Unit - 1
Number System

Number System :-

In digital electronic Number System is used for the representing the number System

* The base or radix of the number System is the total number of the digits used in The number System.
* Suppose if the number System representing the digit from 0-9 then the base of The System is 10.

Number System	Base	1^{st} digit	last digit	All digits
Binary	2	0	1	0,1
octal	8	0	7	0,1,2,3,4,5,6,7
Decimal	10	0	9	0 . . . 9
Hexa decimal	16	0	F	A, B, C, D, E, 0 - - - - 9

In Hexa Decimal A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Types of Number System

Number Systems are of 4 types

Decimal number System
binary number System
octal
Hexa decimal.

Decimal number System :-

The position of each digit in a weighted number System is assigned a weight based on the base (or) radics of the System

* The base of the decimal number System is 10. It Contains numbers like 0, 1, 2 - - - - 9

Eg : 1245.42

1	2	4	5	.	4	2
10^3	10^2	10^1	10^0	↑ Decimal point	10^{-1}	10^{-2}

Eg:- Represent binary number 98.72, in power

* $9\times10^1 + 8\times10^0 + 7\times10^{-1} + 2\times10^{-2}$

5432.423

* $5\times10^3 + 4\times10^2 + 3\times10^1 + 2\times10^0 + 4\times10^{-1} + 2\times10^{-2} + 3\times10^{-3}$

Binary number System:-

The position of each digit in a weighted number System is assigned a weight based on the base (or) radics of the System.

* The base of the decimal number System is 2. It contains numbers likes 0 and 1.

Eg:- 1101.10

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

Eg:- Represent Binary number 1100.01 is power of 2 and find its decimal equivalent

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

Octal number System:

The position of each digit in a weighted number System is assigned a weight based on the base (or) radics of The System.

* The base of the decimal number System is 8. It Contains numbers like 0, 1, 2, - - - 7

Eg:- $(234)_8$

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

$= (156)_{10}$

* Represent octal number 234 in power of 8 and find its decimal equivalent

Hexa decimal number System.

The position of each digit in a weighted number System is assigned a weight based

on the base (or) radics of the system
* The base of the decimal number System is 16
It Contains numbers like A, B, C, D, E, F and 0 - - 9

Eg:- Represent Hexa decimal number 3FD in power of 16 and find its decimal equivalent 3FD

$3 \times 16^2 + F \times 16^1 + D \times 16^0$
$\quad (15) \qquad (13)$

$(1021)_{10}$

The below table shows the relation between decimal, binary, hexa decimal.

Decimal	Binary	Hexa decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1) $76 = (1001100)_2$

2 | 76
2 | 38 - 0
2 | 19 - 0
2 | 9 - 1
2 | 4 - 1
2 | 2 - 0
1 - 0

2) $84 = (1010100)_2$

2 | 84
2 | 42 - 0
2 | 21 - 0
2 | 10 - 1
2 | 5 - 0
2 | 2 - 1
1 - 0

3) $18 = (10010)_2$

2 | 18
2 | 9 - 0
2 | 4 - 1
2 | 2 - 0
1 - 0

Counting in Radix (or) Base (r) :-

* Each number System has 'r' Set of characters
* for example, In decimal number System $r = 10$ has 10 characters from 0 to 9.
* In Binary number System $r = 2$ has 2 characters 0 and 1
* A number represented in radix or has 'r' characters in its Set. and 'r' can be any value If it shown in below table.

Radix (Base) r	character in set
2 (Binary)	0, 1
3	0, 1, 2
4	0, 1, 2, 3
⋮	
7	0, 1, 2, 3, 4, 5, 6
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
⋮	
10 (Decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
⋮	
16 (Hexa decimal)	0, 1 - - - 9, A, B, C, D, E, F

Ex :- Find The decimal equivalent of $(231.23)_4$

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

$= 32 + \cancel{12 + 3 + \frac{1}{2} + 6}$

$= (48.6875)_{10}$

Number Base Conversions :-

It is necessary to convert decimal number into its equivalent binary by displaying the result of operation to The human beings. has showing the below table binary, decimal, octal and hexadecimal.

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversion of Decimal number to any radix number:-

* Decimal to any radix number has two conversion steps

Step-1

We have to convert integer part

Step-2

We have to convert fraction part

The conversion of integer part is successive division method. The conversion of fractional part is successive multiplication method.

Decimal to Binary conversion:-

1) Convert decimal number 37 to its Binary equivalent

$(37)_{10} = (100101)_2$

2 | 37
2 | 18 − 1 least
2 | 9 − 0 significant
2 | 4 − 1 bit
2 | 2 − 0
1 − 0
most significant bit

2) Convert Decimal number 61 to its binary equivalent.

$$(61)_{10} = (111101)_2$$

$$\begin{array}{r|l} 2 & 61 \\ \hline 2 & 30 - 1 \\ \hline 2 & 15 - 0 \\ \hline 2 & 7 - 1 \\ \hline 2 & 3 - 1 \\ \hline & 1 - 1 \end{array}$$

3) Convert $(0.8125)_{10}$ decimal number to its binary number.

$(0.8125)_{10}$

$0.8125 \times 2 = 1.625 = 0.625$ with carry 1 MSB

$0.625 \times 2 = 1.25 = 0.25$ with carry 1

$0.25 \times 2 = 0.5 = 0.5$ with carry 0

$0.5 \times 2 = 1.0 = 0.0$ with carry 1 LSB

$(0.8125)_{10} = (0.1101)_2$

4) Convert 0.95 decimal number to its binary number.

$0.95 \times 2 = 1.9 = 0.9$ with carry 1

$0.9 \times 2 = 1.8 = 0.8$ with carry 1

$0.8 \times 2 = 1.6 = 0.6$ with carry 1

$0.6 \times 2 = 1.2 = 0.2$ with carry 1

$0.2 \times 2 = 0.4 = 0.4$ with carry 0

$0.4 \times 2 = 0.8 = 0.8$ with carry 0

$0.8 \times 2 = 1.6 = 0.6$ with carry 1

$(0.95)_{10} = (0.1111001)_2$

5) Convert decimal number 24.6 to its binary number

$$(24)_{10} = (11000)_2$$

$$\begin{array}{r|l} 2 & 24 \\ \hline 2 & 12 - 0 \\ \hline 2 & 6 - 0 \\ \hline 2 & 3 - 0 \\ \hline & 1 - 1 \end{array}$$

$0.6 \times 2 = 1.2 = 0.2$ with carry 1
$0.2 \times 2 = 0.4 = 0.4$ with carry 0
$0.4 \times 2 = 0.8 = 0.8$ with carry 0
$0.8 \times 2 = 1.6 = 0.6$ with carry 1
$0.6 \times 2 = 1.2 = 0.2$ with carry 1

$\therefore (24.6)_{10} = (11000.10011)_2$

6) convert decimal number 34.6 to its binary number

$$\begin{array}{r|l} 2 & 34 \\ \hline 2 & 17 - 0 \\ \hline 2 & 8 - 1 \\ \hline 2 & 4 - 0 \\ \hline 2 & 2 - 0 \\ \hline & 1 - 0 \end{array}$$

$0.6 \times 2 = 1.2 = 0.2$ with carry 1

$(34.6)_{10} = (100010.10011)_2$

7) convert decimal number 42.385 into binary number

$$\begin{array}{r|l} 2 & 42 \\ \hline 2 & 21 - 0 \\ \hline 2 & 10 - 1 \\ \hline 2 & 5 - 0 \\ \hline 2 & 2 - 1 \\ \hline & 1 - 0 \end{array}$$

$0.385 \times 2 = 0.77 = 0.77$ with carry 0
$0.77 \times 2 = 1.54 = 0.54$ with carry 1
$0.54 \times 2 = 1.08 = 0.08$ with carry 1
$0.08 \times 2 = 0.16 = 0.16$ with carry 0
$0.16 \times 2 = 0.32 = 0.32$ with carry 0
$0.32 \times 2 = 0.64 = 0.64$ with carry 0
$0.64 \times 2 = 1.28 = 0.28$ with carry 1
$0.28 \times 2 = 0.56 = 0.56$ with carry 0
$0.56 \times 2 = 1.12 = 0.12$ with carry 1

$(42.385)_{10} = (101010.011000)_2$

1) Convert decimal number 214 to its octal equivalent

$$\begin{array}{r|l} 8 & 214 \\ \hline 8 & 26 - 6 \\ \hline & 3 - 2 \end{array}$$

$(214)_{10} = (326)_8$

2) convert 0.640625 decimal number into its octal number.

