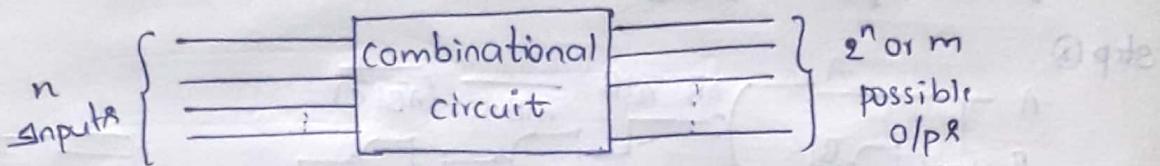


UNIT-III  
COMBINATIONAL CIRCUITS

Combinational circuit (or) combinational logics:-

combinational circuit are the circuits having 'n' inputs and 'm' possible outputs with zero memory and which works only with the present state inputs.



Design procedure for combinational circuits:

- (1) Statement of the problem.
- (2) Identify the inputs & outputs.
- (3) Assign letters (or) symbols to S/p's and outputs.
- (4) construct the truth table and mention the relation between Sinputs & outputs.
- (5) Write the boolean expression for the o/p variables with the terms of S/p variables.
- (6) Simplify using k-map & Tabular method
- (7) Implement the simplified boolean expression using logic gates.
- (8) Draw the combinational circuits.

Ex(1):- Design a 2bit add, (or) Binary adder on Half adder?

(A) Step ① :- Design a half-adder

	S/p		o/p	
	A	B	S	C
m <sub>0</sub>	0	0	0	0
m <sub>1</sub>	0	1	1	0
m <sub>2</sub>	1	0	1	0
m <sub>3</sub>	1	1	0	1

Step ⑤:- Boolean expression for o/p

$$S = \sum m(1,2), C = \sum m(3)$$

Step 6

	B	0	1
A	0	0	1
	1	1	0

	B	0	1
A	0	0	1
	1	0	0

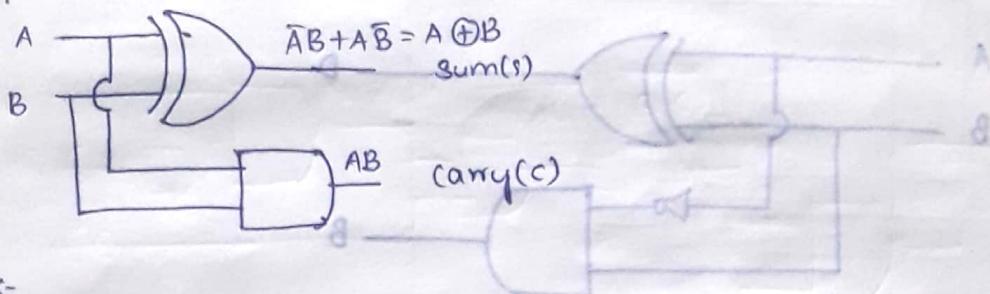
$$S = \sum m(1, 2) = \bar{A}B + A\bar{B}$$

$$= A \oplus B$$

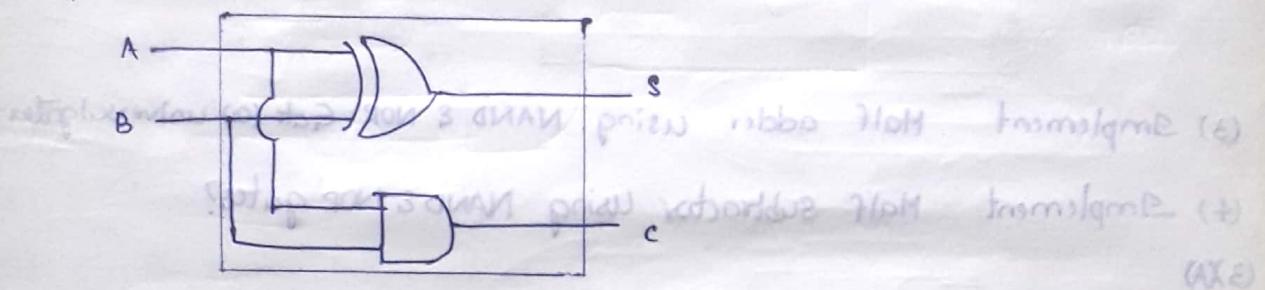
(Q12)

$$C = \sum m(3) = AB$$

Step 7:



Step 8:-



② Design a half subtractor (or) 2bit subtractor (or) Binary subtractor.

(A) Step 1:- Design a half subtractor.

	s/p		o/p	
Step 2,3,4:-	A	B	D	B
m <sub>0</sub>	0	0	0	0
m <sub>1</sub>	0	1	1	1
m <sub>2</sub>	1	0	1	0
m <sub>3</sub>	1	1	0	0

Step 5 Boolean expression for o/p

$$D = \sum m(1, 2), B = \sum m(1)$$

Step 6

	B	0	1
A	0	0	1
	1	1	0

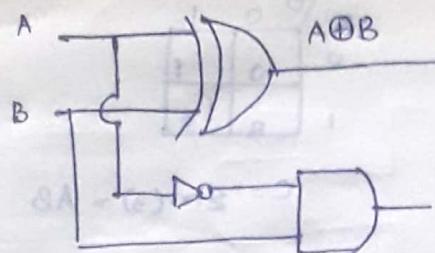
	B	0	1
A	0	0	1
	1	0	0

$$D = \sum m(1, 2) = \bar{A}B + A\bar{B}$$

$$= A \oplus B$$

$$B = \sum m(1) = \bar{A}B$$

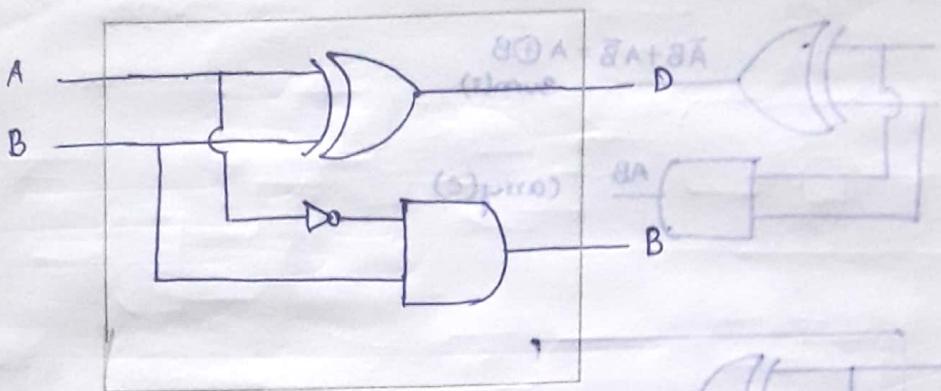
Step 7:-



$$\bar{A}B + B\bar{A} = (1,1) \text{ m3} = 2$$

$$B \oplus A =$$

Step 8



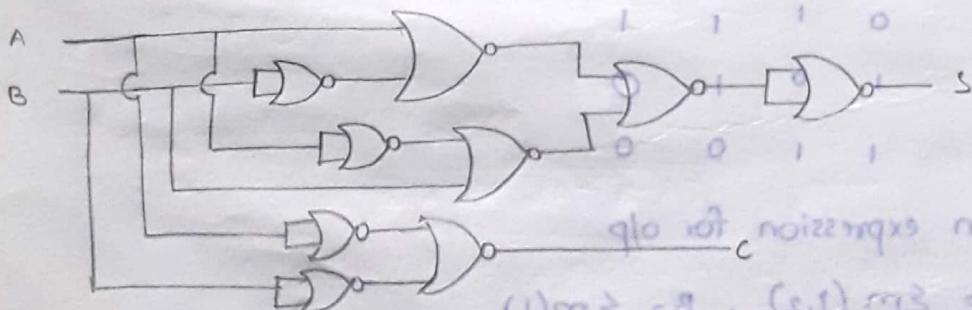
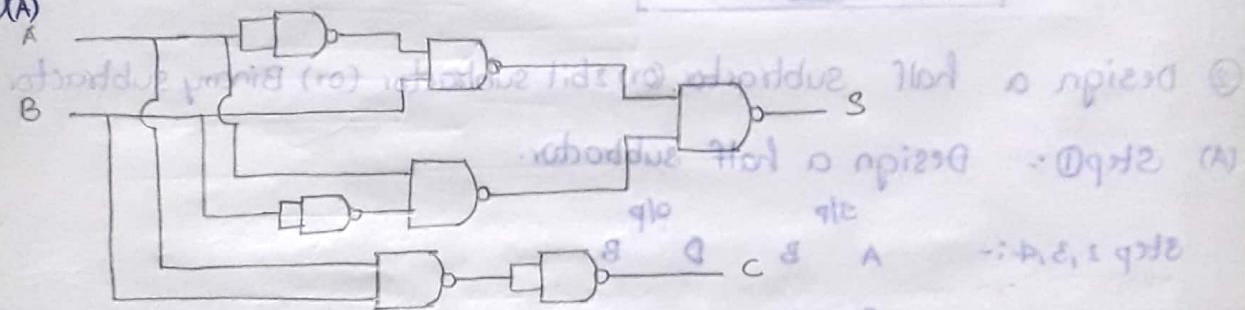
Step 8

Step 8

(3) Implement Half adder using NAND & NOR Gate (or) universal gates?

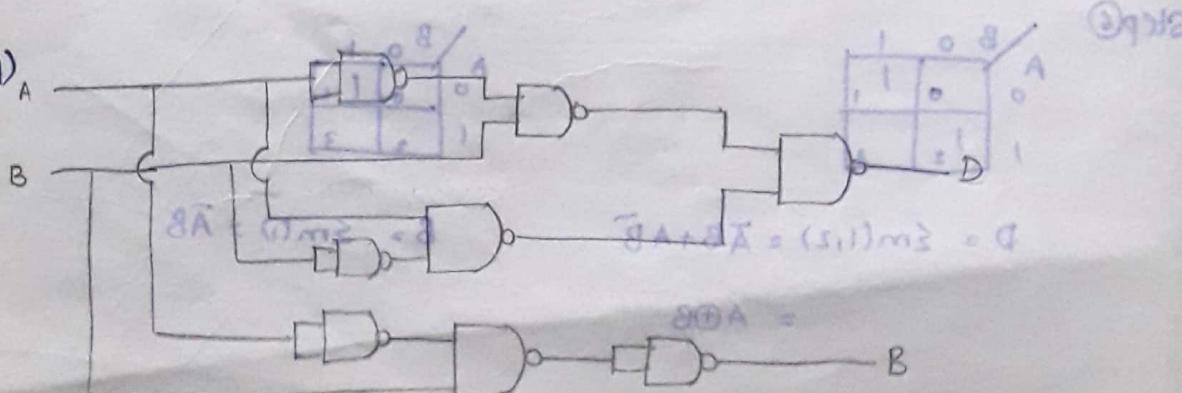
(4) Implement Half subtractor using NAND & NOR gates?

(3)(A)

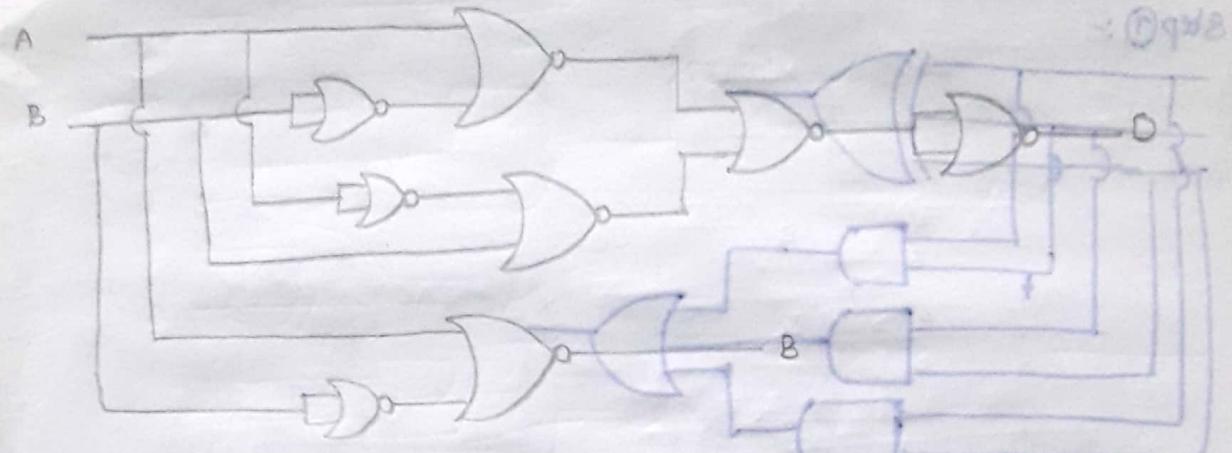


$$(1)m3 - 0, (2,1)m3 = 4$$

4(A)



Half subtractor using NOR gate



Q18

(6) Implement full adder?

A) step ① design full adder

	S/I/P			O/P	
Step 2,3,4 :-	A	B	Cin	S	Cout
m <sub>0</sub>	0	0	0	0	0
m <sub>1</sub>	0	0	1	1	0
m <sub>2</sub>	0	1	0	0	1
m <sub>3</sub>	1	0	0	1	0
m <sub>4</sub>	0	1	1	0	1
m <sub>5</sub>	1	0	1	1	0
m <sub>6</sub>	1	1	0	0	1
m <sub>7</sub>	1	1	1	1	1

Step ②:- m<sub>0</sub> = 0A + 0B + 0C<sub>in</sub> = 0  
 m<sub>1</sub> = 0A + 0B + 1C<sub>in</sub> = C<sub>in</sub>  
 m<sub>2</sub> = 0A + 1B + 0C<sub>in</sub> = B  
 m<sub>3</sub> = 0A + 1B + 1C<sub>in</sub> = B + C<sub>in</sub>  
 m<sub>4</sub> = 1A + 0B + 0C<sub>in</sub> = A  
 m<sub>5</sub> = 1A + 0B + 1C<sub>in</sub> = A + C<sub>in</sub>  
 m<sub>6</sub> = 1A + 1B + 0C<sub>in</sub> = A + B  
 m<sub>7</sub> = 1A + 1B + 1C<sub>in</sub> = A + B + C<sub>in</sub>

Step ③:- Boolean expression of o/p  $(\bar{B}A + \bar{B}\bar{A})_{n_1} + n_1B\bar{A} =$

$$S = \sum m(1, 2, 4, 7), \text{ Cout} = \sum m(3, 5, 6, 7)$$

	00	01	11	10
0	0	1	3	12
1	4	5	7	6

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

$$S = A\bar{B}\bar{C}_{in} + \bar{A}\bar{B}C_{in} + AB\bar{C}_{in} + \bar{A}BC_{in}$$

$$S = C_{in}(AB + \bar{A}\bar{B}) + \bar{C}_{in}(\bar{A}B + A\bar{B})$$

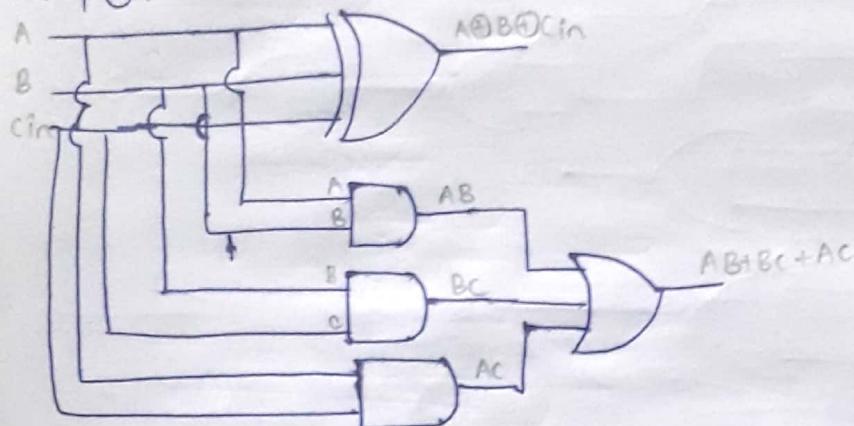
$$\text{Cout} = AC_{in} + BC_{in} + AB$$

$$= C_{in} (\overline{A \oplus B}) + \overline{C}_{in}(A \oplus B)$$

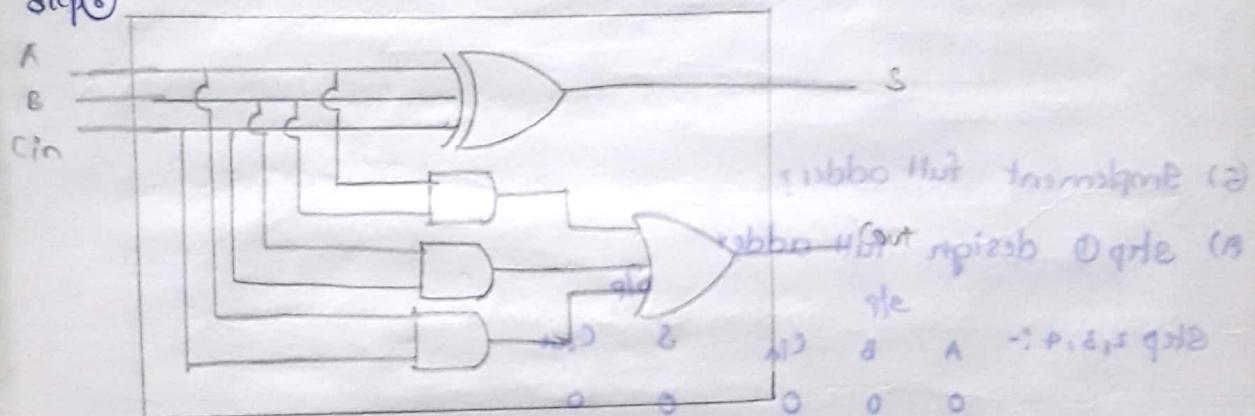
$$= A \oplus B \oplus C_{in}$$

step 7 on page 100

Step 7:-



Step 8



(7) Implementation of full adder using two half adders.

