Number Systems

1.What is Data?

- Data is defined as an un- processed collection of raw facts,
- The data is a fact about people, places or some objects.
- suitable for communication, interpretation or processing.
- It is an input of the computer.
- It will not giving any meaningful message.
  Ex. 134, 16 'Kavitha', 'C'

2.Define Bit or What is the basic unit of data?

- A bit is the short form of Binary digit.
- Which can be '0' or '1'.
- It is the basic unit of data in computers.

3.Define nibble

- A nibble is a collection of 4 bits (Binary digits).

4.Define Byte. What is the basic unit of memory size?

- A collection of 8 bits is called Byte.
- A byte is considered as the basic unit of measuring the memory size in the computer.

5.Define Word length

- Word length refers to the number of bits processed by a Computer's CPU.
- Ex. 8bits, 16 bits, 32 bits and 64 bits

6.How Computer memory is represented?

- Computer memory (Main Memory and Secondary Storage)is normally represented in terms of KiloByte (KB) or MegaByte (MB).
- In binary system, 1 KiloByte represents 1024 bytes that is 2^{10} .

7.How computers are handle the data? What is Machine language?

- Computer handles data in the form of '0'(Zero) and '1' (One).
- Any kind of data like number, alphabet, special character should be converted to '0' or '1' which can be understood by the Computer.
- Computer understandable language is called Machine language(0 and 1)

8.How characters are represented in computer explain with examples?

- Bytes are used to represent characters in a text.
- Different types of coding schemes are used to represent the character set and numbers.
- The most commonly used coding scheme is the **American Standard Code for Information Interchange (ASCII).**

9.How speed of computer is described?

- The speed of a computer depends on the number of bits it can process at once.
- For example, a 64- bit computer can process 64-bit numbers in one operation
- While a 32-bit computer break 64-bit numbers down into smaller pieces, making it slower.

10.What is radix of a number system? Give example What are the different types of Number System?

- Radix or base is number of digits in each number system.
- Each number system is uniquely identified by its base value or radix.
- Radix or base is the general idea behind positional numbering system.
- A numbering system is a way of representing numbers. They are,
- Decimal number system(Base Value 10) 0,1,2,3,4,5,6,7,8,9
- Binary number system(Base Value 2)0,1
- Octal number system(Base Value 8) 0,1,2,3,4,5,6,7
- Hexadecimal number system(Base Value 16) 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

11. Explain 1's Complement representation.

- Used to represent signed numbers.
- This is for negative numbers only.

Step 1:Convert given Decimal number into Binary

Step 2: Check if the binary number contains 8 bits , if less add 0 at the left most bit, to make it as 8 bits.

Step 3: Invert all bits (i.e. Change 1 as 0 and 0 as 1)

12.Write short note on Decimal Number system

- It consists of 0,1,2,3,4,5,6,7,8,9
- The base is 10.
- It is the oldest and most popular number system used in our day to day life.
- The positional value as a power of 10.Ex. 28,11

13.Write short note on Binary Number System

- It consists of 0 and 1. The base is 2.
- The positional value as a power of 2.
- The left most bit in the binary number is called as the Most Significant Bit (MSB)
- It has the largest positional Value.
- The right most bit in the binary number is called as the Least Significant Bit (LSB)
-
- It has the smallest positional Value..

14.Write short note on Octal Number System

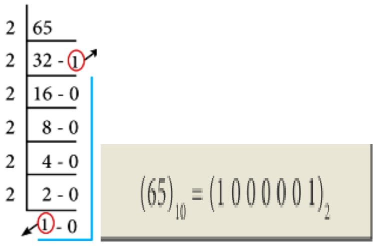
- It consists of 0,1,2,3,4,5,6,7
- The base is 8.
- Each octal digit has its own positional value as a power of 8

15.Write short note on Hexa decimal Number System

- It consists of 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- The base is 16.
- The positional value as a power of 16.

