QUESTION No. ______________ Exam No ______________ Date ______________

DO NOT WRITE ON THE MARGIN

DO NOT WRITE ON THE MARGIN

## LOGICAL EXPRESSION

A logical expression is a statement that can either be true or false. For example, $a<b$ ?.<? is a logical expression. It can be true or false depending on what values of a? and b? are given. [Note that this differs from a mathematical expression which denotes a truth statement. eg $a<b$?.<?]

Logical expressions form the basis of Computing. Boolean expression is the digital logic used to analyse gates and switching circuits such as those for the AND, OR, and NOT gate functions. Boolean functions are implemented by using logic gates. Logic gates are used to carry out logical operations on single or multiple binary inputs and give one binary output. In simple terms, logic gates are the electronic circuits in a digital system

## TYPES OF BASIC LOGIC GATES

there are several basic logic gates used in performing operations in digital systems. The common ones are

OR GATE , AND GATE, NOT GATE

XOR GATE

Additionally, these gates can also be found in a combination of one or two. Therefore, we get other gates, such as NAND Gate, NOR Gate, EXOR Gate and EXNOR Gate.

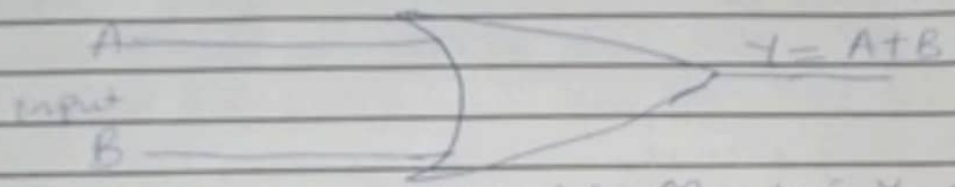
QUESTION No. ______________ Exam No ______________ Date __________

DO NOT WRITE ON THE MARGIN

DO NOT WRITE ON THE MARGIN

## OR GATE

In an OR gate, the output of an OR gate attains state 1 if one or more inputs attain state 1



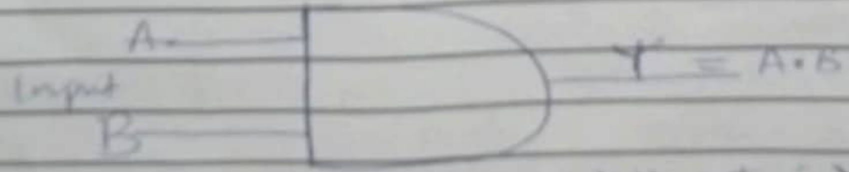
The Boolean expression of the OR gate is Y= A+B, reads as Y equals A 'OR' B.

The truth table of a two-input OR basic gate is given as

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## AND GATE

In the AND gate, the output of an AND gate attains state 1 if and only if all inputs are in state 1.



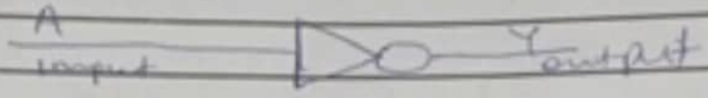
The Boolean expression of AND gate is Y=A•B

The truth table of a two-input AND basic gate is given as

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

## NOT GATE

In a NOT gate, the output of a NOT gate attains state 1 if and only if the input does not attain State 1.

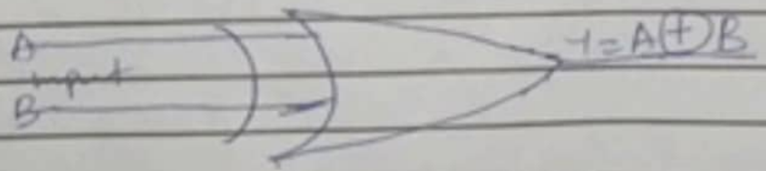


The Boolean expression is $A = \overline{A}$
it is read as Y equals NOT A.
the truth table of NOT gate is as follows

| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

## Exclusive - OR gate (XOR Gate)

In an XOR gate, the output of a two-input XOR gate attains state 1 if one adds only input and attains state 1



The Boolean expression of the XOR gate is
$A \cdot \bar{B} + \bar{A} \cdot B$ OR $Y = A \oplus B$
the truth table of an XOR gate is

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

QUESTION No. ______ Exam No ______ Date ______

DO NOT WRITE ON THE MARGIN

Assignment

Draw the diagram write the boolean expression and the truth table of the following gates

NOR, NAND, XNOR, EXOR AND EXNOR

## BOOLEAN FUNCTION REPRESENTATION

A Boolean expression is an expression which consists of variables, constants (0-false and 1-true) and logical operators which results in true or false. A Boolean function is an algebraic form of Boolean expression. A Boolean function of n-variables is represented by f(x1, x2, x3 . . . xn). By using Boolean laws and theorems, we can simplify the Boolean functions of digital circuits.

Different ways of representing a Boolean function are

- Sum-of-product (SOP) form
- Product-of-Sums (POS) form
- Canonical forms

There are two types of Canonical forms:

Sum-of-Min terms or Canonical SOP,

Product-of-Max terms or Canonical POS

Boolean expressions can be standardized by using theses two Standard forms.

SOP form – Sum of products form

POS form – Product of Sums form

Standardization of Boolean equations make the implementation, evolution and simplification easier and more systematic

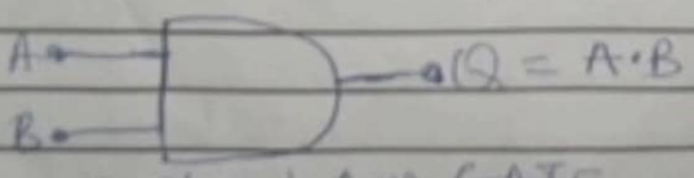
## Sum of Product (SoP) form

The Sum-of-Products (SoP) form is a method (or form) of Simplifying the Boolean expressions of logic gates. In this SoP form of Boolean function representation, the variables are operated by AND (Product) to form a product term and all these product terms are ORed (Summed or added) together to get the final function.

A Sum-of-products form can be formed by adding (or summing) two or more product terms using a Boolean addition operation. Here the product terms are defined by using the AND operation and the Sum term is defined by using OR operation.

In Boolean algebra, the multiplication of two integers is equivalent to the logical AND operation thereby producing a "Product" term when two or more input variables are "AND'ed" together. In other words, in Boolean algebra the AND function is the equivalent of multiplication and so its output state represents the product of its inputs.

AND GATE (Product)



2-Input AND GATE

Thus the Boolean equation for a 2-input AND gate is given as: $Q = A \cdot B$, that is Q equals both