

Unit-2:-

Minimization of Boolean functions and
combinational logic circuits

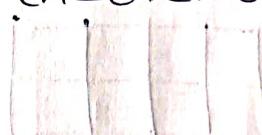
Introduction:-

Map method:- In previous chapter we have
some causes to simply the Boolean expressions
algebraically. But it is not a systematic
method.

⇒ We can never say whether the expression
minimal obtained is the real minimal
expression or not.

⇒ Effectiveness of the algebraic simplification
depends on the Boolean Algebraic rules,
laws and theorems.

K-map:-



. K-map method is a systematic method
of simplifying the boolean expression.

⇒ K-map is a chart or a graph composed
of an arrangement of adjacent cells, each
cell representing a 1x combination of
variables in SOP form or POS form.

⇒ K-map can be used for the problems
involving any no. of variables, it becomes
difficult for problems involving four or more
variables.

⇒ An 'n' variable function can have 2^n possible combinations of product terms; SOP form and 2^n possible combinations of sum terms in POS form.

⇒ ∵ K-map is a graphical representation of Boolean expression,

⇒ For two variables the K-map will have 2^2 cells, $2^3 = 8$ cells, $2^4 = 16$ cells.

Variable map:-

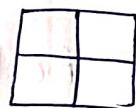
⇒ if $n=1$,

$$\text{cells} - 2^n = 2^1 = 2$$



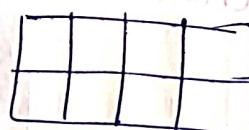
⇒ if $n=2$,

$$\text{cells} - 2^n = 2^2 = 4$$



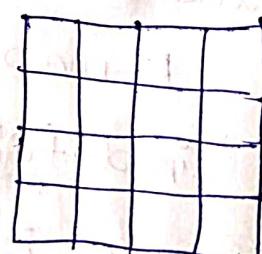
⇒ if $n=3$,

$$\text{cells} - 2^n = 2^3 = 8$$



⇒ if $n=4$

$$\text{cells} \rightarrow 2^n = 2^4 \Rightarrow 16$$



Plotting of K-map using SOP form.

One Variable map:-

⇒ $n=1$,

$$\text{cells} - 2^n$$

$$= 2^1 \Rightarrow 2$$

0	A(0)
1	A(1)

Two-variable map:-

$$n=2,$$

$$\text{cells} - 2^n \Rightarrow 2^2 = 4$$

$\bar{A}\bar{B}$	0	1
$\bar{A}B$	$\bar{A}B$	AB
$A\bar{B}$	AB	AB

0 is represented with A, B, \dots
 1 is repn with A, B, \dots
 $A = 1$
 $\bar{A} = 0$

Three-variable map:-

$$n=3,$$

$$\text{cells} - 2^n \Rightarrow 2^3 = 8$$

$\bar{A}\bar{B}\bar{C}$	000	011	111	110
$\bar{A}\bar{B}C$	$\bar{A}\bar{B}C$	$\bar{A}BC$	$A\bar{B}C$	$A\bar{B}\bar{C}$
$\bar{A}BC$	ABC	ABC	ABC	ABC

Four-variable map:-

$$n=4,$$

$$\text{cells} = 2^n = 16.$$

$\bar{A}\bar{B}\bar{C}\bar{D}$	0000	0111	1111	1100
$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}\bar{C}D$	$A\bar{B}\bar{C}D$	$A\bar{B}\bar{C}D$
$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}CD$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$
$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$
$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$
$AB\bar{C}D$	$AB\bar{C}D$	$AB\bar{C}D$	$AB\bar{C}D$	$AB\bar{C}D$
$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$
$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$

Plotting of k-map using POS form:-

One variable:-

$$n=1,$$

$$\text{cells}, 2^n = 2,$$

0	$A(0)$
1	$\bar{A}(1)$

$$0 = A, B, \dots$$

$$1 = \bar{A}, \bar{B}, \dots$$

$$A = 0$$

$$\bar{A} = 1$$

Two-variable:-

$$n=2,$$

$$\text{cells}, 2^n = 4.$$

$\bar{A}\bar{B}$	0	1
$\bar{A}B$	$\bar{A}B$	AB
$A\bar{B}$	$A\bar{B}$	$A\bar{B}$

Three-variable:-

$$n=3,$$

$$\text{cells}, 2^n = 8$$

$\bar{A}\bar{B}\bar{C}$	000	011	111	110
$\bar{A}\bar{B}C$	$\bar{A}\bar{B}C$	$\bar{A}BC$	$A\bar{B}C$	$A\bar{B}\bar{C}$
$\bar{A}BC$	ABC	ABC	ABC	ABC

Four-variable:-

$$n=4,$$

$$\text{cells} = 2^n = 16$$

$\bar{A}\bar{B}\bar{C}\bar{D}$	0000	0111	1111	1100
$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}\bar{C}D$	$A\bar{B}\bar{C}D$	$A\bar{B}\bar{C}D$
$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}CD$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$
$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$	$\bar{A}B\bar{C}D$
$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$	$\bar{A}BCD$
$AB\bar{C}D$	$AB\bar{C}D$	$AB\bar{C}D$	$AB\bar{C}D$	$AB\bar{C}D$
$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$
$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$

Sum term representation of K-map -
 A Boolean expression with sum terms
 can be represented on the K-map by placing
 one in each cell corresponding to a min-term.

In the SOP expression,

\Rightarrow The remaining cells are '0'.

$$f = \bar{A}B + A\bar{B}$$

	0	1	
0	0	1	$\bar{A}B$
1	1	0	$A\bar{B}$

3-variable

$$\star f = \bar{A}\bar{B}C + A\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} + ABC$$

$$\begin{array}{ccccc} 001 & 101 & 010 & 110 & 111 \\ \hline 1 & 5 & 2 & 6 & 7 \end{array}$$

$$f = \sum m(1, 2, 5, 6, 7)$$

A	00	01	11	10
0	0	1	0	1
1	0	1	1	1

4 variable:

$$\star f = \bar{A}B\bar{C}\bar{D} + A\bar{B}C\bar{D} + A\bar{B}CD + ABC\bar{D}$$

$$\begin{array}{cccc} 0100 & 1010 & 1011 & 1101 \\ \hline 4 & 10 & 11 & 13 \end{array}$$

$$f = \sum m(4, 10, 11, 13)$$

AB	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	0	1	1	0
10	0	0	1	1

~~Product term representation on K-map!~~

A boolean expression with product terms (POS) can be plotted on the K-map by placing '0' corresponding to the max. term. in the expression and remaining cells are filled with '1'.

ex: $f(A+B)(\bar{A}+B)(\bar{A}+\bar{B}) \rightarrow 2\text{-variable}$

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

* 3-variable

$$f(A + \bar{B} + C) (A + \bar{B} + \bar{C}) (\bar{A} + \bar{B} + C) (A + B + \bar{C})$$

$$\begin{array}{r} 0 \ 1 \ 0 \\ \underline{-\ 2} \\ 3 \end{array} \quad \begin{array}{r} 0 \ 1 \ 1 \\ \underline{\quad 3} \\ 6 \end{array} \quad \begin{array}{r} 1 \ 1 \ 0 \\ \underline{\quad 6} \\ 1 \end{array}$$

$\pi M(2,3,6,1)$

	1	0	0	0
	4	5	7	6
	1	1	1	0

* 4-variable!

$$f(A, B, C, D) = (A + \bar{B} + C + D) \cdot (\bar{A} + \bar{B} + \bar{C} + \bar{D}) \cdot (\bar{A} + \bar{B} + C + D)$$

$$\frac{0 \ 1 \ 0 \ 0}{4} \quad \frac{0 \ 1 \ 1 \ 1}{7} \quad (A+B+\bar{C}+\bar{D})$$

$$\pi M(4,7,12,3)$$

	6	1	0	3	1	2
-	4	5	7			6
0	1	0	-	1		
13	13	15			14	
0	1	1	1	1		
8	6	11	10			
1	1	1	1	1		

Minimization of Boolean expression by using K-map:-

⇒ To minimize the boolean expression using K-map we have to know about the grouping techniques.

⇒ Grouping is the process of combining the terms in adjacent cells (the binary numbers with only one bit difference).

Grouping to adjacent one's (pair):-

$$\text{Ex:- } Y = \bar{A}Bc + \bar{A}Bc$$

$\frac{00}{1}$	$\frac{01}{3}$	pair		
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0

$\Sigma m(1,3)$

if like first it be neglected

$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ \bar{A} & & C \end{pmatrix}$

$$\& \quad Y = \bar{A}Bc + A\bar{B}c$$

$$\frac{011}{B} + \frac{111}{\bar{B}}$$

$$\Sigma m(3,7)$$

$\frac{\bar{A}Bc}{00}$	$\frac{01}{1}$	$\frac{11}{3}$	$\frac{10}{2}$
0	0	1	0
0	1	1	0

(C+D)(A+B)

(A+B+C)(A+B+D)

(A+B+C)(A+B+D)

BC

$$\& \quad Y = A\bar{B}\bar{C} + A\bar{B}C$$

$$\frac{100}{4} + \frac{110}{6}$$

$$\Sigma m(4,6)$$

$\frac{A\bar{B}\bar{C}}{00}$	$\frac{01}{1}$	$\frac{11}{3}$	$\frac{10}{2}$
0	0	0	0
1	1	0	0

if it get closed.

pair

$$Y = \sum m(2, 3, 4, 5)$$

2 - 010
 $\bar{A}BC$

3 - 011
 $\bar{A}BC$

4 - 100
 $A\bar{B}\bar{C}$

5 - 101
 $A\bar{B}C$

* $Y = \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D$

$$\begin{array}{r} 0001 \\ + 1001 \\ \hline 1 \end{array}$$

$\sum m(1, 9)$

$$\begin{array}{r} 0/001 \\ 1001 \\ \hline \end{array}$$

$\oplus \bar{B}\bar{C}D$

Grouping of 4 adjacent 1's (quadratic)

Ex: $Y(A, B, C, D) = \sum m(3, 7, 11, 15)$

	00	01	11	10
3 - 0011 - $\bar{A}\bar{B}CD$	0	0	1	0
7 - 0111 - $\bar{A}BCD$	0	1	1	0
11 - 1011 - $A\bar{B}CD$	0	0	1	0
15 - 1111 - $ABCD$	0	0	1	0

$\oplus CD$

* $Y(A, B, C, D) = \sum m(5, 7, 13, 15)$

5 - 0101

7 - 0111

13 - 1101

15 - 1111

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

$\oplus AB$

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

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

$\oplus CD$

$\oplus BD$

$$Y(A, B, C, D) = \sum m(8, 10, 12, 14) \quad AD$$

$$Y(A, B, C, D) = \sum (0, 2, 8, 10)$$

$$Y = \sum (4, 5, 6, 7, 12, 13, 14, 15)$$

$$\sum (8, 10, 12, 14)$$

$$8 - 1000$$

$$10 - 1010$$

$$12 - 1100$$

$$14 - 1110$$

$$A \cdot \overline{D}$$

	0	1	2	3
	4	5	6	7
	11	12	13	14
	8	9	10	11
A	0	1	2	3

Grouping of 8 adjacent 1's (Octect),
 we see the 8 adjacent 1's are in

$$\text{Ex: } Y = \sum (4, 5, 6, 7, 12, 13, 14, 15) \quad (\text{P. 11})$$

$$4 - 0100 - \bar{A}B\bar{C}\bar{D}$$

$$5 - 0101 - \bar{A}B\bar{C}D$$

$$6 - 0110 - \bar{A}BC\bar{D}$$

$$7 - 0111 - \bar{A}BCD$$

$$12 - 1100 - A\bar{B}\bar{C}\bar{D}$$

$$13 - 1101 - A\bar{B}\bar{C}D$$

$$14 - 1110 - ABC\bar{D}$$

$$15 - 1111 - ABCD$$

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

B

$$Y = \sum m(1, 3, 5, 7, 9, 11, 13, 15) \sim D$$

$$Y = \sum m(0, 1, 2, 3, 8, 9, 10, 11) \sim B$$

$$Y = \sum m(0, 2, 4, 6, 8, 10, 12, 14) \sim D$$

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

Minimize the expression:-

$$Y = \overline{ABC} + \overline{AB}\overline{D} + ABC\overline{D} + A\overline{B}CD + A\overline{B}\overline{C}D + A\overline{B}\overline{C}\overline{D}$$

$$Y = \overline{ABC}\overline{D} + \overline{ABC}D + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}\overline{C}\overline{D}$$

$$Y = \overline{ABC}\overline{D} + \overline{ABC}D + A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}\overline{C}\overline{D}$$

$$\begin{array}{ccccc} 0100 & 0101 & 1100 & 1101 & 1001 \\ 4 & 5 & 12 & 13 & 9 \end{array}$$

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

$$\& \overline{CD} + BC$$

Simplify the expressions-

$$f(x,y,z) = \sum m(0,1,5,7)$$

$\overline{x}\overline{y}$	$\overline{y}\overline{z}$	$\overline{y}z$	$y\overline{z}$	yz
1	1	0	0	
0	1	1	0	

($\overline{y}z$) (xz)

$$f(x,y,z) = xz + \overline{x}\overline{y} + \overline{y}z.$$

$$f(x,y,z) = \sum m(1,2,3,6,7)$$

$$0 - 000$$

$$1 - 001$$

$$5 - 101$$

$$7 - 111$$

$$\begin{array}{l} ① \overline{x}\overline{y}\overline{z} \\ ② \overline{x}\overline{y}z \\ ③ x\overline{y}\overline{z} \\ \hline \overline{y}z \end{array}$$

\overline{x}	$\overline{y}\overline{z}$	$\overline{y}z$	$y\overline{z}$	yz
0	1	1	1	1
1	1	1	1	1

($\overline{x}z'$) (y)

$$f(x,y,z) = y + \overline{x}z$$

$$f(x,y,z) = \pi M(1,3,5)$$

	y_3	y_2	y_1	y_0
x	1	0	0	1
\bar{x}	1	0	1	1

$$f(x,y,z) = \pi M(1,3,5) = (x+z)(y+z)$$

5-variable k-map:-

A 5-variable k-map requires 2^5 cells (32 cells) which is very difficult to identify adjacent cells in single 32 cell ~~in~~ k-map.

⇒ The 5 variable k-map generally consist of 2×4 variable k-map with variables A, B, C, D, E.

⇒ The variable A is different for two maps (as indicated at the top of the diagram).

⇒ Two identical 16 cell map containing B, C, D & E is constructed on '1' & 16 cell map, the value of A is '0'. i.e; \bar{A} & A.

$$\underline{\underline{A=0}}$$

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

$$\underline{\underline{A=1}}$$

	$\bar{D}\bar{E}$	$\bar{D}E$	$D\bar{E}$	DE
$\bar{B}\bar{C}$	16	17	19	18
$\bar{B}C$	20	21	23	22
$B\bar{C}$	28	29	31	30
BC	24	25	27	26

Simplify,

(sop form)

$$f(A, B, C, D, E) = \sum m(0, 2, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16, 18, 20, 21, 22, 23, 24, 25, 26, 27, 29, 31)$$

BC	DE	DE	DE	DE	DE
BC	1	4	5	6	1
BC	1	1	1	1	1
BC	1	1	1	1	1
BC	1	1	1	1	1
CE					
BC					

BC	DE	DE	DE	DE	DE
BC	1	6	7	8	1
BC	1	1	1	1	1
BC	1	1	1	1	1
BC	1	1	1	1	1
BC	1	1	1	1	1
CE					
BC					

$$\Rightarrow BC\bar{D}\bar{E} + B\bar{C}\bar{D}E + BCDE + B\bar{C}D\bar{E}$$

$$\Rightarrow BC(\bar{D}\bar{E} + \bar{D}E + DE + D\bar{E})$$

$$\Rightarrow \underline{BC}$$

~~HAPPY~~

$$\begin{array}{l} \cancel{BC\bar{D}\bar{E}} + \cancel{B\bar{C}\bar{D}E} \\ \cancel{DE(\bar{B}\bar{C} + \bar{B}C)} \\ \Rightarrow \\ \begin{array}{c} \cancel{BC\bar{D}\bar{E}} \\ \cancel{B\bar{C}\bar{D}E} \\ \cancel{\bar{B}\bar{C}DE} \\ \cancel{\bar{B}C\bar{D}\bar{E}} \\ \hline \underline{\bar{B}\bar{E}} \end{array} \end{array}$$

$$f(A, B, C, D, E) = \bar{B}\bar{C} + \bar{B}\bar{E} + CE$$

$$\& f(A, B, C, D, E) = \sum m(0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)$$

$$CE + BE + B\bar{D}\bar{E}$$

BC	DE	DE	DE	DE	DE
BC	1	4	5	6	1
BC	1	1	1	1	1
BC	1	1	1	1	1
BC	1	1	1	1	1
CE					
BC					

BC	DE	DE	DE	DE	DE
BC	1	6	7	8	1
BC	1	1	1	1	1
BC	1	1	1	1	1
BC	1	1	1	1	1
CE					
BC					

18

$$\begin{array}{r}
 BC\bar{D}\bar{E} \\
 \bar{B}C\bar{D}\bar{E} \\
 \bar{B}\bar{C}D\bar{E} \\
 \bar{B}CDE \\
 \hline
 \bar{B}E
 \end{array}$$

$$\begin{array}{r}
 BC\bar{D}\bar{E} \\
 \bar{B}C\bar{D}E \\
 \hline
 \bar{B}\bar{D}E
 \end{array}$$

$$\begin{array}{r}
 \bar{B}C\bar{D}\bar{E} \\
 \bar{B}C\bar{D}E \\
 \bar{B}C\bar{D}\bar{E} \\
 \bar{B}CDE \\
 \hline
 CCE
 \end{array}$$

$$f(A, B, C, D, E) = CE + \bar{B}\bar{E} + B\bar{D}E$$

Don't care conditions :-

In some applications the function is not specified for certain combinations of variable functions that have unspecified outputs. For some input combinations which are called incompletely specified functions.

⇒ So, we don't care about the unspecified minterms. For this reason unspecified minterms of a function are called don't care conditions.

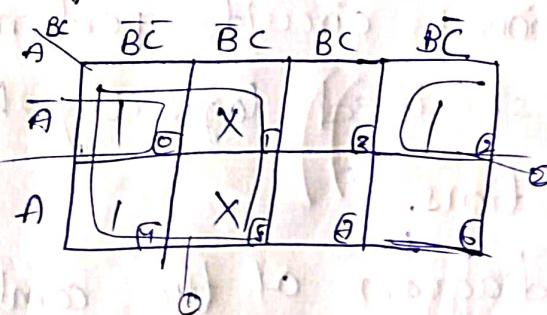
⇒ In SOP, the value of 'x' is '1'.

⇒ In POS, the value of 'x' is '0'.

Simplify the Boolean function $f(a, b, c) = \sum_m(0, 2, 4)$

which has don't care conditions $D(A, B, C) = \sum_m(1, 5)$

by using K-map.



$$f(A, B, C) = \sum_m(0, 2, 4) + D_m(1, 5) = \bar{B} + \bar{A}C$$

$$f(A, B, C, D) = \sum_m (1, 3, 7, 11, 15) + d(0, 2, 4)$$

	\bar{CD}	$\bar{C}D$	$C\bar{D}$	CD	$\bar{C}\bar{D}$
\bar{AB}	1	1	1	1	X
$\bar{A}B$	X	1	1	1	1
$A\bar{B}$	1	1	1	1	1
$A\bar{B}$	1	1	1	1	1

$$\Rightarrow CD + \bar{AB}$$

$$f(A, B, C, D) = \sum_m (0, 6, 8, 13, 14) + d(1, 2, 4, 10)$$

	\bar{CD}	$\bar{C}D$	$C\bar{D}$	CD	$\bar{C}\bar{D}$
\bar{AB}	1	X	1	1	X
$\bar{A}B$	X	1	1	1	1
$A\bar{B}$	1	1	1	1	1
$A\bar{B}$	1	1	1	1	X

$$\Rightarrow ABC\bar{D} + C\bar{D} + \bar{B}\bar{D}$$

logic

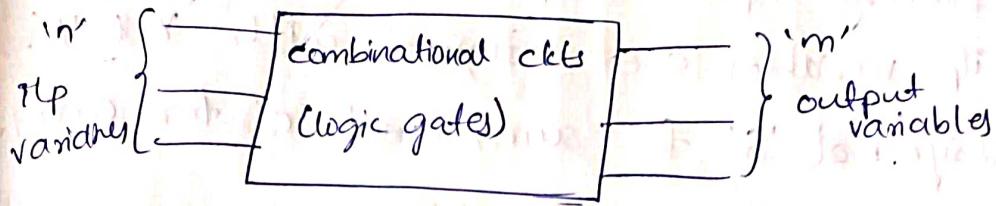
Combinational circuits:-

Combination cbs consists of logic gates whose o/p's at any time are determined from only the present combinations of inputs.

⇒ A combinational circuit performs an operation that it is specified logically by a set of boolean functions.

⇒ The block diagram of the combinational circuit with n inputs binary variables \$ 'm' for o/p variables is shown below.

Block diagram:-



Design Procedure:-

- ⇒ from the specification of the circuit determine the no. of ips & ols, that defines assign a symbol to each.
- ⇒ Design the truth table that defines the required relation b/w ips & ols.
- ⇒ Obtain the simplified boolean function for each ol as a function of ip variables.
- ⇒ Draw the logic diagram & verify the correctness of the design.

Binary adder, subtractor:-

- ⇒ Arithmetic operation for the addition of two binary digits consist of four possible combinations.

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=10$$

- ⇒ A combination ckt that performs the addition of two bits is called Half adder.

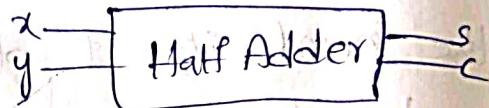
- ⇒ Combinational ckt that performs 3 bit addition (two significant bits and '1' previous carry) is called a Full adder.

Half adder:-

- ⇒ Half adder is a logic circuit which needs two binary inputs & produces two binary outputs.
- ⇒ The input variables are assigned with 'x' & 'y'.
- ⇒ The symbol 'S' for sum, and 'C' for carry.

Truth table of Half adder:-

X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Block diagram

Deriving the boolean expression for above truth table by using k-map (sum)

$$S = \overline{x}y + x\bar{y}$$

$$S = x \oplus y$$

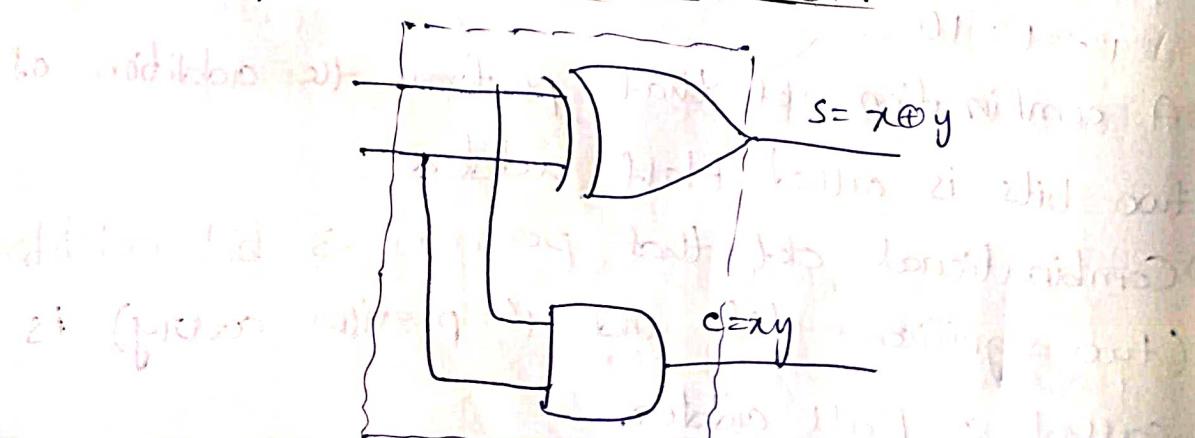
for carry :-

\overline{x}	\overline{y}	y
\overline{x}	0	0
x	0	1

$$C = xy$$

\overline{x}	\overline{y}	y
\overline{x}	0	1
x	1	0

Logic diagram for Half adder.



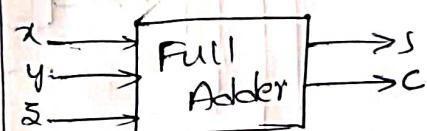
Full Adder:-

As we know that the full adder is a logic ckt which needs 3 binary inputs & 2 binary outputs.

⇒ The input variables are assigned with x, y, z symbols. The symbol 's' for sum & 'c' for carry.

Truth table:-

	x	y	z	s	c
0 - 0	0	0	0	0	0
1 - 0	0	1	0	1	0
2 - 0	1	0	0	1	0
3 - 0	1	1	0	0	1
4 - 1	0	0	1	0	0
5 - 1	0	1	0	0	1
6 - 1	1	0	0	0	1
7 - 1	1	1	1	1	1



Block diagram.

Deriving the boolean function from above truth table:-

K-map for sum:-

	$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}yz$	$xy\bar{z}$	xyz
\bar{x}	0	1	0	1	0
x	1	0	1	0	1

$X \oplus Y$

$$\bar{A}B + A\bar{B}$$

$A \oplus B$

$$\Rightarrow \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + x\bar{y}\bar{z} + xy\bar{z}$$

$$\Rightarrow \bar{x}(\bar{y}\bar{z} + y\bar{z}) + x(\bar{y}\bar{z} + y\bar{z})$$

$$\Rightarrow \bar{x}(y \oplus z) + x(y \oplus z)$$

$$\Rightarrow (\bar{x} + x)(y \oplus z) \Rightarrow y \oplus z$$

$$\begin{array}{l} X-NOR \\ \bar{A}B + A\bar{B} \end{array}$$

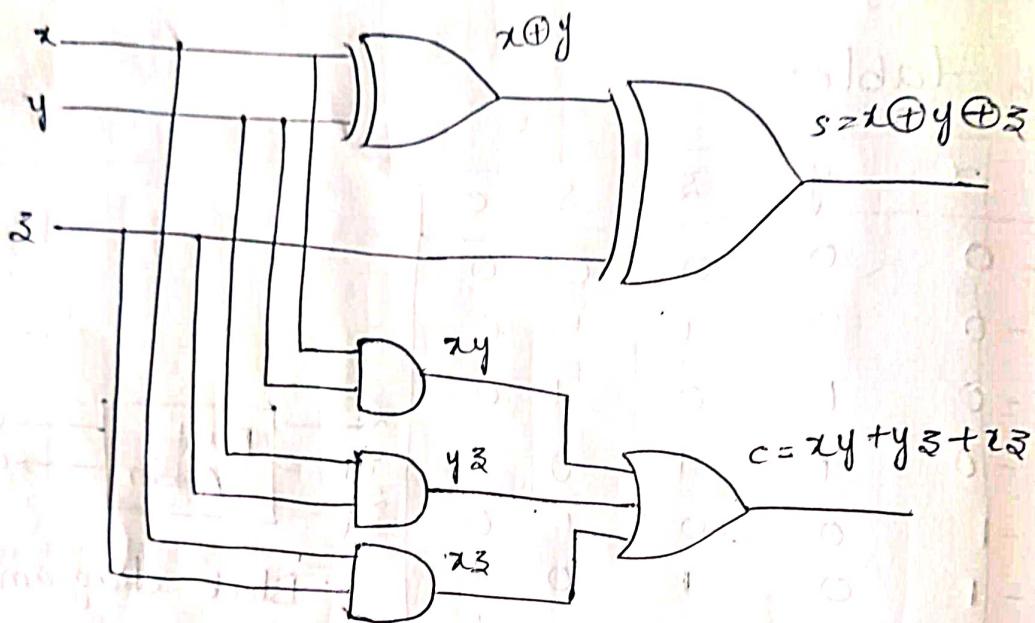
$$\Rightarrow A \oplus B$$

K-map for carry:-

	y_3	\bar{y}_3	y_2	\bar{y}_2	y_1	\bar{y}_1	y_0	\bar{y}_0
x	P	P	1	1	P	P	P	P
y	1	1	1	1	1	1	1	1
z	1	1	1	1	1	1	1	1
c	1	1	1	1	1	1	1	1

$$\Rightarrow c = xy + x\bar{z} + \bar{y}\bar{z}$$

Logic diagram:-



Half subtractor:-

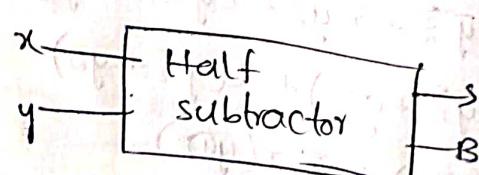
⇒ A combinational circuit that performs 2 bit subtraction (two significant bits and one previous carry) is called Binary ^{Half} subtraction.

⇒ Half subtracter is a logic ckt which needs 2 binary inputs & 2 binary outputs.

⇒ The input variables are assigned with x & y, the symbol 'S' for sum and 'B' for borrow.

Truth table:-

x	y	s	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



K-map for sum:-

$\bar{x}\bar{y}$	$\bar{x}y$	$x\bar{y}$	xy
\bar{x}	0	1	
x	1	0	1

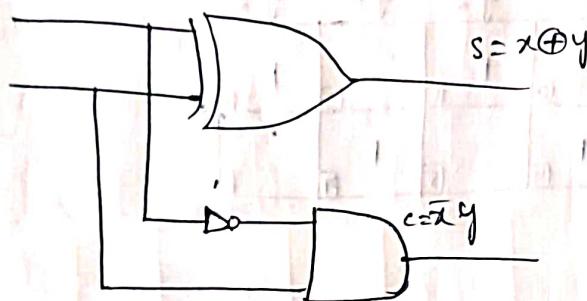
$$S = x \oplus y$$

K-map for borrow

$\bar{x}\bar{y}$	$\bar{x}y$	$x\bar{y}$	xy
\bar{x}	0	1	
x	0	0	0

$$B = \bar{x}y$$

Logic diagram for half subtractor:-



Full subtractor:-

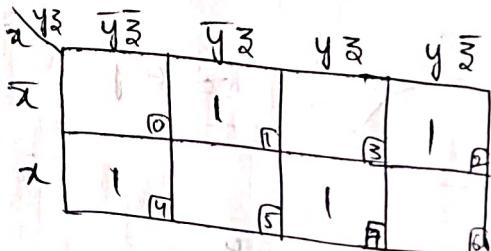
- ⇒ A combinational circuit that performs 3 bit subtraction with 2 significant bits & '1' previous carry is called Binary subtraction.
- ⇒ Full subtractor is a logic circuits which needs 3 binary inputs and 2 binary outputs.
- ⇒ The input variables are assigned with $x, y, \& z$, 'S' for sum & 'B' for borrow.

Block diagram:-

truth table:-

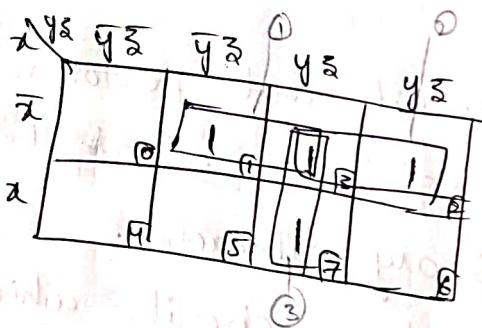
x	y	z	s	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-map for sum:-



$$\Rightarrow s = x \oplus y \oplus z$$

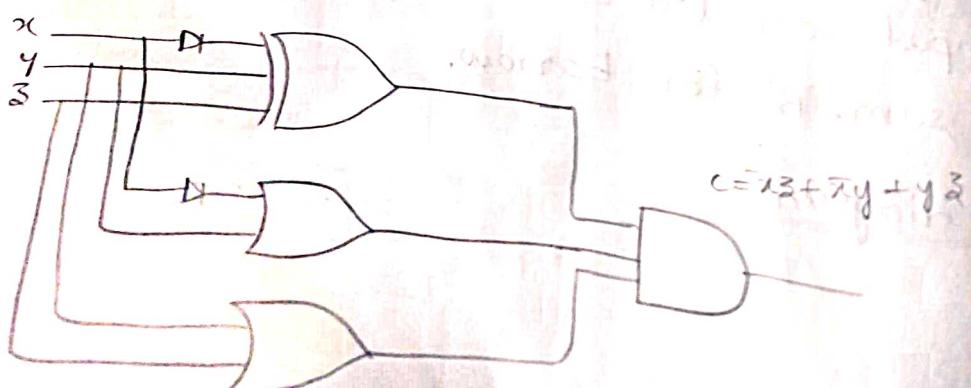
K-map for borrow:-



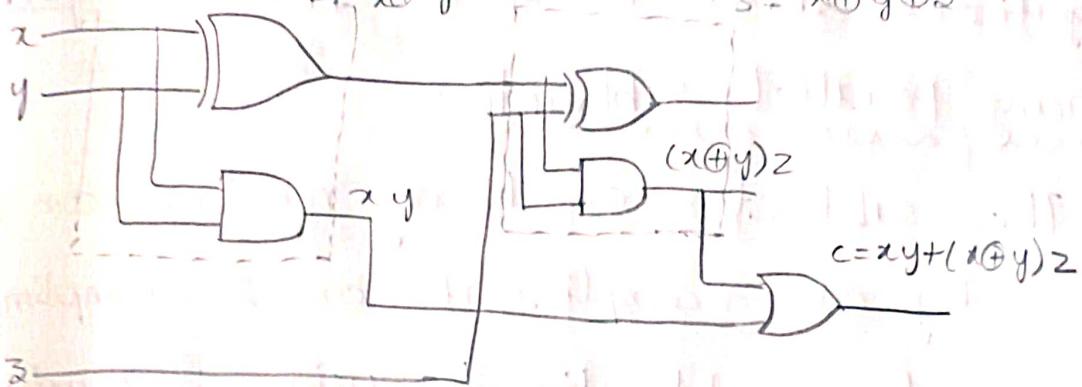
$$\begin{array}{l}
 \overline{x}y'z \\
 \overline{x}y'z \\
 \hline
 \overline{x}z \\
 \overline{x}y'z \\
 \hline
 y'z
 \end{array}$$

$$\Rightarrow b = \overline{x}z + \overline{x}y + yz$$

Logic diagram:-



Full Adder using 2 half adders and OR Gate:-



$$= xy(1+z) + yz + x\bar{y}z$$

$$= xy + yz + x\bar{y}z$$

$$= xy + yz(x+\bar{x}) + x\bar{y}z$$

$$= xy + xyz + \bar{x}yz + x\bar{y}z$$

$$= xy(1+z) + z(\bar{x}y + \bar{y}x)$$

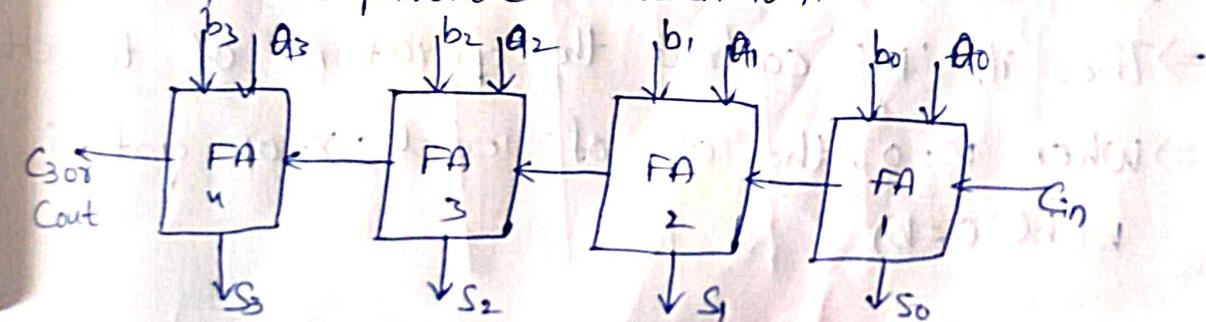
$$= xy(1+z) + z(x \oplus y)$$

Binary parallel adder / Carry Ripple adder:-

The arithmetic sum of 2 binary numbers in parallel. It consists of full adders connected in a chain. with the old carry from each full adder is connected to the new carry of the next full adder in the chain.

⇒ The interconnection of 4 full adders equals

to a 4 bit parallel addition.



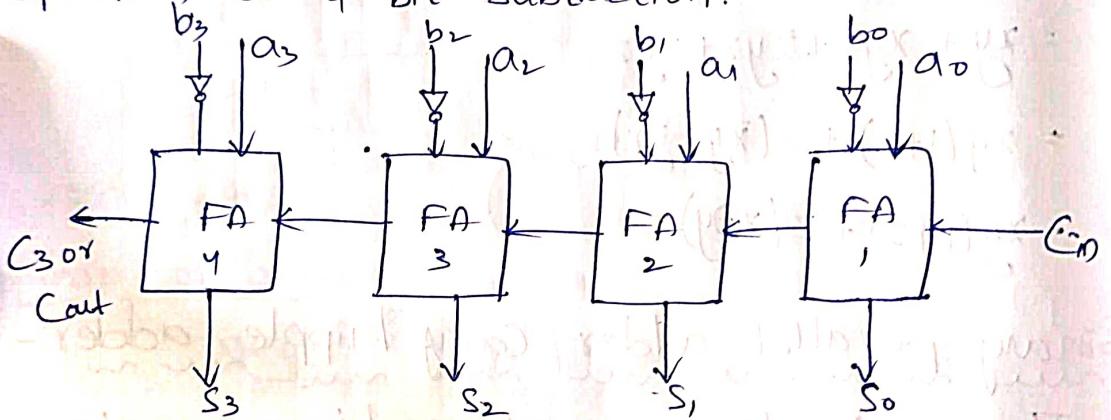
⇒ The adder in which the carry out of each FA is the carry into the next most significant adder, is called sipple carry adder.

Binary parallel subtractors-

The subtraction of binary numbers can be done by one's compliment or 2's compliment.

⇒ To perform subtraction by using 2's compliment method.

⇒ The interconnection of 4 full adders to perform a 4 bit subtraction, is shown below.



Binary Parallel Adder Subtractor

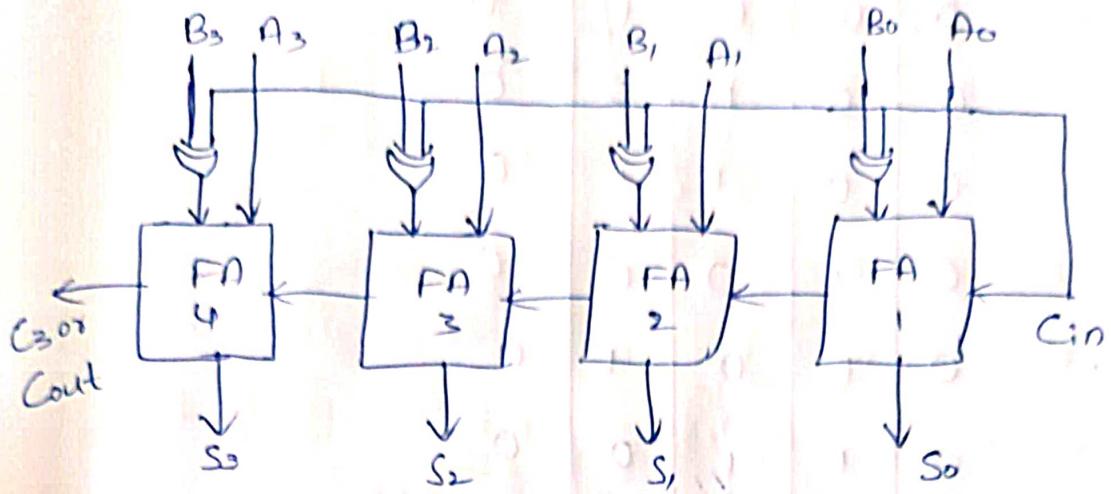
The addition & subtraction of binary numbers can be combined in a single cbt, which is known as binary parallel adder subtractor. This is done by including an X-OR gate with each FA.

⇒ The ip, 'M' controls the operation of the cbt.
When M=0, the o/p of each X-OR gate is $b(b \oplus 0 = b)$.

⇒ If $M=0$, the Cin is '0'. The circuit performs addition ($a+b$).

⇒ If $M=1$, the X-OR gate output is b ($b \oplus 1 = \bar{b}$)

⇒ If $M=1$, Cin is '1' & ckt performs subtraction.



* BCD Adder:-

BCD Adder is a ckt that adds 2 BCD digits & produces a sum digit which is BCD form.

⇒ There are 3 cases in BCD addition.

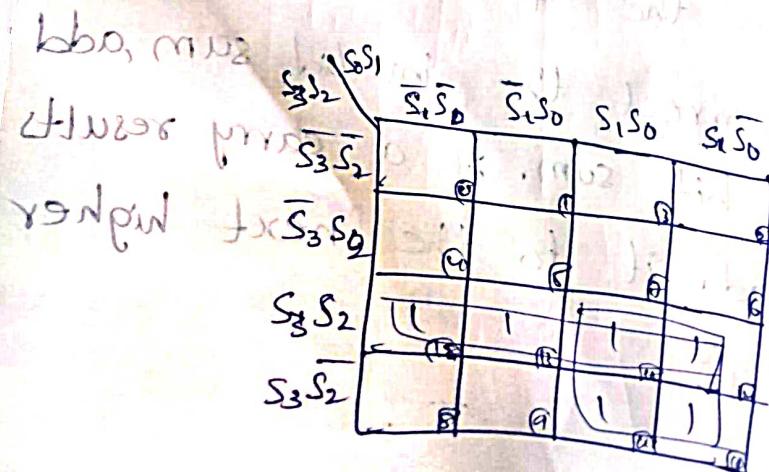
(i) Add 2 BCD numbers using ordinary Binary addition. If the 4 bit sum is ≤ 9 , no correction is needed.

⇒ If the 4 bit sum is > 9 or carry is generated from the 4 bit sum, the sum is invalid. To correct the invalid sum, add 6 to the 4 bit sum. If a carry results addition, then add it to the next higher adder BCD digit.

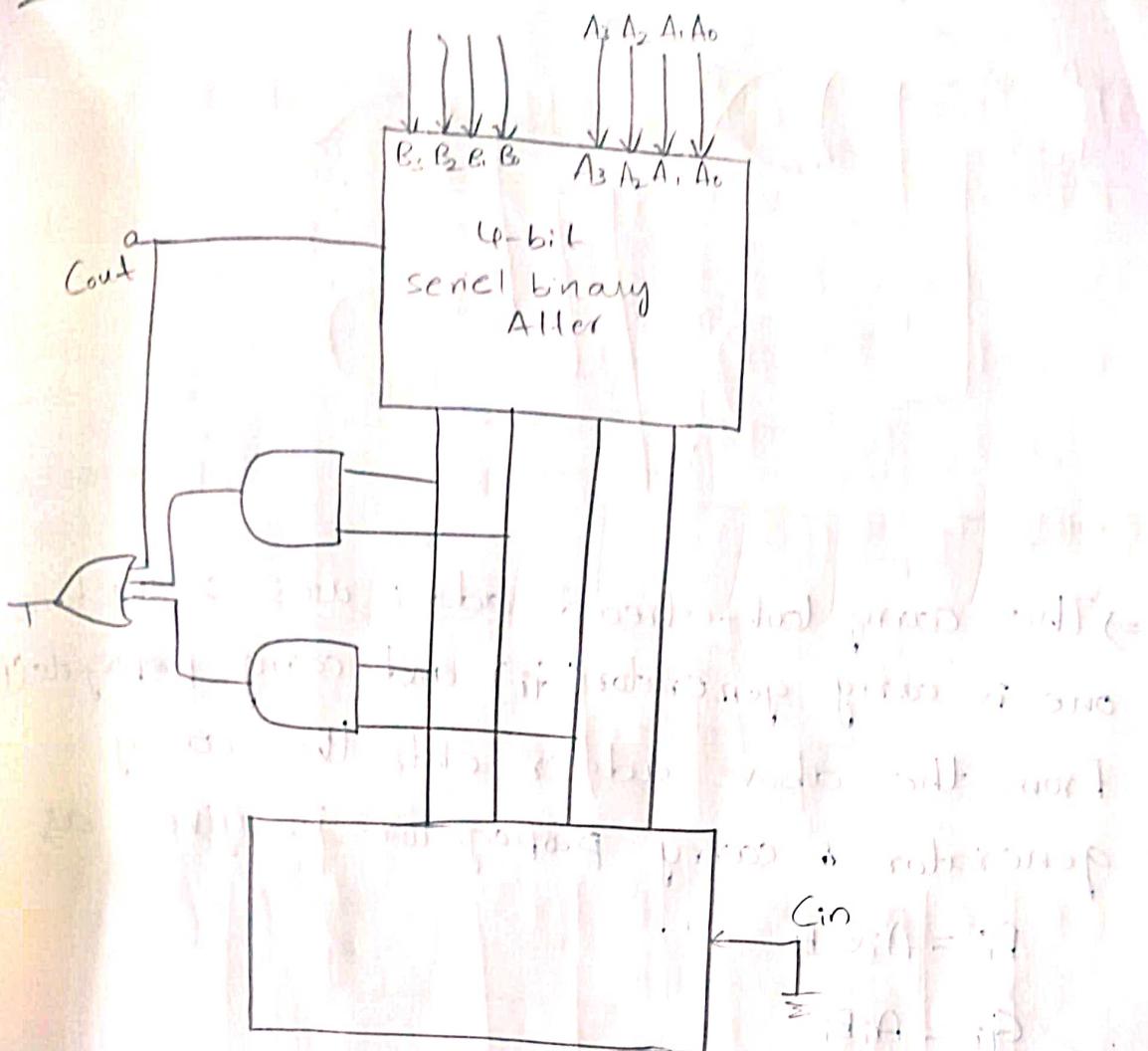
Truth tables

inputs				outputs
S_3	S_2	S_1	S_0	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

K-map-



Logic diagram:-



Carry look-a-head Adder! -

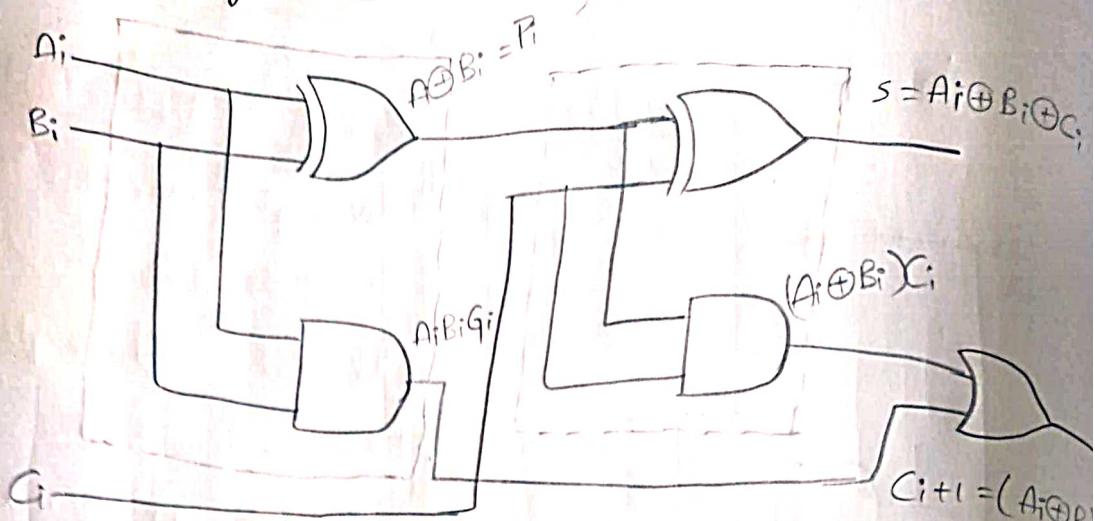
⇒ The IEEE adder is a ripple carry adder in which the carry output of each FA is connected to the carry input of next higher adder stage.

⇒ ∵ The sum and carry outputs of any stage can't be produced until the flip-flop occurs.

⇒ This leads to the time delay in addition process. This delay is known as carry propagation delay.

⇒ One method of reducing the delay by eliminating interspace carry delay is known as carry look-a-head adder.

Logic diagram:-



⇒ The carry look-ahead adder uses 2 functions, one is carry generator (G_i) and carry propagator. From the above adder ckt the carry generator & carry propagator is given as

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = (A_i \oplus B_i) C_i + A_i B_i$$

$$C_{i+1} = P_i C_i + G_i \quad \text{--- (1)}$$

The boolean function of carry out of various full adder as follows

$$i \geq 1,$$

$$C_{i+1} = P_i C_i + G_i$$

$$\text{or } C_2 = P_1 C_1 + G_1$$

C_1 is the initial carry,
 $i \geq 2$,

$$C_3 = P_2 C_2 + G_2$$

$$\text{and } P_2 = [P_1 G_1 + G_2] + G_{12}$$

$$i \geq 3, \text{ now } i = 3$$

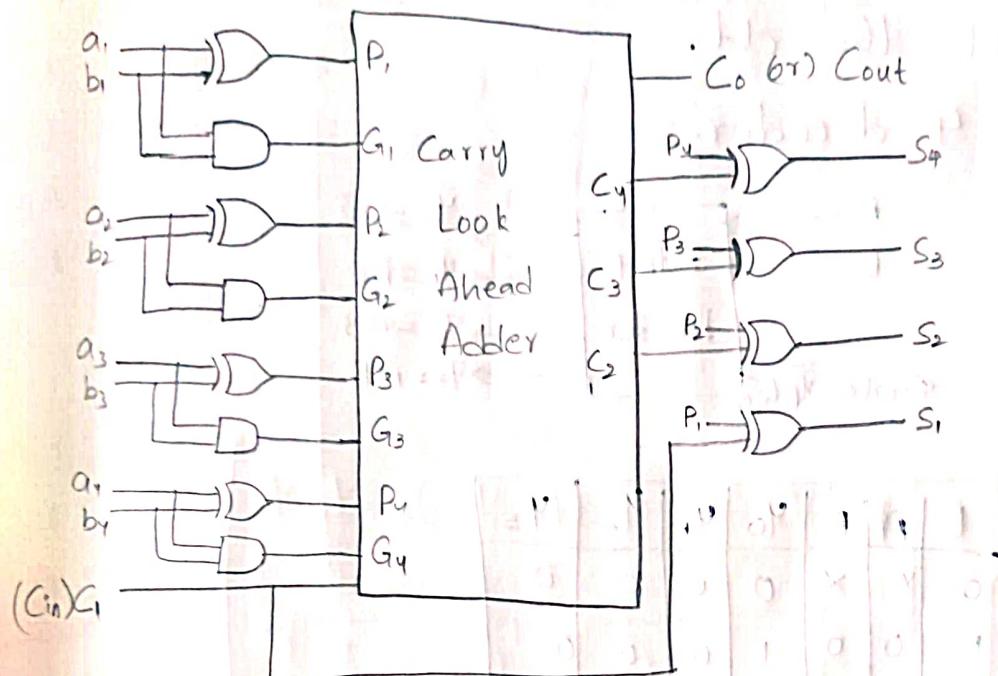
$$C_4 = P_3 C_3 + G_3$$

$$\Rightarrow P_3 \left(P_2 P_1 C_1 + P_2 G_1 + G_2 \right) + G_3$$

$$\Rightarrow P_3 P_2 P_1 C_1 + P_2 P_3 G_1 + P_2 G_2 + G_3.$$

Logic diagram with 4 bit Lee adder
with a carry look-ahead adder

Pin diagram:

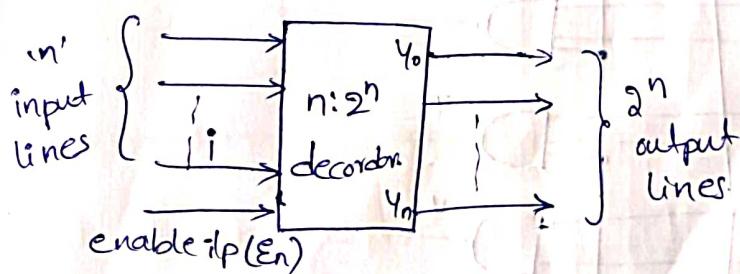


$$S_i = P_i + C_{i-1}$$

De-coder:-

Decoder is a combinational ckt that converts binary information from n input lines to a maximum no. of 2^n unique output lines.

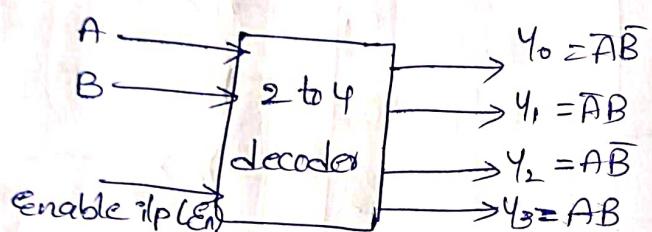
General structure of decoder is



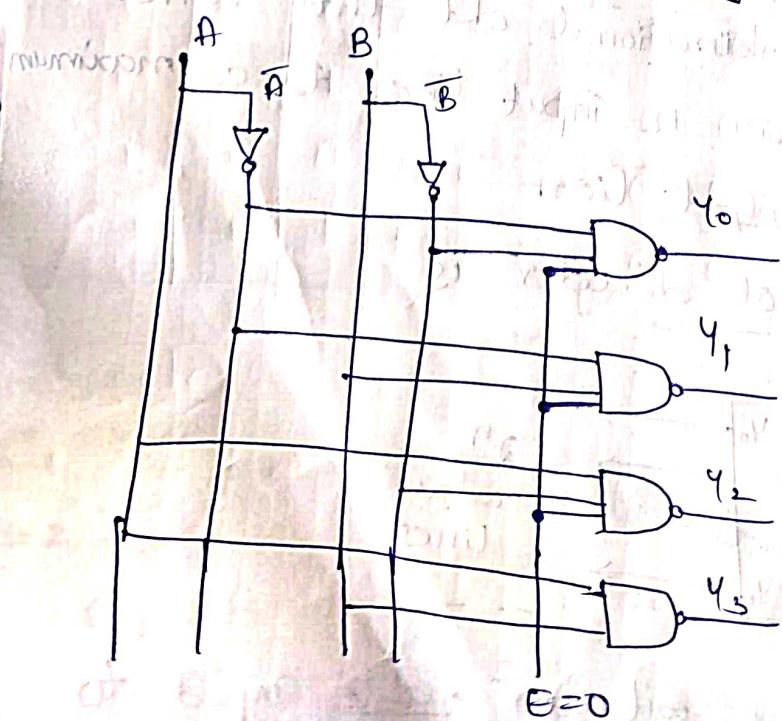
Decoder is provided with an enable input to active the decoder. outputs based on input data.

- ⇒ The enable input may be activated with a single signal or one signal.
- ⇒ The output is disabled when enable 'E=0' when minterm output values are zero.
- ⇒ Some decoders have, ~~two~~, two or more enable inputs that satisfy a given logic condition to enable the output.

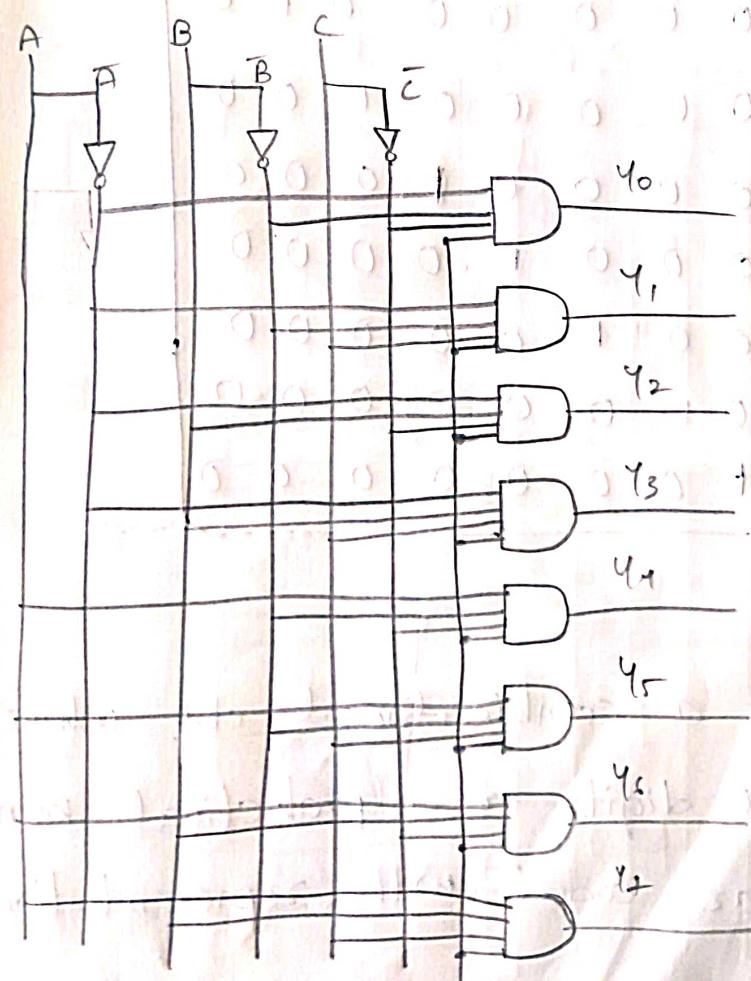
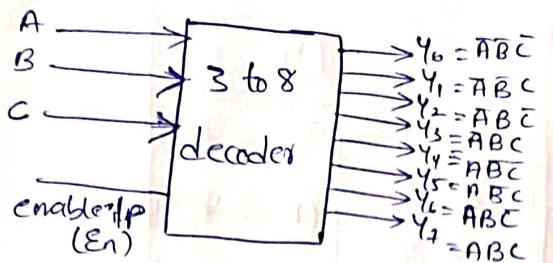
Ex:- 2-4 decoder.



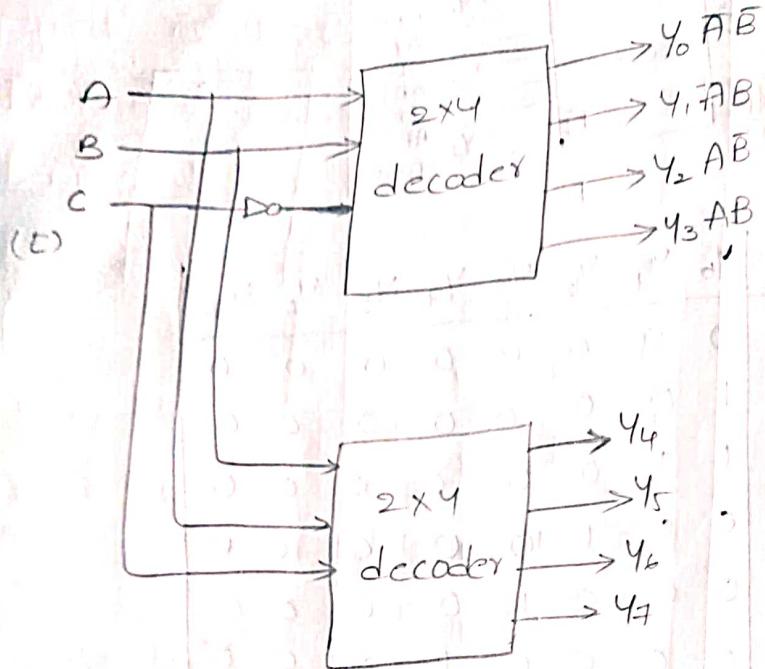
E	A	B	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



3-8 decoder



Design 3-8 decoder using two four decoders

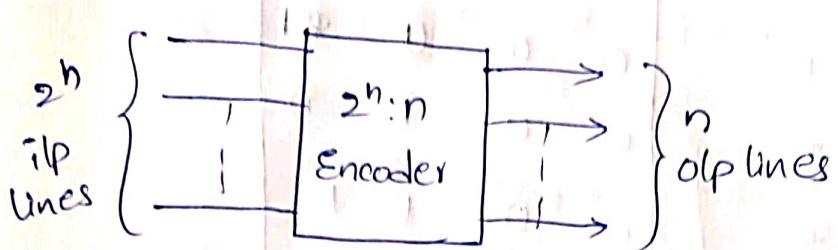


E	B	A	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Encoder:

→ Encoder is a combinational cbt whose inputs are decimal digits or alphabetical numbers and op's are coded with representation of those ip's.

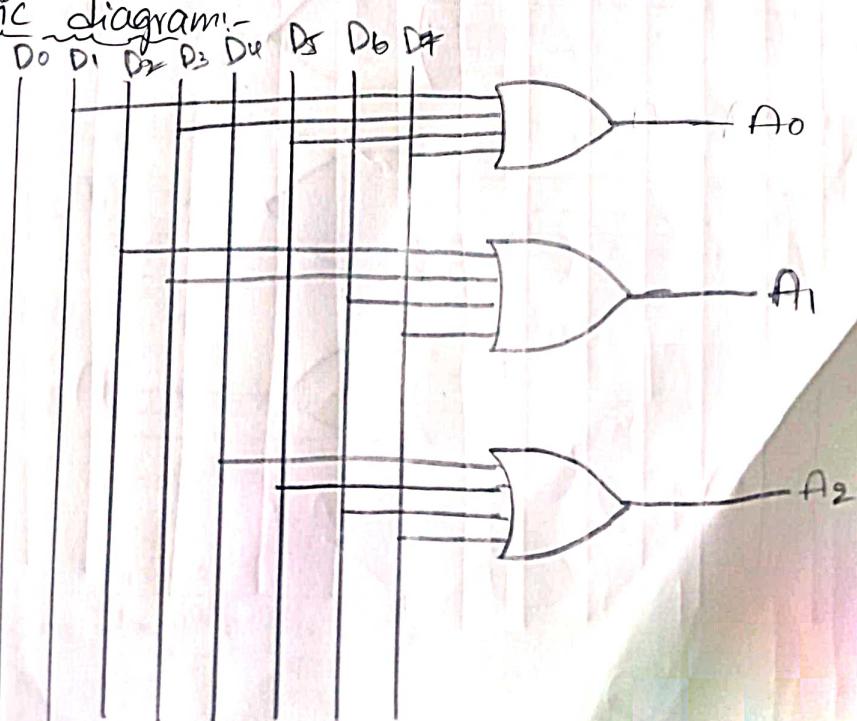
- ⇒ An Encoder is a combinational cbt that have 2^n ilp's. and n o/p lines.
- ⇒ The structure of Encoder is,



Octal - Binary Encoder:-

ilp's Decimal number		A ₂	A ₁	A ₀	
D ₀ — 0		0	0	0	$A_0 = D_1 + D_3 + D_5 + D_7$
D ₁ — 1		0	0	1	$A_1 = D_2 + D_3 + D_6 + D_7$
D ₂ — 2		0	1	0	$A_2 = D_4 + D_5 + D_6 + D_7$
D ₃ — 3		0	1	1	
D ₄ — 4		1	0	0	
D ₅ — 5		1	0	1	
D ₆ — 6		1	1	0	
D ₇ — 7		1	1	1	

Logic diagram:-

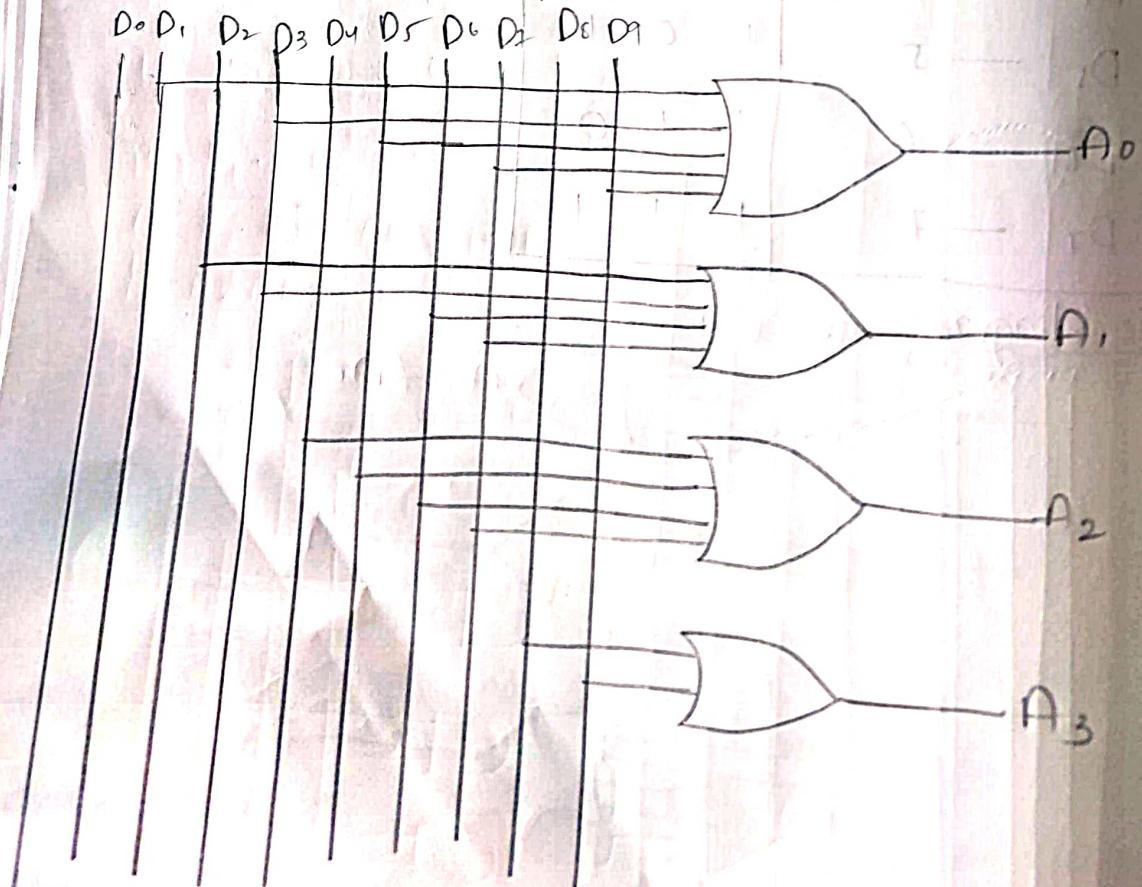


Binary to BCD Encoder

<u>Decimal number</u>	A_3	A_2	A_1	A_0
$D_0 \rightarrow 0$	0	0	0	0
$D_1 \rightarrow 1$	0	0	1	0
$D_2 \rightarrow 2$	0	0	1	1
$D_3 \rightarrow 3$	0	0	0	0
$D_4 \rightarrow 4$	0	1	0	0
$D_5 \rightarrow 5$	0	1	0	0
$D_6 \rightarrow 6$	0	1	1	0
$D_7 \rightarrow 7$	0	1	1	1
$D_8 \rightarrow 8$	1	0	0	0
$D_9 \rightarrow 9$	1	0	0	1

$$A_0 = D_1 + D_3 + D_5 + D_7 + D_9, \quad A_1 = D_2 + D_3 + D_6 + D_7.$$

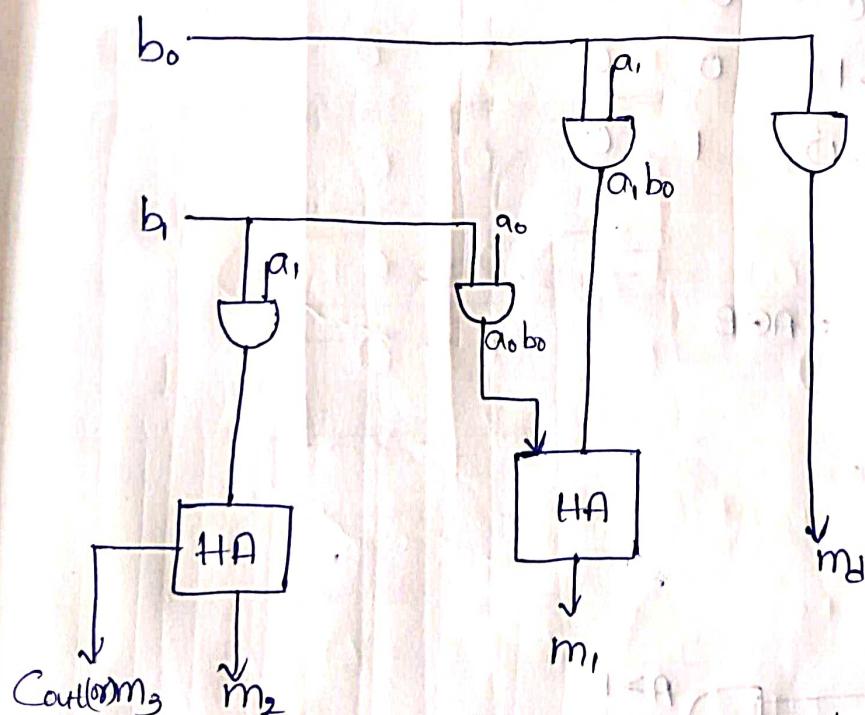
$$A_2 = D_4 + D_5 + D_6 + D_7, \quad A_3 = D_8 + D_9.$$



Binary Multiplier:-

- ⇒ In multiplication binary numbers will perform in the same way the multiplication of decimal numbers.
- ⇒ The multiplicand is multiplied by each bit of the multiplier starting from LSB.
- ⇒ Each such multiplication process gives partial product.
- ⇒ The successive partial products are shifted one position to the left.
- ⇒ The final product is obtained by the addition of all partial products.

2-bit binary multiplier:-



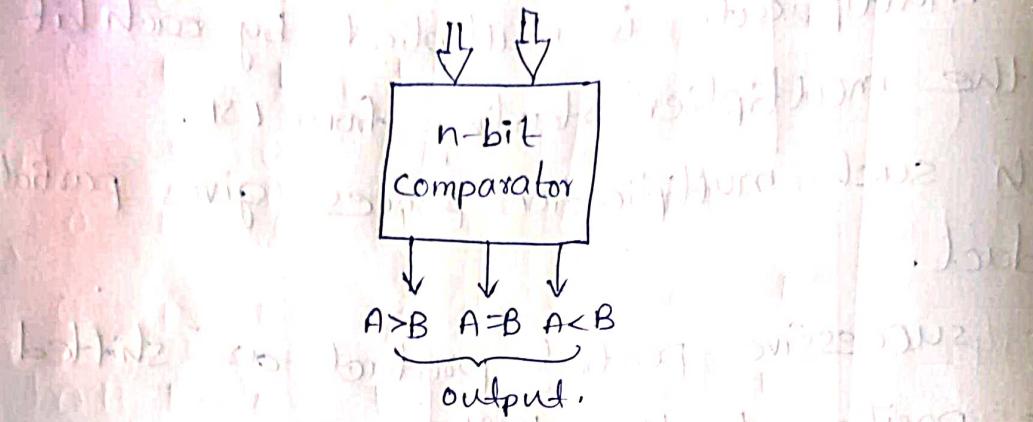
$$\begin{array}{r} a_1, a_0 * b_1, b_0 \\ \hline a_1, a_0 b_1, b_0 \end{array}$$

Why
Pr.
 $\begin{array}{r} a_1, b_1, a_0 b_1, X \\ \hline a_1, b_1, a_1 b_0 + a_0 b_1, a_0 b_0 \\ \hline m_2 \quad m_1 \quad m_0 \end{array}$

* Magnitude comparators -

One bit magnitude comparator:-

⇒ It is a digital circuit which compare two binary numbers with one bit length.



Truth table:-

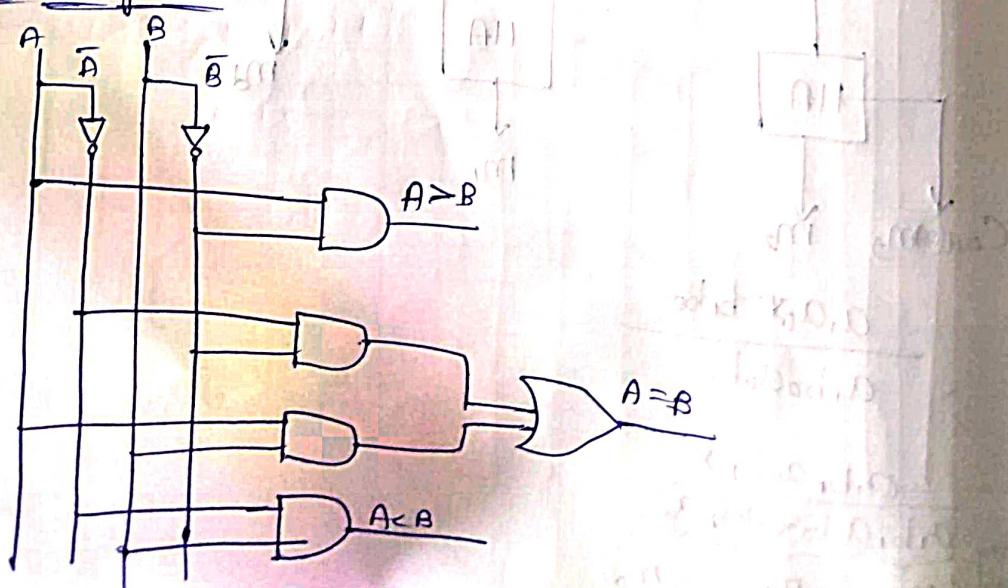
i/p	o/p			
A	B	$A > B$	$A \geq B$	$A < B$
0	0	→ 0	1	0
0	1	→ 0	0	1
1	0	→ 1	0	0
1	1	→ 0	1	0

$$A > B = \overline{AB}$$

$$A = B = \overline{A}\overline{B} + AB = A \oplus B$$

$$A < B = \overline{A}\overline{B}$$

Logic diagram:-



Two bit magnitude comparator:-

Assume two binary numbers A [A₁, A₀] and B [B₁, B₀] respectively, then the ~~cond~~ conditions for outputs are,

Case(i):- A will be Greater than B [A > B] if any of the two conditions satisfied.

Condition:-

1:- when A₁=1 and B₁=0 irrespective of other bits

2:- when A₁=B₁ and A₀=1 and B₀=0

$$A_1 = B_1 \text{ } \& \text{ } A_0 = 1 \text{ } \& \text{ } B_0 = 0$$

Case(ii):- when A [A < B].

when any one of the following conditions satisfied.

a) A=0 and B=1 irrespective of other bits.

b) A₀=0, B₀=1, or A₁=B₁,

Case(iii):-

A=1, when A₁=B₁ and A₀=B₀

functional table:-

inps		A > B	outs	
			A = B	A < B
A ₁ >B ₁	x	→ 1	0	0
A<B ₁	x	→ 0	0	1
A ₁ =B ₁ , A ₀ >B ₀		→ 1	0	0
A ₁ =B ₁ , A ₀ <B ₀		→ 0	0	1
A ₁ =B ₁ , A ₀ =B ₀		→ 0	1	0

Truth tables:-

A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0-0	0	0	0	0	1	0
1-0	0	0	1	0	0	1
2-0	0	1	0	0	0	1
3-0	0	1	1	0	0	1
4-0	1	0	0	1	0	1
5-0	1	0	1	0	1	0
6-0	1	1	0	0	0	1
7-0	1	1	1	0	0	1
8-1	0	0	0	1	0	0
9-1	0	0	1	1	0	0
10-1	0	1	0	0	1	0
11-1	0	1	1	0	0	1
12-1	1	0	0	1	0	0
13-1	1	0	1	1	0	0
14-1	1	1	0	1	0	0
15-1	1	1	1	0	0	1

$$A > B = 4, 8, 9, 12, 13, 14$$

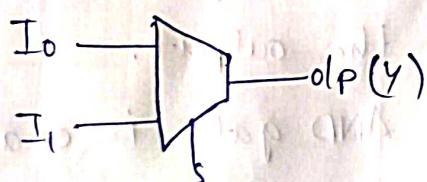
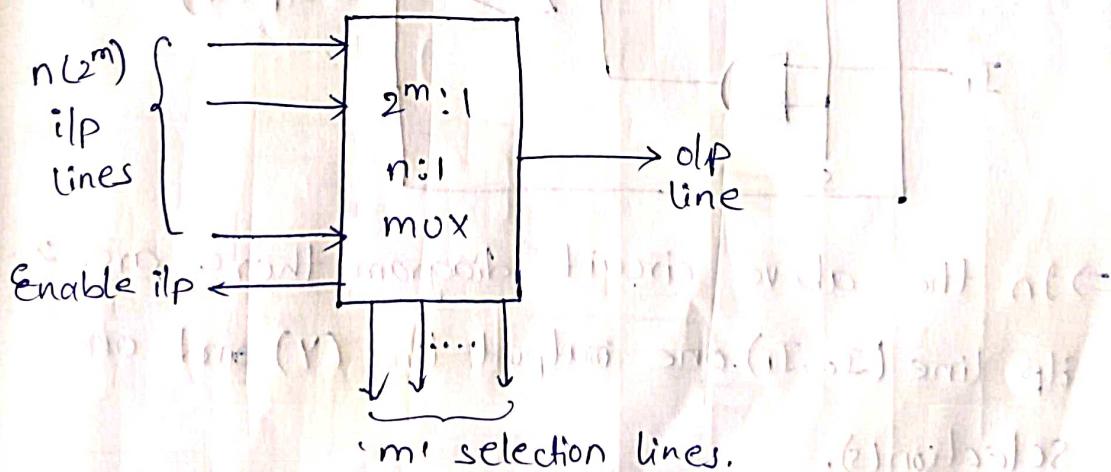
$$A = B = 0, 5, 10, 15$$

$$A < B = 1, 2, 3, 6, 7, 11$$

Multiplexer

- ⇒ Multiplexer is a combination circuit that selects binary information from one of many "ilp" lines and directs it to a single olp line.
- ⇒ The selection of a particular input line is controlled by a set of selection lines.
- ⇒ Normally there are 2^n ilp lines n selection lines whose bit combinations determine which input is selected.
- ⇒ The multiplexer is also known as data selector because it select one input at a time and send it to the output line.
- ⇒ The multiplexer is often labeled "mux" in block diagram.

⇒ The block diagram of an n input multiplexer with m selection lines is shown in the below figure.

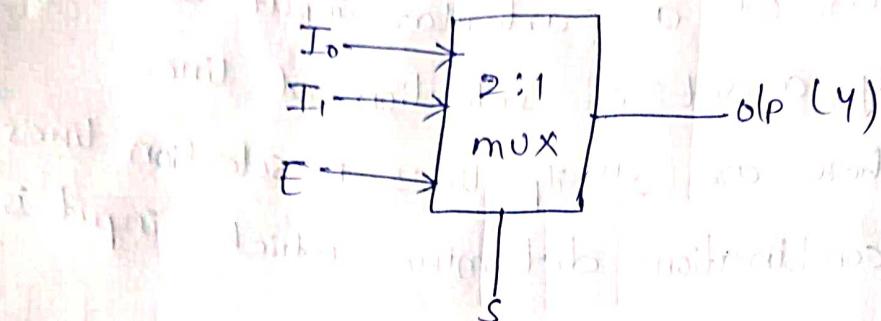


Two-to-one line mux:-

$$m=1$$

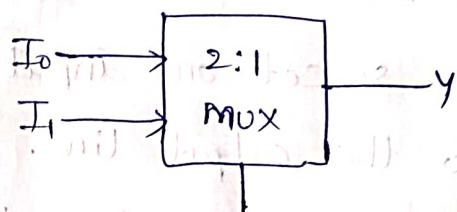
$$n=2^m = 2^1 = 2$$

Block diagram:-

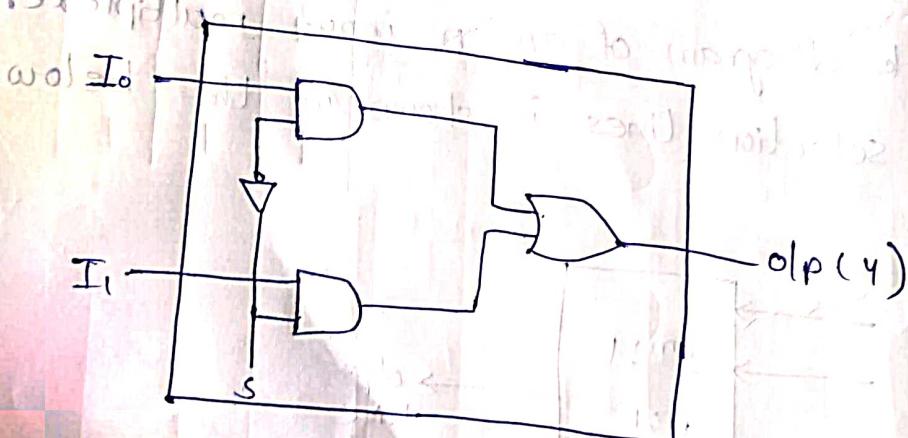


Truth table:-

<u>S</u>	<u>o/p (Y)</u>
0	I_0
1	I_1



Logic diagram:-



⇒ In the above circuit diagram, there are 2d i/p line (I_0, I_1), one output line (Y) and one selection(S).

⇒ When $S=0$, the upper AND gate is enabled I_0 as a patch to the output.

⇒ When $S=1$ the lower AND gate is enabled and as a patch to the o/p.

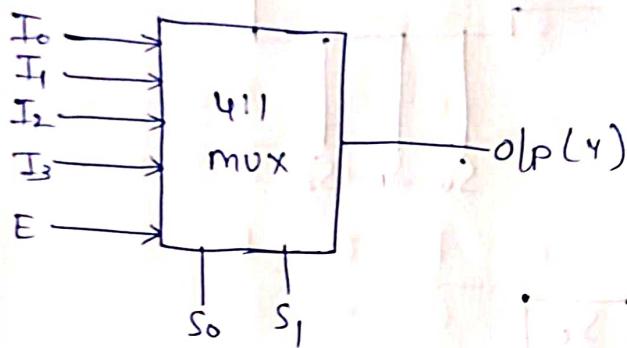
→ The multiplexer acts like an electronic switch that selects one of two sources.

4-1 line mux:-

$$n=4=2^2$$

$$m=2$$

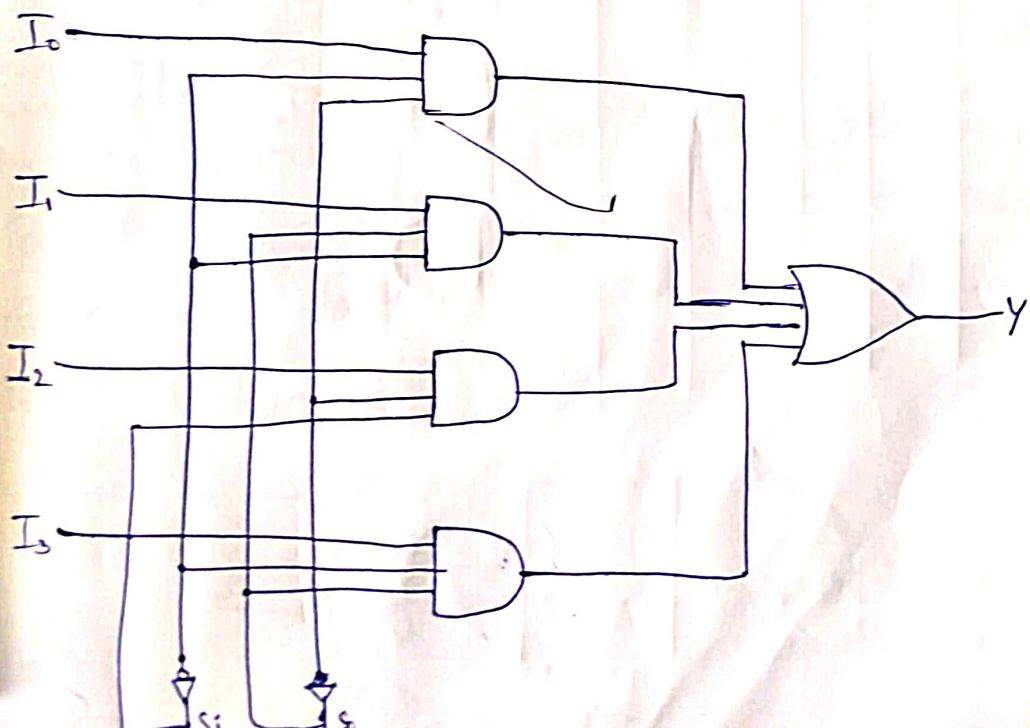
Block diagram:-



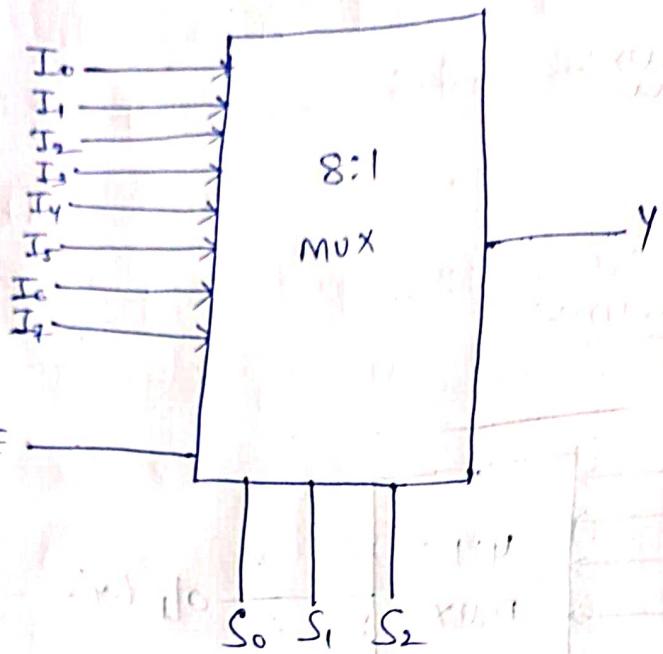
Truth table:-

S_1	S_0	$o/p(4)$
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Logic diagram:-



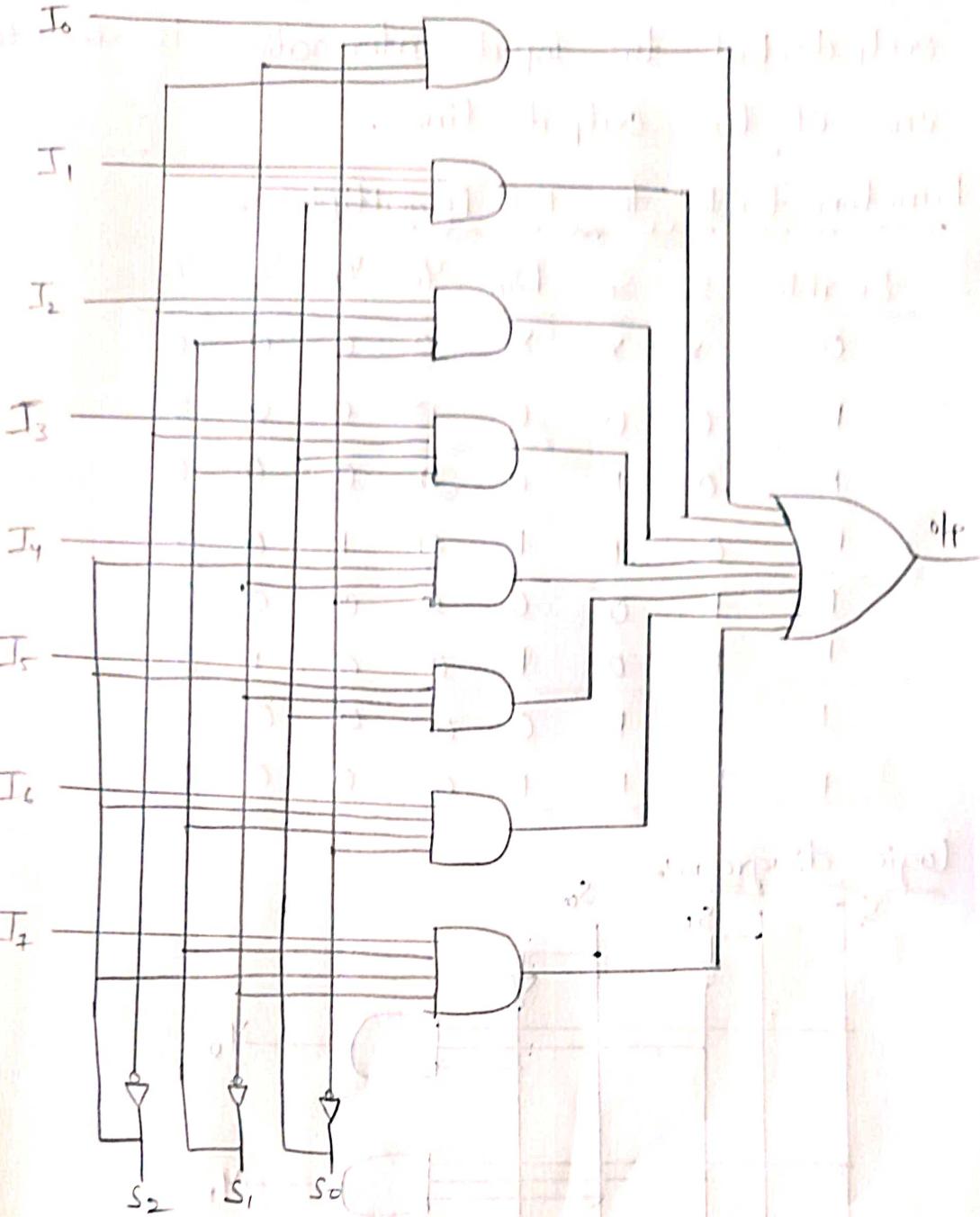
8-to-1 line MUX



Truth table:-

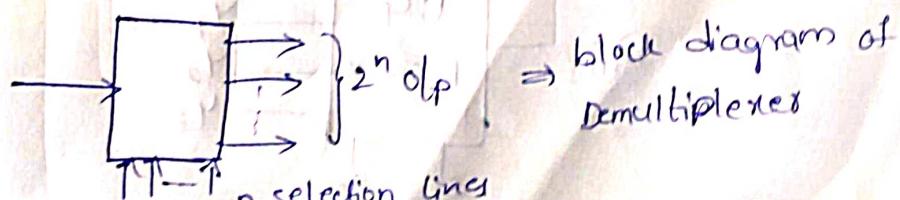
S_2	S_1	S_0	
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Logic diagram



De-multiplexer:-

- ⇒ A Demultiplexer is a circuit that receives information on a single line and transmit this information on one of 2^n possible output lines.
- ⇒ The selection of specific o/p line is controlled by the values of n selection lines.

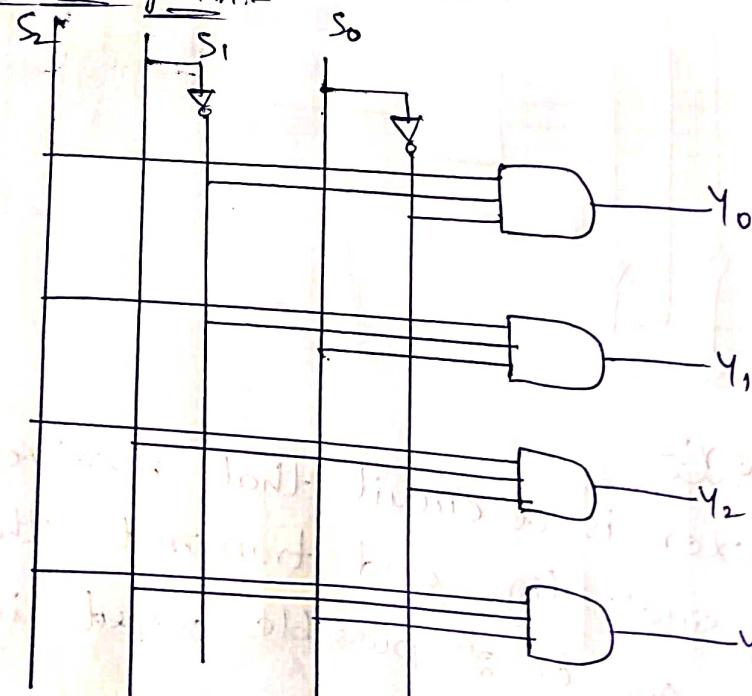


⇒ The below fig. shows 1:4 demultiplexer. The single input variable is D_{in} has a path to all four outputs, but the input information is directed to one of the output lines.

Function table for 1:4 Demultiplexer:-

Enable	S_1	S_0	D_{in}	Y_0	Y_1	Y_2	Y_3
0	x	x	x	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1

Logic diagram:-



Logic symbol:-

