

- 02-08-18  
Thursday S is the most popular and widely used number system.
- ## Number Systems
- (1) Binary Numbers
- ⇒ The Binary Number System has only two digits 0 and 1.
  - ⇒ The Binary System with its 2<sup>n</sup> digit is a base two system.
  - ⇒ The positions of a 1 or 0 in a binary number indicates its weight or value with respect to the Number. Just as in the decimal system, the position of a digit determines the value of that digit.

⇒ The weights in the binary No. are based on power of two.

### Counting in Binary

0 - 0000      13 - 1101

1 - 0001      14 - 1110

2 - 0010      15 - 1111

3 - 0011

4 - 0100

5 - 0101

6 - 0110

7 - 0111

8 - 1000

9 - 1001

10 - 1010

11 - 1011

12 - 1100

13 - 1101

14 - 1110

15 - 1111

⇒ To count from 0 to 15 4 beats are required.

⇒ The value of a beat is determined by its position in the Number's system.

⇒ With n beats we can count up to a number equal to

$$2^n - 1$$

$\Rightarrow$  The largest Number is  $2^n - 1$

$\therefore$  eg: with 6 beats we can count from 0 to 63 ie,  $2^6 - 1 = 64 - 1 = 63$

### Weighting Structure of a Binary Number.

Random word (1)

$\Rightarrow$  Binary Number is a Weighted Number

$\Rightarrow$  Right Most beat is the LSB (least Significant beat) and has a weight of  $2^0 = 1$

$\Rightarrow$  The weight increases from Right to left by a power of two

$\Rightarrow$  The left most beat is the MSB (most Significant beat) and its weight depend on the size of the binary Number.

$\Rightarrow$  Fractional number can also be represented in binary by placing beats to the right of the binary point.

$\Rightarrow$  Left most beat is the MSB in a binary fractional number and has a weight of  $2^{-1} = \frac{1}{2} = 0.5$

$\Rightarrow$  Fractional weight decreases from left to right by a negative power of two for each beat.

$\Rightarrow$  The weight structure of binary Number is  $2^{n-1}, 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}$

where  $n$  - number of beat in binary

$\Rightarrow$  Thus all the beat to the left of the binary point that have +ve power of two.

$\Rightarrow$  All the beat to the right of the binary point have weight that are negative power of two or fractional weight.

2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>
256	128	64	32	16	8	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
of large	numbers								0.5	0.25	0.125	0.0625	0.03125	0.015625

## Binary to decimal conversion

2. Convert the binary Whole Number 11011010 to decimal.

$$(110110)_2 \rightarrow \text{Binary representation}$$

$6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0$

$2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2^0$

$1x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$

$64 + 32 + 0 + 8 + 4 + 0 + 1 = \underline{\underline{(109)_{10}}}$

$$(1101101)_2 \stackrel{S(X)}{\rightarrow} (109)_{10} \stackrel{S(X^2) + S(X)}{\rightarrow} 1 + 8 = 9$$

$$0.011 = \frac{S}{100} + \frac{X}{1000} + \frac{I}{10000} + \frac{K}{100000} = \frac{4}{100} + \frac{8}{1000} + \frac{1}{10000} + \frac{6}{100000} = 41.86\% \text{ to decimal}$$

(1) Convert the fractional binary Number 0.1011 to decimal.

$$(2)^{1001} \cdot (101111.1101)_2 = 1001 \cdot 10$$

$$0.1011 = \underline{2} \\ \begin{array}{r} 0 \ 1 \ 2 \ 3 \ 4 \\ 2 \ 2 \ 2 \ 2 \end{array}$$

$$\begin{aligned}
 & 0x^6 + 1x^1 + 2x^2 + 1x^3 + 1x^4 \\
 & 0x^0 + 0.5 + 0 + 0.125 + 0.625 \\
 & = \underline{\underline{0.250}} \quad = \underline{\underline{0.6875}}
 \end{aligned}$$

$$= \frac{1.250}{\underline{\underline{}} \quad \underline{\underline{}}} = 0.6875$$

$$0.1011 = \pi(1001)$$

$$\begin{array}{r} 0.0625 \\ 0.125 \\ \hline 0.15 \\ \hline 0.6875 \end{array}$$

(2)  $(101111, 1101)_2$

$$101111 \cdot 1101$$

$$262432^3 2^4 2^0 \quad \begin{matrix} -1 & -2 \\ 2 & 2 \end{matrix} \quad \begin{matrix} -3 & -4 \\ 2 & 2 \end{matrix}$$

$$1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2} + 0x2^{-3} + 1x2^{-4}$$

$$32 + 0 + 8 + 4 + 2 + 1 + 0.5 + 0.25 + 0.125 + 0.0625$$

$$= 47.8125$$

(100 11)

$$\begin{array}{r}
 & 0.0625 \\
 & 0.125 \\
 & 0.25 \\
 & 0.5 \\
 & \hline
 & 6.9375
 \end{array}$$

## Decimal to binary conversion

(1) Sum of weight method.

(2) Repeated division by two.

(3) Sum of weight method

$$= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6$$

$$= 1 + 0 + 4 + 0 + 16 + 0 + 64 = 85$$

Convert the following decimal number to binary

$$(1) 9 = 8 + 1 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1001_2$$

$$(2) 12 = 8 + 4 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 1100_2$$

$$(3) 25 = 16 + 8 + 1 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 11001_2$$

$$(1) 1001$$

$$(2) 1100$$

$$(3) 11001$$

## II Repeated division by two.

$$(1) \begin{array}{r} 2 | 9 \\ \hline 2 | 4 - 1 \\ \hline 2 | 2 - 0 \\ \hline 1 - 0 \end{array}$$

$$\begin{array}{r} 2 | 82.0 \\ \hline 2 | 41 - 0 \\ \hline 2 | 20 - 0 \\ \hline 10 - 0 \end{array} = (1001)_2$$

$$\begin{array}{r} 2 | 9 \\ \hline 2 | 4 - 1 \\ \hline 2 | 2 - 0 \\ \hline 1 - 0 \end{array}$$

$$(2) \begin{array}{r} 2 | 12 \\ \hline 2 | 6 - 0 \\ \hline 2 | 3 - 0 \\ \hline 1 - 1 \end{array}$$

$$\begin{array}{r} 2 | 25 \\ \hline 2 | 12 - 1 \\ \hline 2 | 6 - 0 \\ \hline 2 | 3 - 0 \\ \hline 1 - 1 \end{array} = (1011.111101)_2$$

$$= 2560.0 + 2510 + 280 + 2.0 + 1 + 2 + 4 + 8 + 16 + 32$$

$$(3) \begin{array}{r} 2 | 25 \\ \hline 2 | 12 - 1 \\ \hline 2 | 6 - 0 \\ \hline 2 | 3 - 0 \\ \hline 1 - 1 \end{array}$$

$$\begin{array}{r} 2 | 18.0 \\ \hline 2 | 9 - 0 \\ \hline 2 | 4 - 0 \\ \hline 2 | 2 - 0 \\ \hline 1 - 1 \end{array} = (11001)_2$$

converting decimal fraction to binary

Two method.

## (I) Sum of weight method

(II) Repeated multiplication by two. Last two

(1) 0.625

$$\left( \begin{array}{cccc} 0.5 & 0.25 & 0.125 & 0.0625 \\ 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \end{array} \right)$$

$$0.5 + 0.125 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4}$$

$$= 0.\underline{1}010$$

(8) Repeated multiplication by two

0.4285

0.4285 x 2

0.8570 x 2 = 0

1.7140x2 = 1

$$1.428 \times 2, -1$$

1.8560 x 2 - 0

1712082-9

~~1 = 8x + 5x + 2x~~

21/1 2013 ⑧

• 100 •

## Octal Numbers

- ⇒ Octal Number System uses the digit  $0, 1, 2, 3, 4, 5, 6, 7$  (not 8)
  - ⇒ So the base or radix of the system is  $8$
  - ⇒ and LSB has a weight of  $8^0$  (not 1)

## Decimal to octal conversion.

Q1:- (1) convert  $(444.456)_{10}$  to an octal No. base 8.

## Octal to decimal conversion

(1) Convert the octal no.  $\underline{\underline{237}}_8$  into decimal.

$$\begin{array}{r}
 11 \quad (237)_8 \\
 \downarrow \quad \downarrow \quad \downarrow \\
 8^2 \quad 8^1 \quad 8^0 \\
 \hline
 8 \times 2 + 8 \times 3 + 7 \times 8 = 128 + 24 + 7 \\
 = 159 = (159)_{10}
 \end{array}$$

(b)  $(120)_8$

3 bits/4 bits/5 bits

$$1 \times 8^2 + 2 \times 8^1 + 0 \times 8^0 = 64 + 16 + 0 = 80$$

~~120 = 80~~  $\underline{(80)_{10}}$  3 bits/4 bits/5 bits

Octal to binary conversion

For obtaining the binary equivalent of an octal number each significant digit in the given number is replaced by a 3-bit binary equivalent.

e.g.:  $(376)_8 \rightarrow \underline{(011111110)_2}$

$$(3) \rightarrow 011$$

$$(7) \rightarrow 111$$

$$(6) \rightarrow 110$$

Binary to octal

For converting binary to octal starting from LSB each group of 3 bits is replaced by its decimal equivalent.

$(011) (111) (110)$

3 7 6

Q: (1)  $(\underline{00} \underline{110} \underline{10} \underline{01})_2$  = 48EA

$(010)(011)(010)(101)$   
4 2 1 = 2325

$$= 8 \times 8 + 3 \times 4 + 2 \times 1$$

$$= 64 + 12 + 2 =$$

$$= 78$$

$$\underline{\underline{78}}$$

## Hexadecimal Number

Q(021)

(d)

- ⇒ Has a radix of  $16^0 + 16^1 + 16^2 + \dots = 8 \times 0 + 8 \times 1 + 8 \times 2 + 8 \times 3 + \dots$   $\text{co } 11 \text{ to } 15$
- ⇒ uses 16 symbols namely  $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \text{ and } \text{a, b, c, d, e, f}$
- The symbols a, b, c, d, e and f represents the decimal values 10, 11, 12, 13, 14, and 15
- LSB has a weight of  $16^0 = 1$
- i.e., higher significant position are given in the ascending power of 16.

## Decimal-to hexadecimal conversion

Q:- (1)  $(115)_{10}$

$110 \leftarrow (\text{E})$

(2)  $(235)_{10}$

$111 \leftarrow (\text{F})$

(1)  $16 \mid 115$

$011 \leftarrow (\text{3})$

~~16~~  $\underline{67} - 3$

(2)  $16 \mid 235$

$140 \leftarrow (\text{E})$

(011) (111) (110)

## Hexadecimal to Decimal conversion.

~~A3B1~~  $A3B1 - (A3B)_{16} (101100)_2 (1100)$

$(A3B)_{16} \quad (101)(010)(101)(010)$

$A \times 16^2 + 3 \times 16^1 + B \times 16^0$

$= 10 \times 16^2 + 3 \times 16 + 11 \times 16^0$

$= 2560 + 48 + 11$

$= (2619)_{10}$

Ex-5.3

Ques. Convert  $(21)(B2F)_{16}$  into binary & add to convert  $(7B5)_{16}$

$$11 \times 16^2 + 2 \times 16 + 15 \times 16 = 2816 + 32 + 15 = (2863)_{10}$$

Ans.  $\underline{\underline{21}} + \underline{\underline{7B5}} \quad (1)$

### Hexadecimal to binary

8421

$$\begin{aligned} \text{Ques. } & 21P = 10^P = 01010110 = 010 + 0110 = 0101 \\ & (2D5)_{16} \\ \text{Ans. } & 01E = 01110010 = 011 + 010 = 0010 \\ & \underline{\underline{2 - 0010}} \\ & D = 1101 \quad \text{Ans.} \\ & S = 0101 \end{aligned}$$

$$\begin{aligned} \text{Ques. } & 01A = 01(001011010101)_2 = 010 + 010 = 0100 \\ & \underline{\underline{0100}} \quad (\Sigma) \end{aligned}$$

### Binary to Hexadecimal

$$\begin{aligned} \text{Ques. } & 011110110101 = (0111 \quad 1011 \quad 0101)_2 \\ & = 01 + 01 = 010 + 010 = 0100 \\ & (7B5)_{16} \\ & \underline{\underline{0100}} \quad \text{Ans.} \quad 8421. (\Sigma) \end{aligned}$$

$$1 + 0101 = 0101 + 0101 = 01A + 01C$$

Ques. Hexadecimal addition and Subtraction.

### Addition rules

Rule-1

In any given column of an addition problem think of two hexadecimal digits in terms of their decimal value.

e.g.,  $(5)_{16}$  is equivalent to  $01010101_2$  (1)

whose addition can take place in  $(2)$

$(C)_{16} = (12)_{10}$  whose sum can take place in  $(8)$

Rule-2

If the sum of the two digits is  $(15)_{10}$  or less. bring down the corresponding hexadecimal digits.

### Rule-3

Rule-3  
If the sum of the 2 digit is greater than ~~10~~(1s), bring down the amount of the sum that exceeds ~~10~~(1s) and carry a one to the next column.

$$21 + 28 + 18 = 21 \times 2 + 21 \times 4 + 21 \times 1$$

eg + -

$$(1) \quad (23)_{16} + (16)_{16}$$

1543

$$(23)_{16} + (16)_{16} \quad \begin{array}{r} 23 \\ 16 \\ \hline 16 \end{array}$$

period of 10 months

$$3_{16} + 6_{16} = 9_{10} + 6_{10} = 9_{10} = 9_{16}$$

$$2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16} \quad \text{E(39)}$$

$$(39)_{16} = 1011 - 1 \\ 1010 = 2$$

$$(2) \quad \begin{array}{r} 58 \\ \times 16 \\ \hline 22 \\ + 58 \\ \hline \end{array}$$

$$8_{16} + 2 \frac{(1010)}{16} = 8_{10} + 2 \frac{(100)}{10} = A_{10}$$

$$5_{16} + 2_{16} \rightarrow 10_{10} = 0_{16}$$

$$100 \cdot 5_{16} + 12_{16} = 5_{10} + 2_{10} \quad \text{Ex: } 7_{10} = 7_{16} =$$

$$(3) \quad \begin{array}{c} \text{!} \\ \text{DF}_{16} \\ \hline AC_{16} \end{array}$$

$$\rightarrow F_{16} + C_{16} = \frac{(10)(11)(12)(13)}{10} = (27)_{10} - 16 = (11)_{10} = 8_{16}$$

(૨૮)

FIG. C1C

$$B_{10} + A_{10} = 13_{10} + 10_{10} = 23_{10}$$

$$\text{Resultant vector } \vec{R} = \vec{A} + \vec{B} = (11)_{10} \quad \boxed{= (11)_{10}}$$

二〇一〇

~~8 + 1 = 8 with carry!~~

$$= (T_{8B})_{16}$$

## Assignment-1

Method to find the error in approximating the definite integral.

Method to find two's complement of

Method to find cross complement of antibody

- (1) Representation of negative numbers in Binary
  - (2) Error detection and correction code.
  - (3) parity and hamming codes.

(4) universal gate

(4) Universal gate  
also known as (cl) 2<sup>nd</sup> stage of OR-NAND form

It's like I'm reborn after this program so I'll

MSB is one for -ve binary numbers

MSB is zero for +ve binary numbers

## 1's complement

⇒ The 1's complement of a binary Number is obtained when each beat of a binary number is Substracted from one.

ie, the 1's complement of zero is  $= 1-0 = 1$

ie, 1's complement of 1 is  $= 1-1 = 0$

ie, 1's complement of a binary number is obtained by replacing "0 by 1" and "1 by 0".

ie, eg:- 1's complement of 00001000 is

$10111111$  (10)

## 2's complement

⇒ The 2's complement of a binary number is obtained by adding 1's complement of a binary number.

In these representation

MSB is 0 for +ve number.

MSB is 1 for -ve number.

eg:- 2's complement of 8 is,

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

$$\begin{array}{r} 00001000 \\ 11110111 \\ \hline 11111000 \end{array}$$

$$+1+1=0$$

$$\begin{array}{r} 4 \rightarrow 0100 \\ 4 = 1011 \\ 4'' = 1100 \rightarrow -4 \end{array}$$

$$\begin{array}{r} 9 = 01001 \rightarrow 01101 \\ 01001 \\ 01111 \\ \hline -9 \end{array}$$

$$\begin{array}{r} 1+1=0 \text{ with carry } 1 \\ 1+0=1 \\ 0+1=1 \\ 0+0=0 \end{array}$$

$$\begin{array}{r} 1010 \\ 1001 \\ \hline 1001 \end{array}$$

03-08-18  
Friday

## Binary arithmetic operation

Binary addition & subtraction

### (1) Binary addition.

$$1 = 0+1 =$$

$$0+0 = 0$$

$$0+1 = 1$$

$$1+0 = 1$$

$$1+1 = 10$$

carry

### Binary subtraction

$$1 = 1 - 0 =$$

$$0-0 = 0$$

$$1-1 = 0$$

$$1-0 = 1$$

$$0-1 = 1$$

with a borrow of 1

Add

(1)

$$(11)_2 + 0.11 = 1.1$$

$$(01)_2$$

$$\begin{array}{r} 0 \\ 100 \\ \hline 100 \end{array}$$

$$(1) 11 - 01$$

$$\begin{array}{r} 11 \\ - 01 \\ \hline 10 \end{array}$$

(2)

$$(11)_2 + (11)_2$$

$$\begin{array}{r} 10 + 1 \\ + 0 \\ \hline 110 \end{array}$$

$$(2) 11 - 10$$

$$\begin{array}{r} 11 \\ - 10 \\ \hline 1 \end{array}$$

(3)

$$100 + 10$$

$$\begin{array}{r} 0 = 100 + \\ 10 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 100 \\ + 00010000 \\ + 11011111 \\ \hline 00011111 \end{array}$$

$$(3) 1101 - 001$$

$$\begin{array}{r} 1101 \\ - 001 \\ \hline 010 \end{array}$$

(4)

$$111 + 11$$

$$\begin{array}{r} 111 + \\ 11 \\ \hline 1010 \end{array}$$

$$\begin{array}{l} 0 = 111 \\ 1 = 011 \\ 1 = 1+0 \\ 0 = 010 \end{array}$$

$$\begin{array}{r} 0011 \\ + 0110 \\ \hline 1110 \end{array}$$

## Binary Multiplication

(a) & is short and easy, (b) is binary form, (c) is long or difficult.

$0 \times 0 = 0$	$0 \times 1 = 0$	if 1's digit is 0 then product will be 0
$1 \times 1 = 1$		
$1 \times 0 = 0$		if 1's digit is 1 then product will be 1

is also known as short multiplication. It is to develop a quick method.

Method (c) is used for all cases of multiplication.

$$(1) 1101 \times 111 = (1001)_2 \quad (2) 111 \times 101 = (100011)_2$$

Method (a) is simple and easy to understand.

$$\begin{array}{r} 1101 \\ \times 111 \\ \hline 1101 \\ 0110 \\ \hline 100011 \end{array}$$

Method (b) is just like decimal multiplication.

$$\begin{array}{r} 111 \\ \times 101 \\ \hline 111 \\ 000 \\ \hline 100011 \end{array}$$

01001000

## Binary division

Method (a) is simple and easy to understand.

$$(1) 110 \div 11 = 10$$

$$\begin{array}{r} 110 \\ \hline 11 \end{array}$$

$$\begin{array}{r} 000 \\ \hline 000 \end{array}$$

00011001 - 81

$$(2) 110 \div 10 = 11$$

$$\begin{array}{r} 1010 \\ \hline 110 \\ 10 \\ \hline 10 \\ 010 \\ 010 \\ \hline 000 \end{array}$$

$$\begin{array}{r} 10 \\ \hline 110 \\ 10 \\ \hline 10 \\ 010 \\ 010 \\ \hline 000 \end{array}$$

10101100 - 28

## Binary codes

(1) BCD codes (Binary coded decimal)

\* In digital system it is possible to represent decimal number simply by encoding each digit in binary form.

\* This is called binary coded decimal representation.

\* The BCD uses the binary number system to specify the decimal number 0 to 9.

\* It has 4 bits, the weights are assigned according to the position occupied by these digits.

- \* The weights of the first position is  $2^0(1)$  + 1 unit weight
- \* The second  $2^1(2)$ , The third  $2^2(4)$ , and the forth  $2^3(8)$ .
- \* Then Reading from left to right the weight are 8-4-2-1  
8-4-2-1 and hence it is called 8421 code.
- \* The binary equivalent of 7 is  $(111)_2$  but the same number is represent in BCD in 4 bit form  $(0111)_2$   
so also the numbers from 10 to 9 are represented in the same way as in the binary systems. but after 9 the representations in BCD are different.
- \* For eg:- The decimal no 12 in the binary systems is  $(1100)_2$  but the same no is represented BCD as 0001 0010

2. Convert each of the following decimal No to BCD.

$$(1) \frac{35}{11} \quad (2) \frac{98}{11} \quad (3) \frac{2469}{11}$$

$$\begin{array}{r} (1) \frac{35}{11} \\ \underline{35} \\ 00110101 \end{array} \quad \begin{array}{r} (2) \frac{98}{11} \\ \underline{98} \\ 10011000 \end{array}$$

$$(3) \frac{2469}{11} = \underline{\underline{001000110}} = \underline{\underline{0010010001101001}}$$

Q. To convert each of the following BCD code to Decimal.

$$(1) \frac{10000110}{86} = (86)$$

$$(2) \frac{001101010001}{351} = (351)$$

## BCD addition.

- \* BCD is a numerical code and can be used in arithmetic operation.
- \* Addition is the most important operation, because other three operations are accomplished by the use of addition.

### Steps

- (1) Add the two BCD NO using the rules of binary addition.
  - (2) If a 4 beat sum is equal to or less than 9 it is a valid BCD code.
  - (3) If a 4 beat sum is greater than 9 or if a carry out of the 4 beat group is generate it is an invalid result.
  - 4 Add 6 (0110) to the 4 beat sum in order to keep the 6 invalid states and return the code to 8421
- If a carry results when 6 is added Simply add the carry to the next 4 beat group.

### Q Add the following BCD numbers

$$(1) \quad 0011 + 0100$$

$$\begin{array}{r} 0011 \\ + 0100 \\ \hline 0111 \end{array}$$

$$(2) \quad (00100011) +$$

$$\begin{array}{r} (00000101) \\ \hline (00111000) \end{array}$$

$$23$$

$$15$$

$$38$$

$$+ 10001$$

$$8421$$

$$(3) \quad 1001 + (9)$$

$$0100 \quad (4)$$

$$\hline 1100 \quad (13) > 9 \text{ invalid BCD}$$

add 6 to the number to make valid BCD.

$$\begin{array}{r} 101101 + (13) \\ 0110 \quad (6) \\ \hline (00010011) \end{array}$$

$$13$$

$$\text{valid BCD.}$$

(4) 0011

$$(4) \quad 0001 \uparrow \quad 0110 + 16$$

$$\underline{0001} \quad 0101$$

$$1011 \quad 8+21$$

invalid BCD

0001 0110

0001 0110

1011 +

0110  
10001

10001

0001 +

0001 +

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

0011

&lt;

## Gray Code

- \* The Gray Code is unweighted and is not an arithmetic code. i.e., there is no specific weight assigned to the best position.
  - \* Important feature of the Gray Code is that exactly one bit changes from one code to the next in sequence.

## Binary to Gray Code Conversion

- \* The MSB in the Gray Code is the same as the corresponding MSB in the binary number.
  - \* Going from left to right, add each adjacent pair of binary code bits to get the next gray code bit. This is called a carry.   
 discard

The binary No is 10110

प्राचीन रूपमें  $(10110)_2$ -Binary नंबर का उदाहरण है।

$$\begin{array}{r} 1207171+0 \\ \downarrow \\ 11101 \end{array} \quad \left. \begin{array}{l} | \\ 1+0 \text{ carry off} \\ 0+1 \\ 1+1 \text{ carry eliminate} \\ 1+0 \end{array} \right\} \text{Graey}$$

at a time  $(1011)_2$  - Binary

$(1 \cdot 1 \cdot 100)$  - Gray point  $\frac{10}{100}$

## Gray to binary conversion

step 1: ~~skip first~~

- \* The MSB in the binary code is the same as the corresponding bit in the Gray code.
- \* Add each binary code generated to the Gray code bit in the next adjacent position, ~~discard rightmost~~ this called ~~carry~~ ~~sum~~.

eg: (1)  $110_2$  Gray  
 $(10010)_2$  binary  
 $\downarrow + \downarrow + \downarrow + \downarrow$   
 $11011$

(2)  $(1010)_2$  Gray  
 $\downarrow + \downarrow + \downarrow + \downarrow$   
 $10101$

## Excess-three Code

- \* As the name indicate the excess 3 represent a decimal no in binary form, as a number greater than three.

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

- \* An excess-3 code is obtained by adding '3' to a decimal number.

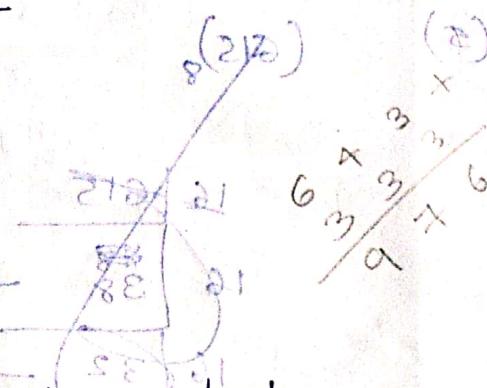
eg:- To encode the decimal no. '6' in to a excess - 3 code. we must first add '3' in order to obtain '9'. The '9' is then encoded in its 4 bit binary code  $(1001)_2$

The Excess-3 code is the self complementing code and helps in performing Subtraction operation in digital computers.

and the excess-3 code is also known as reflective code.

eg:- convert  $(643)_{10}$  into its excess-3 code.

$$\begin{array}{r} \text{Decimal no} & 6 & 4 & 3 \\ \text{adding } 3 & + & + & + \\ 3 & 3 & 3 & 3 \\ \hline 213 & 21 & (9 & 7 & 6) \\ 8+ & (1001 & 0111 & 0110) \end{array}$$



Ques

- (1) Convert  $(111101100)_2$  to octal equivalent
- (2) convert  $(1101100010011011)_2$  to hexadecimal equivalent
- (3) convert  $3FDH$  to binary.
- (4) convert  $5A9.B4$  to binary.
- (5) convert  $(8615)_8$  to its hexadecimal number.
- (6) convert  $(FAB25B)_2$  to its octal equivalent.
- (7) Determine the value of base  $x$ .

$$\begin{array}{l} \text{if } (1) \quad (193)_{10} = (245)_8 \\ \text{if } (2) \quad (225)_{10} = (341)_8 \\ \text{if } (3) \quad (211)_{10} = (152)_8 \end{array}$$

Ans:

$$(1) \quad \begin{array}{r} (111101100)_2 \\ (754)_{10} \\ 8421. \end{array}$$

bin octal

$$(2) \quad (1101100010011011)_2$$

10 11 12 13 14 15 16  
A B C D E F H

$$13 \quad 8 \quad 9 \quad 10 \quad 11 =$$

$$(D89B)_{16}$$

01(802)

$$8421.$$

0.090

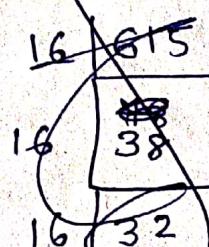
$$(3) \quad \begin{array}{r} (3FD)_{16} \\ (0011111101)_2 \\ 8811 \end{array}$$

$$(4) \quad \begin{array}{r} (5A9.B4)_{16} \\ (010110101001.00110100)_2 \end{array}$$

$$z^3 z^4.$$

$$\underline{0011}$$

(5)  $(615)_8$



$$\begin{array}{r} \text{E} \\ + \\ \text{E} \end{array} \quad \begin{array}{r} \text{E} \\ + \\ \text{E} \end{array} \quad \begin{array}{r} \text{E} \\ + \\ \text{E} \end{array} = \underline{\underline{307}}$$

$$16 \overline{)615}$$

$$\begin{array}{r} 397 \\ 48 \\ 135 \\ 128 \end{array}$$

(5)  $(615)_8$

$$\downarrow \downarrow$$

$$6 \times 8^2 + 1 \times 8^1 + 5 \times 8^0 = 384 + 8 + 5$$

(6)  $(25B)_{16}$

$$\begin{array}{r} (397)_{10} \\ (148) = (243)_{16} \\ (181) = 16 \cdot (14)_{16} \end{array}$$

↓

(6)  $(25B)_{16}$

~~8+8~~

$$8 \times 8 + 2 \times 16^2 + 5 \times 16^1 + 11 \times 16^0 = 11512 + 80 + 11041$$

1	1	1	1	1	1	0	1
4	7	3	4	5	8	9	

$$= (603)_{10}$$

$(603)_{10}$

1133

$$8 \overline{)603}$$

$$8 \overline{)758} - 3$$

$$8 \overline{)925} - 3$$

$$1 - 1$$

$(A.A.PA2)$

$(0010 \ 1001 \ 1001 \ 0101 \ 1010)$

$$(7) \text{ (g)} (193)_2 = (G23)_{10} \quad \text{8 digits A max 1000}$$

$$1x^2 + 9x + 3x^0 = 6x^2 + 2x^8 + 5x^8$$

$$x^2 + 9x + 3 = 384 + 16 + 3$$

~~Re(74) = 402~~ ~~new antilog to 220 Hz SW~~

Now let's solve the quadratic equation  $x^2 + 9x - 40 = 0$  by factoring.

$x = 16, -25$

(193)<sub>16</sub> - *Proposed revision of*

12. ~~कृष्ण~~ राम के द्वारा अवधारणा की गई थी।

$$Q \quad (2) \quad (225)_x = (341)_8$$

$$2x^2 + 2x + 5x^0 = 3x^2 + 4x^1 + 1x^0$$

$$2x^2 + 2n + 5 = 192 + 32 + 8$$

$$2x^2 + 2x = 225 - 5$$

$$2n^2 + 2n = 22$$

$$2n^2 + 2n - 220 = 0$$

$$x^2 + x - 10 = 0$$

$$n = 10_{\text{base}11} \quad \text{to base 10} \quad 10000000000_2$$

(225)<sub>10</sub>

ozi i fersor sligoro' bau'pi ei oras fe fasosilgoras odi? <sup>225/16</sup>

$$(3) \quad (211)_m \stackrel{?}{=} (152)_8$$

$$2x^2 + 1x^1 + 1x^0 = 1 \times 8 + 5 \times 8 + 2 \times 8$$

$$2x^2 + x + 1 = 64 + 40 + 2$$

$$2n^2 + 2n + 1 = \cancel{2n^2} + 106 \text{ (divide by } 2\text{)} \Rightarrow n = 21 \times 21$$

$$2n^2 + n - 105 = 0$$

$$(211)_- = (152)_s$$

(211)<sub>7</sub> = (152)<sub>8</sub>

~~correct A~~ correct B ~~incorrect C~~ incorrect D

କେବଳ ପାଦମର୍ଗ ଏବଂ ପାଦମର୍ଗରେ ଯାଏନ୍ତି କିମ୍ବା କିମ୍ବା କିମ୍ବା

~~1~~ ~~1~~ ~~1~~ ~~1~~ ~~1~~

## Boolean Algebra

$$g(x_2 + g(x_1 + g(x_3 + g(x_4 + g(x_5 + g(x_6 + g(x_7 + g(x_8 + g(x_9 + g(x_{10})))))))) = g(g(g(g(g(g(g(g(g(g(g(x_1 + g(x_2 + g(x_3 + g(x_4 + g(x_5 + g(x_6 + g(x_7 + g(x_8 + g(x_9 + g(x_{10})))))))))))))))))))$$

- \* The basic mathematics needed for the study of logic design of digital system is Boolean Algebra.
- \* We will use a Boolean variable such as  $x$  or  $y$  to represent the input or output of a switching network.
- \* The symbols zero and one are used to represent these two values.
- \* Thus  $x$  is a Boolean variable either  $x$  is 0 or  $x=1$ .
- \* ~~True~~

### Basic Operations

$$g(x_1 + g(x_2 + g(x_3 + g(x_4 + g(x_5 + g(x_6 + g(x_7 + g(x_8 + g(x_9 + g(x_{10})))))))))) = g(g(g(g(g(g(g(g(g(g(g(x_1 + g(x_2 + g(x_3 + g(x_4 + g(x_5 + g(x_6 + g(x_7 + g(x_8 + g(x_9 + g(x_{10})))))))))))))))))))$$

The basic operations are  $SPI = 2 + 10x_1 + 5x_2$

\* AND  $2 - 2x_1 - 2x_2 = 0x_1 + 0x_2$

\* OR

\* complement / invert / inverse

$$0 = 0x_1 - 0x_2 + 0x_3$$

### (A) AND

### (I) complement / invert / inverse

- \* The complement of zero is 1 and complement 1 is 0.

$$\text{Complement of } 0 = \bar{0} = 1 = 0' \quad (\text{E})$$

$$\text{Complement of } 1 = \bar{1} = 0$$

$$g(x_1 + g(x_2 + g(x_3 + g(x_4 + g(x_5 + g(x_6 + g(x_7 + g(x_8 + g(x_9 + g(x_{10})))))))))) = g(g(g(g(g(g(g(g(g(g(g(x_1 + g(x_2 + g(x_3 + g(x_4 + g(x_5 + g(x_6 + g(x_7 + g(x_8 + g(x_9 + g(x_{10})))))))))))))))))))$$

where '1' or ' $\bar{x}$ ' denotes the complementation.

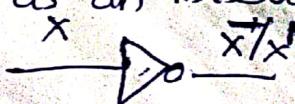
- \* If  $x$  is a switching variable

$$x' = 1 \text{ if } x = 0 \quad 0 + x = 1$$

$$x' = 0 \text{ if } x = 1$$

- \* The alternate name of complementation is inversion.

- \* The electronic circuit which forms inverse of  $x$  is referred to as an inverter. Symbolically we can represent it as



circle at the output indicate inversion.

\* complementation sometimes refers to as NOT operation.

2.1.2 AND Gate

### (2) AND operation

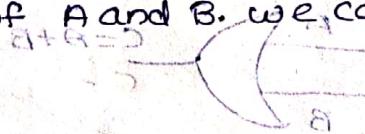
following is truth

AND operation can be defined as follows.

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

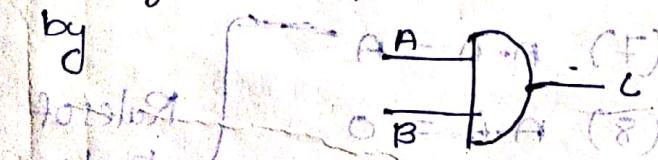
where the dot represent AND operation.

\* If we write the Boolean expression.  $A \cdot B = C$ , then given the values of A and B, we can determine C from the following table



\* Note that logical 1 is word order if  $C=1$  if and only if A and B are both 1, hence the name AND operation.

\* The logic Gate which performs the AND operation is represented by



$$A = 0 + A \quad (1)$$

$$I = 1 \cdot 1 \quad (2)$$

$$0 = 0 \cdot A \quad (3)$$

\* The AND operation is also referred to as logic multiplication or Boolean multiplication. The product term is zero

when 1 or more literals are zero

$$A + A = (A + A)(A + A) \quad (\text{variables})^2$$

$$1 = 1 + 1 \quad (4)$$

### (3) OR operation

\* The OR operation can be defined as follows.

$$A = 0 + A \quad (1)$$

$$1 = 0 + 1$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 1 + 0$$

$$1 = 1 + 0$$

$$0 = 0 + 1$$

$$1 = 0 + 0$$

$$0 = 1 + 1$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

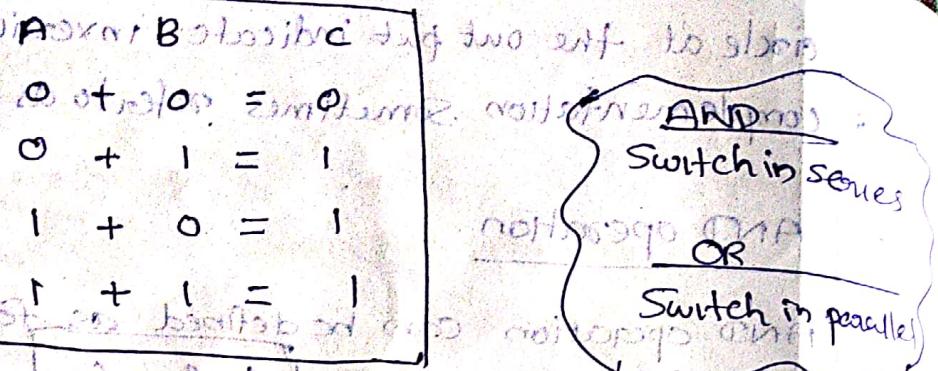
$$0 = 0 + 0$$

$$1 = 1 + 1$$

$$0 = 0 + 1$$

$$1 = 1 + 0$$

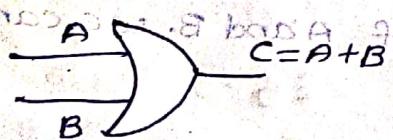
$$0 = 0 + 0$$



- \* Thus, where '+' denotes OR operation.
- If we write  $C = A + B$  note that  
 $C = 1$  if and only if, A or B (or both) is one.  
 hence the name OR operation.

\* The logic gate which performs the OR operation is

represented by



\* OR operation is also known as logical addition.

### Basic theorems

$$(1) A + 0 = A$$

$$(7) A \cdot A = A$$

$$(2) A + 1 = 1$$

$$(8) A \cdot \bar{A} = 0$$

$$(3) A \cdot 0 = 0$$

$$(9) A \cdot \bar{A} = A$$

$$(4) A \cdot 1 = A$$

$$(10) A + A\bar{B} = A$$

$$(5) A + A = A$$

$$(11) A + \bar{A}B = A + B$$

$$(6) A + \bar{A} = 1$$

$$(12) (A + B)(A + C) = A + BC$$

Rules of  
Boolean  
Algebra

$$(1) A + 0 = A$$

$$(2) A + 1 = 1$$

if  $A = 1$  then  $A + B = 1$

$$1 + 0 = 1$$

$$\text{if } B = 0$$

$$0 + 0 = 0$$

$$\therefore A + 0 = A$$

$$(2) A+1 = 1 \quad \text{additive identity}$$

$$\text{eg:- } A+1 = 1 \quad 1+1 = 1 \quad A=1 \quad 1+1=1$$

$$A=0 \quad 0+1=1$$

$$A=0 \quad 0+0=0 \quad (1)$$

$$(G) A+A = \emptyset \quad (2)$$

$$A=0 \quad 0+\emptyset = \emptyset \quad (3)$$

$$A=1 \quad 1+\emptyset = \emptyset \quad (4)$$

$$(3) A \cdot 0 = 0$$

$$A=0 \quad 0 \cdot 0 = 0$$

$$A=1 \quad 1 \cdot 0 = 0$$

$$A=0 \quad 0 \cdot 1 = \emptyset \quad (5)$$

$$A=1 \quad 1 \cdot 1 = \emptyset \quad (6)$$

$$(4) A \cdot 1 = A$$

$$A=0 \quad 0 \cdot 1 = 0$$

$$A=1 \quad 1 \cdot 1 = 1 \quad A=1 \quad 1 \cdot 1 = 1$$

$$A=0 \quad 0 \cdot 1 = 0$$

$$A \cdot 1 = 1 \cdot A$$

$$(7) A \cdot A = A$$

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

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

$$(8) A \cdot \bar{A} = 0$$

$$A=0 \quad 0 \cdot \bar{0} = 0$$

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

$$(9) \bar{A} = A$$

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

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

$$(10) A + AB = A$$

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

$$(11) A + \bar{A}B = B$$

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

$$\text{and } \bar{A}A = \emptyset \quad (2)$$

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

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

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

$$(11) A + \bar{A}B + \bar{A}B = A + B(A + \bar{A}) = A + B(1+C) = A + BC$$

$$= B(1+C) + BC = B + BC$$

$$= B + BC = A + BC$$

$$A + BC = A + BC$$

$$(12) (A+B)(A+C) = A \cdot A + AC + A \cdot B + BC$$

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

$$= AA + A\bar{C} + A\bar{A}B + B\bar{A}C = A + A\bar{C} + RB + B\bar{A}C$$

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

# Laws of Boolean algebra

(1) commutative law

(2) Associative law

(3) Distributive law.

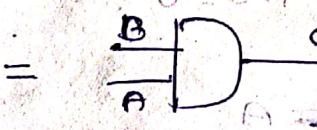
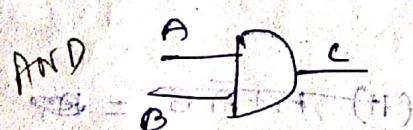
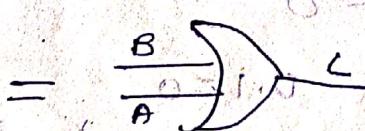
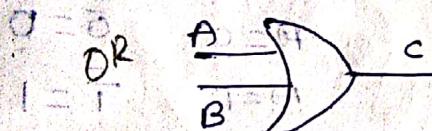
(1) commutative law

commutative law of addition is,

$$A+B = B+A$$

commutative law of multiplication is,

$$A \cdot B = B \cdot A$$

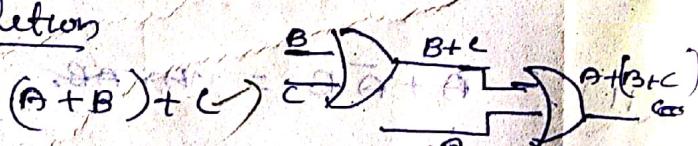


(2) Associative law

addition

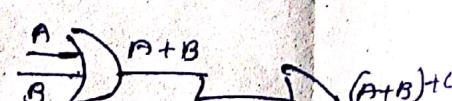
ATB addition

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

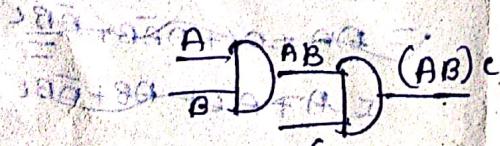
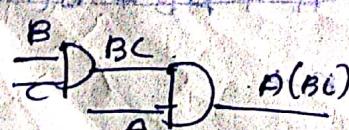


multiplication

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

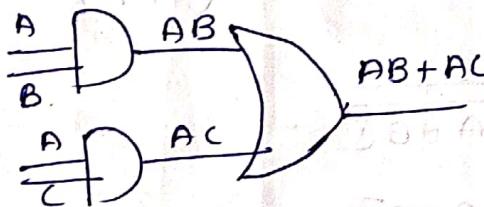
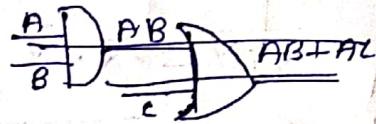
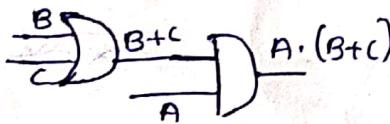


multiplication



### (3) Distributive law

$$A \cdot (B+C) = AB + AC$$



02-09-18  
Monday

### De Morgan's Law Theorem

#### (1) Theorem I

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

The complement of product of variable is equal to the sum of the product of the variable (or complement of two or more variable ANDed) is equivalent to the OR of the complements of the variable.

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

The complement of sum of variable is equal to the product of complements of the variable.

Simplify the following expression using De Morgan's theorem, using Boolean algebra

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

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

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

and by contradiction (8)

$$\overline{A+B\bar{C}} = x \quad \text{and } \overline{D(E+F)} = y \quad (\text{say})$$

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$

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

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

$$\overline{x} = \frac{\overline{A+B\bar{C}}}{\overline{A+B\bar{C}}} \cdot \overline{D(E+F)}$$

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

$$\overline{y} = \overline{D(E+F)}$$

$$= \overline{D} + E$$

Ans-

Let

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

$$\overline{x+x} = \overline{xx}$$

and  $\overline{D(E+F)} = y$

$$= \overline{A+B\bar{C}} \cdot \overline{D(E+F)} \rightarrow (1)$$

using rule  $\overline{\overline{A}} = A$

(1) becomes

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

$$\begin{aligned} \overline{y} &= \overline{\overline{D(E+F)}} \\ &= \overline{D+E+F} \end{aligned}$$

$$\therefore \overline{x+y} = \overline{A+B\bar{C}} \cdot \overline{D+E+F}$$

$$(2) \overline{(A+B+C)D}$$

$$(3) \overline{ABC + DEF}$$

$$(4) \overline{AB + CD + EF}$$

$$(5) \overline{(A+B)} + C$$

$$(6) \overline{(A+B)} + CD$$

$$(7) \overline{(A+B)CD + EF}$$

$$(2) \overline{(A+B+C)D}$$

$$= \overline{AD + BD + CD}$$

$$= \overline{AD} \cdot \overline{BD} \cdot \overline{CD}$$

$$(2) \overline{(A+B+C)D}$$

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

$$(3) \overline{ABC + DEF}$$

$$= \overline{ABC} \cdot \overline{DEF}$$

$$(4) \overline{AB + CD + EF}$$

$$= \overline{AB} \cdot \overline{CD} \cdot \overline{EF}$$

$$(2) \overline{(A+B+C)D}$$

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

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

$$= \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{D}$$

$$(3) \overline{ABC + DEF}$$

$$= \overline{ABC} \cdot \overline{DEF}$$

$$\overline{x+y}$$

$$= \overline{x} \cdot \overline{y}$$

$$= \overline{ABC} \cdot \overline{DEF}$$

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

$$(4) \overline{AB + CD + EF}$$

$$= \overline{AB} \cdot \overline{CD} \cdot \overline{EF}$$

$$\overline{x+y+z}$$

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

$$= \overline{AB} \cdot \overline{CD} \cdot \overline{EF}$$

$$(5) \quad \overline{(A+B)+C}$$

$$(4) \quad \overline{AB + \overline{CD} + EF}$$

$$x = AB \quad y = \overline{CD} \quad z = EF$$

$$\overline{x+y+z} = \overline{x} \cdot \overline{y} \cdot \overline{z}$$

$$= \overline{AB} \cdot \overline{\overline{CD}} \cdot \overline{EF}$$

$$= \overline{AB} \cdot C\bar{D} \cdot \overline{EF}$$

$$= (\overline{A+B}) \cdot (C+D) \cdot (\overline{E+F})$$

$$x = \overline{AB}$$

$$y = C\bar{D}$$

$$z = E\bar{F}$$

$$x \cdot y \cdot z = \overline{x} + \overline{y} + \overline{z}$$

$$(5) \quad \overline{(\overline{A+B})+C}$$

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$

$$x = \overline{A+B} \quad = \quad \overline{A+B} \cdot \overline{C}$$

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

$$(6) \quad \overline{(\overline{A+B})+CD}$$

$$x = \overline{A+B} \quad y = CD$$

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$

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

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

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

$$(7) (A+B)\bar{C}\bar{D} + E\bar{F}$$

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

$$y = E\bar{F}$$

$$z = \bar{F}$$

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

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

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

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

~~difficulties~~  
05-09-18  
Wednesday

Sop and poset Boolean expression and truth table.

Boolean expressions are formed by application of the basic operations to one or more variables or constants.

The simplest expressions consist of a single constant or variable such as zero,  $x, y$  etc.

More complicated expressions are formed by combining two or more other expressions using AND or OR or by complementing other expressions.

e.g.;-  $A\bar{B} + C, \overline{A \cdot C + D} + BE$ , etc.

Truth table (Table of combination)

Specifies the value of a Boolean expression for every possible combination of values of the variables in the expression.

The name truth table comes from a similar table which is used in symbolic logic to list the truth or falsity of a table under all possible conditions.

We can use a truth table to specify output values of a network of logic gates.

In terms of the values of the input logic gate

$$\text{eg: } f = \bar{A} + f = A' + B$$

Truth table

A	A'	B	F
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

### SOP (Sum of product term)

When expression is said to be in sum of product form when all product are the products of a single variable only. These form is the result when an expression is fully multiplied out.

It is easy to recognize a sum of product expression since it consist of a sum of product term.

$$\text{eg: } AB' + CD'E + AC'E'$$

\* If one or more of the product term may consist of a single variable are also considered as in Sop form.

\* The expression

$(A+B)CD + E + F$  is not a Sop form, because  $A+B$  terms enters into a product by is not a single variable.

### Domain of a Boolean expression

The Domain of a general Boolean expression is a set of variable contain in an expression either complimented or uncomplimented form.

eg:- (1)  $\bar{A}B + A\bar{B}C$  is a set of variable A, B, C  
for expression.

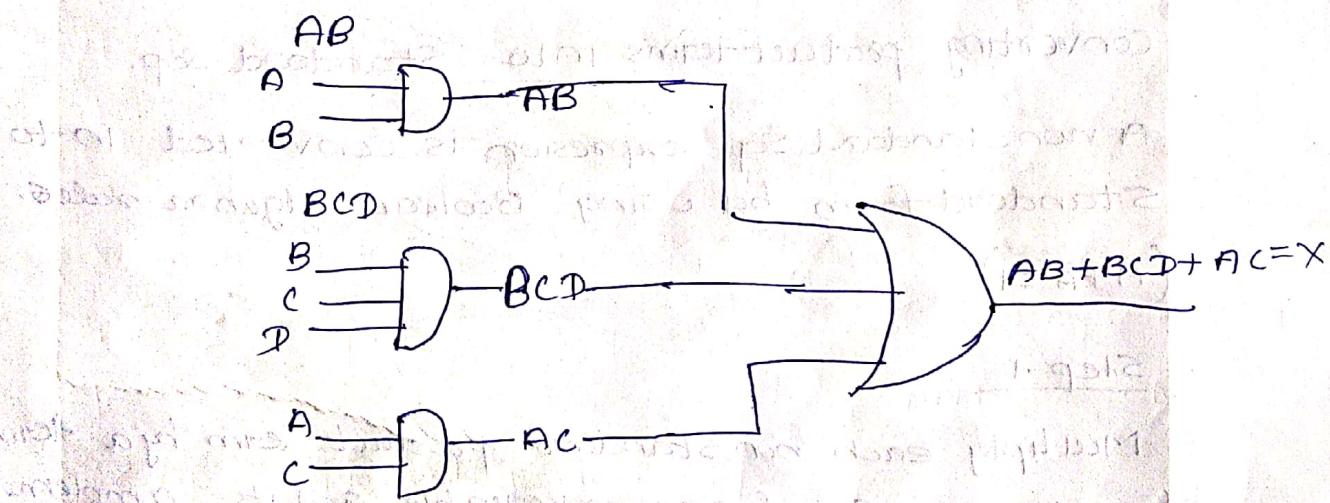
$ABC' + CD'E + B'CD$  is a set of variable, A, B, C, D, E

### Implementation of a Sop expression

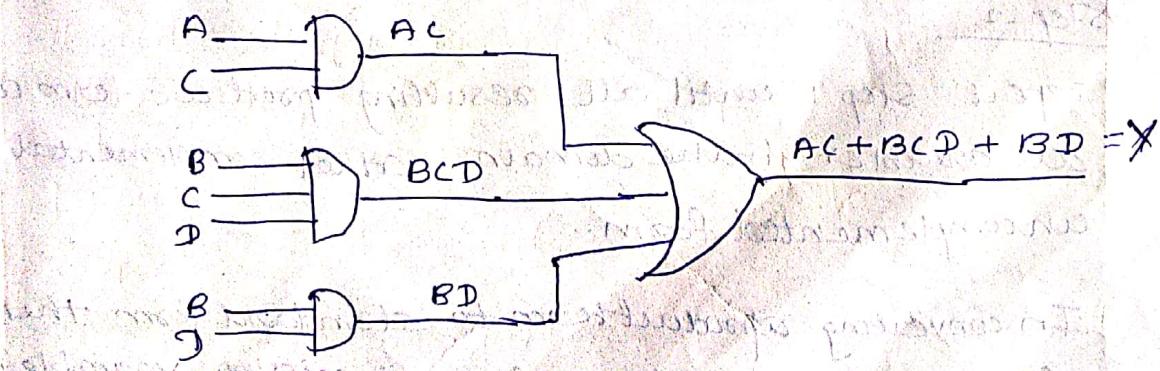
- \* Implementing a sop expression simply require OR all in the output of two or more AND gates.
- \* A product term is produced by an AND operation and OR sum of two or more product term is produced by OR operation.
- \* though the Sop operation is implemented by AND - OR.

in which the output of a number of AND gate is connected to input of a OR gate.

$$\text{eg: (1)} X = AB + BC'D + AC$$



$$(2) AC + BC'D + BD$$



## conversion of a general expression to Sop form

\* Any logic expression can be change in Sop form by applying Boolean algebra techniques.

for

Eg:- The expression  $A(B+C)$  can be converted to Sop form by applying the Distributive law.

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

### Standard Sop form

A Standard Sop expression is one in which all the variables in the domain appear in each product term in the expression.

Eg:-  $ABCD + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD$  is a Standard Sop form

$A\bar{B}D + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD$  is not a Standard Sop form

Converting product terms into Standard Sop.

A non standard Sop expression is converted into Standard form by using Boolean algebra rules.

$$(A+\bar{A})=1$$

#### Step-1

Multiply each non standard product term by a term made up of sum of missing variable and its complement which result two product terms.

#### Step-2

repeat step 1 until all resulting product terms contain all variables in the domain either complemented or uncomplemented form.

In converting a product term to standard form the number of product term is doubled for each missing variable.

Convert the following Boolean expression into standard SOP form

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

Domain = A, B, C, D

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

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

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

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

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

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

convert the following expression.

$w\bar{x}y + \bar{x}y\bar{z} + w\bar{x}\bar{y}$  to standard SOP form.

Domain = ~~x, y, z~~, w, x, y, z.

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

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

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

Ans:

$$w\bar{x}yz + w\bar{x}y\bar{z} + w\bar{x}y\bar{z} + \bar{w}\bar{x}y\bar{z} + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z}$$

pos [product of sum]

cubes or more sum terms are multiplied the resulting expression is a product of sum.

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

Implementation of a pos expression

Implementing a pos expression simply requires AND in the output of two or more OR gates.

## UNIVERSAL GATES

The NAND and NOR gates are called as universal gates because it is possible to implement any Boolean expression with the help of only NAND or only NOR gates. Hence, a user can build any combinational circuit with the help of only NAND gates or NOR gates. This is great advantage because a user will have to make a stock of only NAND or NOR gate ICs.

### NAND gate as a universal gate

In this article, let us illustrate the conversion of each logic function into an expression using only NAND gates.

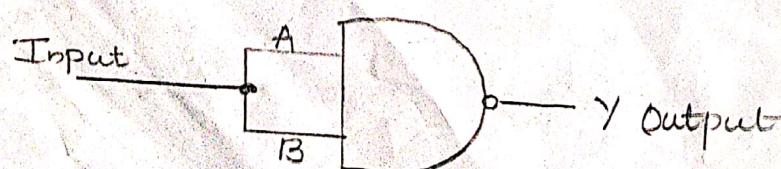
As discussed earlier, a NAND gate is expressed mathematically as under:

$$\text{Working form is } Y = \overline{A \cdot B}$$

Hence, we have to bring the given Boolean expression into this form

#### 1. NOT gate (Inverter) using NAND gate

Figure shows the realization of a NOT gate (inverter) using a two input NAND gate. Both the inputs of the NAND are connected together, we can write that



Input = A = B

Hence, output is given by

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

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

$$A \cdot A = A$$

$$Y = \overline{A}$$

$$Y = \overline{\overline{A}}$$

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

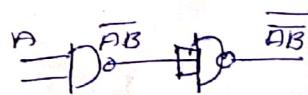
$$Y = \overline{\overline{A}} = 1$$

This expression is the Boolean expression for an inverter  
Hence the figure is an inverter

## 2. AND gate using NAND gate

The Boolean expression for an AND gate is

$$Y = A \cdot B$$



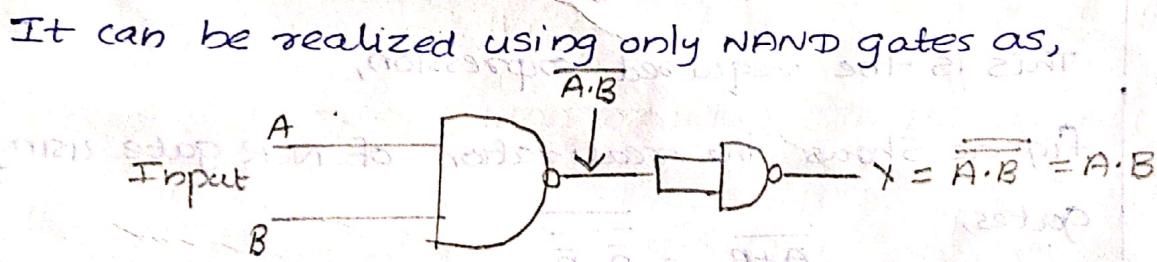
Taking double inversion of RHS, we get

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

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

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

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



## 3. OR gate using NAND gate

The Boolean expression for OR gate is,

$$Y = A + B$$

Taking double inversion on RHS, we get

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

But as per De Morgan's theorem, we know that

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

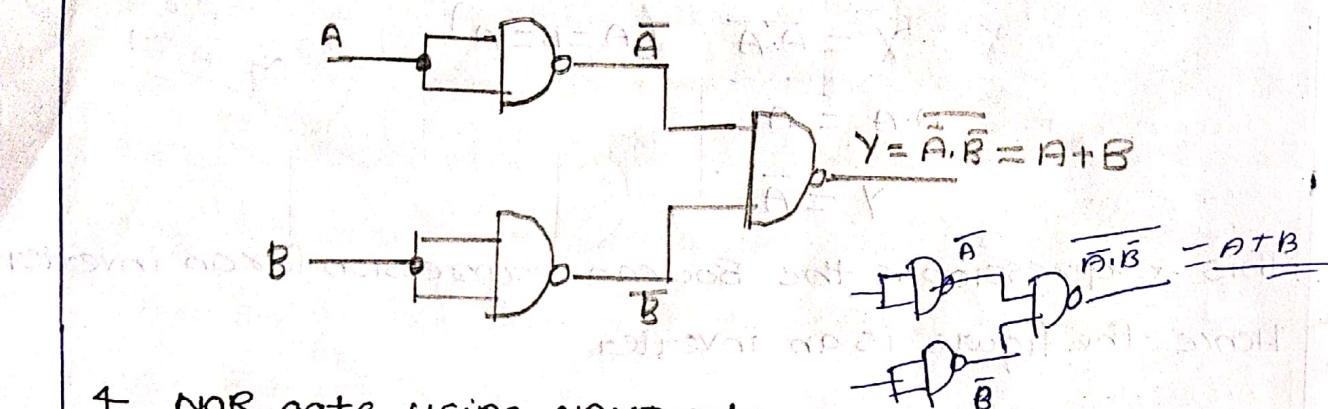
$$Y = \overline{\overline{A} \cdot \overline{B}}$$

Therefore,

BY De Morgan's

This is the required expression

So,  $Y = A + B = \overline{\bar{A} \cdot \bar{B}}$  and figure shows the realization



#### 4. NOR gate using NAND gate

The Boolean expression for NOR gate is given by

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

According to De Morgan's theorem, we have

$$Y = \bar{A} \cdot \bar{B}$$

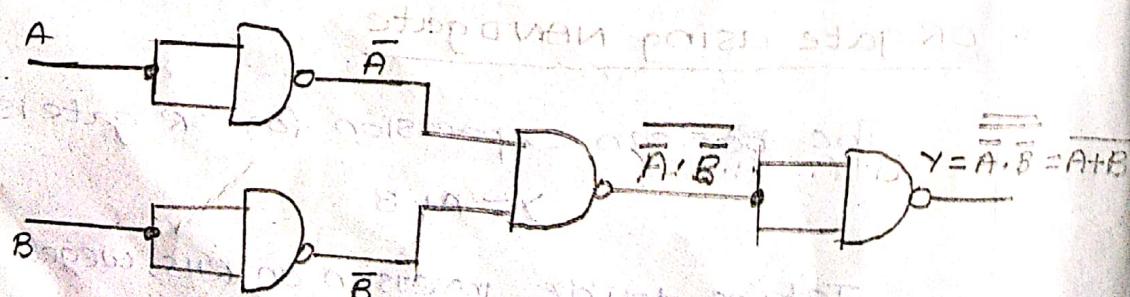
Taking double inversion of RHS, we write

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

This is the required expression,

Figure shows the realization of NOR gate using NAND gates,

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



(5) EX-OR gate using NAND gates only

The Boolean expression for EX-OR gate is,

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

Taking double inversion of RHS, we get

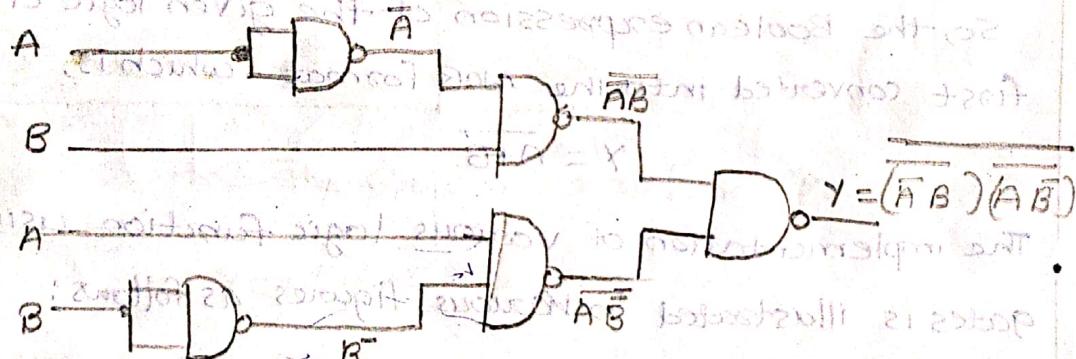
$$\underline{\underline{Y = \bar{A}\bar{B} + A\bar{B}}}$$

$$X = \bar{A}\bar{B} \text{ and } Z = A\bar{B}$$

$$Y = \underline{\underline{X + Z}}$$

Using De Morgan's theorem, we have  $\underline{\underline{X + Z = \bar{X} \cdot \bar{Z}}}$

Therefore, we have  $\underline{\underline{Y = \bar{X} \cdot \bar{Z} = (\bar{A}\bar{B}) \cdot (\bar{A}\bar{B})}}$



(6) EX-NOR using NAND gates only

The Boolean expression for EX-NOR is given by,

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

$$Y = \underline{\underline{X + Z}}$$

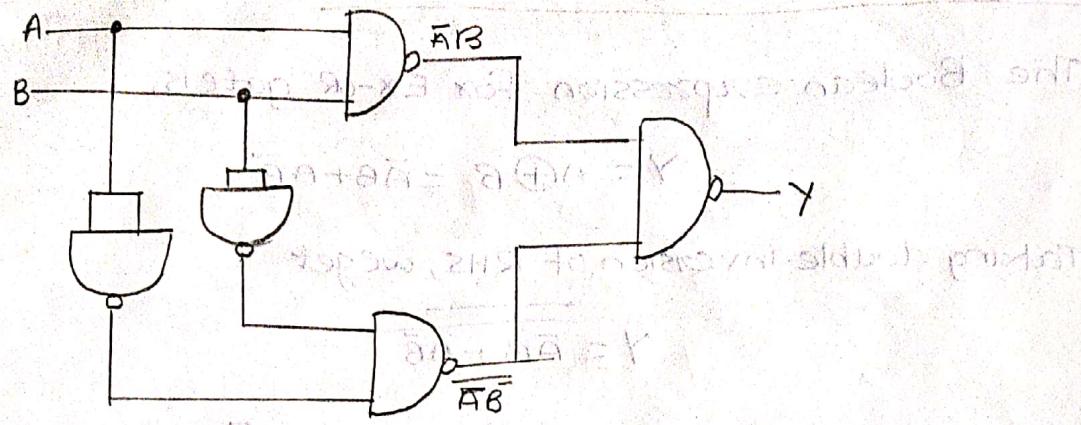
Taking double inversion of RHS, we get

$$\underline{\underline{Y = \bar{X} + \bar{Z} = \bar{X} \cdot \bar{Z}}}$$

Therefore, we have

$$\underline{\underline{Y = (\bar{A}\bar{B}) \cdot \bar{A}\bar{B}}}$$

EX-OR gate



### NOR gate as a universal gate

In this article, let us illustrate the implementation of every logic operation using only NOR gates. In order to do.

So, the Boolean expression of the given logic circuit must be first converted into the NOR format which is,

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

The implementation of various logic function using only NOR gates is illustrated in various figures as follows:

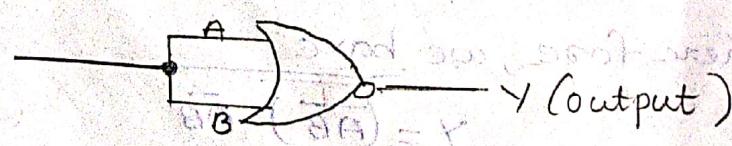
#### 1. NOR gate (inverter) using NOR gate.

We have the figure shows the realization of a NOT gate using only NOR gate. Since both the inputs of the NOR gate are connected together, we can write.

$$A = B = A + BA = Y$$

$$\text{Output of NOR, } Y = \overline{A+B} = \overline{AA} = Y$$

$$Y = \overline{A} \quad (\text{Invertor})$$



## 2. OR gate using NOR gates only.

We know that the Boolean expression for an OR gate is given by.

$$Y = A + B$$

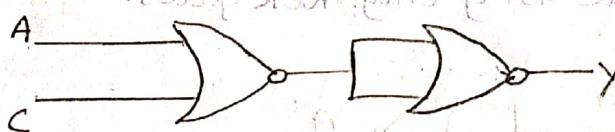
Taking double inversion of RHS, we get

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

This is the required expression.

Figure shows the realization of OR gate using only NOR gates.

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



## 3. AND gate using only NOR gates

We know that the Boolean expression for an AND gate is given by

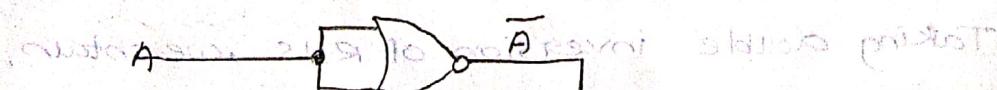
$$Y = A \cdot B$$

Taking double inversion of RHS, we obtain

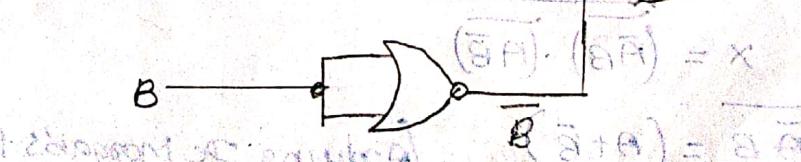
$$Y = \overline{\overline{A \cdot B}}$$

$$Y = \overline{\overline{A}} + \overline{\overline{B}} = \overline{A} \oplus \overline{B} = X$$

The required expression is,



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



#### (4) NAND gate using NOR gates

We know that Boolean expression for a NAND gate is given by

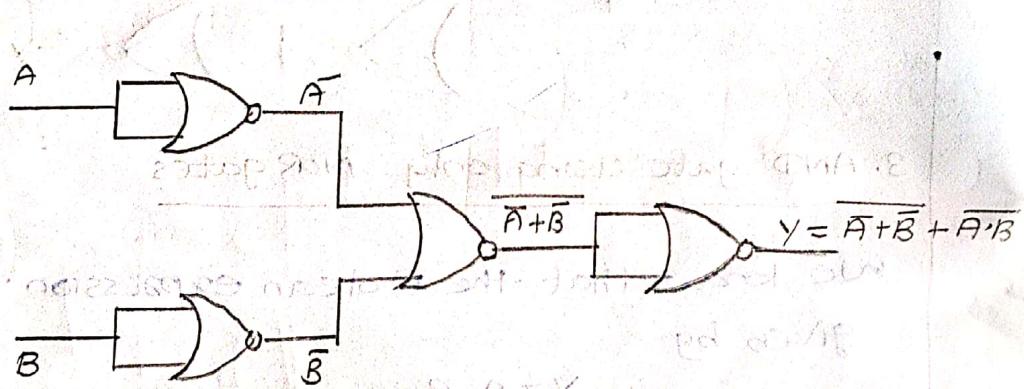
$$Y = \overline{A \cdot B}$$

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

Taking double inversion of RHS, we obtain.

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

- This is the required expression. Figure shows the realization of NAND gate using only NOR gates.



#### 5. EXOR gate using only NOR gates

We know that Boolean expression for a EX-OR gate is

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

$$X = \bar{A}B \text{ and } Z = A\bar{B}$$

$$Y = X + Z$$

Taking double inversion of RHS, we obtain,

$$Y = \overline{\overline{X} + \overline{Z}}$$

$$Y = \overline{\overline{X} \cdot \overline{Z}}$$

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

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

(Applying De Morgan's theorem)

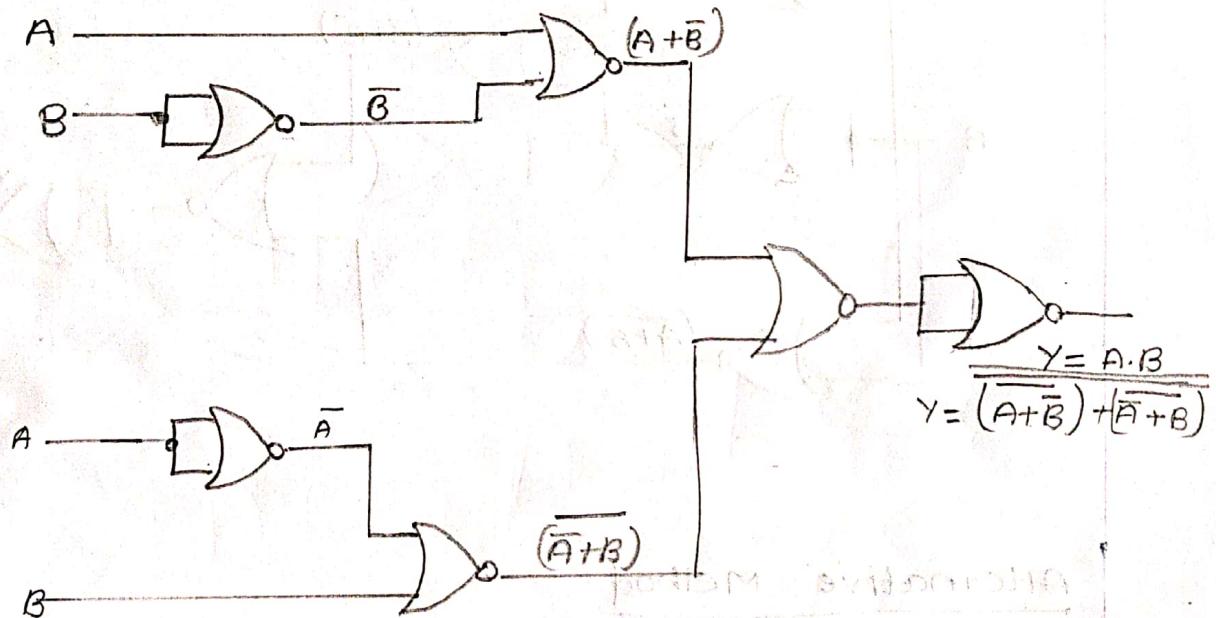
$$Y = \overline{(A+B)} \cdot \overline{(A+B)} \quad (\text{Applying De Morgan's theorem})$$

$$Y = \overline{(A+B)} \cdot \overline{(A+B)} \quad \text{Additional steps omitted}$$

Taking double inversion of RHS, we obtain,

$$Y = \overline{\overline{(A+B)} + \overline{(A+B)}} \quad \text{Additional steps omitted}$$

This is the required expression for EXOR gate using only NOR gates.



### (6) EX-NOR gate using only NOR gates

We know that the Boolean expression for an EX-NOR gate is given by

$$(A+B) + (A+B) = A \oplus B$$

$$Y = \overline{AB} + AB = X + Z$$

$$(A+B) + (A+B) = X \oplus Y \oplus Z$$

Taking double inversion of RHS, we obtain.

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

$$Y = \overline{\overline{AB} + AB} = X \oplus Y \oplus Z$$

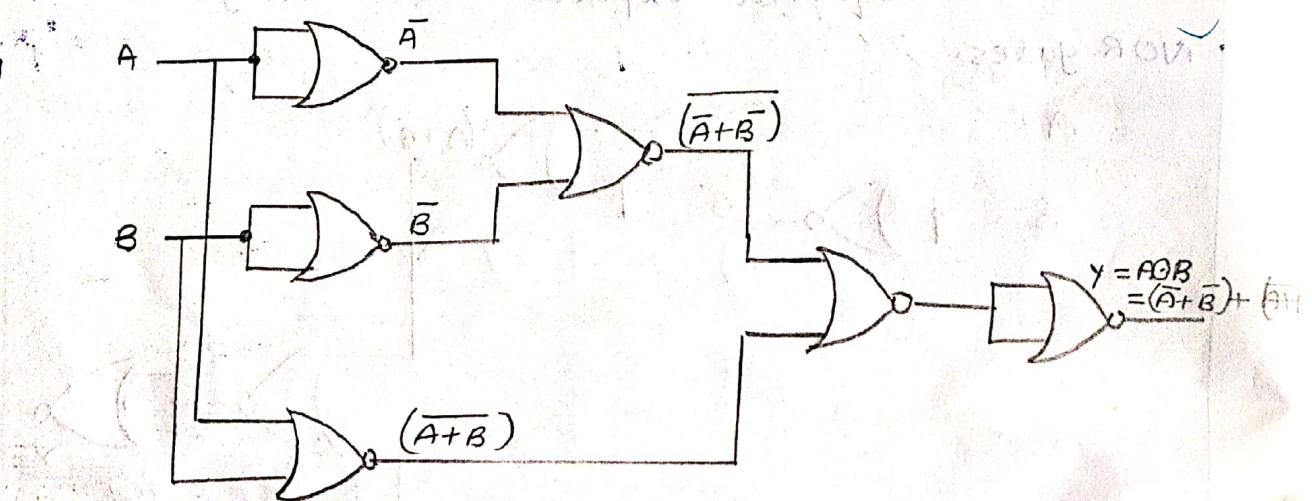
$$Y = (A+B) \cdot (\overline{A}+\overline{B}) \quad (\text{Applying De Morgan's theorem})$$

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

Taking double inversion of RHS, we obtain

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

This is the required expression. figure shows the realization of an EX-NOR gate using only NOR gates.



### Alternative Method

We know that EX-NOR function is given by

$$Y = A \oplus B = \overline{A \oplus B} = \overline{EX-OR}$$

We have already proved that for an EX-OR gate

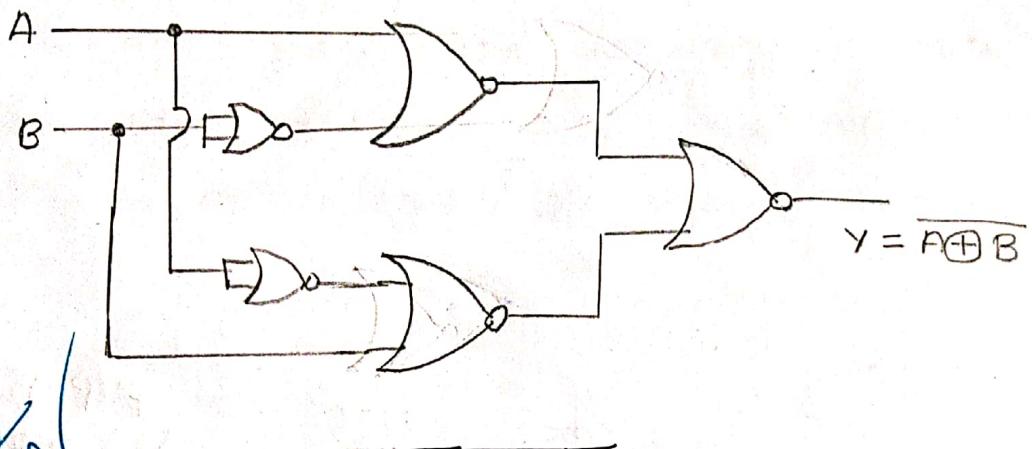
$$A \oplus B = \overline{(A+B)} + \overline{(\bar{A}+B)}$$

Therefore, EX-NOR  $Y = \overline{\overline{(A+B)}} + \overline{\overline{(\bar{A}+B)}}$

We can eliminate the upper 2 bars out of three, so that we may write

$$EX-NOR Y = \overline{(A+B)} + \overline{(\bar{A}+B)}$$

This is the required expression and has been shown in figure.



~~John  
To the~~

# Logic Gates

1. The output of an AND gate with three inputs, A, B, and C, is HIGH when \_\_\_\_\_.
  - A. A = 1, B = 1, C = 0
  - B. A = 0, B = 0, C = 0
  - C. A = 1, B = 1, C = 1
  - D. A = 1, B = 0, C = 1
  
2. If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?
  - A. 1
  - B. 2
  - C. 7
  - D. 8
  
3. If a signal passing through a gate is inhibited by sending a LOW into one of the inputs, and the output is HIGH, the gate is a(n):
  - A. AND
  - B. NAND
  - C. NOR
  - D. OR
  
4. A device used to display one or more digital signals so that they can be compared to expected timing diagrams for the signals is a:
  - A. DMM
  - B. spectrum analyzer
  - C. logic analyzer
  - D. frequency counter

5. A truth table illustrates how the input level of a gate responds to all the possible output level combinations.

- A. True
- B. False

6. A NOR gate output is LOW if any of its inputs is LOW.

- A. True
- B. False

7. As a rule, CMOS has the lowest power consumption of all IC families.

- A. True
- B. False

8. A popular waveform generator is the Johnson shift counter.

- A. True
- B. False

Ref:

<https://www.indiabix.com/digital-electronics/questions-and-answers/>

1. Convert hexadecimal value 16 to decimal.

A.  $22_{10}$

B.  $16_{10}$

C.  $10_{10}$

D.  $20_{10}$

---

2. Convert the following decimal number to 8-bit binary.

187

A.  $10111011_2$

B.  $11011101_2$

C.  $10111101_2$

D.  $10111100_2$

---

3. Convert binary  $11111110010$  to hexadecimal.

A.  $EE2_{16}$

B.  $FF2_{16}$

C.  $2FE_{16}$

D.  $FD2_{16}$

---

4. Convert the following binary number to decimal.

$01011_2$

A. 11

B. 35

C. 15

D. 10

---

5. Convert the binary number  $1001.0010_2$  to decimal.

A. 90.125

B. 9.125

C. 125

D. 12.5

7. The voltages in digital electronics are continuously variable.

A. True

B. False

11. Convert  $59.72_{10}$  to BCD.

A. 111011

B. 01011001.01110010

C. 1110.11

D. 0101100101110010

[View Answer](#) [Discuss in Forum](#) [Workspace Report](#)

8. Convert  $8B3F_{16}$  to binary.

A. 35647

B. 011010

C. 1011001111100011

D. 1000101100111111

[View Answer](#) [Discuss in Forum](#) [Workspace Report](#)

9. Which is typically the longest: bit, byte, nibble, word?

A. Bit

B. Byte

C. Nibble

D. Word

[View Answer](#) [Discuss in Forum](#) [Workspace Report](#)

10. Assign the proper odd parity bit to the code 111001.

A. 1111011

B. 1111001

C. 0111111

D. 001111



Reg. No.

**AUHIPPO.COM\*****Question Paper Code : 40952****B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2018****Third Semester****Electronics and Communication Engineering****EC6302 – DIGITAL ELECTRONICS****(Common to Mechatronics Engineering/Robotics and Automation Engineering)  
(Regulations 2013)****Time : Three Hours****Maximum : 100 Marks****Answer ALL questions****PART – A****(10×2=20 Marks)**

1. State De Morgan's theorem and mention its use..
2. What is meant by "maxterm" and "true maxterm" ?
3. What is the basic principle used in order to check or generate the proper parity bit in a given code word ?
4. Draw the logic diagram of a 4-bit parallel subtractor.
5. Bring out the difference between synchronous sequential circuits and asynchronous sequential circuits.
6. A binary ripple counter is required to count up to  $16,383_{10}$ . How many Flip-flops are required ? If the clock frequency is 8.192 MHz, what is the frequency at the output of the MSB ?
7. What is memory expansion and why is it required ?
8. A certain memory has a capacity of  $32K \times 16$ . How many bits are there in each word ? How many words are being stored and how many memory cells does this memory contain ?
9. Differentiate between an ASM chart and a conventional flow chart.
10. What is dynamic hazard ? When do they occur ?

**★ WWW.AUHIPPO.COM ★**



## PART - B

(5×13=65 Marks)

11. a) Explain the Tri-State configuration with neat diagram. (13)

(OR)

- b) Minimize the following expression : using Tabulation method

$$f = \sum m(0, 1, 2, 8, 9, 15, 17, 21, 24, 25, 27, 31) \quad (13)$$

12. a) Implement the following Boolean function using an 8 : 1 multiplexer considering D as the input and A, B, C as selection lines :

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

(OR)



(13)

- b) With a neat diagram, explain in detail about the working of a 4-bit look ahead carry adder. Also mention its advantage over conventional adder. (13)

13. a) Explain in detail about the Ring Counter with its logic diagram, state diagram and its sequence table. (13)

(OR)

- b) Discuss in detail about the Pulse-Triggered S-R flip flop. Also draw the output waveform of this flip flop and explain it with an example. (13)

14. a) Write the program table to implement a BCD to Excess-3 code conversion using a PLA. (13)

(OR)

- b) Explain in detail about the working of bipolar SRAM cell and single transistor DRAM cell with neat sketches. (13)

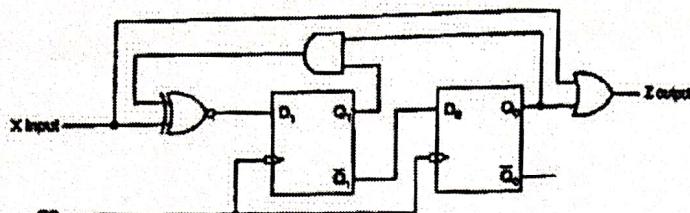
15. a) A clocked sequential circuit with single input x and single output z produces an output  $z = 1$  whenever the input x completes the sequence 1011 and overlapping is allowed :

i) Obtain the state diagram. (5)

ii) Obtain its minimum state table and design the circuit with D flip-flops. (8)

(OR)

b) Draw an ASM chart and the state diagram for the circuit as shown in the figure. (13)



PART - C

(1×15=15 Marks)

16. a) Design a J-K counter that goes through states 3, 4, 6, 7 and 3 .... Is the counter self-starting ? Modify the circuit such that whenever it goes to an invalid state it comes back to state 3. (15)

(OR)

b) A staircase light is controlled by two switches, one is at the top of the stairs and the other at the bottom of the stairs :

- i) Make a truth table for this system.
- ii) Write the logic equation in the SOP form.
- iii) Realize the circuit using AOI logic.
- iv) Realize the circuit using minimum number of (A) NAND gates and (B) NOR gates.

(15)

★ WWW.AUHIPPO.COM ★

{

## **Question Paper Code : 20408**

**B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2018.**

**Third Semester**

**Electronics and Communication Engineering**

**EC 6302 — DIGITAL ELECTRONICS**

**(Common to Mechatronics Engineering, Robotics and Automation Engineering)**

**(Regulations 2013)**

**(Also common to PTEC 6302 — Digital Electronics for B.E. (Part-time) Second Semester – Electronics and Communication Engineering – Regulations 2014)**

**Time : Three hours**

**Maximum : 100 marks**

**Answer ALL questions.**

**PART A — (10 × 2 = 20 marks)**

1. What are the universal gates? Justify.
2. State De-Morgan's Theorem.
3. Write the characteristic equation of  $4 \times 1$  Multiplexer.
4. State the differences between combinational and sequential circuits.
5. Draw the excitation table for D Flip Flop.
6. Draw the state diagram of 3 bit up counter.
7. Compare PAL, PLA and PROM.
8. Define setup time with timing diagram.
9. Mention the types of sequential circuits and give the difference between them.
10. Define Mealy machine with a state diagram.



PART B — (5 × 13 = 65 marks)

11. (a) Simplify  $f(W, X, Y, Z) = \sum m(2, 6, 8, 9, 10, 11, 14, 15)$  using Quine – Mc Cluskey method of minimization.

Or

- (b) Draw and Explain NAND, NOT and NOR gate CMOS representation.

12. (a) Draw  $4 \times 1$  multiplexer and  $1 \times 4$  Demultiplexer using gates and explain its operation.

Or

- (b) Design a 2 bit magnitude comparator and draw its logic circuit.

13. (a) Explain the logic circuit, characteristic and excitation table of JK, SR and D flip flop.

**AUHIPPO.COM**

Or

- (b) Design a 3 bit synchronous binary up-down binary using T flip flop.

14. (a) Design and implement a BCD to gray code converter using PAL.

Or

- (b) Write short notes on Static, Bipolar and MOSFET RAM cell.

15. (a) Elucidate the design procedure of synchronous sequential circuits.

Or

- (b) Design a sequential circuit whose state tables are specified in the Table below using D flip-flops.

Present State $Q_0, Q_1$	Next State		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
0 0	0 0	0 1	0	0
0 1	0 0	1 0	0	0
1 0	1 1	1 0	0	0
1 1	0 0	0 1	0	1

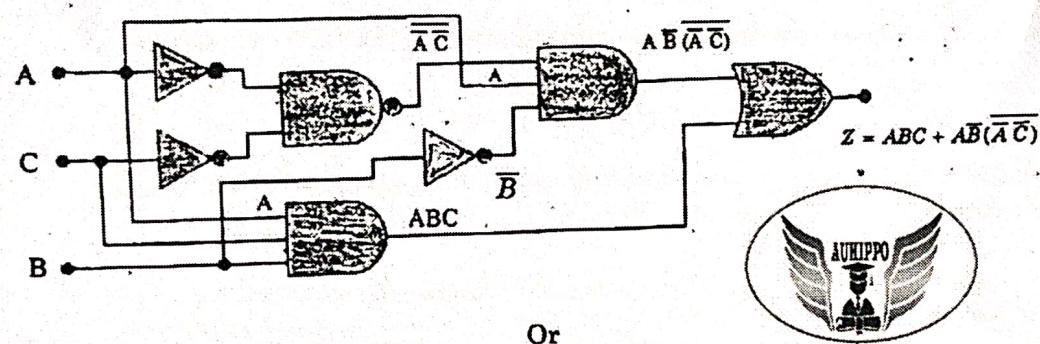
PART C — (1 × 15 = 15 marks)

16. (a) (i) Design a circuit that has a 3-bit binary input and a single output ( $Z$ ) specified as follows : (8)

$Z = 0$ , when the input is less than  $5_{10}$

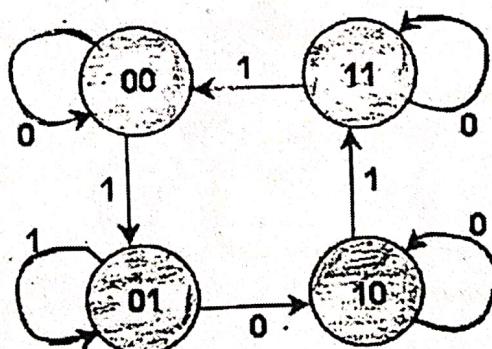
$Z = 1$ , otherwise

- (ii) Simplify the given logic circuit using Boolean Simplification. (7)



Or

- (b) Design a synchronous sequential circuit whose state diagram is shown below. (15)



**CBGS Scheme**

USN

--	--	--	--	--	--	--

15EC33

**Third Semester B.E. Degree Examination, Dec.2016/Jan.2017  
Digital Electronics**

Time: 3 hrs.

Max. Marks: 80

**Note: Answer FIVE full questions, choosing one full question from each module.****Module-1**

1. a. Define the following: i) TVUM Table; ii) Combinational circuit; iii) Canonical SOP; iv) Canonical POS. (04 Marks)  
 b. Obtain minimal expression using k-map for the following incompletely specified function:  
 $F(a, b, c, d) = \sum_m(0, 1, 4, 6, 7, 9, 15) + \sum_d(3, 5, 11, 13)$  and draw the circuit diagram using gates. (06 Marks)  
 c. Write the truth table and design a circuit to generate o/p using K-map for the problem statement given: o/p of a combinational circuit having 4 inputs and an o/p, becomes logical '1' when two or more inputs goto logic level '1'. (06 Marks)

**OR**

2. a. Define K-map, incompletely specified function, essential prime implicants and grey code. (04 Marks)  
 b. Obtain minimal logical expression for the given minterm expression using K-map.  
 $f(a, b, c, d) = \sum_m(0, 1, 4, 5, 6, 7, 9, 14, 11, 15, 15)$ . (04 Marks)  
 c. Use Quine McCluskey's method of minimization to obtain essential prime implicants and minimal expression for the following minterm expression:  
 $f(a, b, c, d) = \sum_m(0, 1, 4, 5, 6, 8, 13, 15) + \sum_d(2)$ . (08 Marks)

**Module-2**

3. a. Define encoder, decoder, priority encoder and multiplexer. (04 Marks)  
 b. Write block diagram representation of a full adder using 3:8 decoders. (04 Marks)  
 c. Design full adder using i) 2:1 MUX and ii) 4:1 MUX. (08 Marks)

**OR**

4. a. Explain carry lookahead adder with neat diagram and relevant expressions. (08 Marks)  
 b. Design 2-bit comparator and briefly explain. (08 Marks)

**Module-3**

5. a. Define bistable element, latch, flip-flop and function table. (04 Marks)  
 b. Sketch timing diagrams for JK flipflop and D-flip-flop. (06 Marks)  
 c. Explain M/S JK flip-flop with the help of circuit diagram and waveforms. (06 Marks)

**OR**

Find characteristic equations for T and SR-flip-flops with the help of function tables. (06 Marks)

Write circuit diagram for the edge triggered D-flip-flop and provide explanation for different input condition. (06 Marks)

c. Explain the operation of a switch debouncer built using SR-latch with the help of waveforms. (04 Marks)

1 of 2

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.  
 2. Any revealing of identification, appeal to evaluator and/or equations written e.g. 42+8 = 50, will be treated as malpractice.

15EC33

Module-4

- 7 a. Define register, asynchronous ripple counter synchronous counter and ring counter. (04 Marks)  
 b. Design mod-8 counter using right shift register. Use D-flip-flop to build register circuit. Explain the operation using function table. (06 Marks)  
 c. Write timing diagrams, counting sequence and the logic diagram for 4-bit ripple counter and briefly explain. (06 Marks)

**OR**

- 8 a. Explain PIPO and SIPO operations using single diagram. (04 Marks)  
 b. Design Mod-6 synchronous counter using JK flip-flop. The sequence is 000, 001, 011, 100, 101, 111...000. (07 Marks)  
 c. Write state diagram for Mod-5 self correcting counter and briefly explain. The sequence is 000, 001, 101, 110, 111, 000. (03 Marks)

Module-5

- 9 a. What are Melay and Moore models of a sequential circuit? Briefly explain with diagrams. (04 Marks)  
 b. Write characteristic/excitation table for JK flip-flop and explain. (03 Marks)  
 c. Analyze the following sequential circuit. Write excitation equations K-maps and state diagrams to analyze. (09 Marks)

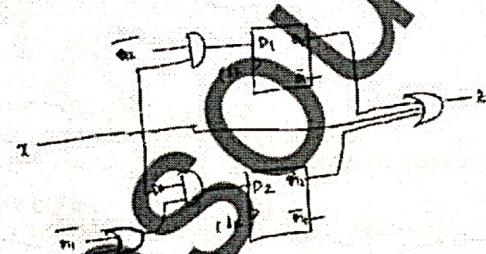


Fig.Q.9(c)

**OR**

- 10 a. Write state diagrams for a four state machine using Melay and Moore models and briefly explain. (04 Marks)  
 b. What is a state table? Give an example. (02 Marks)  
 c. Design a counter circuit for the following state table. Follow the standard steps for design. (10 Marks)

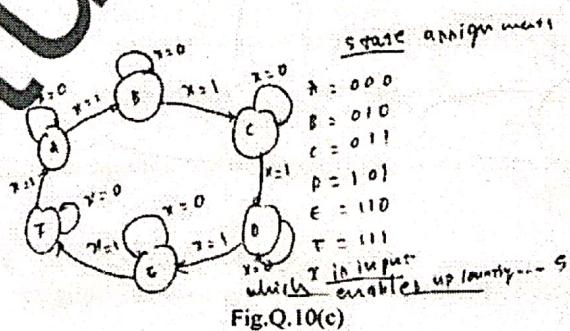


Fig.Q.10(c)

\*\*\*\*\*

2 of 2



Reg. No.

<b>AUHIPPO.COM*</b>												



## Question Paper Code : 77112

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2015.

Third Semester

Electronics and Communication Engineering

EC 6302 — DIGITAL ELECTRONICS

(Common to Mechatronics Engineering and Robotics and Automation Engineering)

(Regulation 2013)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)



1. Convert  $Y = A + B\bar{C} + AB + \bar{A}BC$  into canonical form.
2. State the advantages of CMOS logic.
3. Draw the two bit comparator circuit using logic gates.
4. Write down the difference between demultiplexer and decoder.
5. Draw the truth table of RS flipflop.
6. What is the minimum no of flip-flop needed to design a counter of modulus 60?
7. What is the basic difference between the RAM & ROM circuitry?
8. Compare a static and dynamic RAM cell.
9. Distinguish between a combinational logic circuit and a sequential logic circuit.
10. What is the most important consideration in making state assignments for asynchronous network?

★ [WWW.AUHIPPO.COM](http://WWW.AUHIPPO.COM) ★

PART B — (5 × 16 = 80 marks)

11. (a) Simply using Quine Mccluskey method and verify your result using K-map  $F = \Sigma(0,1,2,5,7,8,9,10,13,15)$ . (16)

Or

- (b) (i) Express the Boolean functions  $F = A + \bar{B}C$  in a sum of minterms. (10)  
(ii) Simplify the following Boolean expression using Boolean algebra.  
(1)  $\bar{x}\bar{y}z + \bar{x}yz + x\bar{y}$  (3)  
(2)  $x\bar{y}z + \bar{x}z + yz$ . (3)



12. (a) (i) Design a 4\*1 multiplexer circuit. (8)  
(ii) Implement the function using multiplexer  $F = \Sigma(0,1,3,4,8,9,15)$ . (8)

Or

- (b) (i) Draw the logic diagram of Binary to octal decoder and explain the working in detail. (8)  
(ii) How is the carry look ahead adder faster than a ripple carry adder? Explain in detail with neat sketches. (8)

13. (a) Using D flipflops design a synchronous counter which counts in the sequence. 000, 001, 010, 011, 100, 101, 110, 111, 000. (16)

Or

- (b) (i) Discuss the working of a 4 bit Johnson counter with neat block diagram. (8)  
(ii) Explain the functioning of a recirculating shift register with various modes of operation. (8)
14. (a) (i) Explain memory decoding. Compare the RAM, ROM, PROM & EPROM. (8)  
(ii) Draw a RAM Cell and explain its working in detail. (8)

Or

- (b) Write short notes on with suitable schematic (16)  
(i) Programmable Logic Array (PLA).  
(ii) Field Programmable Gate Arrays (FPGA).

15. (a) (i) Explain how a state graph for a sequential machine can be converted to an equivalent ASM chart. (8)  
(ii) Derive the ASM chart for binary multiplier. (8)

Or

- (b) (i) When is a sequential machine said to be strongly connected. (2)  
(ii) Design a sequential pattern detector that receives a stream of input bits. The circuit should recognize the pattern 010 and produce an output whenever this pattern is received. (14)

Alphanumeric codes are codes used to encode the characters of alphabet in addition to the decimal digits. They are used primarily for transmitting data between computers and its I/O devices such as printers, keyboards and video display terminals. Because the no. of bits used in most alphanumeric codes is much more than those required to encode 10 decimal digits and 26 alphabet characters, these code include bit patterns for a wide of other symbols and functions as well. The most popular alphanumeric codes are the ASCII code and the EBCDIC code.

### The ASCII Code:-

The American Standard Code for Information Interchange (ASCII) pronounced as 'ASKEE' is a widely used alphanumeric code. This is basically a 7-bit code. Since the number of different bit patterns that can be created with 7-bits is  $2^7 = 128$ , the ASCII can be used to encode both lowercase and uppercase characters characters of the alphabet (52 symbols) and some special symbols as well, in addition to the

terminals that interface with small computer system (T).  
 Make Many Large systems also make provisions for its accomodation. Because characters are assigned in ascending binary numbers. ASCII is very easy for a computer to alphabetize & sort.

MSBs

LSBs

	000	001	010	011	100	101	110	111
0000	NUL	DEL	Space	0	①	P	P	
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	I	I
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	1	n	~
1111	SI	US	/	?	O	-	o	DLE

ACK - Acknowledge

DC3 - Direct Control 3

BEL - Bell

DC4 - Direct Control 4

BS - Backspace

DEL - Delete idle

CAN - Cancel

DLE - Data Link Escape

CR - Carriage Return

EM - End of Medium

~

ENQ - Enquiry