HSSLiVE.IN

Chapter 2: Data Representation and Boolean Algebra

Base of a number system: The number of symbols used in a number system is called base or **radix** of a number system

MSD and LSD: Left most digit of a number is MSD and right most digit of a number is LSD.

Different Number Systems:

Number System	Base	Symbols used
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Number Conversion procedure:

Conversion	Procedure
Decimal to Binary	Repeated division by 2 and grouping the remainders
Decimal to Octal	Repeated division by 8 and grouping the remainders
Decimal to Hexadecimal	Repeated division by 16 and grouping the remainders
Binary to Decimal	Multiply binary digit by place value(power of 2) and find their sum
Octal to Decimal	Multiply octal digit by place value (power of 8) and find their sum
Hexadecimal to Decimal	Multiply hexadecimal digit by place value (power of 16) and find their sum
Octal to Binary	Converting each octal digit to its 3 bit binary equivalent
Hexadecimal to Binary	Converting each hexadecimal digit to its 4 bit binary equivalent
Binary to Octal	Grouping binary digits to group of 3 bits from right to left
Binary to Hexadecimal	Grouping binary digits to group of 4 bits from right to left
Octal to Hexadecimal	Convert octal to binary and then binary to hexadecimal
Hexadecimal to Octal	Convert hexadecimal to binary and then binary to octal

Number representation methods: (i) Sign and magnitude representation (ii) 1's complement representation (iii) 2's complement representation.

ASCII: American Standard Code for Information Interchange. It uses 7 bits to represent a character in computer memory. It can represent only128 characters. Another version is ASCII-8 uses 8 bits for each character, can represent 256 different characters.

EBCDIC: Extended Binary Coded Decimal Interchange Code. This is similar to ASCII and is an 8 bit code. It can represent 256 characters.

ISCII: Indian Standard Code for Information Interchange or Indian Script Code for Information Interchange. It uses 8-bits and can represent various writing systems of India.

Unicode: Originally used 16 bits which can represent up to 65,536 characters. Nowadays Unicode uses more than 16 bits and hence it can represent more characters. Unicode can represent characters in almost all written languages of the world.

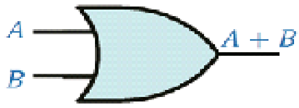
Boolean operations: The operations performed on the Boolean values 0s and 1s. The operations are OR (Logical Addition), AND (Logical Multiplication), NOT (Logical Negation).

Logic gate: It is a physical device that can perform logical operations on one or more logical inputs and produce a single logical output.

Truth Table: It is a table that shows Boolean operations and their results.

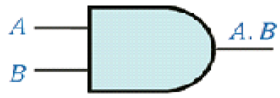
OR operation & Gate

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



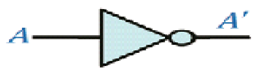
AND Operation & Gate

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



NOT Operation & Gate

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



Boolean Laws:

No.	Boolean Law	Statement 1	Statement 2
1	Additive Identity	$0 + X = X$	$1 + X = 1$
2	Multiplicative Identity	$0 . X = 0$	$1 . X = X$
3	Idempotent Law	$X + X = X$	$X . X = X$
4	Involution Law	$\bar{\bar{X}} = X$	
5	Complimentary Law	$X + \bar{X} = 1$	$X . \bar{X} = 0$
6	Commutative Law	$X + Y = Y + X$	$X . Y = Y . X$
7	Associative Law	$X + (Y + Z) = (X + Y) + Z$	$X . (Y . Z) = (X . Y) . Z$
8	Distributive Law	$X . (Y + Z) = X . Y + X . Z$	$X + (Y . Z) = (X+Y) . (X+Z)$
9	Absorption Law	$X + (X . Y) = X$	$X . (X + Y) = X$

De Morgan's theorems:

(i) $\overline{X+Y} = \bar{X} \cdot \bar{Y}$

(ii) $\overline{X.Y} = \bar{X} + \bar{Y}$

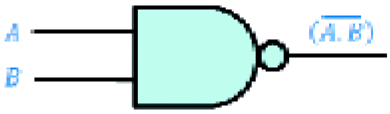
Universal Gates: The NAND and NOR gates are called universal gates. A universal gate is a gate which can implement any Boolean function without using any other gate type.

NAND: It is an inverted AND operation. NAND gate is an inverted AND gate.

NOR: It is an inverted OR operation. NOR gate is an inverted OR gate.

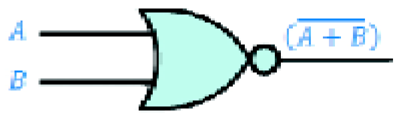
A	B	Y= $\overline{(A.B)}$
0	0	1
0	1	1
1	0	1
1	1	0

NAND operation and NAND gate

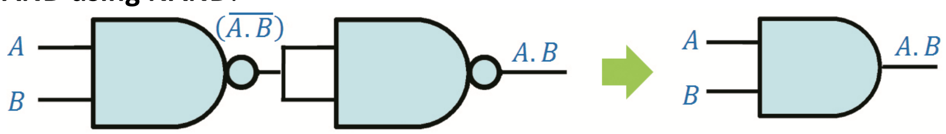


A	B	Y= $\overline{(A+B)}$
0	0	1
0	1	0
1	0	0
1	1	0

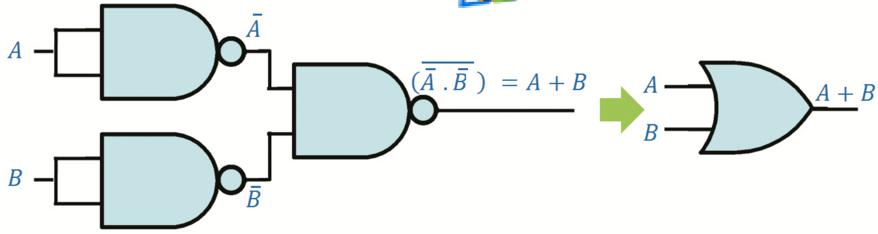
NOR operation and NOR gate



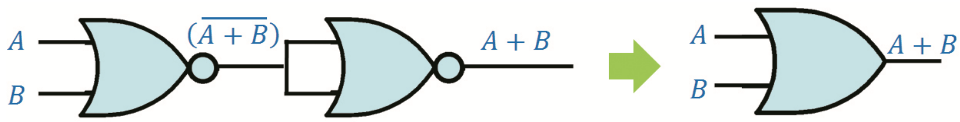
AND using NAND:



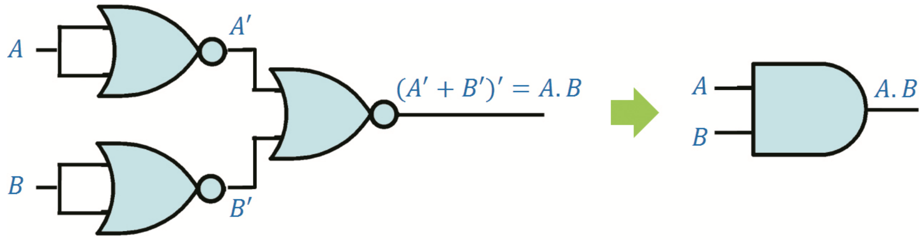
OR using NAND:



OR using NOR:



AND using NOR:



NOT using NAND and NOR: