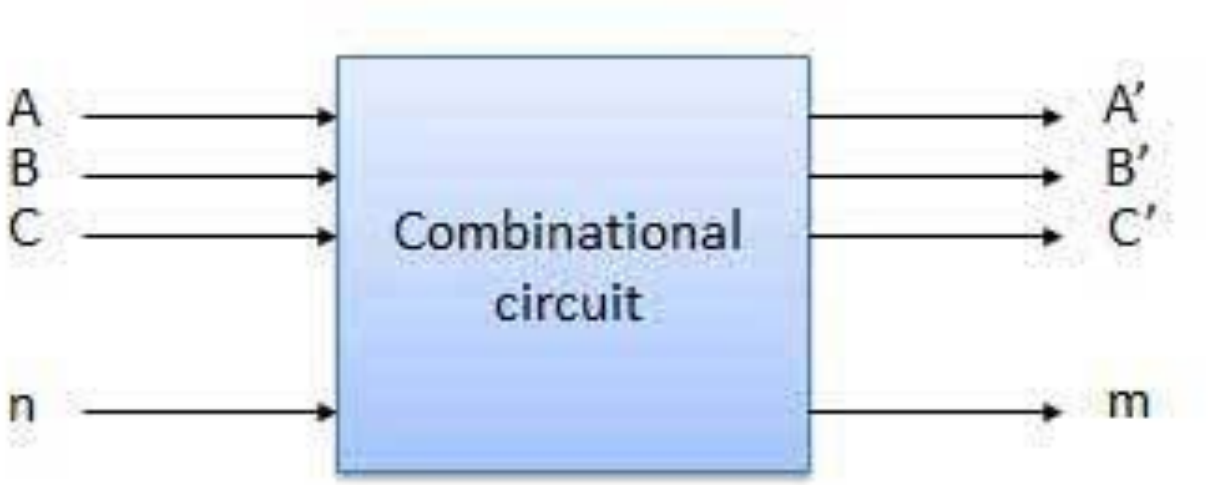
4	**Combinational Logic:** Different types of Codes:- BCD, excess-3, Gray code , binary code and their conversion BCD addition and subtraction circuits: - half- adder full adder half subtractor full subtractor BCD adder using IC7483 look ahead and carry parity generator and checker using 74180 magnitude comparator using 7485 Multiplexers (MUX):- working of MUX implementation of expression using MUX (IC 74153 74151) Demultiplexers (DEMUX):- implementation of expression using DEMUX decoder (IC 74138)	**10**	**10**

Unit 4: Combinational Logic Circuits

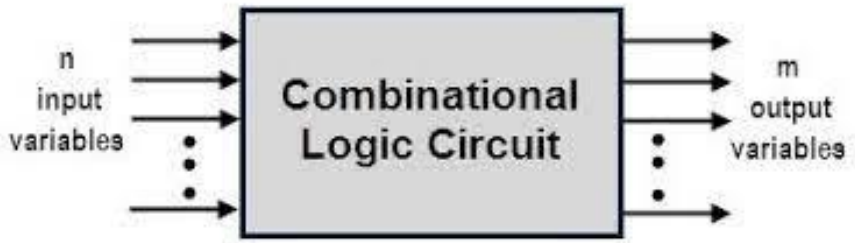


Introduction

1. The logic operations and Boolean algebra have already been discussed earlier
2. The Boolean theorems and De-Morgan's theorems are useful in manipulating the logic expressions. We can then realize the logical expressions using gates.
3. The number of logic gates required for the realization of a logical expression should be reduced to the minimum possible value.
4. This is possible if we can simplify the logical expressions. In this chapter we will discuss one of the simplification techniques called Karnaugh map or K-map.

Combinational circuits :

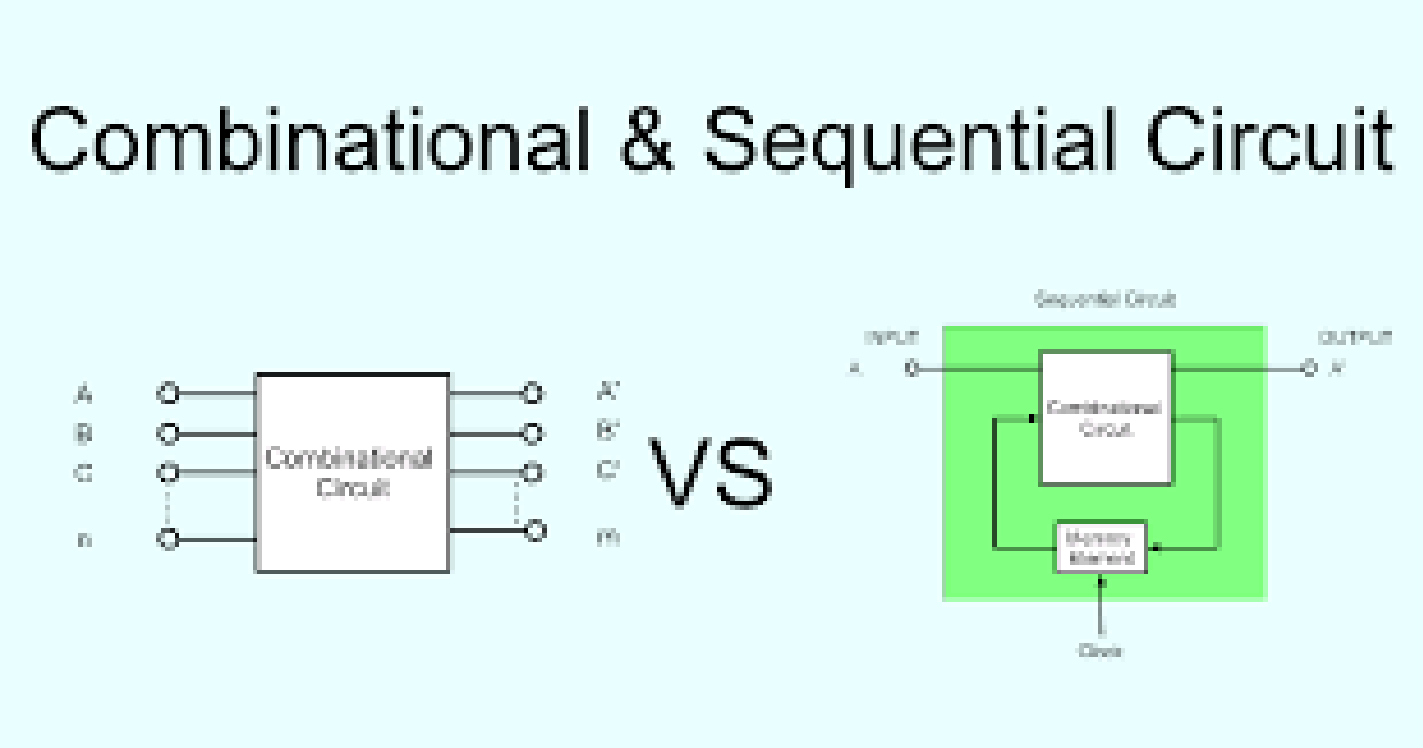
1. A combinational circuit is a **logic** circuit, the output of which depends only on the combination of the inputs.
2. The output does not depend on the past value of inputs or outputs.
3. Hence combinational circuits do not require **any memory.**
4. The block diagram of a combinational circuit is shown in Fig.



Classification of Digital Circuits :

- The digital systems in general are classified into two categories namely:

1. Combinational logic circuits
2. Sequential logic circuits.



Examples of combinational circuits :

Following are the examples of some combinational circuits:

1. Adders, subtractors
2. Comparator
3. Code converters
4. Encoders, decoders
5. Multiplexers, demultiplexers

SOP and POS Representations for Logic Expressions :

- Consider that the logic expression given to us is as follows:

$$Y = AC + BC$$

Then it can be realized using basic gates as shown in Fig

In this Boolean expression, Y is the result or output and A, B,C are called as **literals.**

Any logic expression can be expressed in the following two standard forms:

1. **Sum-of-products form (SOP) and**

2 Product-of-sums form (POS)

These two forms are suitable for **reducing the given logic expression to its simplest form.**

1. **Sum-of-products form (SOP)**

Y =ABC+BCD+ABD

A = XY+XY+XỸ

Y = PO+POR + POR

2 Product-of-Sums form (POS)

Y = (A+B+C).(A + B).(A + C)

A = (X+Y). (X+Y+Z)

Y = (P+R).(P+Q). (P+R)

Z=(A+B).(C+D)

- **Standard Form-In** this form each term may contain one, two or any number of literals. It is not necessary that each term should contain all the literals
- **Canonical Form-This** rule states that each term used in a equation **must contain all the available input variables.**

Sr.No	Expression	Type
1	Y=AB+ ABC + ABC	
2	$Y=AB+\overline{AB}+\overline{A}B$	
3	Y=(A+B).(A+B).(A+B)	
4	Y=(A+B).(A+B+C)	

Concept of Minterm and Maxterm

- **Minterm-** Each individual term in the canonical SOP Form is called as Minterm
- **Maxterm-** Each individual term in the canonical POS form is called as maxterm.
- Canonical SOP –$Y=A\bar{B}\bar{C}+A\bar{B}C+\bar{A}\bar{B}C$ each individual term **is minterm**
- m0 m1 m2
- Canonical POS-$Y=(A+B).(A+\bar{B})$
- M0 M1

Conversion from Standard SOP to Canonical SOP form :

- Steps to be followed:

Step 1: For each term in the given standard SOP expression find the missing literal.

Step 2: Then AND this term with the term formed by ORing the missing literal and its complement.

Step 3: Simplify the expression to get the canonical SOP expression

Convert the expression Y = AB + $\overline{AC}$ + BC into the canonical SOP form :

Soln. :

Step 1: Find the missing literal for each term :

Y= AB+ $\overline{AC}$ + BC

Missing literal → C B A

Step 2: AND each term with (Missing literal + Its C(complement) :

Y=AB.(C+$\overline{C}$) + $\overline{AC}$.(B+$\overline{B}$) + BC.(A + $\overline{A}$)

(Missing literal + Its complement)

Original product term

Step 3: Simplify the expression to get the canonical SOP :

Y = AB (C+$\overline{C}$) + $\overline{AC}$ (B+$\overline{B}$) + BC (A+$\overline{A}$)

= ABC + ABC + ABC + ABC + ABC + ABC

= (ABC + ABC) +(ABC+ AB C) + ABC + AB C

But A +A=A & (ABC + ABC) = ABC and (ABC + $\overline{ABC}$) = $\overline{ABC}$

:: Y = ABC + $\overline{ABC}$ + $\overline{ABC}$ + $\overline{ABC}$ Canonical SOP form

Each term contains all the literals

Soln. :

1. Given expression is,

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

$$\therefore \quad Y = \bar{A}\,(B + \bar{B})\,(C + \bar{C})\,(D + \bar{D}) + B\bar{C}\,\bar{D}\,(A + \bar{A})$$

$$= \bar{A}\,BCD + \bar{A}\,BC\,\bar{D} + \bar{A}\,B\,\bar{C}\,D + \bar{A}\,B\,\bar{C}\,\bar{D}$$

$$+ \bar{A}\,\bar{B}\,CD + \bar{A}\,\bar{B}\,C\,\bar{D} + \bar{A}\,\bar{B}\,\bar{C}\,D + \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$$

$$+ AB\,\bar{C}\,\bar{D} + \bar{A}\,B\,\bar{C}\,\bar{D}$$

$$\therefore \quad Y = \bar{A}\,BCD + \bar{A}\,BC\,\bar{D} + \bar{A}\,B\,\bar{C}\,D + \bar{A}\,B\,\bar{C}\,\bar{D} + \bar{A}\,\bar{B}\,CD$$

$$+ \bar{A}\,\bar{B}\,C\,\bar{D} + \bar{A}\,\bar{B}\,\bar{C}\,D + \bar{A}\,\bar{B}\,\bar{C}\,\bar{D} + AB\,\bar{C}\,\bar{D}$$

$$\ldots (\because A + A = A)$$

This is the required expression in the standard SOP form.

1. Convert following equation into standard SOP form (canonical) :
$Y = \bar{A} + BC\bar{D}$

2. Convert following equation into standard POS form (canonical) :
$Y = (\bar{A} + B)(B + \bar{C})(\bar{A} + \bar{C})$

Soln. :

1. Given expression is,

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

$$\therefore Y = \bar{A}(B + \bar{B})(C + \bar{C})(D + \bar{D}) + B\bar{C}\bar{D}(A + \bar{A})$$

$$= \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}CD$$
$$+ \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}$$
$$+ AB\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D}$$

$$\therefore Y = \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}B\bar{C}D$$
$$+ \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D$$
$$+ \bar{A}\bar{B}\bar{C}\bar{D} + AB\bar{C}\bar{D} \quad \ldots (\because A + A = A)$$

This is the required expression in the canonical SOP form.

2. $$Y = (\bar{A} + B)(B + \bar{C})(\bar{A} + \bar{C})$$

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

But $A + BC = (A + B)(A + C)$

$$\therefore Y = (\bar{A} + B + C)(\bar{A} + B + \bar{C}) \cdot (B + \bar{C} + A)(B + \bar{C} + \bar{A})$$
$$\cdot (\bar{A} + \bar{C} + B)(\bar{A} + \bar{C} + \bar{B})$$

$$\therefore Y = (\bar{A} + B + C)(\bar{A} + B + \bar{C}) \cdot (A + B + \bar{C})(\bar{A} + B + \bar{C})$$
$$\cdot (\bar{A} + B + C)(\bar{A} + B + \bar{C}) \quad \ldots \text{Ans.}$$

Ex. 4.2.6 : Convert following equation to canonical SOP form : $Y = (A + B\bar{C})(B + AC)$ **S-09, 2 Marks**

Soln. :

$$Y = (A + B\bar{C})(B + AC)$$

$$= AB + AAC + BB\bar{C} + ABC\bar{C}$$

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

$[\because A \cdot A = A, B \cdot B = B \text{ and } C \cdot \bar{C} = 0]$

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

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

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

This is required canonical SOP form.

Ex. 4.2.7 : Convert following expression into canonical SOP form : $Y = A + B$ **W-09, 2 Marks**

Soln. :

$$Y = A + B$$

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

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

This is canonical SOP form.

Conversion from standard POS to canonical POS form :

Steps to be followed:

Step 1: For each term find the missing literal

Step 2: Then OR each term with the term formed by ANDing the missing literal in that term with its complement.

Step 3: Simplify the expression to get the canonical POS

Ex. 4.2.8 : Convert the following into canonical SOP :

1. $AC + CD + BC$ 2. $\bar{A}(B + \bar{C})$

W-12, 4 Marks

Soln. :

1. $AC + CD + BC$:

$$\therefore Y = AC(B + \bar{B})(D + \bar{D}) + CD(A + \bar{A})(B + \bar{B}) + BC(A + \bar{A})(D + \bar{D})$$

$$= ABCD + ABC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}CD + ABCD + \bar{A}BCD + A\bar{B}CD + \bar{A}\bar{B}CD + ABCD + \bar{A}BCD + ABC\bar{D} + \bar{A}BC\bar{D}$$

$$= ABCD + ABC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}CD + \bar{A}BCD + \bar{A}\bar{B}CD + \bar{A}BC\bar{D}$$

2. $\bar{A}(B + \bar{C})$:

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

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

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

Convert the following SOP equation into standard SOP equation.

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

S-15, 2 Marks

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

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

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

$$Y = ABC + AB\bar{C} + \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$ ...Ans

Ex. 4.2.11 : Convert the following expression into its standard forms :

1. $Y = \bar{A}BC + AC + \bar{B}$
2. $Y = (B + \bar{C}) \times (A + D) \times (\bar{B} + \bar{D})$

W-16, 4 Marks

Soln. :

1. $Y = \bar{A}BC + AC + \bar{B}$

$= \bar{A}BC + AC(B + \bar{B}) + \bar{B}(A + \bar{A})(C + \bar{C})$

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

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

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

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

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

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

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

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

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

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

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

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

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

Ex. 4.4.3 : Convert the following boolean expression into its standard forms :

1. $Y = A\bar{B} + AC + \bar{B}C$
2. $Y = (A + \bar{B}) \cdot (A + C) \cdot (B + \bar{C})$

S-14, 4 Marks

Soln. :

1. $Y = A\bar{B} + AC + \bar{B}C$

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

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

$Y = ABC + AB\bar{C} + A\bar{B}C + \bar{A}\,\bar{B}C$

...($\because ABC + ABC = ABC, A\bar{B}C + A\bar{B}C = A\bar{B}C$)

This is required standard SOP form.

2. $Y = (A + \bar{B})(A + C)(B + \bar{C})$

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

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

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

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

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

...($\because (A + \bar{B} + C)(A + \bar{B} + C) = A + \bar{B} + C$)

This is required standard POS form.

Karnaugh-Map Simplification :

What is K-map ? Explain the rules to simplify

1. This is another simplification technique to reduce the Boolean equation.
2. It overcomes all the disadvantages of the algebraic simplification technique.
3. K-map (short form of Karnaugh map) is a graphical method of simplifying a Boolean equation.
4. K-map is a graphical chart made up of rectangular boxes.
5. The information contained in a truth table or available in the
6. SOP or POS form can be represented on a K-map. The K-map can be used for systematic simplification of Boolean expression.
7. K-maps can be written for 2, 3, 4 ... upto 6 variables. Beyond that the K-map technique becomes very cumbersome.
8. K-map is ideally suitable for designing the combinational logic circuits using either a SOP method or a POS method.

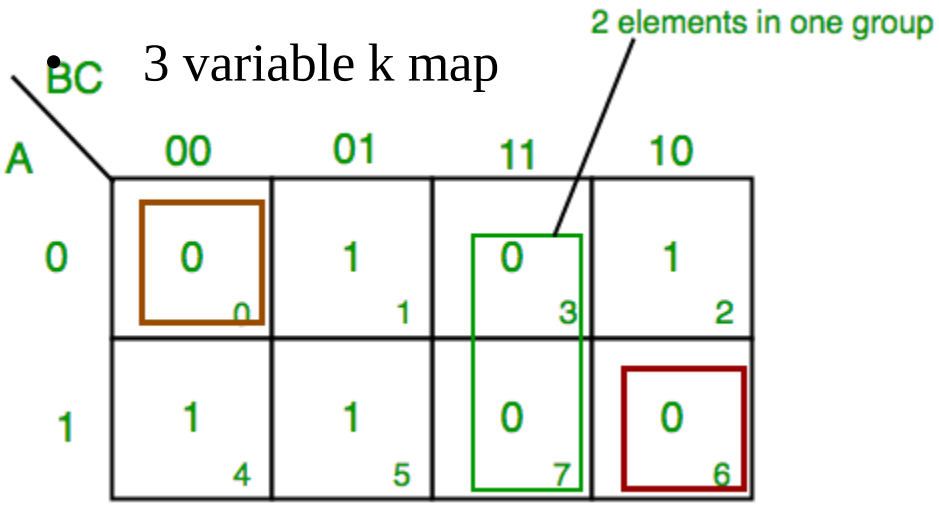
K-map Structure

- 2 variable k map

A. SOP: -

A \ B	$\overline{B}$ 0	B 1
$\overline{A}$ 0	$\overline{A}.\overline{B}$	$\overline{A}.B$
A 1	$A.\overline{B}$	$A.B$

- 3 variable k map



4 variable k map

A B \ CD	00	01	11	10
00				
01				
11				
10				

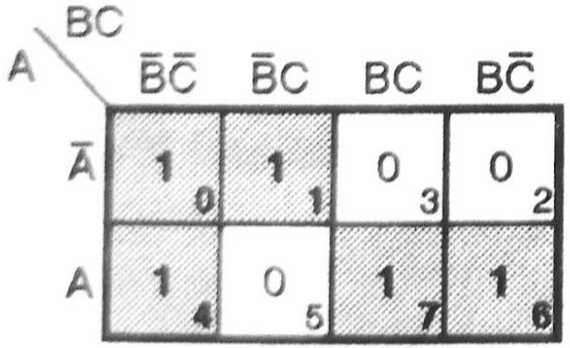
	C'D'	C'D	CD	CD'
A'B'	0	1	3	2
A'B	4	5	7	6
AB	12	13	15	14
AB'	8	9	11	10

Ex. 4.5.1 : Represent the equation given below on Karnaugh map :

$$Y = \bar{A}\,\bar{B}\bar{C} + \bar{A}\,\bar{B}\,C + A\bar{B}\bar{C} + AB\bar{C} + ABC.$$

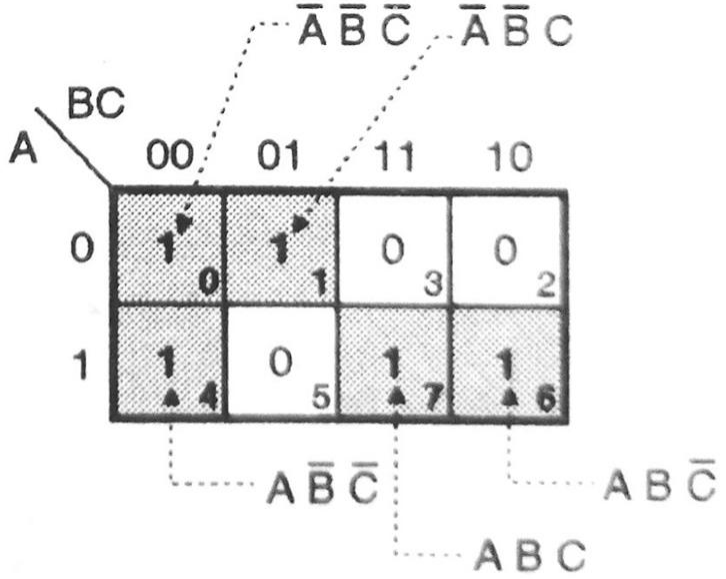
Soln. : The given expression is in the standard SOP form. Each term represents a minterm.

We have to enter 1's in the boxes corresponding to each minterm, as shown in Fig. P. 4.5.1.



"1"s correspond to the minterms in the given Boolean expression

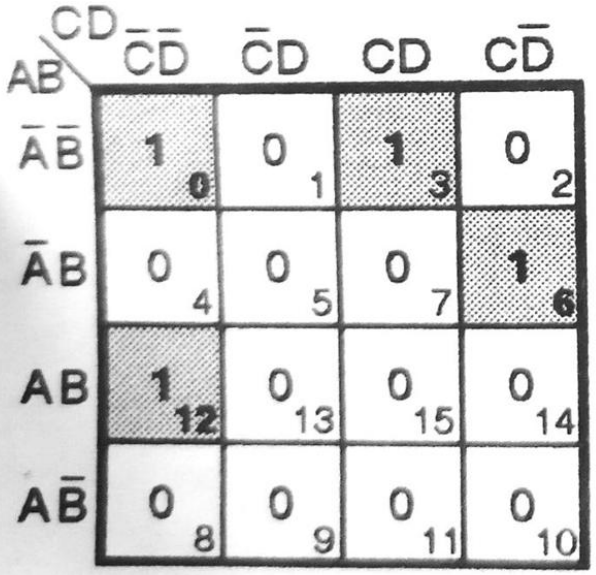
OR



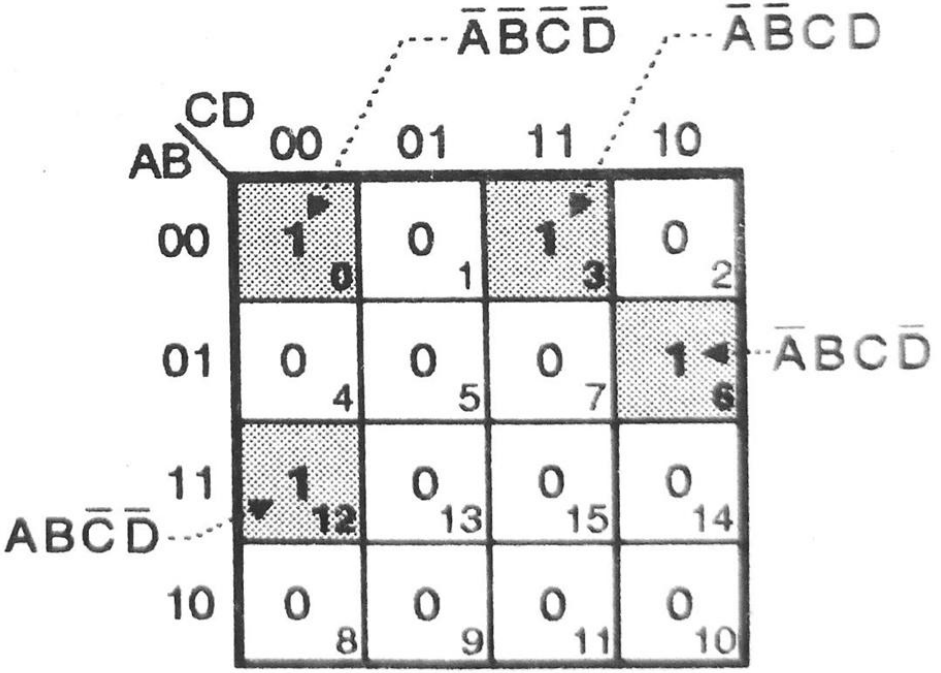
Ex. 4.5.2 : Plot the following Boolean expression on K-map :

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

Soln. : Refer Fig. P. 4.5.2.



"1"s correspond to the minterms in the given Boolean expression



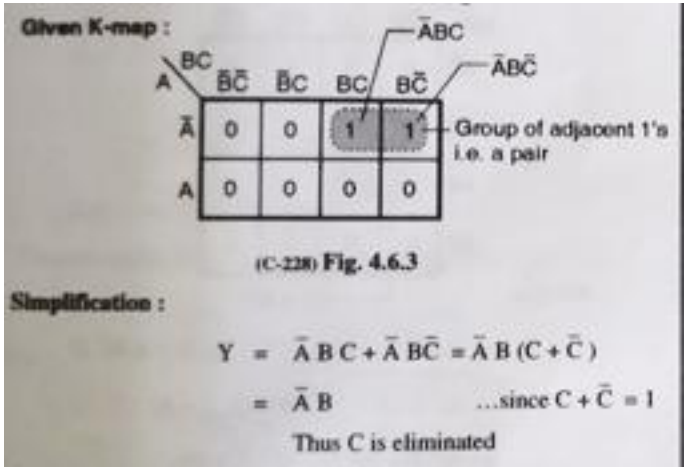
Way of Grouping (Pairs, Quads and Octets):

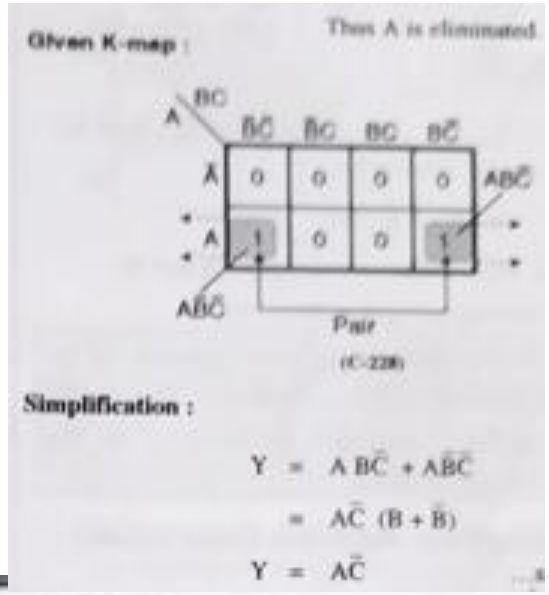
While grouping, we **should group most number of 1's** (or 0's).

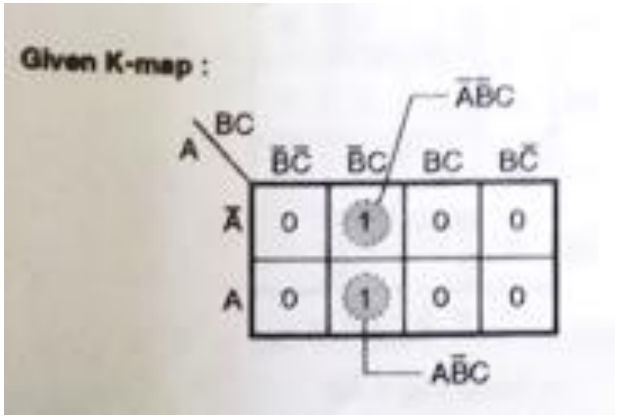
The grouping follows the binary rule i.e. we can group 1, 2, 4, 8, 16, 32 number of l's or 0's. We cannot group 3, 5, 7, ... number of 1's or 0's.

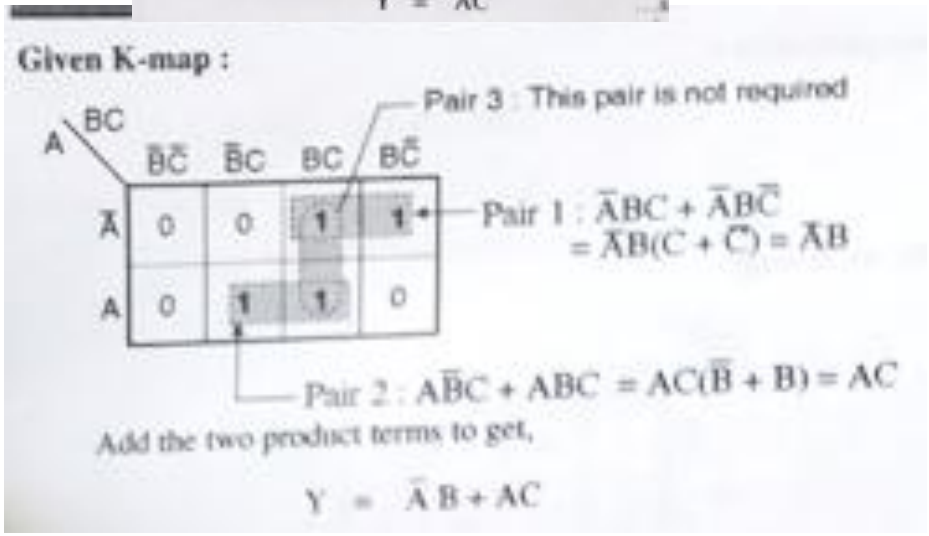
1. **Pairs:** A group of two adjacent 1's or 0's is called as a pair.
2. **Quad:** A group of four adjacent l's or 0's is called as a quad.
3. **Octet:** A group of eight adjacent l's or 0's is called as octet.

Grouping of 2 Adjacent Ones

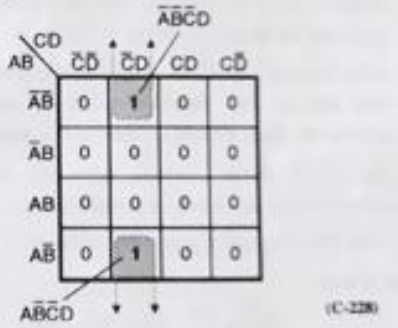








Given K-map :



(C-228)

Simplification :

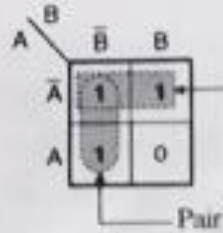
$$Y = \bar{A}\,\bar{B}\bar{C}\,D + A\bar{B}\bar{C}\,D = \bar{B}\bar{C}\,D\,(\bar{A} + A)$$

$$\therefore\ Y = \bar{B}\bar{C}\,D$$

Conclusion : A is eliminated.

Example of overlap :

Given K-map :



Pair 1 : $\bar{A}\bar{B} + \bar{A}B$
$= \bar{A}(\bar{B} + B) = \bar{A}$

Note that B is eliminated

Pair 2 : $\bar{A}\bar{B} + A\bar{B} = \bar{B}$

Note that A is eliminated

$\therefore$ Final expression $Y = \bar{A} + \bar{B}$

Note that in order to cover all the 1's, we have to overlap two pairs as shown.

Grouping of 4 Adjacent Ones

Given K-map :

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	0	0
$A\bar{B}$	0	1	0	0

$Y = \bar{C}D$

(C-231)

Simplification :

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

$$= \bar{C}\,D\,[\bar{A}\,\bar{B} + \bar{A}\,B + A\,B + A\bar{B}]$$

$$= \bar{C}\,D\,[\bar{A}\,(\bar{B} + B) + A\,(B + \bar{B})]$$

$$= \bar{C}\,D\,[\bar{A} + A] = \bar{C}\,D$$

Thus A and B are eliminated.

Given K-map :

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	0	0	0	0
$A\bar{B}$	1	1	1	1

$Y = A\bar{B}$

(C-230)

Simplification :

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

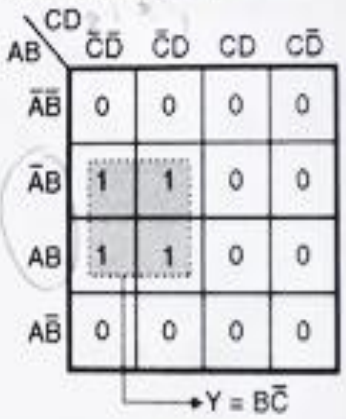
$$= A\bar{B}\,[\bar{C}\,\bar{D} + \bar{C}\,D + C\,D + C\,\bar{D}]$$

$$= A\bar{B}\,[\bar{C}\,(\bar{D} + D) + C\,(D + \bar{D})]$$

$$\therefore\ Y = A\bar{B}\,[\bar{C} + C] = A\bar{B}$$

Thus C and D are eliminated.

Four adjacent ones forming a square



(C-231(a))

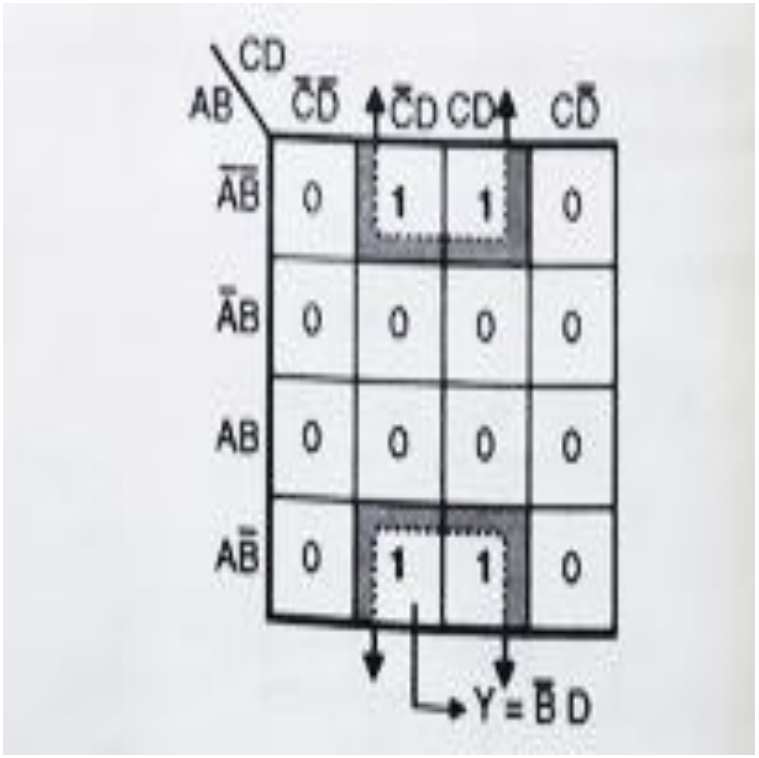
$$Y = \bar{A}B\bar{C}\bar{D} + AB\bar{C}\bar{D} + \bar{A}B\bar{C}D + AB\bar{C}D$$

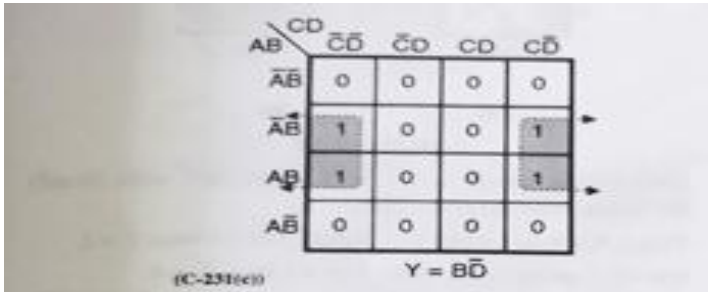
$$= B\bar{C}\bar{D}(\bar{A} + A) + B\bar{C}D(\bar{A} + A)$$

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

$$\therefore Y = B\bar{C}$$

Thus A and D are eliminated





(C-231(c))

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

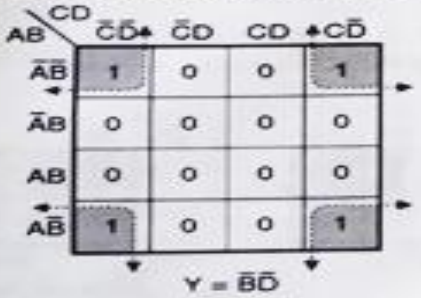
$$= B\bar{C}\bar{D}(\bar{A} + A) + BC\bar{D}(\bar{A} + A)$$

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

$$Y = B\bar{D}(\bar{C} + C) = B\bar{D}$$

Thus A and C are eliminated

1's corresponding to corners forming a Quad :



(C-232)

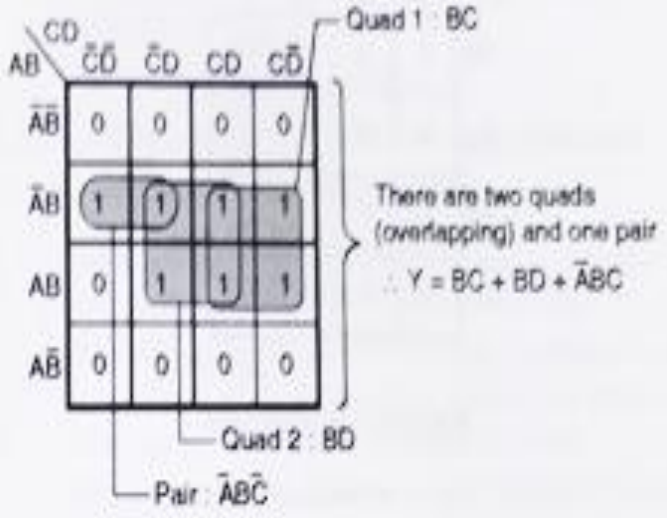
$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$

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

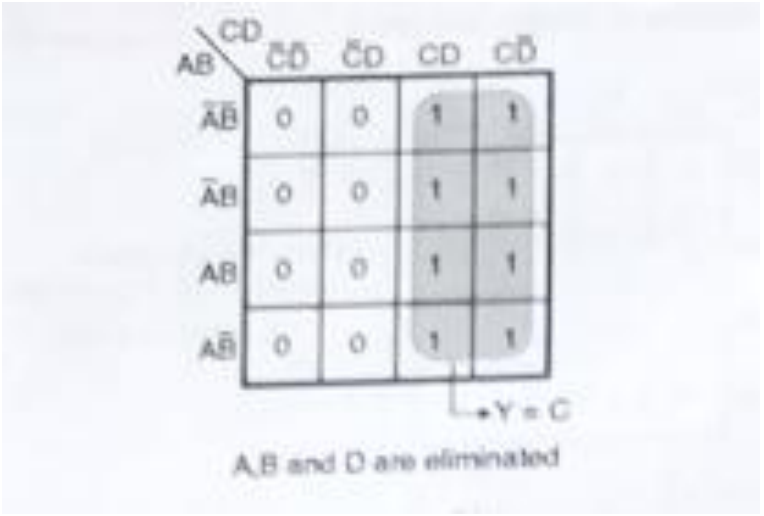
$$= \bar{B}\bar{C}\bar{D} + \bar{B}C\bar{D} = \bar{B}\bar{D}(\bar{C} + C)$$

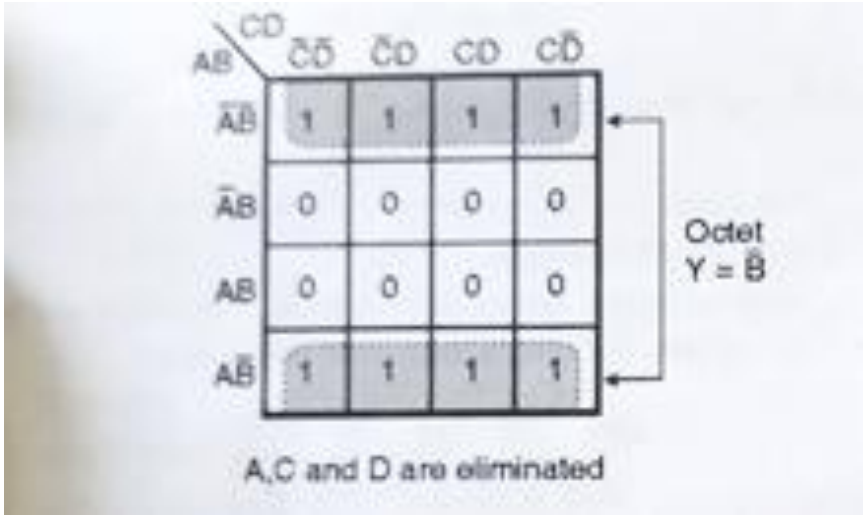
$$\therefore Y = \bar{B}\bar{D}$$ Thus A and C are eliminated

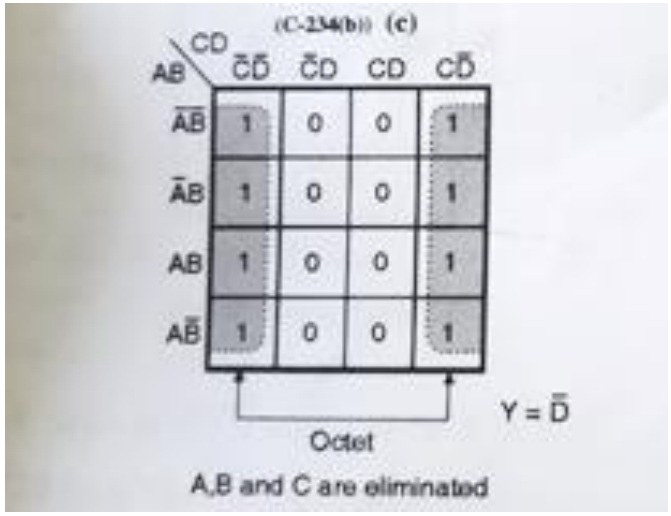
Overlapping of Quads and pairs

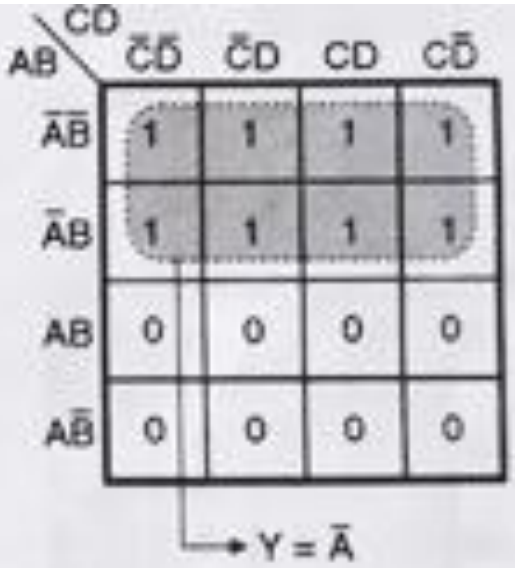


Grouping of 8 Adjacent Ones









B,C and D are eliminated

(C-234) **Fig. 4.6.6(a)**

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

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

$$\therefore Y = \bar{A}\,\bar{B}\bar{C}\,(\bar{D} + D) + \bar{A}\,\bar{B}C$$

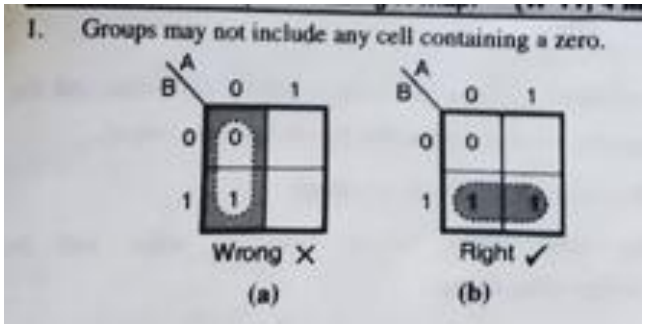
$$(D + \bar{D}) + \bar{A}\,B\,\bar{C}\,(\bar{D} + D) + \bar{A}\,B\,C\,(D + \bar{D})$$

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

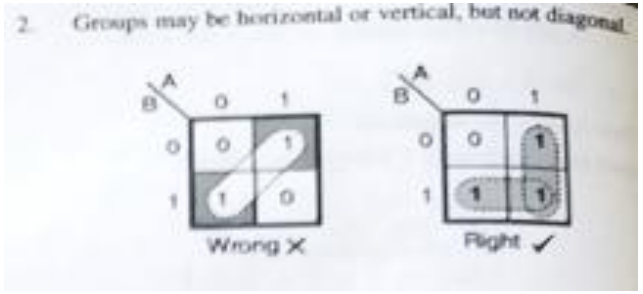
$$\therefore Y = \bar{A}\,(\bar{B} + B) = \bar{A}$$

Rules followed for K-Map Simplification

1. Groups may not include any cell containing a zero.



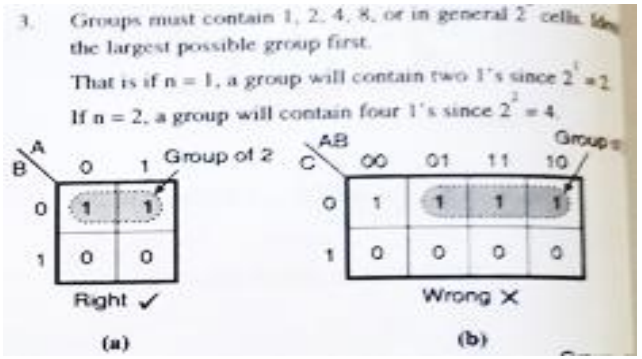
2. Groups may be horizontal or vertical, but not diagonal.



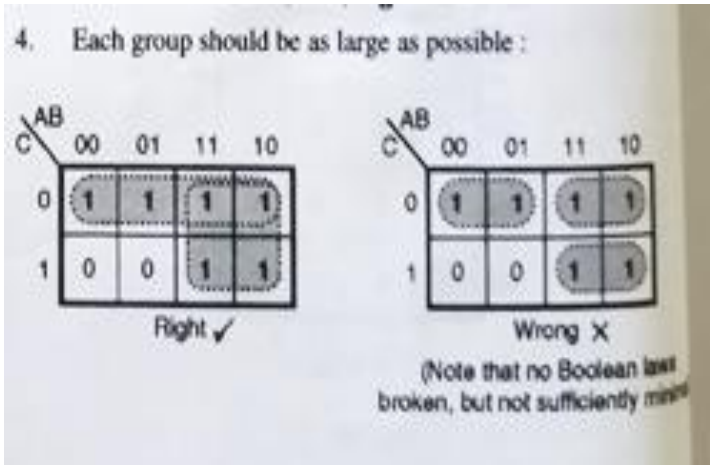
3. Groups must contain 1, 2, 4, 8, or in general 2^n cells. Ide... the largest possible group first.

That is if n = 1, a group will contain two 1's since $2^1 = 2$.

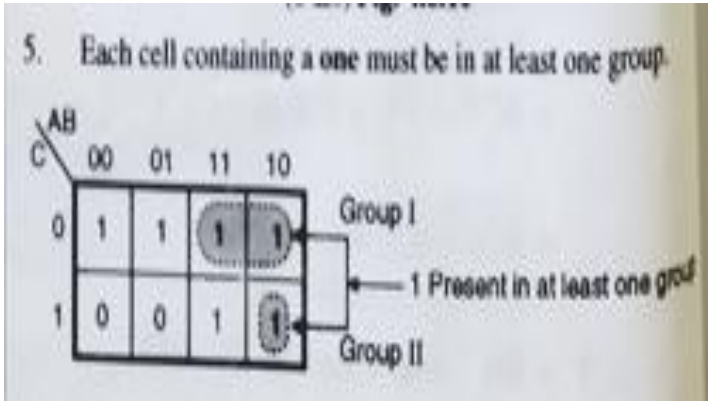
If n = 2, a group will contain four 1's since $2^2 = 4$.



4. Each group should be as large as possible :

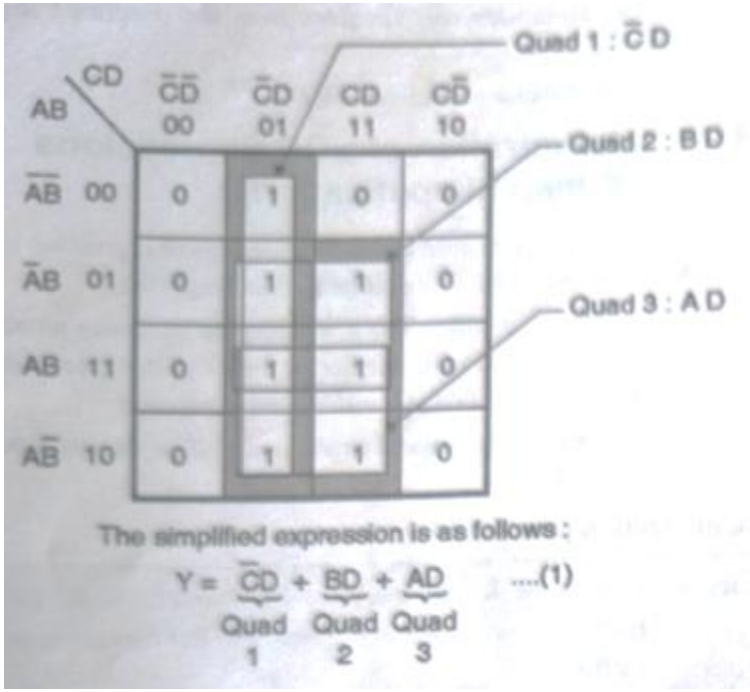


5. Each cell containing a one must be in at least one group.



1. For the logical expression given below draw the K-map and obtain the simplified logical expression:

Y = Σm (1, 5, 7, 9, 11, 13, 15). Realize the minimized expression using the basic gates.



Examples

2. Using K-map realize the following expression using minimum number of gates : $\Sigma m\ (1, 3, 4, 5, 7, 9, 11, 13, 15)$

3. Simplify the following expressions using K-map :

a. $f\ (A, B, C) = \Sigma m\ (0, 1, 3, 4, 6)$

b. $f\ (A, B, C, D) = \Sigma m\ (0, 1, 2, 4)$

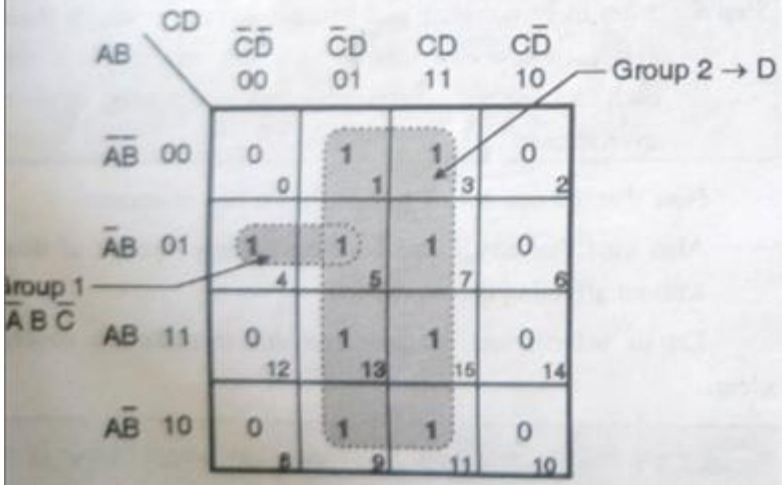
Ex. 4.7.2 : Using K-map realize the following expression using minimum number of gates :

$Y = \sum m\,(1, 3, 4, 5, 7, 9, 11, 13, 15)$

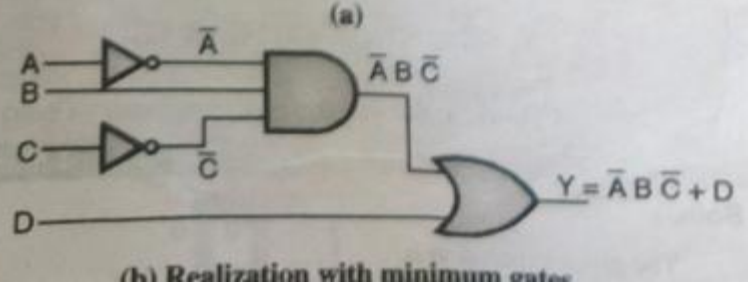
S-09. 4 Marks

Soln. :

The required K-map is shown in Fig. P. 4.7.2(a) and the realization using minimum gates is shown in Fig. P. 4.7.2(b).



