ABES ENGINEERING COLLEGE, GHAZIABAD

DEPARTMENT OF ELECTRONICS
&
COMMUNICATION ENGINEEING

COURSE MATERIAL

Subject Name: Fundamentals of Electronics Engineering.

Subject Code: BEC-101/201

Branch/Semester: All Branches / 1st or 2nd

Session: 2023-24 (Odd & Even-Semester)

Faculty: Dr. Manish Zadoo, Mr. Mudit Saxena, Ms. Pooja Pathak, Ms. Arpita Johri, Mr. Shahbaz Alam, Ms. Kavita Chaudhary & Dr. Shivam Singh

EVALUATION - SCHEME, B. Tech- I YR./ I SEM (AKTU)

B. Tech. First Year, Semester- I

(All Branches except Agriculture Engineering and Biotechnology)

3- WEEKS STUDENT INDUCTION PROGRAMME in the beginning of the session														
								Evaluation Scheme						
SN	Subject Code	Subject Name	Type	Category	Period			Sessional Component		Sessional (SW) (TS/PS)	End Semester Examination (ESE)	Total	Credit
					L	T	P	CT	TA	CT+TA	TE/PE	SW+ESE	Cr
1.	BAS101/ BAS102	Engineering Physics/ Engineering Chemistry	T	BS	3	1	0	20	10	30	70	100	4
2.	BAS103	Engineering Mathematics-I	T	BS	3	1	0	20	10	30	70	100	4
3.	BEE101/ BEC101	Fundamentals of Electrical Engineering/ Fundamentals of Electronics Engineering	T	ES	2	1	0	20	10	30	70	100	3
4.	BCS101/ BME101	Programming for Problem Solving/ Fundamentals of Mechanical Engineering	T	ES	2	1	0	20	10	30	70	100	3
5.	BAS104/ BAS105	Environment and Ecology/ Soft Skills	T	BS/ HS	3	0	0	20	10	30	70	100	3
6.	BAS151/ BAS152	Engineering Physics Lab/ Engineering Chemistry Lab	P	BS	0	0	3	-	50	50	50	100	1
7.	BEE151/ BEC151	Basic Electrical Engineering Lab/ Basic Electronics Engineering Lab	P	ES	0	0	3	-	50	50	50	100	1
8.	BCS151/ BAS155	Programming for Problem Solving Lab/ English Language Lab	P	ES/ HS	0	0	3	-	50	50	50	100	1
9.	BCE151 / BWS151	Engineering Graphics & Design Lab/ Workshop Practice Lab	P	ES	0	1	3	-	50	50	50	100	2
					13	5	12			350	550	900	22

UNIT-4

(Digital Electronics)

Contents as per syllabus:

1. Introduction to Digital Electronics

- Digital Electronics is the sub-branch of electronics that deals with digital signals for processing and controlling various systems and sub-systems.
- Digital electronics are the most common representation of Boolean algebra and are the basis of all digital circuits for computers, mobile phones, and numerous other consumer products.
- In various applications like sensors and actuators, usage of digital electronics is increasing extensively.
- The most common fundamental unit of digital electronics is the logic gate. By combining numerous logic gates (from tens to hundreds of thousands) more complex systems can be created.
- The complex system of digital electronics is collectively referred to as a digital circuit.

1.1 Need of Digital Electronics

- The more accurate representation of the digital signal can be generated using more binary digits and has a unique way of representation.
- Moreover, the immunity to noise in the digital system allows the data to be retrieved and stored without any degradation.
- In an analog system, the aging noise may vary due to wear and tear properties in the stored information. But in digital, if the noise is below the suggested level, the data can be restored perfectly. Even if there is any presence of significant noise, the redundancy use allows the restoring of actual data and provides a strong resistance to errors.

1.2 Analog Signal vs Digital Signal

Almost all the signals in the World are analog i.e. they are continuously varying values. There are lots of continuously variable signals or simply analog signals in nature like light, motion, sound, temperature, pressure etc.

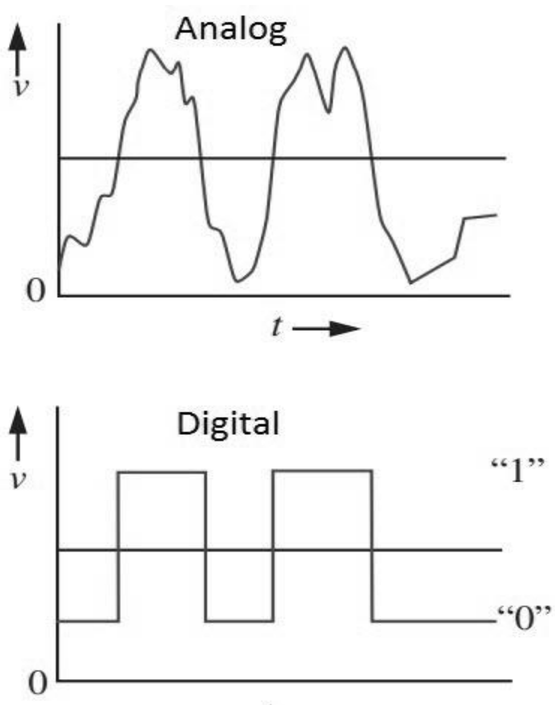


Figure 1.1 Representation of analog and digital signal

Digital Signals vary in discrete levels, in contrast to the continuous representation of analog signals. Generally, the discrete levels in a digital signal are just two values: ON and OFF. Even though all the physical signals of the nature are continuous analog signals, representing signals as discrete values has its own advantages.

Traditionally, all electronic devices processed analog information only but the development in technology has led to using digital signals for easy processing and such techniques are called Digital Signal Processing.

2. Number System and Representation

2.1 Introduction

In digital electronics, the number system is used for representing the information. The number system has different bases and the most common of them are the decimal, binary, octal, and hexadecimal. The base or radix of the number system is the total number of the digit used in the number system.

Table 2.1. Characteristics of Commonly used Number Systems

Number System	Base or radix(b)	Symbols used (d_i or d_{-j})	Weight assigned to position i	-j	Example
Binary	2	0,1	2^i	2^{-j}	1101.11
Octal	8	0,1,2,3,4,5,6,7	8^i	8^{-j}	3647.53
Decimal	10	0,1,2,3,4,5,6,7,8,9	10^i	10^{-j}	3874.65
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	16^i	16^{-j}	4DAC.54

2.2 Binary to Decimal Conversion

Conversion of binary to decimal (base-2 to base-10) numbers and back is an important concept to understand as the binary numbering system forms the basis for all computer and digital systems.

Examples:

- Convert binary $(10101)_2$ to decimal.

Solution:

$10101B = 1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0 = 16+4+1= 21$

- Convert binary $(10110.001)_2$ to decimal.

Solution:

First, multiply each bit of $(10110.001)_2$ with its respective positional weight, and then we add the products of all the bits with its weight.

$(10110.001)_2=(1\times2^4)+(0\times2^3)+(1\times2^2)+(1\times2^1)+(0\times2^0)+(0\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$

$(10110.001)_2=(1\times16)+(0\times8)+(1\times4)+(1\times2)+(0\times1)+ (0\times1/2)+(0\times1/4)+(1\times1/8)$

$(10110.001)_2=16+0+4+2+0+0+0+0.125$

$\mathbf{(10110.001)_2=(22.125\ )_{10}}$

2.3 Decimal to Binary Conversion

For converting decimal to binary, there are two steps required to perform, which are as follows:

i. Divide the given number by 2.
ii. Get the integer quotient for the next iteration.
iii. Get the remainder for the binary digit.
iv. Repeat the steps until the quotient is equal to 0.

Examples:

- Convert $(13)_{10}$ to binary:

Solution:

Division by 2	Quotient	Remainder
13/2	6	1(LSB)
6/2	3	0
3/2	1	1
1/2	0	1(MSB)

So, $(13)_{10}$ = $(1101)_2$

- Convert $(152.25)_{10}$ to binary

Solution:

Step 1: Divide the number 152 and its successive quotients with base 2.

Divide by 2	Quotient	Remainder
152/2	76	0 (LSB)
76/2	38	0
38/2	19	0
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1(MSB)

$(152)_{10}$=$(10011000)_2$

Step 2: Now, perform the multiplication of 0.25 and successive fraction with base 2.

Operation	Result	carry
0.25×2	0.50	0
0.50×2	0	1

$(0.25)_{10}=(.01)_2$

So, $(152.25)_{10}$ = $(10011000.01)_2$

2.4 Binary to Hexadecimal Conversion

The base numbers of binary and hexadecimal are 2 and 16, respectively. In a binary number, the pair of four bits is equal to one hexadecimal digit. There are also only two steps to convert a binary number into a hexadecimal number which are as follows:

i. In the first step, we have to make the pairs of four bits on both sides of the binary point. If there will be one, two, or three bits left in a pair of four bits pair, we add the required number of zeros on extreme sides.
ii. In the second step, we write the hexadecimal digits corresponding to each pair.

Examples:

- Convert binary $(1101100)_2$ to hexadecimal.

Solution:

Convert every 4 binary bits (starting from LSB) to hex digit i.e.

$$1101100_2 = \overbrace{0\ 1\ 1\ 0}^{6}\ \underbrace{1\ 1\ 0\ 0}_{C} = 6\ C = (6\ C)_{16}$$

Note: In the above example, the left side of the binary point has only three bits. To make it a complete pair of 4 bits, add one zero on the extreme side.

Table 2.2. Reference table for binary to hexadecimal equivalent

Binary ( )$_2$	**Hex Equivalent ( )$_{16}$**
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

- Convert binary **(10110101011.0011)$_2$** to hexadecimal.

Solution:

Firstly, make pairs of 4 bits on both sides of the binary point.

111 1010 1011.0011

On the left side of the binary point, the first pair has three bits. To make it a complete pair of four bits, add one zero on the extreme side i.e.

0111 1010 1011.0011

Now, write the hexadecimal digits correspond to each pair 4 bit pair i.e.

(0111 1010 1011.0011)$_2$= (7AB.3)$_{16}$

2.5 Decimal to Hexadecimal Conversion

For converting decimal to hexadecimal, there are two steps required to perform, which are as follows:

i. Divide the number by 16.
ii. Get the integer quotient for the next iteration.
iii. Get the remainder for the hex digit.
iv. Repeat the steps until the quotient is equal to 0.

Examples:

- Convert $(7562)_{10}$ to hex

Solution:

Division by 16	Quotient (integer)	Remainder (decimal)	Remainder (hex)
7562/16	472	10	A
472/16	29	8	8
29/16	1	13	D
1/16	0	1	1

So, $\mathbf{(7562)_{10} = (1D8A)_{16}}$

- Convert $(152.25)_{10}$ to hex

Solution:

First, divide the number 152 and its successive quotients with base 16.

Operation	Quotient	Remainder
152/16	9	8
9/16	0	9

$(152)_{10}=(98)_{16}$

Now, perform the multiplication of 0.25 and successive fraction with base 16.

Operation	Result	carry
0.25×16	0	4

$(0.25)_{10}=(4)_{16}$

Hence, the hexadecimal number of the decimal number $(152.25)_{10}$ is $\mathbf{(230.4)_{16}}$.

2.6 Hexadecimal to Decimal Conversion

A regular decimal number is the sum of the digits multiplied with power of 10. For example, $(137)_{10}$ is equal to each digit multiplied with its corresponding power of 10, i.e.

$(137)_{10} = 1\times10^2+3\times10^1+7\times10^0 = 100+30+7.$

Likewise, hex numbers are read the same way, but each digit counts power of 16 instead of power of 10.

For hex number with n digits, i.e. dn-1 ... d3 d2 d1 d0

Multiply each digit of the hex number with its corresponding power of 16 and sum:

Hence, decimal $= dn\text{-}1\times16^{n-1} + ... + d_3\times16^3 + d_2\times16^2 + d_1\times16^1+d_0\times16^0$

Examples:

- Convert $(E7A9)_{16}$ to decimal

Solution

E7A9 in base 16 is equal to each digit multiplied with its corresponding 16^n i.e.

$E7A9_{16} = 14\times16^3+7\times16^2+10\times16^1+9\times16^0 = 57344+1792+160+9 = 59305_{10}$

Hence, $\mathbf{(E7A9)_{16} = (59305)_{10}}$

- Convert $(3B)_{16}$ to decimal

Solution

3B in base 16 is equal to each digit multiplied with its corresponding 16^n i.e.

$3B_{16} = 3\times16^1+11\times16^0 = 48+11 = 59_{10}$

Hence, $\mathbf{(3B)_{16} = (59)_{10}}$

- Convert $(152A.25)_{16}$ to decimal

Solution

First, multiply each digit of **152A.25** with its respective positional weight, and then we add the products of all the bits with its weight.

$\mathbf{(152A.25)_{16}}=(1\times16^3)+(5\times16^2)+(2\times16^1)+(A\times16^0)+(2\times16^{-1})+(5\times16^{-2})$

$\mathbf{(152A.25)_{16}}=(1\times4096)+(5\times256)+(2\times16)+(10\times1)+(2\times16^{-1})+(5\times16^{-2})$

$\mathbf{(152A.25)_{16}}=4096+1280+32+10+(2\times1/16)+(5\times1/256)$

$\mathbf{(152A.25)_{16}}=5418+0.125+0.125$

$\mathbf{(152A.25)_{16}}=5418.14453125_{10}$

Hence, $\mathbf{(152A.25)_{16}=(5418.14453125)_{10}}$

Table 2.3. Reference table for Hexadecimal to Decimal equivalent

Hexadecimal (base 16)	**Decimal (base 10) Equivalent**
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

2.7 Octal to Decimal Conversion

The process starts from multiplying the digits of octal numbers with its corresponding positional weights. And lastly, we add all those products.

Example:

- Convert $(152.25)_8$ to decimal

Solution

First, multiply each digit of **152.25** with its respective positional weight, and then we add the products of all the bits with its weight.

$(152.25)_8$ $=(1\times8^2)+(5\times8^1)+(2\times8^0)+(2\times8^{-1})+(5\times8^{-2})$

$(152.25)_8$ $=64+40+2+(2\times1/8)+(5\times1/64)$

$(152.25)_8$ $=64+40+2+0.25+0.078125$

Hence, **$(152.25)_8$**$=106.328125$

2.8 Binary Subtraction using Complements (1s & 2s)

Binary Subtraction using 1's & 2's complement

1's complement :- To Calculate the 1's complement of a binary number just flip each value (bit) of the original binary no.

i.e $0 \rightarrow 1$, $1 \rightarrow 0$

Ex 0 1 0 1 1 1

$\downarrow$ 1's complement

1 0 1 0 0 0

2's complement :- Just add '1' to 1's complement of that number.

