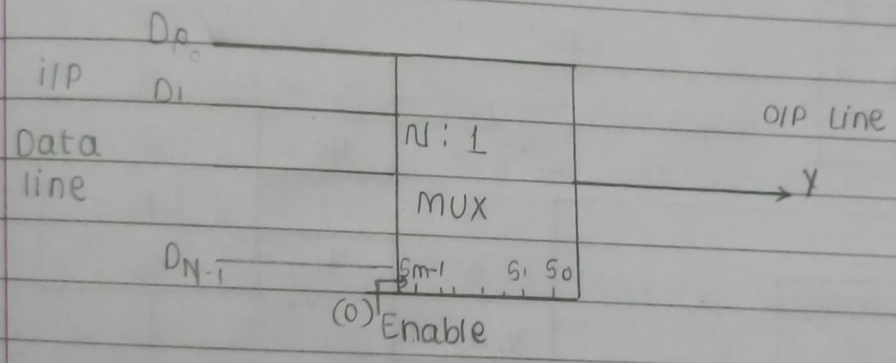
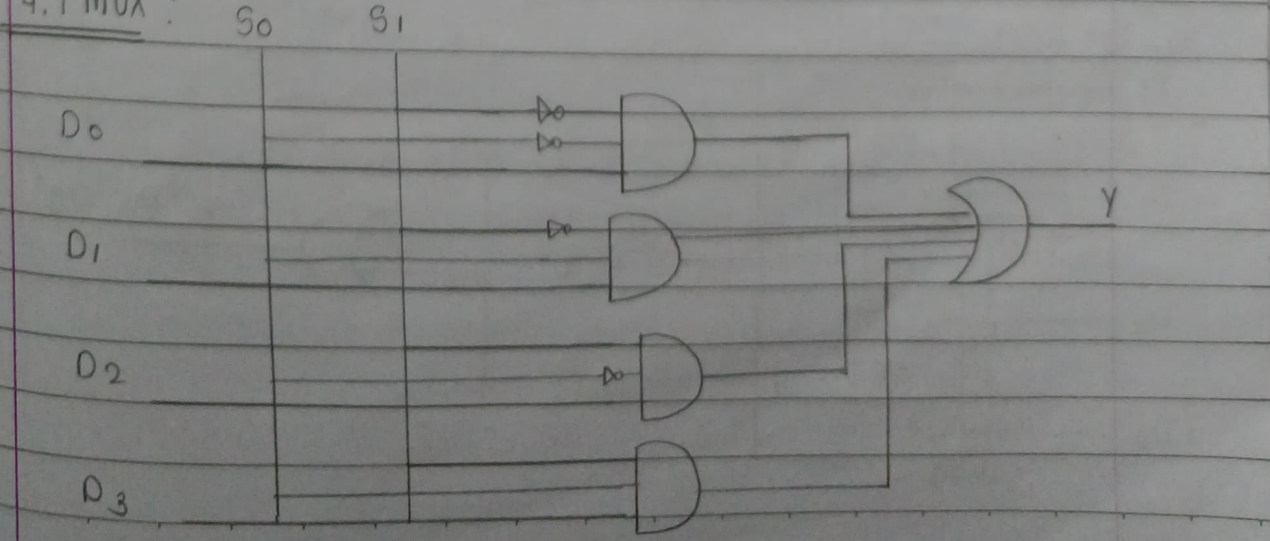
* N: 1 MUX



N = No. of data lines.
m = No. of Select lines.
$N = 2^m$

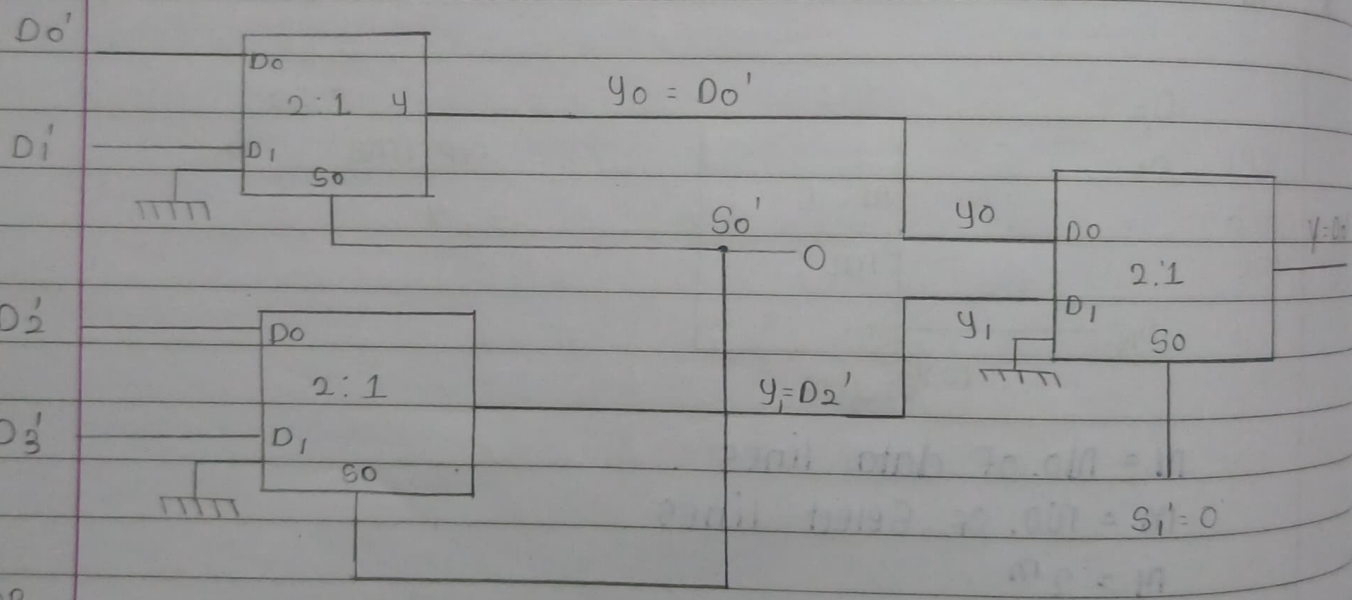
4:1 MUX :



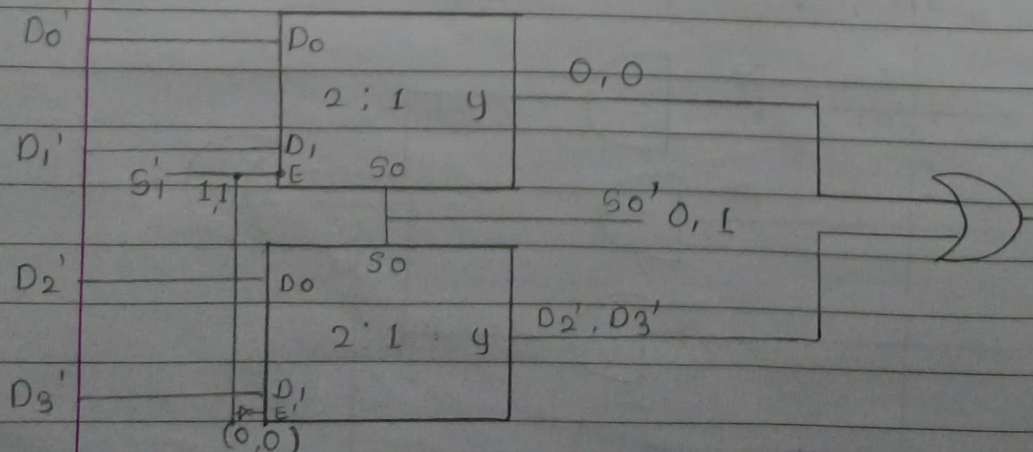
Truthtable :

E	S_1	S_0	Y
1	X	X	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3

Que Design 4:1 multiplexer using 2:1 (cascading)



OR



Que 2] Design 8:1 multiplexer using 4:1 multiplexer

Imp Que 3] Implement given boolean expression using multiplexer.

@ $f(A,B,C) = A\bar{B} + Bc + \bar{A}\bar{C}$

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

	A	B	C	Y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

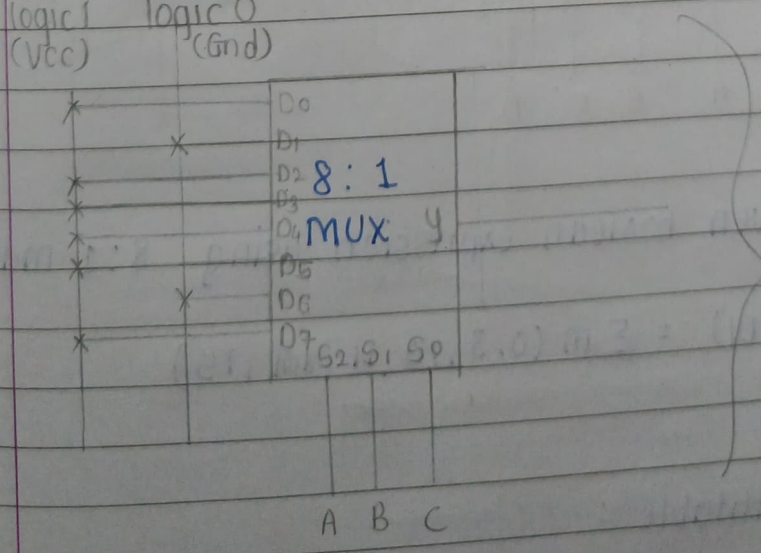
$\therefore f(A,B,C) = A\bar{B} + Bc + \bar{A}\bar{C}$

$= \Sigma m(0,2,3,4,5,7)$

No. of variables = 3

No. of conditions = $2^3 = 8$

$\therefore$ 8:1 MUX is required.



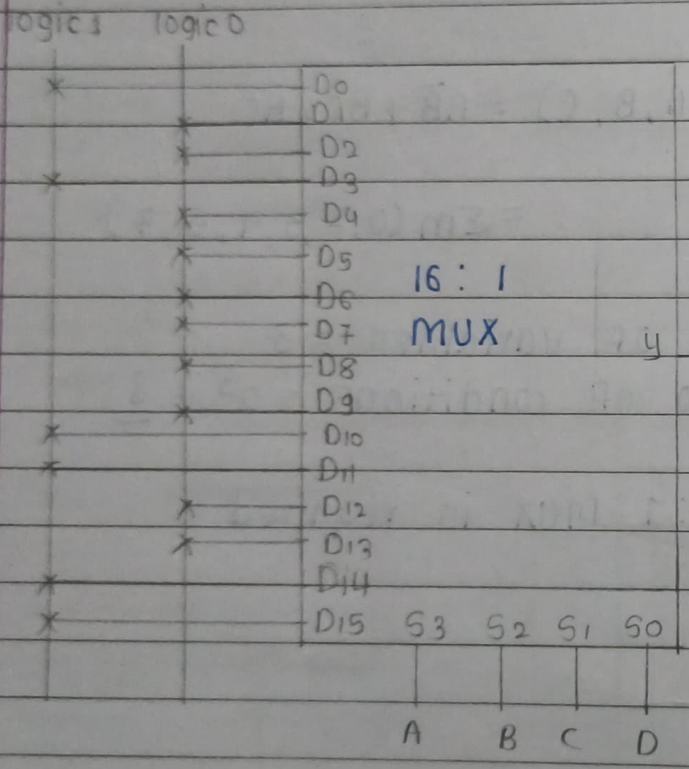
ⓑ Implement given boolean expression using multiplexer

$y = f(A, B, C, D) = \Sigma m(0, 3, 10, 11, 14, 15)$

→ No. of variables = 4

No. of conditions $= 2^4 = 16$

∴ 16 : 1 MUX is required.