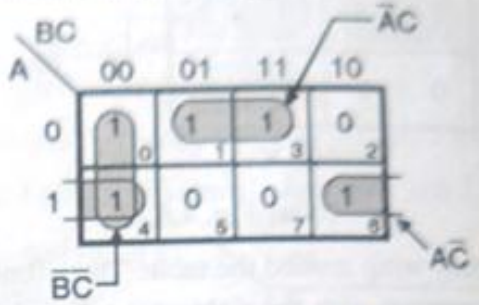
$\therefore Y = \bar{A}B\bar{C} + D$



(b) Realization with minimum gates

1. $f(A,B,C) = \Sigma m\,(0,1,3,4,6)$:

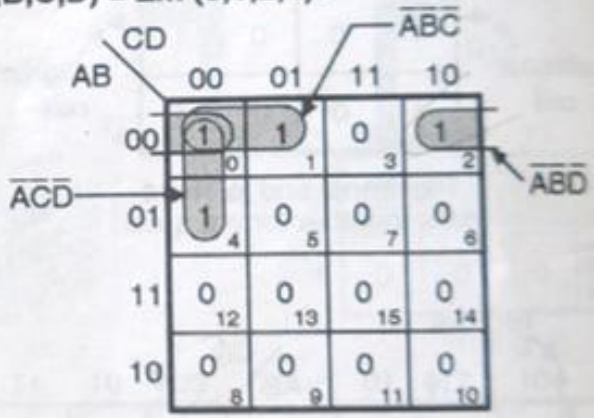
Simplification using K-map :



(C-2600) **Fig. P. 4.7.3(a)**

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

2. $f(A,B,C,D) = \Sigma m\,(0,1,2,4)$:



(C-2601) **Fig. P. 4.7.3(b)**

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

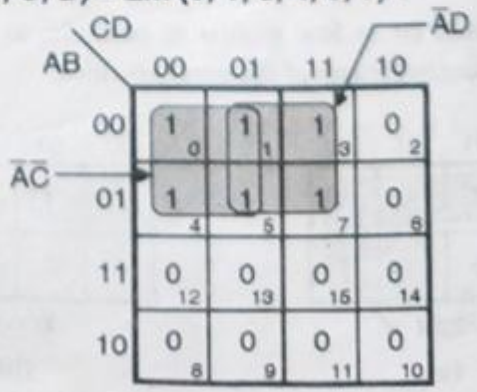
Ex. 4.7.4 : Solve the following SOP expression with K-map :

1. $f(A, B, C, D) = \Sigma m\ (0, 1, 3, 4, 5, 7)$
2. $f(A, B, C) = \Sigma m\ (0, 1, 4, 5, 6, 7)$

W-11, W-17, 4 Marks

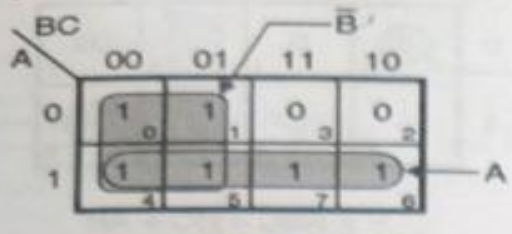
Soln. :

1. $f(A, B, C, D) = \Sigma m\ (0, 1, 3, 4, 5, 7)$:



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

2. $f(A, B, C) = \Sigma m\ (0, 1, 4, 5, 6, 7)$:



(C-2603) **Fig. P. 4.7.4(b)**

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

Ex. 4.7.5 : Convert the following functions into SOP form and plot the K map :
$Y = AB + AC + BC.$ **W-11**

Soln. :

$$Y = AB + AC + BC$$

$$= AB(C + \bar{C}) + AC(B + \bar{B}) + BC(A + \bar{A})$$

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

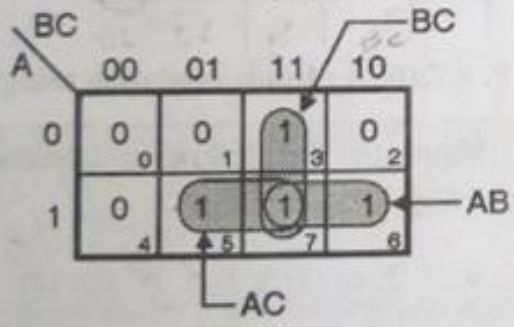
$$\therefore Y = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC$$

$$\therefore Y = \Sigma m\ (7,6,5,3)$$

This is the required standard SOP form (canonical

Simplification using K-map :

$$Y = \Sigma m\ (3,5,6,7)$$



(C-2604) **Fig. P. 4.7.5**

$$Y = AB + BC + AC$$

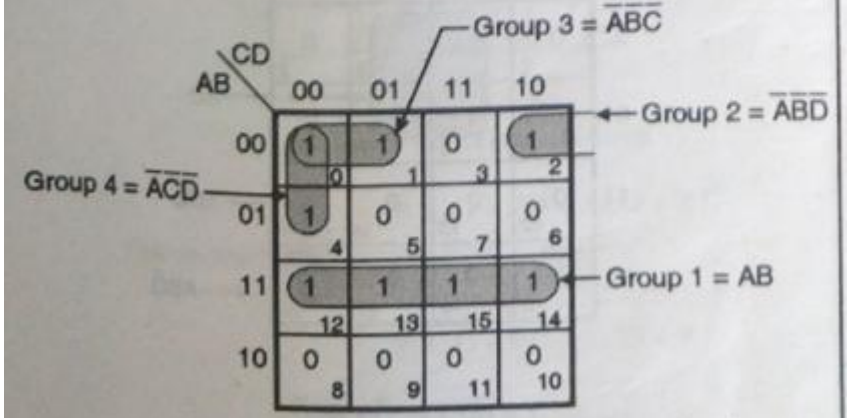
Ex. 4.7.6 : Solve the following Boolean expression using K-map :

$Y = \Sigma m (m_0, m_1, m_2, m_4, m_{12}, m_{13}, m_{14}, m_{15})$

S-12. 4 Marks

Soln. :

Simplification using K-map :



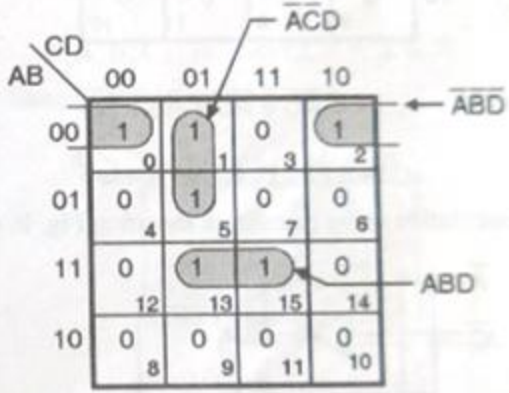
Ex. 4.7.8 : Simplify the following using K-map and realize using NAND – NAND gates.

$f(A, B, C, D) = \Sigma m (0, 1, 2, 5, 13, 15)$

W-12. 4 Marks

Soln. :

Step 1 : Minimization using K-map :



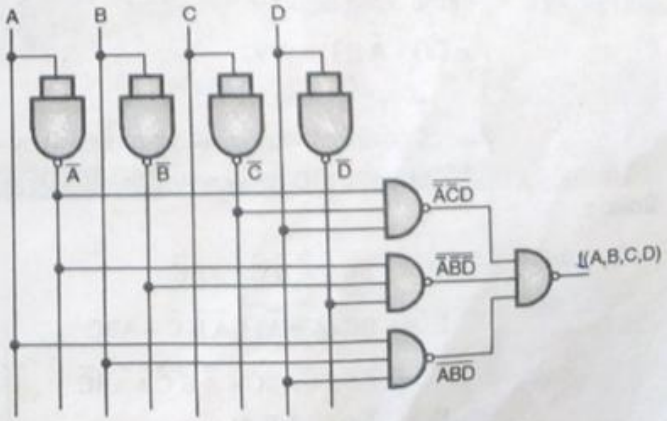
Step 2 : Realization using NAND gates :

$$f(A, B, C, D) = \overline{\overline{A}\,\overline{C}\,D + \overline{A}\,B\,\overline{D} + ABD}$$

...Take double inversion

$$= \overline{\overline{A}\,\overline{C}\,D \cdot \overline{A}\,B\,\overline{D} \cdot \overline{ABD}}$$

...using De Morgan's laws.



.7.9 : Minimize the following Boolean expression using K-map.

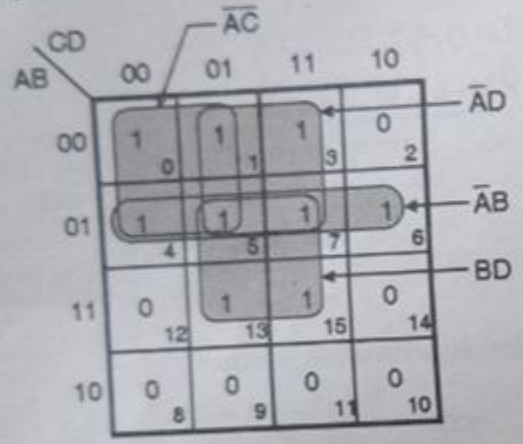
$Y = \Sigma m\ (0, 1, 3, 4, 5, 6, 7, 13, 15)$

Draw the logical circuits diagram of minimized expression using basic gates. **W-13, 4 Marks**

n. :

$$Y = \Sigma m\ (0, 1, 3, 4, 5, 6, 7, 13, 15)$$

The required K-map is as shown in Fig. P. 4.7.9.



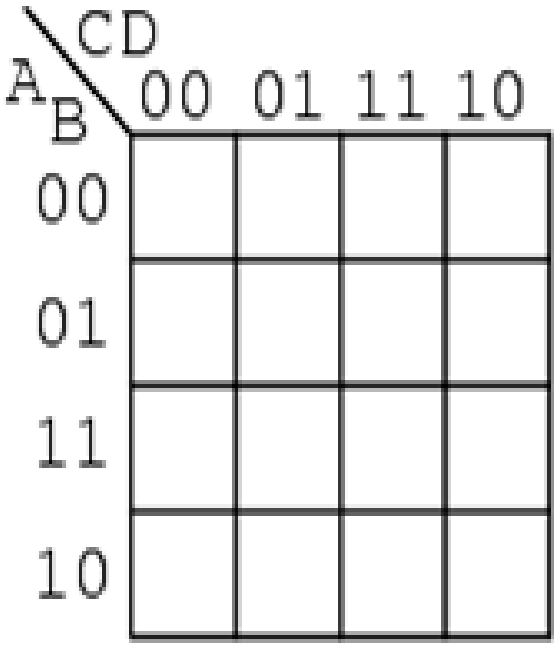
$\therefore\ Y = \bar{A}\bar{C} + \bar{A}D + \bar{A}B + BD$

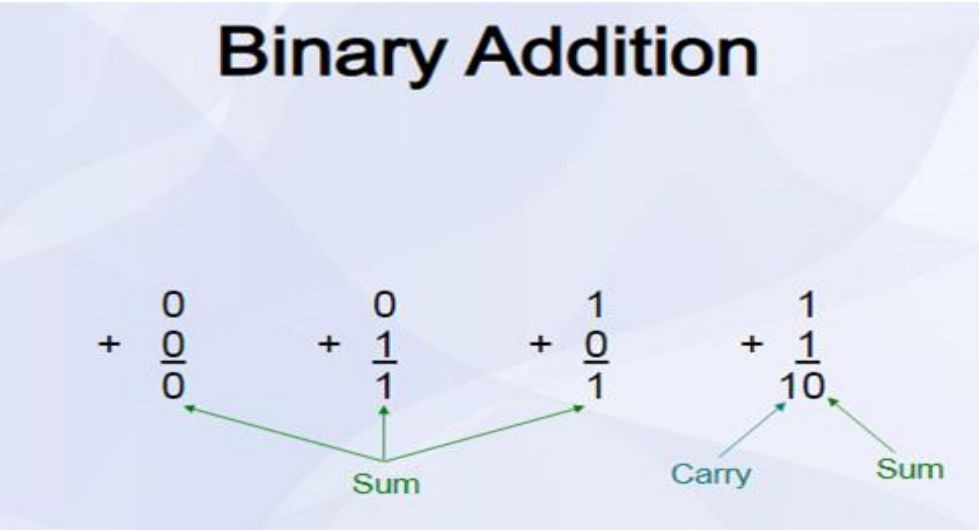
4.Minimize the following expressions using K-map :

a. $Y = \Sigma m\ (1,5,6,7,11,12,13,15)$

b. $F = \Sigma m\ (0,1,2,3,11,12,14,15)$

Soution:Simplification using K-Map is as shown below





Adders :

- Adders are important not only in the computer but also in many types of digital systems in which the numeric data are processed.
- **Types of adder:**
- **Half adder**
- Full adder

What is Adder?

- **Adder** : In electronics an adder is digital circuit that perform addition of numbers. In modern computer adder reside in the arithmetic logic unit (ALU).

Half Adder-

- Half Adder is a combinational logic circuit.
- It is used for the purpose of adding two single bit numbers.
- It contains 2 inputs and 2 outputs (sum and carry).

Step-01:

Identify the input and output variables-

- Input variables = A, B (either 0 or 1)
- Output variables = S, C where S = Sum and C = Carry

Step-02:

Draw the truth table-

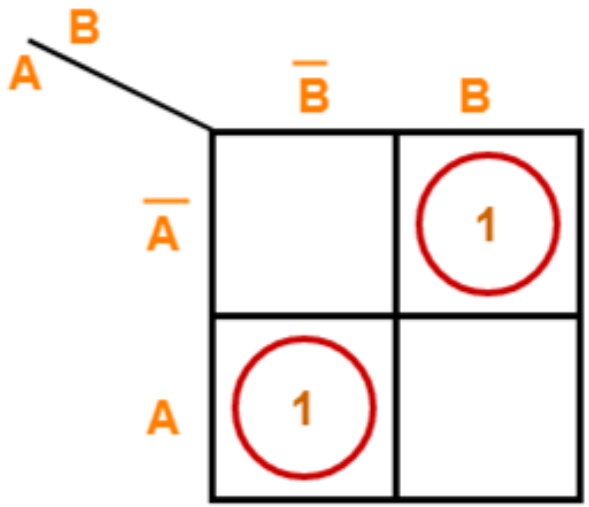
Inputs		Outputs	
A	B	C (Carry)	S (Sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Truth Table

Step-03:

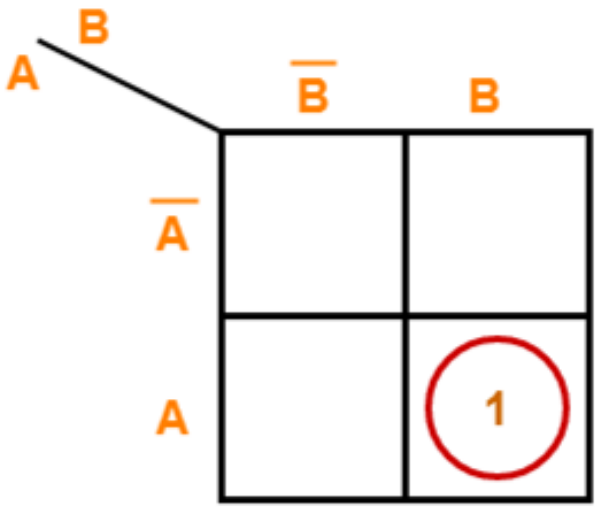
Draw K-maps using the above truth table and determine the simplified Boolean expressions-

For S:



$S = A \oplus B$

For C:

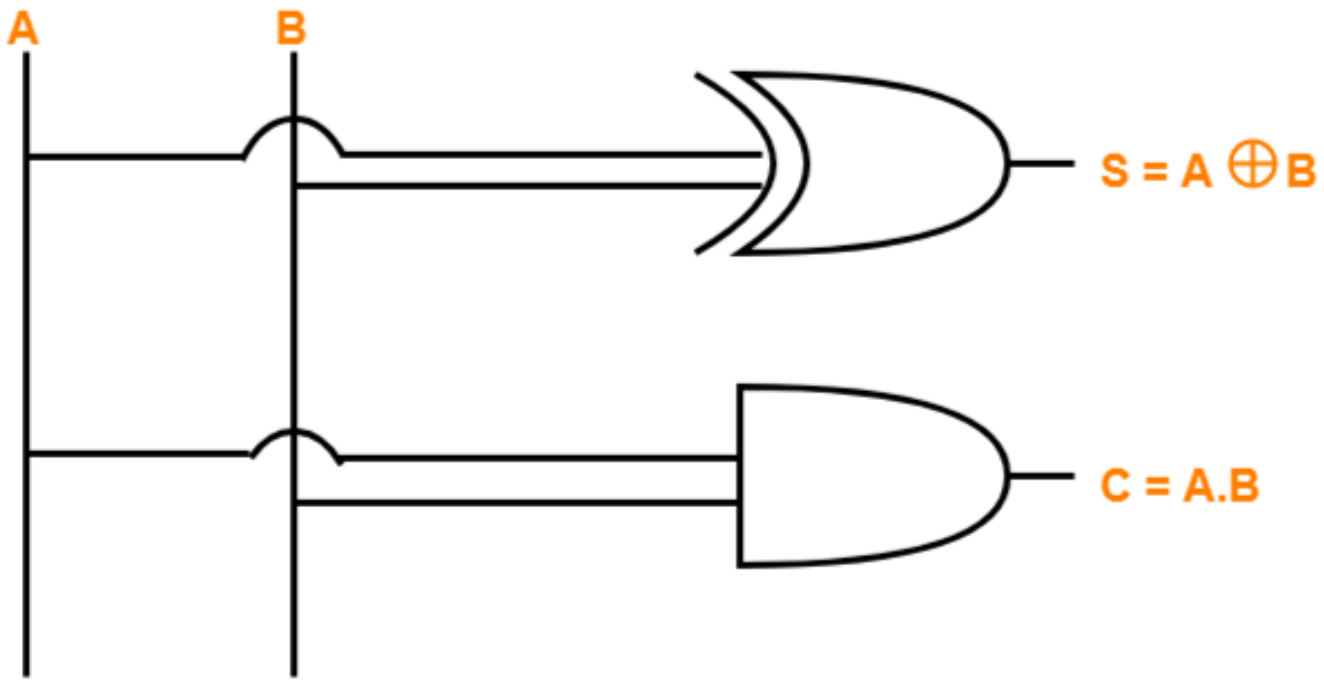


$C = A \cdot B$

Step-04:

Draw the logic diagram.

