**NAME: DHVANI PATEL**

**ROLL NO: 21BCP116**

**ASSIGNMENT 1**

**AIM: Implement and operate logic gate with its truth table.**

**Theory:**

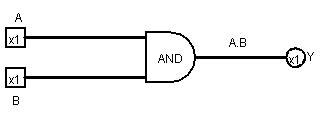
A logic gate is a digital gate that allows data to be transferred by some set of rules , where information flows based of that rules. The following type of logic gate are commonly used:

1. AND
2. OR
3. NOT
4. NOR
5. NAND
6. XOR
7. XNOR

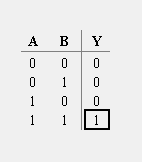
**AND Gate:** A high output results only if all the inputs to the AND gate are high.

Expression: Y= (A.B)

Circuit:



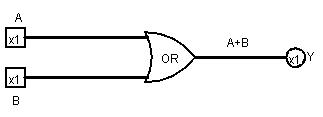
Truth Table:



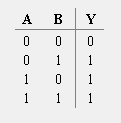
**OR Gate:** A high output results if one or both the inputs to the gate are high.

Expression: Y= (A + B)

Circuit:



Truth Table:



**NOT Gate:** A NOT gate is a logic gate that inverts the digital input signal.

Expression: Y=~(A)

Circuit:



Truth Table:

A Y

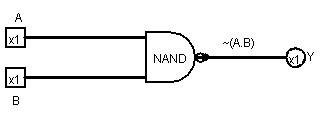
0 1

1 0

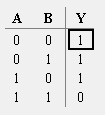
**NAND Gate:** A low input results only if all the inputs to NAND gate are high.

Expression: Y= ~(A.B)

Circuit:



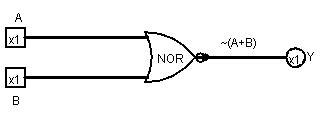
Truth Table:



**NOR Gate:** A high input results only if all the inputs to NOR gate are low.

Expression: Y= ~(A + B)

Circuit:



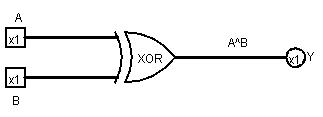
Truth Table:



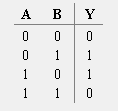
**XOR Gate:** XOR gate gives high output when the number of high inputs is odd.

Expression: Y= (A ^ B)

Circuit:



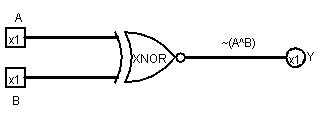
Truth Table:



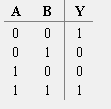
**XNOR Gate:** XNOR gate gives low output when the number of low inputs is odd.

Expression: Y= ~(A ^ B)

Circuit:



Truth Table:



**NAND Gate is also known as universal gate**

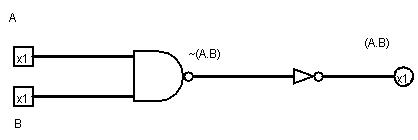
A universal gate is type gate where we can perform any Boolean expression with the presence of only that gate. We have two universal gate NAND and NOR gate which we use them in the circuits also they are easy to manufacture and use in industry as in IC circuits.

**AND Gate using NAND Gate**

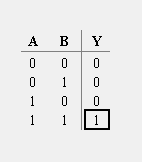
Expression:

Y= (A.B)

Circuit:



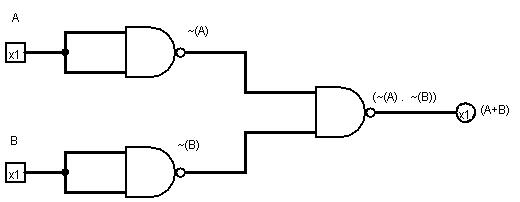
Truth Table:



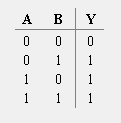
**OR Gate using NAND Gate**

Expression: Y= (A + B)

Circuit:



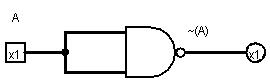
Truth Table:



**NOT Gate using NAND Gate**

Expression: Y=~(A)

Circuit:



Truth Table:

A Y

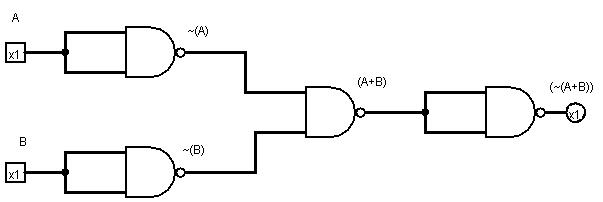
0 1

1 0

**NOR Gate using NAND Gate**

Expression: Y= ~(A + B)

Circuit:



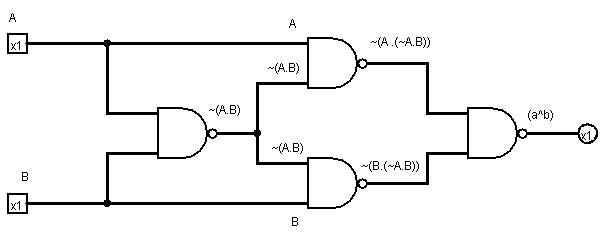
Truth Table:



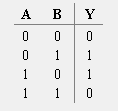
**XOR Gate using NAND Gate**

Expression : Y= (A ^ B)

Circuit:



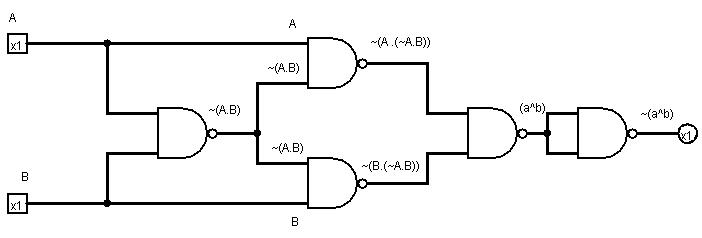
Truth Table:



**XNOR Gate using NAND Gate**

Expression: Y= ~ (A ^ B)

Circuit:



Truth Table:

