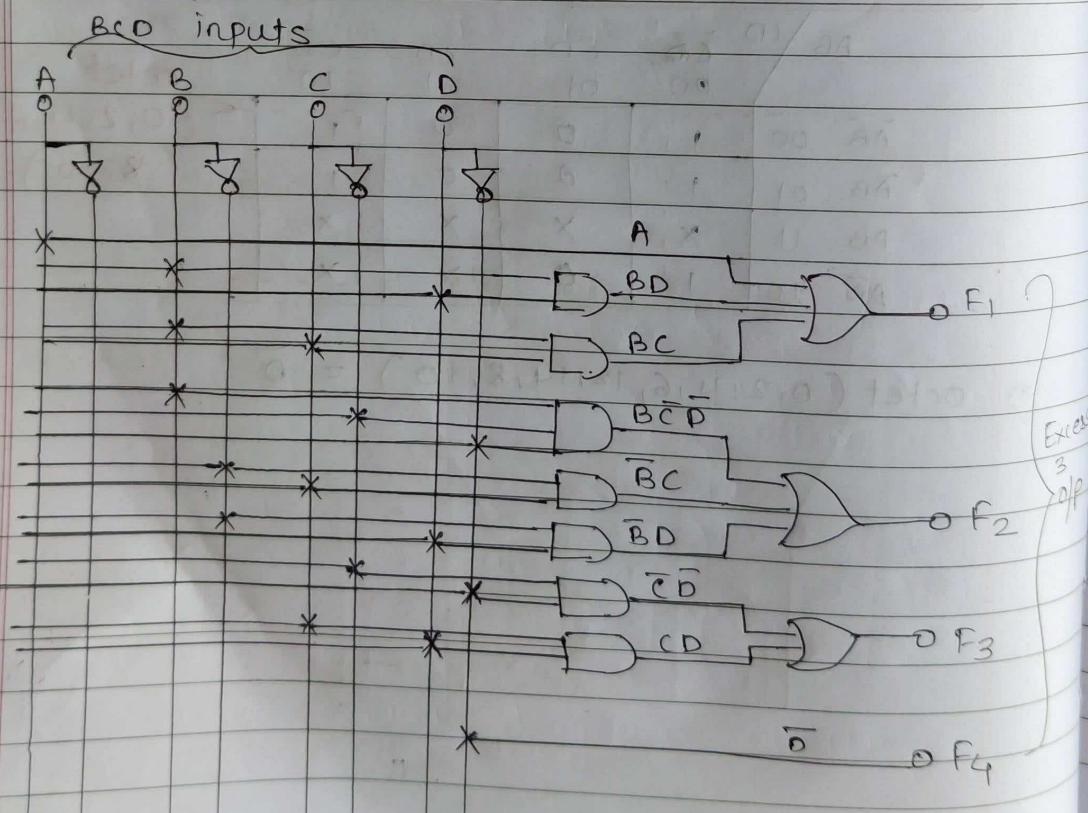
Octet (0, 2, 4, 6, 12, 14, 8, 10)

① Octet (0, 2, 4, 6, 12, 14, 8, 10) = \bar{D}

* Step 3 → PAL Programming Table

Product Term	Inputs				outputs
	A	B	C	D	
A	1	-	-	-	f1
BD	-	1	-	1	
BC	-	1	1	-	
$\bar{B}\bar{C}\bar{D}$	-	1	0	0	f2
$\bar{B}C$	-	0	1	-	
$\bar{B}D$	-	0	-	1	
$\bar{C}\bar{D}$	-	-	0	0	f3
CD	-	-	1	1	
\bar{D}	-	1	-	0	f4

* Step 4 - Implementation of PAL



Practice Examples:-

- ① Implement the following function using PAL

$$F_1(A, B, C, D) = \sum m (1, 3, 4, 6, 9, 12, 14)$$

$$F_2(A, B, C, D) = \sum m (1, 2, 3, 7, 12, 15)$$

② $W(A, B, C, D) = \sum m (2, 12, 13)$

$$X(A, B, C, D) = \sum m (7, 8, 9, 16, 11, 12, 13, 14, 15)$$

- ③ Implement Following Boolean Function using PAL.

$$F_1 = \sum m (0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$F_2 = \sum m (1, 2, 8, 12, 13)$$

Excess

3
of PF₂F₃F₄

PLA* Programmable logic Array :- (PLA)

- PLA is a programmable logic device that has both programmable AND array & programmable OR array.

- It is flexible PLD.
- PLA consist of two level AND-OR circuit on single chip.

AND gate - Product term

OR gate - sum of these product term

- It has M inputs, n Product terms & N outputs.

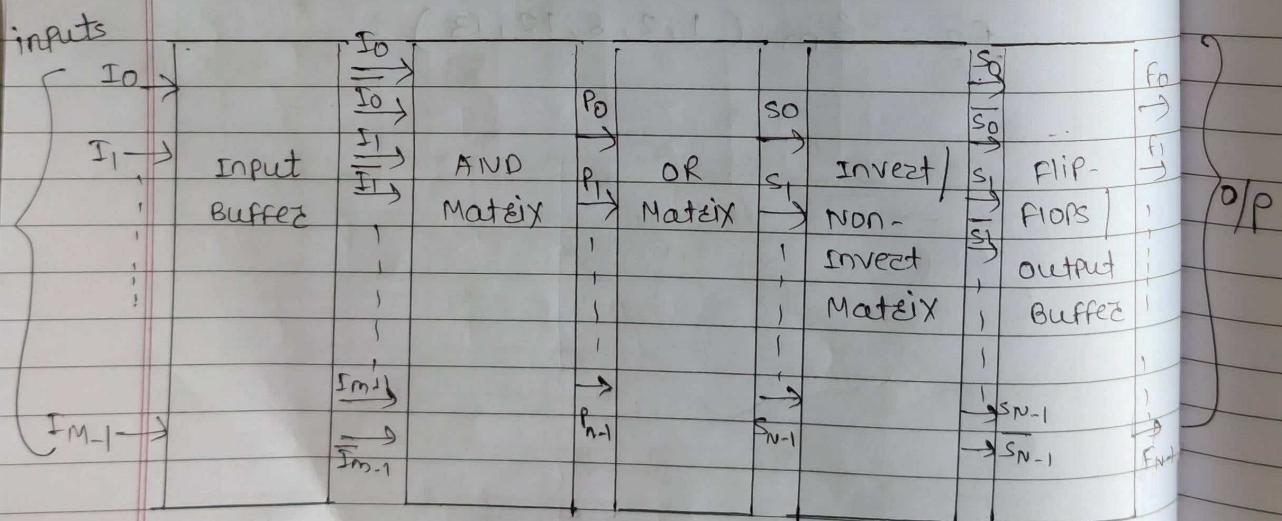
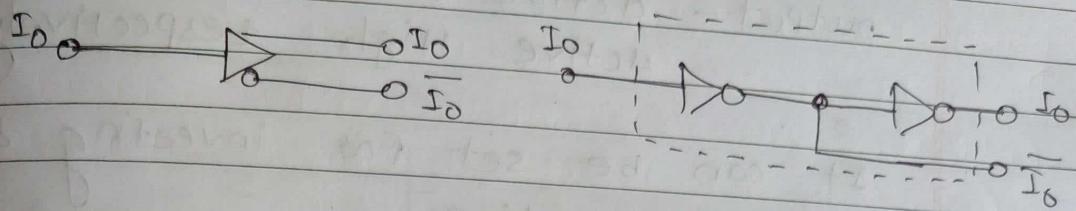
* Block diagram of PLA:-

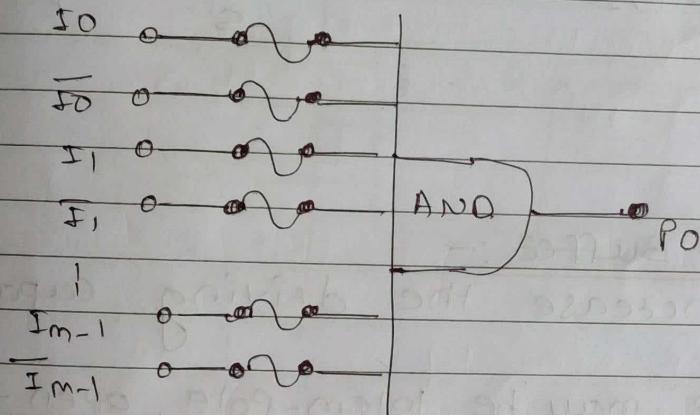
Fig: Block diagram of PLA.

① 1st BLOCK - Input Buffer.



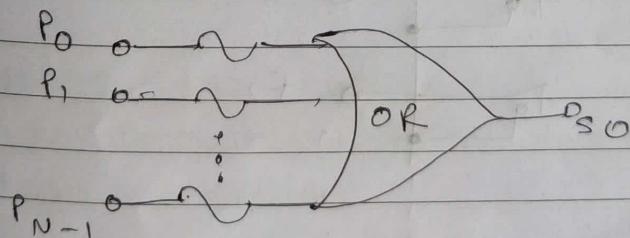
② 2nd Block - AND matrix

(I_0 through I_{M-1} & I_0 through I_{M-1})