The implementation of half adder using 1 XOR gate and 1 AND gate is as shown below-



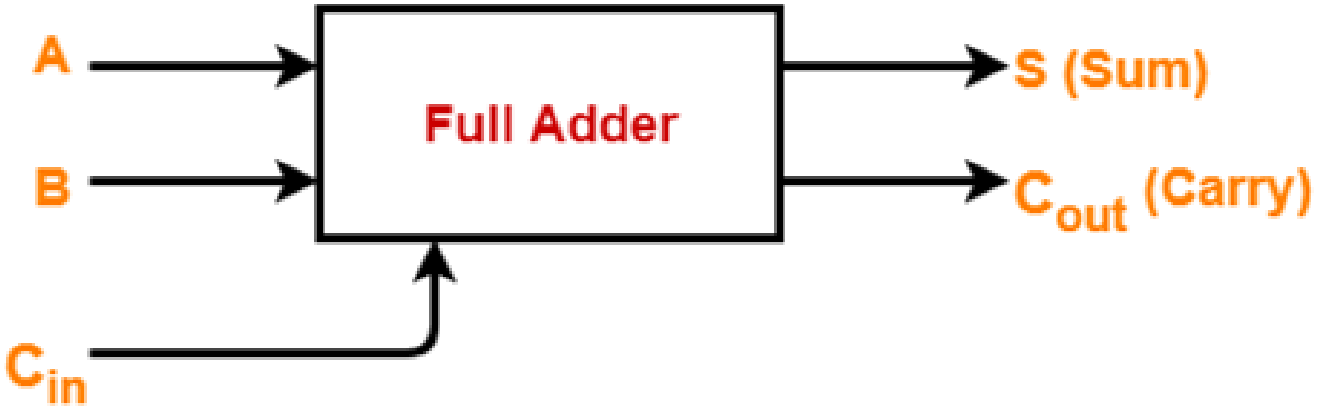
Half Adder Logic Diagram

Limitation of Half Adder-

- Half adders have no scope of adding the carry bit resulting from the addition of previous bits.
- This is a major drawback of half adders.
- This is because real time scenarios involve adding the multiple number of bits which can not be accomplished using half adders.

Full Adder-

- Full Adder is a combinational logic circuit.
- It is used for the purpose of adding two single bit numbers with a carry.
- Thus, full adder has the ability to perform the addition of three bits.
- Full adder contains 3 inputs and 2 outputs (sum and carry) as shown-



Full adder is designed in the following steps-

Step-01:

Identify the input and output variables-

- Input variables = A, B, C_{in} (either 0 or 1)
- Output variables = S, C_{out} where S = Sum and C_{out} = Carry

Step-02:

Draw the truth table-

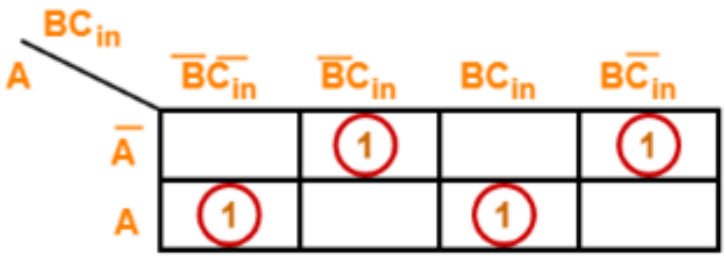
Inputs			Outputs	
A	B	C_{in}	C_{out} (Carry)	S (Sum)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth Table

Step-03:

Draw K-maps using the above truth table and determine the simplified Boolean expressions-

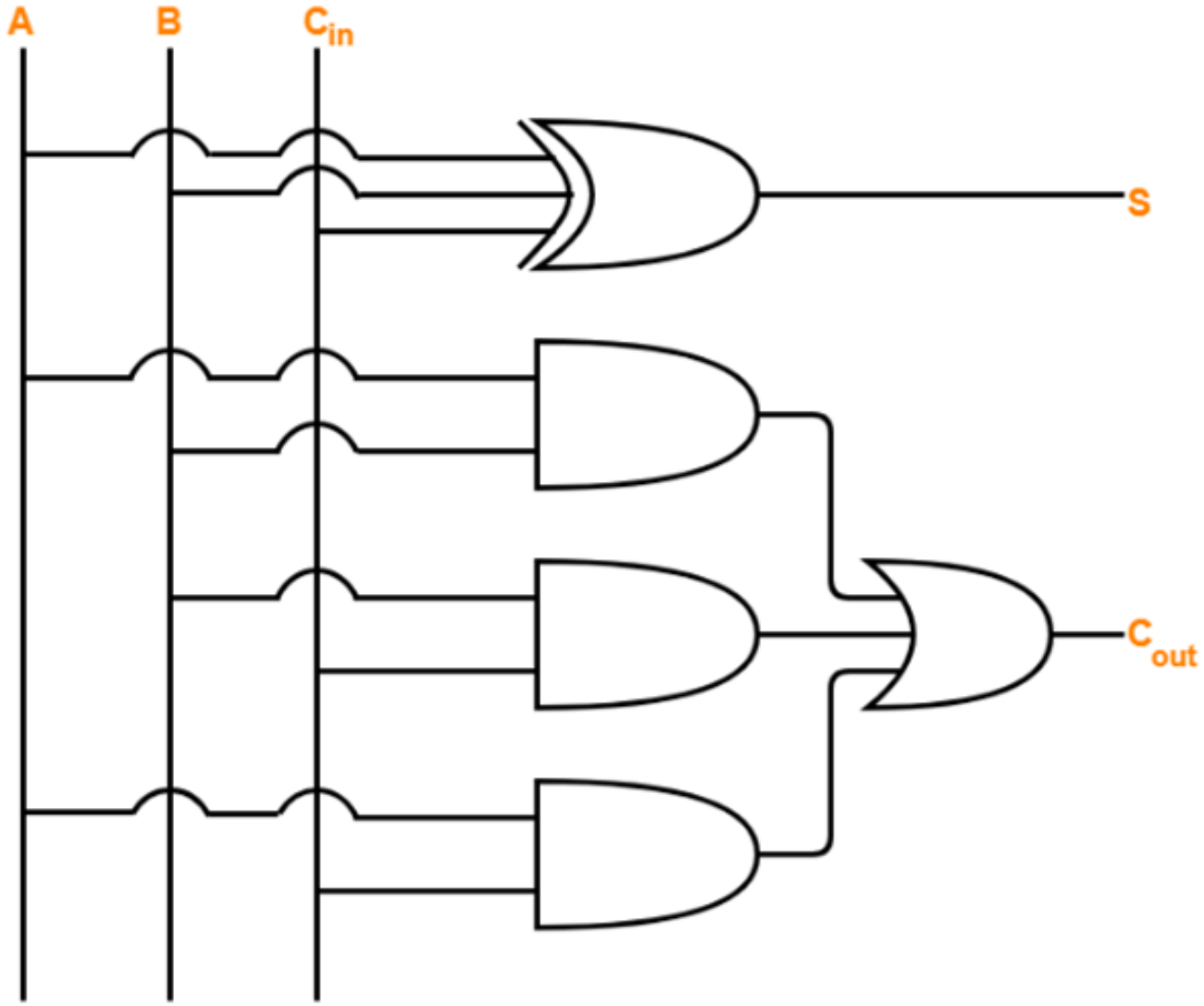
For S:



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

For C_{in}:

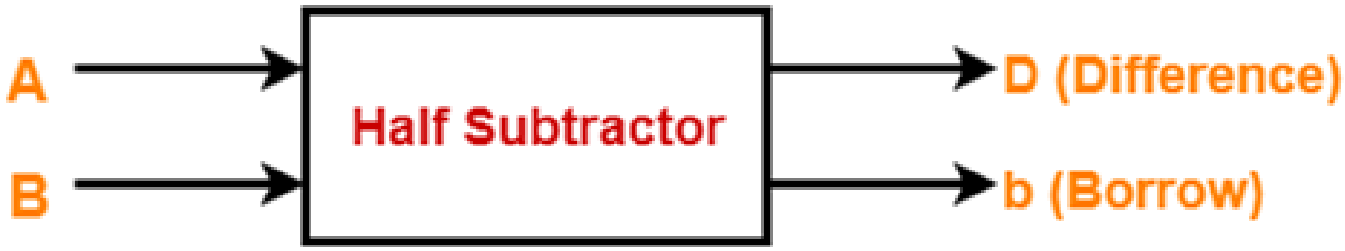
$$C_{out} = AB + BC_{in} + C_{in}A$$



Full Adder Logic Diagram

Half Subtractor-

- Half Subtractor is a combinational logic circuit.
- It is used for the purpose of subtracting two single bit numbers.
- It contains 2 inputs and 2 outputs (difference and borrow).



Step-01:

Identify the input and output variables-

- Input variables = A, B (either 0 or 1)
- Output variables = D, b where D = Difference and b = borrow

Step-02:

Draw the truth table-

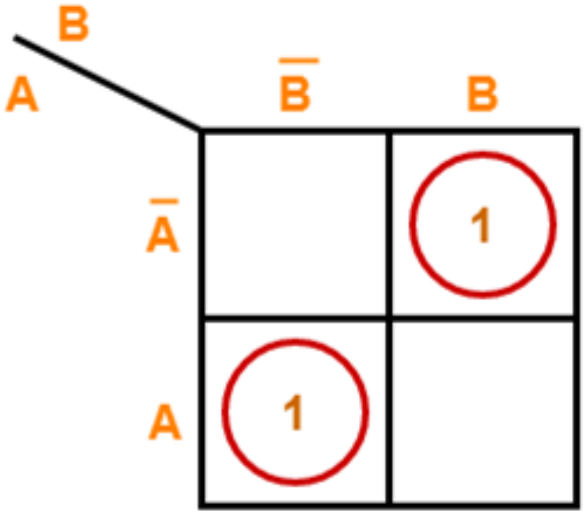
Inputs		Outputs	
A	B	D (Difference)	b (Borrow)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Truth Table

Step-03:

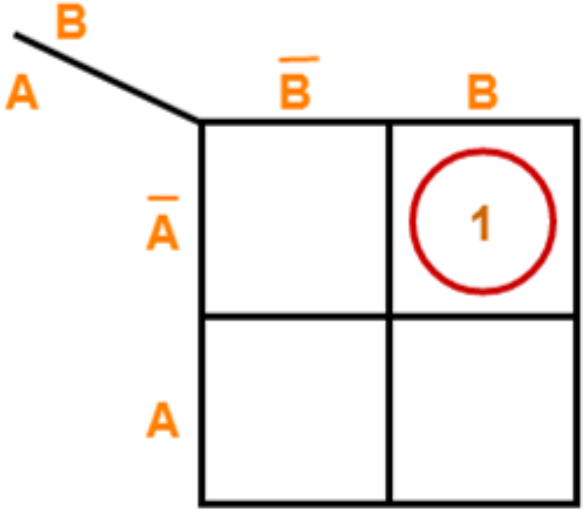
Draw K-maps using the above truth table and determine the simplified Boolean expressions-

For D:



$$D = A \oplus B$$

For b:



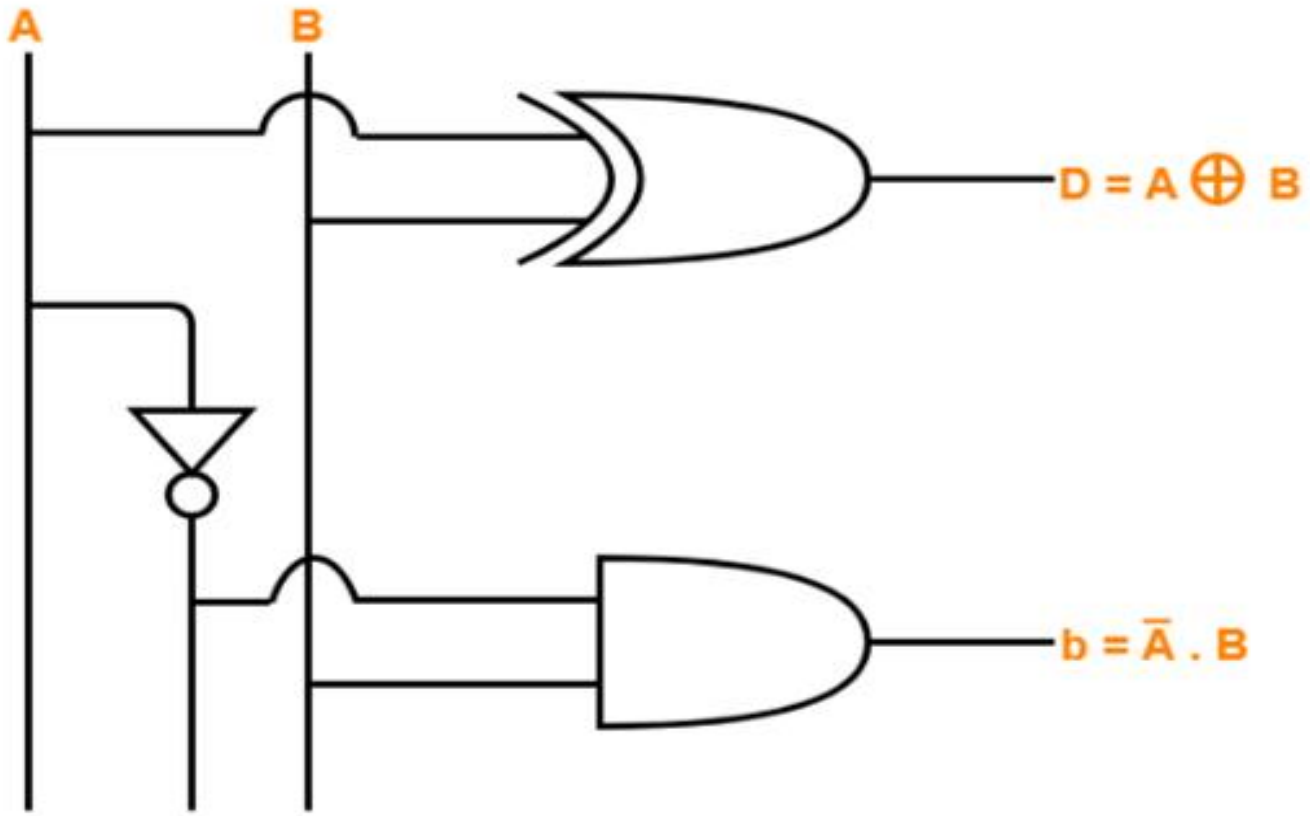
$$b = \overline{A}.B$$

K Maps

Step-04:

Draw the logic diagram.

The implementation of half subtractor using 1 XOR gate, 1 NOT gate and 1 AND gate is as shown below-



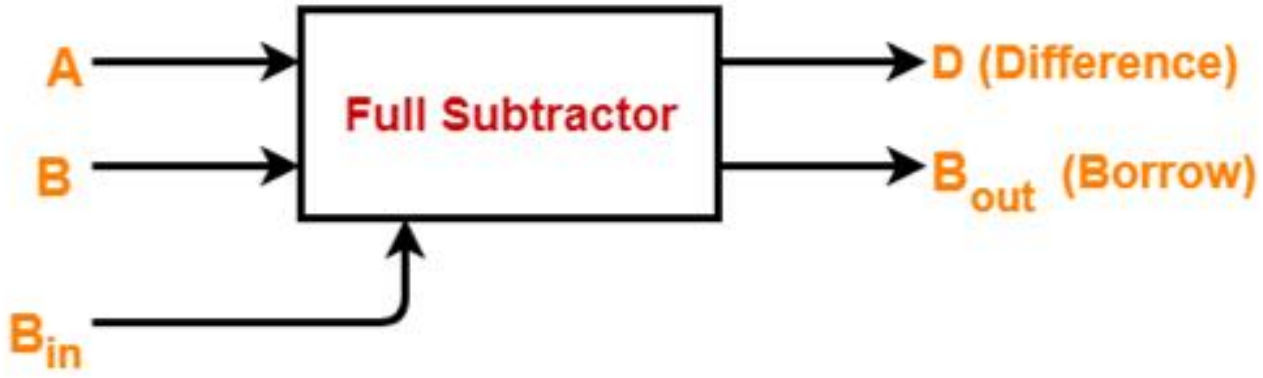
Half Subtractor Logic Diagram

Limitation of Half Subtractor-

- Half subtractors do not take into account “Borrow-in” from the previous circuit.
- This is a major drawback of half subtractors.
- This is because real time scenarios involve subtracting the multiple number of bits which can not be accomplished using half subtractors.

Full Subtractor-

- Full Subtractor is a combinational logic circuit.
- It is used for the purpose of subtracting two single bit numbers.
- It also takes into consideration borrow of the lower significant stage.
- Thus, full subtractor has the ability to perform the subtraction of three bits.
- Full subtractor contains 3 inputs and 2 outputs (Difference and Borrow) as shown-



Designing a Full Subtractor-

Full subtractor is designed in the following steps-

Step-01:

Identify the input and output variables-

- Input variables = A, B, B_{in} (either 0 or 1)
- Output variables = D, B_{out} where D = Difference and B_{out} = Borrow

Step-02:

Draw the truth table-

Inputs			Outputs	
A	B	B_{in}	B_{out} (Borrow)	D (Difference)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Truth Table

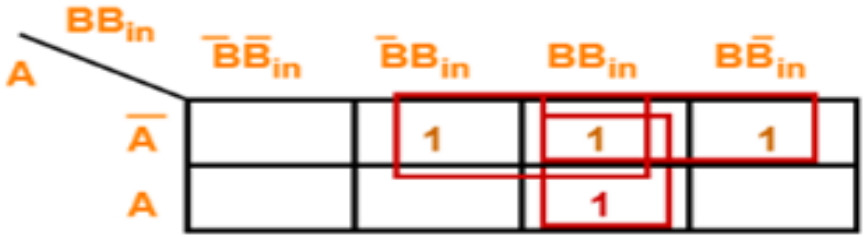
Step-03:

Draw K-maps using the above truth table and determine the simplified Boolean expressions-

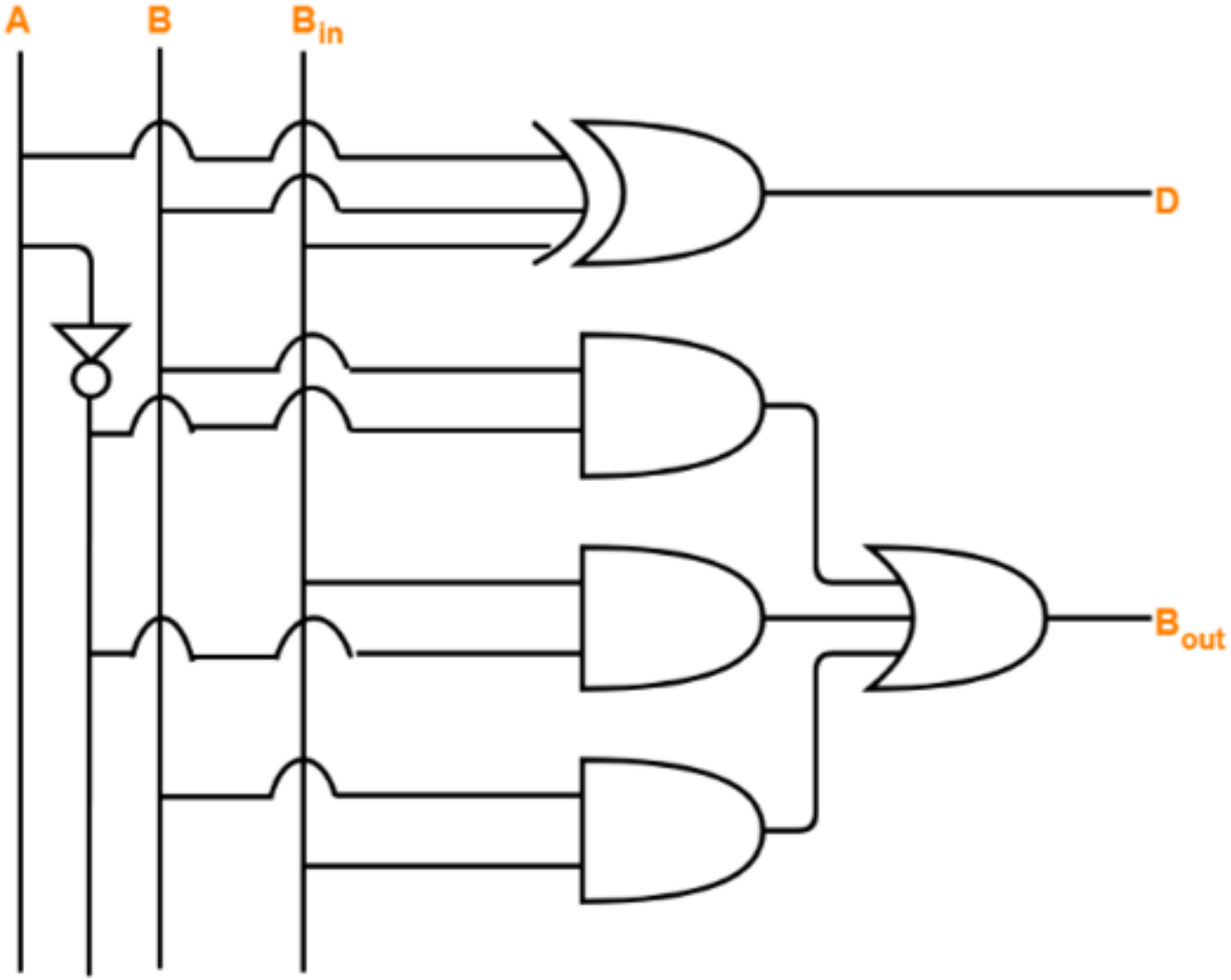
For D:

$$D = A \oplus B \oplus B_{in}$$

For B_{in}:



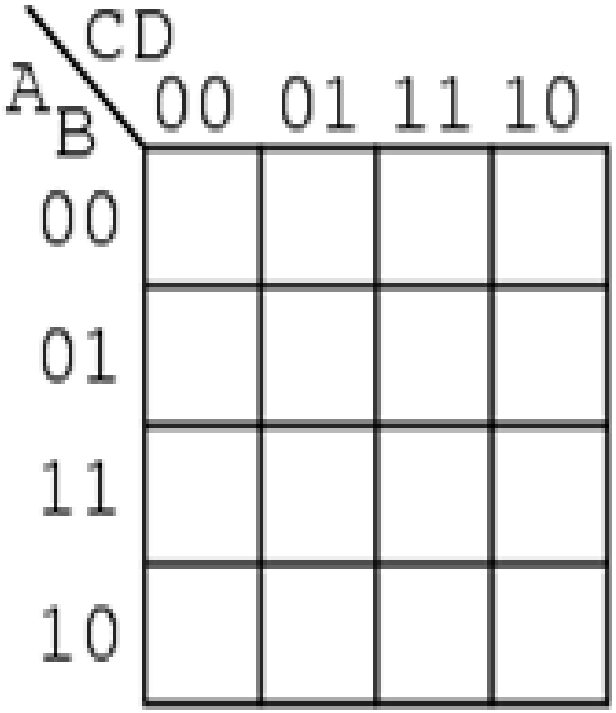
$$B_{out} = \bar{A}\,B + (\bar{A} + B)\,B_{in}$$



Full Subtractor Logic Diagram

Q. Design a combinational logic circuit whose output is high (1)only when majority of inputs (A,B,C,D) are low(0).

- Solution:
  Write the truth table



A	B	C	D	Output
0	0	0	0	**1**
0	0	0	1	**1**
0	0	1	0	**1**
0	0	1	1	0
0	1	0	**0**	**1**
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	**1**
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Q. Write the Boolean equation and draw logic diagram for the logic that will have output as Y and inputs A,B,C.
The logic performs the following operation:
1.Y=1
when A=B=C=1
and when A=B=0 and C=1

3.Y=0 for all other cases.

Digital Logic Gate Symbols

Gate	Symbol	Notation	Truth Table
AND		$A \cdot B$	INPUT / OUTPUT A B A AND B 0 0 0 0 1 0 1 0 0 1 1 1
OR		$A + B$	INPUT / OUTPUT A B A OR B 0 0 0 0 1 1 1 0 1 1 1 1
NOT		$\overline{A}$	INPUT / OUTPUT A NOT A 0 1 1 0
NAND		$\overline{A \cdot B}$	INPUT / OUTPUT A B A NAND B 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$\overline{A + B}$	INPUT / OUTPUT A B A NOR B 0 0 1 0 1 0 1 0 0 1 1 0
XOR		$A \oplus B$	INPUT / OUTPUT A B A XOR B 0 0 0 0 1 1 1 0 1 1 1 0

Multiplexer (Data Selectors)

What are multiplexers? Draw the logical circuits and describe its working.

- The term Multiplexer means many into one.
- Multiplexing is the process of transmitting a large number of information over a single line.
- A Digital Multiplexer (MUX) is a combinational Circuit that selects one digital information from several sources and transmits the selected information on a single output line.
- A Multiplexer is also called a Data Selector.
- The Multiplexer has several data input line and a single output line.

Cont.

➢MUX directs one of the inputs to its output line by using a control bit word (*selection line) to its select lines.*

➢ Multiplexer contains the followings:

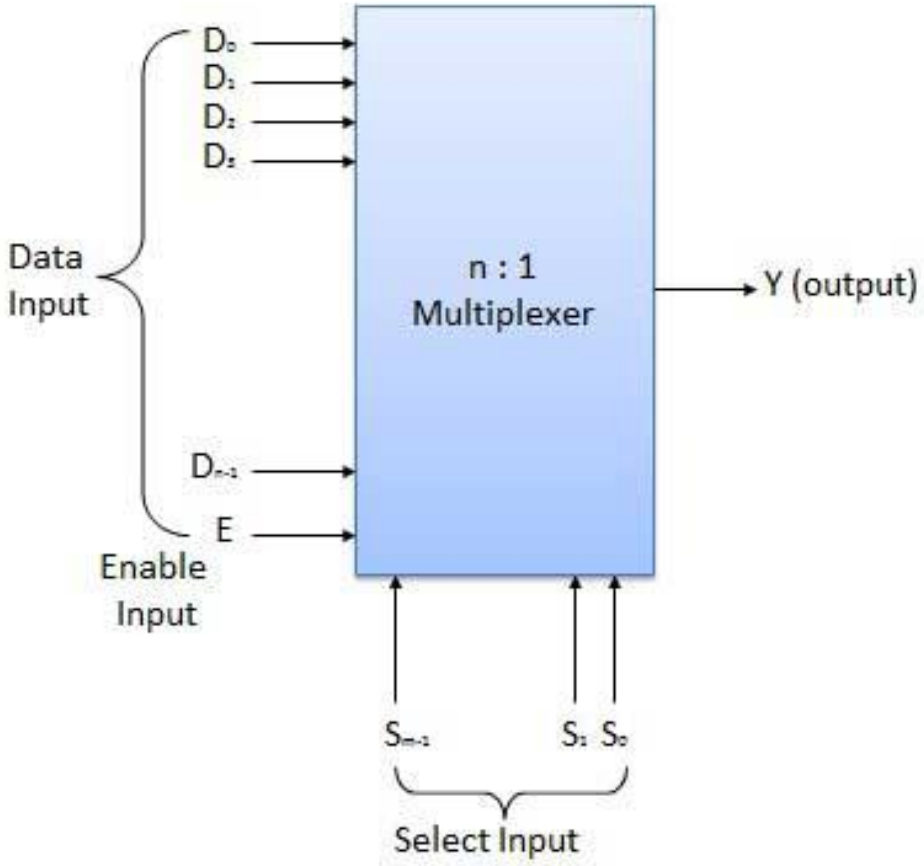
❖ data inputs

❖ selection inputs

❖ a *single output*

➢ Selection input determines the input that should be connected to the output.

➢The multiplexer acts like an electronic switch that selects one from different.


D0
D1
D2
D3
Data
Input
n : 1
Multiplexer
Y (output)
Dn-1
E
Enable
Input
Sn-1
S1
S0
Select Input

Explain the necessity of multiplexers.

- In most of the electronic systems, the digital data is available from more than one sources. It is necessary to route this data over a single line.
- Under such circumstances we require a circuit which selects one of the many sources at a time. This circuit is nothing else but a multiplexer, which has many inputs, one output and some select inputs.
- Multiplexer improves the reliability of the digital system because it reduces the number of external wired connections.

Advantages of Multiplexers:

Write any two advantages of MUX.

It reduces the number of wires, required to be used.

A multiplexer reduces the circuit complexity and cost.

We can implement many combinational circuits using MUX.

It simplifies the logic design.

It does not need the k maps for simplification.

Types of MUX:

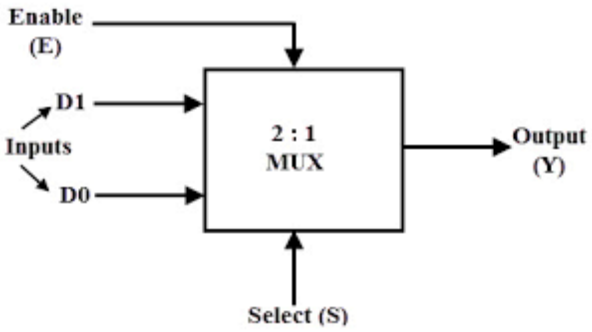
1.2:1 MUX 2. 4:1 MUX 3) 8:1 MUX

4. 16:1 MUX 5. 32:1 MUX

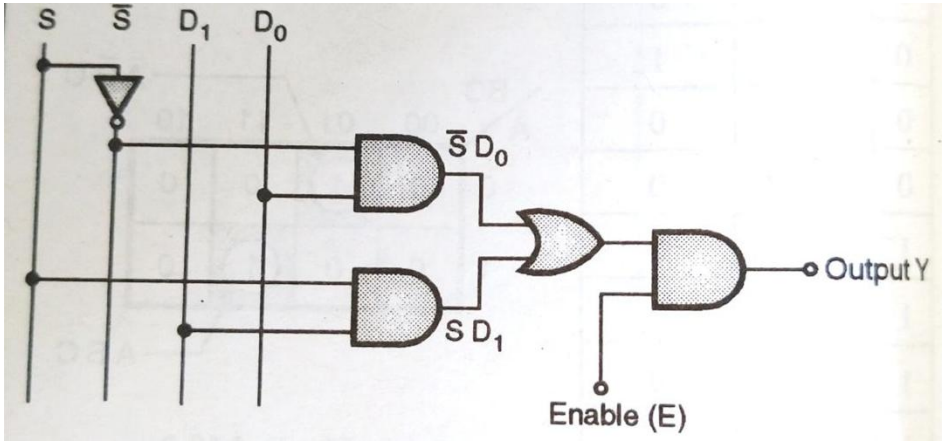
2:1Multiplexer

Draw the symbol and write truth table of 2:1 MUX

1. The block schematic of a 2 : 1 multiplexer is shown in Fig. below.
2. It has two data inputs Do and D_1, one select input S, an enable input E and one output Y.
3. The truth table of this MUX is shown in Fig. below

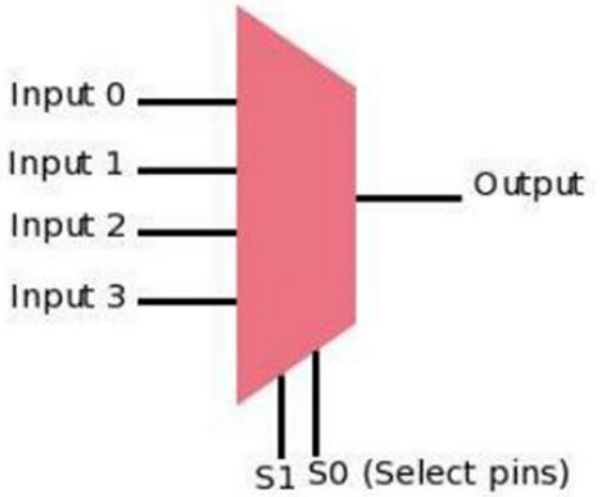


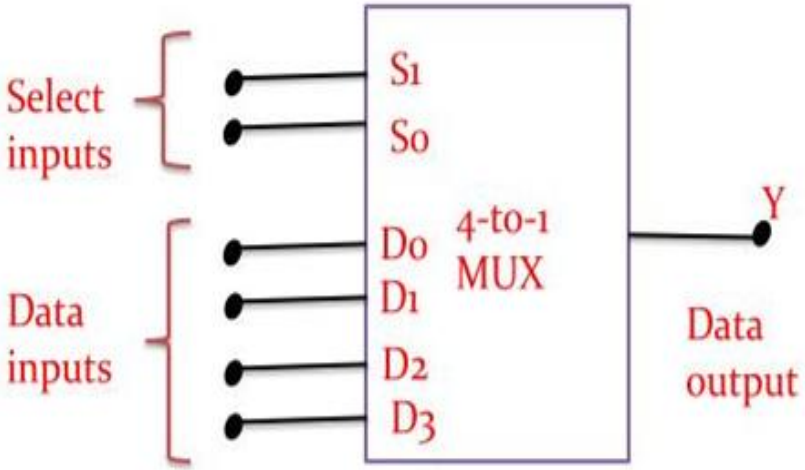
Enable	Select Input S	Output Y
0	X	0
1	0	D0
1	1	D1



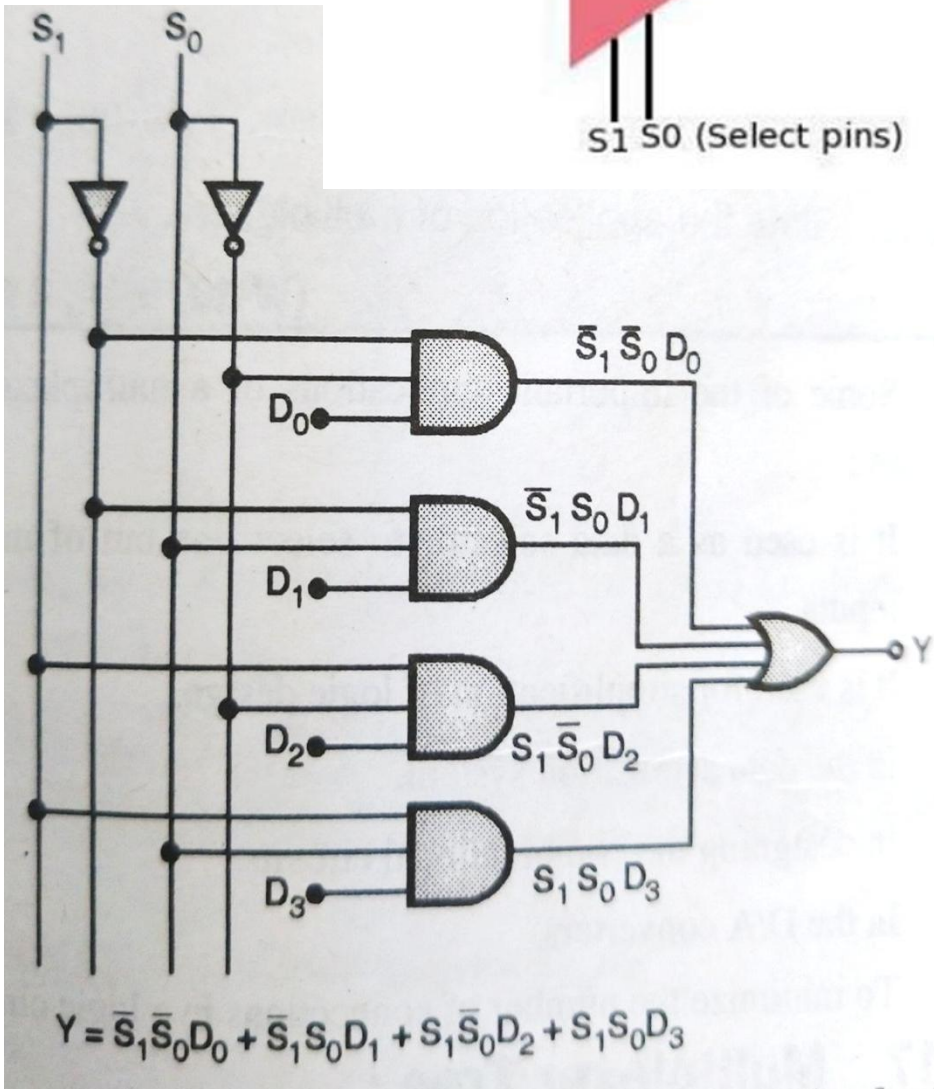
4:1 Multiplexer

Explain the operation of 4:1 Mux using logic diagram





Data select inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

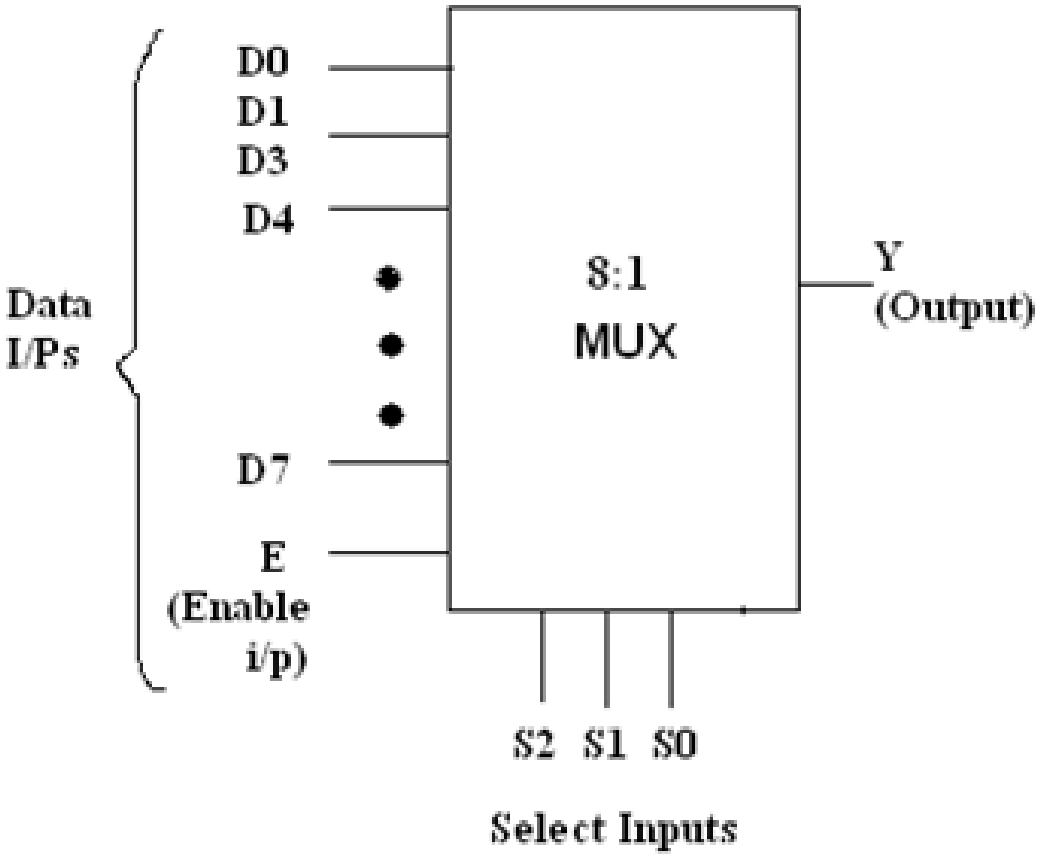


1. Fig. shows the block diagram of a 4: 1 multiplexer and Table gives its truth table.
2. Note that n = 4 hence number of select lines i.e. m = 2 so that 2 = n.
3. The truth table tells us that if S_1 So 00, the data bit Do is selected and routed to output Y = Do
4. Similarly if S_1 So 01, then D_1 is selected and routed to the output Y = D_1
5. Y = D_2 for S_1 So = 10 and Y = D3 for S_1 So 11
6. Hence the logical expression for output in the SOP form is as follows:

$$Y = \overline{S}_1\ \overline{S}o\ Do + \overline{S}_1\ So\ D_1 + \overline{S}_1\ So\ D_2 + S_1\ So\ D3$$

This expression can be realized using basic gates as shown in

8:1 Multiplexer



Enable	Select Inputs			Output
E	S2	S1	S0	Y
0	X	X	X	0
1	0	0	0	D0
1	0	0	1	D1
1	0	1	0	D2
1	0	1	1	D3
1	0	0	0	D4
1	0	0	1	D5
1	0	1	0	D6
1	0	1	1	D7

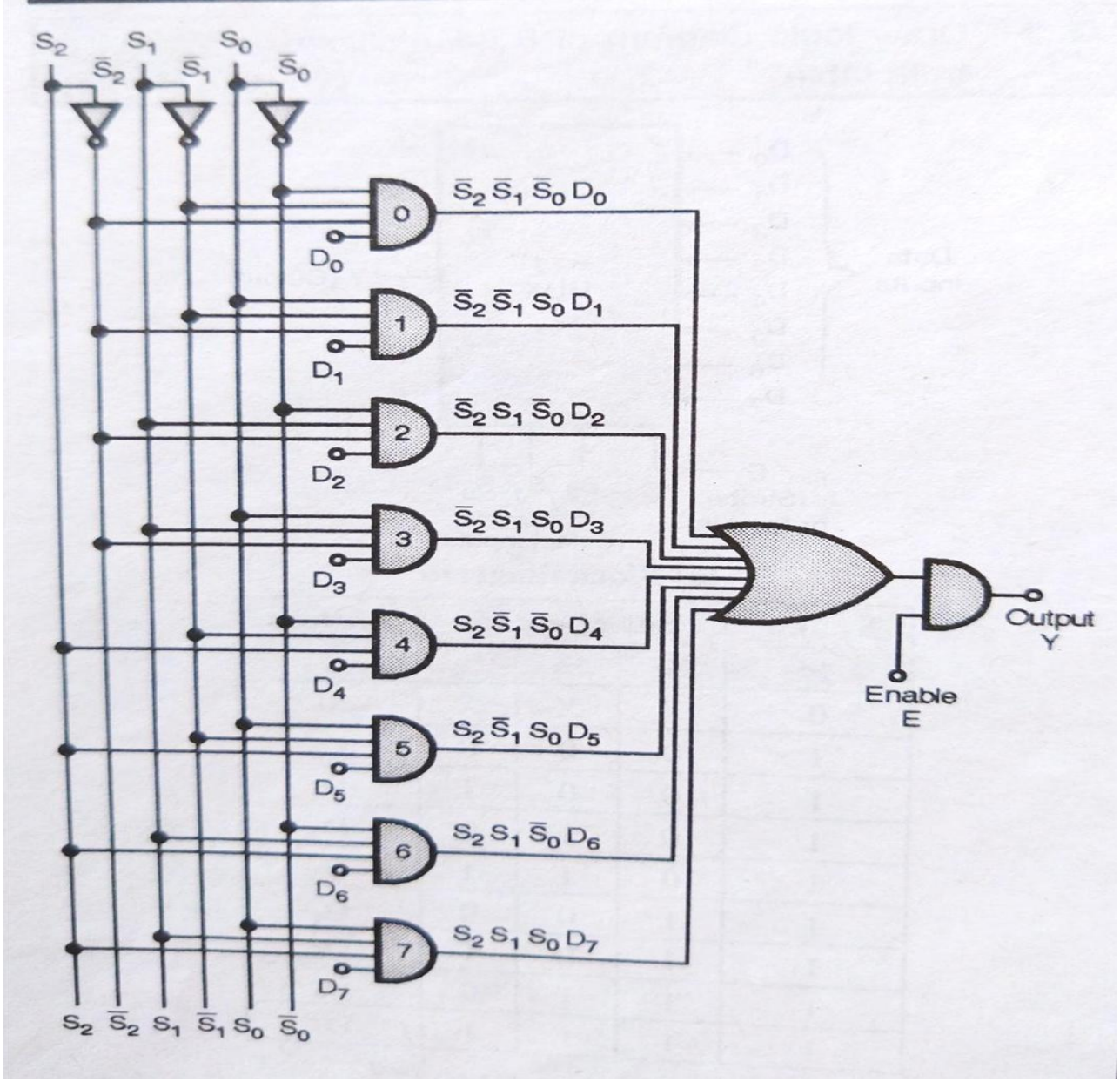
1. The block diagram of an 8 1 MUX is shown in Fig. 1and its truth table is shown in Fig. 2 .It has eight data inputs, one enable input, three select inputs and one output.

Operating principle :

1. When the strobe or enable input is 0, the multiplexer will be 0 irrespective of any input. With E = 1, we can select any one of the eight data inputs and of the connect it to the output.
2. For example if S_2 S, So = 0 1 1 then the data input D, is selected and output Y will follow the selected input D3.
3. Realization using gates :
4. The expression for output Y can be obtained from the truth table of Fig. as

$Y = E.[S_2 S_1 S0 Do + S_2 S_1 So D_1 + S_2 S_1 S0 D_2 + S_2 S_1 S0 D3 + S_2 S_1 So D4 + S_2 S_1 SO D5 + S_2 S_1 So D6 + S_2 S_1 SqD7]$

5. The realization of 8 : 1 MUX using gates is shown in Fig.


S_2 $\bar{S}_2$ S_1 $\bar{S}_1$ S_0 $\bar{S}_0$
$\bar{S}_2\,\bar{S}_1\,\bar{S}_0\,D_0$
$\bar{S}_2\,\bar{S}_1\,S_0\,D_1$
$\bar{S}_2\,S_1\,\bar{S}_0\,D_2$
$\bar{S}_2\,S_1\,S_0\,D_3$
$S_2\,\bar{S}_1\,\bar{S}_0\,D_4$
$S_2\,\bar{S}_1\,S_0\,D_5$
$S_2\,S_1\,\bar{S}_0\,D_6$
$S_2\,S_1\,S_0\,D_7$
0
1
2
3
4
5
6
7
D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7
Output Y
Enable E
S_2 $\bar{S}_2$ S_1 $\bar{S}_1$ S_0 $\bar{S}_0$

Applications of a Multiplexer:

- Some of the important applications of a multiplexer are as follows:

1. It is used as a data selector to select one out of many data inputs.

2. It is used for simplification of logic design.

3. In the data acquisition system. In designing the combinational circuits.

4.In the D/A converters.

5.To minimize the number of connections in a logic circuit.

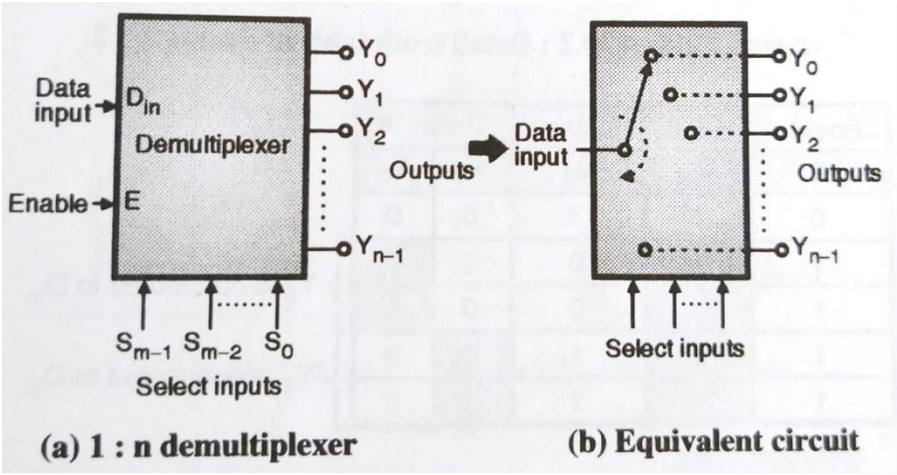
De-Multiplexer

1. The De-Multiplexer is a combinational logic circuit that performs the reverse operation of multiplexer (Several output lines, one input line).
2. De -Multiplexer means one to many.
3. A De-Multiplexer is a circuit with one input and many output. By applying control signal, we can steer any input to the output. Few types of De -Multiplexer are 1-to 2, 1-to-4, 1-to-8 and 1-to 16 De -Multiplexer .
4. De-Multiplexer is the process of taking information from one input and transmitting the same over one of several outputs.

Define and draw the logical symbol of a demultiplexer.

1. The block diagram of a demultiplexer or decoder is shown in Fig
2. It has only one input, "n" outputs, and "m" select inputs. A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs.
3. At a time only one output line is selected by the select lines and the input is routed to the selected output line.
4. Hence a demultiplexer is equivalent to a single pole multiple way switch as shown in Fig. The enable input will enable the demultiplexer. If the enable (E) input is active, then the demultiplexer does not work.
5. The relation between the n output lines and m select lines is as follows:

$n = 2m$



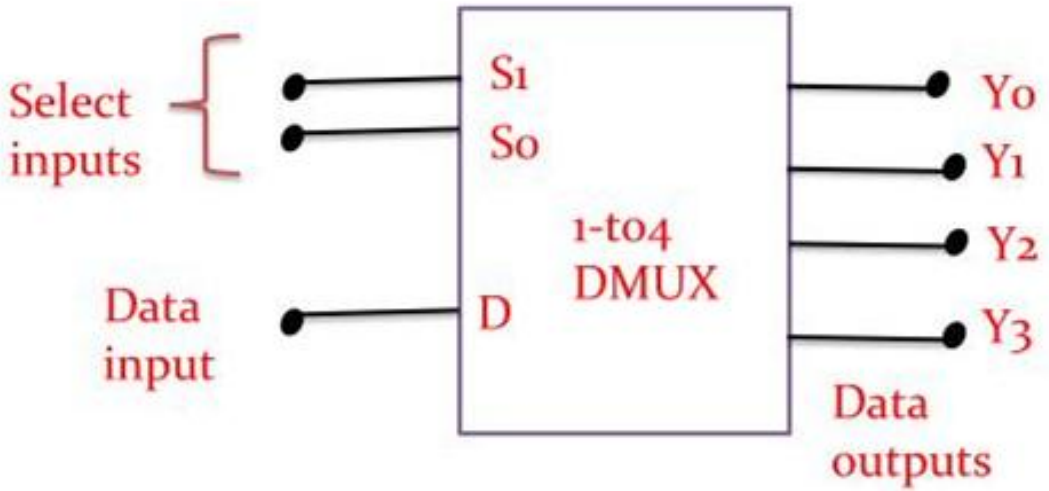
(a) 1 : n demultiplexer (b) Equivalent circuit

Types of Demultiplexers :

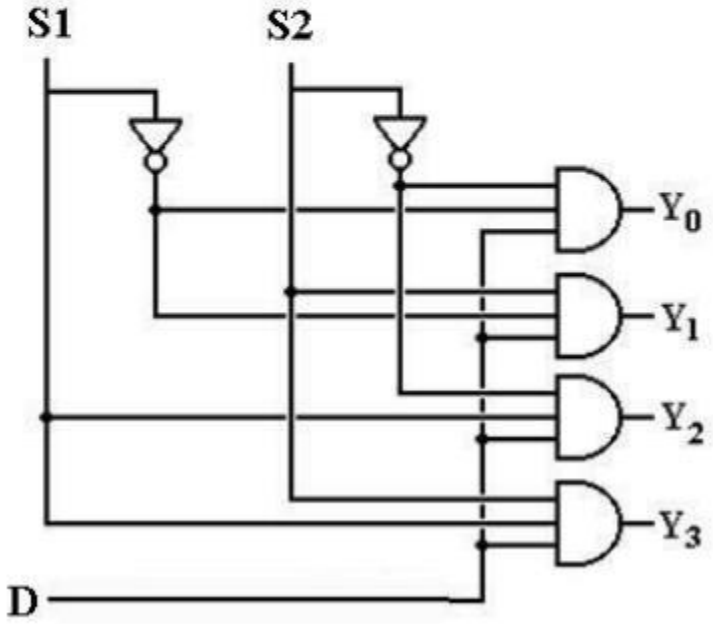
- Similar to the multiplexers, the demultiplexers are classified as follows:

1. 1:2 demultiplexer
2. 1:4 demultiplexer
3. 1:8 demultiplexer
4. 1: 16 demultiplexer

1: 2 Demultiplexer:



1 to 4 DeMultiplexer Logic diagram



Truth Table of 1 to 4 DeMultiplexer

Data Input	Select Inputs		Output			
D	S_1	S_0	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0

Applications of Multiplexer

Communication system – Communication system is a set of system that enable communication like transmission system, relay and tributary station, and communication network. The efficiency of communication system can be increased considerably using multiplexer. Multiplexer allow the process of transmitting different type of data such as audio, video at the same time using a single transmission line.

Computer Memory – A Multiplexer is used in computer memory to keep up a vast amount of memory in the computers, and also to decrease the number of copper lines necessary to connect the memory to other parts of the computer.

Applications of De-Multiplexer

Communication System - Communication system use multiplexer to carry multiple data like audio, video and other form of data using a single line for transmission. This process make the transmission easier. The demultiplexer receive the output signals of the multiplexer and converts them back to the original form of the data at the receiving end. The multiplexer and demultiplexer work together to carry out the process of transmission and reception of data in communication system.

Multiplexer Tree

- The multiplexers having more number of inputs can be obtained by cascading two or more multiplexers with less number of inputs.
- This is called as a multiplexer tree.

Obtain an 8: 1 multiplexer using two 4:1 multiplexers.

- Soln. :
- The cascading of two 4: 1 multiplexer results in 8:1 multiplexer as shown in Fig. below
- There are in all eight data inputs (Do through D4).
- The select lines S, and So of both 41 multiplexers are connected in parallel whereas a third select input S_2 is used for enabling one multiplexer at a time.
- S_2 is connected directly to the enable (E) terminal of MUX-1
- whereas S_2 is connected to the enable terminal of MUX-2.

Ex: Implement the following Boolean function using 8:1 multiplexer.

$$F(A, B, C, D) = \sum m(2, 4, 5, 7, 10, 14)$$

Implement a 16 : 1 multiplexer using 4 : 1 multiplexers

Design 8: 1 MUX using 2:1 MUX and 4:1

Implement the following logic expression using 16: 1 mux : $Y = \Sigma m\ (0, 3, 5, 7, 8, 9, 12, 13)$

Implement the following function using 4 : 1 Mux: $Y = \Sigma m\ (0, 1, 3)$ MUX.

Design a 32:1 MUX using 8:1 multiplexers.

Draw 1:64 Demultiplexer tree using 1:16 demultiplexer

Draw 1:16 Demux using 1:4 demultiplexer

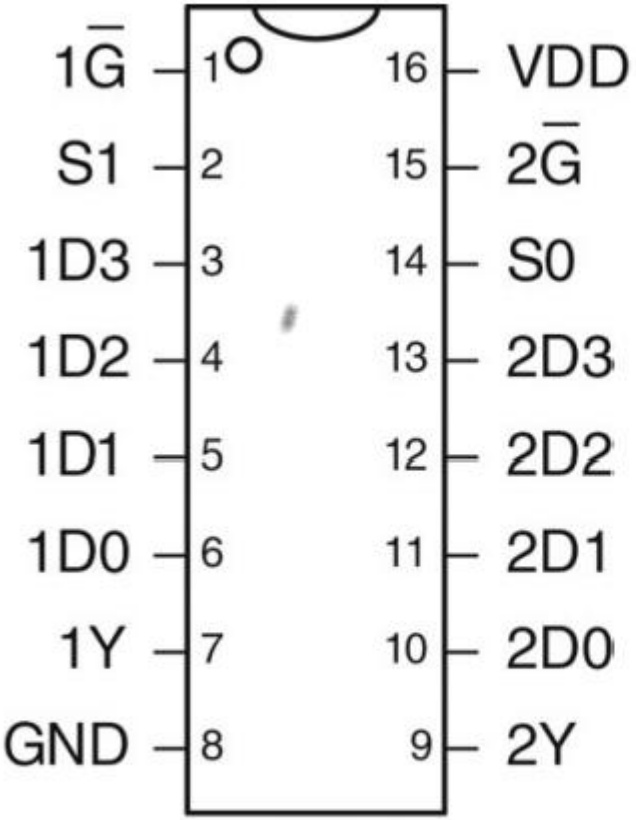
Implement a full adder using demultiplexer.

A	B	Cin	S	Cout	Y
0	0	0	0	0	Y0
0	0	1	1	0	Y1
0	1	0	1	0	Y2
0	1	1	0	1	Y3
1	0	0	1	0	Y4
1	0	1	0	1	Y5
1	1	0	0	1	Y6
1	1	1	1	1	Y7

74153 Multiplexer IC

- **Specifications :**
- TTL ICs i.e. Transistor Transistor Logic ICs.
- Total pins = 16 pins.
- Dual 4:1 line multiplexer circuit in single package.
- Supply voltage = 5V DC ±0.25V.
- Operating temperature range = 0°C to +70°C.
- High fan-out = 10.
- Low impedance < 100Ω
- Totem pole outputs.

- **Description**
- The 74153 MUX has two separate 2-input/4-row MUXs on it. To create a single 16-row truth table, we can start by implementing parts of the table on different MUXs, and then combining the two separate outputs into one output. We'll turn on only the MUX needed using the STROBEs. The multiple outputs are combined with an OR gate. Remember, each strobe turns its MUX on when it is low.
  On each MUX, we have to use the MUX doubling technique to fit a 3-input/8-row truth table onto a 2-input/4-row MUX.
  The circuit shown will generate the accompanying truth table. Besides the 74153 MUX, it requires two inverters (C and D), and one two-input OR gate.
-



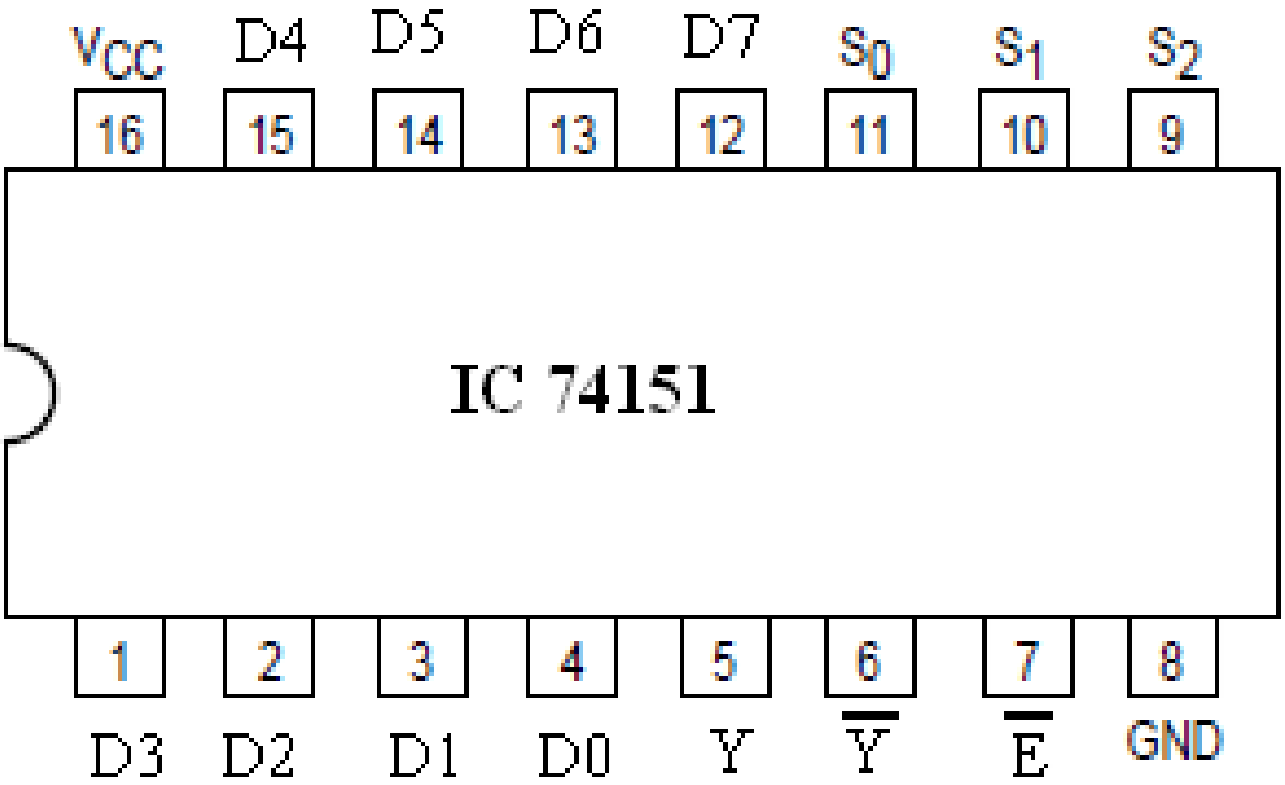
74153 4 :1 Mux

Two 4:1 Multiplexers

$D_{3:0}$:	data
$S_{1:0}$:	select
Y:	output
Gb:	enable

```
always_comb
  if (1Gb) 1Y = 0;
  else     1Y = 1D[S];
always_comb
  if (2Gb) 2Y = 0;
  else     2Y = 2D[S];
```

MULTIPLEXER IC 74151



- **THEORY:**

-
- Multiplexer is a combinational circuit that is one of the most widely used in digital design.
- The multiplexer is a data selector which gates one out of several inputs to a single o/p. It has n data inputs & one o/p line & m select lines where 2^m= n shown in fig a.
- Depending upon the digital code applied at the select inputs one out of n data input is selected & transmitted to a single o/p channel.
- Normally strobe (G) input is incorporated which is generally active low which enables the multiplexer when it is LOW. Strobe i/p helps in cascading.
- IC 74151A is an 8: 1 multiplexer which provides two complementary outputs Y & Y. The o/p Y is same as the selected i/p & Y is its complement. The n: 1 multiplexer can be used to realize a m variable function. (2^m= n, m is no. of select inputs)

Necessity of multiplexers:

-
- In most of the electronic systems, the digital data is available on more than one line. It is necessary to route this data over a single line.
- Under such circumstances we require a circuit which selects one of the many inputs at a time.
- This circuit is nothing else but a multiplexer, which has many inputs, one output & some select inputs.
- Multiplexer improves the reliability of the digital system because it reduces the number of external wired connections.

- In digital electronics, an encoder is a device that converts information from one format to another. The most common type of encoder is used to convert digital signals from parallel to serial format. An encoder in digital electronics is a combinational circuit that converts a set of input signals into a binary code. Encoders are used in various applications such as data compression, communication systems, and digital signal processing.

- An encoder is a combinational logic circuit that accepts multiple inputs and generates a binary output. The encoder is often used when a digital system needs to identify which input line is active from multiple input lines.
- Encoders are used to compress data or convert it into a format that can be easily transmitted, stored, or processed.

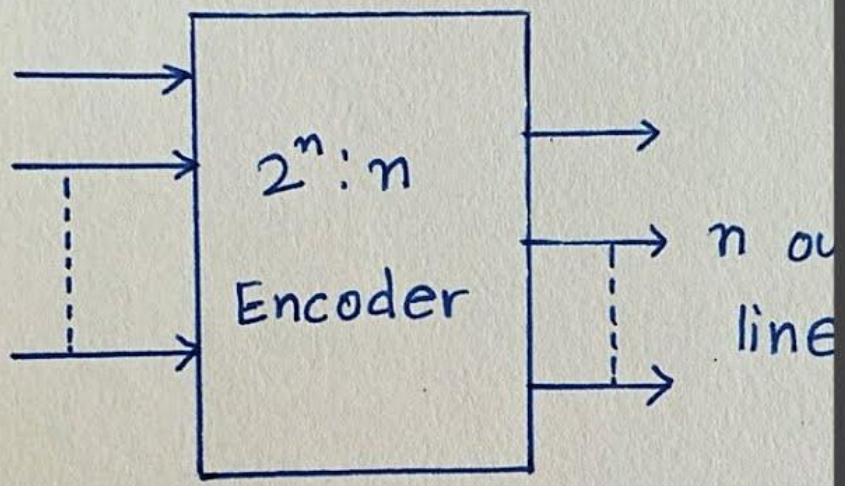
- An encoder is a digital combinational circuit that transforms human-readable information into a coded format for machine processing. Simply put, an encoder converts normal information into a coded form, a process known as encoding.
- Encoders play an essential role in various digital electronics applications, including data transmission, control and automation systems, communication, and signal processing.

- **Types of Encoders**
- Encoders can be classified based on the number of input and output lines. The most common types of encoders include:
- 4 to 2 Encoder
- 8 to 3 Encoder (Octal Encoder)
- Decimal to BCD Encoder

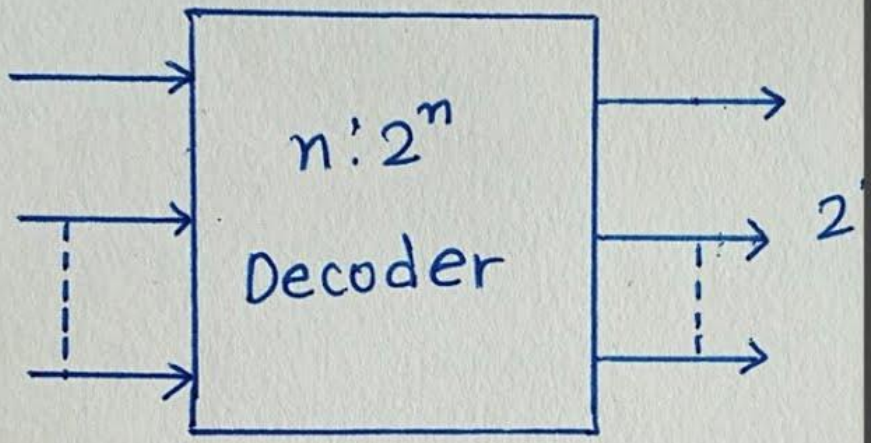
Applications of Encoders

- **Multiplexers:** Converts parallel data bus to serial data stream for transmission or vice versa.
- **Communication Systems:** Encodes digital signals for efficient transmission over limited bandwidth channels.
- **Keyboards and Scanners:** Encodes key presses/scan lines into digital format.
- **Seven Segment Displays:** Uses BCD encoders to display decimal numbers.
- **Barcode Readers:** Decodes barcodes scanned as varying reflectance into digital data format.

Encoders



Decoders



Applications of 74LS138 IC

- The **applications of IC 74LS138** include the following.
- Line decoders
- Memory circuits
- Servers
- Digital systems
- Line De-multiplexing
- Telecom circuits

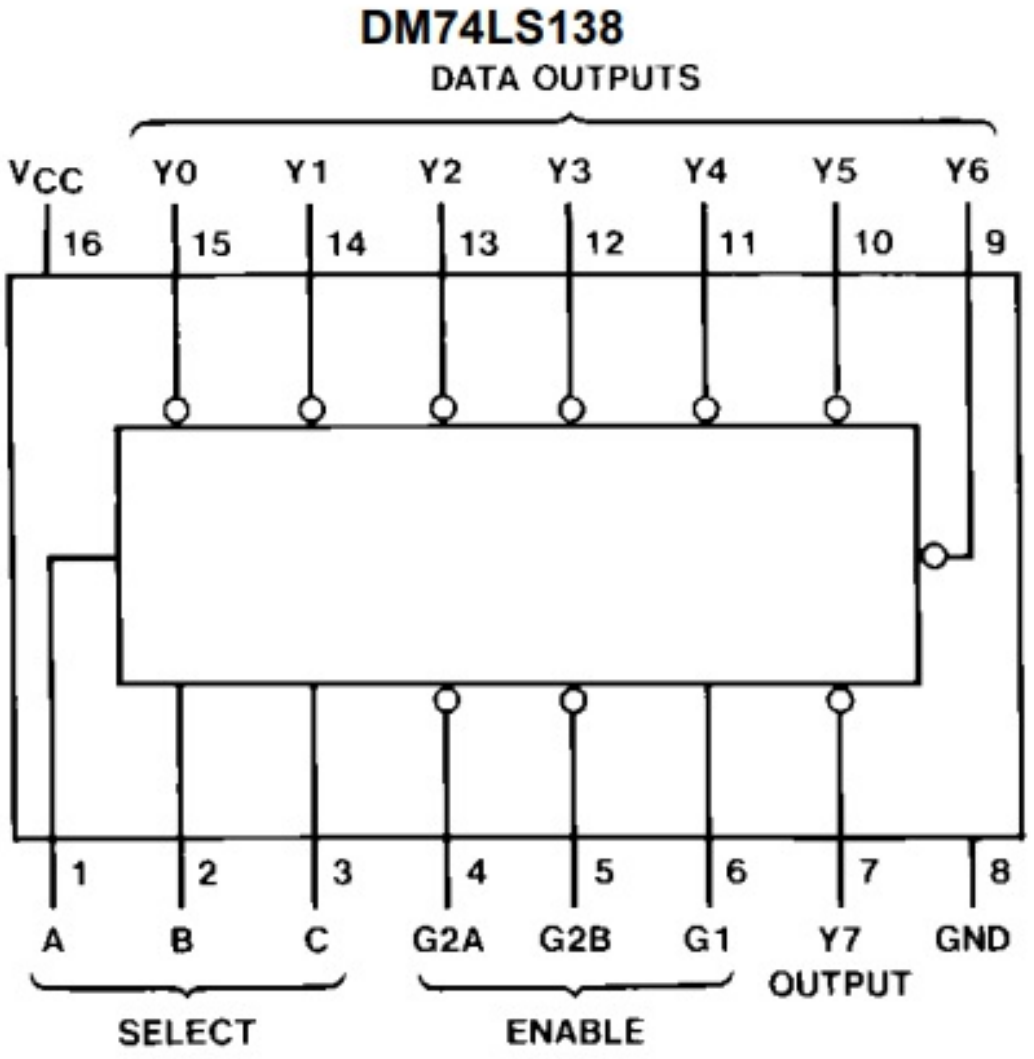
74LS138 IC

3 to 8 Line Decoder IC.

- **What is a 74LS138 IC?**
- The IC 74LS138 is a 3 to 8 line decoder **integrated circuit** from the 74xx family of **transistor-transistor-logic-gates**. The main function of this IC is to decode otherwise demultiplex the applications. The setup of this IC is accessible with 3-inputs to 8-output setup. This IC is mainly used in applications like memory decoding with high performance otherwise data routing, etc. These ICs can be used for minimizing the system decoding effects in memory systems with high performance. This IC includes three enable pins (where two pins are active low and one is active high) decreases the necessity of outside gates. The implementation of 24 line decoder can be done without using outside inverters, as well as a 32-line decoder needs a single inverter
- This IC is mainly used in **de-multiplexing** applications with the help of an enable pin like a data input pin. And also the inputs of this IC is clamped with **Schottky diodes** which are the high performance to contain line ringing as well as system design simplify.

74LS138 Pin Configuration

The **IC 74LS138 is a 16-pin integrated circuit**, and each pin of this IC is discussed below. The similar 74LS138 IC's are



- Pin1 (A): Address input pin
- Pin2 (B): Address input pin
- Pin3 (C): Address input pin
- Pin4 (G2A): Active low enable pin
- Pin5 (G2B): Active low enable pin
- Pin6 (G1): Active high enable pin
- Pin7 (Y7): Output pin
- Pin8 (GND): Ground pin
- Pin9 (Y6): Output pin 6
- Pin10 (Y5): Output pin 5
- Pin11 (Y4): Output pin 4
- Pin12 (Y3): Output pin 3
- Pin13 (Y2): Output pin 2
- Pin14 (Y1): Output pin 1
- Pin15 (Y0): Output pin 0
- Pin16 (VCC): Power supply pin

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	YO	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

why 74LS138 decoder IC generates active low output and why not 1

- Minimizing Power Consumption in TTL Logic.

THANK YOU