Ex $010111 \xrightarrow[\text{complement}]{\text{2's}}$ 1's complement + 1

101000 + 1

$\downarrow$

101001

(2's complement).

NOTE :-

1's & 2's Complement are taken from for only −ve number.

SUBTRACTION USING 1's COMPLEMENT

Ex:- Perform $(9)_{10} - (4)_{10}$ using 1's complement.

$$\boxed{\begin{aligned}&9-4\\&=5\end{aligned}}$$

[AKTU QUESTION].

$(\overset{8\,4\,2\,1}{1001})_2 - (\overset{8\,4\,2\,1}{0100})_2$

Step 1: Take 1's complement of -ve number i.e (0100)

$$\begin{array}{c} 0100 \\ \downarrow \\ 1011 \end{array}$$

Step 2: Then Add first number with 1's complement of second no.

$$\begin{array}{r} 1001 \\ 1011 \\ \hline \boxed{1}\ 0100 \end{array}$$

final carry.

Remember

$0+0 \Rightarrow 0$

$0+1 = 1$

$1+0 = 1$

$1+1 = 1\,0$ Carry.

Step 3: Add final carry to the above result-

$$\begin{array}{r} 0100 \\ +1 \\ \hline (0101)_2 \end{array}$$

Ans.

Answer is positive & True form.

Ex - Perform $(1011001)_2 - (1101010)_2$ using 1's Complement

Step 1: (1101010)

↓ 1's complement

0010101

89 - 106 = 17

Step 2: Add both number

$$\begin{array}{r} 1011001 \\ 0010101 \\ \hline \boxed{0}\ 1101110 \end{array}$$

final carry.

Answer is -ve & 1's complement form

Step 3: Invert all the bits

$$\begin{array}{c} 1101110 \\ \downarrow \\ (0010001)_2 \end{array}$$

Ans

SUBTRACTION USING 2's COMPLEMENT

Ex: Perform following subtraction using 2's complement method.

$(1101)_2 - (1001)_2$ $\quad 13 - 9 = 4$

Step 1: Take 2's complement of -ve number (1001)

$$1001 \downarrow 0110 \text{ (1's complement)} \downarrow 0110 + 1 \text{ (Add 1)} \downarrow$$

2's complement : $(0111)_2$

Step 2: Add both number $(1101)_2 + (0111)_2$

$$\begin{array}{r} 1\,1\,0\,1 \\ 0\,1\,1\,1 \\ \hline \boxed{1}\ \boxed{0\,1\,0\,0} \end{array}$$

final carry $\leftarrow \boxed{1}$ $\quad \boxed{0100} = 4 \rightarrow$ Ans is +ve & True form.

Ex Perform binary subtraction using 2's complement method :-

$(10011)_2 - (1101)_2$ $\quad 19 - 13 = 6$

(Make both number of equal digits)

1. $(01101)_2 \longrightarrow$ 2's complement

$\downarrow$

10010 (1's complement)

$\downarrow$

$10010 + 1 \longrightarrow 10011$ (Add 1)

2. Add both the no.

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

final carry $\boxed{1}$ $\quad \boxed{00110} \rightarrow$ Ans is +ve (6) & true form.

EX Perform binary subtraction using 2's complement method – (28 – 42) = 14

$(011100)_2 - (101010)_2$

1. $(101010)_2 \longrightarrow$ 2's complement

$\downarrow$

010101

$\downarrow$

010101 + 1 = 010110

2. Add

```
      011100
    + 010110
final [0] [110010] → Ans is –ve & its 2's complement form.
carry
```

$\downarrow$

3. Invert all bits + 1 : 001101 (1's complement)

+1

Ans $\boxed{001110}$ 2's complement (14)

NOTE: final carry 1 : Answer is +ve & true form.

0 : Answer is –ve & 2's complement form.

3. Basic and Universal Logic Gates

3.1 Introduction

Logic gates are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output. The relationship between the input and the output is based on certain logic. Based on this, logic gates are named as AND gate, OR gate, NOT gate etc. Moreover, NAND and NOR gates are called universal logic gates.

3.2 Logic Addition

The possible input and output combinations may be arranged as follows:

0 + 0 =0
0 + 1 =1
1 + 0 =1
1 + 1 =1

3.3 Logical Multiplication

We can define the "**.**" (logical multiplication) symbol or AND operator by listing all possible combinations for (input) variables let say X and Y and the resulting (output) value of X**.**Y will be as follows:

X **Y**

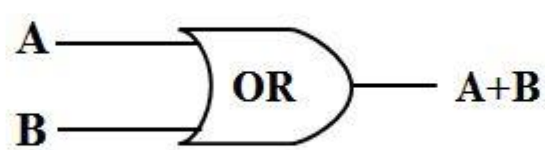
0.0 = 0
0.1 = 0
1.0 = 0
1.1 = 1

3.4 Logic Gates

- A logic gate is defined as an electronics circuit with two or more input signals and one output signal.
- AND, OR & NOT gates are called basic logic gates.
- NAND and NOR gates are called universal logic gates.

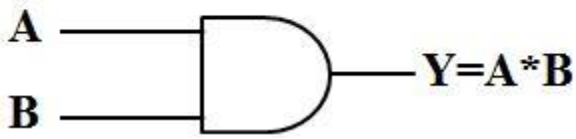
3.4.1 OR Gate

Inputs		**Output**
A	**B**	**Y= A*B**
0	0	0
0	1	1
1	0	1
1	1	1



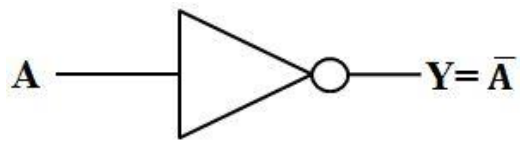
3.4.2 AND Gate

Inputs		Output
A	**B**	**Y= A*B**
0	0	0
0	1	0
1	0	0
1	1	1



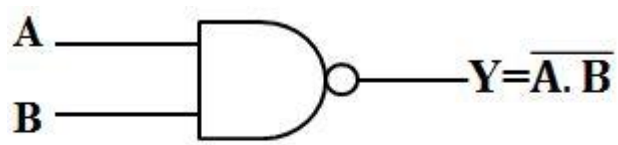
3.4.3 NOT Gate

Input	Output
A	**Y**
0	1
1	0



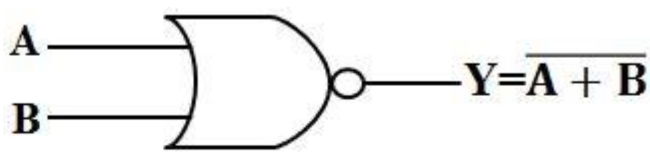
3.4.4 NAND Gate

Inputs		Output
A	**B**	**$Y=\overline{A.B}$**
0	0	1
0	1	1
1	0	1
1	1	0



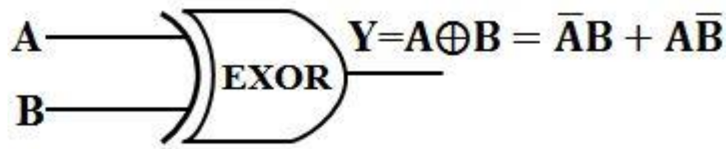
3.4.5 NOR Gate

Inputs		Output
A	**B**	**$Y=\overline{A+B}$**
0	0	1
0	1	0
1	0	0
1	1	0



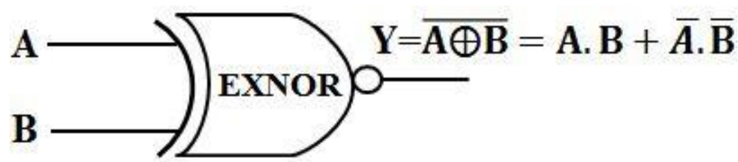
3.4.6 EX-OR Gate

Inputs		Output
A	**B**	**Y=A⊕B**
0	0	0
0	1	1
1	0	1
1	1	0



3.4.7 EX-NOR Gate

Inputs		Output
A	**B**	**Y=$\overline{A\oplus B}$**
0	0	1
0	1	0
1	0	0
1	1	1

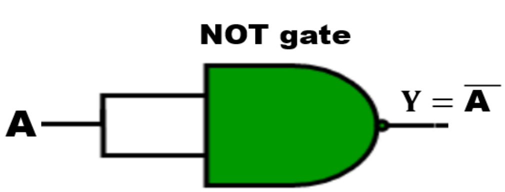


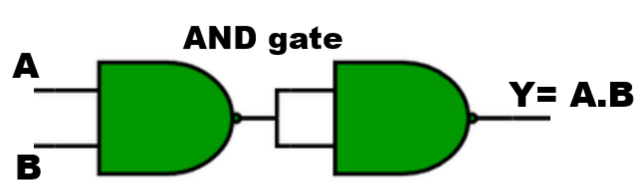
3.5 Realization of Logic Gate Using Universal gates

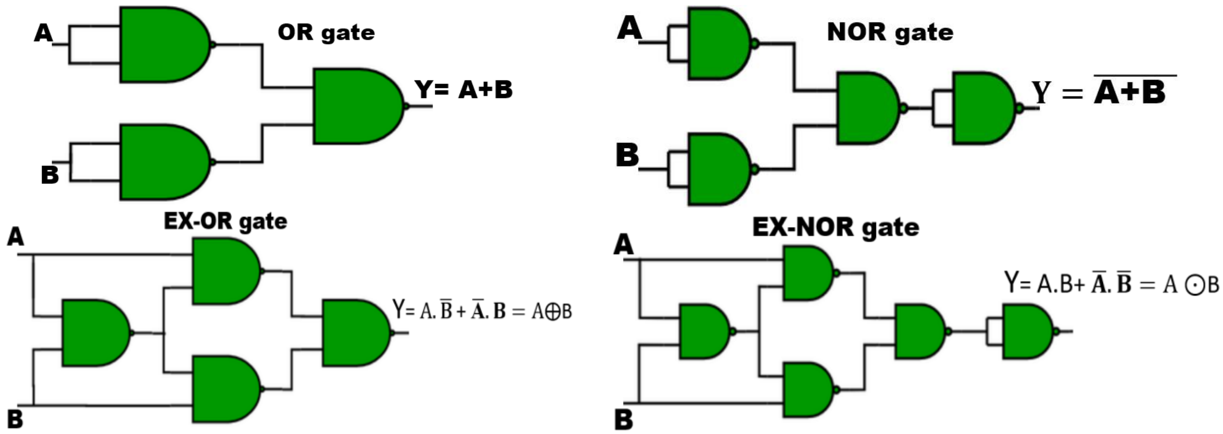
In Boolean Algebra, the **NAND** and **NOR** gates are called **universal gates** because any digital circuit can be implemented by using any one of these two i.e., any logic gate can be created using NAND or NOR gates only.

3.5.1 Implementation of NOT, AND, OR, NOR, NAND, EX-OR, & EX-NOR Gate Using Universal gates

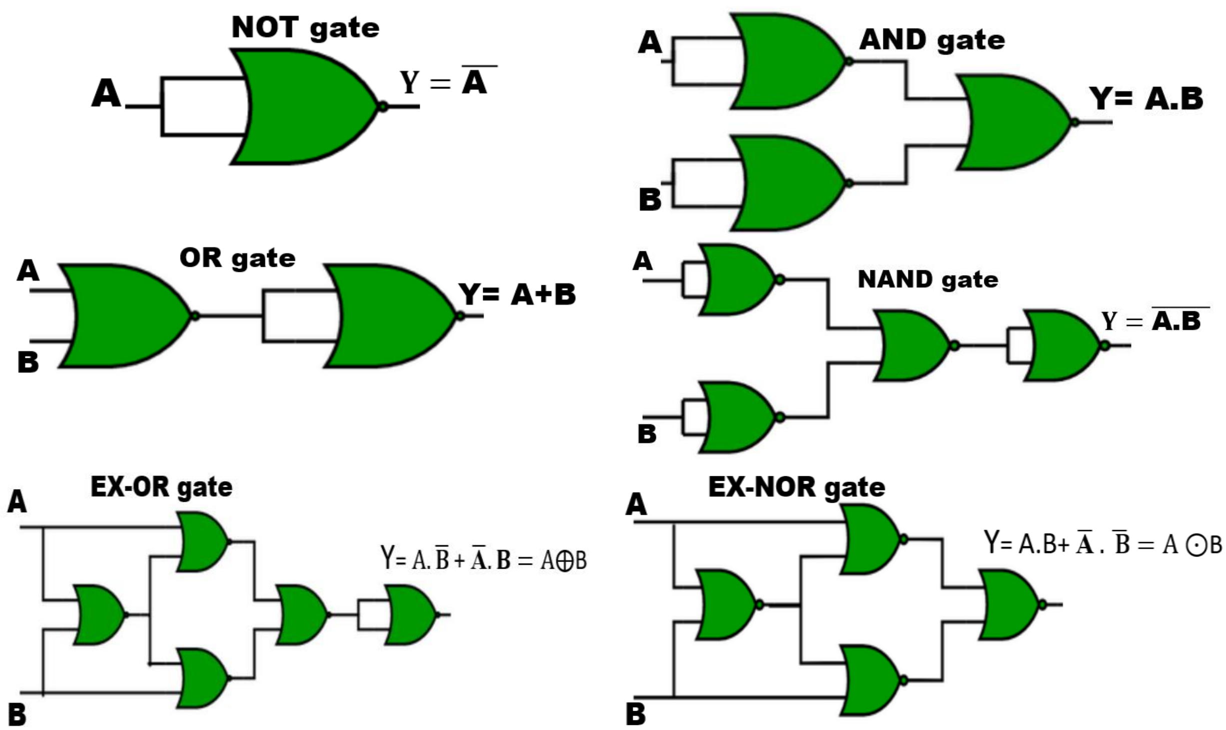
A. *Implementation Using Minimum Number of NAND Gates*







B. Implementation Using Minimum Number of NOR Gates



4. Boolean Algebra Simplification of Boolean Function

4.1 Boolean Algebra

An algebra which deals with the binary number system called Boolean Algebra. Here, we use logic gate concept i.e.

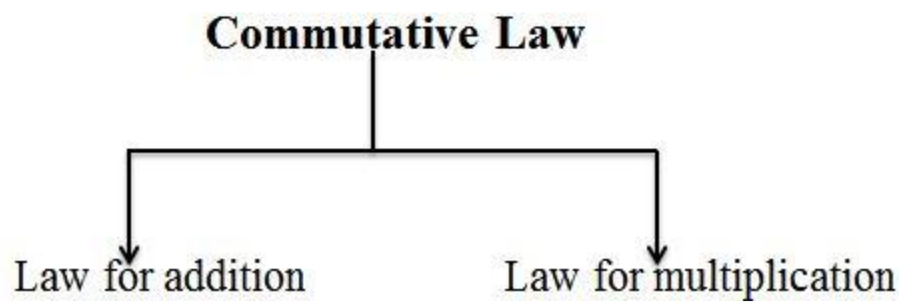
' + ' Addition $\longrightarrow$ **OR Gate**

' ● ' Multiplication ⟶ **AND Gate**

4.2 Laws of Boolean Algebra

- ➢ Commutative Law
- ➢ Associative Law
- ➢ Distributive Law

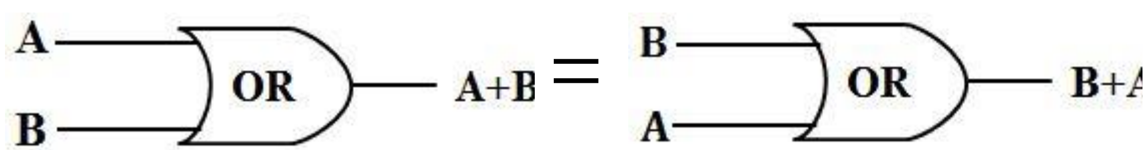
➢ Commutative Law



a) Law for addition

It is independent of order, this means if we change the sequence of variables, result will be same i.e.

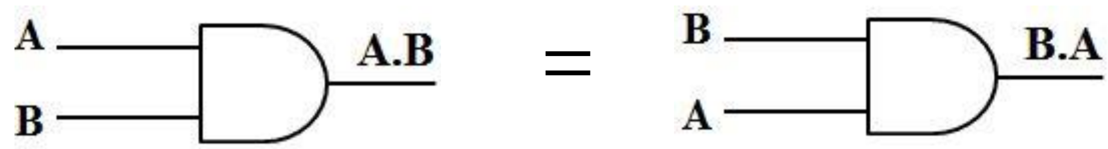
$$A+B = B+A$$



b) Law for multiplication

It is also independent of order. It means changing variable sequence while AND operation, the final result will not affect i.e.

$$A.B = B.A$$

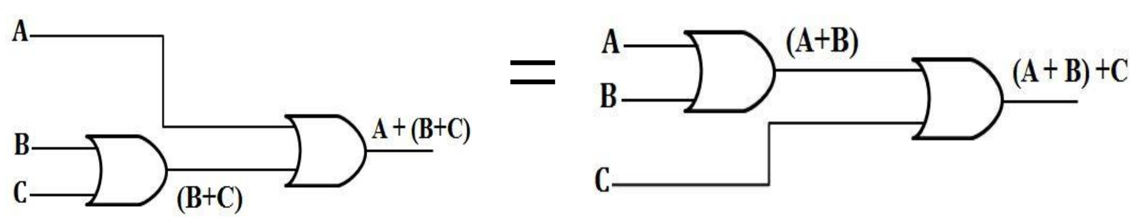


- **Associative Law**

a) Law for addition

Associative law for addition can be justified as follows:

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



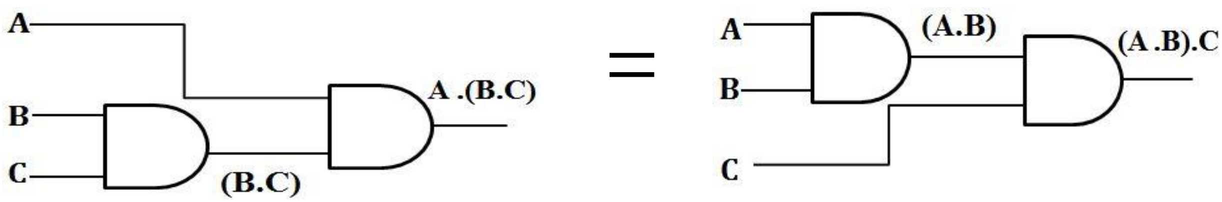
<u>Verification Table</u>

A	B	C	(B+C)	A+(B+C)	(A+B)	(A+B)+C
0	0	0	0	**0**	0	**0**
0	0	1	1	**1**	0	**1**
0	1	0	1	**1**	1	**1**
0	1	1	1	**1**	1	**1**
1	0	0	0	**1**	1	**1**
1	0	1	1	**1**	1	**1**
1	1	0	1	**1**	1	**1**
1	1	1	1	**1**	1	**1**

b) Law for multiplication

Associative law for multiplication can be justified as follows:

$$A \bullet (B \bullet C) = (A \bullet B) \bullet C$$



<u>**Verification Table**</u>

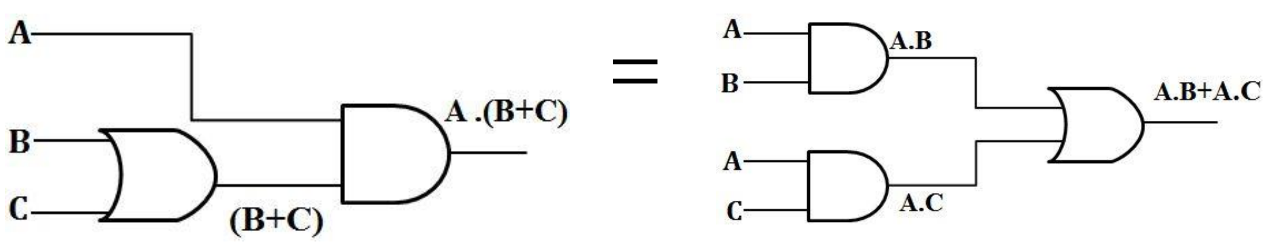
A	B	C	(B.C)	A.(B.C)	(A.B)	(A.B).C
0	0	0	0	**0**	0	**0**
0	0	1	0	**0**	0	**0**
0	1	0	0	**0**	0	**0**
0	1	1	1	**0**	0	**0**
1	0	0	0	**0**	0	**0**
1	0	1	0	**0**	0	**0**
1	1	0	0	**0**	1	**0**
1	1	1	1	**1**	1	**1**

➢ Distributive Law

a) Law using AND operator

Distributive law using AND operator can be justified as follows:

$$A \bullet (B + C) = A \bullet B + A \bullet C$$



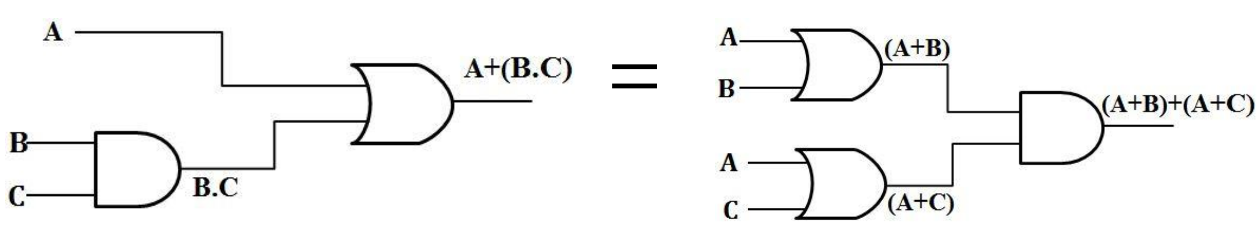
Verification Table

A	B	C	(B+C)	A.(B+C)	(A.B)	(A.C)	(A.B)+(A.C)
0	0	0	0	**0**	0	0	**0**
0	0	1	1	**0**	0	0	**0**
0	1	0	1	**0**	0	0	**0**
0	1	1	1	**0**	0	0	**0**
1	0	0	0	**0**	0	0	**0**
1	0	1	1	**1**	0	1	**1**
1	1	0	1	**1**	1	0	**1**
1	1	1	1	**1**	1	1	**1**

b) Law using OR operator

Distributive law using OR operator can be justified as follows:

$$A + (B \bullet C) = (A+B) \bullet (A+C)$$



Verification Table

A	B	C	(B.C)	A.(B+C)	(A+B)	(A+C)	(A+B).(A+C)
0	0	0	0	**0**	0	0	**0**
0	0	1	0	**0**	0	1	**0**
0	1	0	0	**0**	1	0	**0**
0	1	1	1	**1**	1	1	**1**
1	0	0	0	**1**	1	1	**1**

1	0	1	0	**1**	1	1	**1**
1	1	0	0	**1**	1	1	**1**
1	1	1	1	**1**	1	1	**1**

4.3 Duality in Boolean Algebra

Starting with a Boolean relation, we can derive another Boolean relation by-

i. Changing each OR (+) sign to an AND (.) sign
ii. Changing each AND (.) sign to an OR (+) sign.
iii. Complementary each 0 and 1

For example:

A + 0 = A

The dual relation is A**.**1 = A

Since, A (B + C) = AB + AC (by distributive law).

Its dual relation is A + B C = (A + B) (A + C)

4.4 De Morgan's Theorem

This theorem can be justified as follows:

a) $\overline{A + B} = \overline{A} \,.\, \overline{B}$

b) $\overline{A \,.\, B} = \overline{A} + \overline{B}$

4.5 Boolean Algebra Theorems

Name	AND Form	OR Form
Identity Law	A.1 = A	A+0 = A
Null Law	A.0 = 0	A+1 = 1
Idempotent Law	A.A = A	A+A = A
Inverse Law	A. $\overline{A}$ = 0	A+$\overline{A}$ = 1
Commutative Law	AB=BA	A+B=B+A
Associative Law	(AB)C= A(BC)	(A+B)+C= A+(B+C)
Distributive Law	A+BC=(A+B)(A+C)	A(B+C) = AB+AC
Absorption Law	A(A+B)=A	A+AB=A
De Morgan's Law	$\overline{A.B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}.\overline{B}$

4.6 Simplification of Boolean expressions

In this section, we will discuss some working examples to simplify different Boolean expressions.

Example 1:

Simplify the Boolean expression $AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD$.

Solution:

Given, $AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD$

$\Longrightarrow AB\overline{C}(\overline{D} + D) + ABC\,(\overline{D} + D)$

$\Longrightarrow AB\overline{C}.1 + ABC.1$; (As we know $D + \overline{D} = 1$)

$\Longrightarrow AB\overline{C} + ABC$

$\Longrightarrow AB(\overline{C} + C)$

Hence, the answer is **AB**

Example 2:

Simplify the Boolean expression$AB(\overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + \overline{A}\,B\,C)$.

Solution:

Given, $AB(\overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + \overline{A}\,B\,C)$

$\Longrightarrow A\,\overline{A}\,B\,B\,\overline{C} + A\,A\,B\,\overline{B}\,\overline{C} + \overline{A}\,A\,B\,B\,C$

$\Longrightarrow 0 + 0 + 0$; (As we know $A\,\overline{A} = 0$)

Hence, the final answer is **0**

Example 3:

Simplify the Boolean expression $(\overline{A}\,\overline{B} + A\,B + \overline{A}\,B\,)$.

Solution:

Given, $(\bar{A}\,\bar{B} + A\,B + \bar{A}\,B\,)$

$\Longrightarrow \bar{A}\,\bar{B} + B(A + \bar{A})$

$\Longrightarrow \bar{A}\,\bar{B} + B.1$

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

$\Longrightarrow (B + \bar{A}\,)(B + \bar{B})$

$\Longrightarrow (B + \bar{A}\,).1$

Hence, the answer is $\mathbf{\bar{A} + B}$

Example 4:

Simplify the Boolean expression $(A + B)(A + \bar{B})$.

Solution:

Given, $(A + B)(A + \bar{B})$

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

$\Longrightarrow A.A + A(\bar{B} + B) + 0$

$\Longrightarrow A.A + A(\bar{B} + B)$

$\Longrightarrow$ A + A

Hence, the answer is **A**

Example 5:

Simplify the Boolean expression $AB + A(C\,D + C\,\bar{D})$.

Solution:

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

$\Longrightarrow AB + AC(D + \bar{D})$

$\Longrightarrow AB + AC.(1)$

Hence, the answer is $\mathbf{A(B+C)}$

Example 6:

Simplify the Boolean expression $(B\,\overline{C} + \overline{A}\,D)(A\,\overline{B} + C\,\overline{D})$.

Solution:

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

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

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

$0 + 0 + 0 + 0 = 0$

Hence, the answer is **0**

4.7 SOP and POS Representation for Logic Expressions

Sum of products (SOP) form:-

Let the boolean function $F = AB + AC + BC$

Sum

product terms

Sum of Products (SOP) form

It is in the forms of sum of three terms AB, AC and BC with each individual terms is product of two variables

therefore, such expressions are known as expression in SOP form.

The sums and products in the SOP forms are not the actual addition or multiplication. In fact, they are the OR and AND functions

Some more examples of SOP forms:-

(i) $F_1 = ABC + BCD + AB\bar{D}$

(ii) $F_2 = xy + \bar{x}\bar{y} + x\bar{y}$

(iii) $f_2 = \bar{P}\bar{Q} + PQR + PQ\bar{R}$

Products of Sum (POS) form:-

Let the boolean function $F = (A+B)\cdot(B+C)\cdot(A+C)$

products

Sum terms

Products of Sum (POS) form

It is in the form of product of three terms (A+B) (B+C) and (A+C) with each terms is in the form of sum of two variables.

Some more ~~exp~~. examples of POS forms:-

$$F_1 = (A+\bar{B}+\bar{C})\cdot(A+B)\cdot(\bar{A}+C)$$

$$F_2 = (\bar{x}+\bar{y})\cdot(\bar{x}+y+z)$$

$$F_3 = (P+R)\cdot(P+\bar{Q})\cdot(\bar{P}+R)$$

Canonical or standard form:-

Standard SOP form $f = ABC + A\bar{B}\bar{C} + \bar{A}BC$

Each product term consists of all the literals in the complimented or uncomplimented form

Standard POS form $F = (A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

Each sum term consists of all the literals in the complimented or uncomplimented form.

