**Experiment 2**

**Verify the truth table of Logic Gates NAND and XOR**

**Objective:**

In this lab students will verify

* The truth table of NAND gate
* The truth table of XOR gate

**Components**

* IC74LS00×1
* IC74LS86×1
* AM2000 TRAINER
* Multi-meter
* Cutter
* Single core wire
* Pair of Pliers

**Task 1:**

**To check the operation of NAND gate according to the NAND’s truth table using 74LS00IC.**

***NAND Gate***

The NAND or “Not AND” function is a combination of the two separate logical functions, the AND function and the NOT function in series. The logic NAND function can be expressed by the Boolean expression of, $\overbar{A.B}$.



**Figure 1Two input NAND gate**

The **Logic NAND Function** only produces an output when “ANY” of its inputs are not present and in Boolean Algebra terms the output will be TRUE only when any of its inputs are FALSE.



**Figure 2 NAND gate IC 74LS00**

**Procedure:**

1. Connect the AM2000 trainer to the 220V AC power supply
2. Turn on the trainer and verify the voltage of the power supply using the multimeter. It should be +5V exactly.
3. Install IC74LS00 on the trainer’s board.
4. Wire the circuit according to the diagram by consulting NAND gate IC’s diagram as shown in the Figure 2.
5. Use any of the two logic switches from S2 to S9 for inputs A and B respectively.
6. For output indication use any of the LED’s from L0 to L15.
7. Supply the +5v and GND to the pins 14 and 7 of the IC respectively.
8. Test all the possible combinations of inputs and verify the output according to the truth table of NAND gate.
9. Make truth table according to the results.
10. Similarly replace the IC for checking XOR gate operation.

**In case of trouble:**

1. Check the power supply.
2. Check the Vcc and GND at pins 14 and 7 respectively.
3. Check all the wire connections.
4. Check the circuit wiring and remove the breaks.
5. Check the IC using truth table.

**Truth Table of NAND gate (2)**

|  |  |  |
| --- | --- | --- |
| **Input (A)** | **Input (B)**  | **Output (Q=**$\overbar{A.B}$**)** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Task 2:**

**To check the operation of XOR gate according to the XOR’s truth table using 74LS86 IC.**

***XOR gate***

An XOR gate (sometimes referred to by its extended name, Exclusive OR gate) is a digital logic gate with two or more inputs and one output that performs exclusive disjunction. The output of an XOR gate is true only when exactly one of its inputs is true. If both of an XOR gate's inputs are false, or if both of its inputs are true, then the output of the XOR gate is false.



****

**Figure 3 Two inputs XOR gate**

****

**Figure 4 XOR gate IC 74LS86**

**Truth Table of XOR gate (2)**

|  |  |  |
| --- | --- | --- |
| **Input (A)** | **Input (B)**  | **Output (Q=**$\overbar{A}B+A\overbar{B}$**)** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Exercise**

Give the switch representation of NAND function. (2)

How does XOR gate works if it has more than two inputs? (2)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Conclusion (2)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_