16.Decimal to Binary Conversion

Repeated Division by 2



Sum of Powers of 2 method.

Given Number : 65

Equivalent or value less than power of 2 is : 64

(1) 65 - 64 = 1

(2) 1 - 1= 0

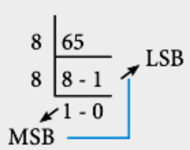
Power's of 2	64	32	16	8	4	2	1
Binary Number	1	0	0	0	0	0	1

$65_{10} = (1000001)_2$

17.Decimal to Octal Conversion

Repeated Division by 8

Convert $(65)_{10}$ into its equivalent Octal number



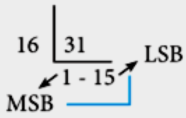
$(65)_{10} = (1\ 0\ 1)_8$

$(65)_{10} = (101)_8$

18.Decimal to Hexadecimal Conversion

Repeated Division by 16

Convert $(31)_{10}$ into its equivalent hexadecimal number.



$(31)_{10} = (1F)_{16}$

fractional Decimal to Binary

19.Conversion of fractional Decimal (0.2)to Binary

	Integer part	
$0.2 \times 2 = 0.4$	0	(first integer part obtained)
$0.4 \times 2 = 0.8$	0	
$0.8 \times 2 = 1.6$	1	
$0.6 \times 2 = 1.2$	1	
$0.2 \times 2 = 0.4$	0	(last integer part obtained)

Note:

- Fraction repeats, the product is the same as in the first step.
- Write the integer parts from top to bottom.
- Hence $(0.2)_{10} = (0.00110011...)_2$

19.Write procedure to convert fractional Decimal to binary with an example.

By using repeated multiplication by 2 method .

Step 1:Multiply the decimal fraction by 2.

The integer part is either 0 or 1.

Step 2: Multiply the fractional part of the previous product by 2.

Step 3:Repeat Step 1until the same fraction repeats or terminates (0).

Step 4: The final answer is to be written from first integer part to the last integer part obtained.

Convert(98.46) to binary

98
49 0
24 1
12 0
6 0
3 0
1 1 (98) = 1100010

0.46 x 2 = 0.92 = 0
0.92 x 2 =1.84 = 1
0.84 x 2 =1.68 =1
0.68 x 2 =1.36 = 1
0.36 x 2 = 0.72 = 0
0.72 x 2 =1.44 = 1
…. Top to Bottom 011101

(98.46)10 = 1100010 . 011101……

Convert $(250)_{10}$ into Binary, then convert that binary number into octal

Binary to Decimal Conversion

20.Convert $(111011)_2$ into its equivalent decimal number.

1	1	1	0	1	1	
x	x	x	x	x	x	
2^5	2^4	2^3	2^2	2^1	2^0	
=	=	=	=	=	=	
32 +	16 +	8 +	0 +	2 +	1	(59)

$(111011)_2$ = $(59)_{10}$

Conversion Table

Hex	Oct	Dec	Binary			
0	0	0	0	0	0	0
1	1	1	0	0	0	1
2	2	2	0	0	1	0
3	3	3	0	0	1	1
4	4	4	0	1	0	0
5	5	5	0	1	0	1
6	6	6	0	1	1	0
7	7	7	0	1	1	1
8		8	1	0	0	0
9		9	1	0	0	1
A			1	0	1	0
B			1	0	1	1
C			1	1	0	0
D			1	1	0	1
E			1	1	1	0
F			1	1	1	1

Binary to Octal Conversion

21.Convert $(11010110)_2$ into octal equivalent number

__Step__ 1: Group the given number into **3 bits** from right to left.

011 010 110

Note: The left most groups have less than 3 bits, so 0 is added to its left to make a group of 3 bits.

__Step__-2: Find Octal equivalent of each group

011 010 110

3 2 6

$(11010110)_2$ = $(326)_8$

Binary to Hexadecimal Conversion

22.Convert $(1111010110)_2$ into Hexadecimal number

Step 1:Group the given number into 4bits from right to left.

0011 1101 0110

Note: 0's are added to the left most group To make it a group of 4 bits

0011 1101 0110

3 D 6

$(1111010110)_2$ = **(3D6)** $_{16}$

Conversion of fractional Binary to Decimal

23.Conversion of fractional Binary to Decimal equivalent

Positional notation	Weight
2^{-1} (1/2)	0.5
2^{-2} (1/4)	0.25
2^{-3} (1/8)	0.125
2^{-4} (1/16)	0.0625
2^{-5} (1/32)	0.03125
2^{-6} (1/64)	0.015625
2^{-7} (1/128)	0.0078125

24.Convert the given Binary number $(11.011)_2$ into its decimal equivalent

Integer part $(11)_2$ = 3

2^1	2^0		2^{-1}	2^{-2}	2^{-3}
↑	↑		↑	↑	↑
1	1	.	0	1	1

$3 + .(0\times0.5 + 1\times0.25 + 1\times0.125)$

$= 3.375$

$(11.011)_2 = (3.375)_{10}$

Octal to Decimal Conversion:

25.Convert $(1265)_8$ to equivalent Decimal number

Positional	8^3	8^2	8^1	8^0
Given				
Weight	512	64	8	1
Number	1	2	6	5

$(1265)_8 = 512 \times 1 + 64\times2 + 8\times6 + 1\times5$

$= 512 + 128 + 48 + 5$

$(1265)_8 = (693)_{10}$

Octal to Binary Conversion

26.Convert $(6213)_8$ to equivalent Binary number

6	2	1	3
↓	↓	↓	↓
110	010	001	011

$(6213)_8 = (110010001011)_2$

Hexadecimal to Decimal Conversion

27.Convert $(25F)_{16}$ into its equivalent Decimal number.

Weight	256	16	1
Positional Notation	16^2	16^1	16^0

Given number = $(2\ 5\ F)_{16}$

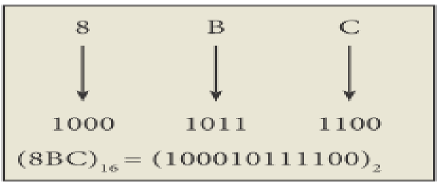
$(25F)_{16} = 2\times256 + 5\times16 + 15\times1$

$= 512 + 80 + 15$

$(25F)_{16} = (607)_{10}$

Hexadecimal to Binary Conversion

28.Convert (8BC) 16 into equivalent Binary number



How to Representation for Signed Numbers in Binary

29.Define sign Bit.

- The left most bit in the binary number is called as the **M**ost **S**ignificant **B**it (MSB)
- It is also called **sign bit or parity bit.**
- If this bit is 0, it is a positive number
- if it 1, it is a negative number**.**
- A signed binary number has 8 bits,
- only 7 bits used for storing values (magnitude) or data and the 1 bit is used for sign.

30.Define Signed Magnitude

- The simplest method to represent negative binary numbers is called **Signed Magnitude**.

31.How Numbers are represented in Computers?

- Signed Magnitude representation
  Ex. +43 or 43 is a positive number
  –43 is a negative number
- 1's Complement
- 2's Complement

32.Explain 1's and 2's Complement representation.

- Used to represent signed numbers.
- This is for negative numbers only.

Step 1:Convert given Decimal number into Binary

Step 2: Check if the binary number contains 8 bits , if less add 0 at the left most bit, to make it as 8 bits.

Step 3: Invert all bits (i.e. Change 1 as 0 and 0 as 1)

2's Complement representation

Step 1. Invert all the bits in the binary sequence.

Step 2. Add 1 to (LSB).

Example

33.Ex.Find 1's complement for (–24) $_{10}$

1's

Binary value of 24 is 00011000

Invert all bits 11100111

2's Complement represent of (-24) $_{10}$

Binary equivalent of +24:	11000
8bit format:	00011000
1's complement:	11100111
Add 1 to LSB:	+1
2's complement of -24:	11101000

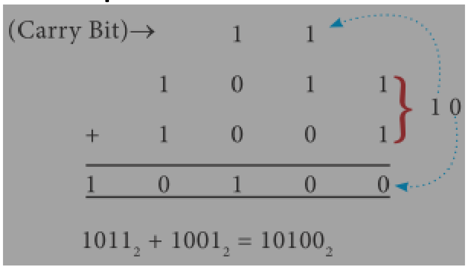
34.We cannot find 1's complement for (28) $_{10}$. State reason: Because 28 is a positive number.
This is for negative numbers only

Binary Arithmetic

Binary Addition Table

A	B	SUM (A + B)	Carry
0	0	0	-
0	1	1	-
1	0	1	-
1	1	0	1

35.Example: Add: 1011 $_2$ + 1001 $_2$



36.Perform Binary addition for the following: .

23 $_{10}$ + 12 $_{10}$

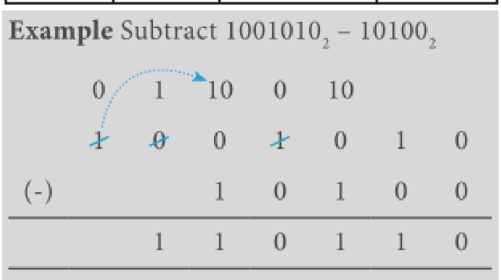
Step 1: Convert 23 and 12 into binary form

23	10111	in 8bits	-00010111
12	1100		-00001100

23+12 =35 => **00100011**

Binary Subtraction

A	B	Difference (A-B)	Borrow
0	0	0	0
1	0	1	0
1	1	0	0
0	1	1	1



37.What are the encoding systems used for computer.?

There are several encoding systems used for computer. They are,

- **BCD** – Binary Coded Decimal
- **EBCDIC** – Extended Binary Coded Decimal Interchange Code
- **ASCII** – American Standard Code for Information Interchange
- **Unicode**
- **ISCII** - Indian Standard Code for Information Interchange

38.Define Binary Coded Decimal (BCD).

- This encoding system is not in the practice right now.
- This is 2^6 bit encoding system.
- This can handle 2^6 = 64 characters only.

39.Define American Standard Code for Information Interchange (ASCII).

- This is the most popular encoding System
- This encoding system can handle English characters only.
- This can handle 2^7 bit which means 128 characters..
- The binary representation of ASCII (7 bits) value is 1000001
- The new edition (version) ASCII -8, has 2^8 bits and can handle 256 characters ..
- The binary representation of ASCII (8 bits) value is 01000001
- Each character has individual number.

The ASCII value for

- blank space is 32
- 0 is 48.
- Thelower case alphabets is from 97 to 122
- The upper case alphabets is from 65 to 90.

40.Extended Binary Coded Decimal Interchange Code (EBCDIC)

- It is 8 bit representation.
- This coding system is formulated by International Business Machine(IBM).
- The coding system can handle 256 characters.
- The input code in ASCII can be converted to EBCDIC system and vice - versa.

41.Indian Standard Code for Information Interchange (ISCII)

- ISCII is the system of handling the character of Indian local languages.
- It is a 8-bit coding system.
- Therefore it can handle 256 (2^8) characters.
- It is recognized by Bureau of Indian Standards (BIS).
- It is integrated with Unicode.

42.Define Unicode

- This coding system is used in most of the modern computers.
- This is 16 bit code and can handle 65536characters.
- Unicode can handle Universal languages.
- Unicode scheme is denoted by hexadecimal numbers.

1. Identify the number system for the following numbers

S. No.	Number	Number system
1	$(1010)_{10}$	Decimal Number system
2	$(1010)_2$	
3	$(989)_{16}$	
4	$(750)_8$	
5	$(926)_{10}$	

2. State whether the following numbers are valid or not. If invalid, give reason.

S.No.	Statement	Yes / No	Reason (If invalid)
1.	786 is an Octal number		
2.	101 is a Binary number		
3.	Radix of Octal number is 7		

3.Convert the following Decimal numbers to its equivalent Binary,Octal,Hexadecimal.

1)1920 2)255 3)126

3)126

126 Divided by 2
127 -1
63 -1
31 -1
15 -1
7 -1
3 -1
1 -1

$(11111111)_2$ - Binary

To Octal (By using Table) Ref.b.Pg.22

011 111 111
3 7 7

$(377)_8$ - Octal

To Hexadec.

1111 1111
15 15

$(ff)_{16}$ - Hexadec.

4.Convert the given binary number into its equivalent Decimal,Octal and Hexadecimal
1)101110101 2)10110 3)101011111

5.Convert the following octal numbers into Binary numbers
1)472 2)145 3)347 4)6247 5)645
<u>1)472 (Use table Method) Ref.b.Pg.22</u>

4	7	2
100	111	010

$(472)_8 = (100111010)_2$

6.Convert the following Hexadecimal numbers to Binary numbers
1)A6 2)BE 3)9BC8 D)BC9
<u>EX.BC9 (Use table Method) Ref.b.Pg.22</u>

B	C	9
1011	1100	1001

$(BC9)_{16} = (101111001001)_2$

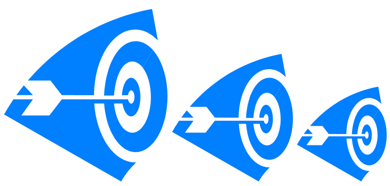
7.Write the 1's complement number and 2's compliment number for the following decimal numbers
1)22 2)-13 3)65 4)-46 5)255
<u>2)-13</u>

13	
6	-1
3	-0
1	-1

Binary Equivalent of +13 =1101
8-bit format =00001101
1'scompliment =11110010
Add 1 to LSB = +1
2's compliment of -13 = **$(11110011)_2$**

8.Perform the following binary component
1)$10_{10}+15_{10}$ 2)$-12_{10} + 5_{10}$ 3)$14_{10} – 12_{10}$
4)$-2_{10} – (-6_{10})$
$-2_{10} – (-6_{10})$
$- 2 + 6 = 4_{10} = (100)_2$
8- Bit = $(00000100)_2$

a) Add $1101010_2 + 101101_2$
b) Subtract $1101011_2 - 111010_2$



Part - II - Boolean Algebra

1.What is Boolean algebra?

- Boolean algebra is a mathematical discipline that is used for designing digital circuits in a digital computer.
- It describes the relation between inputs and outputs of a digital circuit.

2.Define Logical Operations:What are the basic logical operators (fundamental operators)?

The basic logical operations are

- AND, OR and NOT
- Represented by dot (.), plus (+), and by over bar / single apostrophe respectively.

3.Define TRUTH TABLE

- A truth table represents all the possible values of logical variable (input) or statements along with all the possible results(output) of given combination of truth values.

4.What is Gate? What are the fundamental gates?

- Gate is a basic electronic circuit.
- It operates on one or more input signals to produce an output signal.
- There are three fundamental gates namely AND, OR and NOT.

5.Explain about AND operator

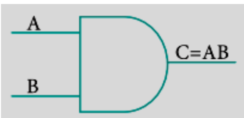
- The AND operator has **two or more** input variables and **one** Output.
- The output is **TRUE** when **all** the Inputs are **TRUE**.

Algebraic expression : Y=A . B

TRUTH TABLE

A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

AND GATE



6.Explain about OR operator

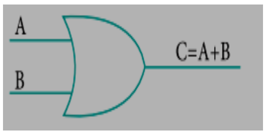
- The OR operator has **two or more input** variables and **one output** .
- The output is TRUE if **at least** one input is TRUE.

Algebraic expression : Y = A + B

TRUTH TABLE

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

OR GATE



7.Explain about NOT operator

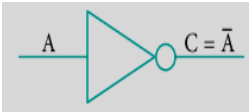
- The NOT Operator has one input and one output
- The NOT operator inverts the input.

Algebraic expression : Y = $\bar{A}$

TRUTH TABLE

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

NOT GATE



8.Consider the following equation
D = A + ($\bar{B}$. C) Write truth table and Find the output of D when inputs A=0,B=1,and C=0.

A	B	C	$\bar{B}$	($\bar{B}$.C)	D =A + ($\bar{B}$ .C)
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

Result: **D=0**

9.What are derived gates

- The gates which are derived from fundamental gates are called derived Gate.
- Ex. NAND ,NOR,XOR,XNOR etc…..

10.Why the NAND and NOR gates are called universal gate?

- NAND and NOR gates are called Universal gates, because the fundamental logic gates can be realized through them

11.Explain NOR Operator with an example

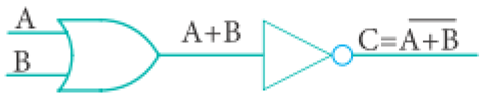
- The NOR is the combination of NOT and OR
- The NOR is generated by inverting the output of an OR operator.

Algebraic expression : Y= $\overline{\boldsymbol{A}+\boldsymbol{B}}$

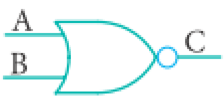
TRUTH TABLE

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

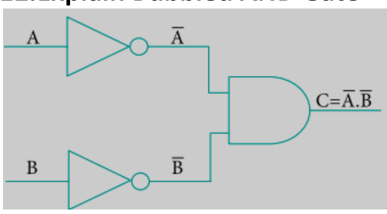
LOGIC CIRCUIT



LOGIC SYMBOL



12.Explain Bubbled AND Gate



- If we compare the truth tables of the bubbled AND gate with NOR gate, they are identical.
- So the circuits are interchangeable

TRUTH TABLE

-

A	B	‘A	**‘B**	**Y =** $\overline{\boldsymbol{A}}\,.\,\overline{\boldsymbol{B}}$
0	0	1	*1*	*1*
0	1	1	*0*	*0*
1	0	0	*1*	*0*
1	1	0	*0*	0

Algebraic expression : $\overline{\boldsymbol{A}+\boldsymbol{B}} = \overline{\boldsymbol{A}}\,.\,\overline{\boldsymbol{B}}$

De Morgan's First theorem – Proved

13.Explain NAND operator with Truth Table.

- The NAND is the combination of NOT and AND
- The NAND is generated by inverting the output of an AND operator

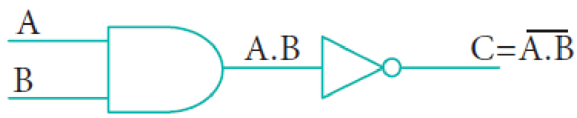
Algebraic expression :Y = $\overline{A.B}$

TRUTH TABLE

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

- The output is "**false**" if ALL inputs are "**true**", otherwise, the output is "**true**"

LOGIC CIRCUIT

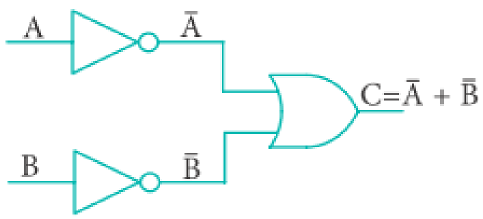


LOGIC SYMBOL

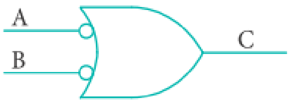
14.Explain Bubbled OR Gate

Algebraic expression : C= $\overline{\boldsymbol{A}} + \overline{\boldsymbol{B}}$

LOGIC CIRCUIT



LOGIC SYMBOL



TRUTH TABLE

A	B	$\bar{A}$	$\bar{B}$	Y = $\bar{A} + \bar{B}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

NAND = BUBBLED OR

- If we compare the truth tables of the bubbled OR gate with NAND gate, they are identical.
- So the circuits are interchangeable.

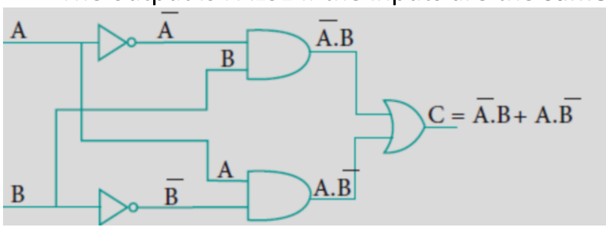
Algebraic expression : $\overline{\boldsymbol{A}.\boldsymbol{B}} = \overline{\boldsymbol{A}} + \overline{\boldsymbol{B}}$

De Morgan's Second theorem – Proved.

How AND and OR can be realized using NAND and NOR gate. (Ref. 11,12,13,14)
Prove and explain De Morgan 's theorem (ref. 11,12,13,14)

15.Explain XOR Gate with Truth Table.

- It is called exclusive - OR gate
- The output is TRUE if the inputs are **different**,
- The output is FALSE if the inputs are the **same**



Algebraic expression : C = $\bar{A}$.B + A . $\bar{B}$

TRUTH TABLE

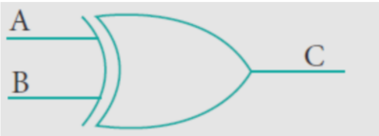
A	B	$\bar{A}$	$\bar{B}$	$\bar{A}$. B	A . $\bar{B}$	$\bar{A}$. B + A . $\bar{B}$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

In boolean algebra. In boolean algebra ⊕ or "encircled plus"stands for the XOR

Therefore

$C = A \oplus B$

Logic Symbol



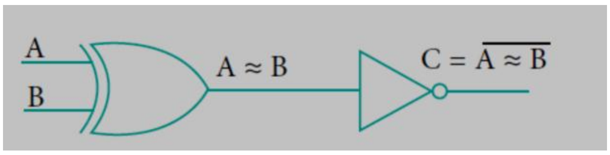
16.Explain XNOR Gate with Truth Table

- It is also called exclusive - NOR gate
- It is a combination XOR gate followed by an inverter.
- The output is FALSE if the inputs are **different**,
- The output is TRUE if the inputs are the **same**

TRUTH TABLE

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

Algebraic expression : C = $\bar{A}$.B + A . $\bar{B}$



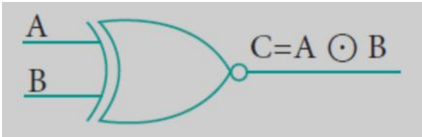
The output of the XNOR is NOT of XOR

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

In boolean algebra, ⊙ or "included dot" stands for the XNOR.

Therefore, C = A ⊙ B

Logic Symbol



17.Prove the following Absorption law by using Truth Table A + (A . B) = A

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

Hence, A + (A . B) = A is proved

18.Write De Morgan's laws

First Law : $\overline{A+B} = \bar{A}.\bar{B}$

Second Law : $\overline{A.B} = \bar{A} + \bar{B}$

19. Write the associative laws?

1)A + (B + C) = (A + B) + C

2) A . (B . C) = (A . B) . C

Theorems of Boolean Algebra

Identity
$A + 0 = A$
$A \cdot 1 = A$

Complement
$A + \bar{A} = 1$
$A \cdot \bar{A} = 0$

Commutative
$A + B = B + A$
$A \cdot B = B \cdot A$

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

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

Null Element
$A + 1 = 1$
$A \cdot 0 = 0$

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

Indempotence
$A + A = A$
$A . A = A$

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

3rd Distributive
$A + \bar{A} \cdot B = A + B$

De Morgan's
$\overline{A + B} = \bar{A} \cdot \bar{B}$
$\overline{(A \cdot B)} = \bar{A} + \bar{B}$