Some more examples of standard or canonical SOP and POS form

	Logical expression	Types of expression
1.	$f = AB + AB\bar{C} + \bar{A}BC$	Non standard SOP
2.	$F = AB + A\bar{B} + \bar{A}\bar{B}$	standard SOP
3.	$F = (\bar{A}+B)(A+B)(A+\bar{B})$	standard POS
4.	$F = (\bar{A}+B)(A+B+C)$	Non standard POS
5.	$F(A,B,C) = ABC + \bar{A}\bar{B}C + \bar{A}B\bar{C}$	Standard SOP
6.	$F(A,B,C) = BC + ABC + AB$	Non Standard SOP

Minterm and Maxterm for three variables:-

x	y	z	Minterm Term	Minterm Designation	Maxterm Term	Maxterm Designation
0	0	0	$\bar{x}\bar{y}\bar{z}$	m_0	$x+y+z$	M_0
0	0	1	$\bar{x}\bar{y}z$	m_1	$(x+y+\bar{z})$	M_1
0	1	0	$\bar{x}y\bar{z}$	m_2	$(x+\bar{y}+z)$	M_2
0	1	1	$\bar{x}yz$	m_3	$(x+\bar{y}+\bar{z})$	M_3
1	0	0	$x\bar{y}\bar{z}$	m_4	$(\bar{x}+y+z)$	M_4
1	0	1	$x\bar{y}z$	m_5	$(\bar{x}+y+\bar{z})$	M_5
1	1	0	$xy\bar{z}$	m_6	$(\bar{x}+\bar{y}+z)$	M_6
1	1	1	xyz	m_7	$(\bar{x}+\bar{y}+\bar{z})$	M_7

Ex: Express the given table in minterm (SOP) and Maxterm (POS)

Decimal no	x	y	z	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

where output (F) is 1, these are minterm (SOP)

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

or $F = m_1 + m_4 + m_7$

$$F = \Sigma m\{1, 4, 7\}$$

where output (F) is 0, these are maxterm (POS)

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

or $F = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$

or $F = \Pi M\{0, 2, 3, 5, 6\}$

Steps to convert Non standard form to standard form:-

1. Find missing literals in each terms.
2. And each product term having missing literals like $(x+\bar{x})$
3. Expand the terms by distributive law.
4. write the repeatitive terms only once.

Ex: Convert the given expression in standard form (SOP)

$$F(A, B, C) = AC + AB + BC$$

Steps.1 Missing literals $\underset{\downarrow}{AC}\to B$, $\underset{\downarrow}{AB}\to C$, $\underset{\downarrow}{BC}\to A$

Step.2 $F(A, B, C) = AC(B+\bar{B}) + AB(C+\bar{C}) + BC(A+\bar{A})$

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

$$\boxed{F(A,B,C) = ABC + A\bar{B}C + AB\bar{C} + \bar{A}BC}$$ Standard (SOP)

For pos form:-

Step 1 Find the missing literals

Step 2 Add that one with the required term as $x \cdot \bar{x}$

Step 3 Expand the expression.

Step 4 write the repeatitive terms once.

Ex: convert the given expression into the standard pos form.

$$F(A,B,C) = (A+B)\,(B+C)\,(C+A)$$

missing literals $\quad C \quad A \quad B$

Add the missing literals $= (A+B+C\bar{C})\,(B+C+A\bar{A})\,(C+A+B\bar{B})$

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

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

standard pos form.

Conversion sop to pos and pos to sop:-

SOP $\quad F = \Sigma_m \{0,1,2,5\}$

POS $\quad F = \Pi_M \{3,4,6,7\}$

Conversion from pos to SOP:-

Ex: Convert F_2 from pos to sop function.

$$F_2 = x(\bar{y}+z)(\bar{x}+\bar{y}+\bar{z})$$

Take compliment

$$\bar{F_2} = \overline{x(\bar{y}+z)(\bar{x}+\bar{y}+\bar{z})}$$

$$\bar{F}_2 = \bar{x} + (y\cdot\bar{z}) + (xyz)$$

$$\boxed{F_3 = \bar{F_2} = \bar{x} + y\bar{z} + xyz}$$ sop representation of F_2 (Ans)

Ex: convert the function form pos to sop.

$$F = \Pi_M \{0,6,7\}$$

Take Compliment $\quad \bar{F} = [\Pi_M \{0,6,7\}]'$

$$\bar{F} = \Sigma_m (1,2,3,4,5) \quad \text{sop form.}$$

Conversion from SOP to POS:-

Ex: convert the following function from SOP to POS

$$F_1 = \bar{y} + xy + \bar{x} y \bar{z}$$

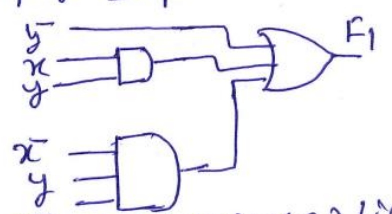
take compliment of F_1

$$F_2 = \bar{F_1} = \overline{(\bar{y} + xy + \bar{x} y \bar{z})}$$

$$\boxed{F_2 = \bar{F_1} = y \cdot (\bar{x} + \bar{y}) \cdot (x + \bar{y} + z)}$$ Ans

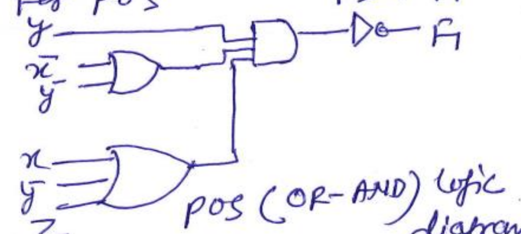
POS representation of F_1

For SOP



SOP (AND–OR) Logic diagram

For POS



POS (OR–AND) Logic diagram

5. Karnaugh map (K-Map)

5.1 Introduction

Karnaugh map or K-map is a map of a function used in a technique used for minimization or simplification of a Boolean expression. It results in less number of logic gates and inputs to be used during the fabrication.

5.2 Rules of Minimization in K-Map

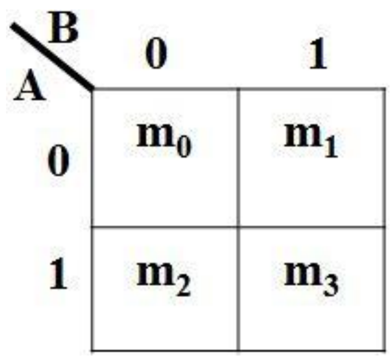
- While grouping, you can make groups of 2^n number where n=0, 1, 2, 3…
- You can either make groups of 1's or 0's but not both.
- Grouping of 1's lead to Sum of Product form and Grouping of 0's lead to product of Sum form.
- While grouping, the groups of 1's should not contain any 0 and the group of 0's should not contain any 1.'
- Groups can be made vertically and horizontally but not diagonally.
- Groups made should be as large as possible even if they overlap.
- All the like term should be in a group even if they overlap.
- Uppermost & lowermost squares can be made into a group together as they are adjacent (1-bit difference). Same goes for the corner squares.
- Each group represents a term in the Boolean expression. Larger the group, smaller and simple the term.
- The product of those literals that remains unchanged in a single group makes the term of the expression.
- Don't care "x" should also be included while grouping to make a larger possible group.

5.3 2-Variable K-Map

2 variables have $2^n = 2^2 = 4$ minterms. Therefore, there are 4 cells (squares) in 2 variable K-map for each minterm.

Consider A & B as two variables.

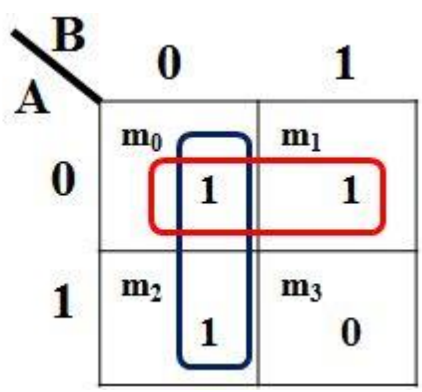
Grouping in 2 variables K-map is easy as there are few squares shown below



Example of 2 Variable K-Map-

F = $\sum (m_0, m_1, m_2) = \bar{A}\bar{B} + \bar{A}B + A\bar{B}$

Now, K-map from Truth table



Hence, the simplifies expression will be the sum of these two terms as given below,

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

5.4 3-Variable K-Map

3 variables make $\mathbf{2^n=2^3=8}$ min terms, so the K- map of 3 variables will have 8 squares(cells) as shown in the figure given below.

AB\C	0	1
00	m_0	m_1
01	m_2	m_3
11	m_6	m_7
10	m_4	m_5

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

Moreover, we can make groups of 2, 4 & 8 cells having same 1s or 0s.

AB\C	0	1
00	1	0
01	1	0
11	0	1
10	0	1

AB\C	0	1
00	1	0
01	0	0
11	0	0
10	1	0

AB\C	0	1
00	0	0
01	1	1
11	1	1
10	0	0

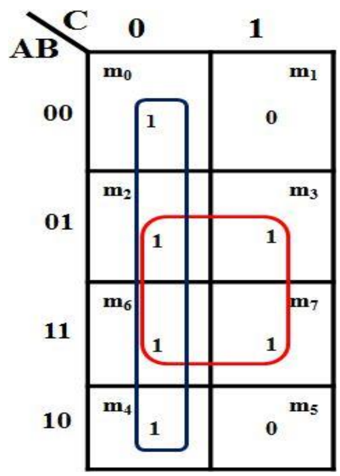
AB\C	0	1
00	1	0
01	1	0
11	1	0
10	1	0

Example of 3 Variable K-Map:

Simplify the given function using K-map.

$$F(A, B, C) = \sum (m_0, m_2, m_3, m_4, m_6, m_7)$$

Solution:

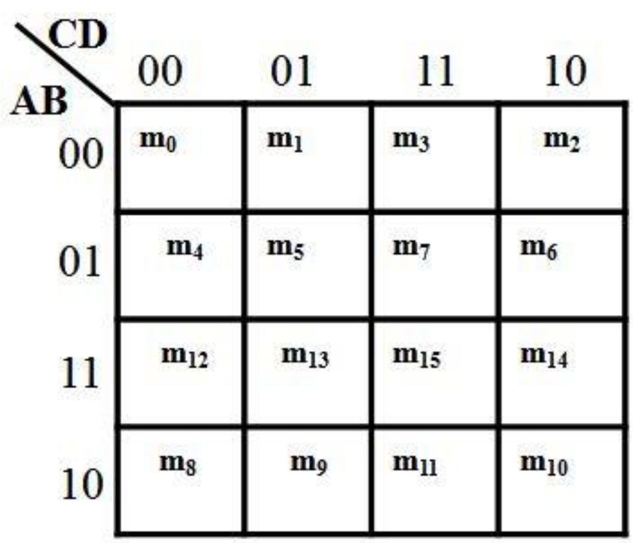


The sum of these two terms will make the simplified expression of the function as given below.

$$\mathbf{F = B + \overline{C}}$$

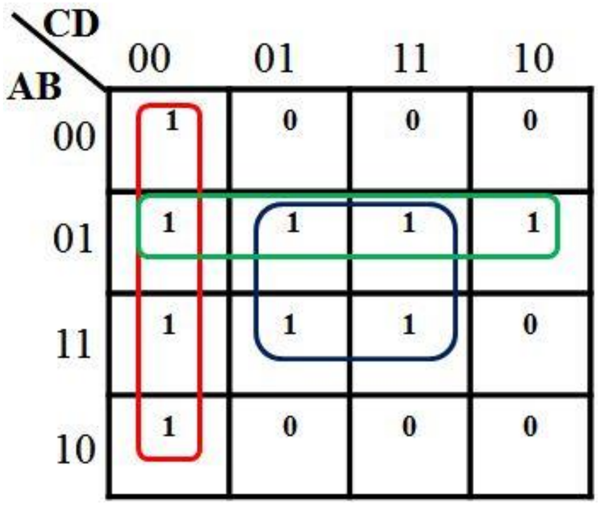
5.5 4-Variable K-Map

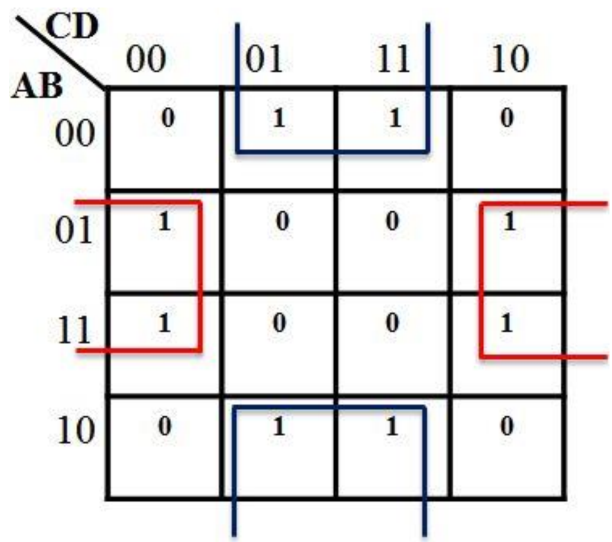
Four variables have $\mathbf{2^n=2^4=16}$ minterms. So, a 4-variable k-map will have 16 cells as shown in the figure given below.



Each cell (min term) represents the variables in front of the corresponding row & column.

Some examples of grouping in 4-variable k-map is given below:



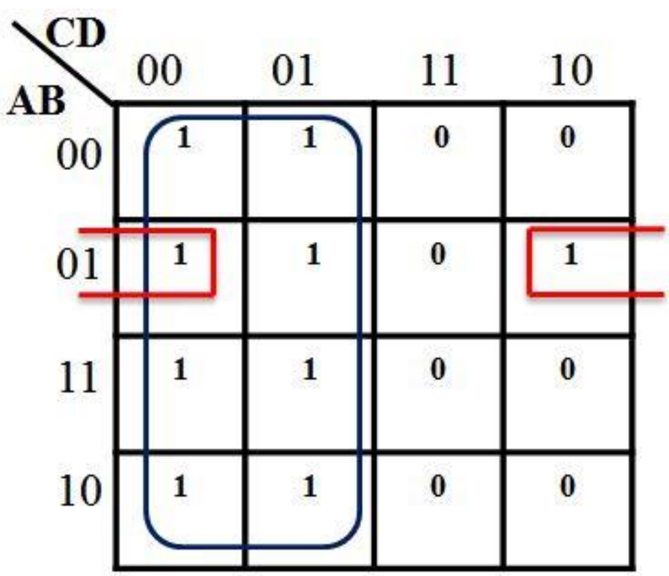


Example of 4 Variable K-Map:

Simplify the given function using K-map.

$$F(A,B,C,D) = \sum(m_0, m_1, m_4, m_5, m_6, m_8, m_9, m_{12}, m_{13})$$

Solution:



Group of 8 will give a term of 1 literal that remains unchanged i.e. $\overline{C}$

So the expression will F = $\mathbf{\overline{C} + \overline{A}B\overline{D}}$

5.6 5-Variables K-Map

5 variables have 32 min terms, which mean 5 variable k- map has 32 squares (cells).

$\bar{A}$

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

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}

A 5-variable K-map is made using two 4-variable K-maps. Consider 5 variables A,B,C,D,E and their 5 variable K-map is given below.

These both 4-variable Karnaugh map together represents a 5-variable K-map for variable A,B,C,D,E. Notice variable A over the top of each 4-variable K-map. For A=0, the left K-map is selected and right map for A = 1.

Each corresponding squares (cells) of these two 4-variable K-maps are adjacent. Visualize these both K-maps on top of each other. m_0 is adjacent to m_{16}, so is m_1 to m_{17} so on until the last square.

The rule (method) of grouping is same for each of the 4-variable k-maps. However, you also need to check the corresponding cells in both K-maps as well. A few example of grouping is given below.

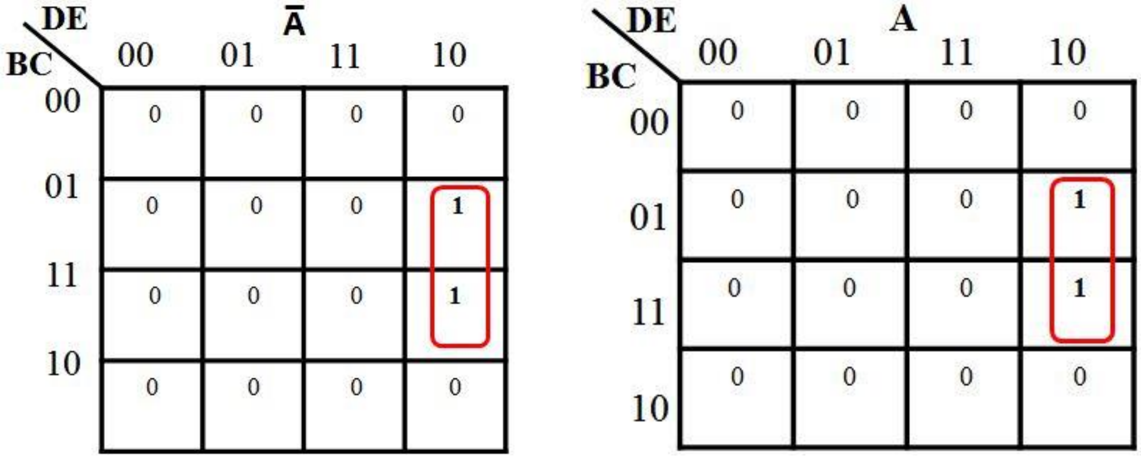
A

BC \ DE	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	0	1	0
10	1	0	0	0

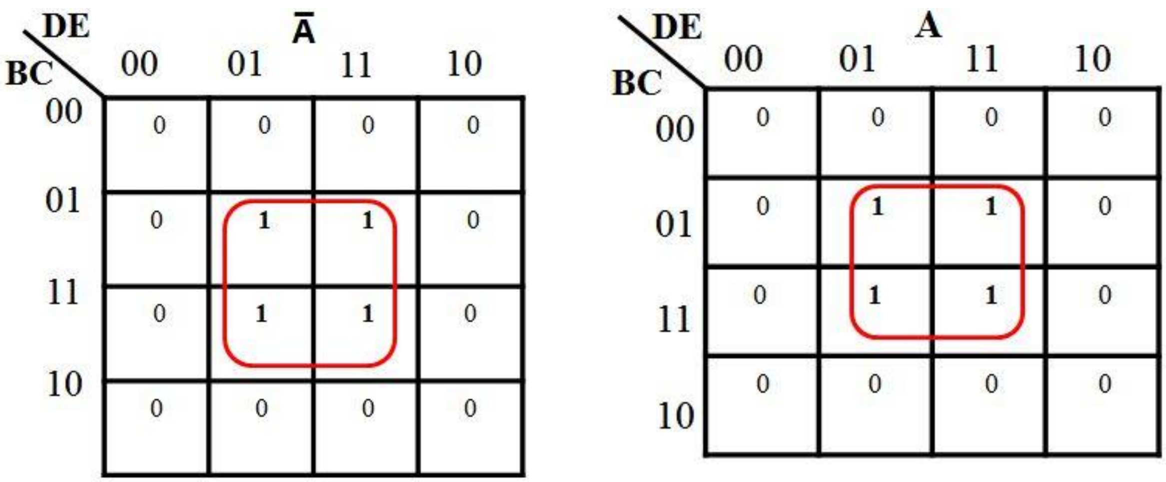
$\bar{A}$

BC \ DE	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	0	1	0
10	1	0	0	0

Groups of 2



Groups of 4



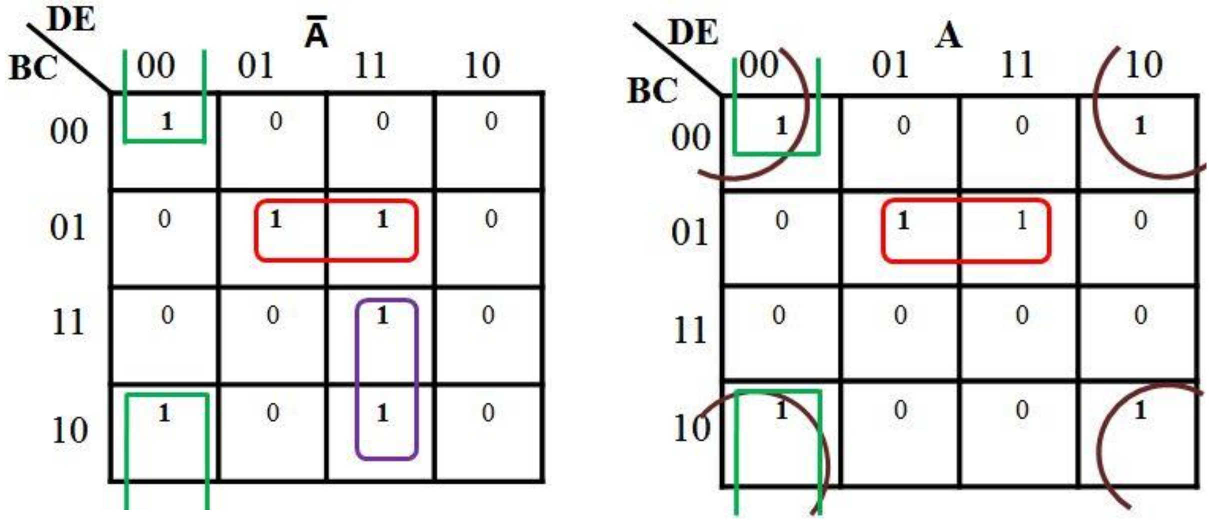
Groups of 8

Example of 5 Variables K-Map

Simplify the given function using K-map.

$$F(A, B, C, D, E) = \sum (m_0, m_2, m_5, m_7, m_8, m_{10}, m_{16}, m_{21}, m_{23}, m_{24}, m_{27}, m_{31})$$

Solution:



This is the 5-variable k-map for the function given above. There are four groups made in this K-map. Each group has a different color to differentiate between them.

The red color group is a group of 4 min terms made between both 4-variable k-maps because they are adjacent cells and it overlaps the green group.

The yellow group is also a group of 4 min terms made between adjacent cells of the 4-variable k-maps.

The green group is a group of 4 min terms made in the left 4-variable k-map. The blue group is of 2 min-terms made in the right 4-variable k-map because there are no common adjacent cells in the other k-map.

Green color group of 4 min term will produce the term $\bar{A}\bar{C}\bar{E}$. The individual 4-variable K-map will produce $\bar{C}\bar{E}$ as they are not changing in the group but variable A should also be taken into account because this individual 4-variable k-map is being represented by $\bar{A}$.

The red color group will produce $\bar{C}\bar{D}\bar{E}$. This group is made between both K-maps which means variable A changes and in individual K-map, B changes so these both variables will be eliminated from the term. Only $\bar{C}\bar{D}\bar{E}$ remains unchanged in this group.

The yellow group will produce $\bar{B}CE$ because these literals are not changing in this group.

Blue group of 2 min terms will produce the term ABDE as they remain unchanged in this group.

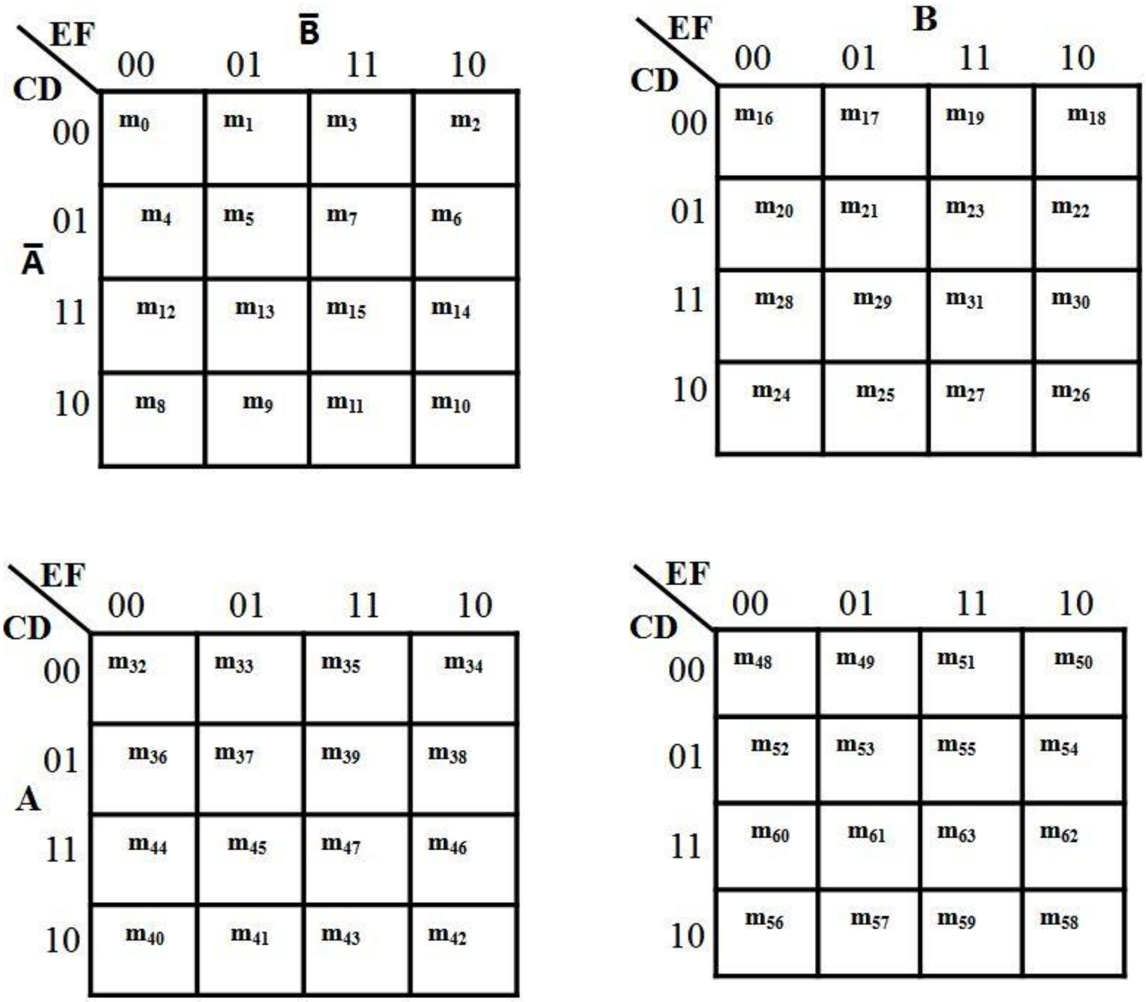
The simplified expression will be the sum of these 4 terms, which is given below:

$$\mathbf{F = \overline{A}\,\overline{C}\,\overline{E} + \overline{C}\,\overline{D}\,\overline{E} + \overline{B}CE + ABDE}$$

5.7 6-Variable K-Map

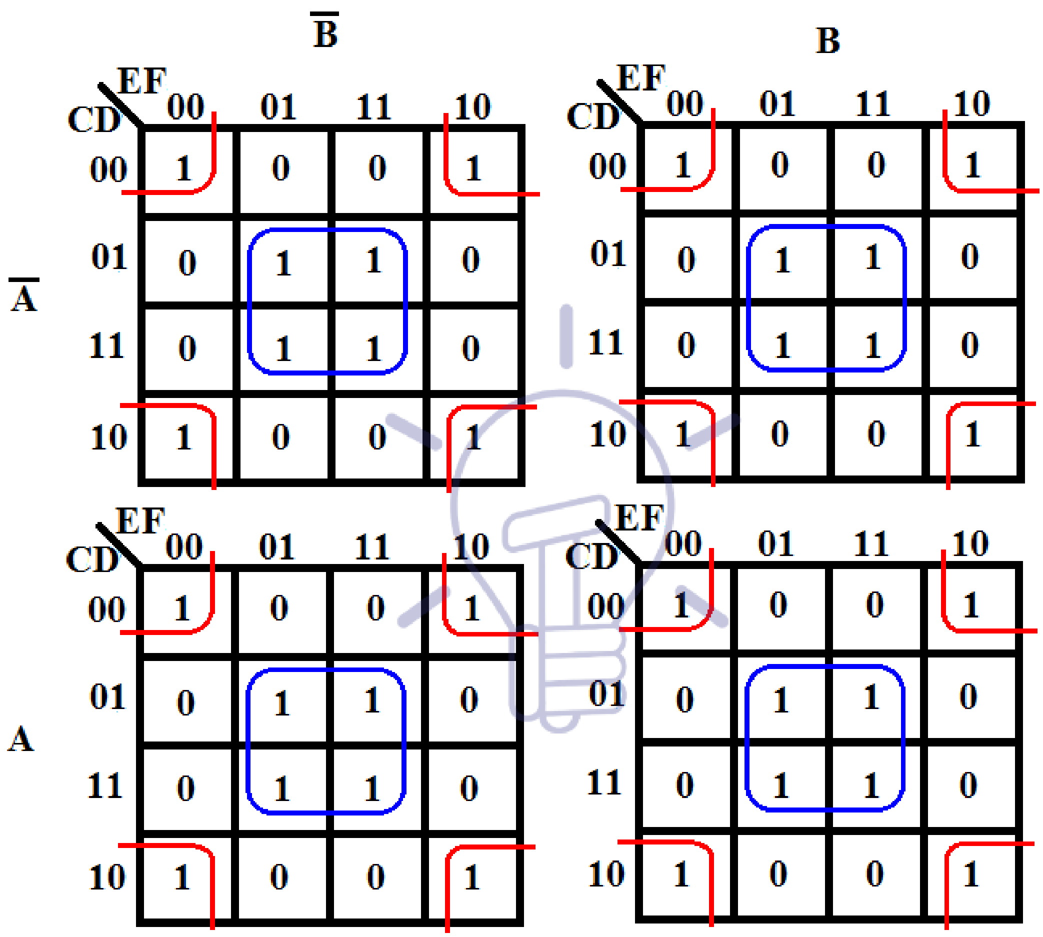
6-variable k-map is a complex k-map which can be drawn. Visualizing 6-variable k-map is a little bit tricky.

6 variables make 64 min terms, this means that the k-map of 6 variables will have 64 cells. Its geometry becomes difficult to draw as these cells are adjacent to each other in all direction in 3-dimensions i.e. a cell is adjacent to upper, lower, left, right, front and back cells at the same time. We can draw it like 5-variable k-map as shown in the figure below.



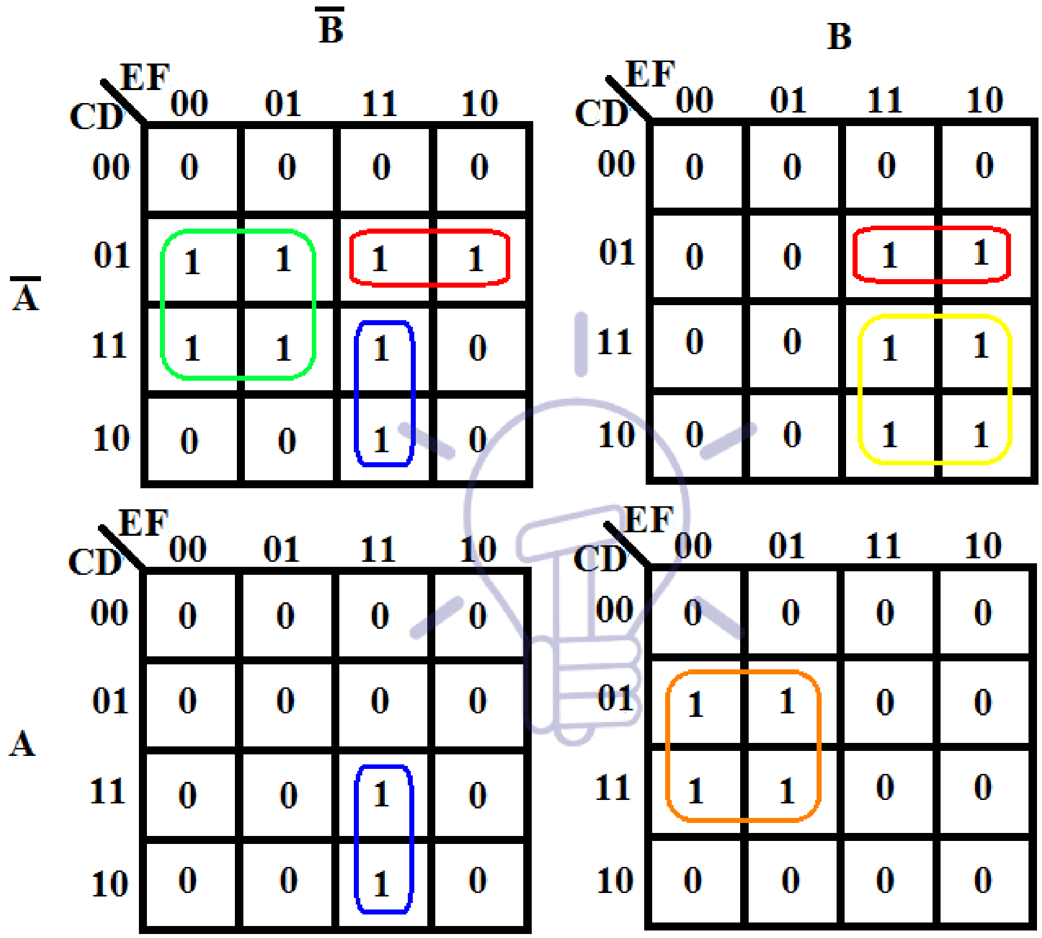
The 6-variable k-map is made from 4-variable 4 k-maps. As you can see variable A on the left side select 2 k-maps row-wise between these 4 k-maps. A = 0 for the upper two K-maps and A = 1 for the lower two K-maps. Variable B on top of these K-maps select 2 k-maps column-wise. B = 0 for left 2 K-maps and B = 1 for right 2 K-maps.

Some examples of grouping in 6-variable K-map are given below.



Groups of 16

Group of 16 min-terms between 4 k-maps as they are all adjacent. Visualize these k-maps on top of each other.



In the above example, there are 5 groups of 4 min-terms. Notice the min-terms in the diagonal K-maps, they make a separate group because these K-maps are not adjacent.

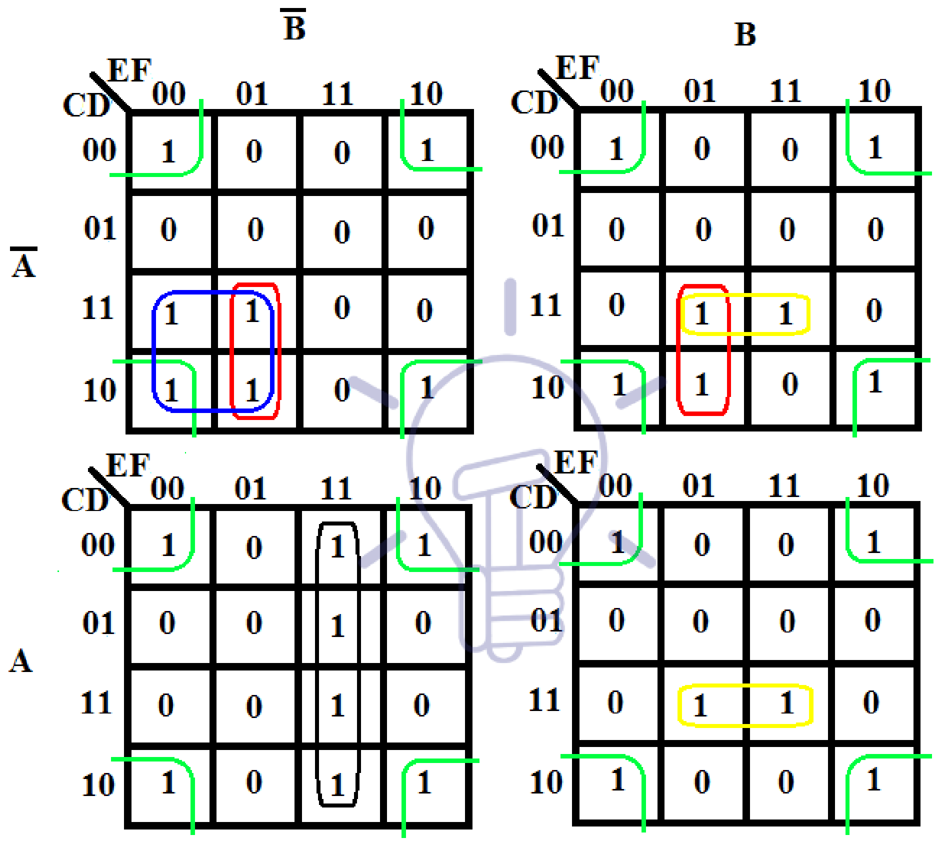
Example of 6 Variable K-Map maping:

Simplify the given function using K-map.

$$F = \sum (m_0, m_2, m_8, m_9, m_{10}, m_{12}, m_{13}, m_{16}, m_{18}, m_{24}, m_{25}, m_{26}, m_{29}, m_{31}, m_{32}, m_{34}, m_{35}, m_{39}, m_{40}, m_{42}, m_{43}, m_{47}, m_{48}, m_{50}, m_{56}, m_{58}, m_{61}, m_{63})$$

Solution:

Its 6 variable K-map is given below-



There are 5 groups (discussed below) in this K-map each colored different.

- Green group is made of 16 min-terms between all 4 individual K-maps. In this group, AB keeps changing so they will be eliminated from the term. C & E are also changing so they will be eliminated from the term too. So the term will become $\overline{D}\overline{F}$ because they remain unchanged throughout the group.
- Red group is made of 4 min-terms. In this group, B is changing so it will be eliminated. D is also changing. So the only remaining unchanged literals will make the term which is $\overline{A}C\overline{E}F$.
- Blue group is also made of 4 min-terms. The only changing variables are DF throughout this group so they will be eliminated from the term. The non-changed literal in this group are $\overline{A}\overline{B}C\overline{E}$ which will be the term produced by this group.
- The Yellow group is also a group of 4 min-terms and the changing variables in this group are AE. The literal that remains unchanged are BCDF in this group.

- The black group is of 4 min-terms too. This group produces the term $A\overline{B}EF$ because they are the unchanged literals in this group.

Hence, the simplified expression of the function will be the sum of these 5 terms from these groups. The expression is given below:

$$\mathbf{F = \overline{D}\overline{F} + \overline{A}C\overline{E}F + \overline{A}\overline{B}C\overline{E} + BCDF + A\overline{B}EF}$$

UNIT-4

Short Type Questions (AKTU)

1. What is a Logic gate?
2. What is meant by a bit, pair, and quad?
3. What are the various logic gates, give the representation along with the truth table.
4. What is DeMorgan's theorem?
5. Define minterms and maxterms.
6. What are universal gates? Why are they called so?
7. Draw the circuit of a 2 input EX-OR gate using 2 input NAND gates.
8. What is the use of don't care combinations? Is it an advantage or disadvantage to include them in a map?
9. Distinguish between SOP and POS.
10. What is the difference between canonical form and standard form.

Long Type Questions (AKTU)

1. Convert the following to the corresponding bases
   i. $(9BCD)_{16} = (\ )_8$
   ii. $(323)_4 = (\ )_5$
   iii. $(53.625)_{10} = (\ )_2$
   iv. $(3FD)_{16} = (\ )_2$
   v. $(A69.8)_{16} = (\ )_{10}$
   vi. $(246.8)_{10} = (\ )_2$, $(\ )_8$ & $(\ )_{16}$
2. Perform Binary subtraction of the following:
   i. $(1011001)_2 - (1101010)_2$ using 1's complement
   ii. $(1101)_2 - (1001)_2$ using 2's complement
3. Simplify the following logic expression using Boolean Algebra:
   i. F= AB+ A(B+C) +B(B+C)
   ii. F= AB'C'D +A'B'D+BCD'+A'B+BC'
4. Consider the following Boolean expression:
   i. F (A, B, C) = A+ AB+ABC, convert it into canonical SOP form
   ii. F (A, B, C) = (A+B) (B+C) (C+A), convert it into canonical POS form
5. Using K map method, determine the minimal SOP expression for the following using decimal notation F=Σ m (1,4,7,9,12,14) +Σ d (2,13). Also, implement using logic gates.
6. Simplify the expression F(a,b,c,d,e)=∑m(1,2,4,7,12,14,15,24,27,29,30,31) using K-map. Also, implement using logic gates.
7. Simplify the following function using K map:
   F (A, B, C, D) = ∑m (1, 3, 4, 5, 6, 7, 9, 11, 13, 15). Also, implement the simplified function using basic gates only.
8. Minimize using K-map and realize output using gates.
   F (A, B, C, D) = Σ m (1, 4, 8, 12, 13,15) + d (3, 14)
9. Reduce the following function using K-Map.

$F(A, B, C, D, E) = \Sigma m(1,4,8,10,11,20,22,24,25,26) + d(0,12,16,17)$

10. Simplify following logic function using K Map and realize using NOR gates.
    i. $f(w,x,y,z) = \pi M(1,2,3,7,10,11)$
    ii. $f(w,x,y,z) = \pi M(3,4,5,6,7,10,11,15)$
11. Simplify the following function with the help of K map:
    $F(A, B, C, D) = \Sigma m(3,5,9,11,15) + d(2,4,6,10)$
12. Minimize the following using K-map:
    $F(A, B, C, D) = AB'C' + A'BC + A'B'CD + ABCD + d(1,5)$
13. Implement EX-OR and EX-NOR gates using minimum NAND gates
14. Implement EX-OR and EX-NOR gates using minimum NOR gates.
15. Define Universal Gates. Implement AND, OR, NOR by using NAND gates only.

Previous Year asked AKTU Questions and their solutions

1. Convert the following to the corresponding bases

- vii. $(73.12)_{10} = (\ )_2$
- viii. $(110110.011)_2 = (\ )_{16}$
- ix. $(231.36)_{10} = (\ )_2$
- x. $(11011.10)_2 = (\ )_{10}$
- xi. $(6FB.67)_{16} = (\ )_{10}$
- xii. $(437)_8 = (\ )_{10}$
- xiii. $(1110111.11011)_2 = (\ )_8$
- xiv. $(1011.10110)_2 = (\ )_{16}$
- xv. $(743.15)_8 = (\ )_2$
- xvi. $(9AC.1B)_{16} = (\ )_2$

Solution:

i.

$(73.12)_{10} = (?)_2$

		Remainder
2	73	
2	36	1 ↑LSB
2	18	0
2	9	0
2	4	1
2	2	0
	1 (MSB)	0

$(73)_{10} = (1001001)_2$

- For fractional part : multiply the fractional part by new base and record the carry

	Carry
$.13 \times 2 =$ (MSB)	0 . 26
$.26 \times 2 =$	0 . 52
$.52 \times 2 =$	1 . 04
$.04 \times 2 =$	0 . 08 (LSB)

$(.13)_{10} = (.0010)_2$

So final answer

$(73.12)_{10} = (1001001.0010)_2$

ii.

$\underbrace{0011}_{8421}\ \underbrace{0110}_{8421} . \underbrace{0110}_{8421}$

$= (36.6)_{16}$

iii.

$(231.36)_{10}$

		Remainder
2	231	
2	115	1
2	57	1
2	28	1
2	14	0
2	7	0
2	3	1
	1	1

$$.36 \times 2 = 0.72$$
$$.72 \times 2 = 1.44$$
$$.44 \times 2 = 0.88$$
$$.88 \times 2 = 1.76$$

$(.36)_{10} = (.0101\ldots)_2$

$(231)_{10} = (11100111)_2$

final answer $(231.36)_{10} = (11100111.0101\ldots)_2$

iv.

$(11011.10)_2$

$= 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2}$

$= (27.5)_{10}$

$(534)_8$

$= 5\times8^2 + 3\times8^1 + 4\times8^0$

$= (348)_{10}$

v.

$$\begin{array}{cccccc} 16^2 & 16^1 & 16^0 & & 16^{-1} & 16^{-2} \\ \uparrow & \uparrow & \uparrow & & \uparrow & \uparrow \\ 6 & F & B & . & 6 & 7 \\ & (15) & (11) & & & \end{array}$$

$= 6\times16^2 + 15\times16^1 + 11\times16^0 + 6\times16^{-1} + 7\times16^{-2}$

$= 1536 + 240 + .11 + .375 + .0273$

$= (1787.4023)_{10}$

vi.

$(437)_8 = (?)_{10}$

$$\begin{matrix} 8^2 & 8^1 & 8^0 \\ \uparrow & \uparrow & \uparrow \\ 4 & 3 & 7 \end{matrix}$$

$= 4 \times 8^2 + 3 \times 8^1 + 7 \times 8^0$

$= 256 + 24 + 7$

$= (287)_{10}$

vii.

$(1\,1\,1\,0\,1\,1\,1 . 1\,1\,0\,1\,1)_2 = (?)_8$

Extra zero to make group of 3

$\underbrace{001}_{421}\ \underbrace{110}_{421}\ \underbrace{111}_{421} . \underbrace{110}_{421}\ \underbrace{110}_{421}$ ← Extra zero to make group of 3

$= (167.66)_8$

viii.

$$(1011.10110)_2 = (?)_{16}$$

$$\underbrace{1011}_{8421} . \underbrace{1011}_{8421}\ \underbrace{0000}_{8421}$$

$$= (B.B0)_{16}$$

ix.

$$(743.15)_8 = (?)_2$$

743.15

7	4	3	1	5
421	421	421	421	421
111	100	011	001	101

$$= (111100011.001101)_2$$

x.

$$(9AC.1B)_{16} = (?)_2$$

$(9\ A\ C\ .\ 1\ B)_{16}$

9	A (10)	C (12)	1	B (11)
8421	8421	8421	8421	8421
1001	1010	1100	0001	1011

$$= (100110101100.00011011)_2$$

2. Simplify the following logic expression using Boolean Algebra:

iii. $F = (A+C)(AD+A\overline{D})+AC+C$

iv. $F = ((A.B)' + C)'. B$

Solution:

i.

$(A+C)\ A(D+\bar{D}) + C[A+1]$

$= (A+C) \cdot A\ (1) + C(\cdot 1)$

$= AA + AC + C$

$= A + C[A+1]$

$= A + C$

ii.

$\{(AB)' + C\}' \cdot B$

$= \{(AB)'' \cdot C'\} \cdot B$

$= (ABC') \cdot B$

$= AB \cdot BC'$

$= ABC'$

3. Consider the following Boolean expression:

iii. $F(A, B, C) = A+B\overline{C}+\overline{A}BC$, convert it into standard SOP form

iv. $F(A, B, C) = A.(\overline{B}+\overline{C}). (A+\overline{B}+\overline{C})$, convert it into standard POS form

v. $F(A, B, C) = (A+\overline{B}). (B+C)$, convert it into standard POS form

Solution

i.

$$F(A,B,C) = A + B\overline{C} + \overline{A}BC$$
$$= A(B+\overline{B})(C+\overline{C}) + B\overline{C}(A+\overline{A}) + \overline{A}BC$$
$$= ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C} + AB\overline{C} + \overline{A}B\overline{C} + \overline{A}BC$$
$$= ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC$$

ii.

$$f(A,B,C) = A\cdot(\overline{B}+\overline{C})\cdot(A+\overline{B}+\overline{C})$$
$$= [A + B\overline{B} + C\overline{C}]\cdot(\overline{B}+\overline{C}+A\overline{A})(\overline{A}+\overline{B}+\overline{C})$$
$$= (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(A+\overline{B}+\overline{C})(A+B+\overline{C})\cdot(\overline{A}+B+\overline{C})(\overline{A}+B+\overline{C})$$
$$= (A+B+C)(A+B+\overline{C})(A+\overline{B}+C)(A+\overline{B}+\overline{C})(\overline{A}+B+\overline{C})$$

iii.

$$F = (A+\overline{B})(B+C)$$
$$\text{Now } (A+\overline{B}) = A+\overline{B}+C\overline{C} = (A+\overline{B}+C)(A+\overline{B}+\overline{C})$$
$$(B+C) = (A\overline{A}+B+C)$$
$$= (A+B+C)(\overline{A}+B+C)$$
$$\text{So } F = (A+\overline{B}+C)(A+\overline{B}+\overline{C})(A+B+C)(\overline{A}+B+C)$$
$$= M_2 * M_3 * M_0 * M_4$$
$$= \prod M(0,2,3,4)$$

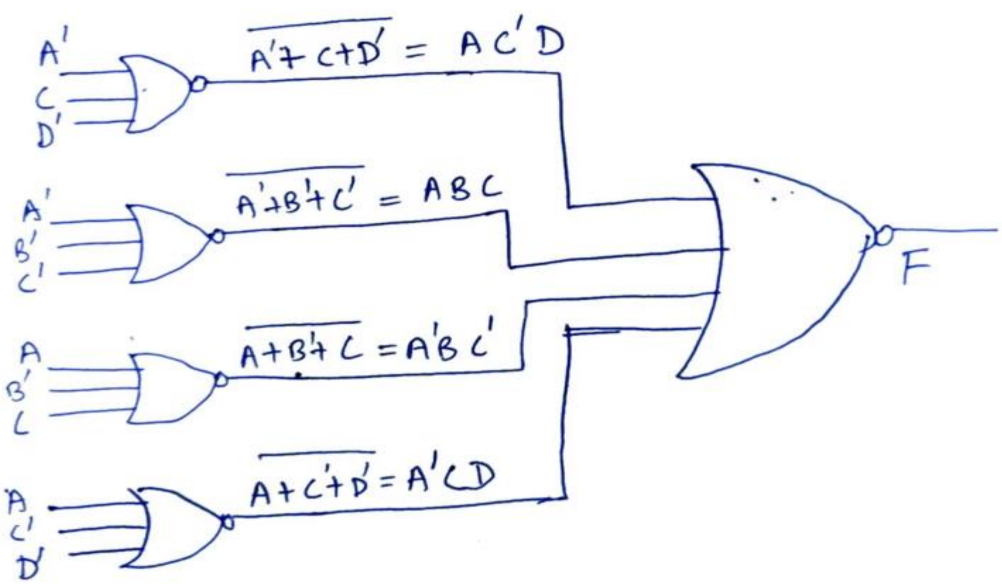
4. Using K-map method, determine the minimal POS expression for the following using decimal notation F (A, B, C, D) = πM (3,4,5,7,9,13,14,15). d (0, 2, 8). Also, implement using NOR gates only.

<u>Solution</u>

$$F(A,B,C,D) = \Pi M(3,4,5,7,9,13,14,15).d(0,2,8)$$

AB \ CD	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	X^{0}	^{1}	0^{3}	X^{2}
$A+\bar{B}$	0^{4}	0^{5}	0^{7}	^{6}
$\bar{A}+\bar{B}$	^{12}	0^{13}	0^{15}	0^{14}
$\bar{A}+B$	X^{8}	0^{9}	^{11}	^{10}

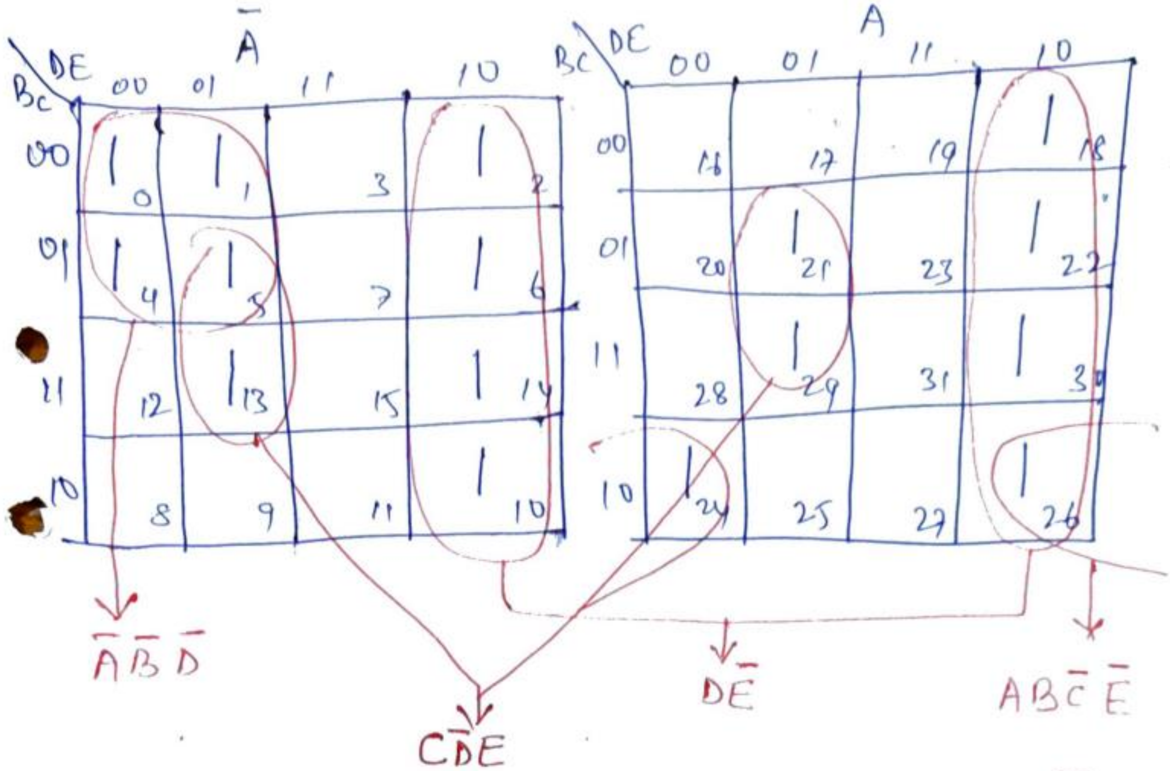
$$f = (A+\bar{C}+\bar{D})(A+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})(\bar{A}+C+\bar{D})$$



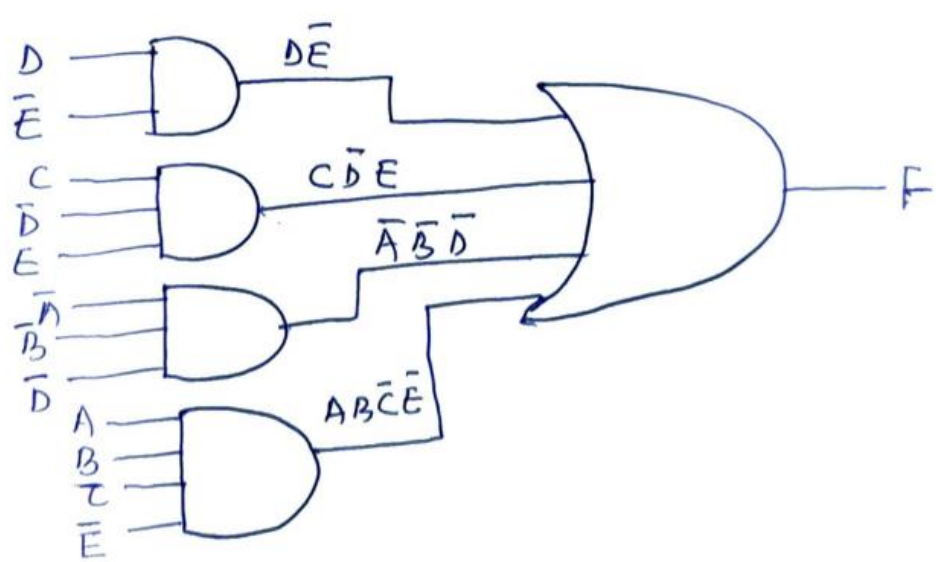
5. Simplify the expression $F(a,b,c,d,e)=\sum m(0,1,2,4,5,6,10,13,14,18,21,22,24,26,29,30)$ using K-map. Also, implement using logic gates.

Solution

$$F(A,B,C,D,E) = \sum m(0,1,2,4,5,6,10,13,14,18,21,22,24,26,29,30)$$

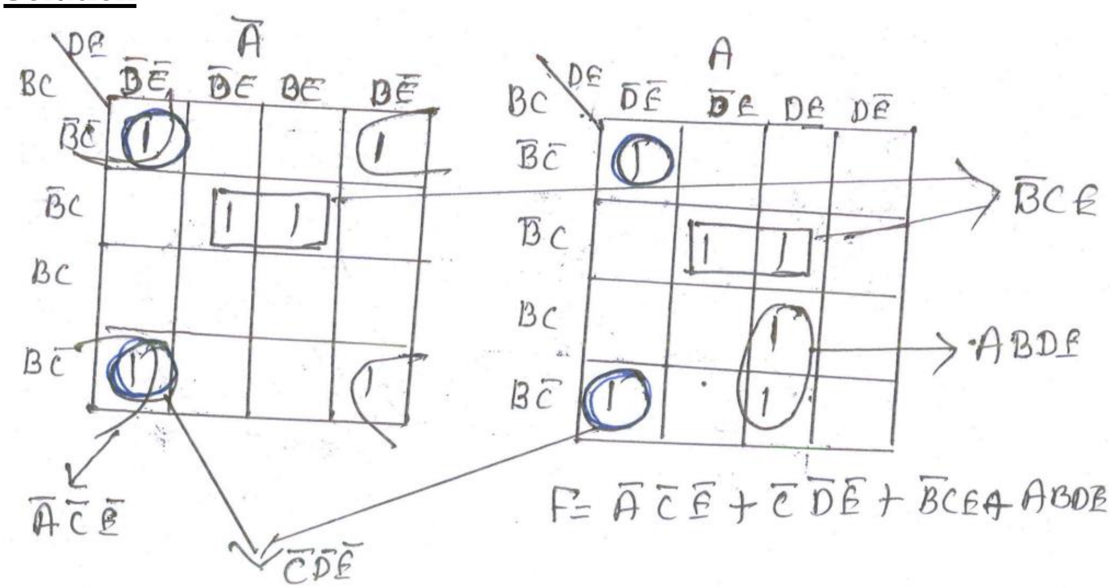


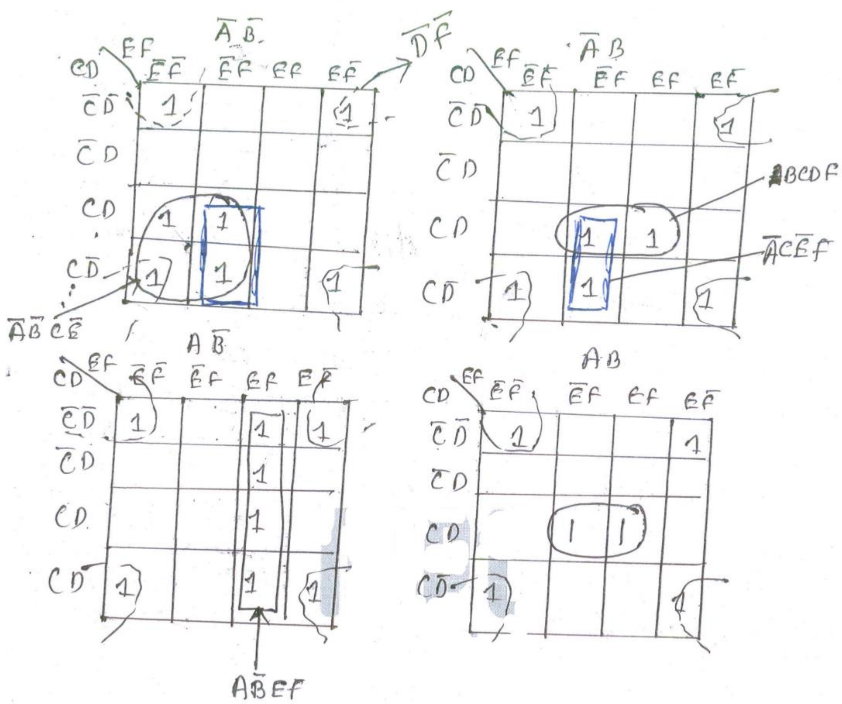
$$F = D\bar{E} + C\bar{D}E + \bar{A}\bar{B}\bar{D} + AB\bar{C}\bar{E}$$



6. Simplify the following function using K map:
F (A, B, C, D, E) = ∑m (0,2,5,7,8,10,16,21,23,24,27,31). Simplify using K-map.

Solution





$F = \overline{A}\,\overline{B}C\overline{E} + \overline{D}\,\overline{F} + BCDF + \overline{A}C\overline{E}F + A\overline{B}EF$

7. Simplify the following using K-map.
F (A, B, C, D) = Σ m (3,5,9,11,15) + d (2,4,6,10)

$F(A,B,C,D) = \Sigma m(3,5,9,11,15) + d(2,4,6,10)$

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

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

$\rightarrow ACD$

$\rightarrow \bar{B}C$

$A\bar{B}D$

So $F = \bar{A}B\bar{C} + A\bar{B}D + \bar{B}C + ACD$

8. Reduce the following function using K-Map. Also, implement with basic gates only.
F (A, B, C, D, E) = Σm (1,3,4,5,6,7,9,11,13,15)

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

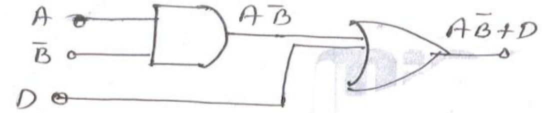
$\rightarrow \bar{A}B$

$\rightarrow D$

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

Basic gates realisation:-

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



9. Simplify following logic expression using K-Map and realize using NOR gates only.
F (A, B, C, D, E) = $\overline{A}\,\overline{B}\overline{C} + A\,\overline{C}\overline{D} + A\,\overline{B} + ABC\overline{D} + \overline{A}\overline{B}\,C$

Solution

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

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

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

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

$\rightarrow A\bar{D}$

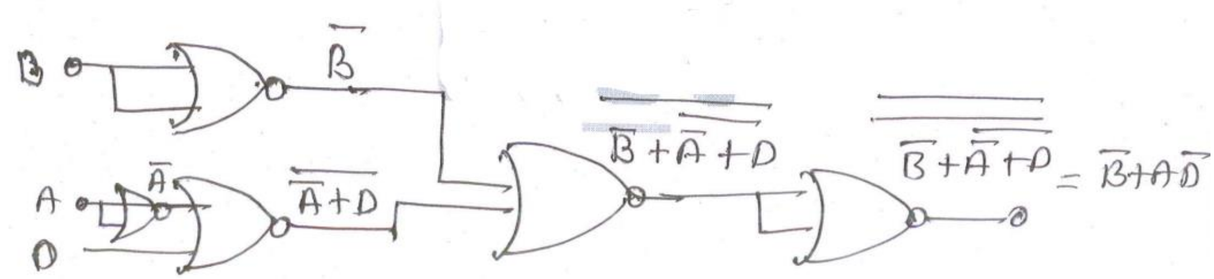
$\rightarrow \bar{B}$

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

NOR realization :

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

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

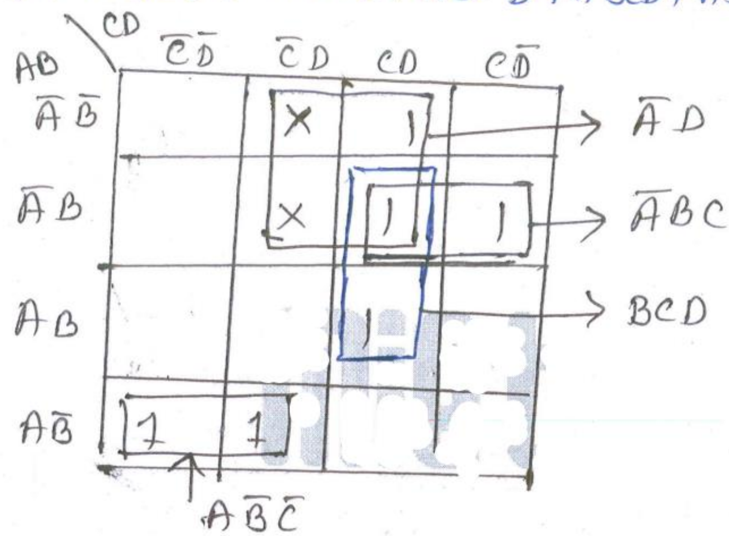


10. Minimize the following using K-map:
F (A, B, C, D) = AB’C’ +A’BC + A’B’CD +ABCD + d (1,5)

$$F(A,B,C,D) = A\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}CD + ABCD + d(1,5)$$

$$F(A,B,C,D) = A\bar{B}\bar{C}(D+\bar{D}) + \bar{A}BC(D+\bar{D}) + \bar{A}\bar{B}CD + ABCD$$

$$= A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}\bar{B}CD + ABCD + d(1,5)$$



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

11. Implement EX-OR and EX-NOR gates using minimum NAND gates.

o/p of Ex-or gate $= \bar{A}B + A\bar{B}$

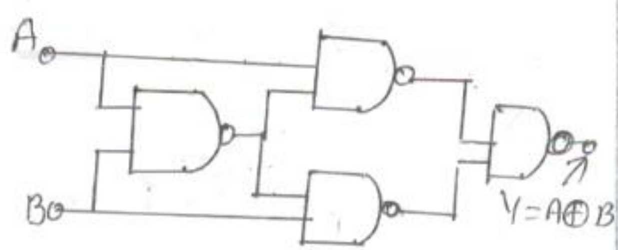
$$= A\bar{A} + A\bar{B} + \bar{A}B + B\bar{B}$$

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

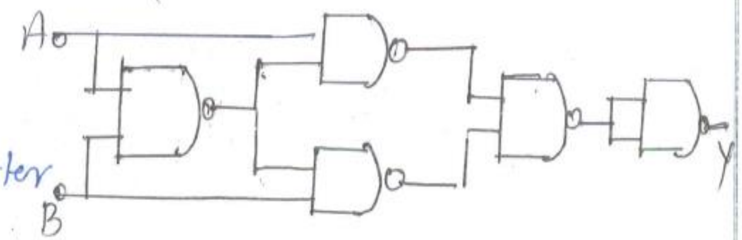
$$= A\cdot(\overline{A\cdot B}) + B(\overline{A\cdot B})$$

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

$$= \overline{\overline{A\cdot(\overline{A\cdot B})}\cdot\overline{B(\overline{A\cdot B})}}$$



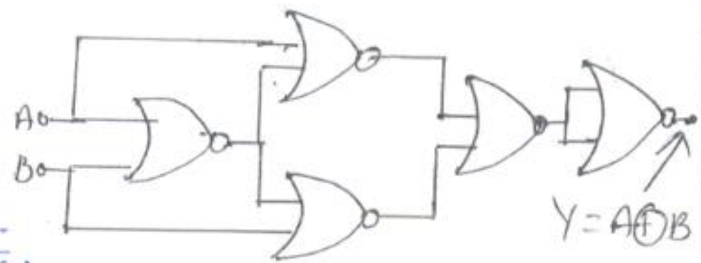
• NAND as Ex-NOR gate.
o/p of Ex-NOR gate
$= AB + \bar{A}\bar{B}$
It is opposite of Ex-OR gate. So by adding one inverter gate we get the Ex-NOR gate.



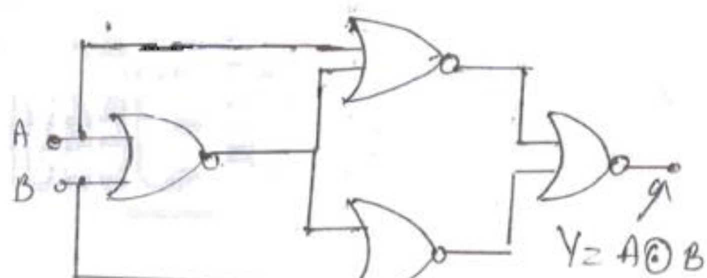
12. Implement EX-OR and EX-NOR gates using minimum NOR gates.

o/p of Ex-OR gate

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



• NOR as Ex-NOR gate:
o/p of Ex-NOR gate
$Y = AB + \bar{A}\bar{B}$
It is opposite of Ex-OR gate. So, by adding one inverter we get the Ex-NOR gate.

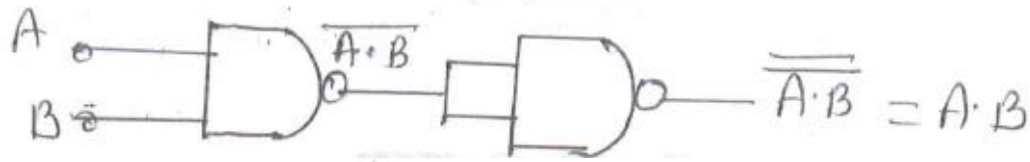


13. Define Universal Gates. Implement AND, OR, NOR by using NAND gates only.

NAND as AND gate :

O/p of AND gate $Y = A \cdot B$

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

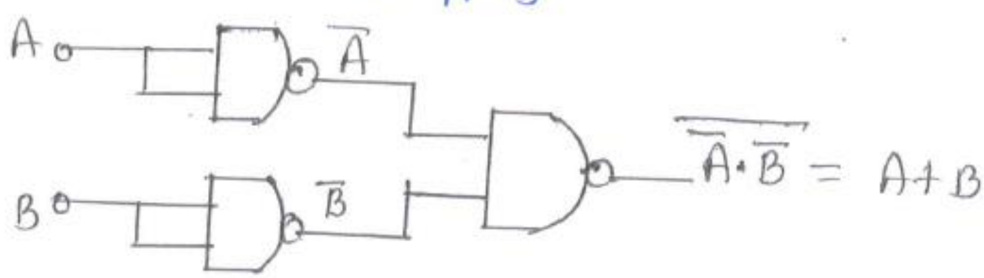


NAND as OR Gate :

O/p of OR gate $= A + B$

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

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

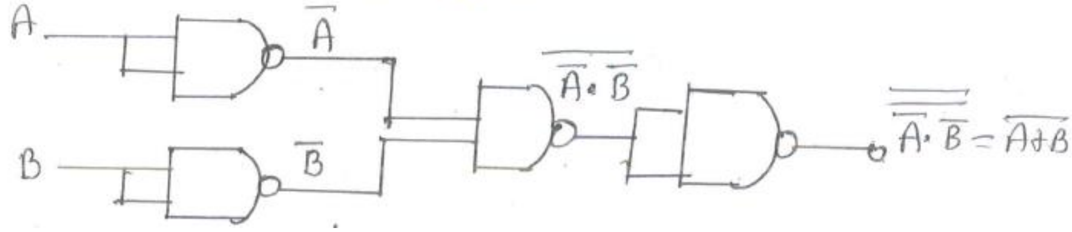


NAND as NOR gate.

O/p of NOR gate $= \overline{A + B}$

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

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



xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx