Date
Page

Number System

A Number System is defined as a system of writing to express the number.

Types of Number System:

- * Positional Number System.
- * Non- Positional Number System.

There are four types of positional Number system:

- * Binary Number System.
- * Octal Number System.
- * Decimal Number System.
- * Hexadecimal Number System.

There are three types of Non-positional Number:

- * Sticks.
- * fingure.
- * Stone.
- * Soil.

Date
Page

1. Defination of Binary Number System?

It is a positional Number System whose base is 2 and it contain two digit that is 0 and 1.

Example: $(1011)_2$

2. Defination of Octal Number System?

It is a positional Number System whose base is 8 and it Contain 8 digits.

Example: (0,1,2,3,4,5,6,7)

: $(412)_8$

3. Defination of Decimal Number System?

It is a positional Number System whose base is 10 and it Contain 10 digits.

Example: $(482)_{10}$

4. Defination of Hexadecimal Number System?

It is a positional number system whose base is 16 and it Contain 16 digits.

Example: $(2AB)_{16}$

Note:

Here positional number System are the type of number system in which there is a fix position between digits. i.e. the value get change if the position get change.

Similarly, Non-positional number system are the type of number system in which there is no fix position among digits.

Noteswift

Binary Subtraction using Complement Method:

0 ⇒ 1
1 ⇒ 0

Binary Subtraction using $1^{'s}$ complement Method:

* Rules for making Binary Subtraction using $1^{'s}$ complement method:

- Make Both upper and Lower bit equal.
- Calculate the $1^{'s}$ complement of Lower bits.
- Add the 1st complement of Lower bits to upper bits.
- Check the extra bit. (Overflow Bits).

a. If there is Overflow bit, remove it and add it to the remaining portion.

b. If there is no Overflow bit, then the answer must be negative. Therefore Again Calculate the $1^{'s}$ complement of the portion.

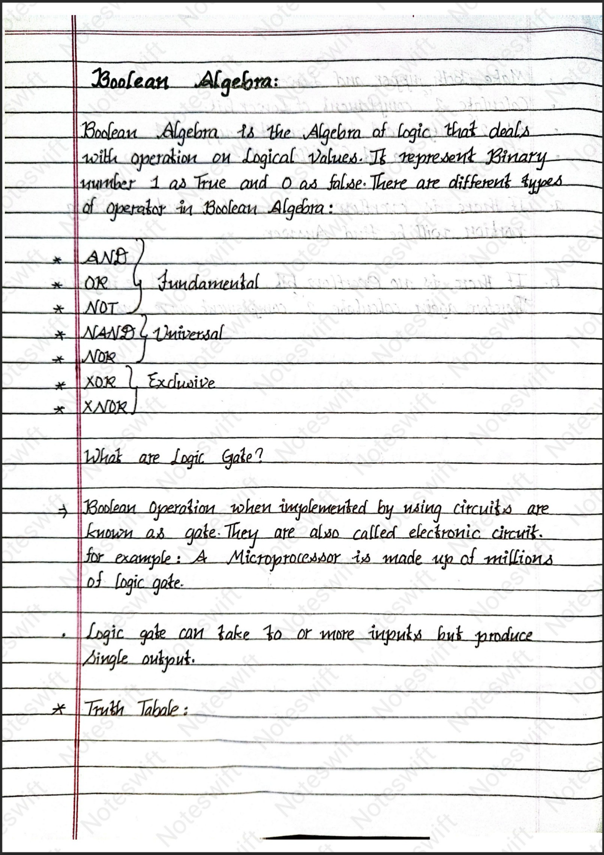
Binary Subtraction using $2^{'s}$ complement Method:

* Rules for making Binary Subtraction using $2^{'s}$ complement method:

- Make Both upper and Lower bit equal.
- Calculate $2^{'s}$ complement of Lower bit.
- Add $2^{'s}$ complement to Lower bit to upper bit.
- Check the extra bit. (overflow bit).

a. If there is overflow bit, removed it, and remaining portion will be find Answer.

b. If there is no Overflow bit, answer must be negative. Therefore again calculate $2^{'s}$ complement above Answer.



Boolean Algebra:

Boolean Algebra is the Algebra of logic that deals with operation on Logical Values. It represent Binary number 1 as True and 0 as false. There are different types of operator in Boolean Algebra:

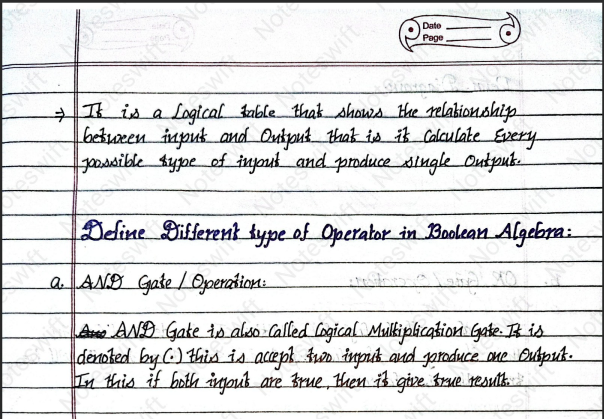
- * AND
- * OR } Fundamental
- * NOT
- * NAND } Universal
- * NOR
- * XOR } Exclusive
- * XNOR

What are Logic Gate?

→ Boolean Operation when implemented by using circuits are known as gate. They are also called electronic circuit. for example: A Microprocessor is made up of millions of logic gate.

• Logic gate can take to or more inputs but produce Single output.

* Truth Tabale:



Date
Page

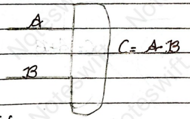
→ It is a Logical table that shows the relationship between input and Output that is it Calculate Every possible type of input and produce single Output.

Define Different type of Operator in Boolean Algebra:

a. AND Gate / Operation:

~~Ano~~ AND Gate is also Called Logical Multiplication Gate. It is denoted by (·) this is accept two input and produce one Output. In this if both input are true, then it give true result.

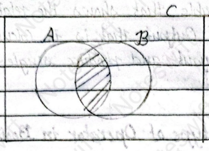
Symbol:



Truth Table:

Input		Output
A	B	C = A·B
0	0	0
0	1	0
1	0	0
1	1	1

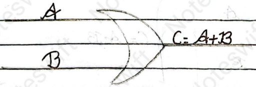
Venn-Diagram:



b. OR Gate / Operation:

→ It is also called Logical Addition. It is denoted (+) in this if any one of the input is true then result will be true, otherwise false.

Symbol:



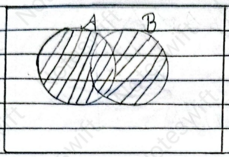
Truth Table:

Input		Output
A	B	C= A+B
0	0	0
0	1	1
1	0	1
1	1	1

Noteswift

Noteswift

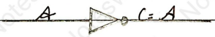
Venn diagram:



c. NOT Gate

This gate is also called complement gate. It is denoted by ($\bar{x}$) in this if input is true the result will be false and vice versa. It can only take single input and produce single Output.

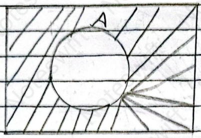
Symbol:



Truth Table:

Input	Output
A	$C = \bar{A}$
0	1
1	0

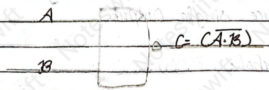
Venn Diagram:



d. NAND Gate.

It is Opposite of AND gate. In this if all the input are true then it gives false result otherwise true.

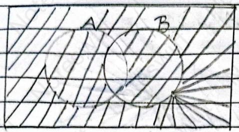
Symbol:



Truth Table:

Input		Output
A	B	C = ($\overline{A\cdot B}$)
0	0	1
0	1	1
1	0	1
1	1	0

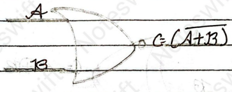
Venn-Diagram:



e. NOR Gate:

It is Opposite of OR Gate. In this if any one of the input is true then the result will false otherwise true.

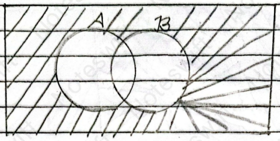
Symbol:



Truth Table:

Input		Output
A	B	C = ($\overline{A+B}$)
0	0	1
0	1	0
1	0	0
1	1	0

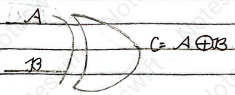
Venn Diagram.



7. XOR Gate:

This gate is called exclusive OR Gate. It produce true result when input are different and vice versa. It is denoted by (⊕).

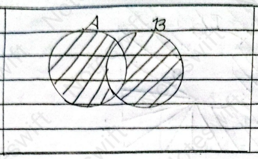
Symbol:



Truth Table:

Input		Output
A	B	C= A⊕B
0	0	1
0	1	1
1	0	1
1	1	0

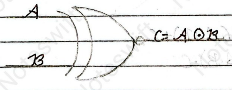
Venn Diagram:



9. XNOR Gate:

This gate is opposite of XOR Gate. It is denoted by (⊙). It gives true result when both input are same.

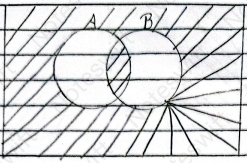
Symbol:



Truth Table:

Input		Output
A	B	$C = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Venn Diagram:



Laws of Boolean Algebra:

The basic laws and theorem of Boolean Algebra are:-

1. Assocative law:

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

2. Commutative law:

⇒ $A+B = B+A$
⇒ $A\cdot B = B\cdot A$

3. Distributive law:

⇒ $(A')' = A$

Date
Page

Noteswift

4. Identity Law:

→ A+0 = A	A·0 = 0
A+1 = 1	A·1 = A
$A+A' = 1$	$A \cdot A' = 0$
A+A = A	A·A = A

5. De-Morgan's Law:

→ $(A \cdot B)' = A' + B'$
→ $(A+B)' = A' \cdot B'$

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

A	B	A·B	(A·B)'	A'	B'	A' + B'
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

* $(A+B)' = A' \cdot B'$

A	B	A+B	(A+B)'	A'	B'	A'·B'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

* A(B+C) = AB+AC

A	B	C	B+C	A(B+C)	A·B	B·C	AB·AC
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	0	1	1
1	0	0	0	0	0	0	0
1	0	1	1	1	0	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Noteswift