$0.640625 \times 8 = 5.125 = 0.125$ with carry 5
$0.125 \times 8 = 1.0 = 0$ with carry 1

$(0.51)_8 = (0.640625)_{10}$

3) convert decimal number 34.45 to its octal number.

$$8 \underline{|34}$$
$$4 - 2$$

$0.45 \times 8 = 3.6 = 0.6$ with carry 3
$0.6 \times 8 = 4.8 = 0.8$ with carry 4
$0.8 \times 8 = 6.4 = 0.4$ with carry 6
$0.4 \times 8 = 3.2 = 0.2$ with carry 3
$0.2 \times 8 = 1.6 = 0.6$ with carry 1

$(34.45)_{10} = (42.3463)_8$

convert decimal to Hexa decimal

1) convert decimal number 3509 to its Hexa decimal

$$16 \underline{|3509}$$
$$16 \underline{|219} - 5$$
$$13 - 11$$

$(3509)_{10} = (DB5)_{16}$

2) convert 0.1289062 decimal number to its hexa decimal

$0.1289062 \times 16 = 2.0625 = 0.0625$ with carry 2
$0.0625 \times 16 = 1.0 = 0.0$ with carry 1

$(0.1289062)_{10} = (0.21)_{16}$

3) convert decimal number 22.64 to its hexadecima number

$$16 \underline{|22}$$
$$1 - 6$$

$0.64 \times 16 = 10.24 = 0.24$ with carry 10
$0.24 \times 16 = 3.84 = 0.84$ with carry 3
$0.84 \times 16 = 13.44 = 0.44$ with carry 13
$0.44 \times 16 = 7.04 = 0.04$ with carry 7
$0.04 \times 16 = 0.64 = 0.64$ with carry 0
$0.64 \times 16 = 10.24 = 0.24$ with carry 10

$(22.64)_{10} = (16.A3D70A)_{16}$

Converting Any radix to decimal :-

In General numbers can be represented as

$$N = A_{n-1} r^{n-1} + A_{n-2} r^{n-2} + \cdots A_1 r^1 + A_0 r^0 + \ldots A_1 r^{-1} + \ldots C_m r^{-m}$$

where N = number in decimal

A = digit

r = radix of base of a number System

n = The No. of digits in the integer position

m = The No. of digits in the fraction part

convert binary to decimal

1) convert binary number $(11{,}01.1)_2$ to its decimal number.

$(1101.1)_2$

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

$= 8 + 4 + 1 + 0.5$

$= 13.5$

2) convert octal to decimal

1) convert $(475.25)_8$ to its decimal number.

$(475.25)_8$

$= 4\times8^2 + 7\times8^1 + 5\times8^0 + 2\times8^{-1} + 5\times8^{-2}$

$= 4\times64 + 56 + 5 + \frac{2}{8} + \frac{5}{64}$

$= 256 + 56 + 5 + 0.25 + \cancel{0.08}\ \cancel{0.03125}\ 0.078125$

$= (317.328)_{10}$

2) convert hexa decimal to decimal

convert $(9B2.1A)_{16}$ to its decimal number.

$(9B2.1A)_{16}$

$9\times16^2 + 11\times16^1 + 2\times1^0 + 1\times16^{-1} + 10\times16^{-2}$

$= 2304 + 176 + 2 + 0.0625 + 0.0391$

$= (2482.1016)_{10}$

3) convert $(3102.12)_4$ to its decimal number

$(3102.12)_4 = 3\times4^3 + 1\times4^2 + 0\times4^1 + 2\times4^0 + 1\times4^{-1} + 2\times4^{-2}$

$= 192 + 16 + 2 + 0.25 + 0.125$

$= (210.375)_{10}$

4) Convert $(614.15)_7$ to its decimal number

$(614.15)_7 = 6\times7^2 + 1\times7^1 + 4\times7^0 + 1\times7^{-1} + 5\times7^{-2}$

$= 294 + 7 + 4 + 0.1429 + 0.10204$

$= (305.2450)_{10}$

Binary to Octal Conversion :-

We know that base for octal number is 8 and the base for binary number is 2

The base for octal number is the third power of the base for binary number.

Therefore by grouping three digits of binary numbers and then converting each group digit to its octal number.

1) convert $(111\ 101\ 100)_2$ to octal equivalent

$(111\ 101\ 100)_2 = (754)_8$

Binary to Hexa decimal :-

We know that base for Hexa decimal number is 16 and the base for binary number is 2.

The base for hexadecimal number is the fourth power of the base for binary number.

Therefore by grouping the four digits of binary number and then converting each group digit to its hexa decimal number.

Convert $(1101100010011011)_2$ to hexa decimal

$(1101\ 1000\ 1001\ 1011)_2 = (D89B)_{16}$

Convert $(010110101001.10110100)_2$ to hexa decimal

$(0101\ 1010\ 1001.1011\ 0100)_2 = (5A9.B4)_{16}$

Octal to Binary conversion :-

Conversion from octal to binary is a reversel process binary to octal. Each digit of the octal number is individually converted to its binary number. to get octal to binary conversion.

Eg:- octal to binary $(432)_8$

$(432)_8 = (100\ 011\ 010)_2$

2) convert $(634)_8$ to binary

$(634)_8 = (110\ 011\ 100)_2$

3) Convert $(725.63)_8$ to binary

$(725.63)_8 = (111\ 010\ 101.110\ 011)_2$

octal to hexadecimal conversion:-

The easiest way to convert octal to hexadecimal number is given below.

convert octal number to its binary number

convert binary number to its hexo decimal number

1) convert $(615)_8$ to its hexa decimal

$(615)_8$

Step-1 octal to binary

6 1 5

110 001 101

Step-2 binary to hexadecimal

6 1 5

000110 001 101

$= (18D)_{16}$

Hexadecimal to binary conversion:-

Convert from hexa decimal to binary is a reversal of the process explained in the binary to hexadecimal Each digit of the hexa decimal number is individually converted to its binary number.

1) convert $(3FD)_{16}$ to its binary

$(3\ F\ D)_{16} = (0011\ 1111\ 1101)_2$

0011 1111 1101

2) convert $(5A9.B4)_{16}$ to its binary

$(5A9.B4)_{16} = (0101\ 1010\ 1001.1011\ 0100)_2$

Hexa decimal to octal conversion:-

* convert hexa decimal to binary
* Convert binary to octal

1) convert $(25B)_{16}$ hexadecimal to its octal number

2 5 B

Step1 0010 0101 1011

Step2 $(1133)_8$

Complements :-

In digital computers to Simplify the Subtraction operation and for logical manupulation complements are used. There are 2 types of complements for each radix System.

1) radix complement 2) diminshed radix complement

The first one is referred as the r's complement and the second one is (r-1)'s complement for example in binary System we Substitute base value is 2 in place of 'r' it refer complements as 2's complement and 1's complement.

In decimal number system we Substitute base value 10 in place of 'r' it refer complements as 10's complement and 9's complement.

1's complement Representation :-

1's complement Representation negative numbers are obtained by complementing each bit of the corresponding positive number.

For example 0100

1's complement is 1011

1) Find 1's complement of i) $(1101)_2$ ii) $(1001001 0011)_2$

i) Given that $(1101)_2$

1's complement = 0010

ii) Given that $(10010010011)_2$

1's complement = 01101101100

2's complement Representation :-

The 2's complement number is obtained by Subtracting corresponding positive number 2^n hence the 2's complement number is obtained by adding 1 to the 1's complement number.

1) find 2's complement of $(1001)_2$

Given that $(1001)_2$

1's complement = 0110
+ 1

2's complement = 0111

2) Find 2's complement of $(1010\ 0011)_2$

Given that $(1010\ 0011)_2$

1's complement = 0101 1100
+ 1

2's complement = 0101 1101

8

9's complement & 10's complement :-

Substraction of decimal numbers can be accomplished by the 9's and 10's complement methods Similar to the 1's and 2's complement method of binary.

9's complement of decimal number is obtained by substracting each bit of the decimal number from 9. 10's complement of a decimal number is obtained by adding '1' to its 9's complement.

Eg :- 9's complement :-

$$20 = \begin{array}{r} 99 \\ -\ 20 \\ \hline 79 \end{array}$$

10's compliment

$$34 \quad \begin{array}{r} 99 \\ -\ 34 \\ \hline 65 \\ +\ 1 \\ \hline 66 \end{array}$$

Find the 9's complement of the following number

i) 3465 ii) 782.54 iii) 4526.075

i) 3465

$$9\text{'s complement} = \begin{array}{r} 9999 \\ -\ 3465 \\ \hline 6534 \end{array}$$

ii) 782.54

9's complement = 999.99
$$\begin{array}{r} 999.99 \\ 782.54 \\ \hline 217.45 \end{array}$$

iii) 4526.075

9's complement = 9999.999
$$\begin{array}{r} 9999.999 \\ 4526.075 \\ \hline 5473.924 \end{array}$$

2) Find the 10's complement of the following decimal numbers

i) 4069

9's complement = 9999
$$\begin{array}{r} 9999 \\ 4069 \\ \hline 5930 \\ 1 \\ \hline 5931 \end{array}$$

10's complement = 5931

ii) 1056.074

9's complement = 9999.999
$$\begin{array}{r} 9999.999 \\ -1056.074 \\ \hline 8943.925 \\ 1 \\ \hline 8943.926 \end{array}$$

10's complement = 8943.926

(r-1)'s complement	r's complement
1's, 9's	2's, 10's
→ Substract b/w two variables a and b i.e a-b	→ Substract b/w two variables a and b i.e a-b
→ a-b = a+(-b)	→ a-b = a+(-b)
→ -b is complement of b	→ -b is complement of b
→ add '-b' to 'a'	→ add '-b' to 'a'
→ If carries exist the result is positive other wise negative	→ If carries exist The result is positive others wise negative
→ If carry = 1 add The carry to Remaining bits	→ If carry = 1, eliminate the carry
→ If carry = 0 find the (r-1)'s complement of Remaining bits	→ If carry = 0, find The r's complement of remaining bits.