ⓒ Implement given boolean expression using 8:1 multiplexer

$y = f(A, B, C, D) = \Sigma m(0, 3, 10, 11, 14, 15)$

→ Step 1 :

Write Truthtable :

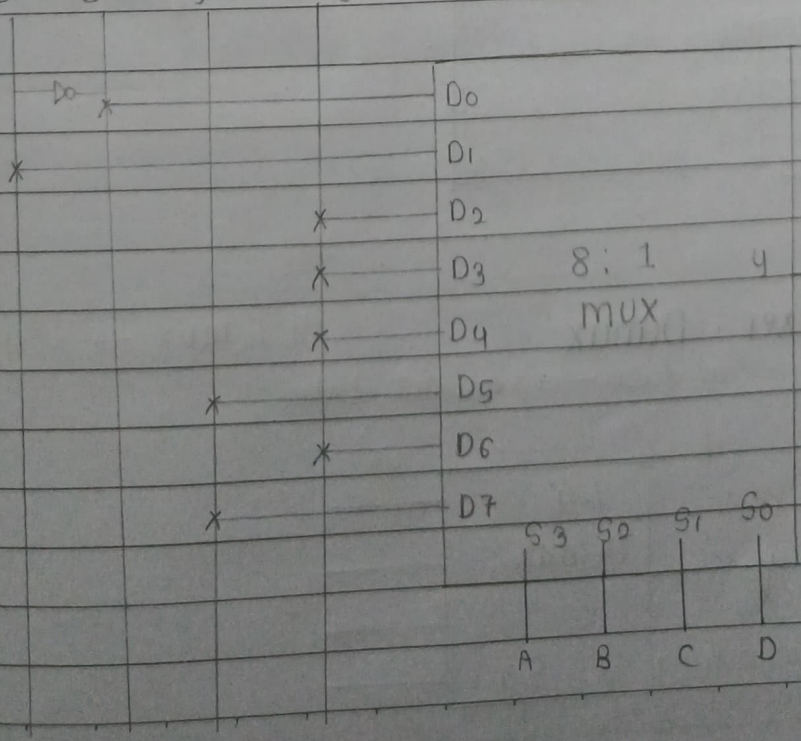
	A	B	C	D	Y	y
0	0	0	0	0	1	$\bar{D}$
1	0	0	0	1	0	
2	0	0	1	0	0	D
3	0	0	1	1	1	
4	0	1	0	0	0	0
5	0	1	0	1	0	
6	0	1	1	0	0	0
7	0	1	1	1	0	
8	1	0	0	0	0	0
9	1	0	0	1	0	
10	1	0	1	0	1	1
11	1	0	1	1	1	
12	1	1	0	0	0	0
13	1	1	0	1	0	
14	1	1	1	0	1	1
15	1	1	1	1	1	

Step 2: Make group of 2 of Y o/p.

Step 3: Draw 8:1 MUX

(Vcc) (Gnd)

D $\bar{D}$ logic 1 logic 0

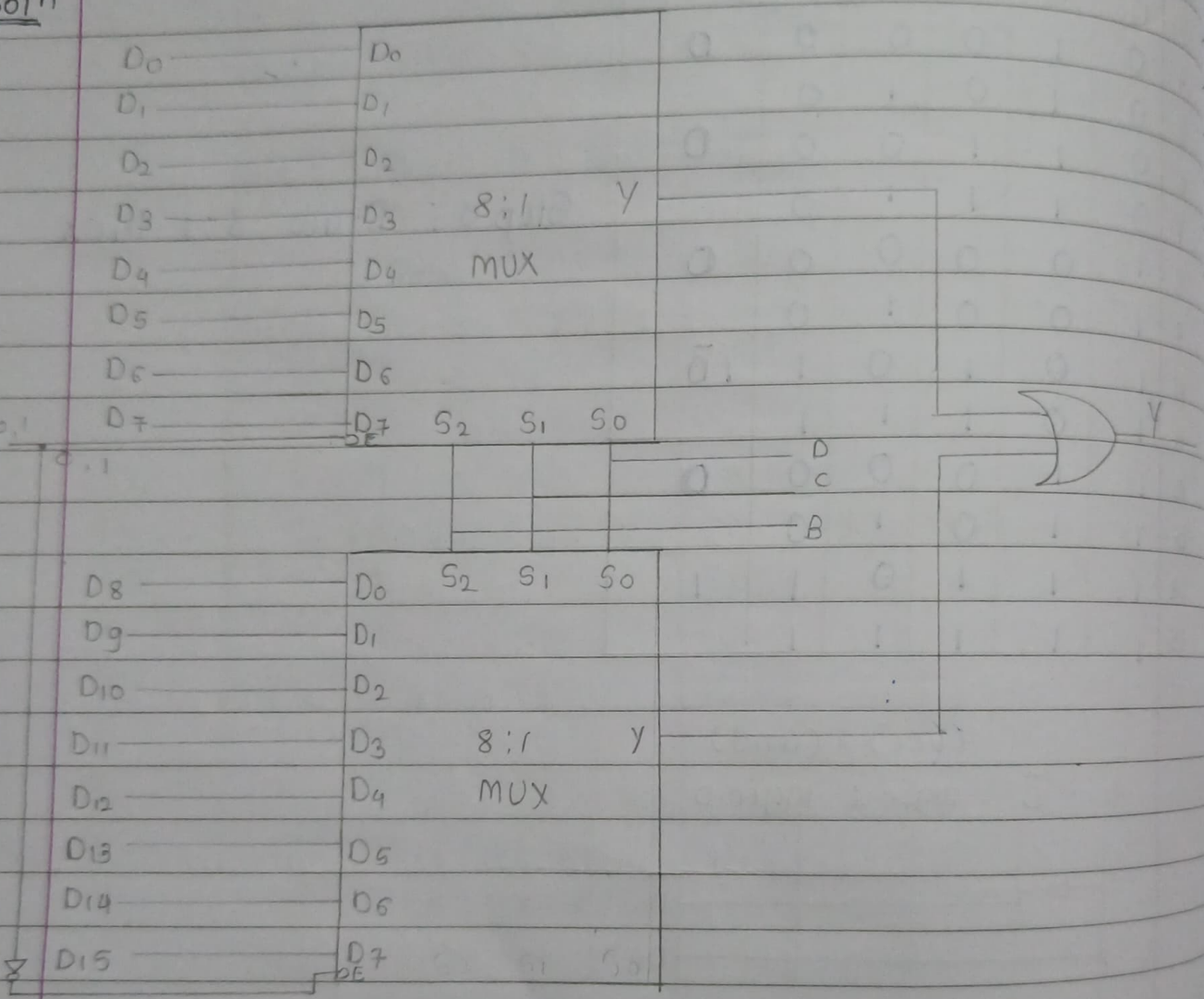


y = f(A, B, C, D)

Page No.:
Date: 16/10/23

Que 4 variable function ① 16 : 1 MUX .
4 variable function ① 8 : 1 MUX
4 variable function ② 8 : 1 MUX . → 16 : 1 → cascading

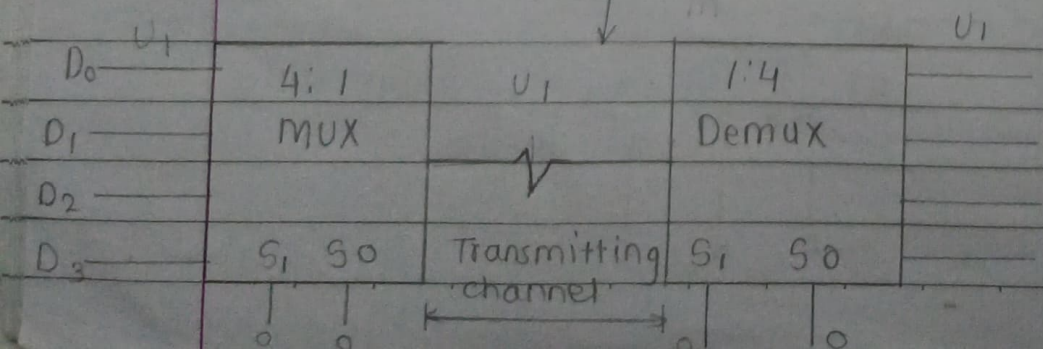
Boln



Demultiplexer : Demux

4:1 MUX → 1:4 Demux

Parallel data → serial data → Parallel data



NAND : A B $Y = \overline{AB}$

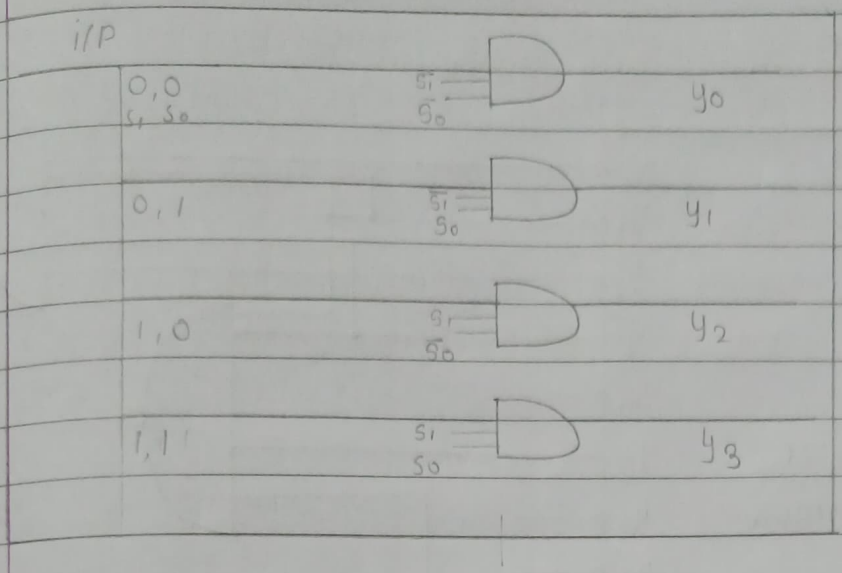
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

M T W T F S S

Page No.:

Date: 16/10/23

YOUVA



Que Implement following boolean expressions using Demux.

① $y_1 = f_1(A,B,C,D) = \sum_m (8, 9, 10, 11)$

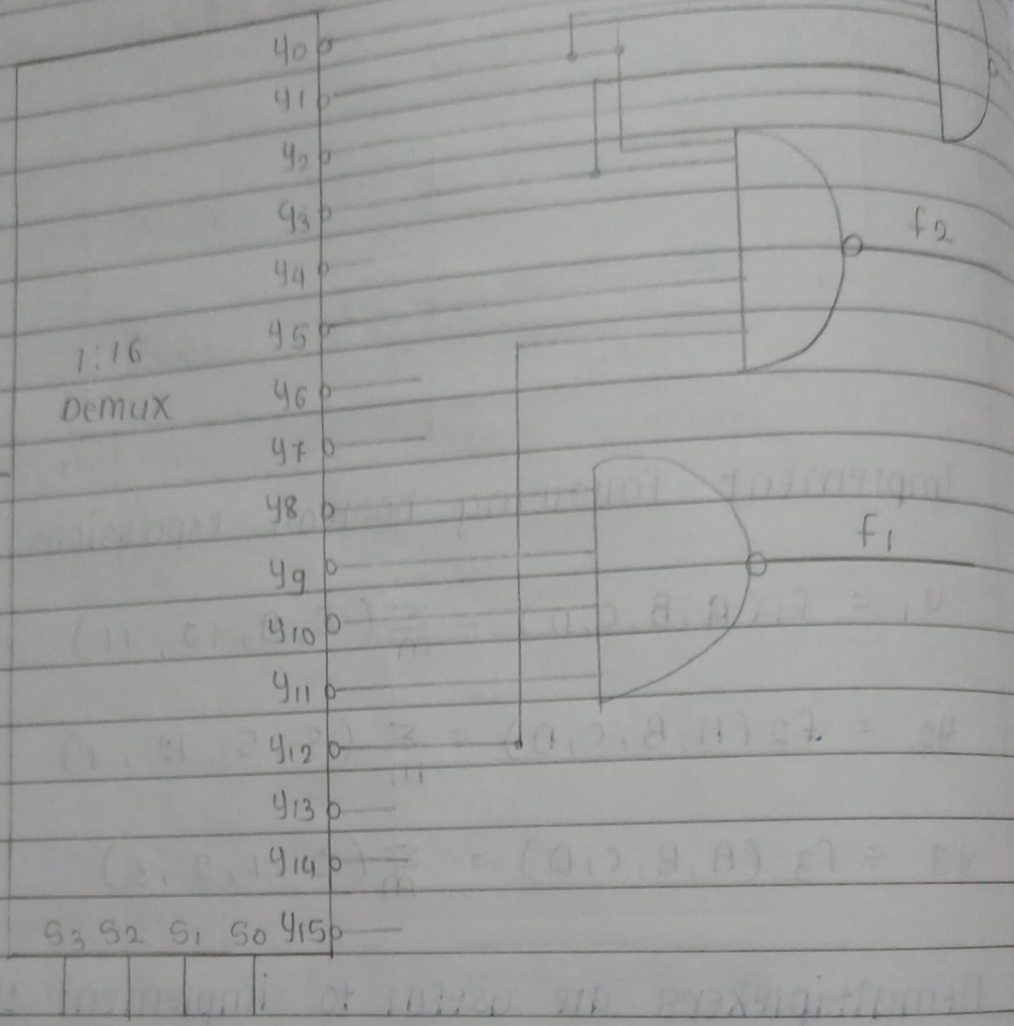
② $y_2 = f_2(A,B,C,D) = \sum_m (3, 5, 12, 1)$

③ $y_3 = f_3(A,B,C,D) = \sum_m (0, 1, 2, 3)$

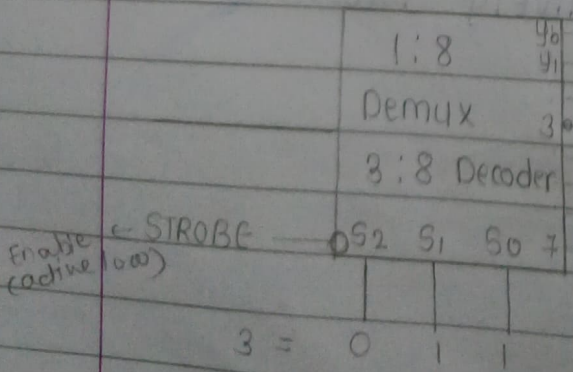
Note: Demultiplexers are useful to implement the multiobject-ive functions in which the boolean variables are common but the functions are different.

Olp of De multiplexers are actively low.

NXT PAGE



Demultiplexer as Decoder :



∴ 3:8 Decoder

Parity Generator Checker :

1 bit Parity : odd parity
even parity
} no. of 1's

que Design 3 bit parity (even) generator :

D_2	Even
D_1	Parity
D_0	Generator

P_e ⟹ | P_2 | D_1 | D_0 | P_e |
O/P

Truth Table

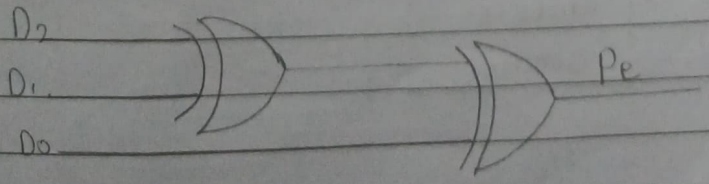
	D_2	D_1	D_0	P_e
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	01	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

no of 1
odd → 1
even → 0

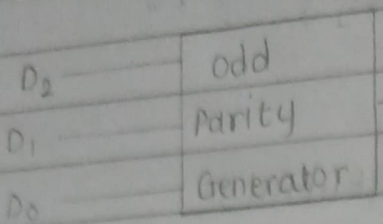
D_2 \ D_1D_0	00	01	11	10
0	0 0	① 1	0 3	① 2
1	① 4	0 5	① 7	0 6

$P_e = Y = \bar{D}_0 D_1 \bar{D}_2 + D_0 \bar{D}_1 \bar{D}_2 + \bar{D}_0 \bar{D}_1 D_2 + D_1 D_0 D_2$

$P_e = y = D_2 \oplus D_1 \oplus D_0$



Que. Design 3 bit odd parity generator:



$P_0 \Rightarrow$ | D_0 | D_1 | D_2 | P_0 |

Truth Table.

	D_2	D_1	D_0	P_0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

even → P → 1
odd → 0

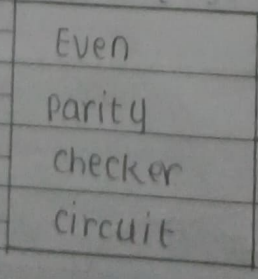
D_2 \ D_1D_0	00	01	11	10
0	(1) 0	0 1	(1) 3	0 2
1	0 4	(1) 5	1 7	(1) 6

$\therefore P_0 = \bar{D}_0\bar{D}_1\bar{D}_2 + D_0D_1\bar{D}_2 + \bar{D}_0D_1D_2 + D_0$

$\therefore$ $P_0 = \overline{D_2 \oplus D_1 \oplus D_0}$

Que Design 4 bit parity checker (Even).

Parity → even → 0
odd → 1



Y $\Rightarrow$ | D_3 | D_2 | D_1 | D_0 |

0101 → 1010
Transmit

M T W T F S S
Page No.:
Date: 18/10/23
YOUVA

Truthtable

	D_3	D_2	D_1	D_0	Y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

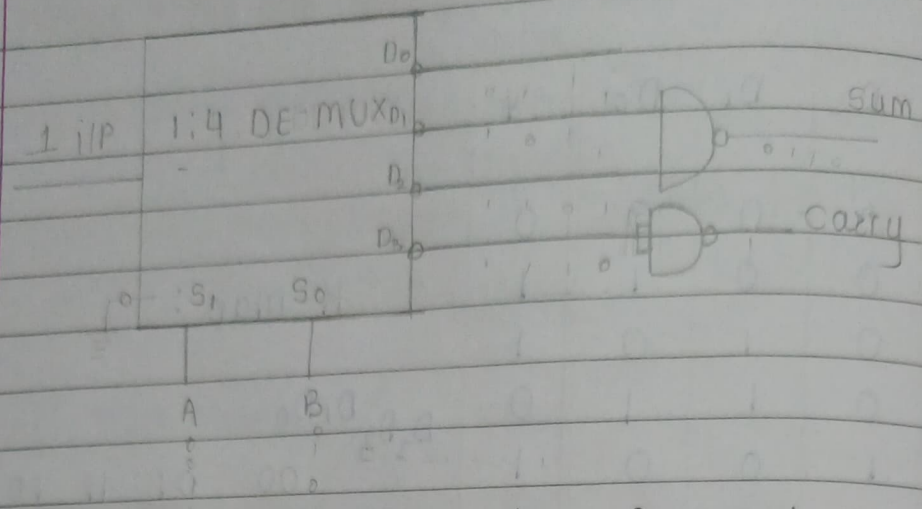
K-Map :

D_2D_3 \ D_1D_0	00	01	11	10
00	0 0	(1) 1	0 3	(1) 2
01	(1) 4	0 5	(1) 7	0 6
11	0 12	(1) 13	0 15	(1) 14
10	(1) 8	0 9	(1) 11	0 10

Implement half adder using De-MUX

Truth Table

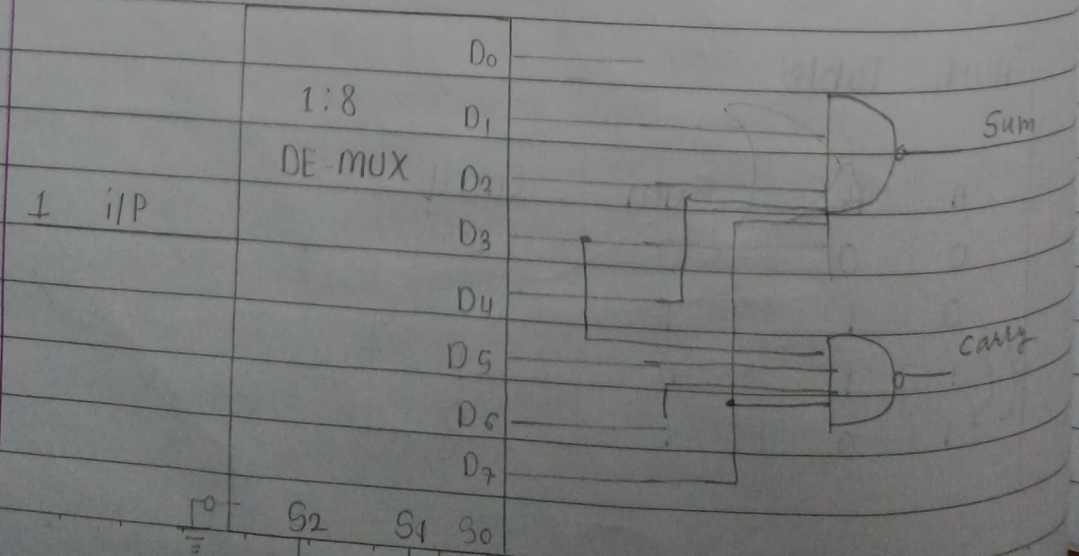
	A	B	Sum	Carry
0	0	0	0	0
1	0	1	1	0
3	1	1	0	1
2	1	0	1	0



Que Implement full adder using De-multiplexer.

Truth-Table :

	A	B	C	Sum	carry
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

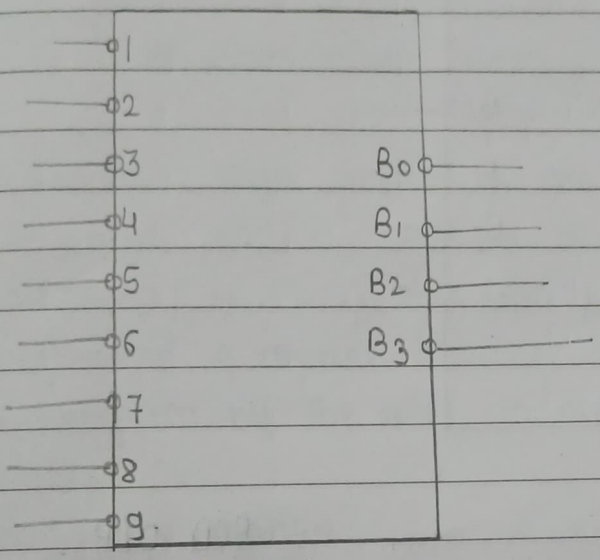


Active low → 1 → off → not activated
0 → on → Activated

Page No.:
Date: 30/10/23
M T W T F S S
YOUVA

Priority Encoders :

Decimal → BCD
Octal → BCD .



Decimal I/P.

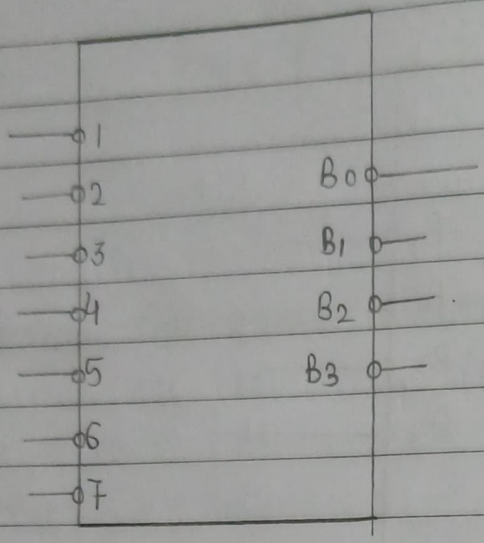
BCD O/P

0 → on
1 → off

1	2	3	4	5	6	7	8	9	B_3 (8)	B_2 (4)	B_1 (2)	B_0 (1)
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
X	0	1	1	1	1	1	1	1	1	1	0	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0

Octal → BCD.

8 symbols are there.



