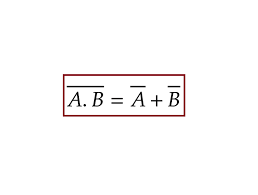
COMPUTER SYSTEM AND ORGANISATION  
(MODULE 6/6)  
  
BY  
Mrs. SUJATA PRADHAN  
PGT(SS),AECS,ANUPURAM

* **Logic Gates & Truth Tables**
* Negative AND & Negative OR gate
* DeMorgan’s First Theorem

According to **De Morgan’s theorem**, a **NAND** **gate** is equivalent to a **bubbled OR gate**. Therefore, the equation can be written as shown below.



The Boolean expressions for **NAND gate** is

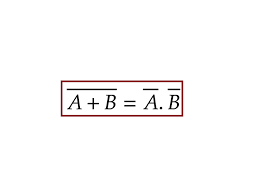
C:\Users\Sujatha\Desktop\demorgans-theorem-eq4-compressor.jpg

The Boolean expressions for the **bubbled OR** gate is C:\Users\Sujatha\Desktop\demorgans-theorem-eq5-compressor.jpg

As the **NAND** and **bubbled OR** gates are interchangeable, i.e., both gates have exactly identical outputs for the same set of inputs.

* **Evaluation of DeMorgan’sTheorem using Truth Table**
* Equality of NAND and Bubbled OR
* DeMorgan’s Second Theorem

According to DeMorgan’s theorem, a NOR gate is equivalent to a bubbled AND gate. Therefore, the equation can be written as shown below



The Boolean equation for the bubbled AND gate is C:\Users\Sujatha\Desktop\demorgans-theorem-eq2-compressor.jpg

The Boolean equation for NOR gate is C:\Users\Sujatha\Desktop\demorgans-theorem-eq1-compressor.jpg

As the **NOR** and **bubbled AND** gates are interchangeable, i.e., both gates have exactly identical outputs for the same set of inputs.

* **DeMorgan’sTheorem using Truth Table**
* Equality of NOR and Bubbled AND
* Boolean equations of De Morgan’s Theorem
* **Universal Gates**

In addition to **AND, OR, and NOT , fundamental gates**, other logic gates like **NAND and NOR** are also used in the design of digital circuits. A **universal gate** is a **gate** which can implement any Boolean function without use of any other **gate** type. **The NAND and NOR gates are universal gates.** In practice, this is advantageous since NAND and NOR **gates** are economical and easier to fabricate all IC digital logic families.

* **NAND Gate**: The NAND gate represents the **complement of the AND** operation. Its name is an abbreviation of **N**OT **AND**. The graphic symbol for the NAND gate consists of an **AND symbol with a bubble** on the output, denoting that a complement operation is perform.
* **NOR Gate:** The NOR gate represents the **complement of the OR** operation. Its name is an abbreviation of **N**OT **OR**. The graphic symbol for the NOR gate consists of an **OR symbol with a bubble** on the output, ed on the output of the AND gate.
* Universality of NAND Gates

**1. Implementing an Inverter (NOT gate) using only NAND Gate**

All NAND input pins connect to the input signal A gives an output A’.

**2. Implementing AND Using only NAND Gates**

**3. Implementing OR Using only NAND Gates**

Thus, the **NAND gate** is a universal gate since it can implement the **AND, OR and NOT.**

* Universality of NOR Gates
* **Implementing an Inverter Using only NOR Gate:** All NOR input pins connect to the input signal A gives an output A’.
* **Implementing OR Using only NOR Gates:** An OR gate can be replaced by NOR gates as shown in the figure (The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter)
* **Implementing AND Using only NOR Gates:** An AND gate can be replaced by NOR gates as shown in the figure (The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters)

Thus, the NOR gate is a universal gate since it can implement the AND, OR and NOT functions

* Boolean Equation of XOR and XNOR Gates

Boolean Equation of **XOR**

**F(A,B)=A’.B+A.B’**

Boolean Equation of **XNOR**

**F(A,B)=A.B+A’.B’**

Like truth tables,it can be proved algebraically (**XOR)’=XNOR**

F(A,B)= **(A’.B+A.B’)’**

=(A’.B)’ . (A.B’)’ (De Morgan’s Theorem)

=((A’)’ + B’) . (A’ + (B’)’) (De Morgan’s Theorem)

=(A + B’) . (A’ + B) ((A’)’=A)

=A.A’ + A.B +B’.A’ + B’.B (A.A’=B.B’=0)

=**A.B + A’.B’**

Hence proved

* Simplification of Boolean Function
* Minimization of Boolean Functions

we need to apply the **rules of Boolean algebra** to reduce the expression to its simplest form.

Minimized version of the Boolean Expression needs less number of logic gates and also reduces the complexity of the circuit substantially. Hence minimization is important to find the most economic equivalent representation of a boolean function. The component reduction results in higher operating speed, less power consumption, less cost, and greater reliability.

**Example:**

**F(X,Y,Z) = X’ Y’ Z + X’ Y Z + X Y’**

**= X’ Z ( Y’ + Y ) + X Y’**

**= X’ Z + X Y’ (Y’ + Y = 1)**

**= X Y’ + X’ Z**

* Boolean equation from a Circuit
* **Laws of Boolean Algebra**
* Simplification of Boolean Expressions
* SUMMARY
* De Morgan’s Theorem
* Negative AND & Negative OR Gates
* Universality of NAND and NOR Gates
* Boolean Equation from a circuit
* Simplification of Boolean Functions