(a) To design full adder using two half adders and an OR gate  
we require

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = A C_{in} + B C_{in} + AB$$

$$= AC_{in}(B + \bar{B}) + BC_{in}(A + \bar{A}) + AB$$

$$= ABC_{in} + \bar{A}\bar{B}C_{in} + \bar{A}BC_{in} + \bar{A}\bar{B}C_{in} + AB$$

$$= ABC_{in} + C_{in}(\bar{A}B + A\bar{B}) + AB$$

$$= AB(1 + C_{in}) + C_{in}(\bar{A}B + A\bar{B})$$

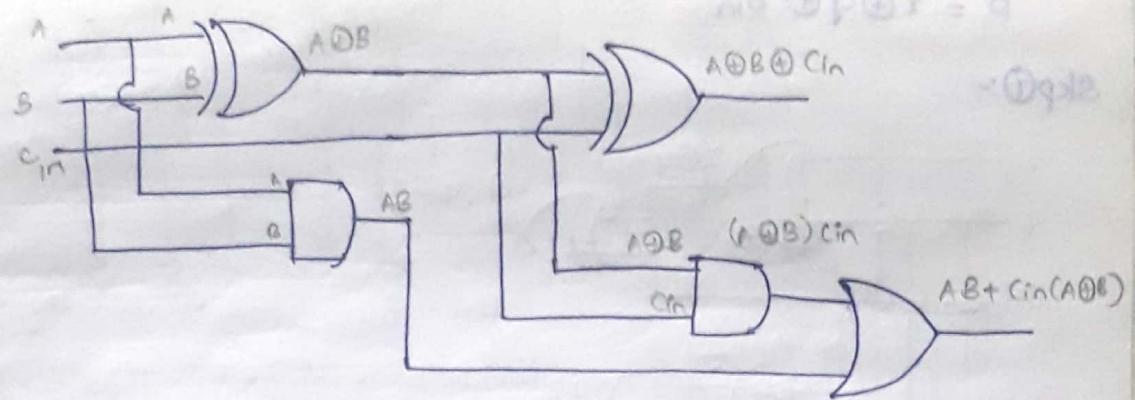
$$\Rightarrow AB + C_{in}(A \oplus B)$$

$$= 1 \ 0 \ 1 \ 0 \ 0$$

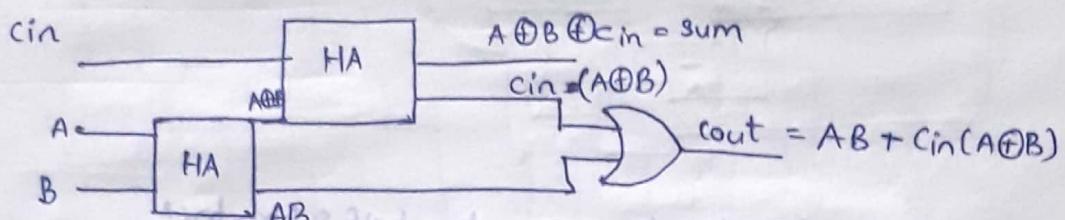
$$= 1 \ 1 \ 1 \ 1 \ 1$$

$$AB + \bar{A}B + A\bar{B} + \bar{A}\bar{B} = 1101$$

$$(\bar{A}B + A\bar{B}) + (\bar{A}\bar{B} + A\bar{B}) = 1101$$



Combinational circuit



(8) Design full subtractor?

(A) Step 0 Design full subtractor

Step 2,3,4 :-			A/p	O/p
x	y	Bin	D <sub>out</sub>	B <sub>out</sub> = p̄x + p̄y = f <sub>out</sub>
m <sub>0</sub> 0	0	(x̄+y) 0	0	(p̄x+p̄y) 0 = f <sub>out</sub>
m <sub>1</sub> 0	0	1	1	1 =
m <sub>2</sub> 0	1	0	0	0 =
m <sub>3</sub> 1	0	0	1	(p̄x+y) 1 =
m <sub>4</sub> 1	0	1	0	(p̄x+y) 0 =
m <sub>5</sub> 1	1	0	0	0 =
m <sub>6</sub> 1	1	1	1	1 =
m <sub>7</sub> 1	1	1	1	1 =

Step 5: Boolean expression of o/p

$$D = \sum m(1, 2, 4, 7) \quad B_{out} = \sum m(1, 2, 3, 7)$$

Step 6 :- k-map

x\y\Bin	00	01	11	10
0	0	1	3	1
1	1	5	1	6

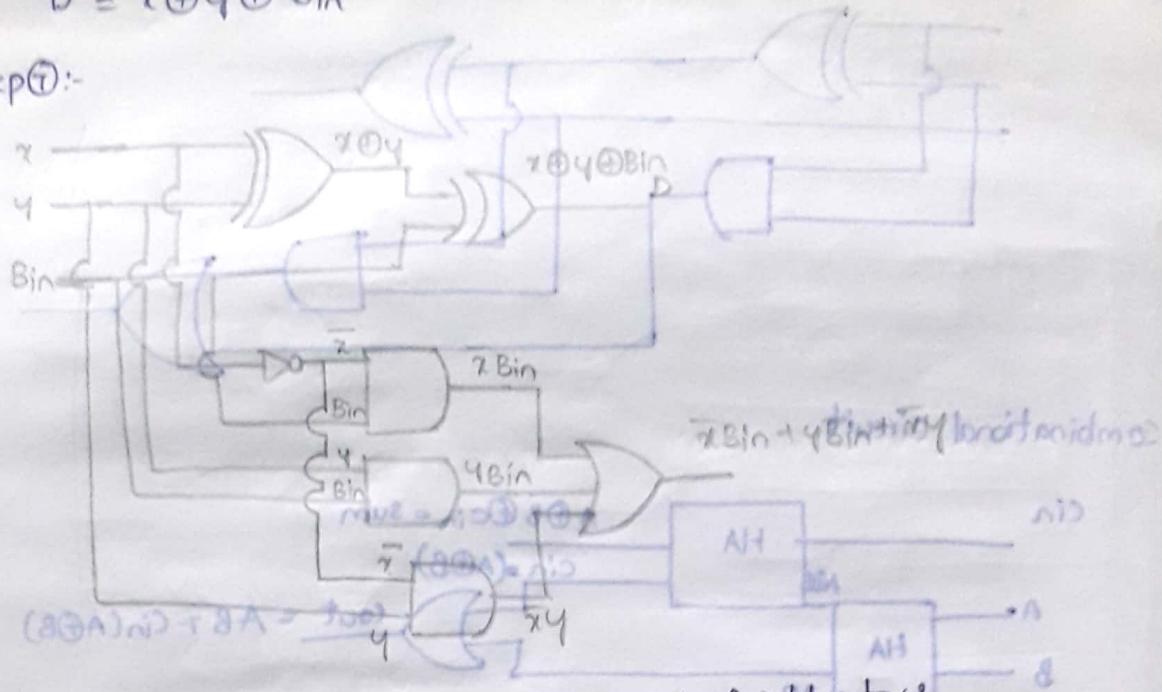
x\y\Bin	00	01	11	10
0	0	1	1	1
1	4	6	1	5

$$D = \bar{x}\bar{y}B_{in} + \bar{x}\bar{y}B_{in} + x\bar{y}B_{in} + \bar{x}y\bar{B}_{in} \quad B_{out} = \bar{x}B_{in} + yB_{in} + \bar{x}y$$

$$D = B_{in}(\bar{x}\oplus y) + \bar{B}_{in}(x\oplus y)$$

$$D = x \oplus y \oplus \text{Bin}$$

Step 7:-

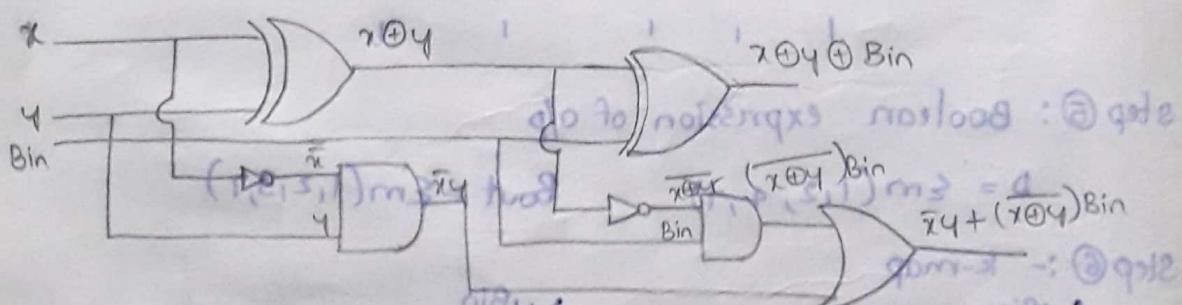


(9) Design full subtractor by using two half subtractors?

(A) This implementation requires two half subtractors and one OR gate.

$$D = x \oplus y \oplus \text{Bin}$$

$$\begin{aligned} \text{Bout} &= \bar{x}y + \bar{x}\text{Bin} + y\text{Bin} \\ &= \bar{x}y + \bar{x}\text{Bin}(y+\bar{y}) + y\text{Bin}(x+\bar{x}) \\ &= \bar{x}y + \bar{x}y\text{Bin} + \bar{x}\bar{y}\text{Bin} + x\bar{y}\text{Bin} + \bar{x}y\text{Bin} \\ &= \text{Bin}(\bar{x}y + x\bar{y}) + \bar{x}y(1 + \text{Bin}) \\ &= \bar{x}y + \text{Bin}(x \oplus y) \end{aligned}$$



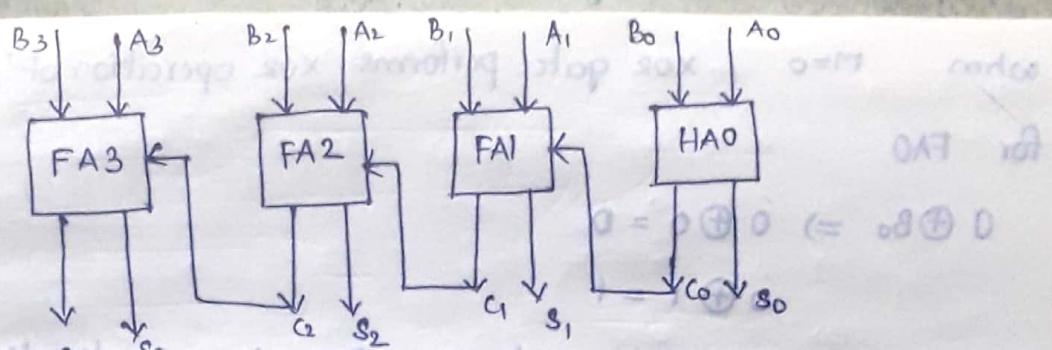
(10) Design 4 bit adder (or) parallel adder by using half adders & full adders.

(A)

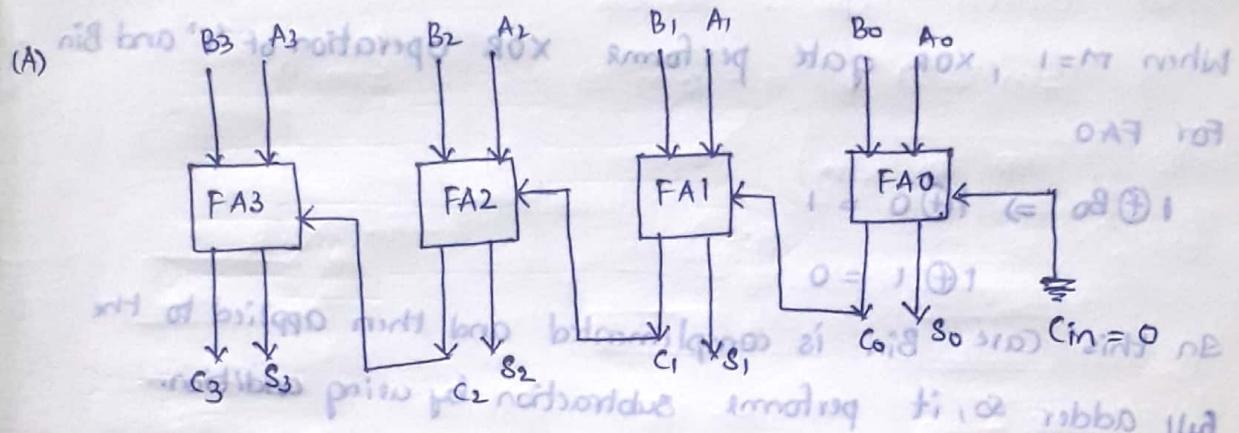
$$\bar{p}\bar{x} + \bar{n}\bar{q}\bar{p} + \bar{n}\bar{q}\bar{x} = \text{full}$$

$$\bar{n}\bar{q}\bar{p} + \bar{n}\bar{q}x + \bar{n}q\bar{p} + \bar{n}q\bar{x} = 0$$

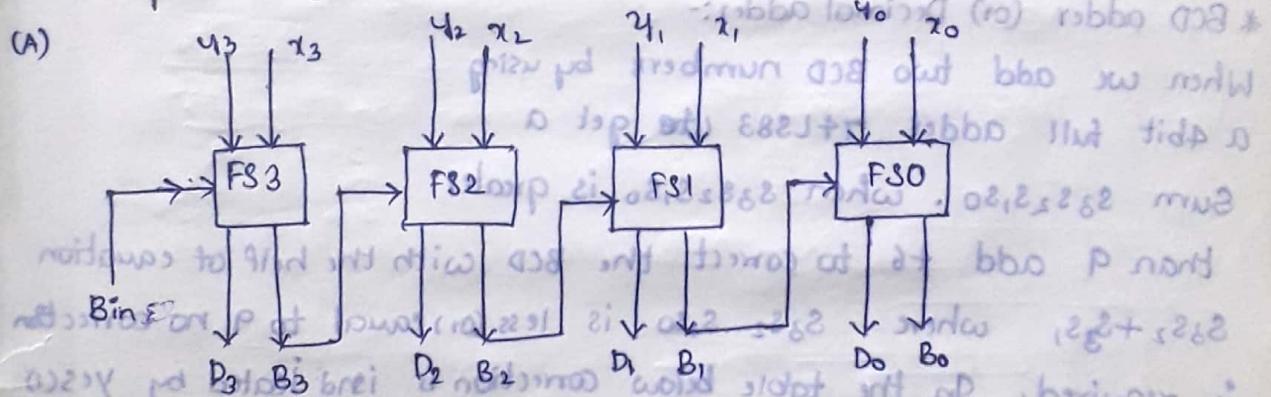
$$(p \oplus q)\bar{n}\bar{q} + (\bar{p} \oplus \bar{q})n\bar{q} = 0$$



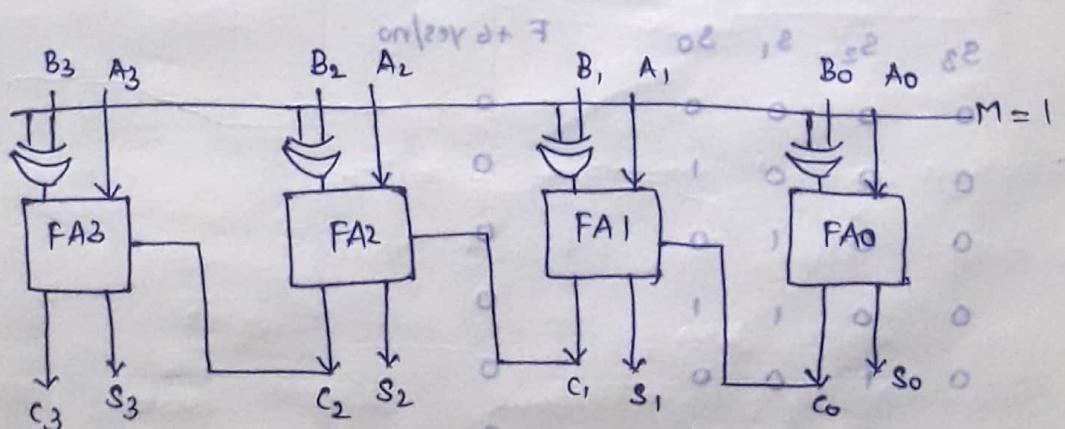
(11) design 4bit adder or parallel adder using only full adders?



(12) Design a 4-bit subtracter



(13) Design 4-bit adder (0) act as subtractor ei noideen on bno



when  $M=0$ , XOR gate performs XOR operation of '0' and Bin.

for FA0

$$0 \oplus B_0 \Rightarrow 0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

In this case whatever is the Bin that is directly applied to the full adder so it performs addition.

When  $m=1$ , XOR gate performs XOR operation of '1' and Bin

For FA0

$$1 \oplus B_0 \Rightarrow 1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

In this case Bin is complemented and then applied to the full adder so, it performs subtraction by using addition.

\* BCD adder (01) Decimal adder:-

When we add two BCD numbers by using a 4bit full adder 74LS83 we get a

Sum  $s_3 s_2 s_1 s_0$ . when  $s_3 s_2 s_1 s_0$  is greater

than 9 add +6 to correct the BCD with the help of equation  $s_3 s_2 + s_3 s_1$ , where  $s_3 s_2 s_1 s_0$  is less (0) or equal to 9 no correction is required. In the table below correction is indicated by Yes(1) and no correction is indicated by No(0).