Octal i/p							BCD o/p			
1	2	3	4	5	6	7	B_3	B_2	B_1	B_0
1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0
X	0	1	1	1	1	1	1	1	0	1
X	X	0	1	1	1	1	1	1	0	0
X	X	X	0	1	1	1	1	0	1	1
X	X	X	X	0	1	1	1	0	1	0
X	X	X	X	X	0	1	1	0	0	1
X	X	X	X	X	X	0	1	0	0	0

PLD : Programmable logic Devices

1] ROM : Read only memory.

2] PLA : Programmable logic array

3] PAL : Programmable array logic

* Programmable logic device is an IC. That is user configurable and is capable of implementing logic functions. It is VLSI chip (Very large scale integration) that contains a regular structure.
It is program by the user to perform the function required.
Several advantages :

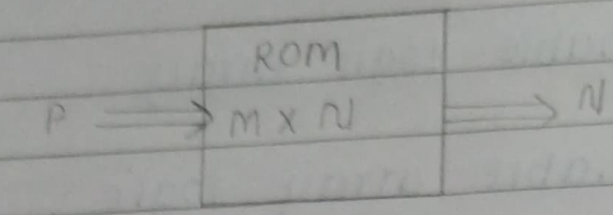
① Short design cycle.
② No development cost.
③ Reduction in board space requirement.
④ Reduction in power requirement.
⑤ Design security
⑥ Compact circuitary
⑦ High switching speed.
⑧ Higher density

1] ROM as PLD :

A Read Only Memory is basically is a combinational circuit and can be used to implement a ROM of size m x n. has m number of locations and n number of bits can be stored at each location.

M=8
P=3
$2^3 = 8$

The number of address input is P where $2^P = M$
And number of data output lines is N.



A P variable in output logic functions can be imp using a Rom of size $2^P \times N$. Since all the possible minterms are effectively generated.

Advantages of ROM:

① No Simplification / minimization required.

② Designs can be changed and modify rapidly.

③ It is faster than SSI / MSI circuits.

④ Cost is reduced.

2] PLA (Programmable Logic Array)

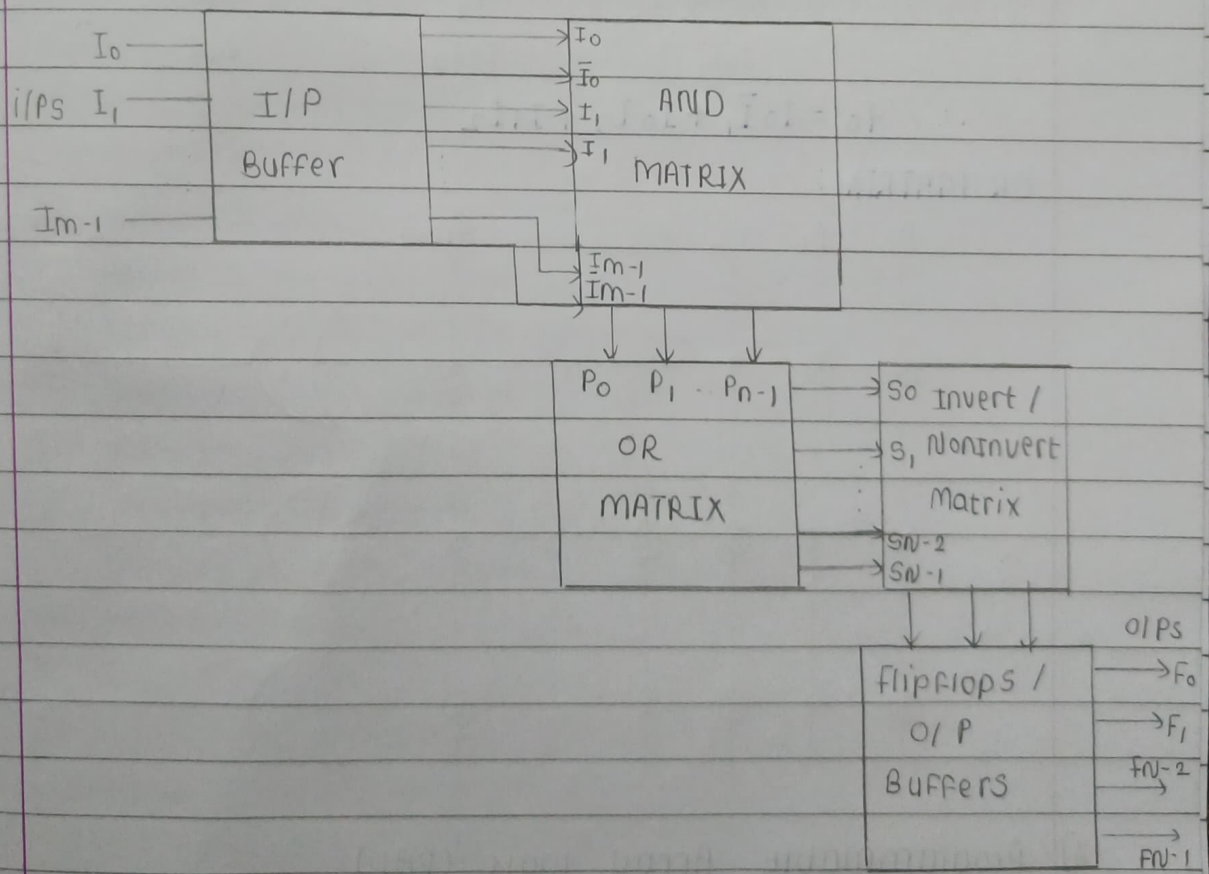
A PLA consist of 2 level and-or circuits on a single chip.
Number and and or gates and their inputs are fixed for given PLA.
The And gates provide product terms.
The OR gates logically sum the product terms
Thereby generating an SOP expressions.
It has a M inputs, n product terms and N o

which $n < 2^m$ and can be used to implement a logic functions of m variables and n outputs.

Since all of the possible 2^m minterms are not available.
Therefore, logic minimization is required to accomodate a give logic function.

2] PLA :

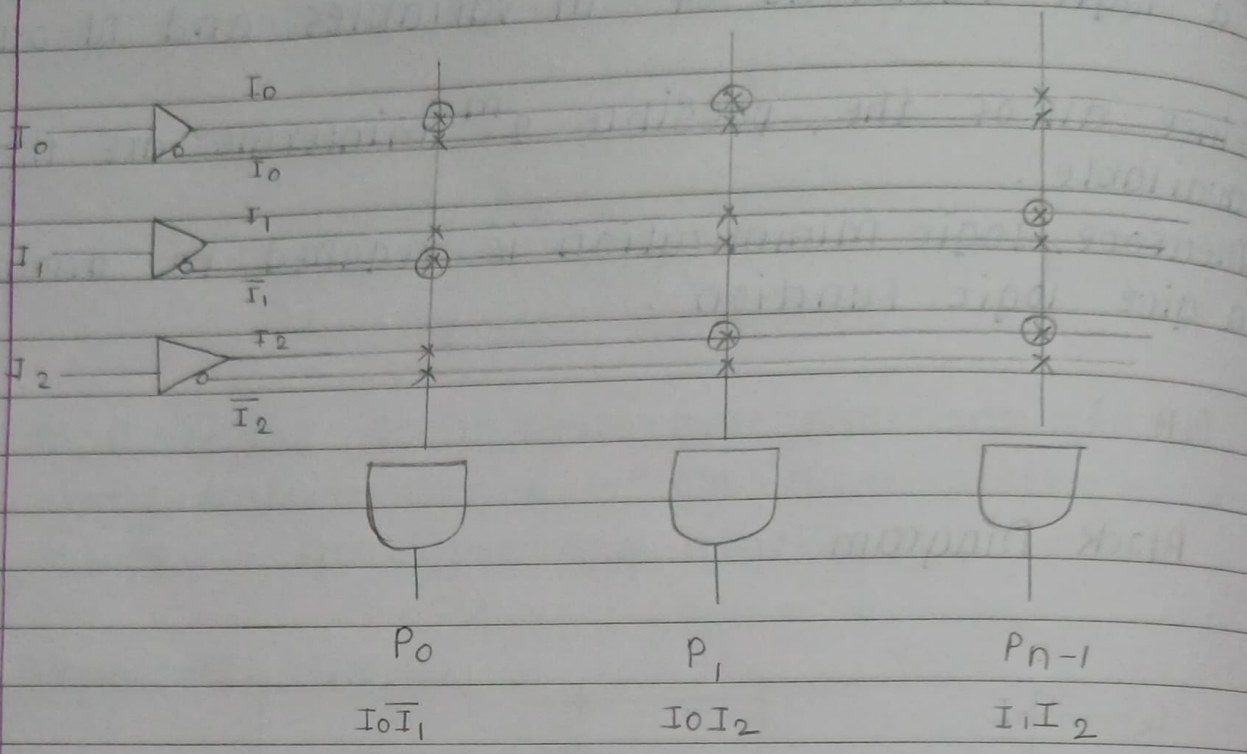
Block Diagram :



PLA consist of 2 level and-or circuit on a single chip. Input Buffer produces inverted and non inverted o/p. for generating req. product terms the unwanted links are open with method of programming.

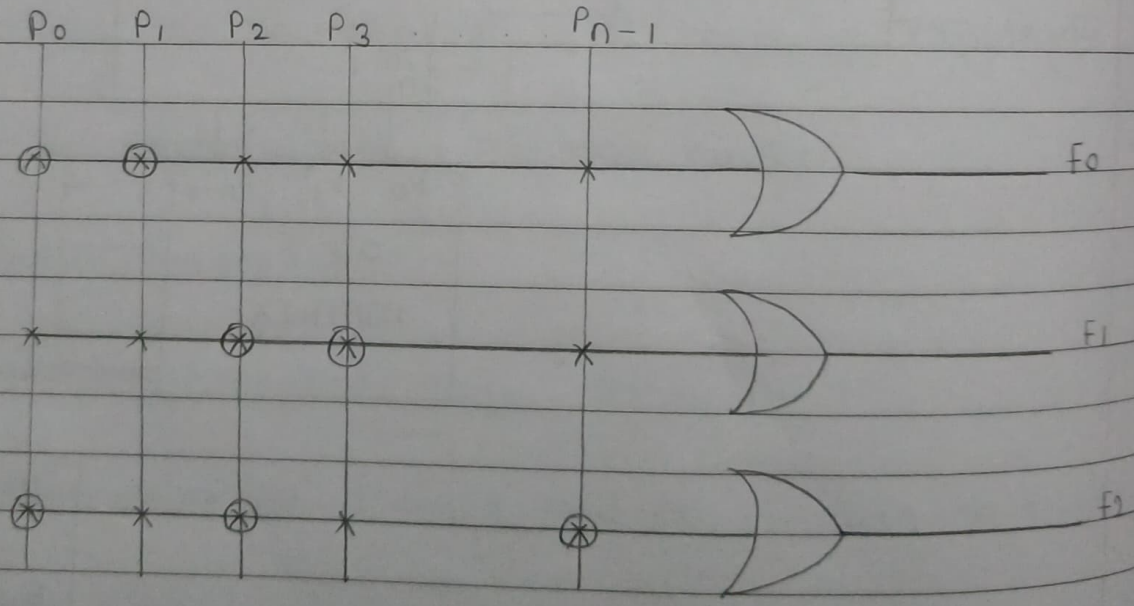
Same for PAL →

AND MATRIX :



$\therefore \quad f_0 = I_0\bar{I}_1 + I_0I_2 + I_1I_2$

OR MATRIX :



3] Programmable Array logic (PAL)

It is programmable array of logic gates on a single chip in AND OR configuration.
In contrast to PAL, it has programmable and

and fixed or array, in which each or gate gets inputs from some of the and gates.
ie. all the and gate outputs are not connected to any or gate.

Input and output circuit of PAL are similar to those of PLA. The number of fusible links in PAL is the product of 2M and n. where M is number of ilp variable and n is no. of product terms.

- AND MATRIX is same as PLA.
- OR MATRIX: