UNIT-II – GATE LEVEL MINIMIZATION

K-Map :-

→ The K map or Karnaugh map is a chart or a graph composed of an arrangement of adjacent cells, each representing a particular combination of variable in SOP form.

→ Any boolean expression can be expressed in standard or canonical SOP or POS form.

→ A standard SOP form is one in which a number of product terms, each one of which contains all variables of the function either in complemented form or normal form.

→ Similarly for POS form also.

Two Variable K-map :-

→ A Two Variable K-map has $2^2 = 4$ squares.

→ These squares are called cells. Each square or cell on the K-map represents the minterm.

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

=

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

=

A \ B	0	1
0	0	1
1	2	3

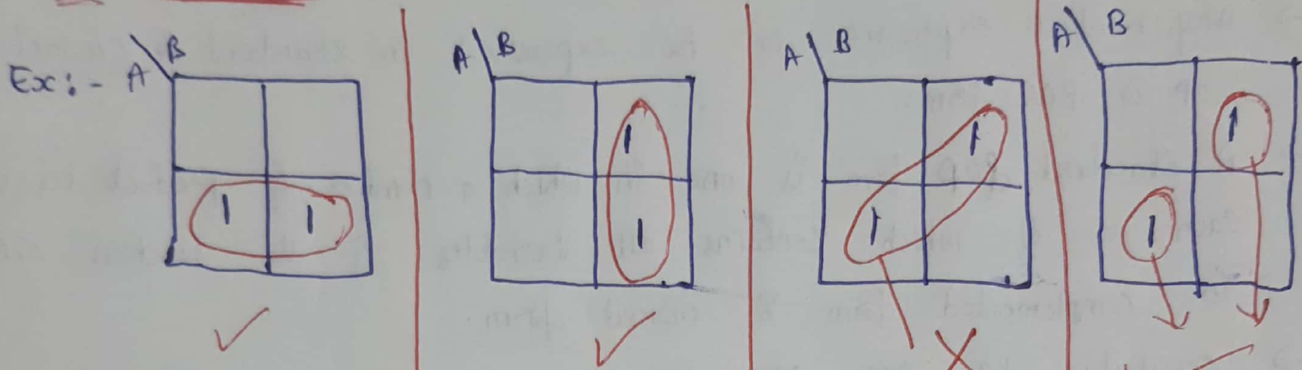
→ To minimize the boolean expression we look for adjacent cells or squares having i.e minterms adjacent to each other & we combine them to form larger squares to eliminate some variables.

→ Two squares are said to be adjacent if they differ in only one variable.

→ A minterm can be combined with any numbers of minterms adjacent to it to form larger squares.

→ Two minterm which are adjacent to each other can be combined to form a larger & bigger square called a 2 square & a pair.

Ex:-

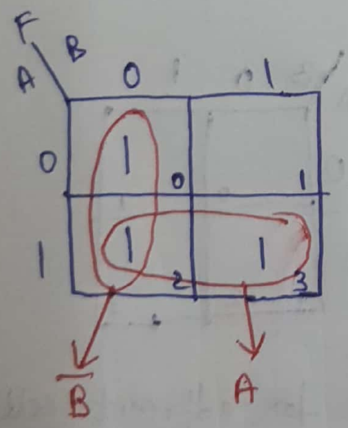


we can group | we can group | Grouping not possible | we can group single

Problems:-

Ⓠ Evaluate the following using 2-Variable K-map:-

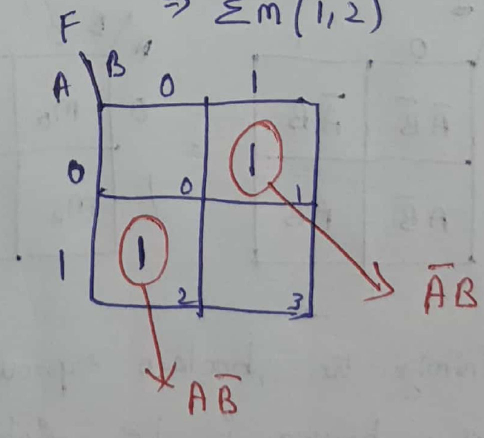
(1) $F = \sum m(0,2,3)$



$$\boxed{F = \bar{B} + A}$$

(2) $F = A\bar{B} + \bar{A}B$

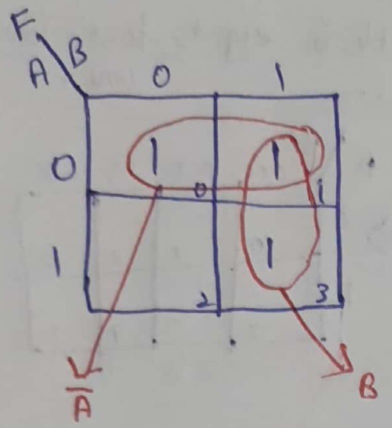
$\Rightarrow \sum m(1,2)$



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

$F = A \oplus B$

(3) $F = \bar{A}\bar{B} + \bar{A}B + AB$



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

$\Rightarrow m_0 + m_1 + m_3$

$\Rightarrow \sum m(0,1,3)$

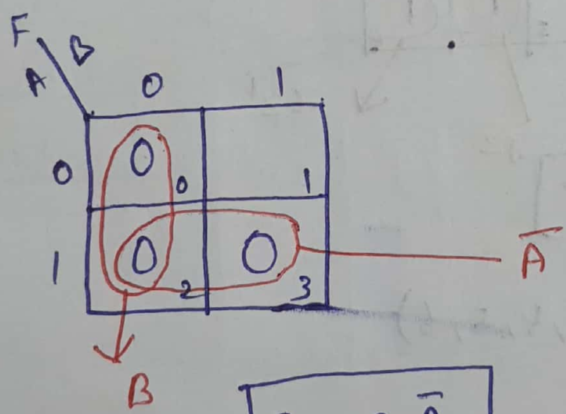
$\therefore \boxed{F = \bar{A} + B.}$

(4) $F = (A+B)(\bar{A}+B)(\bar{A}+\bar{B})$

Sol: The given expression is in POS form. Hence write the maxterm of the expression.

$F \Rightarrow (A+B)(\bar{A}+B)(\bar{A}+\bar{B})$.

$\Rightarrow \prod M(0,2,3)$



$\boxed{F = B.\bar{A}}$

(5) $F = \prod M(0,1,3)$ — H.W

(4)

3- Variable K-map :- It has $2^3 = 8$ Square or cell.
Maximum grouping possible is eight, four, Two, one.

A \ BC	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}BC$	$\bar{A}B\bar{C}$
1	$A\bar{B}\bar{C}$	$A\bar{B}C$	ABC	$AB\bar{C}$

$\Rightarrow$

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

$\Rightarrow$

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

① **Evaluate the following using 3 Variable K-map :-**

① $F(x,y,z) = \Sigma m(3,4,6,7)$

Sol $\Sigma m(3,4,6,7)$

F

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

xz'

yz

$$\boxed{F = x\bar{z} + yz}$$

② $F(x,y,z) = \Sigma m(0,2,4,5,6)$

Sol F

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

$x\bar{y}$

$\bar{z}$

$$\boxed{F = x\bar{y} + \bar{z}}$$

③ $F = A'C + A'B + AB'C + BC$

Sol: Solve the given expression F as it minterms doesn't have all the variables.

$F = A'(B+B')C + A'B(C+C') + AB'C + (A+A')BC$

$\Rightarrow A'(BC+B'C) + A'BC + A'BC' + AB'C + ABC + A'BC$

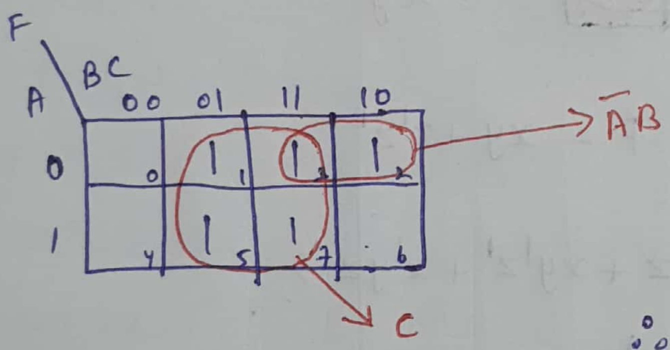
$\Rightarrow \underline{A'BC} + A'B'C + \underline{A'BC} + A'BC' + AB'C + ABC + \underline{A'BC}$

$\Rightarrow$ eliminate the repeated terms

$\Rightarrow A'BC + A'B'C + A'BC' + AB'C + ABC$

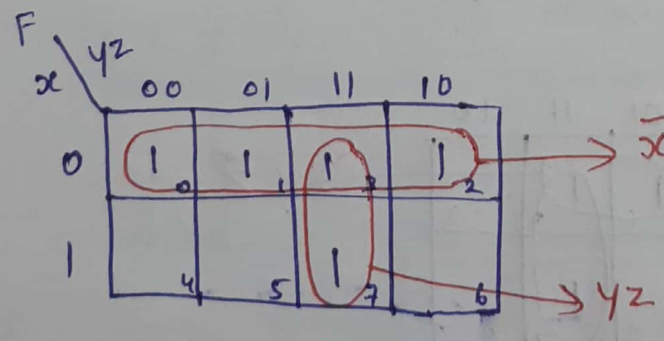
$\Rightarrow m_3 + m_1 + m_2 + m_5 + m_7$

$\Sigma m\,(1, 2, 3, 5, 7)$



$\therefore \boxed{F = \bar{A}B + C}$

④ $F(x, y, z) = \Sigma m\,(0, 1, 2, 3, 7)$



$F = \bar{x} + yz.$

(5) $F(x,y,z) = xy + x'y'z' + x'yz'$

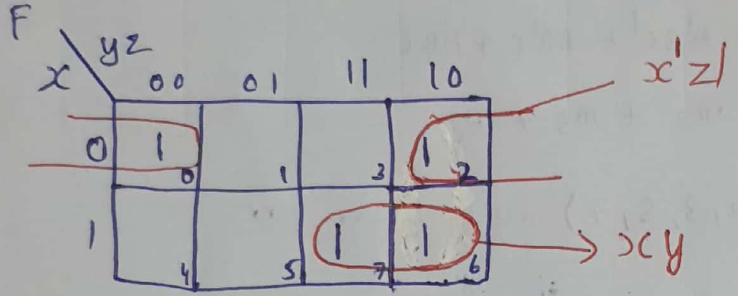
S: Solve for variables in minterms

$\Rightarrow xy(z+z') + x'y'z' + x'yz'$

$\Rightarrow xyz + xyz' + x'y'z' + x'yz'$

$\Rightarrow m_7 + m_6 + m_0 + m_2$

$\Rightarrow \Sigma m(0,2,6,7)$

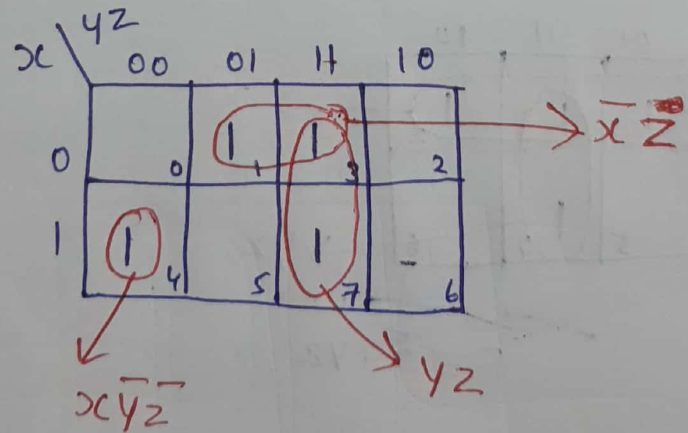


$F = xy + x'z'$

(6) $F(x,y,z) = xyz + x'y'z + xy'z' + x'yz$?

S: $F = \Sigma m(1,3,4,7)$

(1,3,4,7) minterms from above expression.



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

Practice Problems on 3-Variable K-map:-

1. $F(x,y,z) = \sum m(3,4,6,7)$
2. $F(x,y,z) = \sum m(1,2,4,7)$
3. $F(x,y,z) = \sum m(0,2,3,6,7)$
4. $F(x,y,z) = \sum m(0,2,3,4,6)$
5. $F(x,y,z) = \sum m(3,5,6,7)$
6. $F(x,y,z) = \sum m(0,1,2,5,7)$
7. $F(x,y,z) = \sum m(1,2,3,6,7)$
8. $F(x,y,z) = \sum m(0,1,6,7)$
9. $F(x,y,z) = \sum m(0,1,3,4,5)$
10. $F(x,y,z) = \sum m(1,3,5,7)$
11. $F(x,y,z) = \sum m(1,4,5,6,7)$
12. $F(x,y,z) = x'y' + yz + x'yz'$
13. $F(x,y,z) = x'y + yz' + y'z'$.

DEFINITIONS:-

1. **Pair** :- Group of two adjacent minterms. A pair eliminates one Variable in output expression.

2. **Quad** : Group of Four adjacent minterms. A Quad eliminates two Variables in output expression.

3. **Octet** :- Group of eight adjacent minterms. An octet eliminates three Variables in output expression.

4. **Redundant group** :- A redundant group is a group in which all the elements in this group are covered by some other group.

4 - Variable K-map :- It has $2^4 = 16$ Squares & Cells.

wx \ yz	00	01	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

$\Rightarrow$

wx \ yz	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

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

(6) **Evaluate the following using 4 Variable K-map :-**

(1) $F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$

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

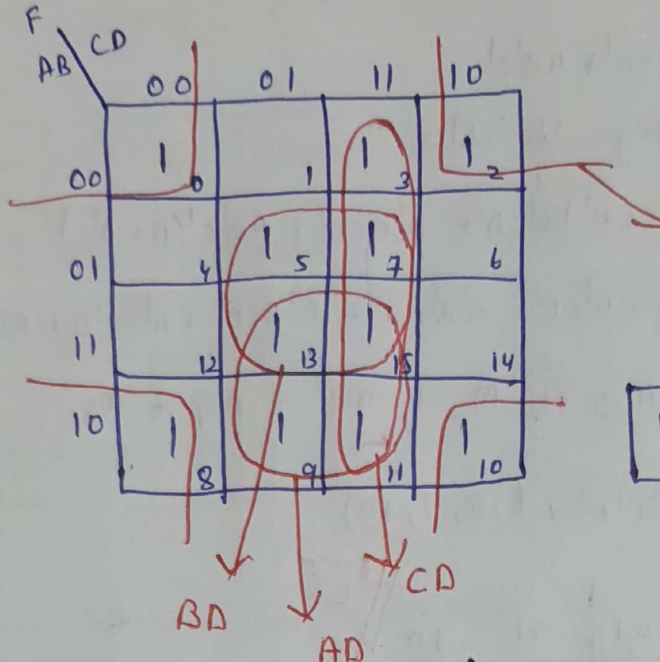
$\rightarrow \bar{w}\bar{z}$

$\rightarrow x\bar{z}$

$\rightarrow \bar{y}$

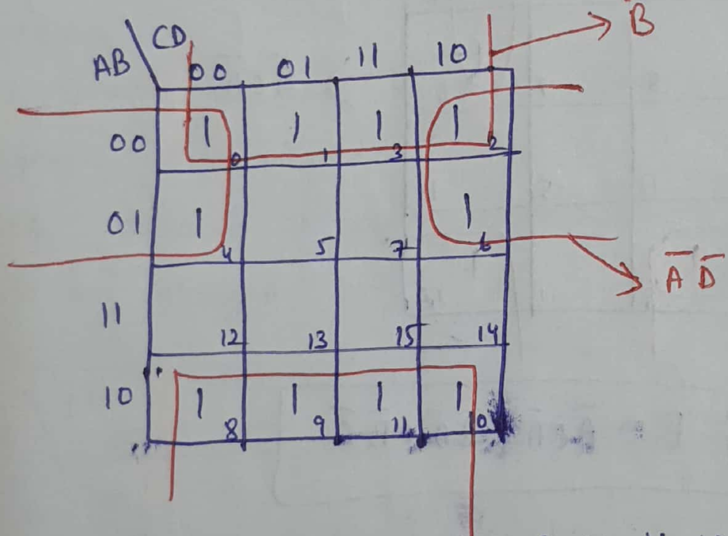
$$\boxed{\therefore F = \bar{y} + \bar{w}\bar{z} + x\bar{z}}$$

(2) $F(A,B,C,D) = \Sigma(0,2,3,5,7,8,9,10,11,13,15)$



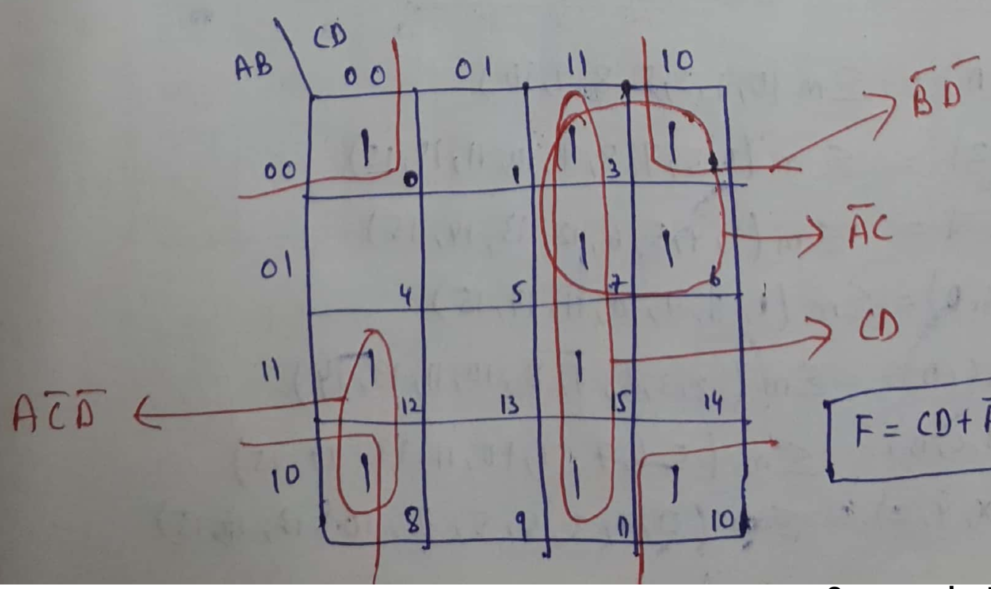
$$\boxed{F = \bar{B}\bar{D} + BD + AD + CD}$$

(3) $F = \Sigma m(0,1,2,3,4,6,8,9,10,11)$



$$\boxed{F = \bar{A}\bar{D} + \bar{B}}$$

(4) $F(A,B,C,D) = \Sigma m(0,2,3,6,7,8,10,11,12,15)$



$$\boxed{F = CD + \bar{A}C + \bar{B}\bar{D} + A\bar{C}\bar{D}}$$

⑤ $F(A,B,C,D) = A'B'C' + B'CD' + A'BCD' + AB'C'$?

Sol: $F = A'B'C' + B'CD' + A'BCD' + AB'C'$

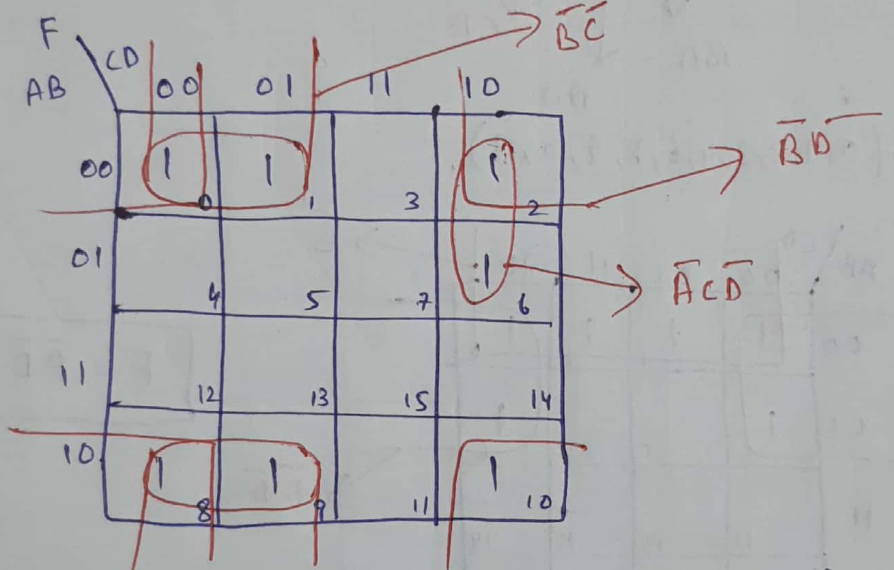
add the missing Variables.

$F = A'B'C'(D+D') + (A+A')B'CD' + A'BCD' + AB'C'(D+D')$

$\Rightarrow A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'$

$\Rightarrow m_1 + m_0 + m_{10} + m_2 + m_6 + m_9 + m_8$

$\Rightarrow \sum m(0,1,2,6,8,9,10)$



$$\boxed{F \Rightarrow \bar{A}C\bar{D} + \bar{B}\bar{D} + \bar{B}\bar{C}}$$

Practice problems on 4-Variable K-map:-

① $F(A,B,C,D) = \sum m(0,1,2,5,8,9,10)$

② $F(W,X,Y,Z) = \sum m(0,5,7,8,9,10,11,14,15)$

③ $F(W,X,Y,Z) = \sum m(1,4,5,6,12,13,14,15)$

④ $F(A,B,C,D) = \sum m(1,5,9,10,11,14,15)$

⑤ $F(A,B,C,D) = \sum m(2,3,6,7,8,10,11,13,14)$

⑥ $F(A,B,C,D) = \sum m(5,6,7,9,10,11,13,14,15)$

⑦ $F(W,X,Y,Z) = \sum m(0,1,3,4,5,7,10,13,14,15)$

5-Variable K-Map :- It consist of $2^5 = 32$ cells.

$A = 0$

BC \ DE	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}$	$\bar{A}\bar{B}\bar{C}\bar{D}E$	$\bar{A}\bar{B}\bar{C}DE$	$\bar{A}\bar{B}\bar{C}D\bar{E}$
01	$\bar{A}\bar{B}C\bar{D}\bar{E}$	$\bar{A}\bar{B}C\bar{D}E$	$\bar{A}\bar{B}CDE$	$\bar{A}\bar{B}CD\bar{E}$
11	$\bar{A}BC\bar{D}\bar{E}$	$\bar{A}BC\bar{D}E$	$\bar{A}BCDE$	$\bar{A}BCD\bar{E}$
10	$\bar{A}B\bar{C}\bar{D}\bar{E}$	$\bar{A}B\bar{C}\bar{D}E$	$\bar{A}B\bar{C}DE$	$\bar{A}B\bar{C}D\bar{E}$

$A = 1$

BC \ DE	00	01	11	10
00	$A\bar{B}\bar{C}\bar{D}\bar{E}$	$A\bar{B}\bar{C}\bar{D}E$	$A\bar{B}\bar{C}DE$	$A\bar{B}\bar{C}D\bar{E}$
01	$A\bar{B}C\bar{D}\bar{E}$	$A\bar{B}C\bar{D}E$	$A\bar{B}CDE$	$A\bar{B}CD\bar{E}$
11	$ABC\bar{D}\bar{E}$	$ABC\bar{D}E$	$ABCDE$	$ABCD\bar{E}$
10	$AB\bar{C}\bar{D}\bar{E}$	$AB\bar{C}\bar{D}E$	$AB\bar{C}DE$	$AB\bar{C}D\bar{E}$

→ It maps for more than 4 Variables are not simple as to use.

→ A 5 Variable k map needs $2^n = 2^5 = 32$ Squares.

→ when number of Variables Increases, number of Squares also increases.

→ It consist of Two - 4 Variable K-map. It has 5 Variables namely A, B, C, D and E.

→ Here Variable A distinguishes b/w two 4 Variable K-maps.

→ It is arranged in 4 Rows & 8 columns.

$A = 0$

BC \ DE	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

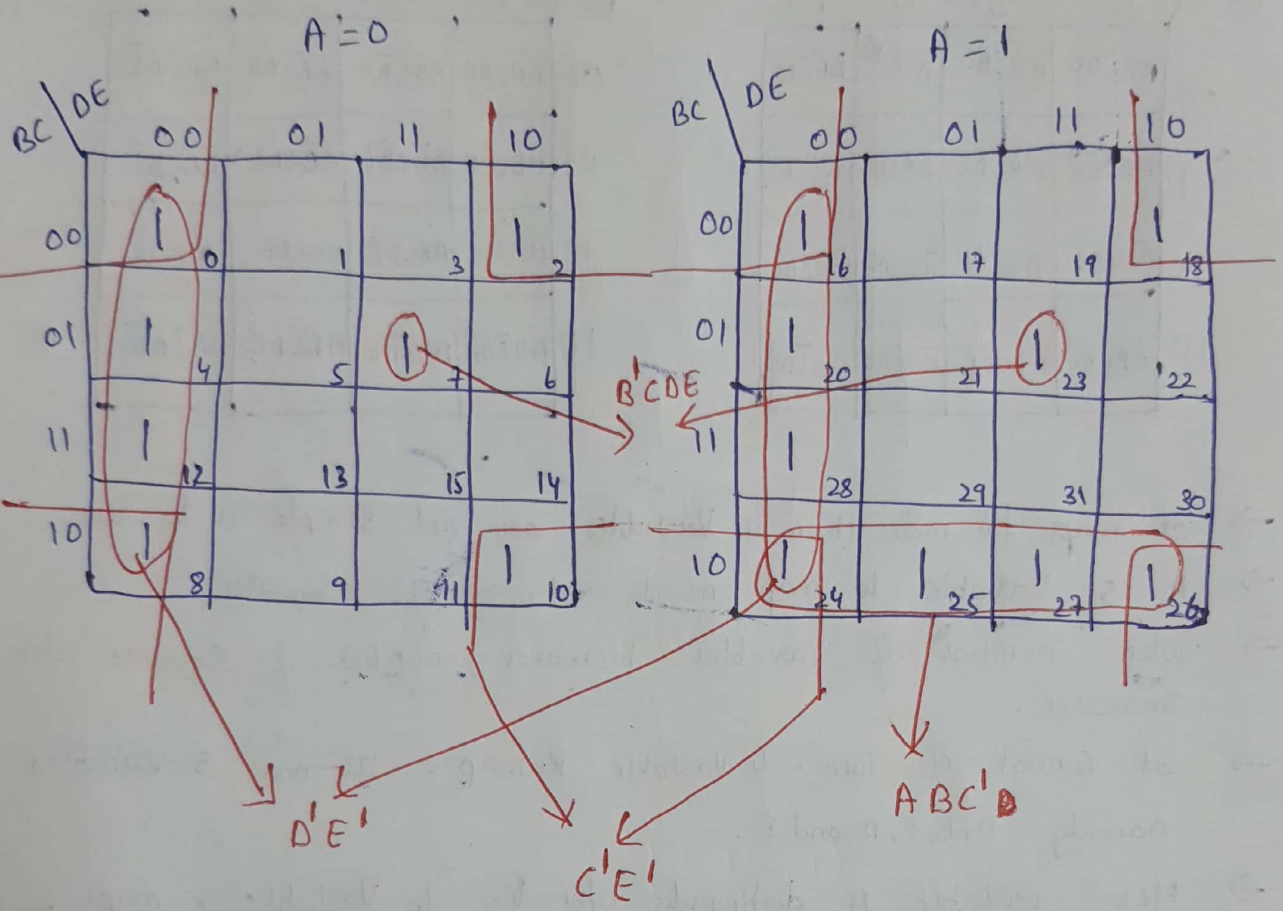
$A = 1$

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

→ minterms 0 to 15 belongs to A=0

→ minterms 16 to 31 belong to A=1.

8 Solve $F(A,B,C,D,E) = \Sigma(0, 2, 4, 7, 8, 10, 12, 16, 18, 20, 23, 24, 25, 26, 27, 28)$.



$$\boxed{F = ABC' + D'E' + C'E' + B'CDE}$$

H.W 1. $F(A,B,C,D,E) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 11, 15, 16, 18, 19, 23, 27, 31)$.

K-Maps using POS form:-

① **3 Variable k-map.**

(a) $F(A,B,C) = (A+B+C)(A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

Sol: write the maxterms of given literals.

$\Rightarrow \prod M(0,1,3,4,6,7)$

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

→ $A+\bar{C}$

→ $\bar{A}+B$

→ $B+C$

$F(A,B,C) = (B+C)(\bar{A}+B)(A+\bar{C})$

(b) $F(A,B,C) = (A+B+\bar{C})(\bar{A}+\bar{B}+C)(A+B+C)(A+\bar{B}+C)(A+\bar{B}+\bar{C})$.

② **4 Variable k-map.**

(a) $F(A,B,C,D) = (A+B+C+D)(A+B+C+\bar{D})(A+B+\bar{C}+D)(A+\bar{B}+C+D)(A+\bar{B}+C+\bar{D})(A+\bar{B}+\bar{C}+\bar{D})(\bar{A}+\bar{B}+\bar{C}+\bar{D})(\bar{A}+B+\bar{C}+D)$

Sol: write the maxterms from the expression

$F = \prod M(0,1,2,4,5,7,15,10)$.

F: AB \ CD	00	01	11	10
00	0	0		0
01	0	0	0	
11			0	
10				0

→ $(B+\bar{C}+D)$

→ $(\bar{B}+\bar{C}+\bar{D})$

← $(A+C)$

$$\boxed{F = (A+C)(B+\bar{C}+D)(\bar{B}+\bar{C}+\bar{D})}$$

Practice :- ① $F(A,B,C,D) = \pi M(0,2,6,7,8,10,12,13)$

② $F(A,B,C,D) = \pi M(2,3,4,5,6,7,12,13)$

Don't Cases in K-Map :-

→ Don't Case Condition Says that we can use the blank Cells of K-map to make a group of variables.

→ To make a group of cells, we can use the 'don't care" cell as either '0' & '1'.

→ We mainly use don't car cell to make a large group of cells.

→ It is represented by symbol 'X'.

⑥ Solve the given expressions :-

① $F(A,B,C) = \Sigma m(2,3,4,5) + d(6,7)$

A \ BC	00	01	11	10
0	$_0$	$_1$	1 $_3$	1 $_2$
1	1 $_4$	1 $_5$	X $_7$	X $_6$

→ B

→ A

$\boxed{F = A + B}$

② $F(A,B,C) = \Sigma m(0,2,6,7) + d(1,3)$.

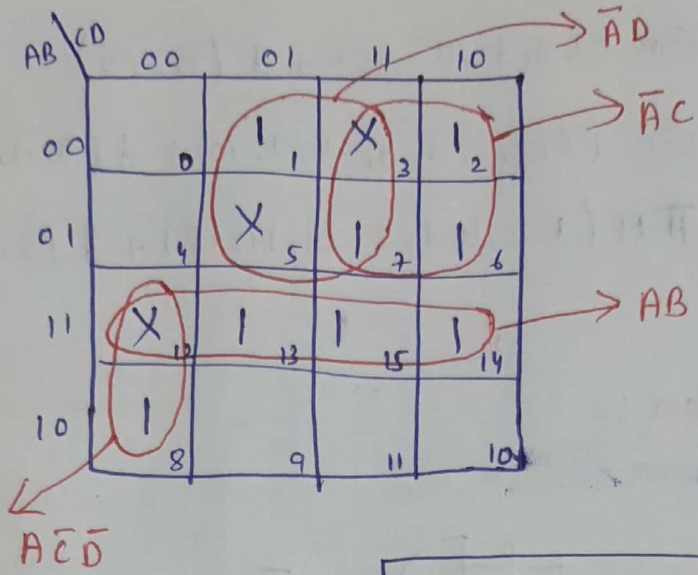
A \ BC	00	01	11	10
0	1 $_0$	X $_1$	X $_3$	1 $_2$
1	$_4$	$_5$	1 $_7$	1 $_6$

→ $\bar{A}$

→ B

$\boxed{F = \bar{A} + B}$

Q $F(A,B,C,D) = \Sigma m(1,2,6,7,8,13,14,15) + d(3,5,12)$.

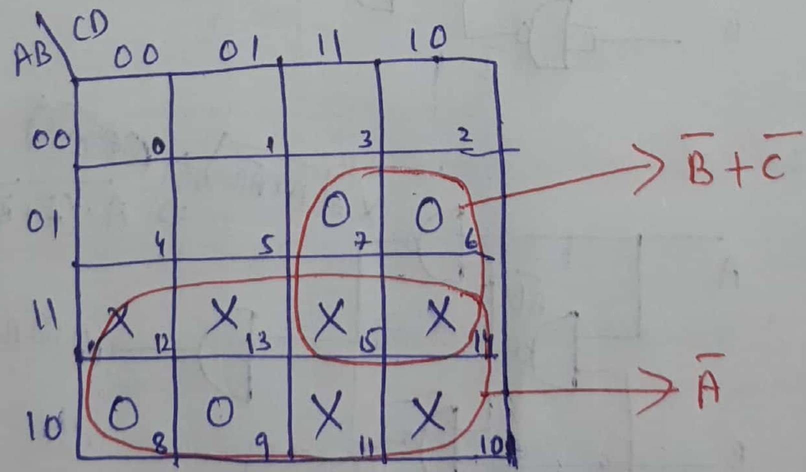


$$\boxed{F = A\bar{C}\bar{D} + AB + \bar{A}D + \bar{A}C}$$

Q Minimize the given expression $F(A,B,C,D) = m(0,1,2,3,4,5) + d(10,11,12,13,14,15)$ in POS form.

Sol The POS form of the given function is given by other terms of the given expression

$$F(A,B,C,D) = \Pi M(6,7,8,9) + d(10,11,12,13,14,15)$$



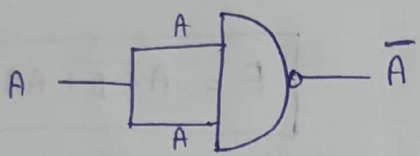
$$\boxed{F = \bar{A}(\bar{B}+\bar{C})}$$

H.W

1. $F(A,B,C,D) = \Sigma m(1,3,7,11,15) + d(0,2,5)$
2. $F(A,B,C,D) = \Sigma m(0,1,2,3,4,6,9,10) + d(7,11,12,13,15)$
3. $F(A,B,C,D) = \Pi M(1,2,6,7,8,13,14,15) + d(3,5,12)$

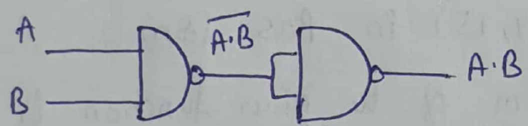
NAND - Realization :-

1. NOT :-

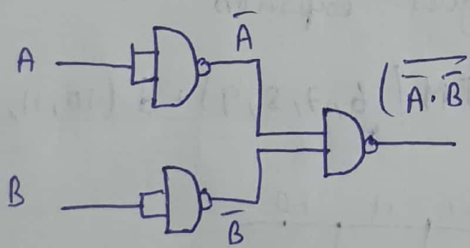


$[\because \overline{A \cdot A} = \bar{A}]$

2. AND gate :-

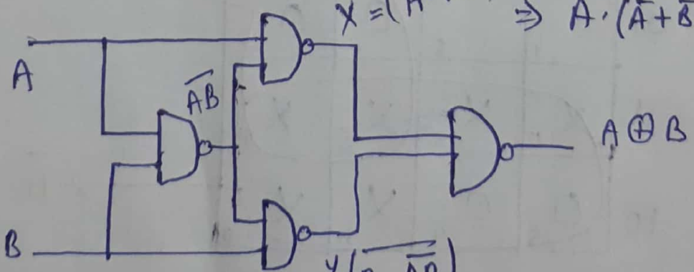


3. OR gate :-



$(\overline{\bar{A} \cdot \bar{B}}) \Rightarrow \bar{\bar{A}} + \bar{\bar{B}} \Rightarrow A + B.$

4. EX-OR gate :-



$X = (\overline{A \cdot \overline{AB}}) \Rightarrow$

$\Rightarrow \overline{A \cdot (\bar{A} + \bar{B})} \Rightarrow \overline{A \cdot \bar{B}}$ $(\because A\bar{A} = 0)$

$\Rightarrow (\bar{A} + B)$

$Y = (\overline{B \cdot \overline{AB}})$

$\Rightarrow \overline{B \cdot (\bar{A} + \bar{B})} = \overline{B \cdot \bar{A}}$ $(\because B \cdot \bar{B} = 0)$

$\Rightarrow \bar{B} + A$

$\Rightarrow \overline{X \cdot Y} \Rightarrow \overline{(\bar{A} + B)(\bar{B} + A)} \Rightarrow \overline{\bar{A}\bar{B} + \bar{A}A + B\bar{B} + BA}$

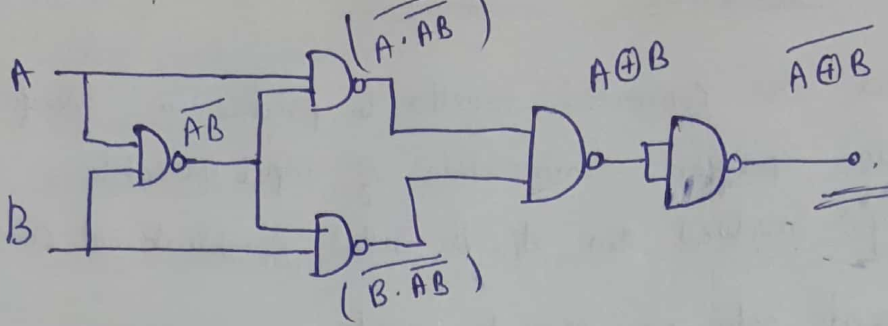
$\Rightarrow \overline{\bar{A}\bar{B} + AB} = \overline{A \odot B} = A \oplus B$

NOTE

To implement EX-OR gate using NAND gate it requires 4-Nand gates

NOTE
To implent EX-NOR gate using NAND gate it requires 5 NAND gates

EX-NOR



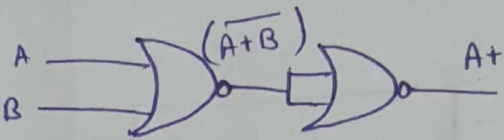
$\overline{A \oplus B} \Rightarrow A \odot B.$

NOR Realization :-

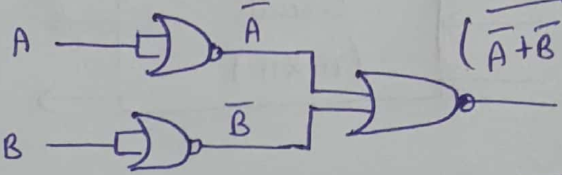
① NOT gate :-

$(\overline{A+A} = \bar{A})$

② OR gate :-



③ AND gate :-

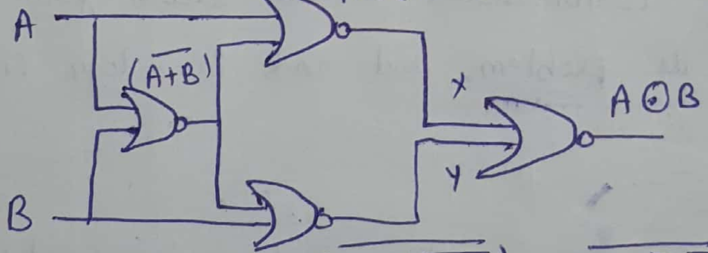


$(\overline{\bar{A}+\bar{B}}) \Rightarrow (\bar{\bar{A \cdot B}}) = A \cdot B.$

④ EX-NOR :-

To implement EX-NOR gate using NOR it requires 4 NOR gates

$X = \overline{A + (\overline{A+B})} = \overline{A+(\bar{A} \cdot \bar{B})} = \bar{A} \cdot \overline{\bar{A} \cdot \bar{B}}$
$\Rightarrow \bar{A}(A+B)$
$\Rightarrow \bar{A}B$

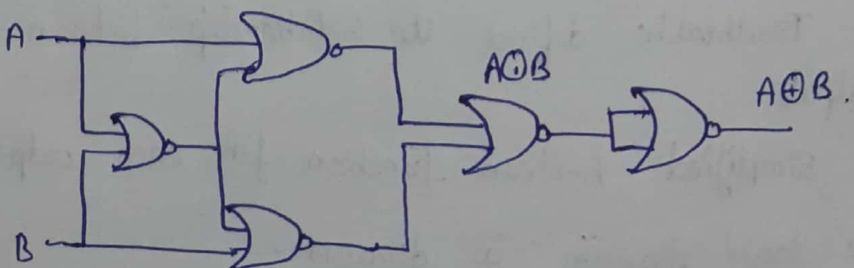


$Y = (\overline{B + (\overline{A+B})}) \Rightarrow \overline{B + \bar{A} \cdot \bar{B}} \Rightarrow \bar{B} \cdot \overline{\bar{A} \cdot \bar{B}}$
$\Rightarrow \bar{B}(A+B)$
$\Rightarrow \bar{B}A$

$\overline{X+Y} \Rightarrow \overline{\bar{A}B + \bar{B}A} \Rightarrow \overline{\bar{A}B} \cdot \overline{\bar{B}A}$
$\Rightarrow (A+\bar{B})(B+\bar{A}) \Rightarrow AB + \bar{B}\bar{A} \Rightarrow A \odot B.$

⑤ EX-OR :-

To implement EX-OR gate by using NOR gates it require 5 NOR gates



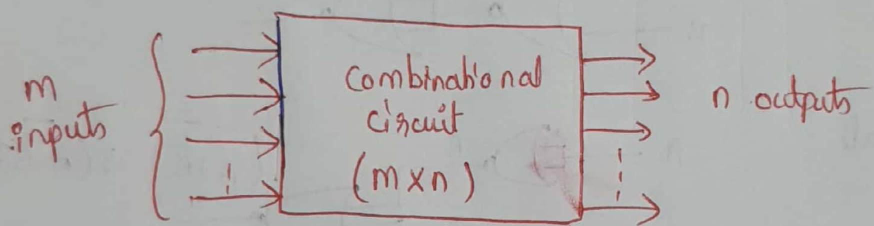
Combinational Circuits :-

⇒ When logic gates are connected together to produce a specified output on certain specified combination of input variable with no memory involved then it is called combinational circuit

→ Here output depends only on present input.

$$\text{output} = F(\text{Input})$$

→ A Combinational circuit will have m-binary inputs and n binary outputs.



Design procedure :-

→ The design of combinational circuit starts from the verbal description of the problem and ends in a logic circuit diagram.

Steps to design :-

1. The problem is stated. The number of available input variables and required output variables is determined.
2. The input and output variables are assigned letter/symbols
3. The Truthtable defines the relationship between inputs and outputs.
4. The simplified boolean function for each output is obtained.
5. The logic diagram is drawn.

Examples of Combinational Circuit

1. Adders and Subtractors
2. Decoder
3. Multiplexer
4. Code - Converters
5. Comparator
6. ROM

Problems

1. Design a combinational circuit with two inputs that produce logic "0" when any one input is one?

S: Given :- Two inputs
One output

1. Let inputs A, B
output C

2. Truth Table

input		output
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

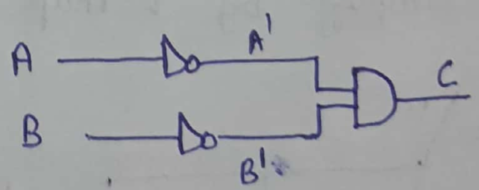
Since any one input is ~~zero~~ one
o/p will be zero.

3. simplify by K-map

A \ B	0	1
0	1	0
1	0	0

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

4. Draw the logic circuit

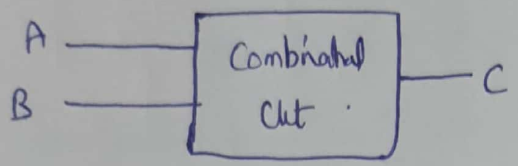


(2) Design a Combinational Circuit with two inputs that produces high output when there are odd numbers of 1's in the input?

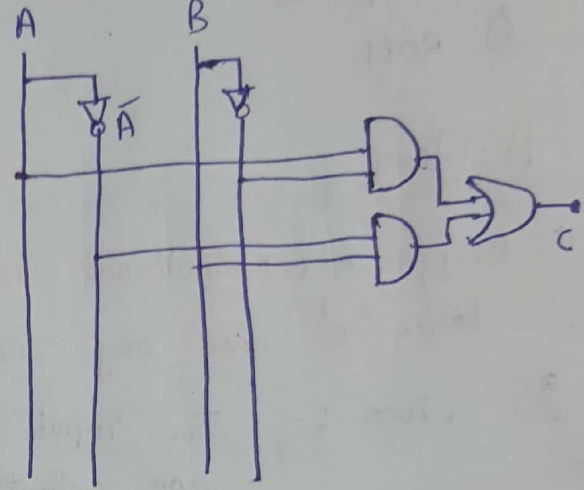
Sol:

Given :- Input - 2
output - 1

(1) Combinational Circuit



(5) logic Circuit



(2) Truth Table

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

(3) Simplify by K-map

A \ B	0	1
0		1
1	1	

$A'B$

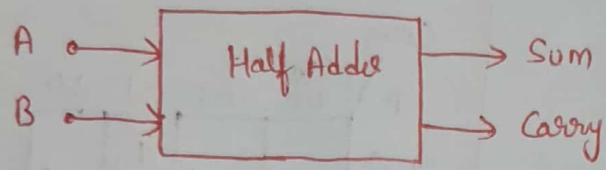
AB'

(4) $C = AB' + A'B$

ADDERS.

→ Half adder is a combinational logic circuit designed to add two single bit numbers.

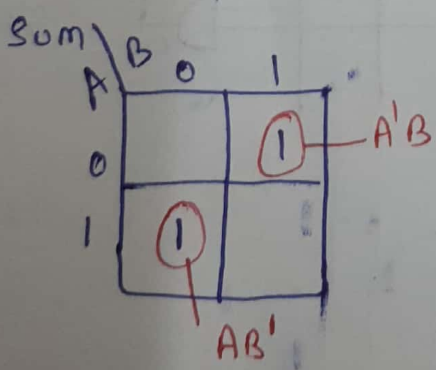
→ It contains two input and two outputs (sum & carry)



Truth Table.

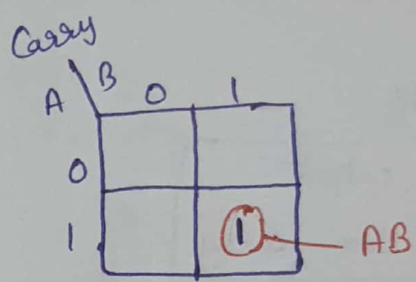
Input		output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Simplify by k-map :-



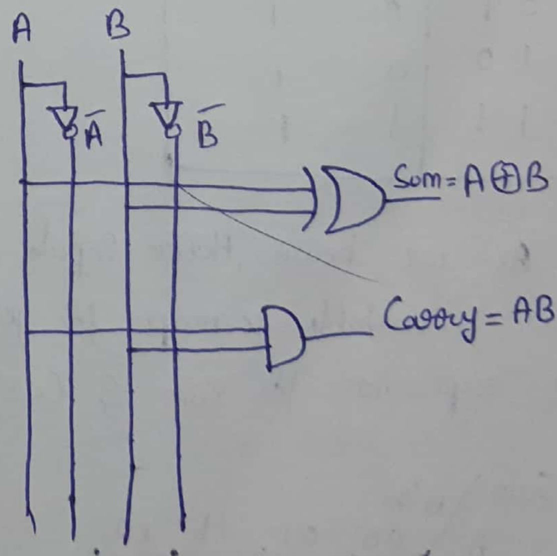
$Sum = AB' + A'B$

$\Rightarrow A \oplus B$



$Carry = AB$

logic circuit.



Full Adder :-

→ It is a arithmetic logic Circuit designed to add two single bit numbers with a Carry.

→ It has three inputs and two outputs.

Truth Table.

input A	input B	input C	output Sum	output Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

→ As we have three inputs, we use 3-Variable K-map for getting the expression for Sum & Carry.

Sum

A \ BC_{in}	00	01	11	10
0	0	1 1	3	1 2
1	1 4	5	1 7	6

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

As all are diagonal take EX-OR of all inputs.

Carry :-

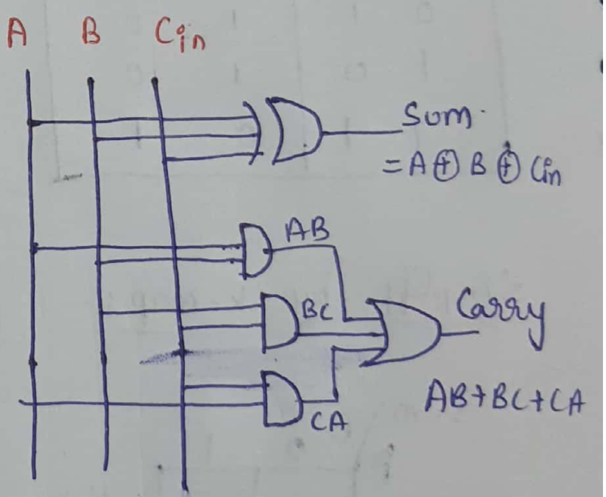
Carry

A \ BC_{in}	00	01	11	10
0	0	1	1 3	2
1	4	1 5	1 7	1 6

AC BC AB

$Carry = AB + BC_{in} + AC_{in}$

logic Circuit



4 - Bit Binary Adder :-

→ Binary adder is a digital circuit that performs arithmetic sum of two binary numbers of any length.

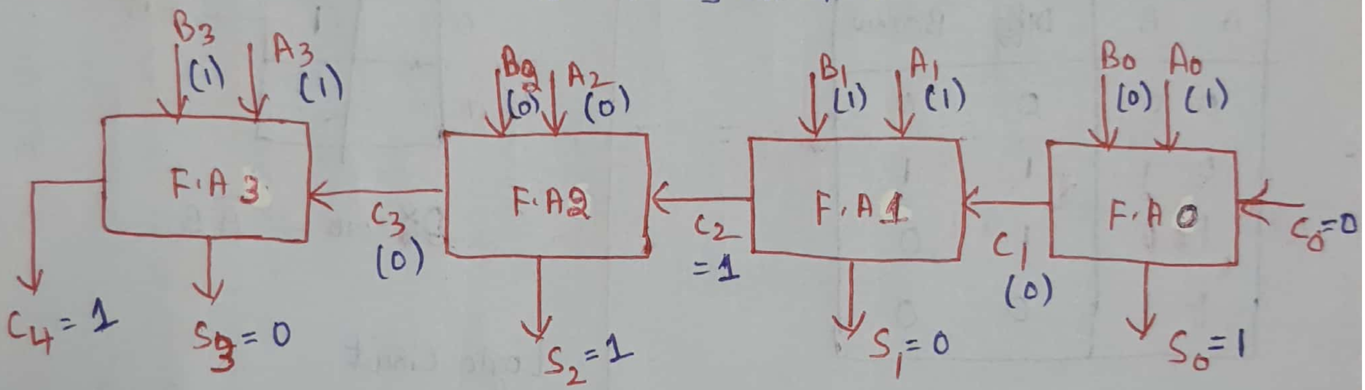
→ It is constructed by using full adders connected in series with output as carry connected to the next full adder and another output as sum.

→ Let the inputs be A, B

As the length is been given as 4 bit

$A \longrightarrow A_3 A_2 A_1 A_0$

$B \longrightarrow B_3 B_2 B_1 B_0$



→ Let's consider an example. Take 4 bits

A → 1 0 1 1

B → 1 0 1 0

A + B ⇒

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ +\ 1\ 0\ 1\ 0 \\ \hline 1\ 0\ 1\ 0\ 1 \\ \hline \end{array}$$

Carry discarded.

Let's simplify it by apply A, B inputs to above diagram.

→ Initially assign $C_0 = 0$, as we don't have carry.

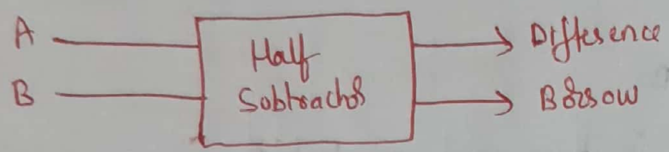
$(S_3 S_2 S_1 S_0) = (0101)$

Subtractor :-

Half Subtractor :-

→ It is a combinational circuit used to get the difference between two single bit numbers.

→ It has two inputs and two outputs difference and borrow.

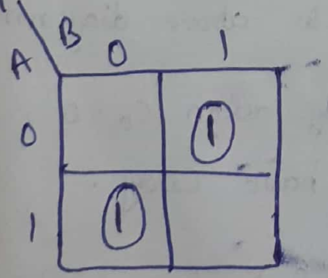


Truth Table

Input		Output	
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Simplify Diffence & borrow to obtain expressions by K-map.

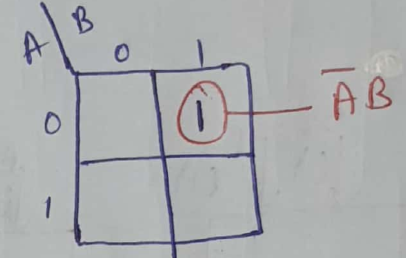
Diff



$Diff = A \oplus B$

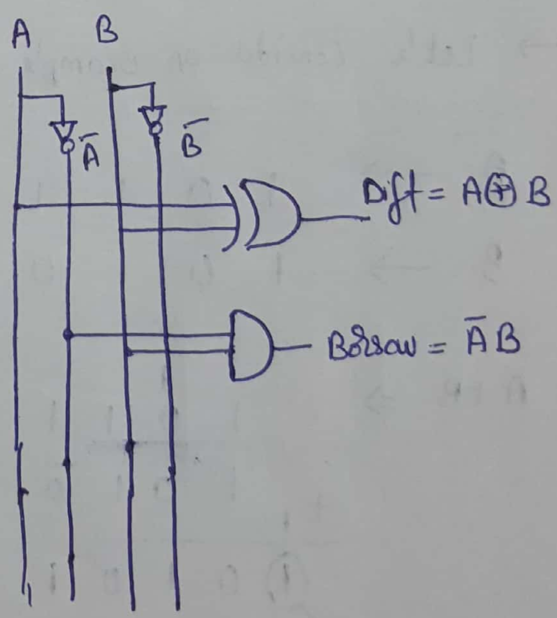
$diff = A'B + AB'$

Borrow



$Borrow = \bar{A}B$

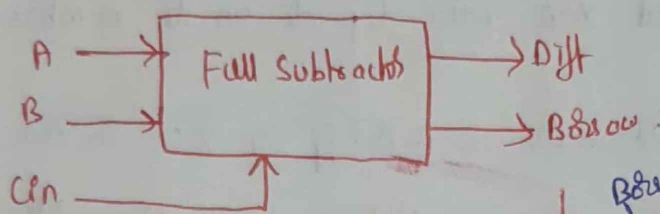
Logic Circuit



Full Subtractor :-

→ Full Subtractor is a Combinational Circuit used to perform Subtraction among 3 bits

→ It has 3 inputs and two outputs (difference & borrow)



Truth Table.

Input			Output	
A	B	Cin	Diff	Borrow
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

→ Simplify Diff & Borrow to get the expression.

→ As we have 3 inputs Consider 3 variable k-map.

Diff

A \ B Cin	00	01	11	10
0	0	1 1	3	1 2
1	1 4	5	1 7	6

$Diff = A \oplus B \oplus Cin$.

Borrow

A \ B Cin	00	01	11	10
0	0	1 1	1 3	1 2
1	4	5	1 7	6

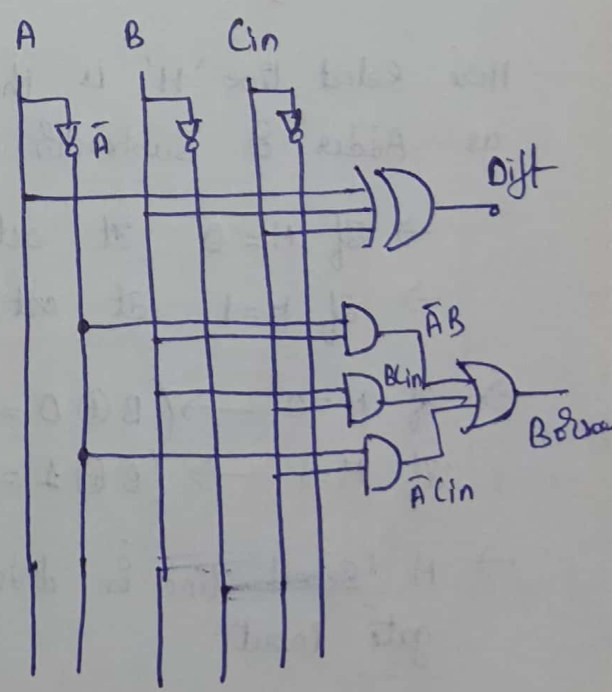
$\bar{A} Cin$

$\bar{A} B$

$B Cin$

$Borrow = \bar{A}B + BCin + \bar{A}Cin$

logic Circuit



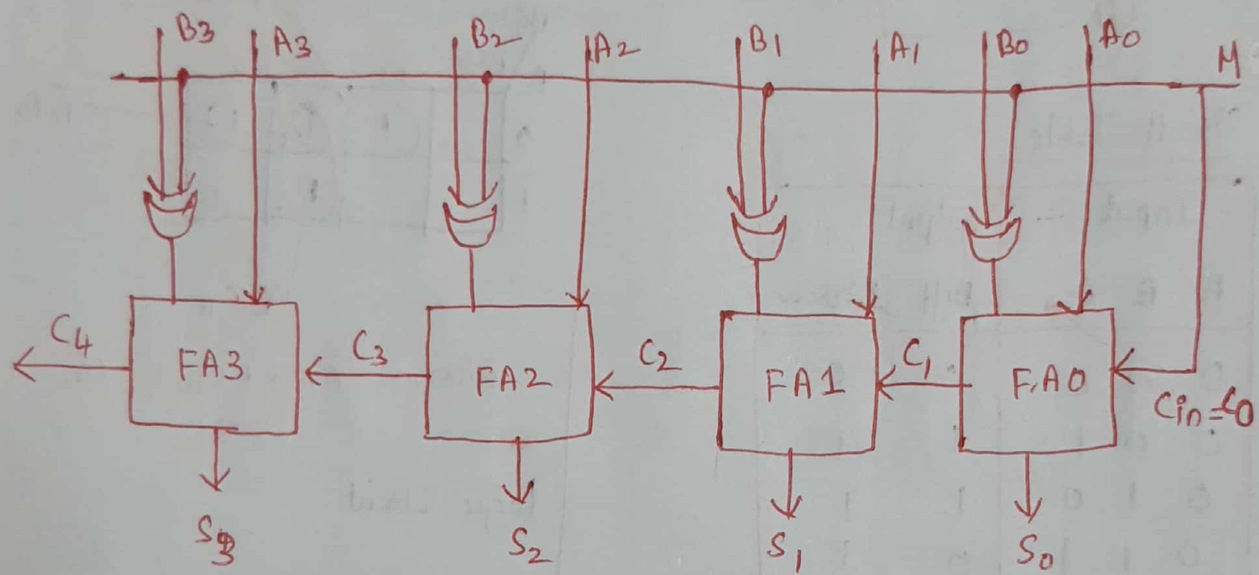
4-Bit Binary Adder / Subtractor :-

→ It is a combinational circuit that performs both Adder & Subtraction at a single time based on the select line "M".

→ The circuit has Full adders and XOR gates in it.

→ Number of full adders and XOR gates depends on the number of bits.

→ Here as it 4 bit, we will have 4 F.A & 4 XOR in our circuit.



Here select line 'M' is the source to make the circuit operate as Adder & Subtractor

→ If M = 0 it act as adder

→ If M = 1 it act as subtractor.

→ If $M = 0 \longrightarrow (B \oplus 0 = B)$

If $M = 1 \longrightarrow (B \oplus 1 = \overline{B})$

→ M select line is drived to all the full adder by XOR gate input

→ Lets add two numbers $6 + 2 = 8$

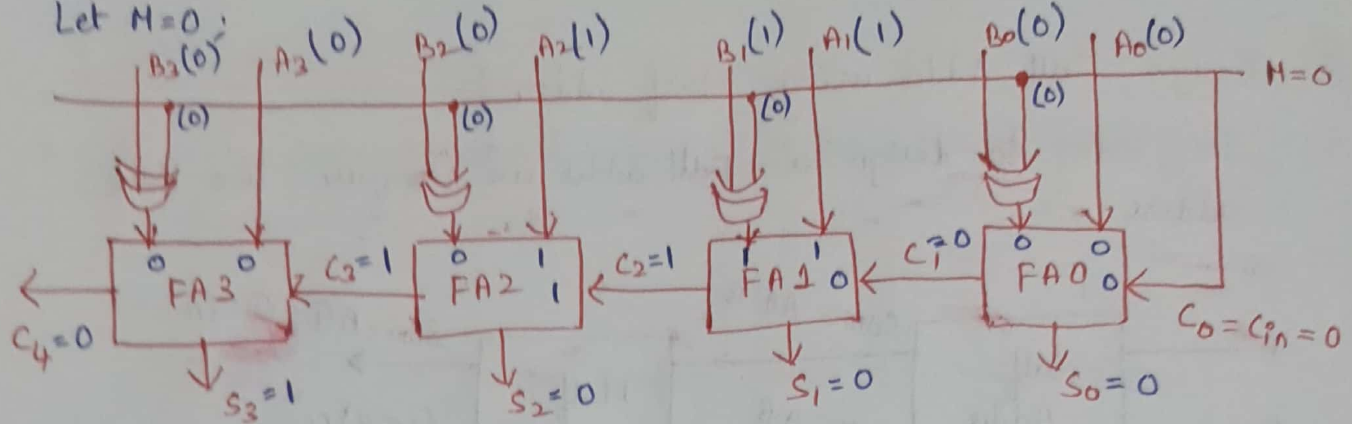
A → 0 1 1 0

B → 0 0 1 0

1 0 0 0

CASE-1

Let $M=0$;



After solving we got $(S_3 S_2 S_1 S_0) = (1\,0\,0\,0)$.

Case-(ii)

Let $M = 1$; now circuit behaves like subtractor.

Let take same numbers $6-2 = 4$.

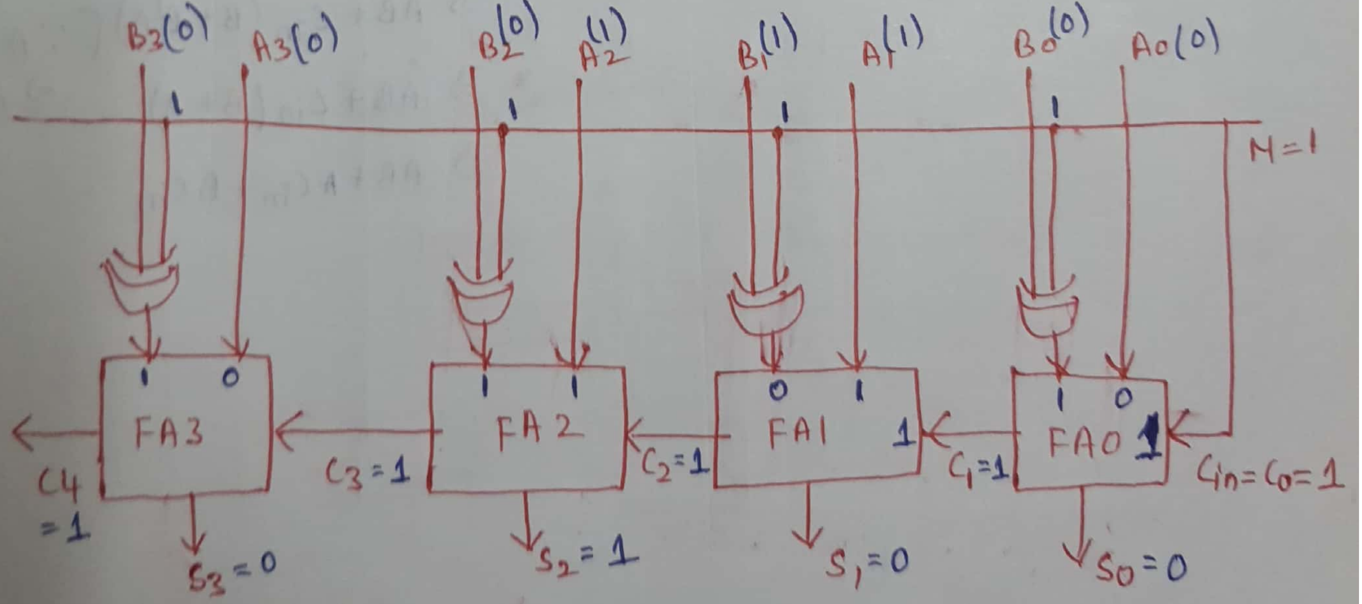
$A = 6 \rightarrow 0110$

$B = -2 \rightarrow 0010$

$0010 \xrightarrow{2's} 1101$
$+1$
1110

$$
\begin{array}{rl}
 & 0110 \rightarrow (6) \\
 & 1110 \rightarrow (-2) \\
\hline
1 & 0100 \rightarrow (4)
\end{array}
$$

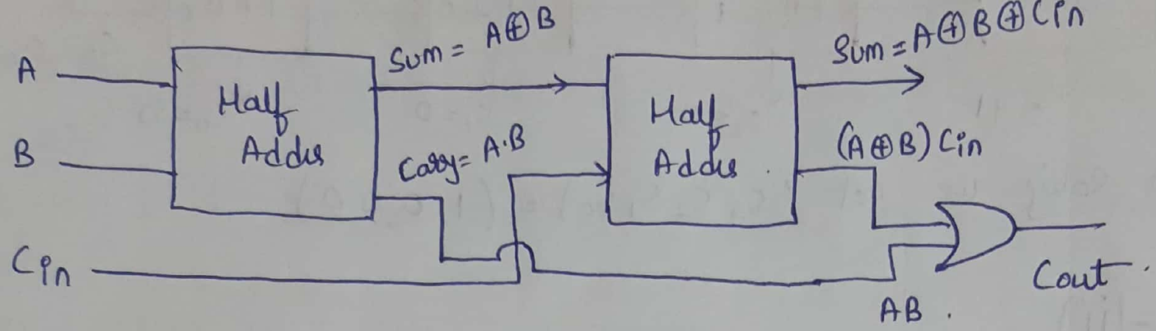
Discard the carry.



$(S_3 S_2 S_1 S_0) = (0100)$.

Q. Design Full Adder using Half adder?

A: In order to design a Full adder we require two Half adders.



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

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

$$\Rightarrow AB + A'BC_{in} + AB'C_{in} \quad \left(\begin{array}{l}\text{from} \\ A + A'B \\ = A+B\end{array}\right)$$

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

$$\Rightarrow B(A + C_{in}) + B'AC_{in}$$

$$\Rightarrow AB + BC_{in} + B'AC_{in}$$

$$\Rightarrow AB + C_{in}(B + B'A) \quad [\because A + A'B \Rightarrow A+B]$$

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

$$\Rightarrow AB + AC_{in} + BC_{in}$$

BINARY MULTIPLIER.

→ It is a Combinational Circuit that performs multiplication of Two binary numbers.

Ex:- A × B

A, B are two binary numbers used for multiplication.

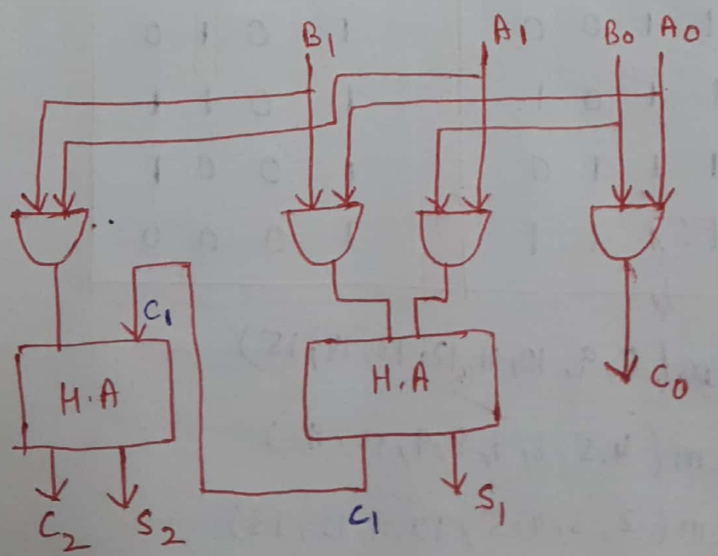
A → multipicand
B → multiplier.

2×2 Multiplier.

A ← $A_1 A_0$
B ← $B_1 B_0$

$$
\begin{array}{ccccc}
 & & A_1 & A_0 & \\
 & \times & B_1 & B_0 & \\
\hline
 & C_1 & B_0A_1 & B_0A_0 & \\
C_2 & B_1A_1 & B_1A_0 & \times & \\
\hline
C_2 & (C_1+B_1A_1) & (B_0A_1+B_1A_0) & B_0A_0 &
\end{array}
$$

↓ Carry (C_2) ↓ Sum (S_2) ↓ Sum (S_1) ↓ Carry (C_0).



CODE CONVERTER:-

→ Code converters are used to convert from one code to another code.

(Q) Design a 4 bit binary to Gray code converter?

S: Inputs = 4 (B_3 B_2 B_1 B_0) ; outputs = 4 (G_3, G_2 G_1, G_0)

Decimal	Binary Input B_3	B_2	B_1	B_0	Gray Code G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

$G_3 = \Sigma m(8, 9, 10, 11, 12, 13, 14, 15)$

$G_2 = \Sigma m(4, 5, 6, 7, 8, 9, 10, 11)$

$G_1 = \Sigma m(2, 3, 4, 5, 10, 11, 12, 13)$

$G_0 = \Sigma m(1, 2, 5, 6, 9, 10, 13, 14)$

G_3

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	1 (12)	1 (13)	1 (15)	1 (14)
10	1 (8)	1 (9)	1 (11)	1 (10)

$\boxed{G_3 = B_3}$

G_2

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	1 (4)	1 (5)	1 (7)	1 (6)
11	12	13	15	14
10	1 (8)	1 (9)	1 (11)	1 (10)

$\bar{B_3} B_2$ $\quad$ $B_3 \bar{B_2}$

$\boxed{G_2 = B_2 \oplus B_3}$

G_1

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	1 (3)	1 (2)
01	1 (4)	1 (5)	7	6
11	1 (12)	1 (13)	15	14
10	8	9	1 (11)	1 (10)

$B_2 \bar{B_1}$ $\quad$ $\bar{B_2} B_1$

$\boxed{G_1 = B_2 \bar{B_1} + \bar{B_2} B_1 \Rightarrow B_2 \oplus B_1}$

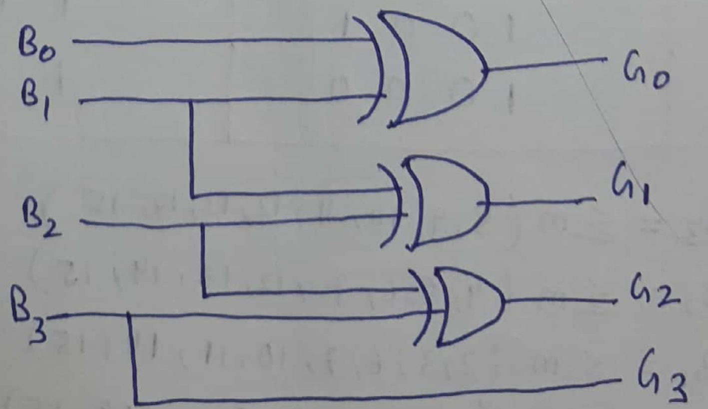
G_0

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1 (1)	3	1 (2)
01	4	1 (5)	7	1 (6)
11	12	1 (13)	15	1 (14)
10	8	1 (9)	11	1 (10)

$\bar{B_1} B_0$ $\quad$ $B_1 \bar{B_0}$

$\boxed{G_0 = \bar{B_1} B_0 + B_1 \bar{B_0} \Rightarrow B_1 \oplus B_0}$

Logic Circuit



Q Design a 4 bit Gray Code to Binary code?

S

Inputs = 4 (Gray Code) (G_3 G_2 G_1 G_0)

outputs = 4 (Binary code) (B_3 B_2 B_1 B_0)

Gray Code				Binary Code			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

B_3 (G_3G_2 \ G_1G_0)

G_3G_2 \ G_1G_0	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

G_3

$$\boxed{B_3 = G_3}$$

B_2 (G_3G_2 \ G_1G_0)

G_3G_2 \ G_1G_0	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$\bar{G}_3 G_2$ $\quad$ $G_3 \bar{G}_2$

$$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2$$

$$\boxed{B_2 \Rightarrow G_3 \oplus G_2}$$

B_1 (G_3G_2 \ G_1G_0)

G_3G_2 \ G_1G_0	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

$\bar{G}_3 \bar{G}_2 G_1$

$\bar{G}_3 G_2 \bar{G}_1$

$G_3 G_2 G_1$

$G_3 \bar{G}_2 \bar{G}_1$

$$B_1 = G_3 \bar{G}_2 \bar{G}_1 + G_3 G_2 G_1 + \bar{G}_3 \bar{G}_2 G_1 + \bar{G}_3 G_2 \bar{G}_1$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

B_0 (G_3G_2 \ G_1G_0)

G_3G_2 \ G_1G_0	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

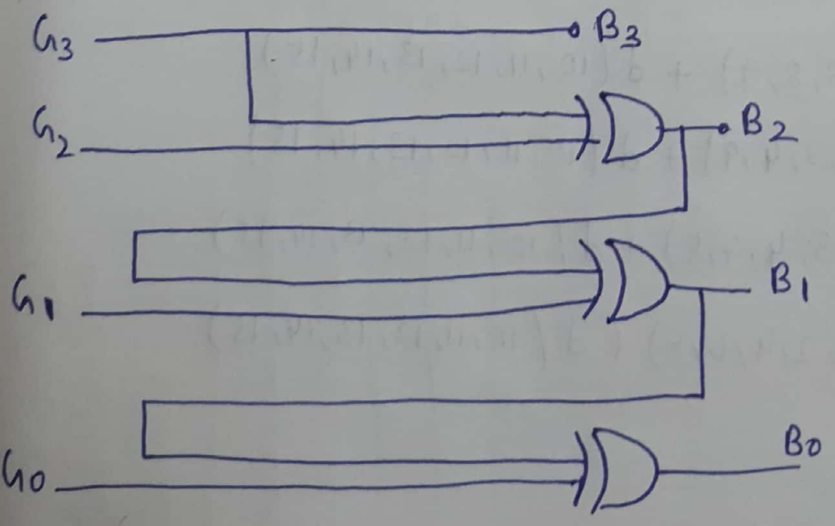
$\bar{G}_3 \bar{G}_2 \bar{G}_1 G_0$ $\quad$ $\bar{G}_3 \bar{G}_2 G_1 \bar{G}_0$

$\bar{G}_3 G_2 \bar{G}_1 \bar{G}_0$ $\quad$ $\bar{G}_3 G_2 G_1 G_0$

$G_3 G_2 \bar{G}_1 G_0$ $\quad$ $G_3 G_2 G_1 \bar{G}_0$

$G_3 \bar{G}_2 \bar{G}_1 \bar{G}_0$ $\quad$ $G_3 \bar{G}_2 G_1 G_0$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0.$$



H.W ① Design 3 bit binary to Gray Converter,
② Design 3 bit Gray to Binary Converter

(6) Convert a 4 bit BCD to Excess-3 Code?

S: inputs – 4 – (A, B, C, D) outputs – 4 (E_3 E_2 E_1 E_0).

→ As it is BCD → 0 to 9 will be active states (10 to 15) will be invalid state.

Decimal	BCD				Excess-3			
	A	B	C	D	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

$$E_3 = \sum m(5,6,7,8,9) + d(10,11,12,13,14,15)$$

$$E_2 = \sum m(1,2,3,4,9) + d(10,11,12,13,14,15)$$

$$E_1 = \sum m(0,3,4,7,8) + d(10,11,12,13,14,15)$$

$$E_0 = \sum m(0,2,4,6,8) + d(10,11,12,13,14,15).$$

E_3

AB \ CD	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

A, BD, BC

$$\boxed{E_3 = A + BD + BC}$$

E_2

AB \ CD	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

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

$$\boxed{E_2 = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C}$$

E_1

AB \ CD	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

$\bar{C}\bar{D}$ CD

$$\boxed{E_1 = C \odot D} \quad (\because \bar{C}\bar{D} + CD = C \odot D)$$

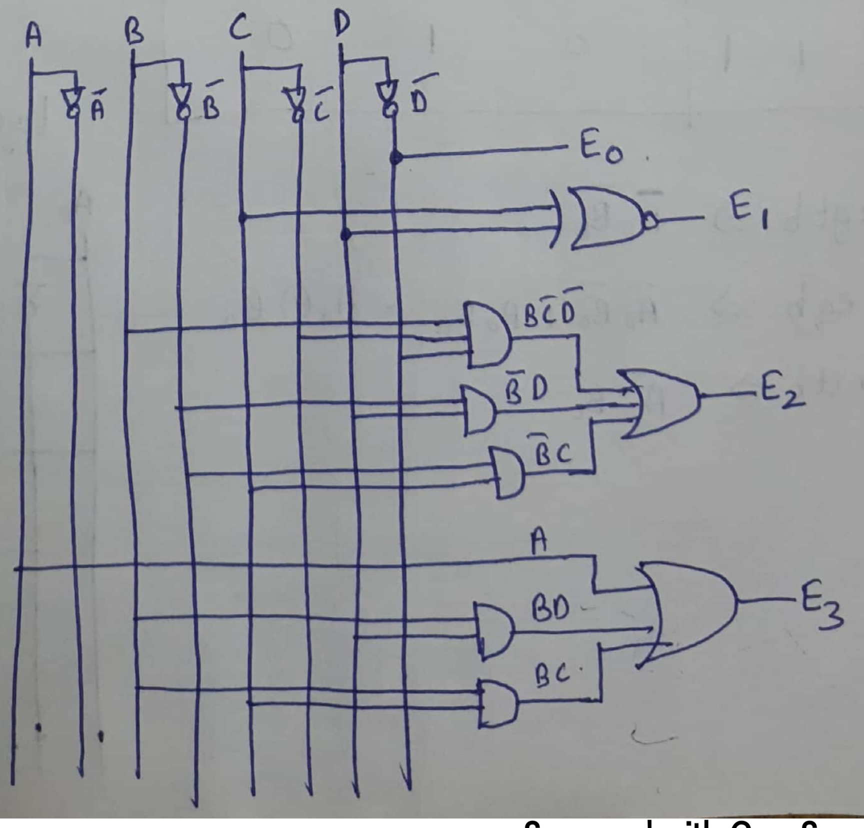
E_0

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$\bar{D}$

$$\boxed{E_0 = \bar{D}}$$

logic Circuit

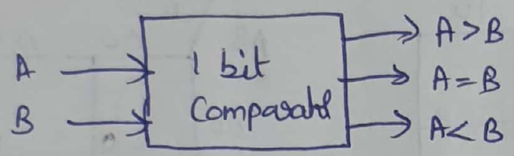


~~Convert BCD to seven segment.~~

COMPARATOR :-

→ Comparator is used to compare two N bit numbers ; let say a & b.
→ It is going to compare magnitude of numbers providing a > b
a < b & a eq b.

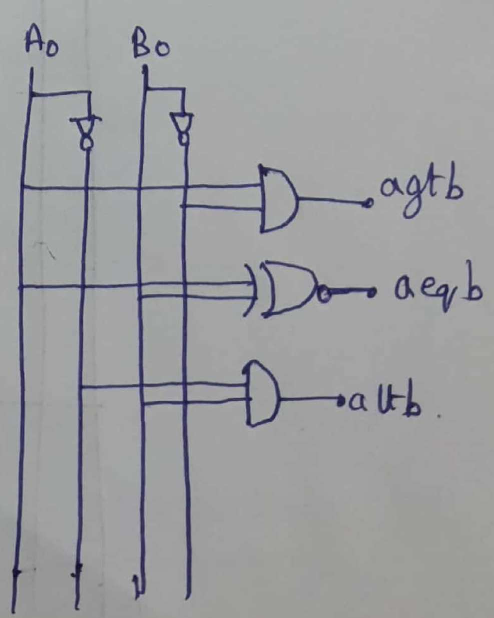
Single bit & 1-bit Comparator :-



Truth Table.

Input		output		
A_0	B_0	agtb	aeqb	altb.
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

logic circuit



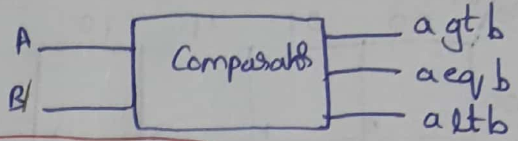
agtb ⇒ $A_0 \overline{B_0}$

aeqb ⇒ $\overline{A_0}\,\overline{B_0} + A_0 B_0 = A_0 \odot B_0$

altb ⇒ $\overline{A_0}\, B_0$.

2- Bit Comparator :- Here let take two Inputs A, B. Each input having length two i.e. $A = A_1 A_0$; $B = B_1 B_0$; also as outputs are agt b, aeq b, alt b.

Truth Table.



Inputs				outputs		
A		B				
A_1	A_0	B_1	B_0	agt b	aeq b	alt b
0	0	0	0	0	1	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	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

$agtb = \Sigma m(4, 8, 9, 12, 13, 14)$

$aeqb = \Sigma m(0, 5, 10, 15)$

$altb = \Sigma m(1, 2, 3, 6, 7, 11)$

(3)

Simplify by K-map.

agtb

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	0	1	3	2
01	1 4	5	7	6
11	1 12	1 13	15	1 14
10	1 8	1 9	11	10

$A_0\bar{B}_1\bar{B}_0$

$A_1A_0\bar{B}_0$

$A_1\bar{B}_1$

aeqb

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	1 0	1	3	2
01	4	1 5	7	6
11	12	13	1 15	14
10	8	9	11	1 10

$\bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0$

$\bar{A}_1A_0\bar{B}_1B_0$

$A_1A_0B_1B_0$

$A_1\bar{A}_0B_1\bar{B}_0$

altb

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	0	1 1	1 3	1 2
01	4	5	1 7	1 6
11	12	13	15	14
10	8	9	1 11	10

$\bar{A}_1B_1$

$\bar{A}_1\bar{A}_0B_0$

$B_1B_0\bar{A}_0$

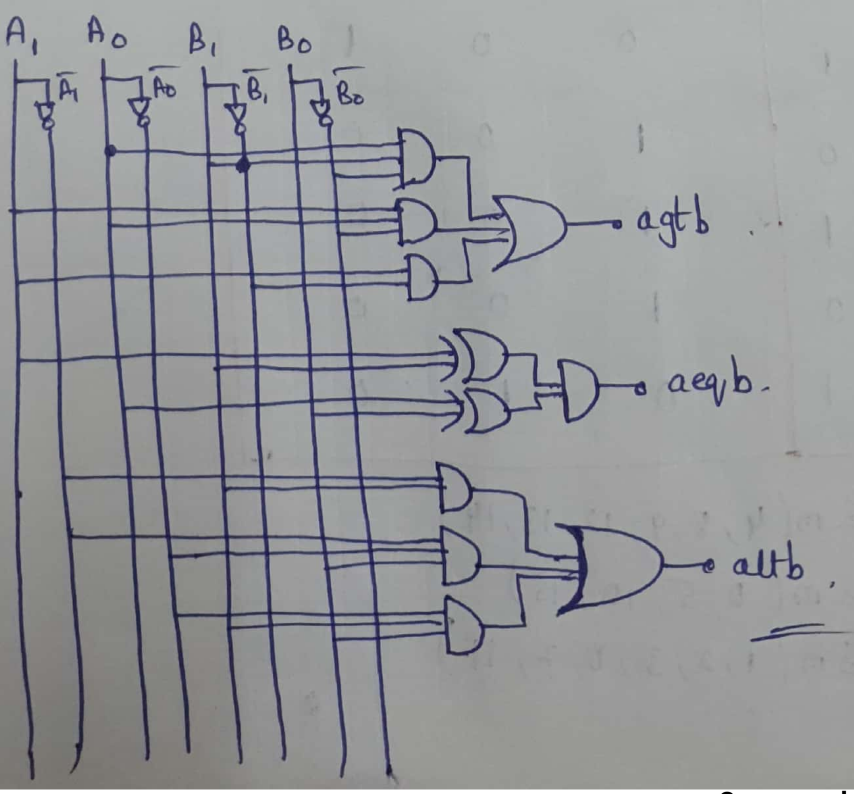
Simplified expression:-

$agtb = A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0 + A_1\bar{B}_1$

$aeqb = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 +$
$A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

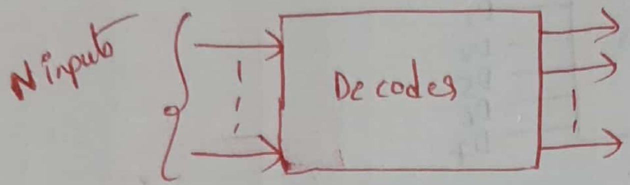
$\Rightarrow (A_1 \odot B_1)(A_0 \odot B_0)$

$altb \Rightarrow \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0$



Decoder :-

→ Decoder is a multi input multi output logic circuit that converts n bit binary input code to M output lines.



$M = 2^N$ output codes.

2 to 4 Decoder. (or) 2×4 Decoder:-

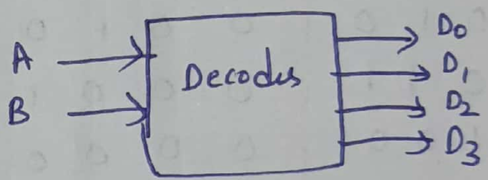
→ 2 to 4 Decoder has two inputs & four outputs.

→ Let two inputs be A, B, outputs be D_0 D_1 D_2 D_3.

NOTE :-

→ If E=0 Decoder is disabled

→ If E=1 Decoder is enabled.



(a) Block diagram.

Truth Table.

Enable	inputs		outputs			
E	A	B	D_0	D_1	D_2	D_3
0	X	X	X	X	X	X
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

Expressions.

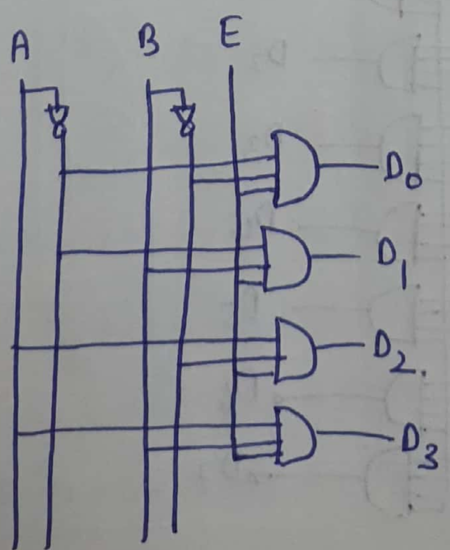
$D_0 = \bar{A}\bar{B}$

$D_1 = \bar{A}B$

$D_2 = A\bar{B}$

$D_3 = AB$.

logic Circuit

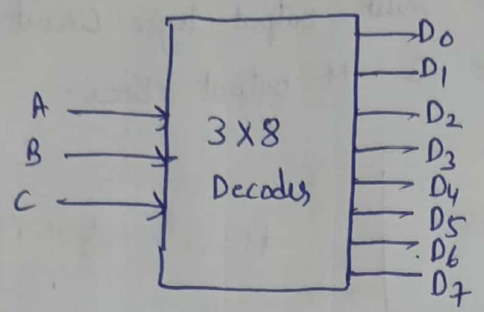


3 to 8 Decoder :-

3x8 Decoder has three inputs and eight outputs.

Let the three inputs be A, B, C and outputs be $D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ D_7$.

Block diagram



Truth Table :-

Inputs				outputs							
E	A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	X	X	X	0	0	0	0	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	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

expression of output

$D_0 = \bar{A}\bar{B}\bar{C}$

$D_1 = \bar{A}\bar{B}C$

$D_2 = \bar{A}B\bar{C}$

$D_3 = \bar{A}BC$

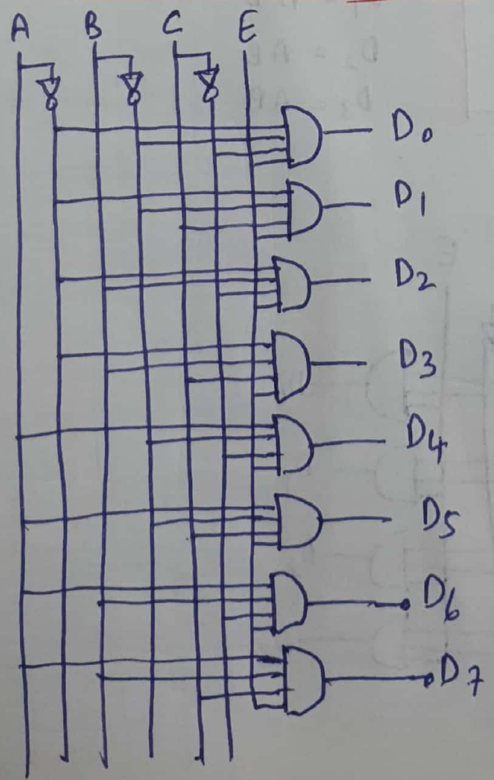
$D_4 = A\bar{B}\bar{C}$

$D_5 = A\bar{B}C$

$D_6 = AB\bar{C}$

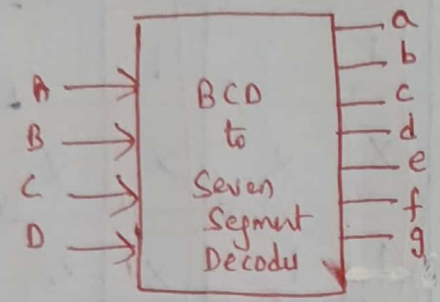
$D_7 = ABC$

logic Circuit

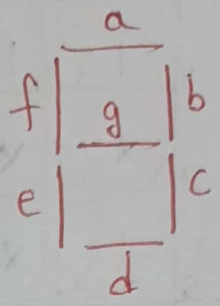


⑧ BCD to Seven Segment Decoder :-

Block diagram.



Seven Segment:



Truth Table.

Decimal Value	Inputs				outputs						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

BCD will have inputs from (0 to 9) & (10 to 15) will be Invalid states.

o/p. Expressions :-

$a = \Sigma m(0,2,3,5,6,7,8,9) + d(10-15)$

$b = \Sigma m(0,1,2,3,4,7,8,9) + d(10-15)$

$c = \Sigma m(0,1,3,4,5,6,7,8,9) + d(10-15)$

$d = \Sigma m(0,2,3,5,6,8,9) + d(10-15)$

$e = \Sigma m(0,2,6,8) + d(10-15)$

$f = \Sigma m(0,4,5,6,8,9) + d(10-15)$

$g = \Sigma m(2,3,4,5,6,8,9) + d(10-15)$

a

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

$\boxed{a = A + C + \bar{B}\bar{D} + BD}$

b

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

$\boxed{b = \bar{B} + A + CD + \bar{C}\bar{D}}$

c

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

$\boxed{c = A + B + \bar{C} + D}$

d

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

$\boxed{d = A + \bar{B}C + \bar{B}\bar{D} + C\bar{D} + B\bar{C}D}$

e

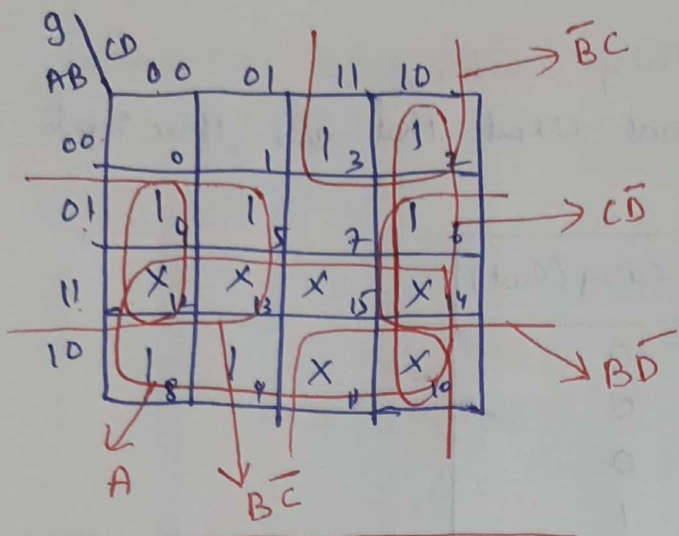
AB \ CD	00	01	11	10
00	1 (0)	(1)	(3)	1 (2)
01	(4)	(5)	(7)	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	(9)	X (11)	X (10)

$\boxed{e = C\bar{D} + \bar{B}\bar{D}}$

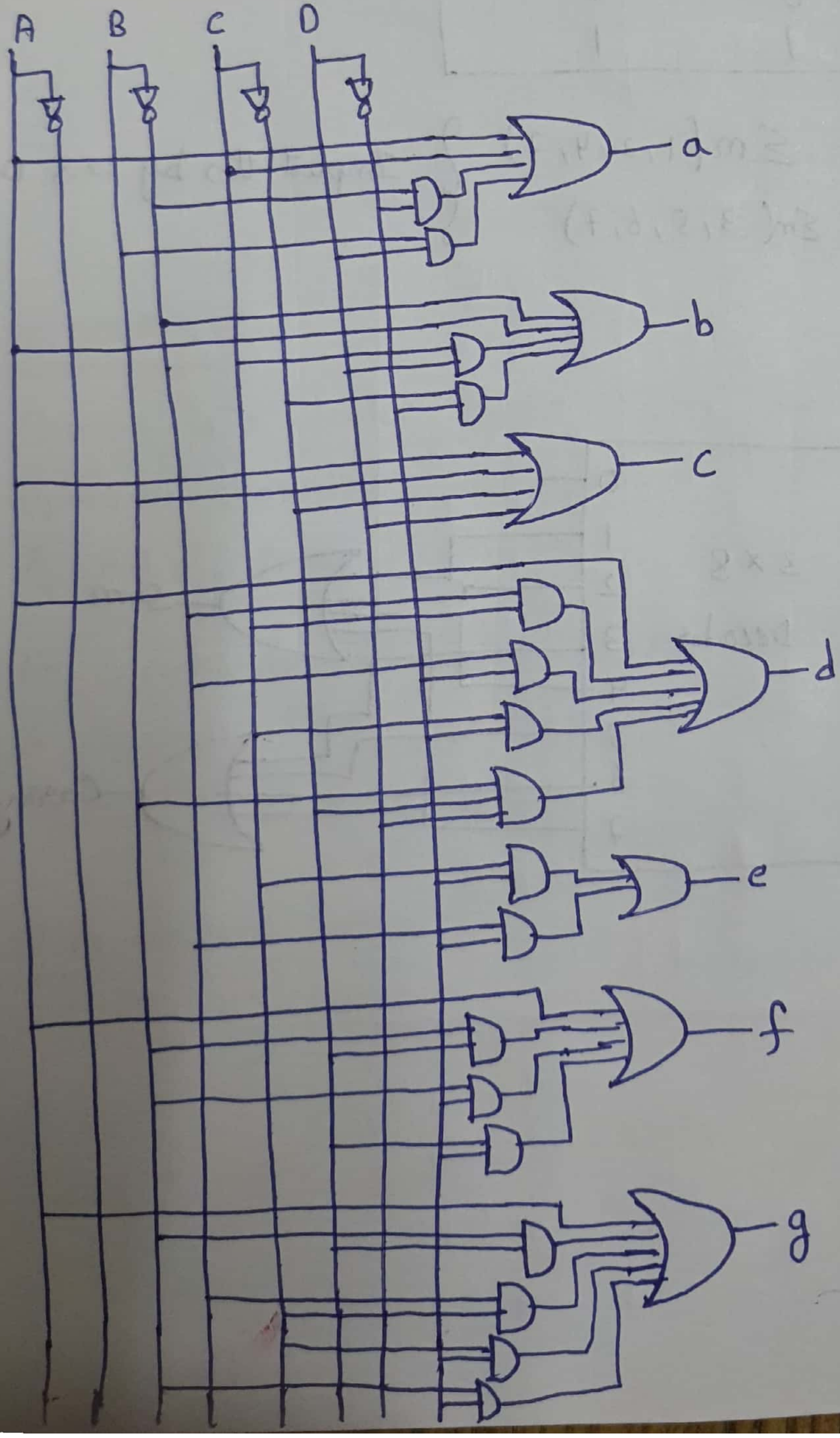
f

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

$\boxed{f = A + B\bar{C} + B\bar{D} + \bar{C}\bar{D}}$



$$\boxed{g = A + B\bar{C} + \bar{B}C + C\bar{D} + B\bar{D}}$$



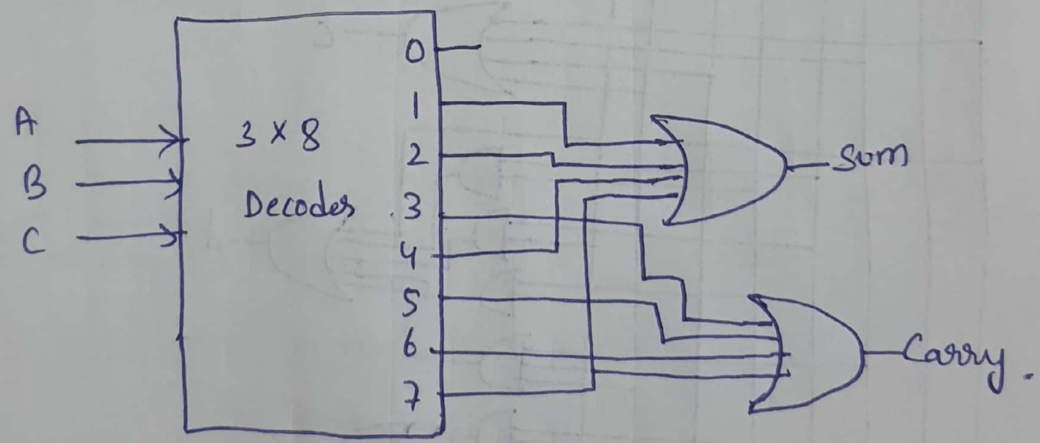
Q Implement the full adder using Decoders ?

S Full adder is a Combinational circuit that add three inputs and have two outputs.

A	B	C	Sum(S)	Carry (C_{out})
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

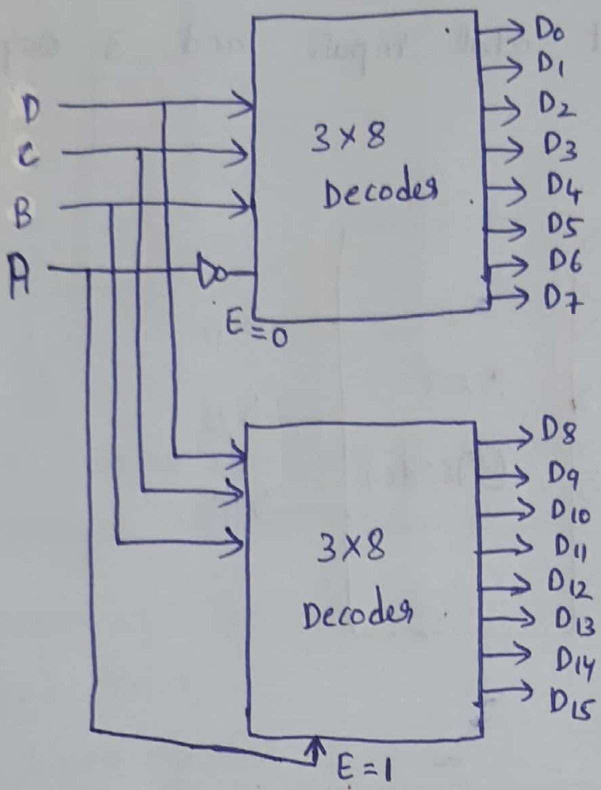
$$\left.\begin{array}{l} Sum = \Sigma m(1,2,4,7) \\ Carry = \Sigma m(3,5,6,7) \end{array}\right\}$$ Implement this by 3x8 Decoders

3x8 Decoder.



5. Implement 4×16 Decoder by using two 3×8 Decoders?

S



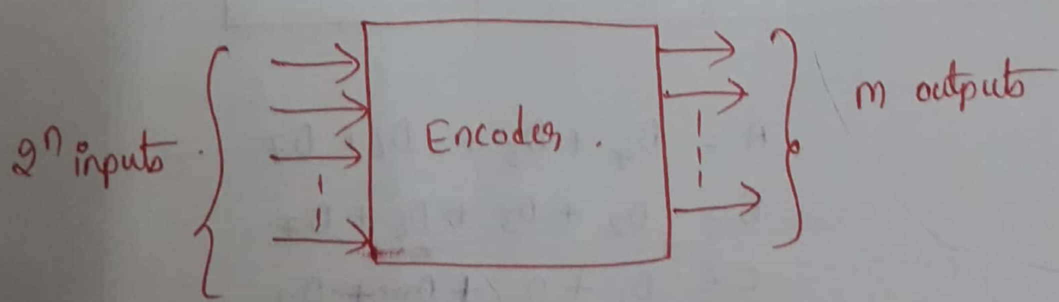
ENCODERS.

→ Encoder is a device whose inputs are decimal digits / or alphabetic characters and outputs are coded representation of those inputs.

→ An encoder is a device which converts familiar numbers or symbols into coded format.

→ It is reverse of decoder.

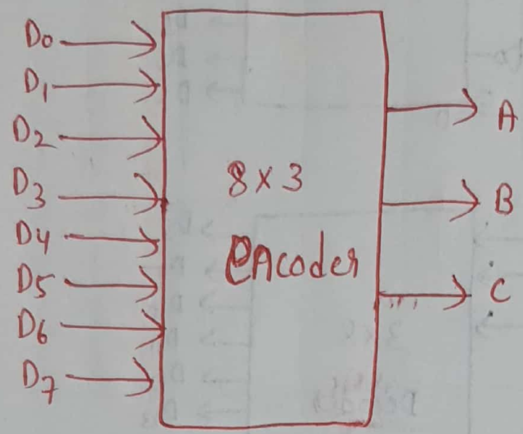
Block diagram



Octal to Binary Encoder :-

→ It is also called as 8 to 3 line encoder.
→ It has eight octal inputs and 3 outputs.

Block diagram.



Truth Table :-

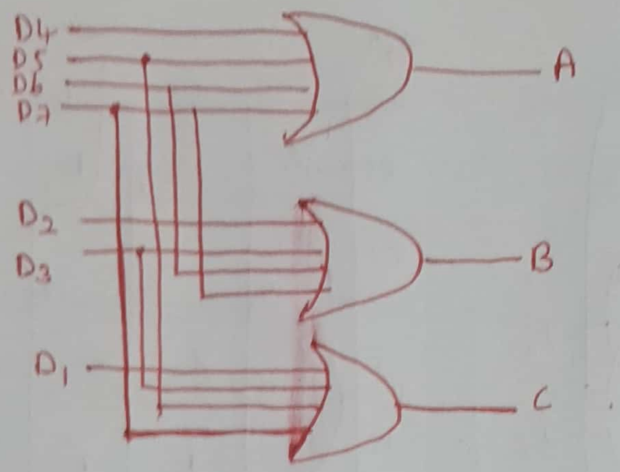
inputs								outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$A = D_4 + D_5 + D_6 + D_7$$

$$B = D_2 + D_3 + D_6 + D_7$$

$$C = D_1 + D_3 + D_5 + D_7$$

logic Circuit :-

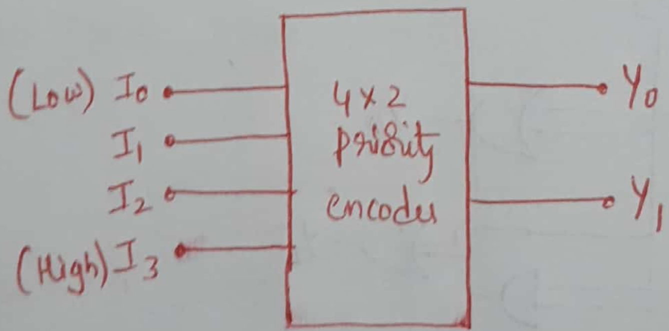


Priority Encoder :- It is one type of encoder where the input signal are prioritized to be driven at output.

→ 4 input priority encoder :- Here we have four inputs I_0, I_1, I_2, I_3 and two outputs Y_0 Y_1.

→ Here I_0 has lowest priority compared to remaining all, where as I_4 holds highest priority

Block diagram :-



Truth Table :

Inputs				outputs		
I_3	I_2	I_1	I_0	Y_0	Y_1	
0	0	0	0	X	X	
0	0	0	1	0	0	when I_0 is high
0	0	1	X	0	1	when I_1 is high
0	1	X	X	1	0	when I_2 is high
1	X	X	X	1	1	when I_3 is high

Y_0: I_3I_2 \ I_1I_0	00	01	11	10
00	X			
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

I_2

I_3

Y_1: I_3I_2 \ I_1I_0	00	01	11	10
00	X		1	1
01				
11	X	X	X	X
10	1	1	1	1

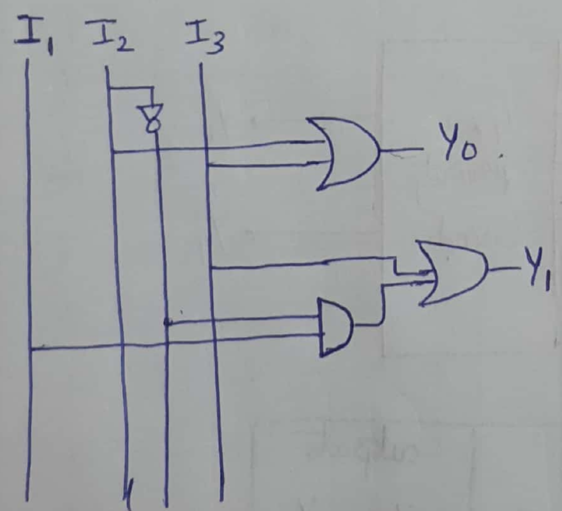
I_3

$\overline{I_2}\, I_1$

$Y_0 = I_2 + I_3$

$Y_1 = I_3 + \overline{I_2}\, I_1$

logic Circuit



H.W

(Q) Design a Decimal to BCD Encoder?

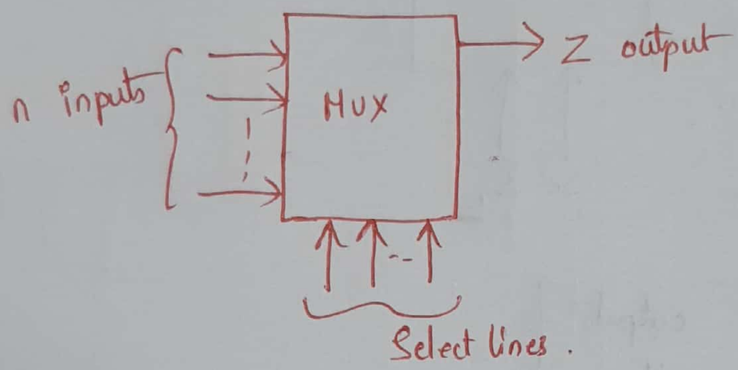
Multiplexers:-

→ Multiplexer is also called as Data Selector.

→ It is a logic circuit that accepts several data inputs and allows only one of them to get through to the output.

→ The routing of the desired data input to the output is controlled by select inputs.

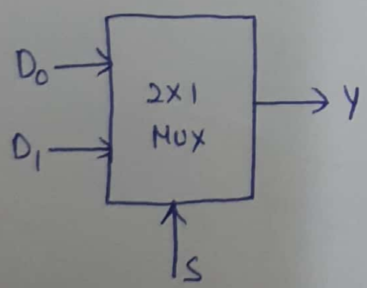
Block diagram:-



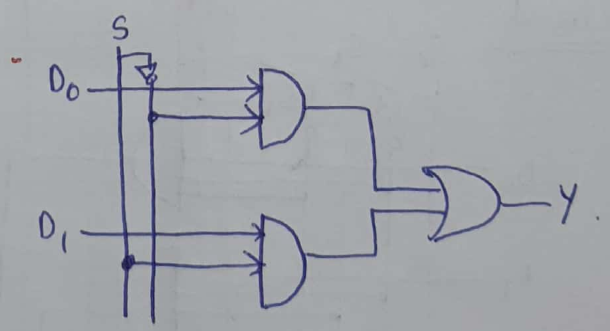
2×1 Mux

→ It has two inputs, one select line and one output.

Block diagram



logic Circuit



Truth Table.

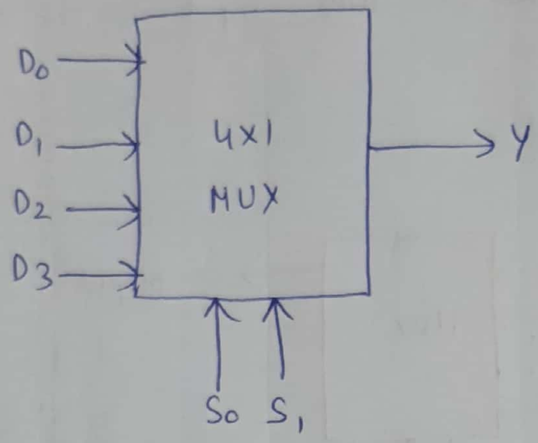
S	output
0	D_0
1	D_1

$4 = 2^{2}$ — Select lines.

4x1 Mux :-

It has 4 inputs , 2 Select lines and one output .

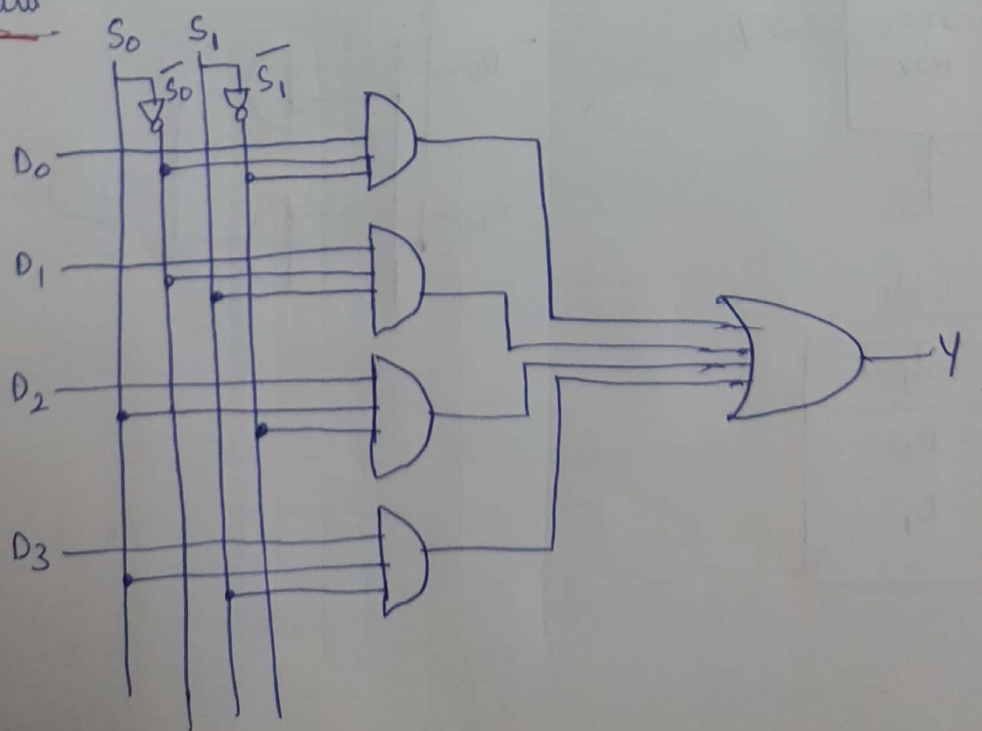
Block diagram .



Truth Table

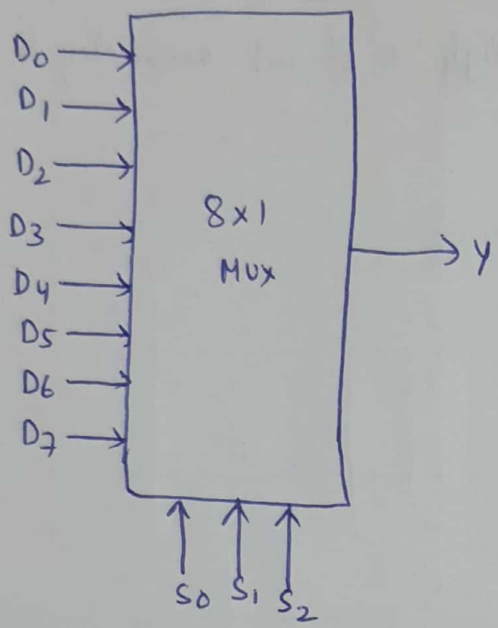
Inputs		outputs
S_0	S_1	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

logic circuit



8x1 MUX : It has 8 inputs, 3 select lines and one output.

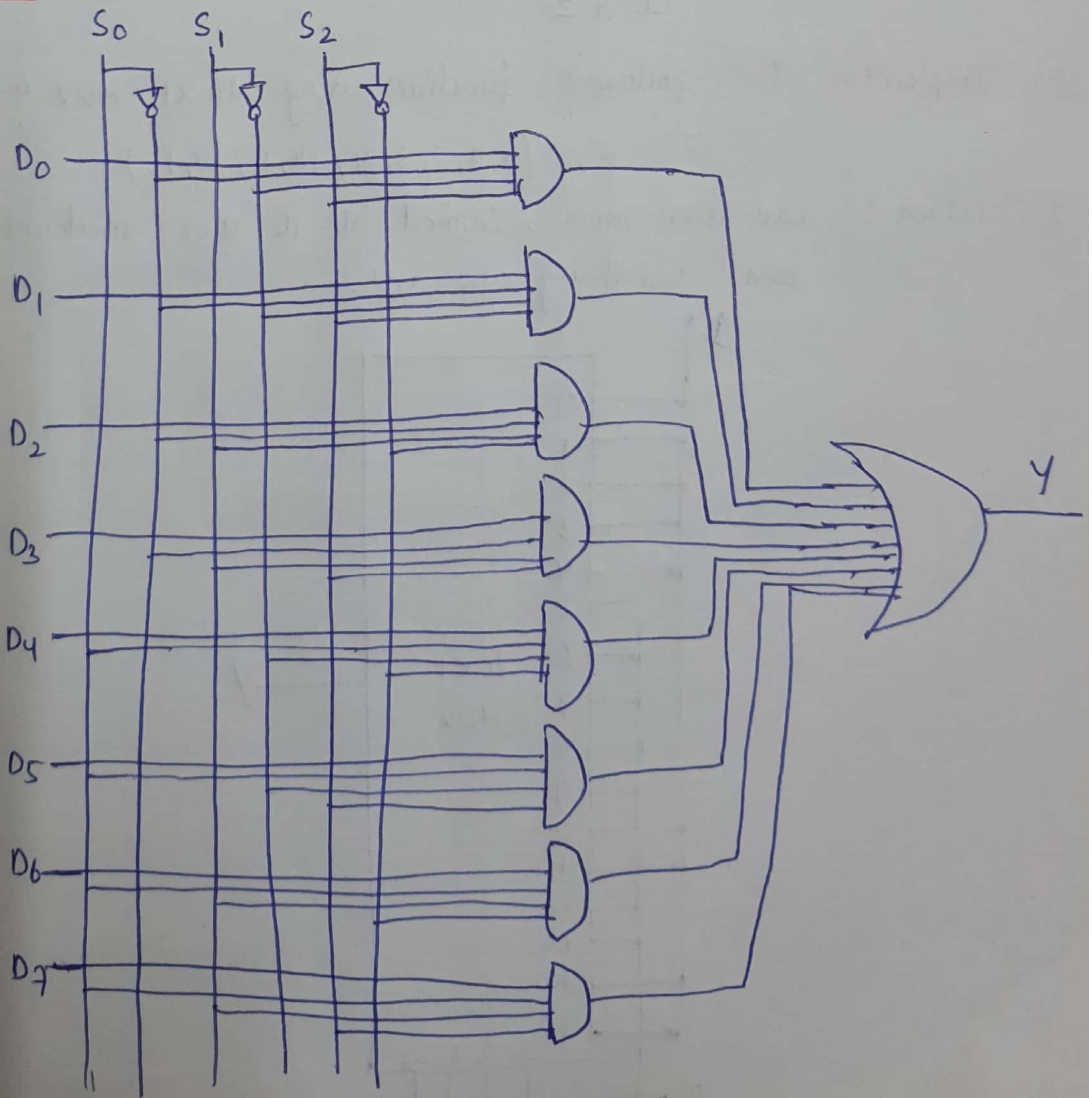
Block diagram.



Truth Table.

Inputs			Output
S_0	S_1	S_2	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Logic Circuit

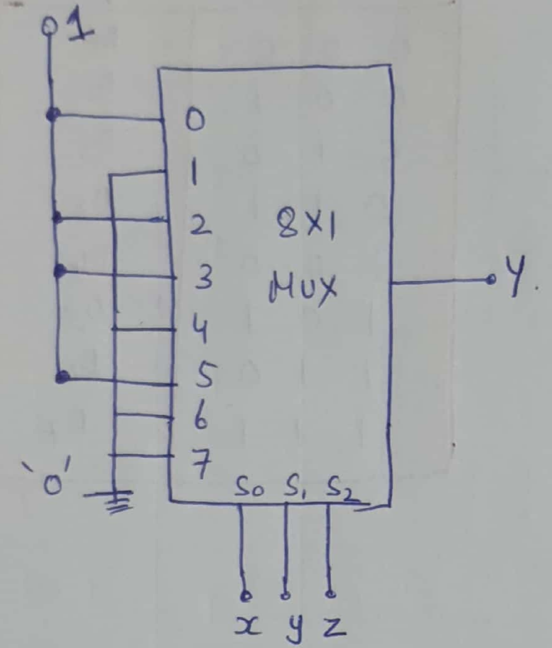


Scanned with CamScanner

Q. Use the MUX to implement the logic function

$$F(x,y,z) = \Sigma m(0,2,3,5)$$

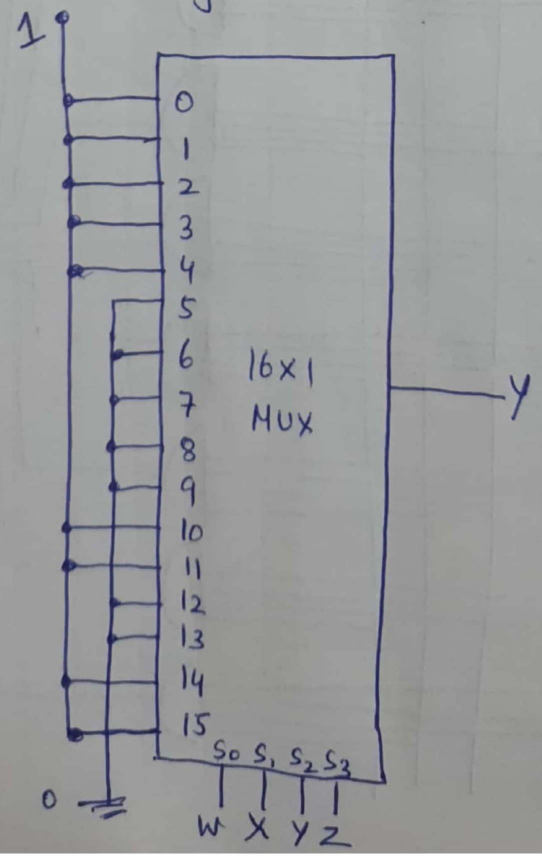
S. As the given function is 3 variable we need to use 8×1 mux. Connect all the given minterms to high i.e '1' and remaining to '0'.



Q. Implement the following function using 16×1 MUX?

$$F = \Sigma m(0,1,2,3,4,10,11,14,15).$$

S. Given :- use 16×1 mux, Connect all the given minterms to high '1' and remaining to '0'.



⑧ use 8×1 Mux to implement the logic function $f(A,B,C,D) = \Sigma m(0,1,2,3,4,10,11,14,15)$?

Sol

S_0 A	S_1 B	S_2 C	D	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
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	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

$F = 1 \rightarrow (D_0)$

$F = 1 \rightarrow (D_1)$

$F = \bar{D} - (D_2)$

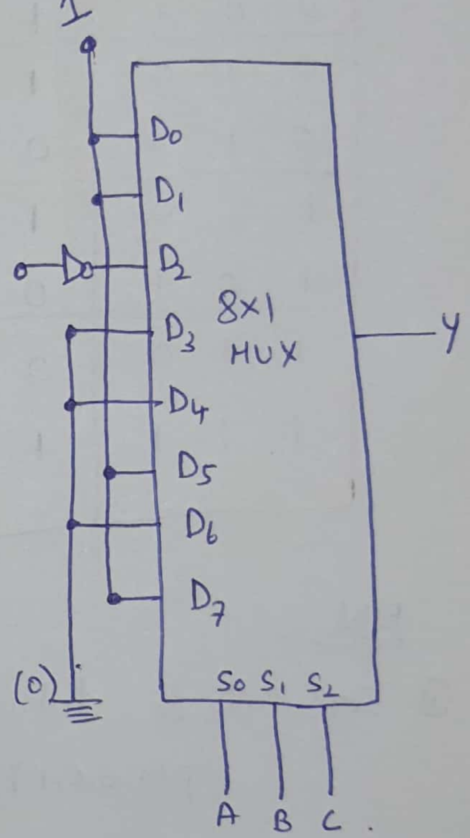
$F = 0 - (D_3)$

$F = 0 - (D_4)$

$F = 1 - (D_5)$

$F = 0 - (D_6)$

$F = 1 - (D_7)$



Q) Implement the logic function $f(A,B,C) = \sum m(1,2,4,7)$ by using 4×1 MUX?

S)

Truth Table

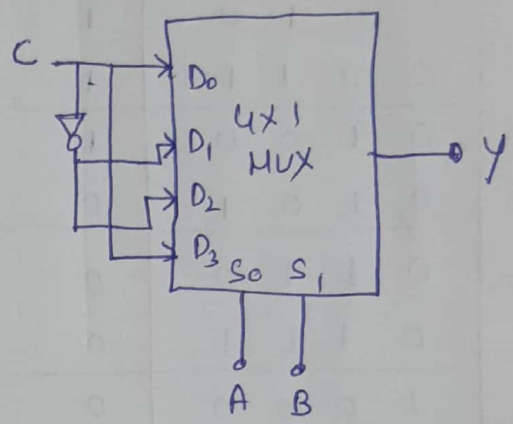
A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$f = c \ (D_0)$

$f = \bar{c} \ (D_1)$

$f = \bar{c} \ (D_2)$

$f = c \ (D_3)$



H.W

① Implement the following function using 8×1 MUX

$F(A,B,C,D) = \sum m(1,3,4,11,12,13,14,15)$

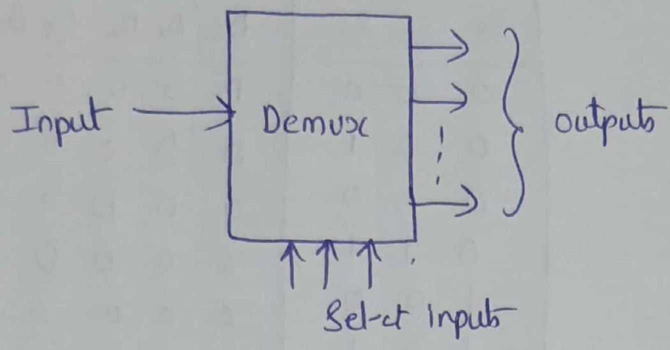
② Implement the following boolean function $F(a,b,c) = ab + \bar{b}c$ using 4:1 mux

③ Implement the boolean function

$F(A,B,C,D) = A\bar{B} + BD + \bar{B}C\bar{D}$ using 16×1 mux and 8×1 mux.

Demultiplexer :-

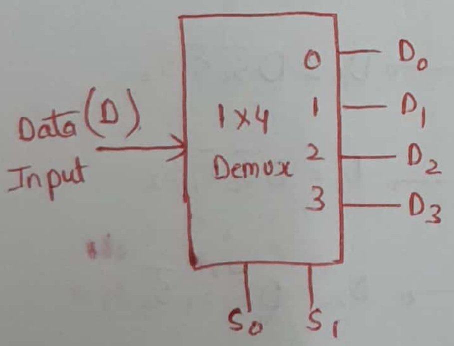
- → It is also called as Data Distributor.
- → It performs reverse operation of Multiplexer.
- → It takes single input and distribute to the several outputs



1 to 4 De-mux.

→ It has one input, two select line (S_0, S_1) and 4 outputs

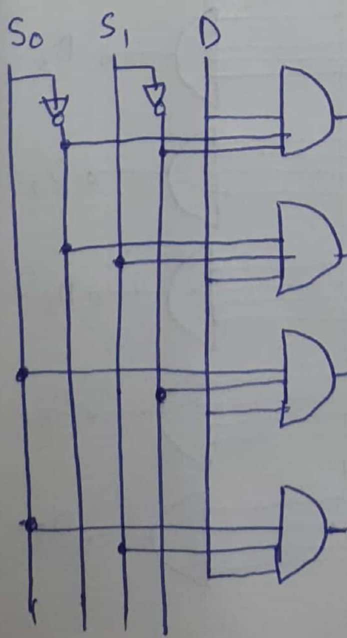
Block diagram.



Truth Table

Inputs		Outputs			
S_0	S_1	D_0	D_1	D_2	D_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

logic Circuit



$D_0 = D\bar{S_0}\bar{S_1}$

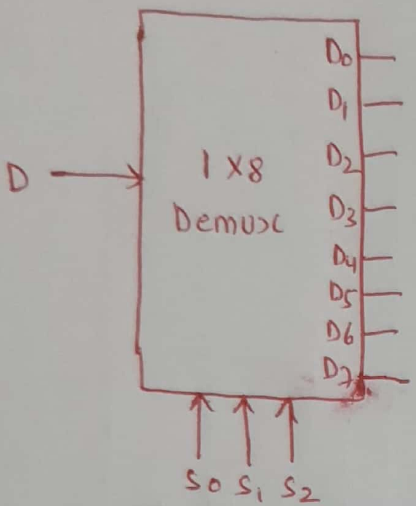
$D_1 = D\bar{S_0}S_1$

$D_2 = DS_0\bar{S_1}$

$D_3 = DS_0S_1$

1×8 Demux :- It has one input 'D', eight outputs ($D_0 - D_7$), three select inputs (S_0, S_1, S_2)

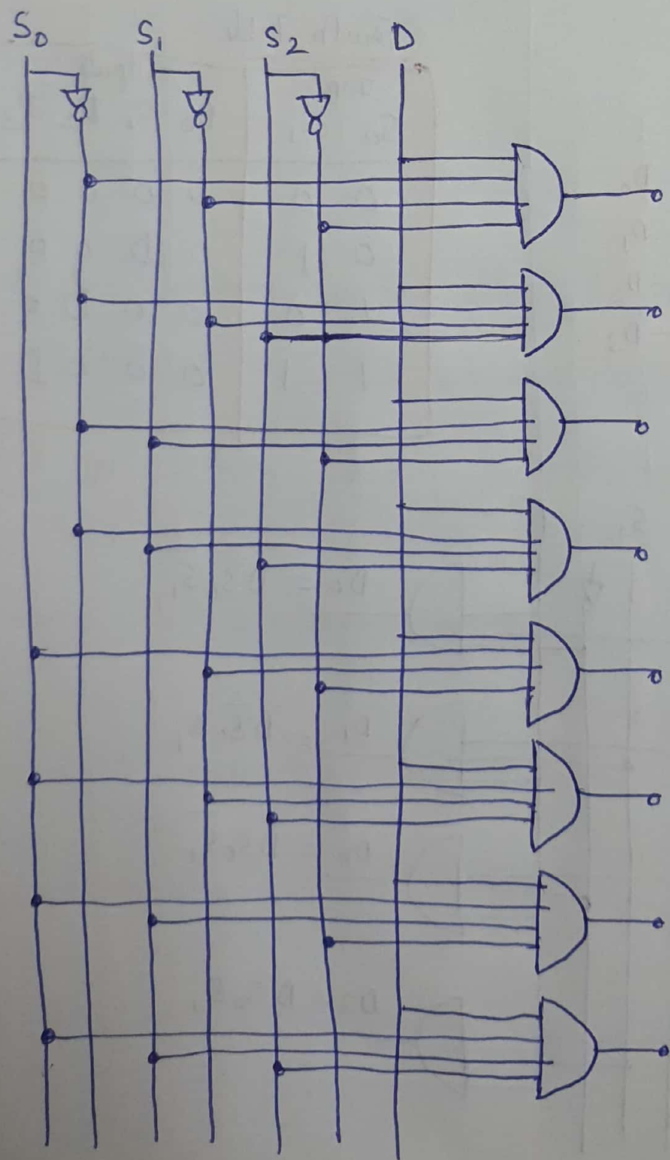
Block diagram



Truth Table

Inputs			outputs							
S_0	S_1	S_2	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	D	0	0	0	0	0	0	0
0	0	1	0	D	0	0	0	0	0	0
0	1	0	0	0	D	0	0	0	0	0
0	1	1	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0
1	0	1	0	0	0	0	0	D	0	0
1	1	0	0	0	0	0	0	0	D	0
1	1	1	0	0	0	0	0	0	0	D

Logic circuit



$D_0 = D\bar{S_0}\bar{S_1}\bar{S_2}$

$D_1 = D\bar{S_0}\bar{S_1}S_2$

$D_2 = D\bar{S_0}S_1\bar{S_2}$

$D_3 = D\bar{S_0}S_1S_2$

$D_4 = DS_0\bar{S_1}\bar{S_2}$

$D_5 = DS_0\bar{S_1}S_2$

$D_6 = DS_0S_1\bar{S_2}$

$D_7 = DS_0S_1S_2$

⑤ Implement 16×1 MUX using two 8×1 Mux?

Ⓢ