1) Substract the following numbers using the 9's complement of the method.

i) 745.81 - 436.62 ii) 436.62 - 745.81

i) 745.81 − 436.62

$a = 745.81$

$b = 436.62$

```
  999.99
  436.62
  ------
  563.37   9's complement of 436.62

  745.81
  563.37
  ------
carry (1)309.18   Intermediate result
          1   Endaround carry
  ------
  309.19
```

ii) 436.62 − 745.81

```
  999.99
 −745.81
  ------
  254.18   9's complement of 742.81

  436.62
 +254.18
  ------
  690.80

  999.99
  690.80
  ------
  309.19
```

2. Substract the following numbers using the 10's complement of the method.

i) 2928.54 − 416.73   ii) 416.73 − 2928.54

```
i)  9999.99              2928.54
(−) 0416.73            + 9583.27
    -------              -------
    9583.26             (1)2511.81
+         1               ↑
    -------            eliminate the carry   then 2511.81
    9583.27
```

ii) 416.73 − 2928.54

```
    9999.99             416.73          9999.99
(−) 2928.54          + 7071.46      (−) 7488.19
    -------            -------          -------
    7071.45            7488.19          2511.80
+         1                                   1
    -------                             -------
    7071.46                             2511.81
```

3) express -45 in eight bit 2's complement form.

45 = 101101 = 101101

$$\begin{array}{r|l} 2 & 45 \\ 2 & 22-1 \\ 2 & 11-0 \\ 2 & 5-1 \\ 2 & 2-1 \\ & 1-0 \end{array}$$

45 in 8 bits = 00101101

1's complement of 45 = 11010010

+1

2's complement -45 = 11010011

3) Substract 14 from 46 using the 8 bit 2's complement arithmatic.

14 = 00001110

1's complement = 11110001

+1

-14 = 11110010

$$\begin{array}{r|l} 2 & 46 \\ 2 & 23-0 \\ 2 & 11-1 \\ 2 & 5-1 \\ 2 & 2-1 \\ & 1-0 \end{array}$$

46
14
(-) 32

0011 0010
1100 1101
1
1100 1110
(CE)

46+(-14) = 00101110
11110010
100001000

eliminate the carry

4) Add -75 to +26 using the 8 bit 2's complement arithmatic.

75 = 01001011

1's complement = 10110100

+1

2's complement = 10110101

26 = 11010 = 00011010

$$\begin{array}{r|l} 2 & 75 \\ 2 & 37-1 \\ 2 & 18-1 \\ 2 & 9-0 \\ 2 & 4-1 \\ 2 & 2-0 \\ & 1-0 \end{array} \qquad \begin{array}{r|l} 2 & 26 \\ 2 & 13-0 \\ 2 & 6-1 \\ 2 & 3-0 \\ & 1-1 \end{array}$$

-75+26 = 00011010
10110101
11001111

5) Add -45.75 to +87.5 using the 12 bit 2's complement arithmatic.

45.75 = 101101.11 = 00101101.1100

1's complement = 11010010.0011

+1

2's complement = 11010010.0100

-45.75 = 11010010.0100

87.5

$$\begin{array}{r|l} 2 & 87 \\ 2 & 43-1 \\ 2 & 21-1 \\ 2 & 10-1 \\ 2 & 5-0 \\ 2 & 2-1 \\ & 1-0 \end{array}$$

0.75 × 2 = 1.5 = 0.5 with carry 1

0.5 × 2 = 1.0 = 0.0 with carry 1

$$\begin{array}{r|l} 2 & 45 \\ 2 & 22-1 \\ 2 & 11-0 \\ 2 & 5-1 \\ 2 & 2-1 \\ & 1-0 \end{array}$$

0.75 × 2 = 1.5 = 0.5 with carry 1

0.5 × 2 = 1.0 = 0.0 with carry 1

87.5 = 1010111.10 = 01010111.1000

-45.75 + 87.5 =

$$\begin{array}{r} 01010111.1000 \\ 11010010.0100 \\ \hline \text{①}00101001.1100 \end{array}$$

eliminate carry

Binary codes:-

The digital data is represented stored and transmitted group of binary digits (bits). The group of bits also known as binary codes. It represent both numbers and letters of the alphabets as well as many Special characters and control functions.

Classification of binary codes

The binary codes can be classified as

i) Weighted code:-

In weighted code each digit position of the number represent specific weight. for ex:- In decimal code if number is 567, the weight of 5 is 100's and weight of 6 is 10's and weight of 7 is 1's,

In weighted binary code each digit has a weight of 8421.

ii) Non-weighted codes:-

These are not assigned with any weight to each digit position i.e. it is not assigned fixed value

* Excess-3 and gray codes are non-weighted codes

iii) Reflective codes:-

A code is said to reflective when the code for 9 is the complement for the code '0', 8 for 1, 7

2, 6 for 3, 5 for 4.

iv) Sequential codes :-

In Sequential codes each succeeding is 1' binary number greater than its preceding code 8421 and excess - 3 are Sequential.

v) Alpha numeric codes :-

The codes which consists of both numbers and alphabetic. characters are called alpha numeric codes.

The most commonly used alpha numeric codes are :- ASCII - American Standard code for Information Inter change

EBCDIC - Extended binary coded Decimal Inter change code.

HOLLERITH code

vi) Error defecting & correcting codes :-

Codes which allows only error detection are called error detecting codes.

Error Correcting codes are Sequence of numbers generated by Specific algorithm for defecting and removing errors in data that has been transmited over noisy channel.

Binary codes

- Weighted
  - Binary
  - BCD
    - 8421
    - 2421
    - 3321
    - 4221
    - 5211
    - 5311
    - 5421
    - 6311
    - 7421
    - $74\bar{2}\bar{1}$
    - $84\bar{2}\bar{1}$
- non-weighted
  - excess-3
  - Gray
  - five bit BCD codes
- Reflective
  - 2421
  - 5211
  - Excess-3
- Sequential
  - 8421
  - Excess-3
- Alpha numeric
  - ASCII
  - EBCDIC
  - Hollerith code
- Error detecting & Error correcting
  - Parity
  - Hamming

BCD Addition:-

The addition of two BCD numbers can be understood by considering the 3 cases that occur when two BCD digits ~~when two~~ are added.

i) Sum equals 9 or less with carry 0

let us consider 6 and 3

$$
\begin{array}{rl}
0110 & \text{- BCD for 6} \\
0011 & \text{- BCD for 3} \\
\hline
1001 & \text{- BCD for 9}
\end{array}
$$

The addition is carried out as in normal binary addition and the sum is 1001 which is BCD code for 9.

ii) Sum greater than 9 with carry 0
let us consider 6 and 8

0110 - BCD for 6
1000 - BCD for 8
1110 - Invalid BCD number (1110>9)

The Sum 1110 is an invalid BCD number, because the sum of the two digits exceeds 9. whenever this occurs the sum has to be corrected by the addition 6 (0110) in the invalid bcd number as shown in below

0110 - BCD for 6
1000 - BCD for 8
1110 - Invalid BCD number (1110>9)
0110 - Add 6 for correction
00010100 - BCD for 14

iii) Sum equals 9 or less with carry 1
let us consider 8 and 9

1000 BCD for 8
1001 BCD for 9
00010001 Incorrect BCD

In these case 0001 0001 is valid BCD number but it is incorrect to get the correct BCD result correction factor of 6 has to be added to the least significant digit sum

1000 - BCD for 8
1001 - BCD for 9
00010001 - Incorrect BCD
0110 - BCD Add for correction
00010111 = BCD for 17.

1) Perform each of the following decimal addition in 8421 BCD

i) 24 + 18   b) 48 + 58   c) 175 + 326   d) 589 + 199

c)
```
  175      0001 0111 0101
+ 326      0011 0010 0110
  ---      --------------
  501      0100 1001 1011
                     0110
           --------------
0101 0000 0001
           0100 1010 0001
                0110
           --------------
           0101 0000 0001
```

a)
```
  24       0010 0100
  18       0001 1000
  --       ---------
  42       0011 1100
                0110
           ---------
           0100 0010
```

b)
```
  48       0100 1000
  58       0101 1000
  ---      ---------
  106      1010 0000
                0110
           ---------
           1010 0110
           0110
           ---------
      0001 0000 0110
```

d)
```
  589      0101 1000 1001
  199      0001 1001 1001
  ---      --------------
  788      0111 0010 0010
                     0110
           --------------
           0111 0010 1000
                0110
           --------------
           0111 1000 1000
```

BCD Substraction :-

① 8 − 2 = 6 9 − 2 = 7 8 + 7 = ①5 — 9's complement

Add carry to result: 5 + 1 = 6

② 28 − 13 = 15 99 − 13 = 86 28 + 86 = ①14, 14 + 1 = 15

i) Perform the following decimal Substraction in 8421 BCD using 9's complement method.

i) 79 − 26 ii) 89 − 54

i) 79 − 26 = 53 99 − 26 = 73 79 + 73 = ①52 = 53

```
0111 1001
0010 0110
```

0111 1001
= 0111 0011

1110 1100
0110

1111 0010
0110

①0101 0010
1

0101 0011

ii) 89 − 54 = 35   99 − 54 = 45

45 : 0100 0101
89 : 1000 1001

①34 1000 1110
1 0110

35 1101 0100
0110

①0011 0100
1

0011 0101

carry = 0 = add
carry = 1

Excess-3 code :-

It is a modified form of a BCD number. It can be derived from the BCD code by adding 3 to the each code number.

for example decimal 12 can be represented in BCD as 0001 0010 now adding 3 to each digit we get excess-3 code as 0100 0101

Decimal digit	Excess-3
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

find the excess-3 code and its 9's complement for following decimal number

a) 592 b) 403

a) 592

0101	1001	0010	
0011	0011	0011	excess-3
1000	1100	0101	

999
592
407

407 =

0100	0000	0111
0011	0011	0011
0111	0011	1010

b) 403

0100	0000	0011	
0011	0011	0011	excess-3
0111	0011	0110	

999
403
596

596 =

0101	1001	0110
0011	0011	0011
1000	1100	1001

Excess-3 addition:-

To perform excess-3 addition add 2 excess-3 numbers.

i) If carry = 1 add 3 to the sum of the two digits

ii) If carry = 0 Substract 3

Eg:-

i) Add 8 and 6

8 → 1011 Excess 3 - 8

6 → 1001 Excess 3 - 6

00010100 carry occur

0011 0011

0100 0111 → Excess 3 - 14

8
6
14
0100 0111

2) Add 1 and 2

1 - 0100

2 - 0101

3 1001 carry not occur

0110

1001

(-) 0011 subtract - 3

0110 excess 3 for 3

D		B
0 - 0 = 0		0
1 - 0 = 1		0
1 - 1 = 0		0
0 - 1 = 1		1

Excess-3 Substraction:-

To perform excess-3 subsraction, we have to

1) Complement the subtrahend
2) Add complement subtrahend to minuend
3) If carry = 1 result is positive add 3 and end around carry
4) If carry = 0 result is negative subtract 3

Eg:- 8 - 5

```
  8          5 → 1000  (excess 3 for 5)
 -5
 ---             0111  (complement the subtrahend)
  3
  ↓
 0110            0111
                 1011
                ------
               (1)0010
                 0011
                ------
                 0101
                  → 1
                ------
                 0110
```

```
  5
 -8          1011  (excess - 3 for 8)
 ---
 -3          0100  (complement

   1000   (excess 3 for 5)
 (+) 0100 [complement of 8]
  ------
   1100
 (-) 0011       subtract 3
  ------
   1001
```

Perform the subtraction $(645)_{10} - (319)_{10}$ in excess-3 using the 9's complement method

```
  999
  319          645 = 1001  0111  1000
  ---          680 = 1001  1011  0011
  680                 --------------------
                  (1)0011  0010  1011
                 (+)0011 (+)0011  0011 (-)
                     --------------------
  645                0110  0101  1000
  319                               → 1
  ---                --------------------
  326                0110  0101  1001
```

Gray code :-

It is a Special case unit distance code. In unit distance code bit pattern for two consecutive numbers differ in only one bit position. These codes are also called as cyclic codes.

XOR operation

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

S.NO	Decimal No	Binary	Gray
	0	0000	0000
	1	0001	0001
	2	0010	0011
	3	0011	0010
	4	0100	0110
	5	0101	0111
	6	0110	0101
	7	0111	0100
	8	1000	1100
	9	1001	1101
	10	1010	1111
	11	1011	1110
	12	1100	1010
	13	1101	1011
	14	1110	1001
	15	1111	1000

Gray to Binary Conversion:-

The most significant bit of the binary number is the same as the most significant bit of the gray code number.

To obtain the next binary digit perform an XOR operation between the bit just return down and the next gray code write down the result.

Repeat step-2 until all gray code

Eg:- Convert gray code 101011 into its Binary equivalent.

1 0 1 0 1 1

1 1 0 0 1 0

Convert gray code 110011 into its Binary equivalent

1 1 0 0 1 1

= 1 0 0 0 1 0

Binary to gray conversion:-

let us represent binary numbers $B_1, B_2, \ldots B_n$ and its equivalent gray code numbers $G_1, G_2, \ldots G_n$ with this representation gray code bits are obtained from the binary bit as follows.

$$G_1 = B_1$$

$$G_2 = B_1 \oplus B_2$$

$$G_3 = B_2 \oplus B_3$$

$$\vdots$$

$$G_n = B_{n-1} \oplus B_n$$

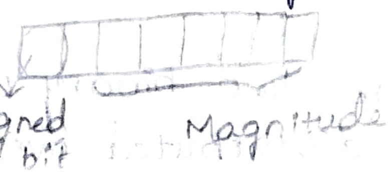
Eg:- Convert 10111011 in binary into its Gr equivalent Gray code

10111011

11100110

Signed binary numbers:-

In general the signed numbers are represented with '+' sign for positive numbers and '−' sign for negative numbers. Because of hardware limitations in computer both the signs are represented with binary numbers only. The left most bit of binary number represent sign of the number. If MSB is '1' → Negative number. If MSB is '0' → positive number. Represent of signed binary number with 8 bits.



Eg:- 1. find out 1's and 2's complement for $(-45)_{10}$ in arithmatic

2 | 45
2 | 22 − 1
2 | 11 − 0
2 | 5 − 1
2 | 2 − 1
  1 − 0

45 = 00101101

−45 = 1101.0010 = 1's
                 1
2's = 11.010011

Boolean algebra:-

It is defined with a set of elements, a set of operators and number of rules, laws, theorems and postulates.

Set:-

It is a collection of objects usually having a common properties

If 'E' is a set x, y are certain elements we can written as $S = \{x, y\}$

Associative law:-

A binary operator '*' on a set 'S' is said to be associative $(A * B) * C = A * (B * C) \ \forall A, B, C \in S$

Commutative law:-

A binary operator '*' on a set 'S' is said to be commutative $A * B = B * A \ \forall A, B \in S$

Identity element:-

A set 'S' is said to have an identity element with respect to a binary operator '*' on 'S' if there exist an element

$e * A = A * e \ \forall \ A \in S$

A set 's' is said to have an identity element with respect to a binary operator '+' on 's' if there exist an element

$e + A = A + e \ \forall \ A \in S$

Distributive law:

If '+' and '.' are two binary operators on a set 's', '+' is distributed over the function

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

* The additive identity is '0'
* The additive inverse is subtraction
* The Multiplicative identity is 1
* The Multiplicative inverse is division.

Rules in Boolean Alzebra:-

* The symbol which represent an orbitary element of an boolean algebra is known as variable. Any single variable or a function of several variable can have either a 'one' or 'zero'
* A complement of a variable is represented by (-) 'bar' (or) '1'. A will be denoted by $\bar{A}$.
* The logical 'AND' operator of two variables is represented either by writing a dot (.) between two variables such as A.B (or) AB. 'AND' operation between three variables can be represented as A.B.C (or) ABC
* The logical 'OR' operator of two variables is represented either by writing a '+' sign between the two variables such as A+B
* The logical 'OR' operator in the boolean algebra with variables having value either a 'zero' (or) 'one'. gives following results.

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

from the above results following rules are defined in the boolean algebra.

Rule 1 :- $0 + A = A$ $\quad$ $1 + \boxed{0} = 0$

$A + 0 = A$ $\quad$ $0 + \boxed{1} = 1$

Rule 2 :- $1 + A = 1$ $\quad$ $1 + \boxed{0} = 1$

$A + 1 = 1$ $\quad$ $1 + \boxed{1} = 1$

Rule 3 :- $A + A = A$ $\quad$ $\boxed{0} + \boxed{0} = \boxed{0}$, $\boxed{1} + \boxed{1} = \boxed{1}$

Rule 4 :- $A + \bar{A} = 1$ $\quad$ $\boxed{0} + \boxed{1} = 1$

(or) $\bar{A} + A = 1$ $\quad$ $\boxed{1} + \boxed{0} = 1$

* The logical AND operator in the boolean Algebra with variables having values either '0' or '1' gives the following result.

$0 . 0 = 0$
$0 . 1 = 0$
$1 . 0 = 0$
$1 . 1 = 1$

Rule 5 :- from The above result following rules are the define the Boolean Algebra.

Rule 5 :- $0 . A = 0$ $\quad$ $0 . \boxed{0} = 0$

$A . 0 = 0$ $\quad$ $0 . \boxed{1} = 0$

Rule 6 :-

$1 . A = A$ $\quad$ $1 . \boxed{0} = 0$

$A . 1 = A$ $\quad$ $1 . \boxed{1} = 1$

Rule 7 :- $A . A = A$ $\quad$ $0 . 0 = 0$

$1 . 1 = 1$

Rule 8 :- $A . \bar{A} = 0$ $\quad$ $0 . 1 = 0$

(or) $\bar{A} . A = 0$ $\quad$ $1 . 0 = 0$

* The NOT operator in the boolean algebra with variable having value either a '0' or '1'. use the following results.

$\bar{0} = 1$, $\bar{\bar{0}} = 0$

$\bar{1} = 0$ $\quad$ $\bar{\bar{1}} = 1$

Rule 9 :- $\bar{\bar{A}} = A$

Axiomatic Definition of Boolean Algebra (postulates):-

There are 6 hunting ~~ton~~ postulates

1. a) The structure is closed with respect to the operator '+'.
   b) The structure is closed with respect to the operator '.'.
2. a) The element '0' is an identity element with respect to 'x'. $x+0 = 0+x = x$
   b) The element '1' is an identity element with respect to 'x' $x.1 = 1.x = x$
3. a) The structure is commutative with respect the operator '+'. $A+B = B+A$
   b) The structure is commutative with respect the operator '.'. $A.B = B.A$
4. a) The operator '+' distributive over '.'
   $x+(y.z) = (x+y).(x+z)$
   b) The operator '.' distributive over '+'
   $x.(y+z) = (x.y)+(x.z)$
5. a) for every element $x \in B$ their exist an element $\bar{x} \in B$
   $$x+\bar{x} = 1$$
   $$x.\bar{x} = 0$$
6. There exist atleast 2 elements $x, y \in B$ Such that $x \neq y$

Laws of Boolean Algebra:-

Three of the basic laws of boolean algebra are the same as in ordinary algebra.

They are
1) commutative law
2) Associative law
3) Distributive law.

1) Commutative law :-

Law 1 :- $A + B = B + A$

This states that the order in which the variables ORed and makes no difference in the output. The truth tables are identical.

A OR B is same as B OR A

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

=

A	B	B+A
0	0	0
0	1	1
1	0	1
1	1	1

Law 2 :- $A . B = B . A$

The commutative law of multiplication states that the order in which the variable are ANDed and makes no difference in the output.

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

=

A	B	B.A
0	0	0
0	1	0
1	0	0
1	1	1

It follows that $A + B + C = B + C + A$ and $A + C = C + A$

Similarly $ABCD = BACD$

2) Associative law :-

Law 1 :- $A + (B + C) = (A + B) + C$

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

=

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

Law 2 = $(AB)C = A(BC)$

A	B	C	AB	(AB)C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

=

A	B	C	BC	A(BC)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

3) Distributive law:-

law 1 :- $A(B+C) = AB + AC$

A	B	C	B+C	A(B+C)	AB	AC	AB + BC
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

law 2 :- $A + (BC) = (A+B)(A+C)$

A	B	C	BC	A+(BC)	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

Additional rules in Boolean Algebra:-

Rule 10:- $A + AB = A$

$L.H.S = A + AB$

$= A(1+B)$ $\qquad 1+B = 1$ (R-2)

$= A$

Rule 11 :-

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

L.H.S $= A + \bar{A}B$ Rule 10 :: $A = A + AB$

$= A + AB + \bar{A}B$ Rule 4 : $A + \bar{A} = 1$

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

$= A + B(1)$

$= A + B$

Rule 12 :- $(A+B)(A+C) = A + BC$

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

$= A.A + AB + AC + BC$ Rule 7 : $A.A = A$

$= A + AB + AC + BC$ Rule 2 : $1 + B = 1$

$= A(1+B) + AC + BC$

$= A + AC + BC$

$= A(1+C) + BC$

$= A + BC$

Theorems in Boolean Algebra:-

Demorgon's theorem :-

Demorgon's suggested two theorems that form an important part of Boolean algebra in the equation form they are

1) $\overline{AB} = \bar{A} + \bar{B}$ ~~2) (A+~~

The complement of a product is equal to the sum of the complements. This is shown in below.

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

2) $\overline{A+B} = \bar{A}.\bar{B}$

The complement of the sum is equal to the product of the complements. The truth table is shown below.

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

Consensus theorem:-

The simplification of Boolean expression in the form of $AB+\bar{A}C+BC$ the term BC is redundant and can be eliminated to form the equivalent expression $AB+\bar{A}C$. The theorem used for the simplification is known as consensus theorem. and it stated as

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

L.H.S

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

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

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

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

$= AB+\bar{A}C$

Eg:- Solve the given expression using consensus theorem

$$\bar{A}\bar{B}+AC+\overline{BC}+\bar{B}C+AB$$

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

$AB+\bar{A}C+BC = AB+\bar{A}C$ (consensus theorem)

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

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

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

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

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

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

Dual of consensus theorem:-

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

L.H.S

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

$= (0 + AC + \bar{A}B + BC)(B+C)$

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

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

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

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

$= \bar{A}B(1+C) + AC(1+B) + BC$

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

R.H.S

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

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

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

Logic Gates:-

1) Basic gates are AND, OR, NOT
2) Universal gates are NAND, NOR
3) Special gates XOR, XNOR.

AND Gate :-

1) It is a multi input and single output
2) It only turn on when both inputs are on
3) If only one input is on it spits output OFF

A	B	Y=AB
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate :-

1) It is a multiple input and single output.
2) OR needs one input to be 'ON'.

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate:-

1) NOT is the most simple logic gate
2) All it does is takeing an input that is either on or OFF and
3) splits out the opposite. So for a '1' it will give '0' and for a '0' it will give 1

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

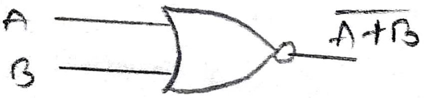
NAND Gate :-

1) NAND Gate is inversion of AND Gate

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

NOR Gate :-

1) NOR Gate is inversion of OR gate



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

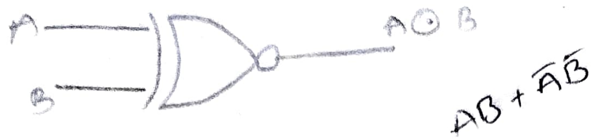
XOR Gate:- (Exclusive OR)

1) When two inputs are same then output is equal to 0.
2) When two inputs are different then output is equal to 1.

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

XNOR Gate:- (Exclusive NOR)

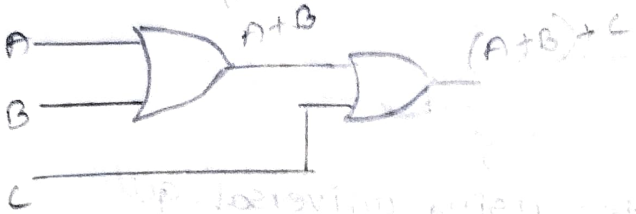
1) inversion of XOR gate

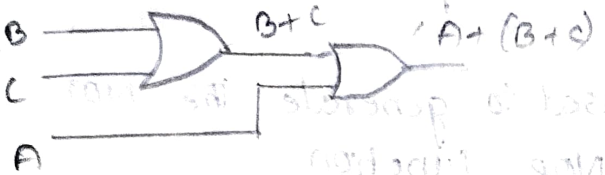


A	B	A⊙B
0	0	1
0	1	0
1	0	0
1	1	1

Logic gate for Associative law :-

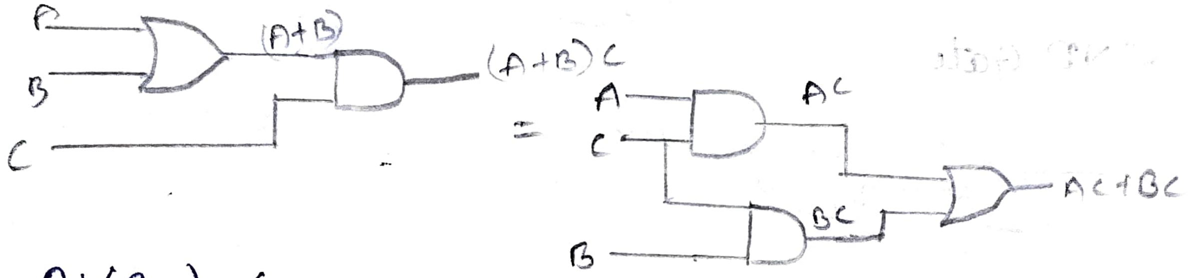
$(A+B)+C = A+(B+C)$



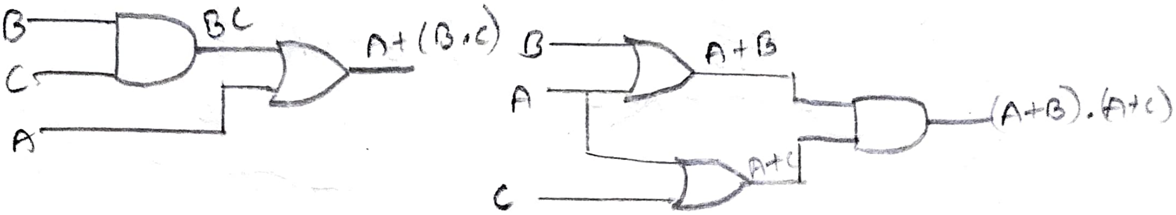


Logic gate for Distributive law :-

$A(B+C) = AB+AC \Rightarrow (A+B)C = AC+BC$

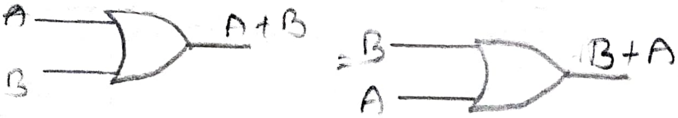


$A+(B.C) = (A+B).(A+C)$



Commutative law :-

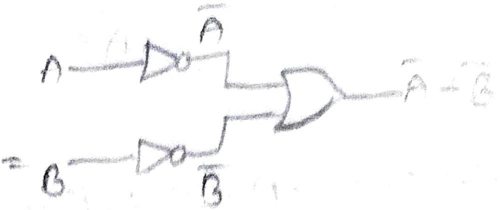
$A+B = B+A$



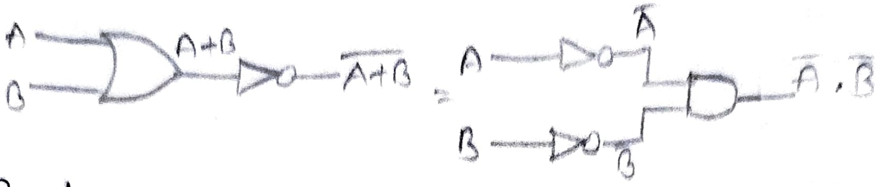
$A.B = B.A$

Demorgan's law

1) $\overline{AB} = \overline{A} + \overline{B}$



2) $\overline{A+B} = \overline{A}.\overline{B}$

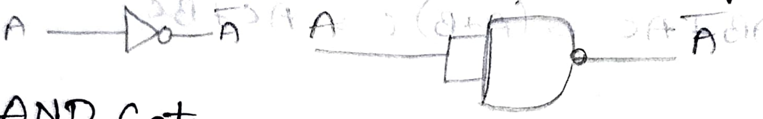


Realization of logic gates using universal gates:-

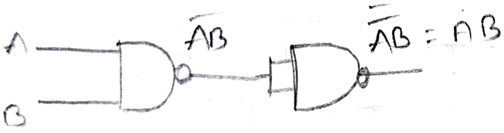
NAND Gate :-

NAND Gate can be used to generate the NOT function, AND, OR and NOR function.

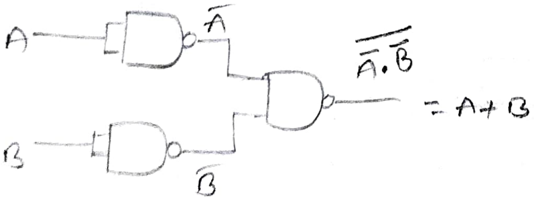
NOT Gate operation using NAND gate:-



AND Gate



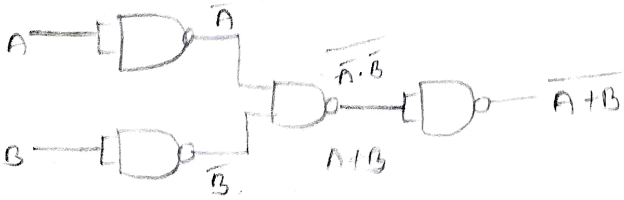
OR Gate:-



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

$= A+B$

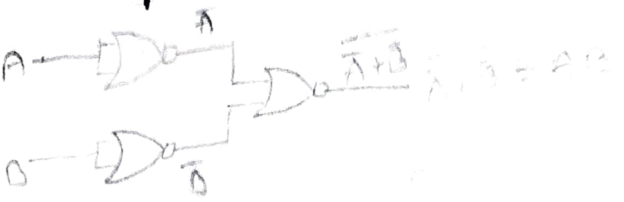
NOR gate :-



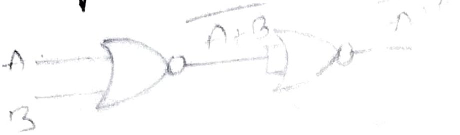
NOR gate realization :-

NOT gate by using NOR gate

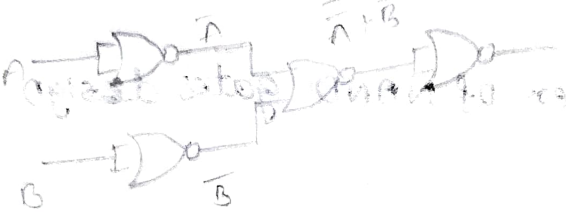
AND gate :-



OR gate :-



NAND gate :-



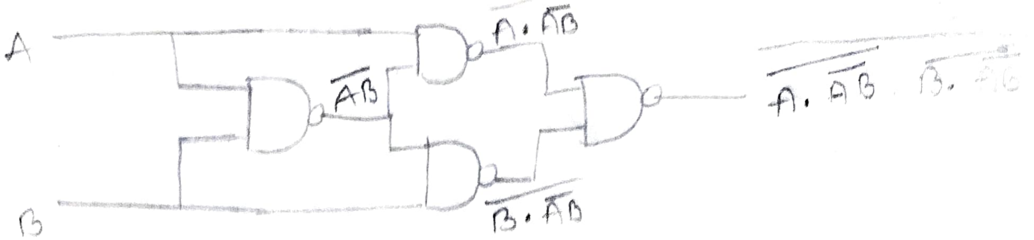
1. EX-NOR and EX-OR are complement to each other

$$\overline{x \oplus y} = \overline{x\bar{y} + \bar{x}y}$$
$$= \overline{x\bar{y}} \cdot \overline{\bar{x}y}$$
$$= (\bar{x}+\bar{\bar{y}}) \cdot (\bar{\bar{x}}+\bar{y})$$
$$= (\bar{x}+y)(x+\bar{y})$$
$$= x\bar{x} + \bar{x}\bar{y} + xy + y\bar{y}$$
$$= 0 + xy + \bar{x}\bar{y} + 0$$
$$= xy + \bar{x}\bar{y}$$
$$= x \odot y$$

$$\overline{x \odot y} = \overline{xy + \bar{x}\bar{y}}$$
$$= \overline{xy} \cdot \overline{\bar{x}\bar{y}}$$
$$= \overline{xy} \cdot xy$$
$$= (\bar{x}+\bar{y}) \cdot (x+y)$$
$$= x\bar{x} + y\bar{x} + y\bar{y} + x\bar{y} = 0 + y\bar{x} + 0 + x\bar{y} = x\bar{y} + \bar{x}y = x \oplus y$$

2. Design Ex-OR gate with minimum number of NAND gates.

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



$$\overline{\overline{A\cdot\overline{AB}}} + \overline{\overline{B\cdot\overline{AB}}}$$

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

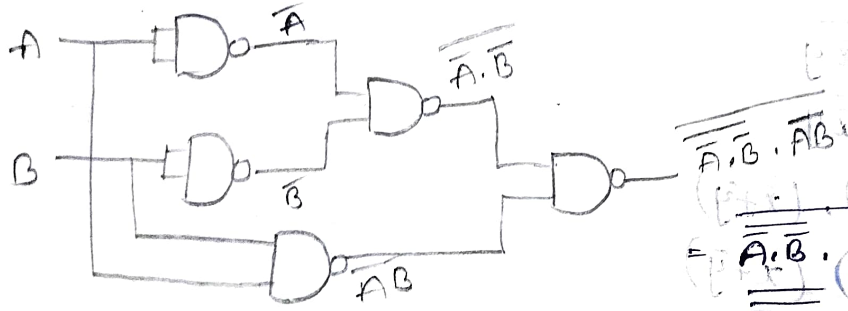
$$= A\cdot(\bar A+\bar B) + B(\bar A+\bar B)$$

$$= A\cdot\bar A + A\bar B + B\cdot\bar A + B\cdot\bar B$$

$$= A\bar B + B\bar A$$

3. With minimum number of NAND gates design EX-NOR gate

$$A\odot B = AB + \bar A\bar B$$



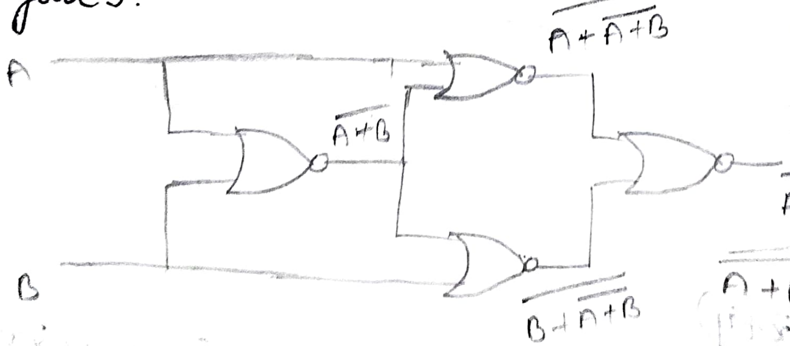
$$\overline{\overline{\bar A\cdot\bar B}\cdot\overline{AB}}$$

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

$$= \overline{\overline{\bar A\cdot\bar B}} + \overline{\overline{AB}}$$

$$= \bar A\cdot\bar B + AB$$

4. Design EX-NOR gate with minimum number of NOR gates.



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

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

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

$$(A+(\bar A\cdot\bar B))\cdot(B+(\bar A\cdot\bar B))$$

$$(A+\bar A)(A+\bar B)(B+\bar A)(B+\bar B)$$

$$(A+\bar B)(B+\bar A)$$

$$AB + A\bar A + B\bar B + \bar A\bar B$$

$$AB + \bar A\bar B$$

Properties of XOR gate:-

1) $A \oplus 1 = \bar{A}$

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

$= A.\bar{1} + \bar{A}\,1$

$= A.0 + \bar{A}$

$= \bar{A}$

2) $A \oplus 0 = A$

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

$= A.\bar{0} + \bar{A}.0$

$= A.1 + 0$

$= A$

3) $A \oplus A = 0$

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

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

$= 0$

$A \oplus A = 0$

4) $A \oplus \bar{A} = 1$

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

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

$= A + \bar{A}$

$= 1$

5) $AB \oplus AC = A(B \oplus C)$

$= AB.\overline{AC} + \overline{AB}.AC$

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

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

$= 0 + AB\bar{C} + 0 + \bar{B}AC$

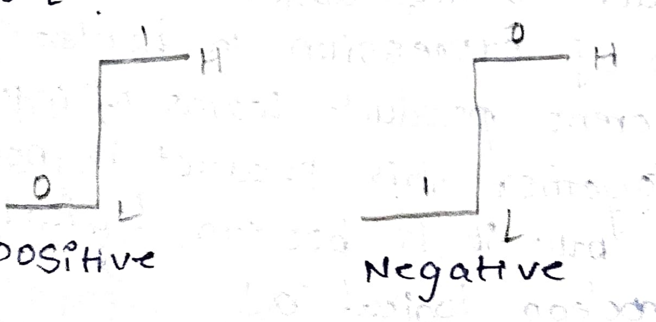
$= A(B\bar{C} + \bar{B}C)$

$= A(B \oplus C)$

t. ~~If A⊕B=c~~ ⊕

Positive and negative logics:-

A binary logic at the input and output of any gate has one of two values except during transition. one signal represents logic 1 the another signal represent logic 0. higher signal Designated as 'H' and lower signal designated as 'L'.



Negative logic OR gate

Negative OR   positive NAND

Negative AND   positive NOR

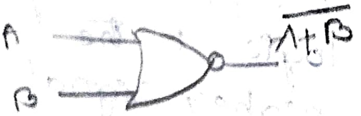
Traditional Symbols for active high

AND:-

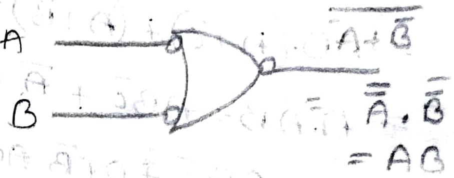
OR:-

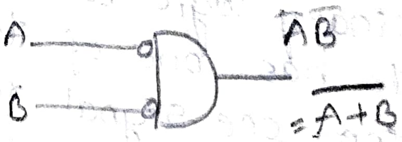
NAND:-

NOR:-



Traditional Symbols for active low





Sum of product (sop) form:-

The Sum of product is the abrevated form of 'Sop'. It is a form of expression in boolean algebra in which different product terms of inputs are being Summed together this product is not arithmatical multiply but it is boolean logical "AND" the Sum is boolean logical 'OR'.

Ex :- $f(A,B,C) = \overbrace{ABC + \bar{A}B\bar{C}}^{\text{sum}}$

product

$f(P,Q,R,S) = \underbrace{\overbrace{P\bar{Q} + QR + RS}^{\text{sum}}}_{\text{product}}$

The SOP form is also known as disjunctive normal form (or) disjunctive

product of sum :-

A product of sum is any groups of sum terms ANDed together

$F(A,B,C) = (A+B) \overset{\text{Product}}{\cdot} (B+C)$

$F(P,Q,R,S) = (P+Q) \cdot \overset{\text{Sum}}{(R+S)} \cdot (P+S)$

product sum

The Pos is also known as conjuctive normal form or conjunctive normal form.

Each of these product of sum expression consists of two or more sum terms (that are ANDed together. Each sum term consist of one or more literals appearing in either complemented or uncomplemented form.

Canonical form :-

In boolean algebra boolean function can be expressed as canonical disjunctive normal form known as minterm and same are expressed as canonical conjunctive normal form known as maxterm.

In minterm we look for the function where output result is '1'. In maxterm we look for the function where output result is 0.

We perform sum of minterm also known as sum of product.

We perform product of maxterm also known a product of sum.

Boolean function expressed as a sum of minterms (or) product of maxterms are said to be canonical forms.

Standard form:-

A boolean variable can be expressed in either True form (or) complemented form. In standard form boolean function will contain all the variables in either true form (or) complemented form while in canonical no. of variables depend on the output of SOP or POS

Truth Table representing minterms and maxterms

x	y	z	Minterms product terms	Maxterms sum terms
0	0	0	$m_0 = \bar{x}.\bar{y}.\bar{z}$ minterms $(\bar{x},\bar{y},\bar{z})$	$M_0 = x+y+z$ maxterms (x,y,z)
0	0	1	$m_1 = \bar{x}.\bar{y}.z$ mintems $(\bar{x},\bar{y},z)$	$M_1 = x+y+\bar{z}$ maxterms $(x,y,\bar{z})$
0	1	0	$m_2 = \bar{x}.y.\bar{z}$ minterms $(\bar{x},y,\bar{z})$	$M_2 = x+\bar{y}+z$ maxterms $(x,\bar{y},z)$
0	1	1	$m_3 = \bar{x}.y.z$ minterms $(\bar{x},y,z)$	$M_3 = x+\bar{y}+\bar{z}$ maxterms $(x,\bar{y},\bar{z})$
1	0	0	$m_4 = x.\bar{y}.\bar{z}$ minterms $(x,\bar{y},\bar{z})$	$M_4 = \bar{x}+y+z$ maxterms $(\bar{x},y,z)$
1	0	1	$m_5 = x.\bar{y}.z$ minterms $(x,\bar{y},z)$	$M_5 = \bar{x}+y+\bar{z}$ maxterms $(\bar{x},y,\bar{z})$
1	1	0	$m_6 = x.y.\bar{z}$ minterms $(x,y,\bar{z})$	$M_6 = \bar{x}+\bar{y}+z$ maxterms $(\bar{x},\bar{y},z)$
1	1	1	$m_7 = x.y.z$ mintrams (x,y,z)	$M_7 = \bar{x}+\bar{y}+\bar{z}$ maxterms $(\bar{x},\bar{y},\bar{z})$

Sum of Minterms :-

The minterms whose sum defines the boolean function are those which gives the one's of a function in a truth table. Since the function can be either '1' or '0' for each ~~term~~ minterm. Since there are 2^n minterms one can calculate all the functions that can be formed with n variables to the $(2^{(2^n)})$

1. Express the boolean function $F = A + \bar{B}C$ as standard sum of minterms.

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

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

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

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

$= 111 \quad 110 \quad 101 \quad 100 \quad 101 \quad 001$

$= m_7 \quad m_6 \quad m_5 \quad m_4 \quad m_5 \quad = m_1$

$F = m_1 + m_4 + m_5 + m_6 + m_7$

$= \Sigma_m (1, 4, 5, 6, 7)$

Product of maxterms:-

To express a boolean function as a product of maxterms it must first to be brought into a form of OR terms

1. Express the boolean function $F = xy + \bar{x}z$ as a product of maxterm

$F = xy + \bar{x}z$

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

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

$= (\bar{x} + y + (zz'))(x + z + (yy'))(y + z + (xx'))$

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

$= 100 \quad 101 \quad 000 \quad 010 \quad 000 \quad 100$

$M_4 \quad M_5 \quad M_0 \quad M_2 \quad M_0 \quad M_4 = \Pi_M (0, 2, 4, 5)$

2. Determine the canonical sum of products represent the following function

$F(x,y,z) = z + (\bar{y}+y)(x+\bar{y})$ The given function is not in standard form

$F(x,y,z) = z + (y'+y)(x+y')$

$= z + (xy' + y'y' + xy + yy')$

$= z + xy' + y'y' + xy + yy'$

$= z + xy' + y' + xy$

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

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

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

The above expression converted into canonical sum of product

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

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

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

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

3. Simplify the following 3 variable expressions using boolean algebra $y = \Sigma m(1,3,5,7)$.

Given boolean algebra

$Y(A,B,C) = \Sigma m(1,3,5,7)$

$= m_1 + m_3 + m_5 + m_7$

001 011 101 111

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

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

$= \bar{A}C + AC$

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

$= C$

4. Convert the given expression in standard POS form. $Y = (A+B)(B+C)(A+C)$

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

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

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

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

$$= \quad 000 \qquad 001 \qquad 100 \qquad 010$$

$$= \quad M_0 \qquad M_1 \qquad M_4 \qquad M_2$$

$$= \Pi_M(0,1,2,4)$$

5. convert the given expression in standard POS form
$Y = A.(A+B+C)$.

Given
$$Y = A.(A+B+C)$$
$$= (A+BB'+CC')(A+B+C)$$
$$= (A+BB'+C)(A+BB'+C')(A+B+C)$$
$$= (A+C+B)(A+C+B')(A+C'+B)(A+C'+B')$$
$$(A+B+C)$$
$$= (A+B+C)(A+B'+C)(A+B+C')(A+B'+C')$$

$$= \quad 000 \qquad 010 \qquad 001 \qquad 011$$

$$= \quad M_0 \qquad M_2 \qquad M_1 \qquad M_3$$

$$\Pi_M(0,1,2,3)$$

Note :-
* Standard SOP form or minterm canonical form
* Standard POS form or maxterm canonical form

Steps to convert SOP to standard SOP form

Step-1 :- find the missing literal in each product.

Step-2 :- AND each product term having missing literal with terms form by ORing the literal and its complement form.

Step-3 :- expand the terms by applying distributive law and reorder the literals in the product form.

Step-4 :- Reduce the expression by omitting repeated product terms

Steps to convert pos to standard pos form.

Step-1 :- find the missing literal in each Sum term

Step-2 : OR each sum term having missing literal with terms form by ANDing the literal and its complement

Step-3 :- Expand the terms by applying distributive law and re-order the literals in the sum term.

Step-4 :- Reduce the expression by omitting repeated sum terms.

Complements of the given function :-

The Complement of the function can be find out in 2 ways.

1) Demorganization 2) Duality

1) Demorganization :-

By applying demorgan's theorem we are going to find out complement of a given function.

Eg :- $F_1 = x(\bar{y}\bar{z} + yz)$

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

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

$$= \overline{x\bar{y}\bar{z}} \cdot \overline{xyz}$$

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

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

2) Duality:-

Step-1:- find out the duality of the given expression.

Step-2:- complement each and every literal

Eg:- $F_1 = xy\bar{z} + xyz$

Step-1:- $(x+\bar{y}+\bar{z}).(x+y+z)$

Step2:- $\bar{F_1} = (\bar{x}+y+z)(\bar{x}+\bar{y}+\bar{z})$

K-map:-(karnaugh map)

In solving of boolean expression by using boolean laws and theorems here so many problems occur because there are some predicted steps to remove such problems to we are going to a systematic procedure to simplify the given boolean expression this method is known as map method.

Map method is proposed by veitch and it is modified by karnaugh hence it is known as veitch diagrams or karnaugh maps.

k-map is used for simplifying both SOP and POS expressions.

The map method presented here provide simple straight forward procedure for minimizing boolean functions. These method may be regarded as pictorial form of truth table.

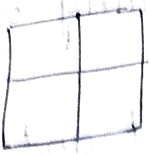
k-map is a diagram made of squares with each square representing one minterm of the function in map minterms are arranged not in a binary sequence but in sequence similar to grey code.

The number of variable in a given function is equal to n then the no. of cells needed in the k-map is equal to 2^n.

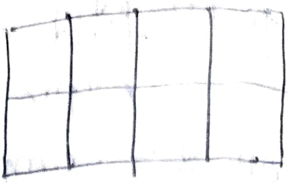
K-maps are used to simplify complex equations

$1 = 2^1 = 2$

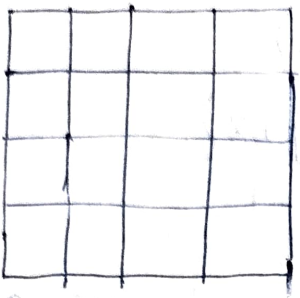
2 variables $= 2^2 = 4$ cells



3 variables $= 2^3 = 8$



4 variables $= 2^4 = 16$ cells



1, 2, 3 and 4 variable maps with product terms

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

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	$\bar{A}\bar{B}$ 0	$\bar{A}B$ 1
A 1	$A\bar{B}$ 3	AB 2

A \ BC	$\bar{B}\bar{C}$ 0	$\bar{B}C$ 1	BC 3	$B\bar{C}$ 2
$\bar{A}$ 0	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}BC$ 3	$\bar{A}B\bar{C}$ 2
A 1	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	ABC 7	$AB\bar{C}$ 6

AB \ CD	$\bar{C}\bar{D}$ 0	$\bar{C}D$ 1	CD 3	$C\bar{D}$ 2
$\bar{A}\bar{B}$ 0	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
$\bar{A}B$ 1	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
AB 3	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
$A\bar{B}$ 2	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 11	$A\bar{B}C\bar{D}$ 10

Another way to represent variable maps.

A	
$\bar{A}$ 0	$\bar{A}$ m_0 0
A 1	A m_1 1

A \ B	$\bar{B}$	B
$\bar{A}$	$\bar{A}\bar{B}$ m_0 0	$\bar{A}B$ m_1 1
A	$A\bar{B}$ m_2 2	AB m_3 3

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$\bar{A}\bar{B}\bar{C}$ m_0 0	$\bar{A}\bar{B}C$ m_1 1	$\bar{A}BC$ m_3 3	$\bar{A}B\bar{C}$ m_2 2
A	$A\bar{B}\bar{C}$ m_4 4	$A\bar{B}C$ m_5 5	ABC m_7 7	$AB\bar{C}$ m_6 6

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	$\bar{A}\bar{B}\bar{C}\bar{D}$ m_0 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
$\bar{A}B$	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
AB	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
$A\bar{B}$	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 12	$A\bar{B}C\bar{D}$ 11

Representation of truth table on k-map

XOR :

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

A \ B	$\bar{B}$	B
$\bar{A}$	0_0	1_1
A	1_2	0_3

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

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

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$_0$	$_1$	$_3$	1_2
A	1_4	1_5	1_7	$_6$

~~$A\bar{B}$~~ $\bar{A}B\bar{C} + A\bar{B} + AC$

A	B	C	D	Y
0	0	0	0	
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	a	1
1	1	1	1	1

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$	
$\bar{A}\bar{B}$	1_0	$_1$	1_3	x_2	(0, 4)
$\bar{A}B$	1_4	1_5	1_7	$_6$	(3, 7)
AB	1_{12}	$_{13}$	1_{15}	1_{14}	(5, 7)
$A\bar{B}$	$_8$	$_9$	$_{11}$	1_{10}	(12, 14)
					(7, 15)
					(14, 10)

$\bar{A}\bar{C}\bar{D} + \bar{A}CD + \bar{A}BD + AB\bar{D} + BCD$

$+ AC\bar{D}$

1. plot boolean expression Y = AB... ...BC...
on The k-map

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

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1 1	3	2
A	4	5	1 7	1 6

2. plot boolean expression $y = \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BC\bar{D} + A\bar{B}CD + AB\bar{C}D$. on the k-map

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

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

3. Simplify the boolean expression $F(x, y, z) \sum(1, 2, 4, 7)$

x \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0	1 1	3	1 2
x	1 4	5	1 7	6

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

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

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

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

$$x \oplus y \oplus z$$

Grouping cells for Simplification :-

Grouping of two adjacent 1's :- (pair)

x \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0	1 1	1 3	1 2
x	1 4	1 5	7	1 6

pair 3 $(\bar{x}y)$

pair 2 $(x\bar{z})$

pair 1 $(\bar{y}z)$

A group of two adjacent cells of a k-map forms a pair & eliminates one variable in simplification by k-map.

pair-1 is formed by combining two vertical adjacent cells

pair-2 is formed by combining leftmost and right most cells.

pair-3 is formed by combining two horizontal adjacent cells.

Grouping of 4 adjacent cells (Quad):-

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1	3	2
A	1 (4)	1 (5)	1 (7)	1 (6)

→ A

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

$A\bar{D}$

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

AB

AD AC

A group of 4 adjacent cells in k-map forms a quad it eliminates two variables in simplification of k-map.

Grouping of 8 adjacent one's (octal):-

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

BD

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

→ B

A group of 8 adjacent cells in k-map forms an octal it reduces 4 variables to a single variable

1. Reduce the following expression using k-map

$F = \Sigma m(1, 3, 4, 5, 8, 9, 13, 15)$

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

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

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

ABD

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

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

2) plot the Boolean expression $Y = (A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)(A+B+\bar{C})$

010   011   110   001

$A \backslash BC$	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	(0)	0 (1)	0 (3)	0 (2)
A	(4)	(5)	(7)	0 (6)

$A+B+\bar{C}$

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

$A+\bar{B}+C$

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

Prime implicant :-

Each Square or rectangle formed by grouping of adjacent minterms is called prime implicant

false prime implicant :

Grouping of maxterms is called false prime implicant.

Non prime implicant and Non false prime implicant:

A group formed by one minterm is Non prime implicant and one maxterm is called Non false prime implicant.

Essential prime implicant and Essential false prime implicant:-

The minimum no. of prime implicant required to include all one's present in the K-map are called essential prime implicant.

The minimum no. of false prime implicant required to include all zero's present in the k-map are called essential false prime implicants.

Redundant prime implicant and Redundant false prime implicant.

All one's in a group are covered by some other group is called Redundant prime implicant and all zero's in a group are covered by some other group is called (RFPI)

Selective prime implicant (SPI) and selective false prime implicant (SFPI)

A group which is neither an ~~es~~ EPI nor RPI is called SPI.

A group which is neither an EFPI nor RFPI is called SFPI.

1. Minimize the expression $y = A\bar{B}C + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$.

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad 101 \quad\;\; 001 \quad\;\; 011$

$\;\;100 \quad\quad 000$

Step-1:- draw the 3 variable k-map

Step-2:-

$A \backslash BC$	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1 (0)	1 (1)	1 (3)	(2)
A	1 (4)	1 (5)	(7)	(6)

Step-2:- There are no isolated ones

Step-3:- 1 in the cell-3 is adjacent only to 1 in the cell-1. This pair is combined referred to as group-1.

Step-4:- There is no octat but there is a quad this quad cells 0, 1, 4, 5 form a quad. This form a group-2.

Step-5 :- All one's have already been grouped

Step-6 :- $y = \bar{A}C + \bar{B}$ each group generated a term in the expression.

1. Minimize the expression

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

001   011   110   100   000

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0 (0)	0 (1)	0 (3)	(2)
A	0 (4)	(5)	(7)	0 (6)

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

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

$\bar{y} = \overline{\bar{B}\bar{C} + \bar{A}C + A\bar{C}}$

$\bar{y} = \overline{\bar{B}\bar{C}} \cdot \overline{\bar{A}C} \cdot \overline{A\bar{C}}$

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

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

Don't Care Condition:-

* The combination for which the value of function is not specified are called don't care condition (or) Incompletely specified function.
* The map entries which does not have any effect on the system on its occurance are referred as dont care map entries or don't care condition
* It is denoted by d, x, ϕ.
* The value of don't care condition can be 0 or 1. for maxterm and minterm.

1. $F(A,B,C) = \Sigma m\,(0,2,7) + \phi\,(3,4)$.

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1 (0)	(1)	X (3)	1 (2)
A	X (4)	(5)	1 (7)	(6)

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

2. Find the Reduced SOP form of the following function $f(A,B,C,D) = \Sigma m(1,3,7,11,15) + \Sigma d(0,2,4)$

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

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