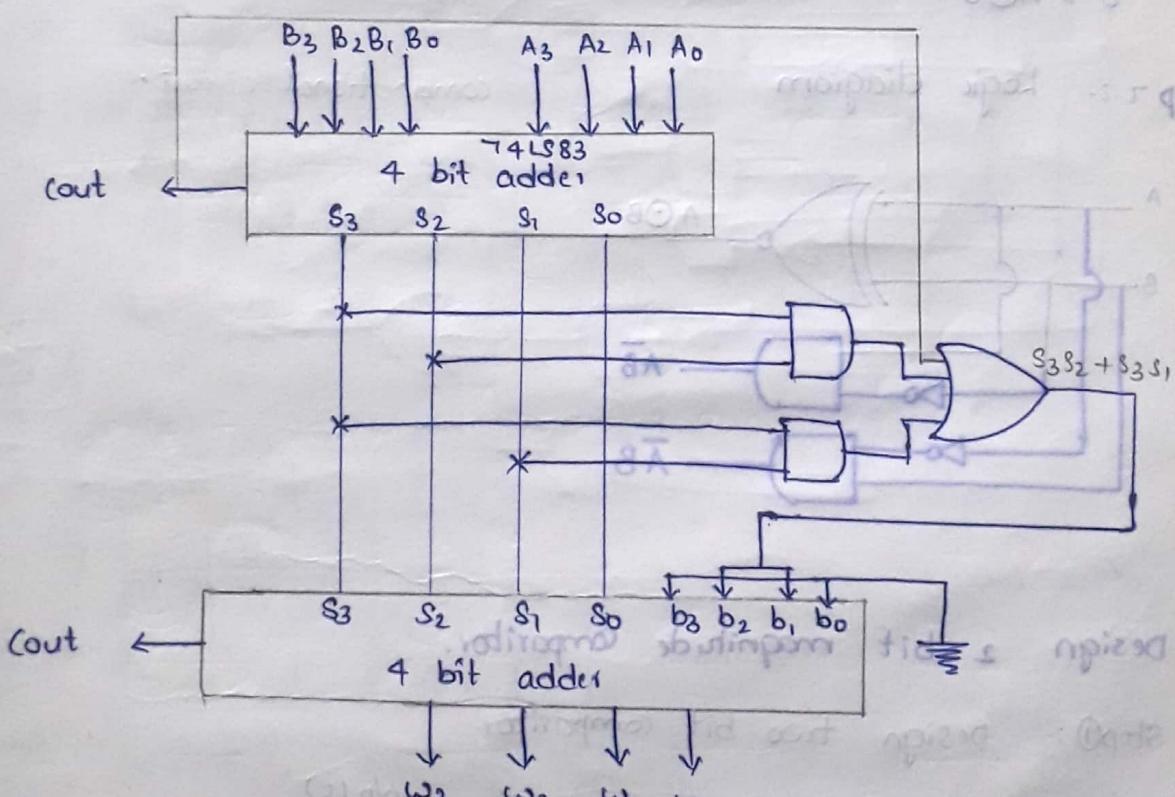
$s_3$	$s_2$	$s_1$	$s_0$	$F +6$ yes/no
$I = M_0$	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
0	0	0	0	0
1	0	0	1	0

- (10) 1 0 1 0
- (11) 1 0 1 1
- (12) 1 1 0 0
- (13) 1 1 0 1
- (14) 1 1 1 0
- (15) 1 1 1 1

kmap :-

$S_3 S_2$	00	01	11	10
$S_1 S_0$	0	1	3	2
00	4	5	7	6
01	12	13	15	14
11	8	9	11	10

$$F_{yes/no} = S_3 S_2 + S_3 S_1$$



\* Design a 1bit magnitude comparator (or) unary comparator.

(A) Step 1:- Design unary comparator.

		S/I/P (S)		O/P (S)				
		A	B	$E$ ( $A=B$ )	$G$ ( $A>B$ )	$L$ ( $A < B$ )		
(0)	0	0		1	0	0	0	0
(1)	0	1		0	0	1	1	0
(2)	1	0		0	1	0	0	1
(3)	1	1		1	0	0	1	1

Step 5:- Boolean expression for o/p :- qand

$$E = \Sigma m(0,3), \quad G = \Sigma m(2), \quad L = \Sigma m(1)$$

Step 6:- Kmap for o/p.

A	B	0	1
0	1	0	1
1	2	1	0

$$E = \bar{A}\bar{B} + AB$$

$$E = A \odot B$$

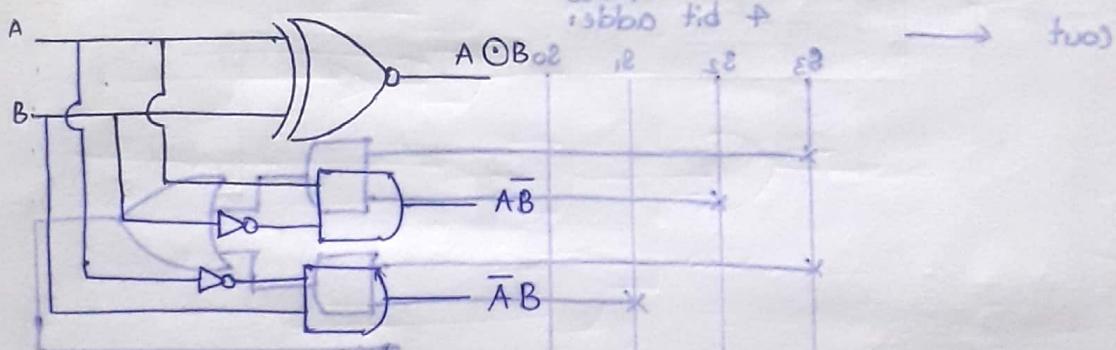
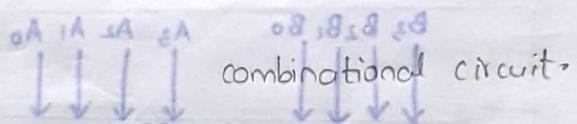
A	B	0	1
0	1	0	1
1	2	1	0

$$G = \bar{A}\bar{B}$$

A	B	0	1
0	1	1	0
1	2	0	1

$$L = \bar{A}B$$

Step 7:- Logic diagram



\* Design 2 bit magnitude comparator.

(A) Step 0:- Design two bit comparator.

Step 2,3,4:- S/I/P (S)      o/w      o/w      o/w      o/p (S)

Comparing A with B		Comparing B with A		$E$ ( $A=B$ )	$G$ ( $A>B$ )	$L$ ( $A < B$ )
$A_0$	$A_1$	$B_0$	$B_1$	1	0	0
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1

0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

Step ⑤ Boolean expression for o/p.

$$F = \sum m(0, 5, 10, 15), G = \sum m(4, 8, 9, 12, 13, 14)$$

$$L = \sum m(2, 3, 6, 7, 11)$$

Step ⑥ Kmap for o/p.

F:-

$A_0 A_1$	$B_0 B_1$	00	01	11	10
00	1	0	1	2	2
01	4	1	5	7	6
11	12	13	1	15	14
10	8	9	11	1	10

$$F = \bar{A}_0 \bar{A}_1 \bar{B}_0 \bar{B}_1 + \bar{A}_0 A_1 \bar{B}_0 \bar{B}_1 + A_0 A_1 B_0 B_1 + A_0 \bar{A}_1 B_0 \bar{B}_1$$

$$F = \bar{A}_0 \bar{B}_0 (\bar{A}_1 \bar{B}_1 + A_1 B_1) + A_0 B_0 (A_1 B_1 + \bar{A}_1 \bar{B}_1)$$

$$F = \bar{A}_0 \bar{B}_0 (A_1 \oplus B_1) + A_0 B_0 (A_1 \oplus B_1)$$

$$F = [A_1 \oplus B_1] [A_0 \oplus B_0]$$

G:-

$A_0 A_1$	$B_0 B_1$	00	01	11	10
00	00	0	1	2	3
01	01	4	5	7	6
11	11	12	13	15	14
10	10	18	19	11	10

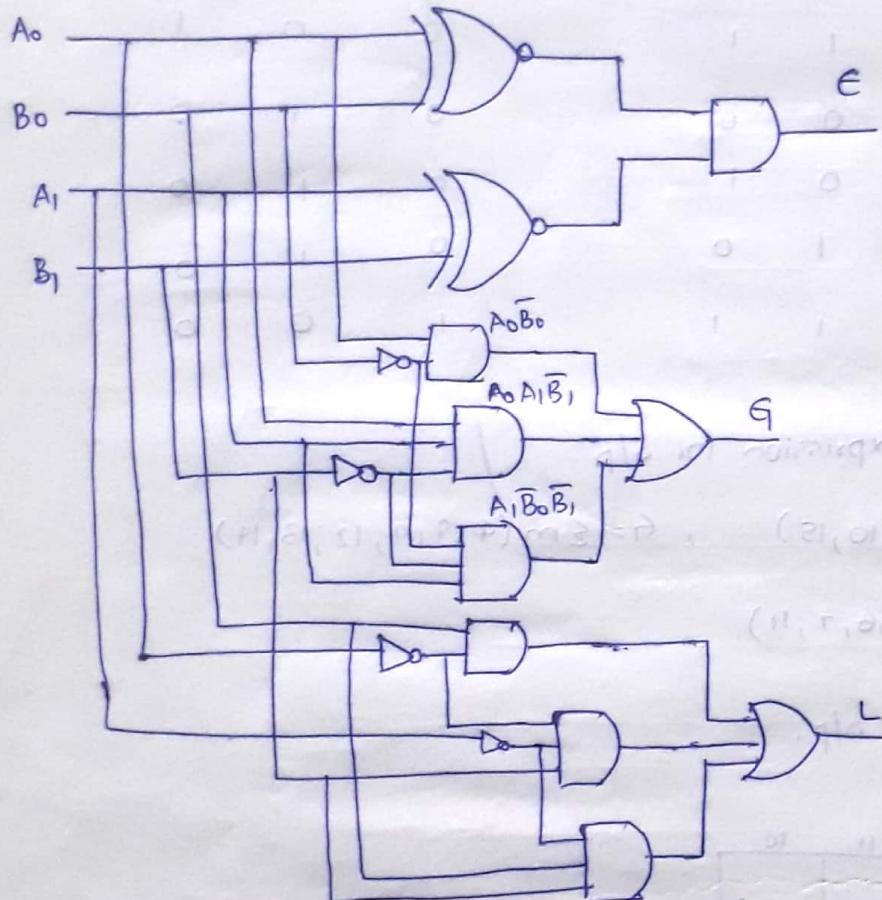
$$G = A_0 \bar{B}_0 + A_0 A_1 \bar{B}_1 + A_1 \bar{B}_0 \bar{B}_1$$

L:-

$A_0 A_1$	$B_0 B_1$	00	01	11	10
00	00	0	1	1	2
01	01	4	5	17	16
11	11	12	13	15	14
10	10	8	9	11	10

$$L = \bar{A}_0 B_0 + \bar{A}_0 \bar{A}_1 B_1 + \bar{A}_1 B_0 B_1$$

combinational circuit:-



00	01	11	10
0	1	1	2
4	5	17	16
12	13	15	14
18	9	11	10

$$(\bar{A}_0 \bar{B}_0 + A_0 \bar{B}_1 + A_1 \bar{B}_0 + A_0 A_1 \bar{B}_1) + (\bar{A}_0 \bar{B}_0 + A_0 \bar{B}_1 + A_1 \bar{B}_0 + A_1 \bar{B}_1) = 3$$

$$(\bar{A}_0 \bar{B}_0 + A_0 \bar{B}_1 + (A_1 \bar{B}_0 + \bar{A}_1 \bar{B}_1)) \bar{A}_0 \bar{B}_1 = 3$$

$$(A_0 \bar{B}_0 \bar{B}_1 + (A_0 \bar{B}_1 \bar{B}_1)) \bar{A}_0 \bar{B}_1 = 3$$

$$(A_0 \bar{B}_0 \bar{B}_1) (\bar{A}_0 \bar{B}_1) = 3$$

\* Design 4-bit magnitude comparator

A                            B  
A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub>      B<sub>3</sub> B<sub>2</sub> B<sub>1</sub> B<sub>0</sub>

1. If A<sub>3</sub> > B<sub>3</sub> then A > B & if A<sub>3</sub> < B<sub>3</sub> then A < B & if A<sub>3</sub> = B<sub>3</sub> then A ≠ B.
2. If A<sub>3</sub> = B<sub>3</sub> then & A<sub>2</sub> > B<sub>2</sub> then A > B, & if A<sub>2</sub> < B<sub>2</sub> then A < B  
& if A<sub>2</sub> = B<sub>2</sub> then A ≠ B.
3. If A<sub>3</sub> = B<sub>3</sub> & A<sub>2</sub> = B<sub>2</sub> & A<sub>1</sub> > B<sub>1</sub> then A > B & if A<sub>1</sub> < B<sub>1</sub> then A < B & if A<sub>1</sub> = B<sub>1</sub> then A ≠ B
4. If A<sub>3</sub> = B<sub>3</sub> & A<sub>2</sub> = B<sub>2</sub> & A<sub>1</sub> = B<sub>1</sub> & if A<sub>0</sub> > B<sub>0</sub> then A > B & if A<sub>0</sub> < B<sub>0</sub> then A < B & if A<sub>0</sub> = B<sub>0</sub> then A = B.

\* If MSB bits A<sub>3</sub> & B<sub>3</sub> are equal then A<sub>3</sub> ⊕ B<sub>3</sub> = x<sub>3</sub>

A<sub>2</sub> & B<sub>2</sub>

A<sub>2</sub> ⊕ B<sub>2</sub> = x<sub>2</sub>

A<sub>1</sub> & B<sub>1</sub>

A<sub>1</sub> ⊕ B<sub>1</sub> = x<sub>1</sub>

A<sub>0</sub> & B<sub>0</sub>

A<sub>0</sub> ⊕ B<sub>0</sub> = x<sub>0</sub>

$$E = (A_3 \oplus B_3)(A_2 \oplus B_2)(A_1 \oplus B_1)(A_0 \oplus B_0)$$

$$G = A_3 \bar{B}_3 + x_3 A_2 \bar{B}_2 + x_3 x_2 A_1 \bar{B}_1 + x_3 x_2 x_1 A_0 \bar{B}_0$$

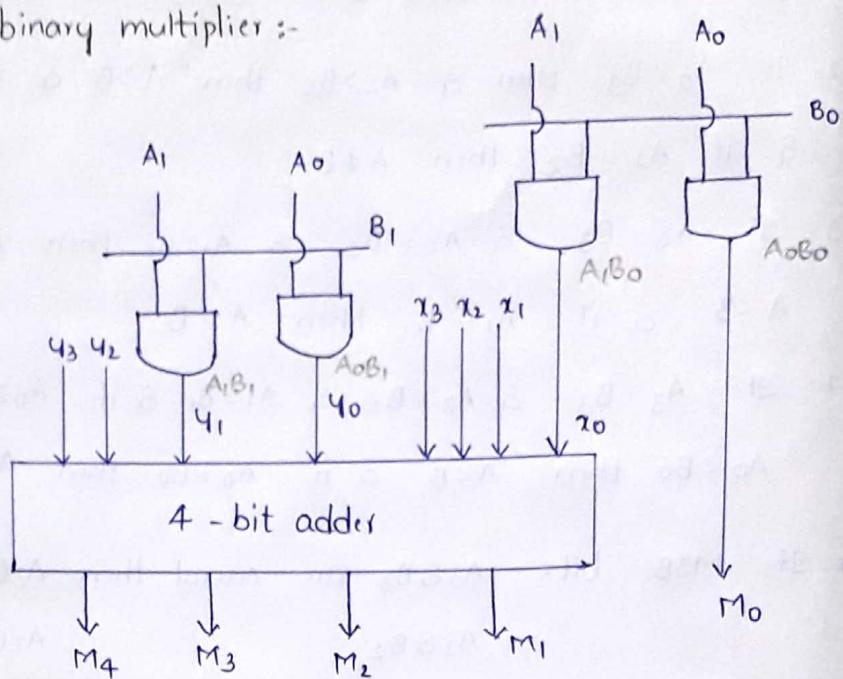
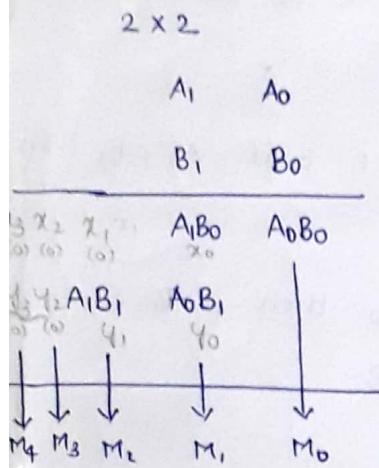
$$L = \bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 + x_3 x_2 x_1 \bar{A}_0 B_0$$

Logic Diagram

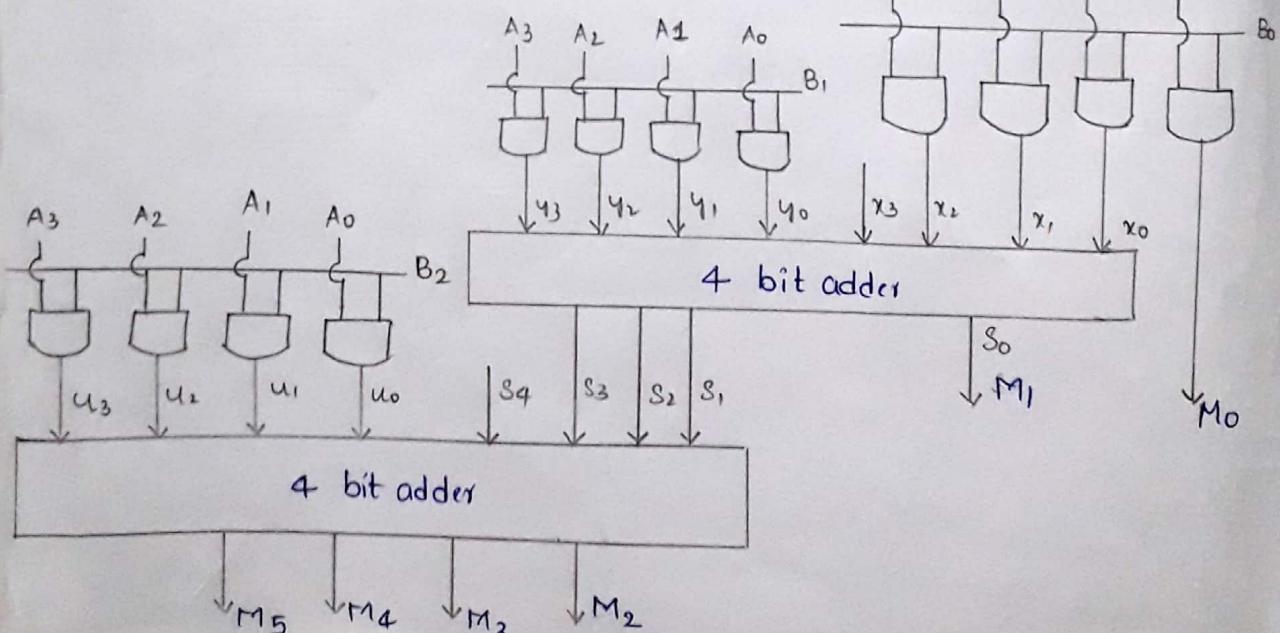
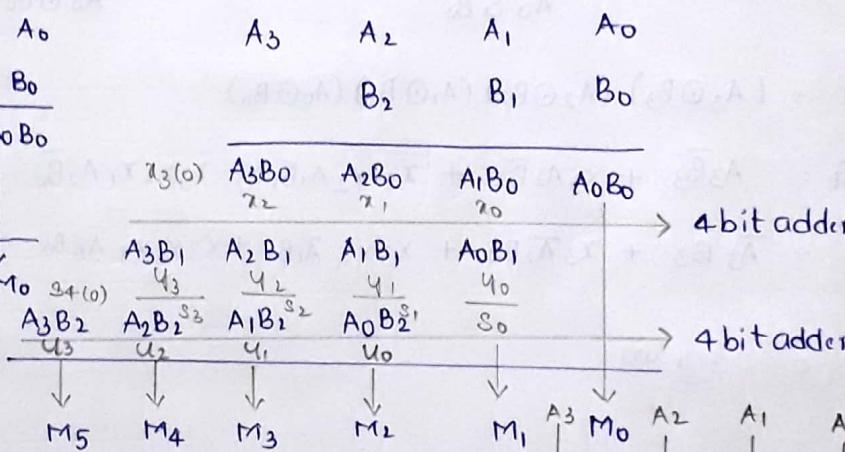
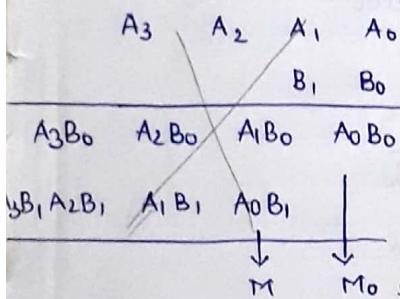
## Binary Multiplier:-

Binary Multiplier is an electronic circuit used in digital electronics such as computers to perform multiplication of 2 binary numbers. It is built by using Binary adder.

\* Design a  $2 \times 2$  binary multiplier:-



\* Design  $4 \times 3$  binary multiplier?

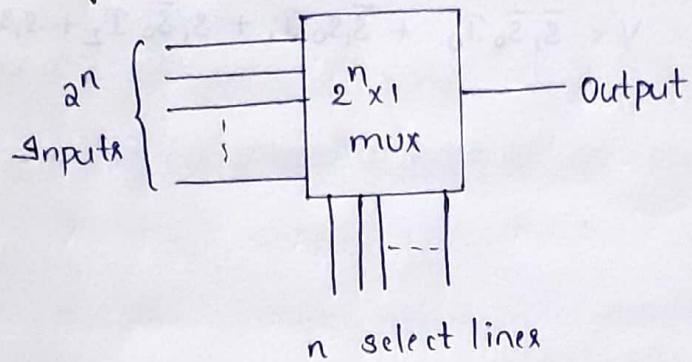


MULTIPLEXER: Multiplexer is a device which selects any one of the  $2^n$  inputs as a output.

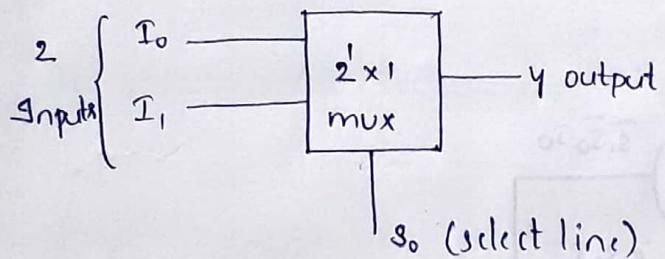
(01)  
Multiplexer is a switching device with  $2^n$  inputs and one output and 'n' select lines.

\* Multiplexer in short form is designated as MUX

Block Diagram:-



\* Design a 2:1 MUX?



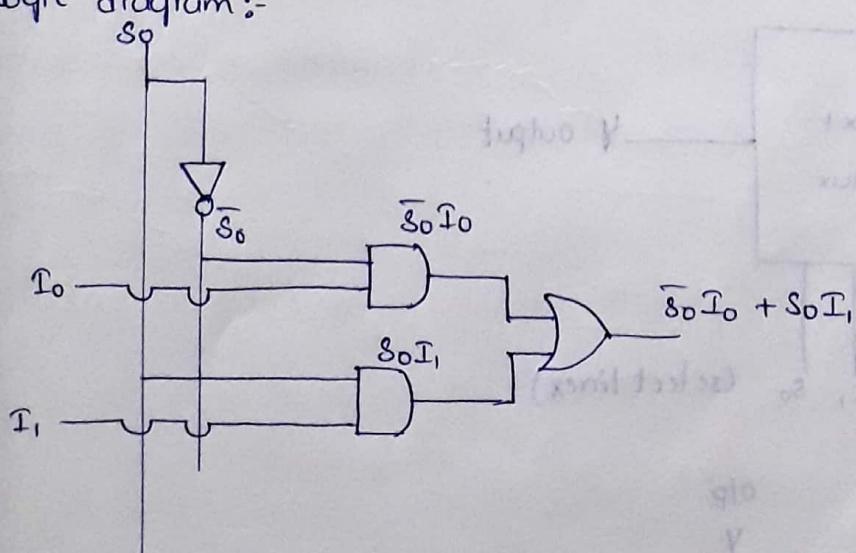
Truth Table:-

Selected lines ( $s_0$ )	output (y)
0	$I_0$
1	$I_1$

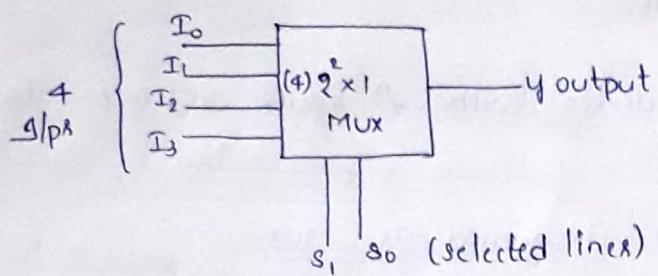
Boolean expression

$$y = \bar{s}_0 I_0 + s_0 I_1$$

Logic diagram:-



\* Design 4:1 MUX



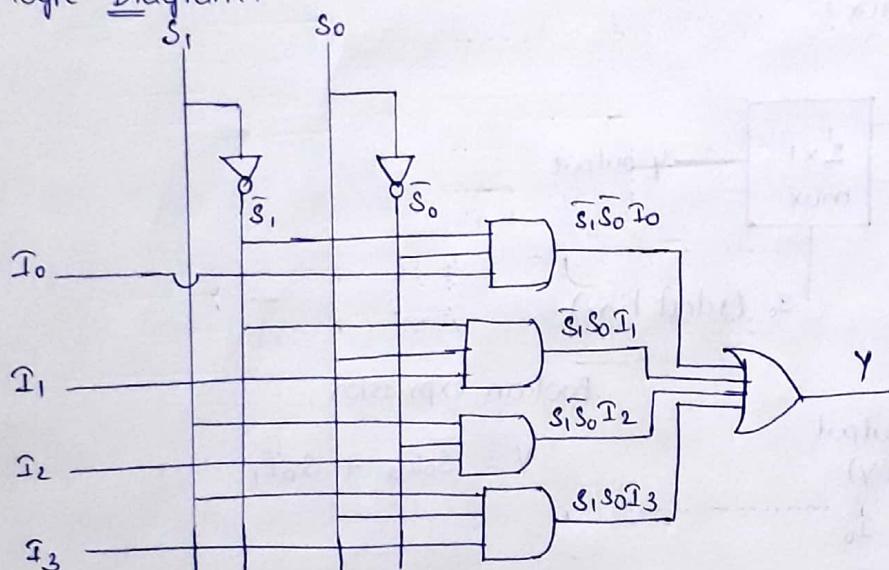
Truth Table:-

Select lines (S <sub>1</sub> )	(S <sub>0</sub> )	output (Y)
0	0	I <sub>0</sub>
0	1	I <sub>1</sub>
1	0	I <sub>2</sub>
1	1	I <sub>3</sub>

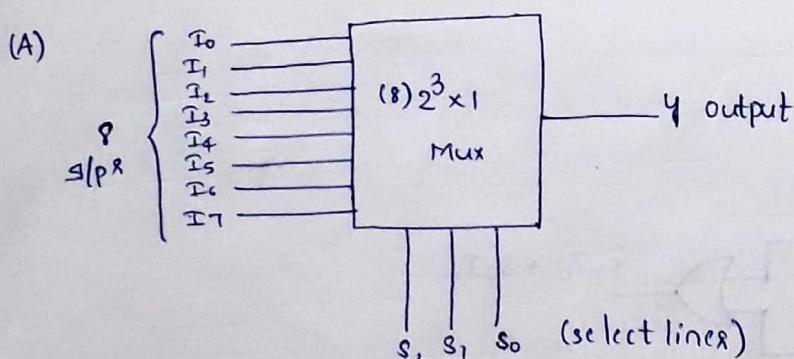
Boolean expression:-

$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

Logic Diagram:-



\* Design 8:1 MUX:-



Truth Table:-

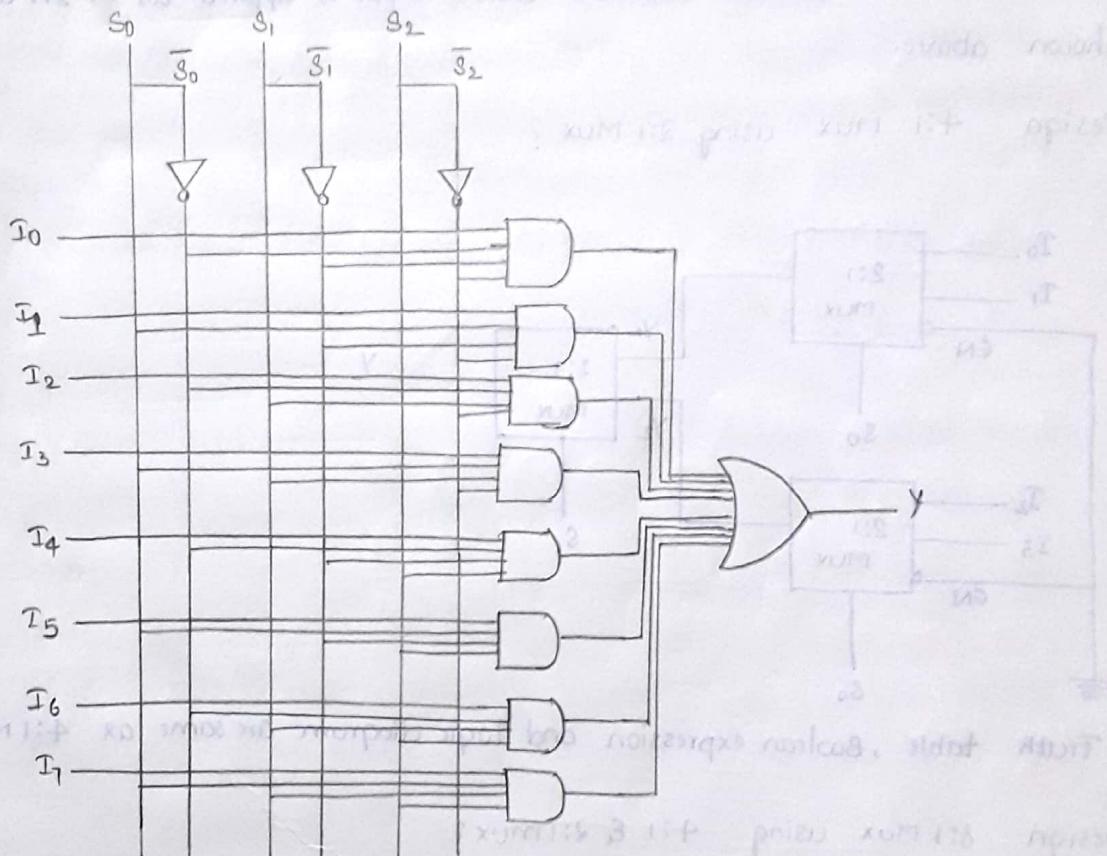
Select lines S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	O/P
0	0	0	Y
0	0	1	I <sub>0</sub>
0	1	0	I <sub>1</sub>
0	1	1	I <sub>2</sub>

1	0	0	I <sub>4</sub>
1	0	1	I <sub>5</sub>
1	1	0	I <sub>6</sub>
1	1	1	I <sub>7</sub>

Boolean expression:-

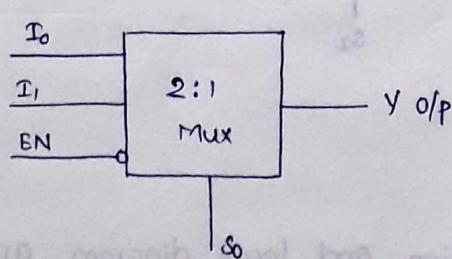
$$Y = \bar{s}_0 \bar{s}_1 \bar{s}_2 I_0 + s_0 \bar{s}_1 \bar{s}_2 I_1 + \bar{s}_0 s_1 \bar{s}_2 I_2 + s_0 s_1 \bar{s}_2 I_3 + \bar{s}_0 \bar{s}_1 s_2 I_4 + s_0 \bar{s}_1 s_2 I_5 \\ + \bar{s}_0 s_1 s_2 I_6 + s_0 s_1 s_2 I_7$$

Logic diagram:-



Enable Input :- Enable is a active low input that is when enable is zero it selects the mux . When enable is one it deselects the multiplexer(MUX) and the output is zero.

\* Design 2:1 Mux with enable o/p:-

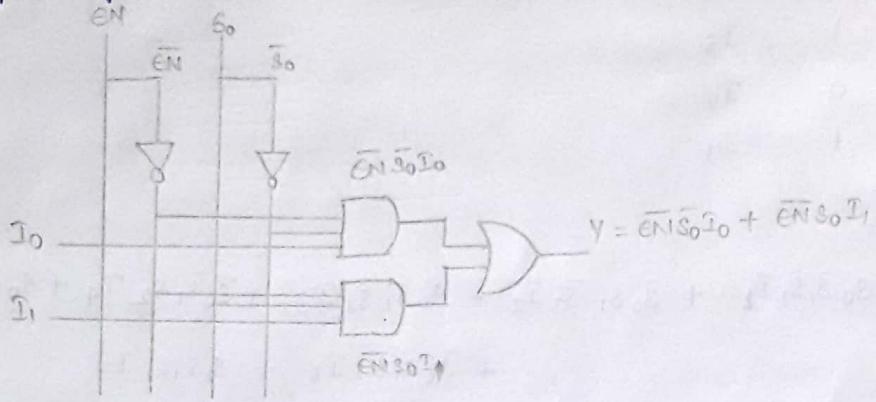


Truth Table		
EN	\$S_0\$	O/P
0	0	I <sub>0</sub>
0	1	I <sub>1</sub>

Boolean expression:-

$$Y = \bar{EN} \bar{s}_0 I_0 + \bar{EN} s_0 I_1$$

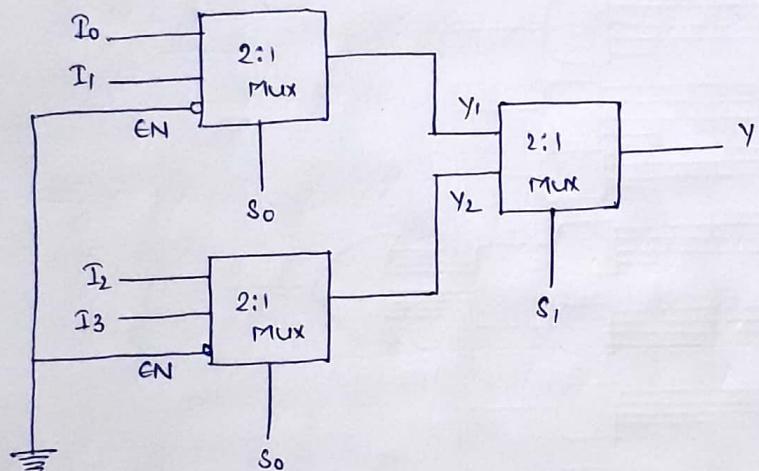
Logic diagram:-



Note:- 4:1 Mux and 8:1 Mux design with enable is same as general 4:1 and 8:1 Mux except an addition enable input is applied as in 2:1 design shown above.

(1) Design 4:1 Mux using 2:1 Mux ?

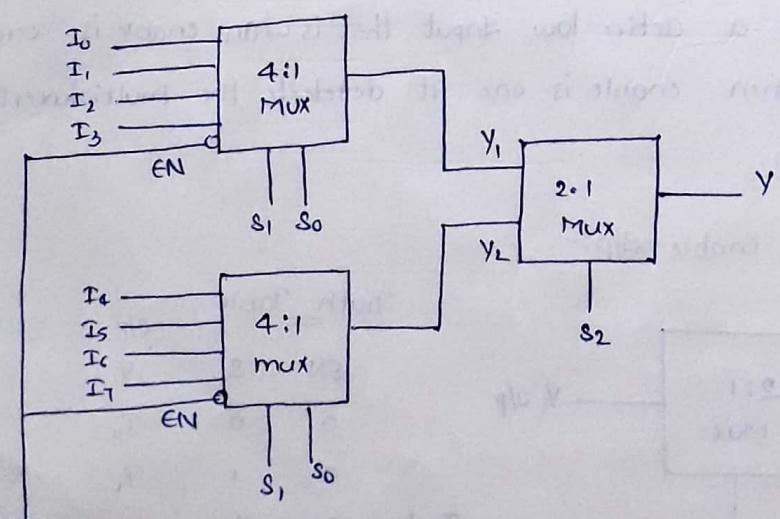
(A)



Truth table, Boolean expression and logic diagram are same as 4:1 Mux.

(2) Design 8:1 Mux using 4:1 & 2:1 mux ?

(A)



Truth table, Boolean expression and logic diagram are same as 8:1 Mux

(3) Design 16:1 Mux by using 8:1 & 2:1?

(4) Implement the following Boolean expression  $F(a,b,c) = \sum m(1,3,5,6)$  by using 4:1 MUX?

(A) Given function F contains three variables a,b,c and the design is by 4:1 MUX.

4:1 MUX requires two select lines so from the three variables a,b,c any two variables can be used as a select lines. (a,b on b,c or a,c) and remaining variable  $(a(0),b(0),c)$  is used as a S/p line.

Implementation Table:-

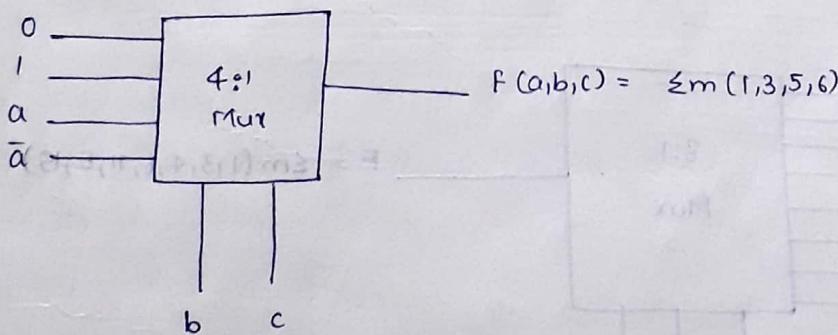
To write implementation table the remaining variable  $(a(0),b(0),c)$  is taken in complement form and normal form and write the minterms equally. If the two minterms are grouped or circled o/p is '1'. If no minterm is grouped o/p is '0'. If any one of the minterm is grouped write the corresponding variable in o/p.

$$\Rightarrow f(a,b,c) = \sum m(1,3,5,6)$$

↓      ↗  
select lines

Implementation Table

$\bar{a}$	0	1	2	3
a	4	5	6	7
	0	1	a	$\bar{a}$



(5) Design Boolean expression  $F = \prod m(0,1,2,3,6,9,11,13,14)$  using 8x1 mux

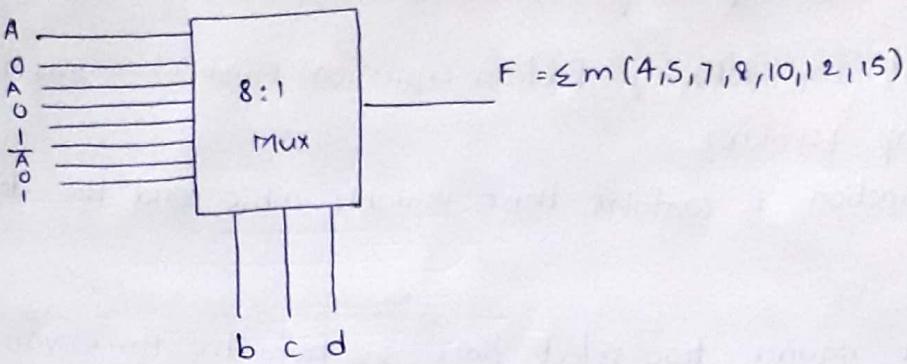
(A) Given pos expression  $F = \prod M(0,2,3,6,9,11,13,14)$

Sop expression  $F = \sum m(4,5,7,8,10,12,15) = F(A,B,C,D)$

$\bar{A}$	0	1	2	3	4	5	6	7
-----------	---	---	---	---	---	---	---	---

↓      ↗  
select lines

A	8	9	10	11	12	13	14	15
	0	A	0	1	$\bar{A}$	0	1	



(6) Design 8x1 Mux for the boolean function?

$$F(A_1B_1C_1D) = \bar{A}B\bar{D} + ACD + \bar{B}CD + \bar{A}\bar{C}D.$$

$$\begin{aligned}
 (A) \quad F(A_1B_1C_1D) &= \bar{A}B\bar{D}(C+\bar{C}) + ACD(B+\bar{B}) + \bar{B}CD(A+\bar{A}) + \bar{A}\bar{C}D(B+\bar{B}) \\
 &= \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + AB\bar{C}D + A\bar{B}CD + A\bar{B}CD + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} \\
 &\quad (6) \quad (4) \quad (15) \quad (11) \quad (11) \quad (3) \quad (5) \quad (1) \\
 &= \Sigma m(1, 3, 4, 5, 6, 11, 15)
 \end{aligned}$$

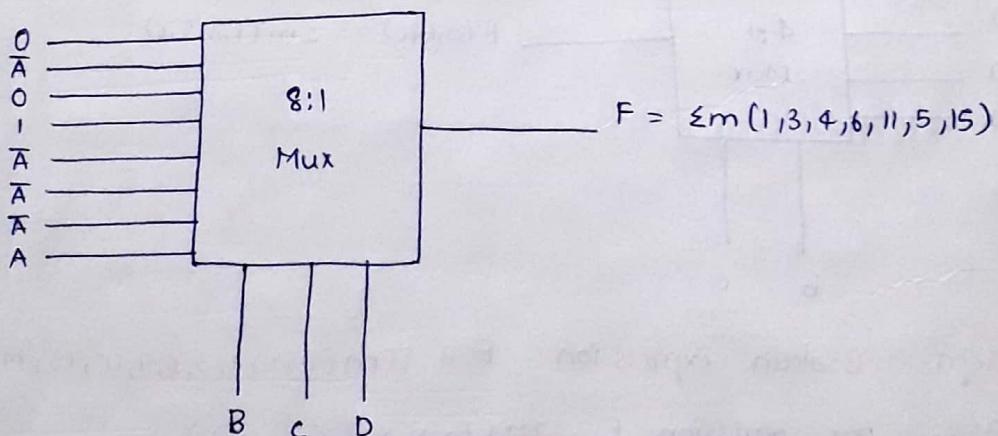
$$F(A_1B_1C_1D) = \Sigma m(1, 3, 4, 5, 6, 11, 15)$$

↓  
select lines

Implementation table:

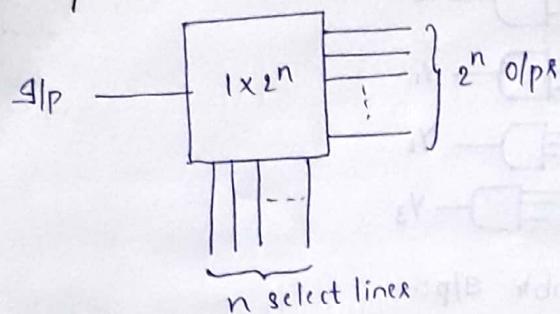
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	0	$\bar{A}$	0	1	$\bar{A}$	$\bar{A}$	$\bar{A}$	A

Block diagram:-



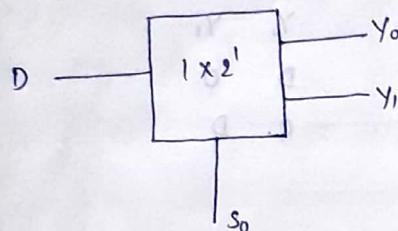
Demultiplexer:- Demultiplexer is a device with one input and  $2^n$  outputs and  $n$  select lines. Demultiplexer is quite opposite to the multiplexer.

Block Diagram:-



(1) Design  $1 \times 2$  Demux?

(A)



Truth Table

Select line

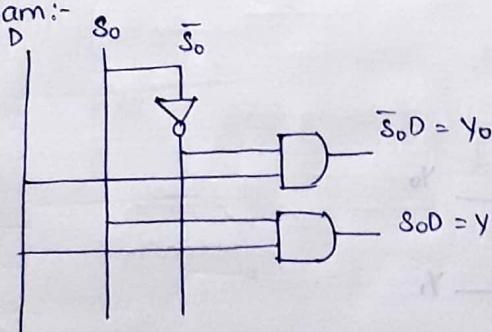
	O/p's	
$s_0$	$y_0$	$y_1$
0	D	0
1	0	D

Boolean expression:-

$$y_0 = \bar{s}_0 D + s_0 \cdot 0 = \bar{s}_0 D$$

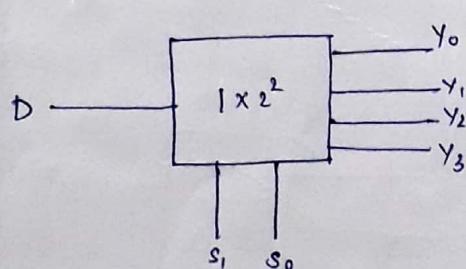
$$y_1 = \bar{s}_0 \cdot 0 + s_0 \cdot D = s_0 D$$

Logic diagram:-



(2) Design  $1 \times 4$  Demux?

(A)



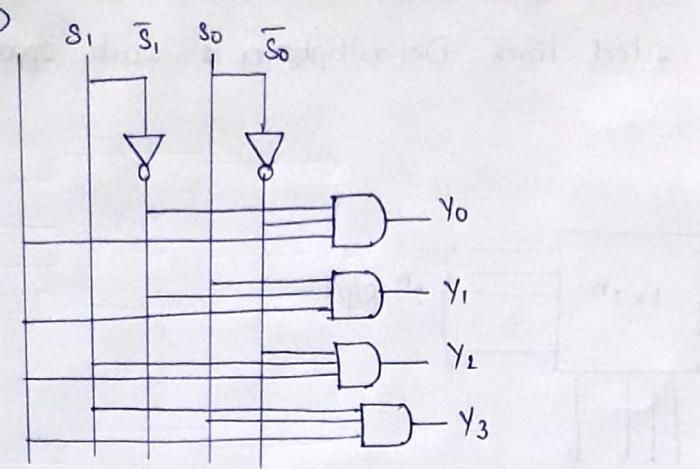
Truth Table :-

$s_1$	$s_0$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Boolean expression:-

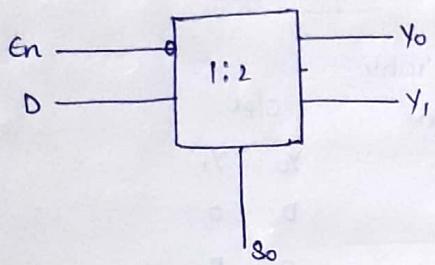
$$y_0 = \bar{s}_0 \bar{s}_1 D, y_1 = s_0 \bar{s}_1 D, y_2 = \bar{s}_0 s_1 D, y_3 = s_0 s_1 D$$

Logic diagram:-



(3) Design 1:2 Demux using enable S/p:-

(A)



Truth table:-

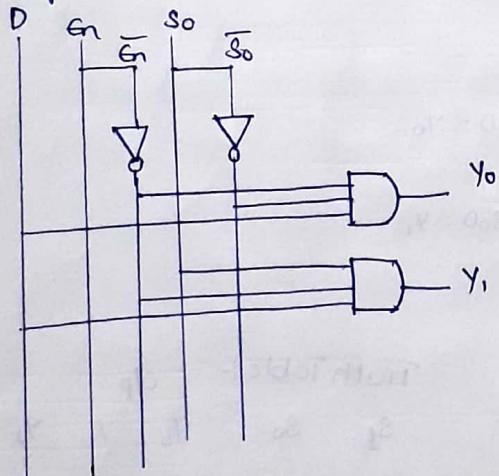
En	$S_0$	o/p	
		$y_0$	$y_1$
0	0	D	0
0	1	0	D

Boolean expression:-

$$y_0 = \bar{E}n \bar{S}_0 D$$

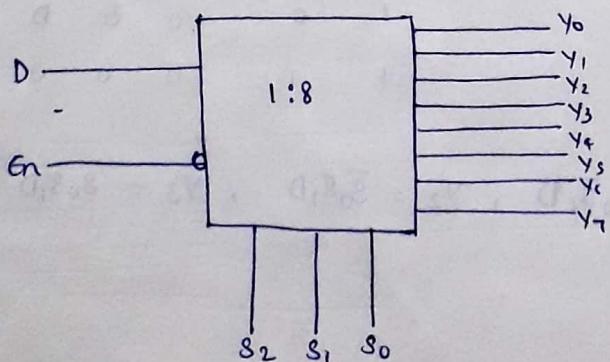
$$y_1 = \bar{E}n S_0 D$$

Logic diagram:-



(4) Design 1:8 Demux using enable S/p?

(A)



Truth Tables

of 8x1

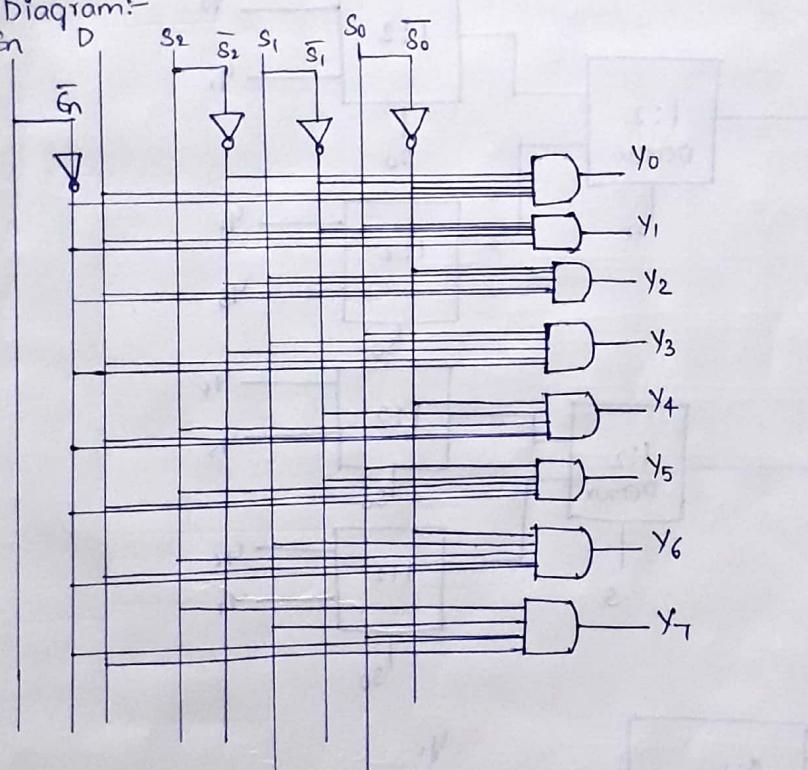
$E_n$	$S_2$	$S_1$	$S_0$	$y_0$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$
0	0	0	0	D	0	0	0	0	0	0	0
0	0	0	1	0	D	0	0	0	0	0	0
0	0	1	0	0	0	D	0	0	0	0	0
0	0	1	1	0	0	0	D	0	0	0	0
0	1	0	0	0	0	0	0	D	0	0	0
0	1	0	1	0	0	0	0	0	D	0	0
0	1	1	0	0	0	0	0	0	0	D	0
0	1	1	1	0	0	0	0	0	0	0	D

Boolean expression:-

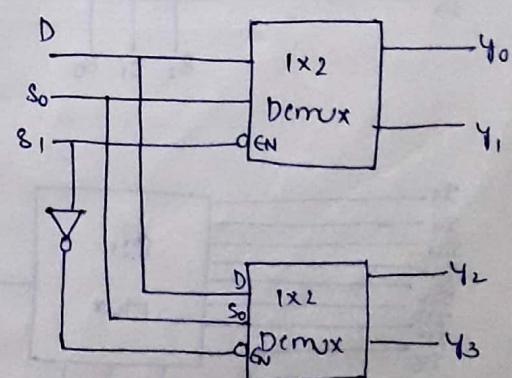
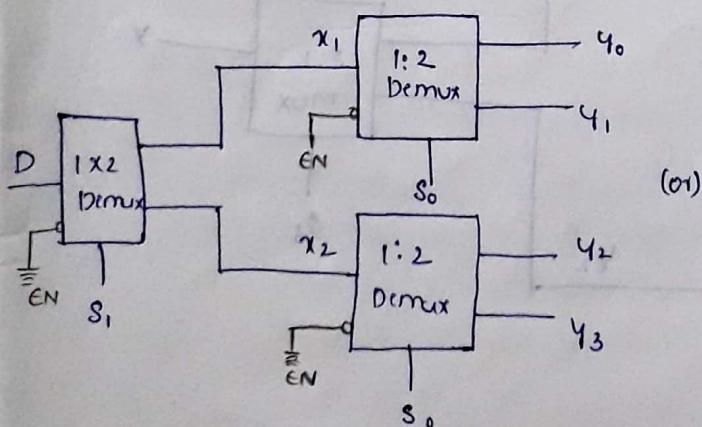
$$y_0 = \bar{S}_0 \bar{S}_1 \bar{S}_2 D \bar{E}_n, \quad y_1 = S_0 \bar{S}_1 \bar{S}_2 D \bar{E}_n, \quad y_2 = \bar{S}_0 S_1 \bar{S}_2 D \bar{E}_n, \quad y_3 = S_0 S_1 \bar{S}_2 D \bar{E}_n,$$

$$y_4 = \bar{S}_0 \bar{S}_1 S_2 D \bar{E}_n, \quad y_5 = S_0 \bar{S}_1 S_2 D \bar{E}_n, \quad y_6 = \bar{S}_0 S_1 S_2 D \bar{E}_n, \quad y_7 = S_0 S_1 S_2 D \bar{E}_n$$

Logic Diagram:-



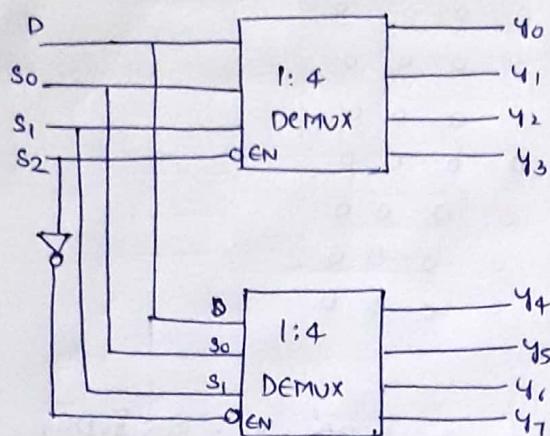
b) Design 1:4 Demux using 1:2 Demux?



Boolean expression, logic diagram, truthtable is same as previous.

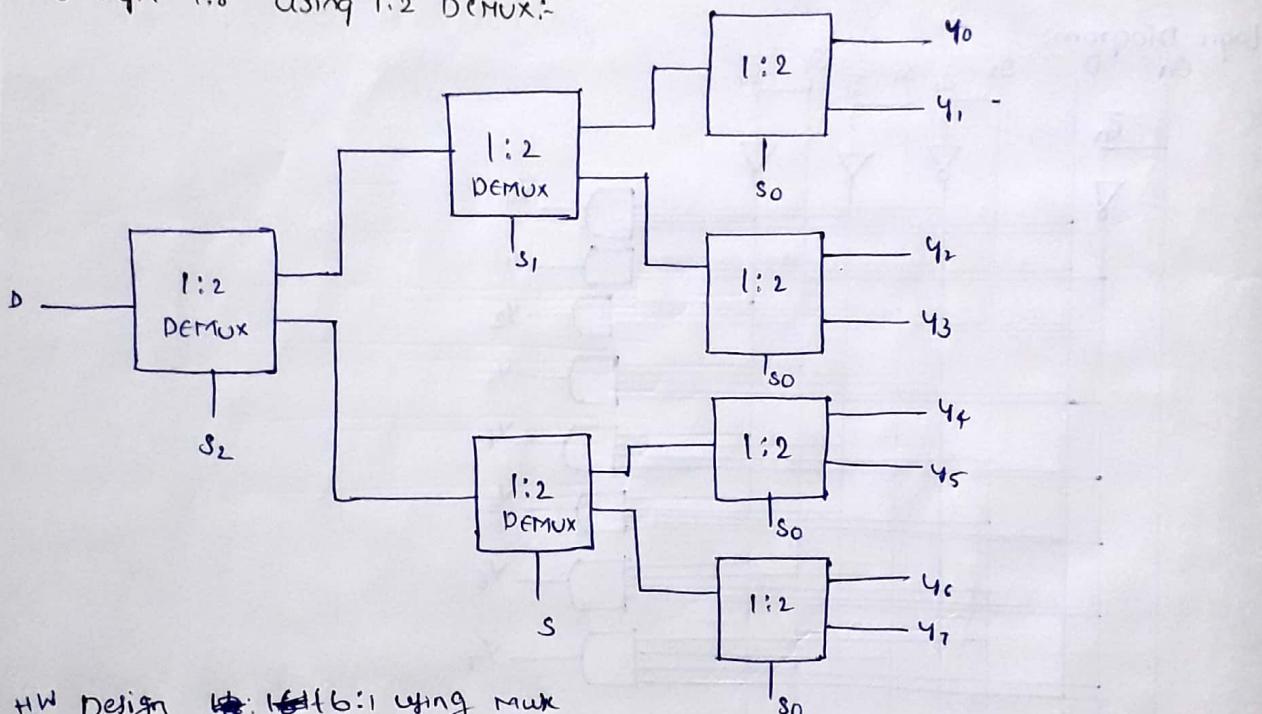
(6) Design 1:8 using 1:4 demux?

(A)



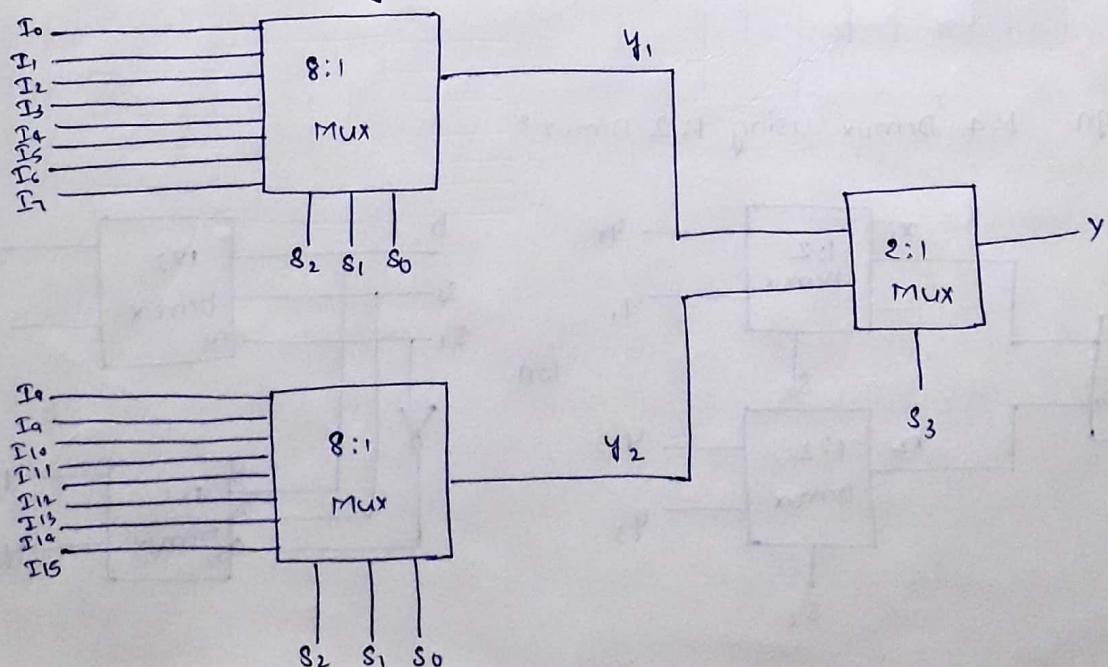
Boolean expression, Truthtable, logic diagram are same as 1:8 DEMUX.

(7) Design 1:8 using 1:2 DEMUX:



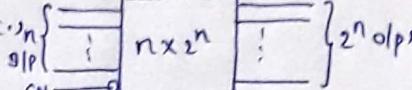
HW Design 1:6 using MUX

(3)

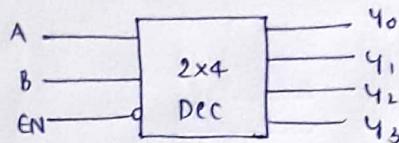


Decoder:- Decoder is a circuit that converts 'n' inputs into  $2^n$  o/p's.

It has single enable s/p. Decoders are a type of demultiplexers.

Eq:-  $2 \times 4$ ,  $3 \times 8$ ,  $4 \times 16$ ,  $5 \times 32$  etc.,  $\{$    $\}$

(1) Design a  $2 \times 4$  Decoder?



Truth Table

S/p			O/p			
EN	A	B	$y_0$	$y_1$	$y_2$	$y_3$
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

Boolean expression:-

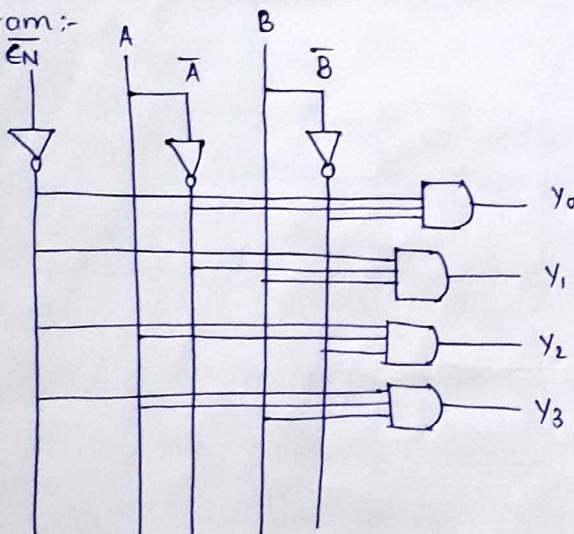
$$y_0 = \overline{EN} \overline{A} \overline{B}$$

$$y_1 = \overline{EN} \overline{A} B$$

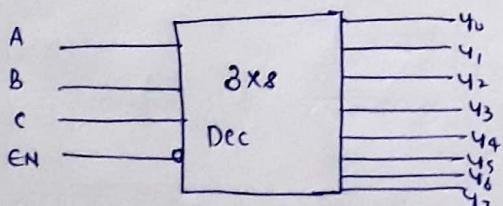
$$y_2 = \overline{EN} A \overline{B}$$

$$y_3 = \overline{EN} A B$$

Logic diagram:-



(2) Design  $3 \times 8$  Decoder?



Truth Table:-

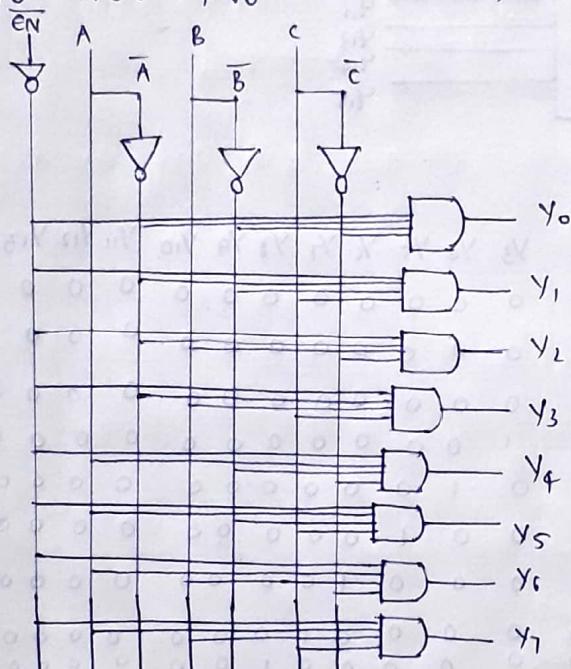
S/p				O/p							
EN	A	B	C	$y_0$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0

0	1	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	0	0	1

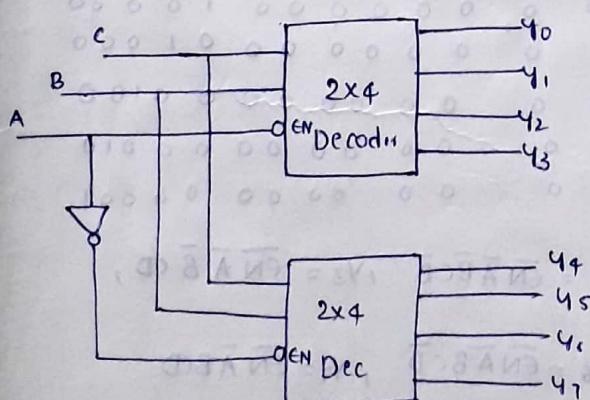
Boolean expression:-

$$y_0 = \overline{EN} \overline{ABC}, y_1 = \overline{EN} \overline{AB}\bar{C}, y_2 = \overline{EN} \overline{A}\overline{B}\bar{C}, y_3 = \overline{EN} \overline{ABC}, y_4 = \overline{EN} \overline{AB}\bar{C}$$

$$y_5 = A\overline{B}C\overline{EN}, y_6 = \overline{EN}ABC, y_7 = \overline{EN}ABC$$

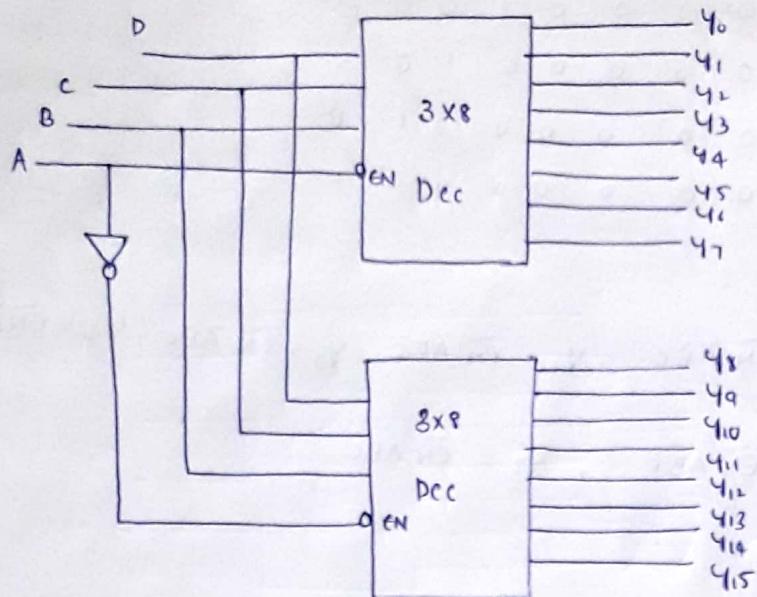


(3) Design 3x8 Decoder using 2x4 Dec:-



The Boolean expression, Truth table & logic diagram are same as above 3x8 Dec.

(4) Design 4x16 by using 3x8 Dec?



Truth Table :-

Inps				Outps																
EN	A	B	C	D	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>	Y <sub>8</sub>	Y <sub>9</sub>	Y <sub>10</sub>	Y <sub>11</sub>	Y <sub>12</sub>	Y <sub>13</sub>	Y <sub>14</sub>	Y <sub>15</sub>
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Boolean expression:-

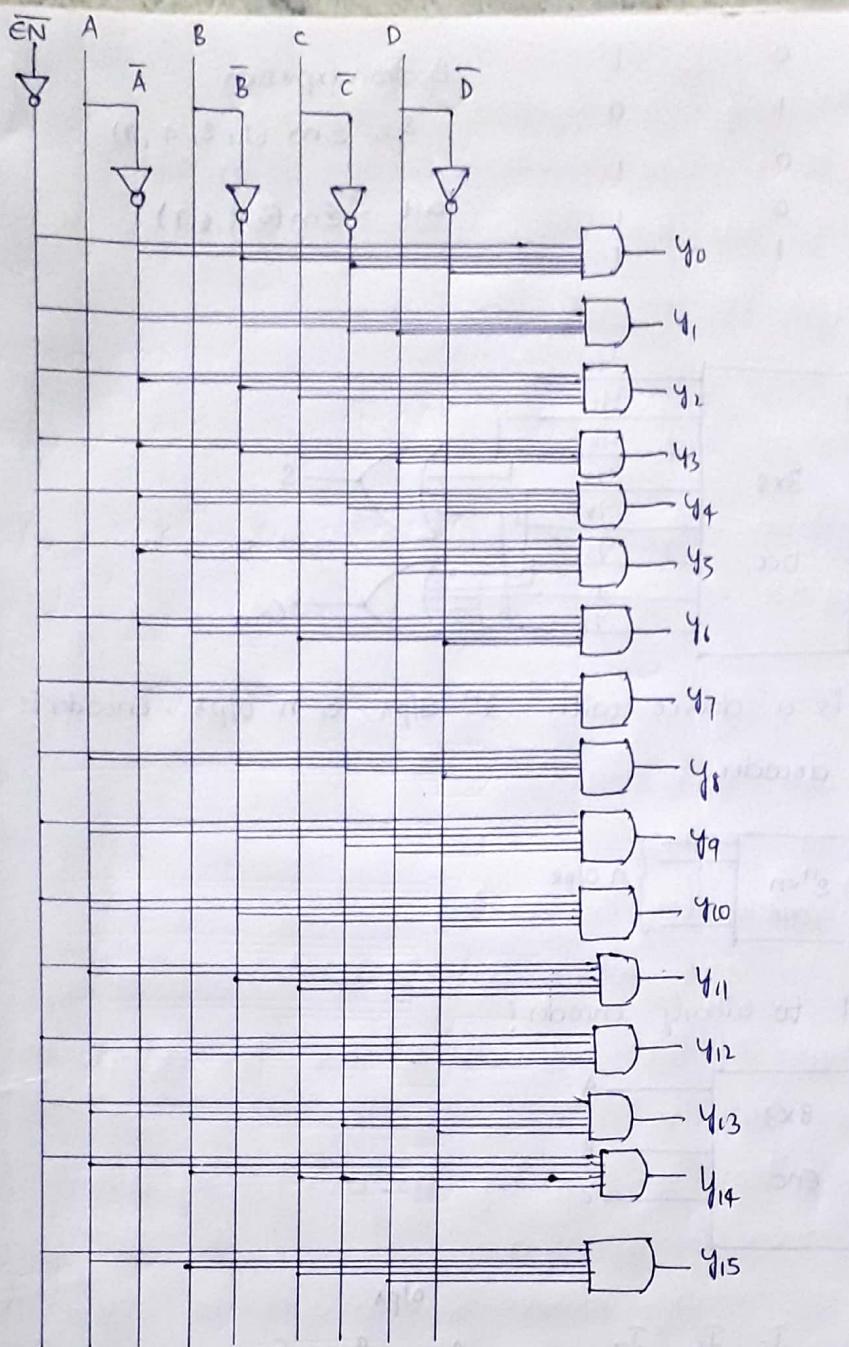
$$Y_0 = \overline{EN} \overline{A} \overline{B} \overline{C} \overline{D}, \quad Y_1 = \overline{EN} \overline{A} \overline{B} \overline{C} D, \quad Y_2 = \overline{EN} \overline{A} \overline{B} C \overline{D}, \quad Y_3 = \overline{EN} \overline{A} \overline{B} C D,$$

$$Y_4 = \overline{EN} \overline{A} B \overline{C} \overline{D}, \quad Y_5 = \overline{EN} \overline{A} B \overline{C} D, \quad Y_6 = \overline{EN} \overline{A} B C \overline{D}, \quad Y_7 = \overline{EN} \overline{A} B C D$$

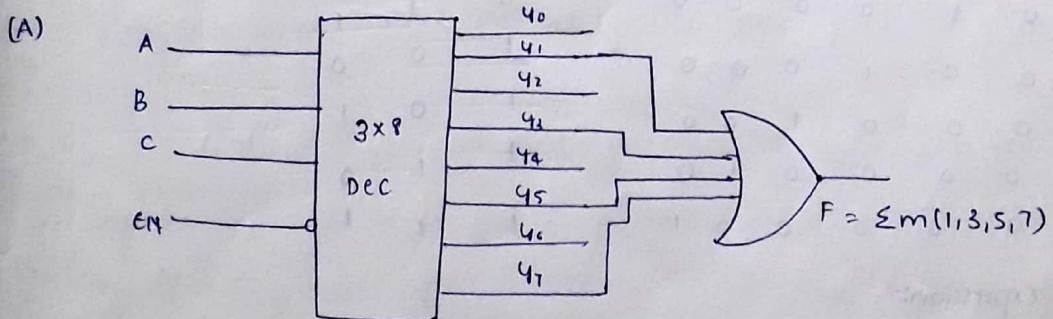
$$Y_8 = \overline{EN} A \overline{B} \overline{C} \overline{D}, \quad Y_9 = \overline{EN} A \overline{B} \overline{C} D, \quad Y_{10} = \overline{EN} A \overline{B} C \overline{D}, \quad Y_{11} = \overline{EN} A \overline{B} C D$$

$$Y_{12} = \overline{EN} A B \overline{C} \overline{D}, \quad Y_{13} = \overline{EN} A B \overline{C} D, \quad Y_{14} = \overline{EN} A B C \overline{D}, \quad Y_{15} = \overline{EN} A B C D$$

Logic diagram :-



(5) Design logic function  $F(A,B,C) = \sum m(1,3,5,7)$  using decoder?

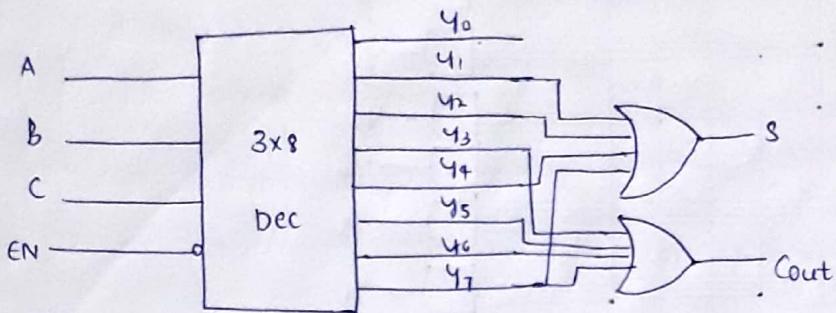


(6) Design full adder using decoder?

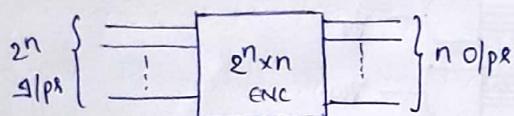
Full adder:-

	A	B	Cin	S	Cout
$m_0$	0	0	0	0	0
$m_1$	0	0	1	1	0
$m_2$	0	1	0	1	0

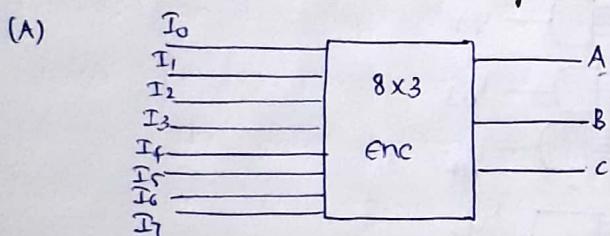
$m_3$	0	1	1	0	1		Boolean expression
$m_4$	1	0	0	1	0		$S = \sum m(1, 2, 4, 7)$
$m_5$	1	0	1	0	1		
$m_6$	1	1	0	0	1		$(out = \sum m(3, 5, 6, 7))$
$m_7$	1	1	1	1	1		



ENCODER:- Encoder is a device with  $2^n$  inputs &  $n$  outputs. Encoder is quite opposite to decoder.



① Design a octal to binary encoder?



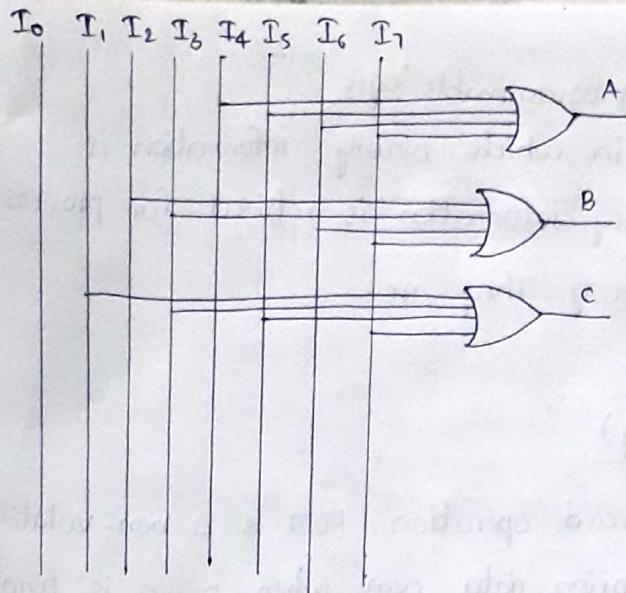
I/p's							O/p's			
$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Boolean expression:-

$$A = I_4 + I_5 + I_6 + I_7, \quad B = I_2 + I_3 + I_6 + I_7,$$

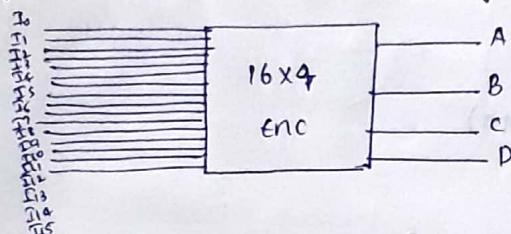
$$C = I_1 + I_3 + I_5 + I_7$$

Logic diagram:-



(2) Design a Hexa decimal to binary encoder?

(A)



$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$I_8$	$I_9$	$I_{10}$	$I_{11}$	$I_{12}$	$I_{13}$	$I_{14}$	$I_{15}$	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	

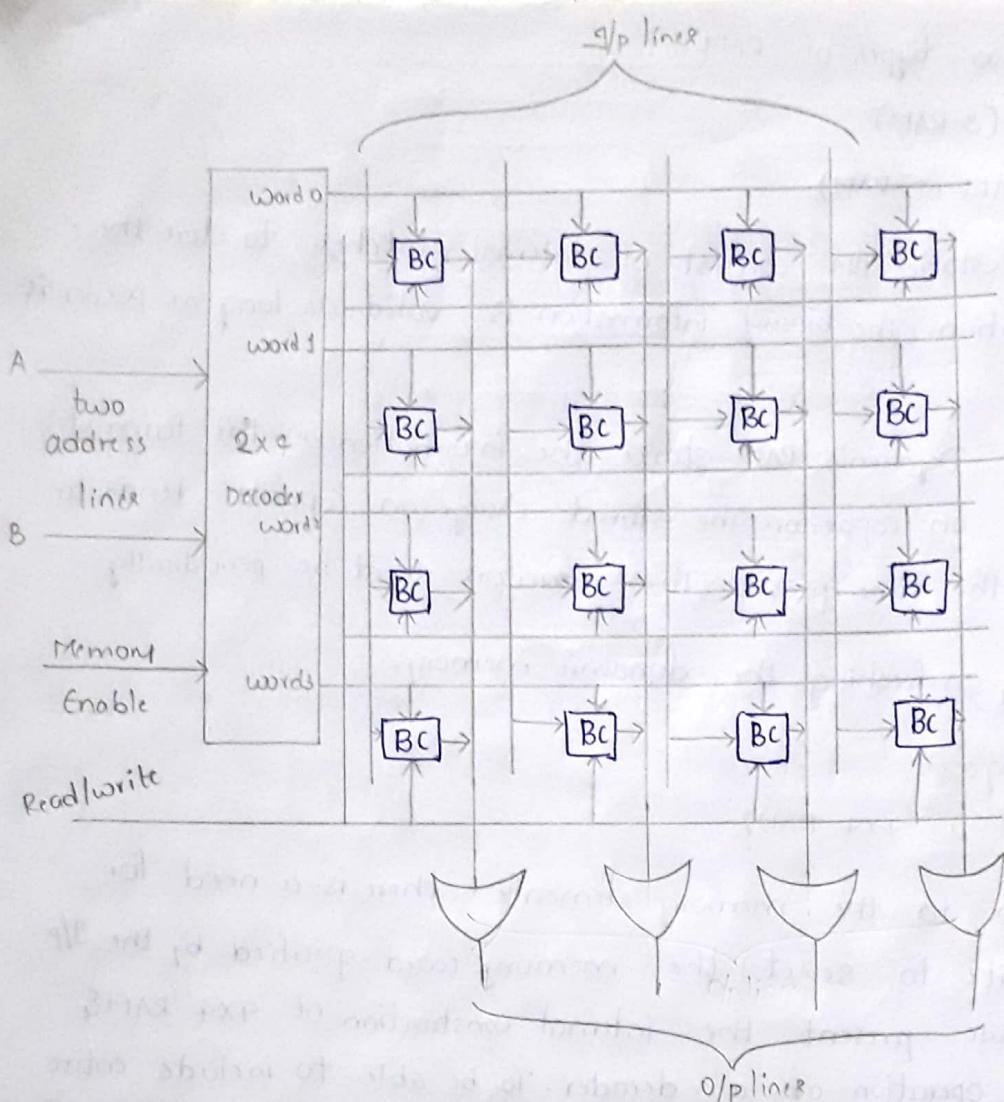
Boolean expression:-

$$A = I_8 + I_9 + I_{10} + I_{11} + I_{12} + I_{13} + I_{14} + I_{15}$$

$$B = I_4 + I_5 + I_6 + I_7 + I_{12} + I_{13} + I_{14} + I_{15}$$

$$C = I_2 + I_3 + I_6 + I_7 + I_{10} + I_{11} + I_{14} + I_{15}$$

$$D = I_1 + I_2 + I_5 + I_7 + I_9 + I_{11} + I_{13} + I_{15}$$



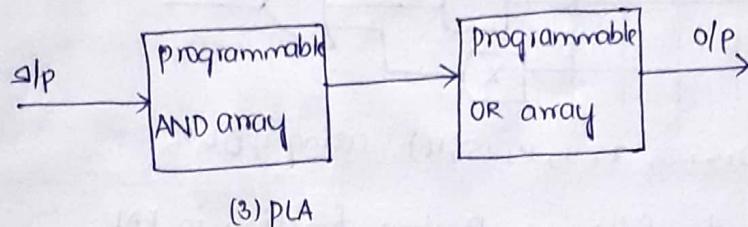
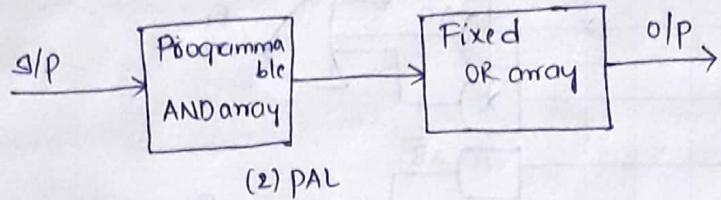
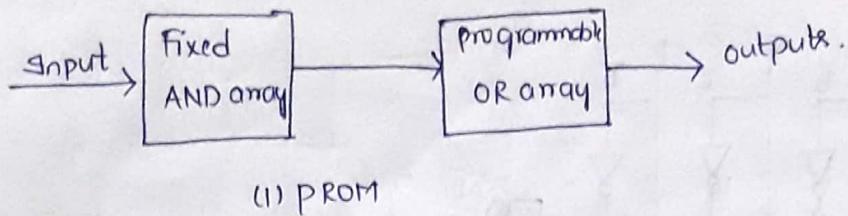
\* The logical construction of RAM shown above consist of 4 words 4 bits each and has a total of 16 binary cells. The two address lines go through the  $2 \times 4$  decoder to provide 4 words for RAM when enable  $S/I/P$  is zero O/p's of decoder are zero. When enable is '1', one of the 4 words is selected as dictated by the  $S/I/P$  lines.

Combinational PLD's:- (PLD - programmable logic device)

\* It is an integrated circuit constructed with AND array and OR array. AND-OR array gives the sum of product terms. There are three types of combinational PLD's.

- (1) programmable ROM (P-ROM)
- (2) programmable array logic (PAL)
- (3) programmable logic array (PLA)

The block diagrams for combinational PLD's are shown in fig below.



**Programmable Array Logic:-** PAL is a PLD device with programmable AND array & fixed OR array. PAL is easier to program & compute compared to PLA. Note:- The number of product terms or minterms are limited to  $(n-1)$ .

Eq: Implement the following Boolean functions.

$F_1 = \Sigma m(0, 1, 2, 3, 7, 8, 10, 12, 14)$ ,  $F_2 = \Sigma m(0, 1, 2, 3, 4, 6, 8, 10, 12, 14)$  using PAL?

	AB	CD	00	01	11	10
00	1	1	1	1	1	1
01	4	6	7	6		
11	1	12	13	15	14	
10	1	9	11	1	10	

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

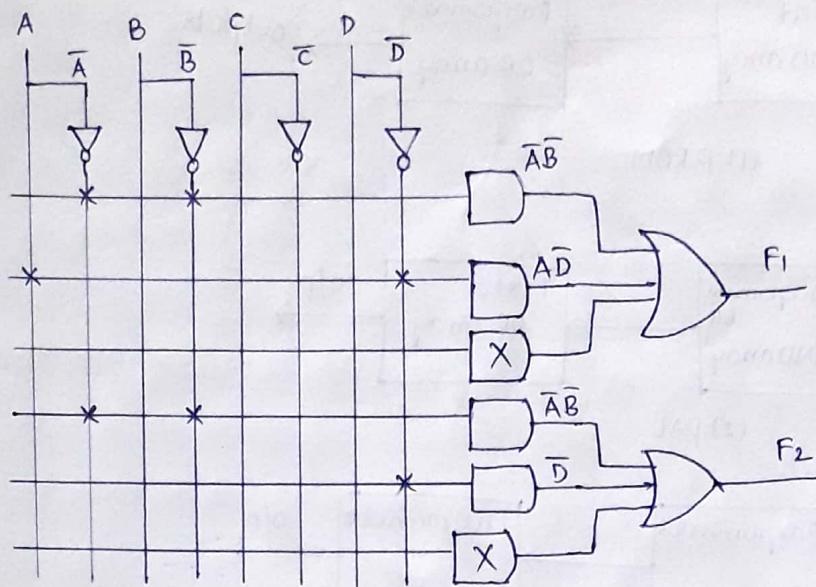
	AB	CD	00	01	11	10
00	1	1	1	1	1	1
01	4	5	6	7	6	5
11	1	12	13	15	14	1
10	1	9	11	1	10	1

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

PAL programming table

product term	AND Inputs			
A	B	C	D	
1 $\bar{A}\bar{B}$	0	0	-	-
2 $A\bar{D}$	1	-	-	0
3 -	-	-	-	-
4 $\bar{A}\bar{B}$	0	0	-	-
5 $\bar{D}$	-	-	-	0
6 -	-	-	-	-

Logic diagram:-



② Implement  $F = \sum m(0, 2, 3, 7, 9, 11, 15, 16)$  using PAL?

③ Design & Implement following Boolean function in PAL

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

$$X(A, B, C, D) = \sum (1, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \sum (0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \sum (1, 4, 8, 12, 13)$$

		DE		BC		A=0	
		00	01	11	10		
		00	1	1	1	1	1
		01	4	5	1	7	6
		11	12	18	1	5	14
		10	8	19	1	11	10

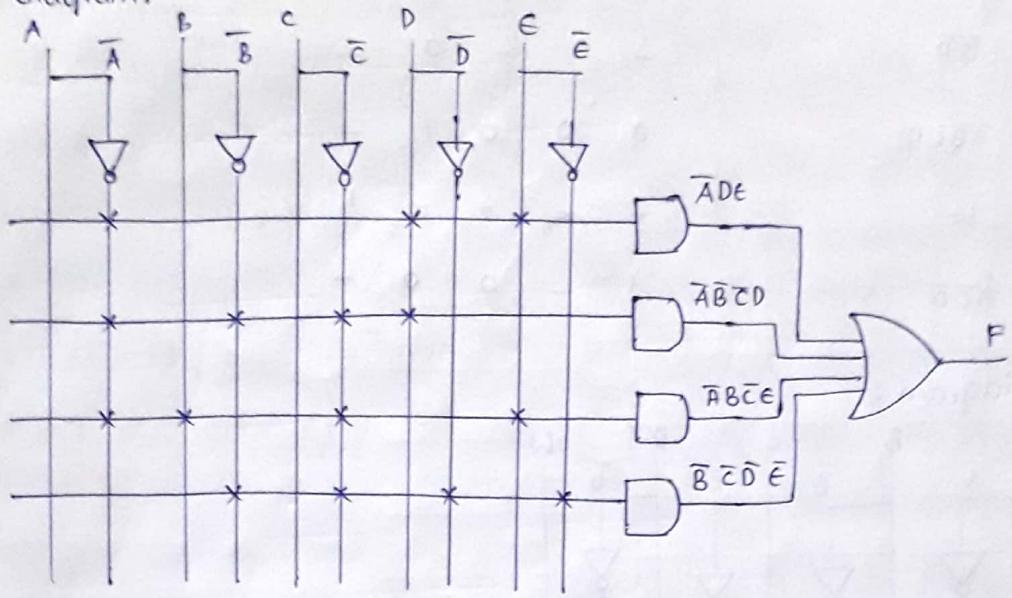
		DE		BC		A=1	
		00	01	11	10		
		00	1	16	17	19	18
		01	20	21	23	22	
		11	28	29	31	30	
		10	24	25	27	26	

$$F = \bar{A}DE + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}E + \bar{B}\bar{C}\bar{D}\bar{E}$$

PAL Programming table

product term	AND S/p				
	A	B	C	D	E
1 $\bar{A}DE$	0	—	—	1	1
2 $\bar{A}\bar{B}\bar{C}D$	0	0	0	1	—
3 $\bar{A}B\bar{C}E$	0	1	0	—	1
4 $\bar{B}\bar{C}\bar{D}\bar{E}$	—	0	0	0	0

Logic diagram:-



③ (a)

	AB	CD	00	01	11	10
W → 00	0	1	3	1	2	
01	4	5	7	6		
11	12	13	15	14		
10	8	9	11	10		

$$W = ABC + \bar{A}\bar{B}C\bar{D}$$

Y

	AB	CD	00	01	11	10
00	1	0	1	3	1	2
01	1	1	5	17	1	6
11			12	13	15	14
10	1	1	9	11	1	10

$$Y = CD + \bar{A}B + \bar{B}\bar{D}$$

X →

	AB	CD	00	01	11	10
00	0	1	1	3	1	2
01	4	5	7	6		
11	12	13	15	14		
10	1	1	9	11	1	10

$$X = A + \bar{B}\bar{C}\bar{D}$$

Z →

	AB	CD	00	01	11	10
00	0	1	1	3	1	2
01	4	5	7	6		
11	12	13	15	14		
10	1	1	9	11	1	10

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

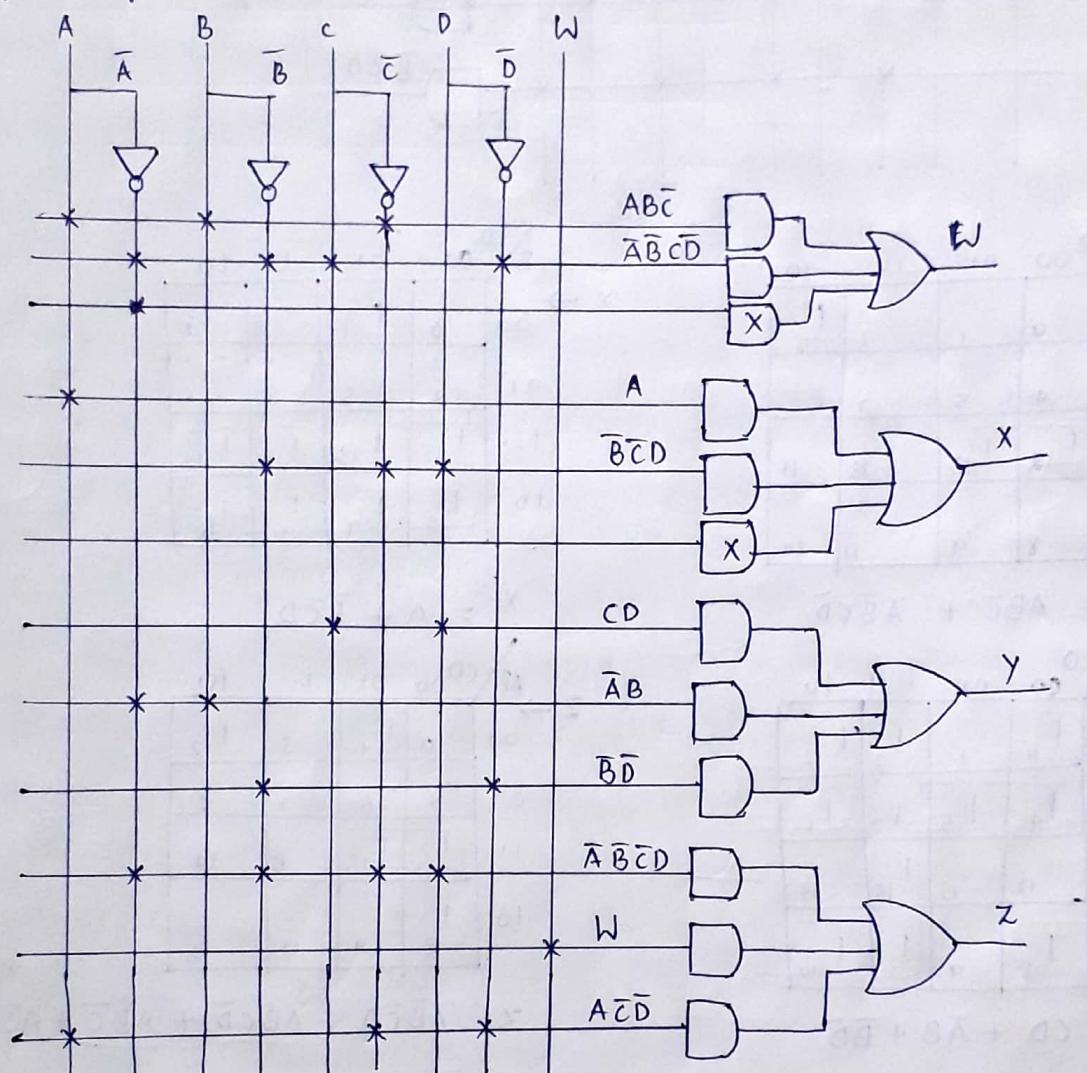
$$Z = \bar{A}\bar{B}\bar{C}\bar{D} + W + AC\bar{D}$$

PAL programming table.

product term	AND gate					W
	A	B	C	D		
1 ABC	1	1	0	-	-	
2 $\bar{A}\bar{B}C\bar{D}$	0	0	1	0	-	
3 -	-	-	-	-	-	
4 A	1	-	-	-	-	
5 $\bar{B}\bar{C}D$	-	0	0	1	-	
6 -	-	-	-	-	-	
7 CD	-	-	1	1	-	

8	$AB$	0	1	-	-	-
9	$B\bar{D}$	-	0	-	0	-
10	$\bar{A}\bar{B}\bar{C}D$	0	0	0	1	-
11	$W$	-	-	-	-	1
12	$\bar{A}\bar{C}\bar{D}$	1	-	0	0	-

Logic diagram :-



PLA :- (Programmable Logic array)

PLA is a PLD device with programmable AND array & programmable OR array.

Q1:- How PLA is similar to concept of PROM explain?

(A) PLA is similar to the concept of PROM except the decoder is replaced by an array of AND gates. That can be programmed to generate any product term of the 3 input variables. The product terms are connected to the OR gates to provide the SOP.

Q2:- Design & Implement the following Boolean functions using PLA:-

$$F_1(A, B, C) = \sum m(3, 5, 6, 7), \quad F_2(A, B, C) = \sum m(0, 2, 4, 7)$$

(A)

A	BC			
	00	01	11	10
0	0	1	1	2
1	4	5	7	6

$$F_1 = AC + AB + BC$$

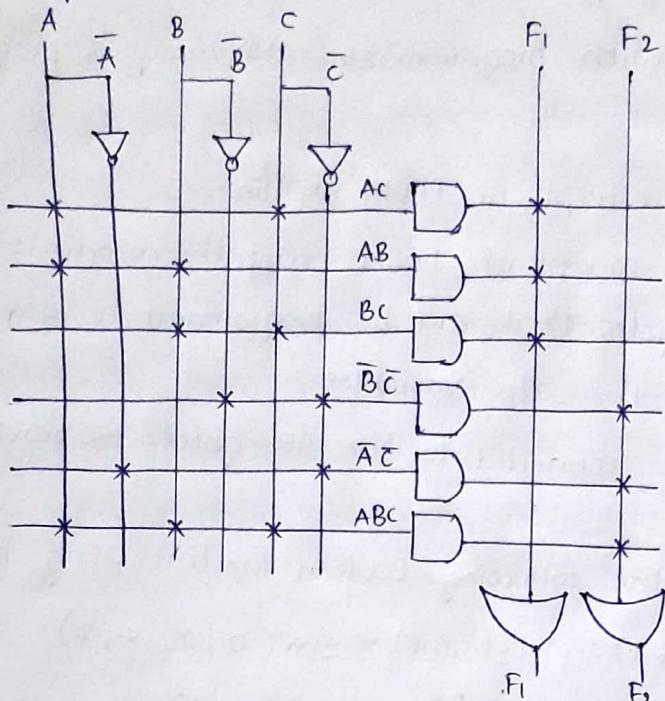
A	BC			
	00	01	11	10
0	0	1	3	2
1	4	5	7	6

$$F_2 = \overline{B}\overline{C} + \overline{A}\overline{C} + ABC$$

PLA programming table

product terms	AND outputs			OR outputs		(No limit for minterms)
	A	B	C	F <sub>1</sub>	F <sub>2</sub>	
1 AC	1	-	1	1	-	
2 AB	1	1	-	1	-	
3 BC	-	1	1	1	-	
4 $\overline{B}\overline{C}$	-	0	0	-	1	
5 $\overline{A}\overline{C}$	0	-	0	-	1	
6 ABC	1	1	1	-	1	

Logic diagram:-



② Design & Implement the following Boolean expression using PLA

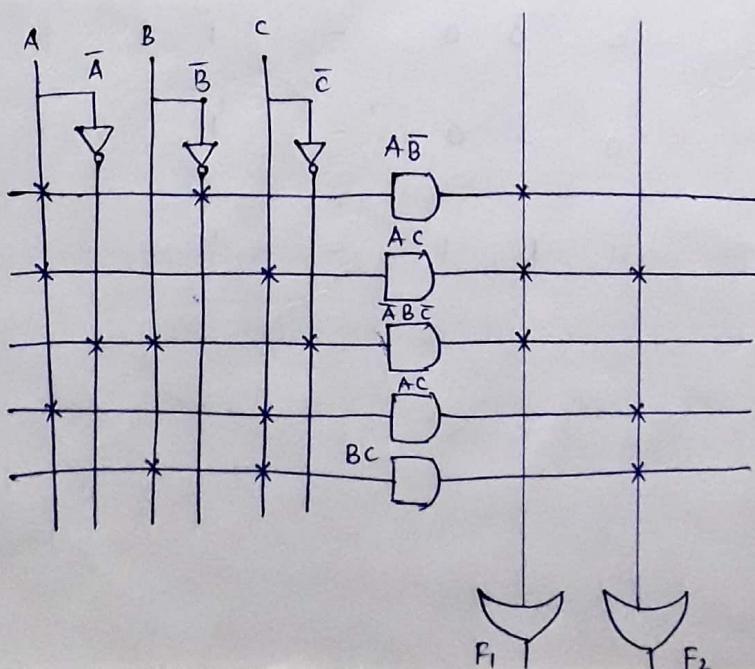
$$(i) F_1 = A\bar{B} + Ac + \bar{A}B\bar{C}, F_2 = Ac + Bc$$

$$(ii) F = \sum m(0, 2, 4, 5, 7, 9, 11, 15, 16)$$

(i)(A) PLA programming table

product term	AND S/PI			OR O/PI	
	A	B	C	$F_1$	$F_2$
1 $A\bar{B}$	1	0	-	1	-
2 $Ac$	1	-	1	1	1
3 $\bar{A}B\bar{C}$	0	1	0	1	-
4 $Ac$	1	-	1	-	1
5 $Bc$	-	1	1	-	1

logic Diagram:-



$$(iii) F = \sum m(0, 2, 4, 5, 7, 9, 11, 15, 16)$$

	DE	00	01	11	10
BC					
00		0	1	3	1
01		4	5	7	6
11		12	13	15	14
10		8	9	11	10

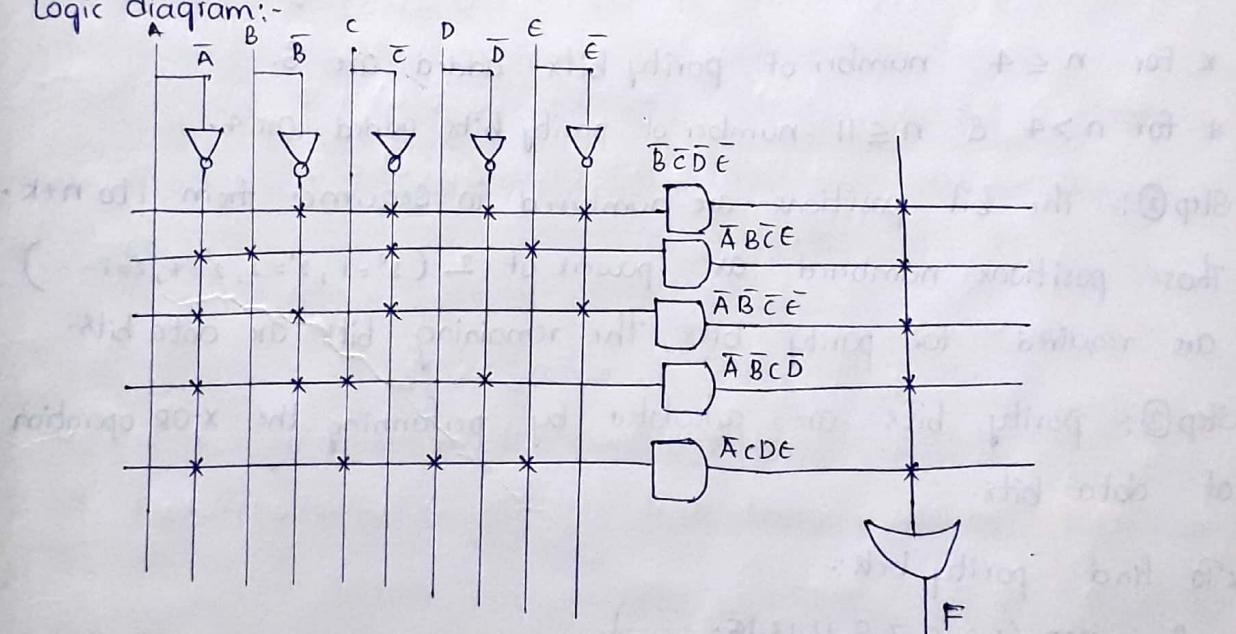
	DE	00	01	11	10
BC					
00		16	17	19	18
01		20	21	23	22
11		28	29	31	30
10		24	25	27	26

$$F = \overline{B}\overline{C}\overline{D}\overline{E} + \overline{A}\overline{B}\overline{C}E + \overline{A}\overline{B}\overline{C}\overline{E} + \overline{A}\overline{B}C\overline{D} + \overline{A}CDE$$

PLA programming Table:-

product term	AND g/p 8					OR O/P <sub>A</sub> (F)
	A	B	C	D	E	
1 $\overline{B}\overline{C}\overline{D}\overline{E}$	-	0	0	0	0	1
2 $\overline{A}\overline{B}\overline{C}E$	0	1	0	-	1	1
3 $\overline{A}\overline{B}\overline{C}\overline{E}$	0	0	0	-	0	1
4 $\overline{A}\overline{B}C\overline{D}$	0	0	1	0	-	1
5 $\overline{A}CDE$	0	-	1	1	1	1

Logic diagram:-



ERROR DETECTION AND CORRECTION:-

ERROR DETECTION:-

Def:- A single error occurs when a bit changes from 0 to 1 & 1 to 0. The most common method used to detect errors is parity scheme.

Parity means adding the extra bit to the transmitted data to make the transmitted data as even (even parity) (0) or odd (odd parity).

Even parity:- It means even number of ones in the transmitted or received data.