③ 3rd BLOCK - OR Matrix

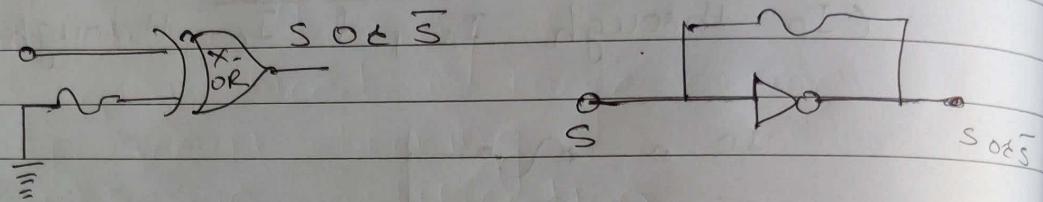
$$S_0 = P_0 + P_1 + \dots + P_{N-1}$$



4) 4th BLOCK - Invert | Non Invert Matrix

output - active low or
active High respectively.

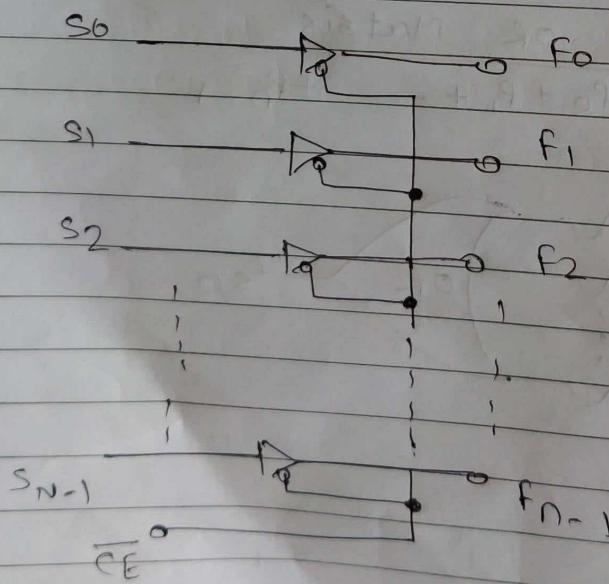
It can be set for inverting & non-inverting operation.



5) Output Buffered :-

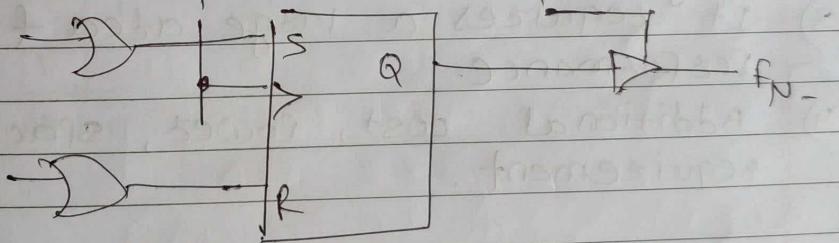
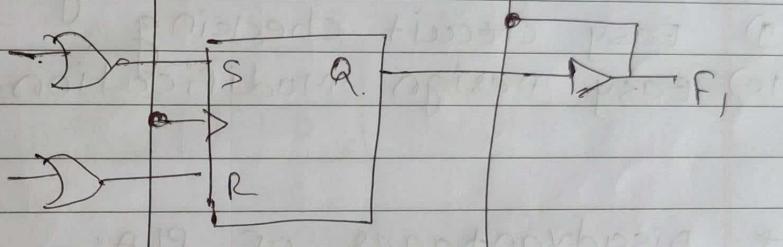
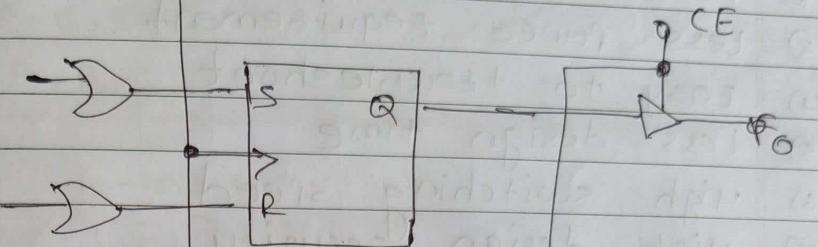
- increase the driving capability of PLA.

The OLP may be totem-Pole, open-collector, or three state.



output through flip-flops & buffers.

clock



* Applications of PLA :-

- 1) Used to implement combinational & sequential logic circuits.
- 2) Used as a counter.
- 3) Used as a decoder.
- 4) Provide control over datapath.

* Advantages OF PLA:-

- 1) Low development cost.
- 2) Less space requirement
- 3) Less power requirement.
- 4) Easy to troubleshoot.
- 5) Less design time
- 6) High switching speed
- 7) High design security.
- 8) can be produced in large volume.
- 9) Easy circuit checking.
- 10) Easy Design modification.

* Disadvantages OF PLA:-

- 1) Lack of security
- 2) Large power requirement
- 3) It requires a large area of good performance.
- 4) Additional cost, power, space requirement.

Examples:-

1) Implement the following Boolean function using PLA.

$$F_1(A, B, C) = \sum m(0, 1, 3, 4)$$

$$F_2(A, B, C) = \sum m(1, 2, 3, 4, 5)$$

→ Step 1 :- k-map (F1)

		BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$	
		00	01	11	10		
		$\bar{A}0$	1 0	1 1	1 3	0 2	$F_1 = \bar{B}\bar{C} + \bar{A}C$
		$A1$	1 4	0 5	0 7	0 6	
			↓ $\bar{B}\bar{C}$	↓ $\bar{A}C$			

(F2)

		BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$	
		00	01	11	10		
		$\bar{A}0$	0 0	1 1	1 3	1 2	$F_2 = A\bar{B} + \bar{A}C + \bar{A}B$
		$A1$	1 4	1 5	0 7	0 6	
			↓ $A\bar{B}$	↓ $\bar{A}C$	↓ $\bar{A}B$		

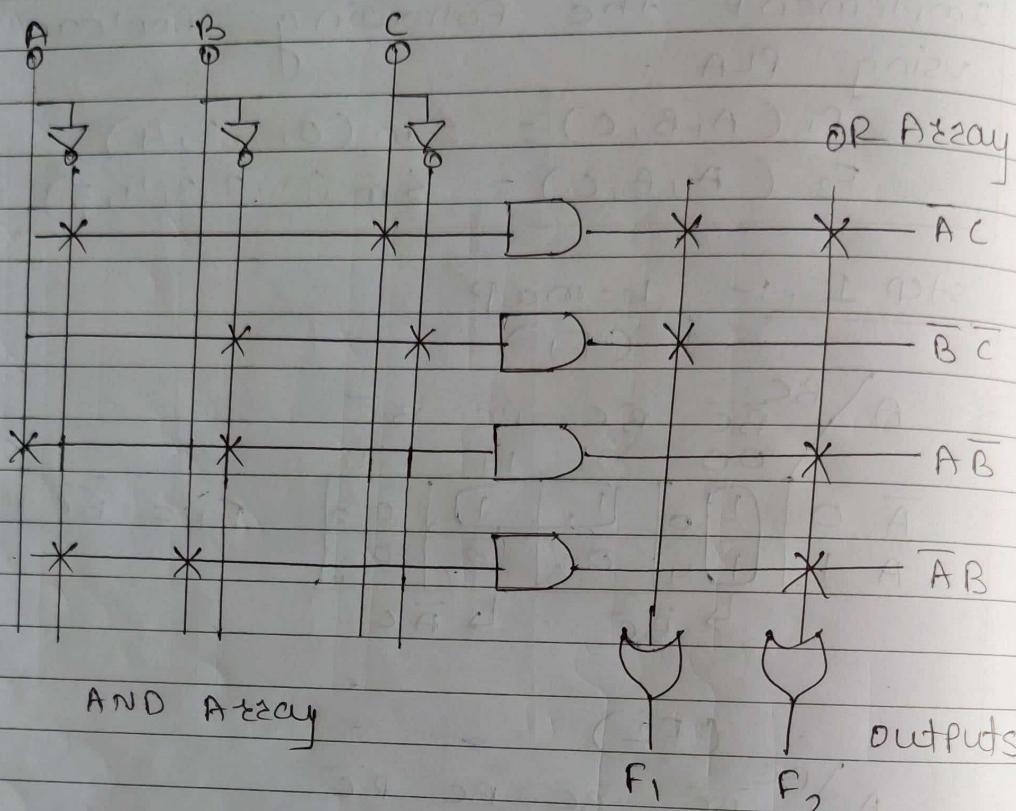
Step 2 → Programming Table of PLA

Product Term	Inputs			Outputs	
	A	B	C	F_1	F_2
$\bar{A}C$	0	-	1	1	1
$\bar{B}\bar{C}$	-	0	0	0	-
$A\bar{B}$	1	0	-	-	1
$\bar{A}B$	0	1	-	-	1

F1+F2

* Step 3 → PLA Implementation

Inputs



② Implement the following Boolean function using $3 \times 4 \times 2$ PLA.

$$(A, B, C) F_1 = \sum m(0, 1, 2, 4)$$

$$(A, B, C) F_2 = \sum m(0, 5, 6, 7)$$

→ Step 1 → K-map.

(F_1)

A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	0	0

[$3 \times 4 \times 2$]
↑
Product Term
OP

$$F_1 = \bar{B}\bar{C} + \bar{A}\bar{B} + \bar{A}\bar{C}$$

K-map (F_2)

		BC				
		00	01	11	10	
		0	1	0	0	0
		1	0	1	1	0

$$F_2 = AC + AB + \bar{A}\bar{B}\bar{C}$$

③ K-map for \bar{F}_1

A \ BC

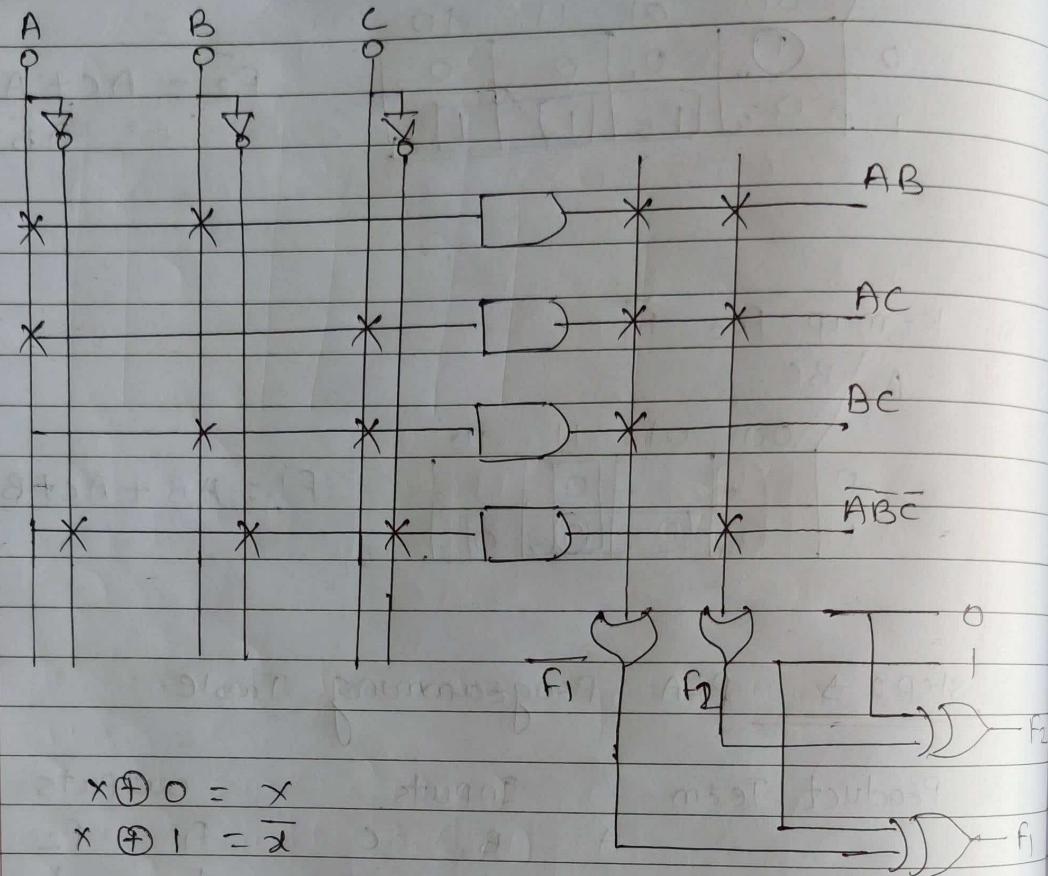
		00	01	11	10	
		0	1	0	3	2
		1	0	0	7	6

$$F_1 = AB + AC + BC$$

Step 2 → PLA Programming Table.

Product Term	Inputs			outputs	
	A	B	C	F_1	F_2
AB	1	1	-	1	1
AC	1	-	1	1	1
BC	-	1	1	1	-
$\bar{A}\bar{B}\bar{C}$	0	0	0	-	1

* PLA Implementation!

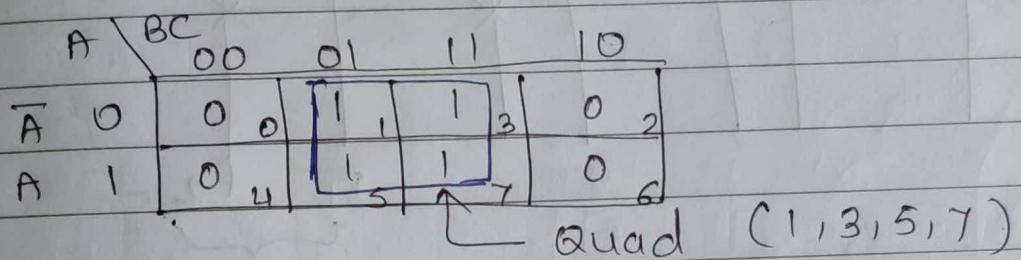


② Implement the following functions using PLA

$$F_1 = \sum m(1, 3, 5, 7)$$

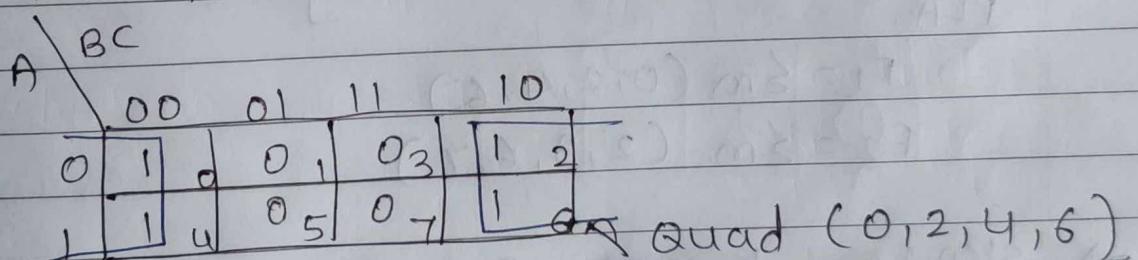
$$F_2 = \sum m(0, 2, 4, 6)$$

→ Step 1 - K-map for F_1



$$F_1 = C$$

K-map for F_2

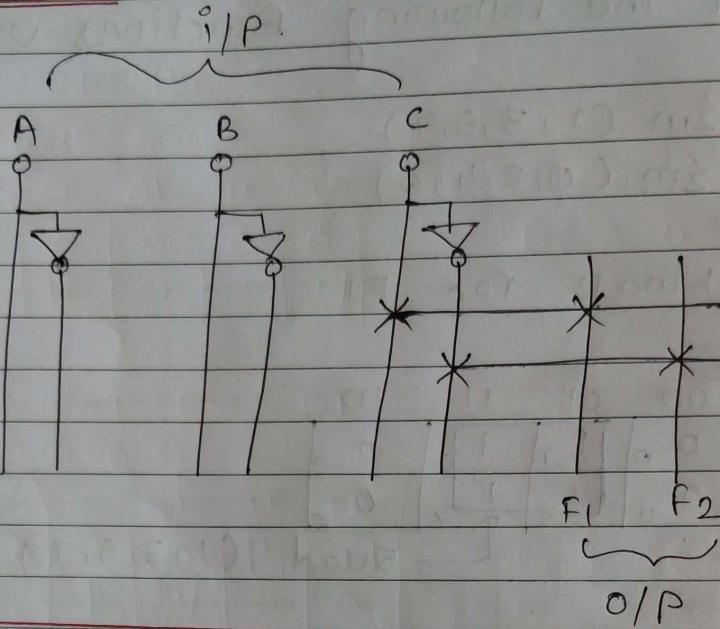


$$F_2 = \bar{C}$$

Step 2 - Programming Table of PLA

Product Term	Inputs			Outputs	
	A	B	C	F_1	F_2
$\frac{C}{\bar{C}}$	-	-	1	1	0
	-	-	0	0	1

* Step 3 - Implementation of PLA.

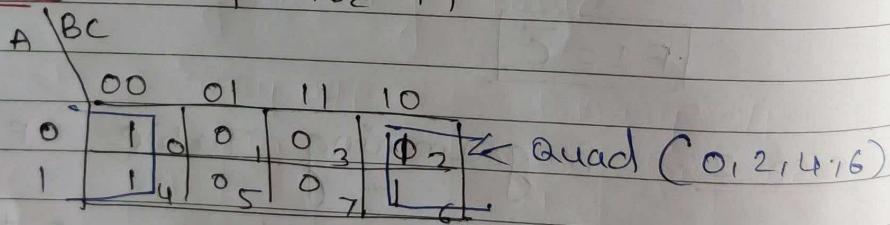


④ Implement the following functions using PLA.

$$F_1 = \sum m(0, 2, 4, 6)$$

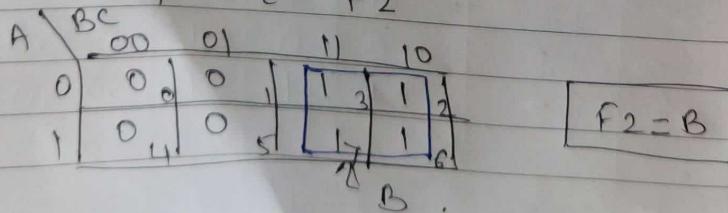
$$F_2 = \sum m(2, 3, 6, 7)$$

→ Step 1 - K-map for F₁



$$F_1 = \bar{C}$$

* K-map for F₂



⑤ Imp
conv

→ Decim

0
1
3
2
6
7
5
4

Step 2 - Programming table

Product team

inputs

outputs

 \bar{C} \ominus

-

0

1

0

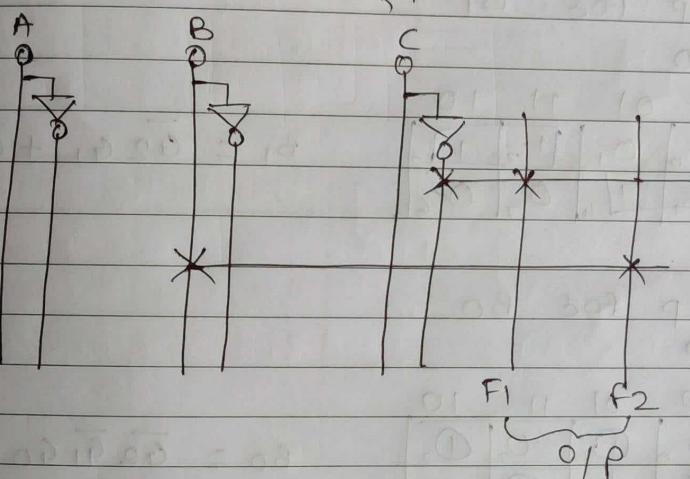
B

-

1

0

1

Step 3 - Implementation

- ⑤ Implement 3 bit binary to gray code converter using PLA.

Truth Table

Decimal	Gray			i/P	Binary O/P		
	G ₂	G ₁	G ₀		B ₂	B ₁	B ₀
0	0	0	0		0	0	0
1	0	0	1		0	0	1
3	0	1	1		0	1	0
2	0	1	0		0	1	1
6	1	1	0		1	0	0
7	1	1	1		1	0	1
5	1	0	1		1	1	0
4	1	0	0		1	1	1

* Step 2 - K-map

K-map for B_2

		G ₁ , G ₀			
		00	01	11	10
G ₂	0	0, 0	0, 1	0, 2	0, 2
	1	1, 0	1, 1	1, 1	1, 0

$$B_2 = G_2$$

K-map for B_1

		G ₁ , G ₀			
		00	01	11	10
G ₂	0	0, 0	0, 1	1, 1	1, 1
	1	1, 1	1, 0	0, 1	0, 0

$$B_1 = \overline{G}_2 G_1 + G_2 \overline{G}_1$$

K-map for B_0

		G ₁ , G ₀			
		00	01	11	10
G ₂	0	0, 0	0, 1	0, 2	0, 2
	1	1, 1	0, 0	0, 1	0, 0

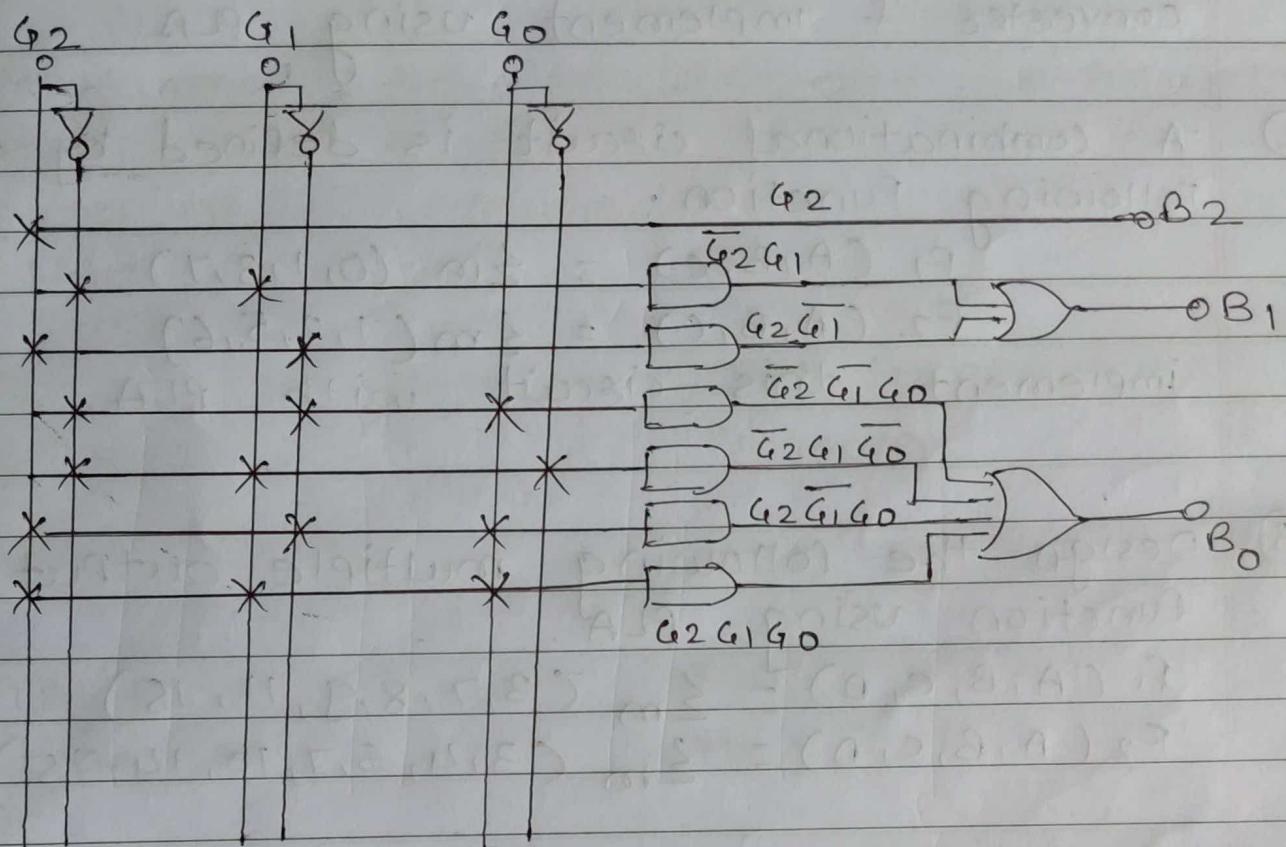
$$B_0 = \overline{G}_2 \overline{G}_1 G_0 + \overline{G}_2 G_1 \overline{G}_0 + G_2 \overline{G}_1 G_0 + G_2 G_1 G_0$$

* Step 3 - Programming Table

Decimal	Gray i/P			Binary o/P		
	G ₂	G ₁	G ₀	B ₂	B ₁	B ₀
G ₂	1	-	-	1	-	-
G ₂ G ₁	0	1	-	-	1	-
G ₂ G ₁	1	0	-	-	1	-
G ₂ G ₁ G ₀	0	0	1	-	1	-
G ₂ G ₁ G ₀	0	1	0	-	-	1
G ₂ G ₁ G ₀	1	0	0	-	-	1
G ₂ G ₁ G ₀	1	1	1	-	-	1

* Step 4 - Implementation

Inputs:



Practice Examples.

- ① Design of BCD to Excess-3 code converter & implement using PLA.

- ② A combinational circuit is defined by the following function:

$$F_1(A, B, C) = \sum m(0, 1, 3, 7)$$

$$F_2(A, B, C) = \sum m(1, 2, 5, 6)$$

implement this circuit with PLA.

- ③ Design the following multiple outputs function using PLA.

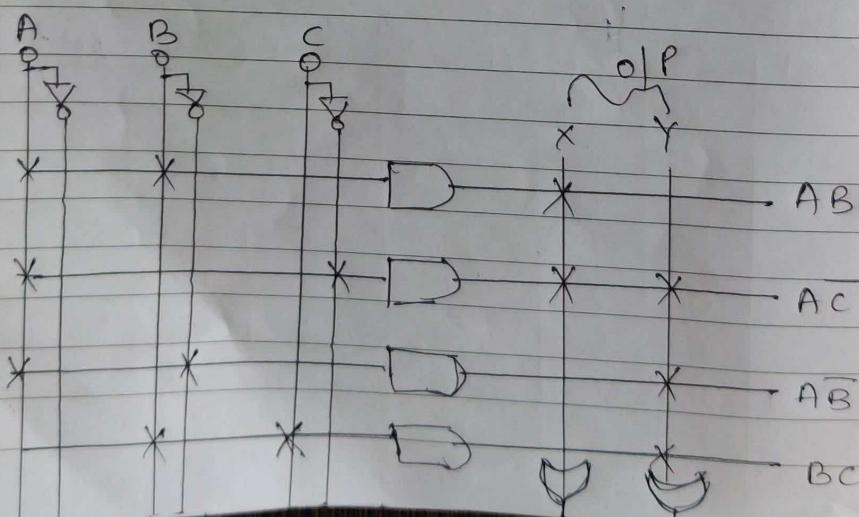
$$F_1(A, B, C, D) = \sum m(3, 7, 8, 9, 11, 15)$$

$$F_2(A, B, C, D) = \sum m(3, 4, 5, 7, 10, 14, 15)$$

- ④ Implement the following Boolean expression with the help of PLA.

$$X = AB + A\bar{C}$$

$$Y = A\bar{B} + BC + A\bar{C}$$



* Difference between PLA & PAL

PLA	PAL
Programmable Logic Array	Programmable Array Logic
1) Both AND & OR gates are programmable	1) AND gate is programmable & OR gate is fixed.
2) PLA is not easier to program.	2) PAL is easier to program.
3) PLA is flexible	3) PAL is not flexible
4) PLA speed is lower than PAL	4) PAL's speed is higher than PLA
5) complexity of PLA is high.	5) complexity of PAL is less.
6) cost is high	6) cost is low
7) It is less used than PAL	7) It is more used than PLA.
8) No limitation of number of AND gates.	8) Limitation is number of AND gates.