**EEE-1212:Digital Logic Design Lab**

1st Year 2nd Semester

Session: 2015-2016

**Experiment Number:** 01

**Name of the Experiment:**

(a)Verification of AND,OR,NOT,NAND and NOR Gates

(b) Verification of the universality of NAND and NOR gates

**Submitted by:**

**Group:** 3

Nusrat Munia

Roll: SK-03

Palash Roy

Roll: JH-24

Abdullahil Baki Arif

Roll: SH-36

**Prepared by:**

Nusrat Munia

Roll: SK-03

**Experiment Date**: 31st July 2016

**Submission Date**: 7th August 2016

**Submitted to:**

1. Dr.Suraiya Pervin, Professor, Dept of CSE,DU

2. Mr. Abu Ahmed Ferdaus, Associate Professor, Dept. of CSE, DU

**Name of the Experiment:**

1(a). Verification of AND,OR,NOT,NAND and NOR Gates

1(b). Verification of the universality of NAND and NOR gates

**Objective:**

1. The objective for this lab is to understand the fundamentals of logic gates. To study the truth tables of various basic logic gates. To verify the truth table of AND gate, OR gate, NOR gate, NAND gate & NOT gate using NAND & NOR gate
2. To verify the truth table of AND gate, OR gate & NOT gate using NAND & NOR gate to verify the universality of NAND and NOR gates.

**Theory:**

 Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/ output combination is called Truth Table. Various gates and their working are explained here.

**AND Gate:** The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high otherwise it gives low output(0).  A dot (.) is used to show the AND operation i.e. A.B.

**OR Gate:** The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high otherwise it gives low output(0).  A plus (+) is used to show the OR operation.

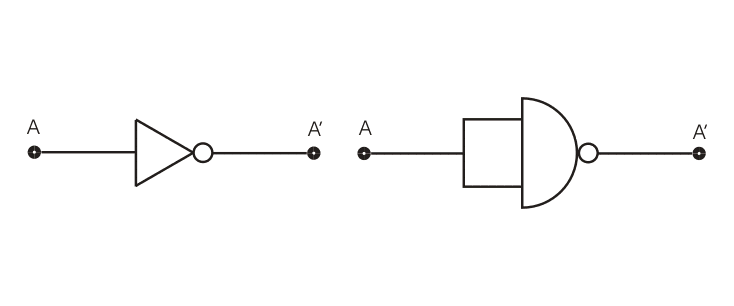
**NOT Gate:** The NOT gate is an electronic circuit that produces an inverted version of the input at its output.  It is also known as an inverter.  If the input variable is A, the output is known as NOT A.  That means if it’s input is 1 it’s output is 0 ,if it’s input is 0 it’s output is 1.

**NAND Gate:** The NAND gate is an electric circuit that gives a low output (0) if it’s all input is high (1) otherwise it gives high output(1). Actually it operates like an AND Gate followed by an inverter.

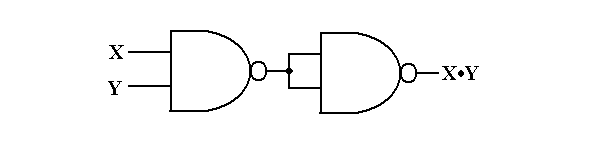
**NOR Gate**: The NOR gate is an electric circuit that gives a high output (1) if it’s all input is low (0) otherwise it gives low output(0). Actually it operates like a OR Gate followed by an inverter.

NAND and NOR gate are called universal gates, using NAND only or NOR only any logic function can be implemented.

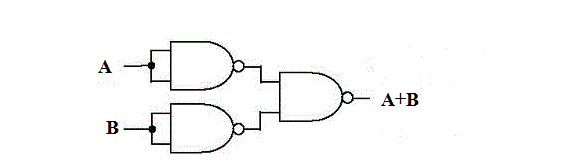
Implementation of NOT gate using NAND gate:

****

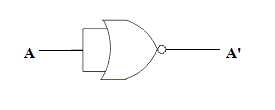
Implementation of AND gate using NAND gate:



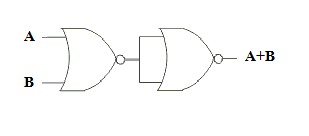
Implementation of OR gate using NAND gate:



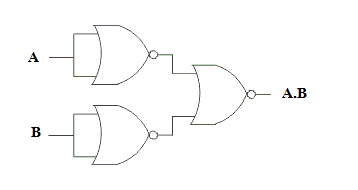
Implementation of NOT gate using NOR gate:

****

Implementation of OR gate using NOR gate:

****

Implementation of AND gate using NOR gate:



**Instruments:**

* Trainer Board
* IC(s) IC-7408,IC-7432,IC-7404,IC-7402,IC-7400
* Connecting wires

**Procedure: (a)**

1. At first we placed the integrated circuits with AND, OR, and NOT gates on a breadboard properly. All of these components is placed across the gap in the center of the breadboard.
2. Then we connected the inputs of any one logic gate to the logic sources and its output to the logic indicator.
3. And constructed the circuits, one at a time, on a breadboard with switches and an LED. Experimentally verified the truth tables for NOT, AND, and OR gates. A and B are switches. The output for each logic gate will be on an LED. (LED Off = 0, LED On = 1).
4. Gave biasing to the ICs with the VCC(5 volt) and GND(0 volt), and do necessary connections according to the circuit diagram shown below.
5. Various input combinations and observe output for each one is applied.
6. The truth table for each input/ output combination is verified.

**Procedure: (b)**

1. At first we placed the integrated circuits with NAND, or NOR gates on a breadboard properly. All of these components is placed across the gap in the center of the breadboard.
2. Gave biasing to ICs with the VCC(5 volt) and GND(0 volt), and do necessary connections according to the circuit diagram shown below.
3. Experimentally verified the truth tables for NOT, AND, and OR gates.

**Result:**

**IC and Truth Table:**

**IC NO: 7408**

**Logic Gate: Truth Table for AND gate:**

|  |  |  |
| --- | --- | --- |
| A | B | Q=A.B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

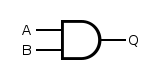
****

Fig: AND Gate

**IC N0: 7432**

**Logic Gate: Truth Table for OR gate:**

|  |  |  |
| --- | --- | --- |
| A | B | Q=A+B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

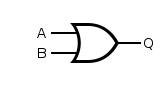
****

Fig: OR Gate

**IC N0: 7404**

**Logic Gate: Truth Table for NOT gate :**

|  |  |
| --- | --- |
| A | Q=A’ |
| 0 | 1 |
| 1 | 0 |

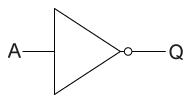
****

Fig: NOT Gate

**IC N0: 7400**

**Logic Gate: Truth Table for NAND gate:**

|  |  |  |
| --- | --- | --- |
| A | B | Q |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**and 1.PNG**

Fig: NAND Gate

**IC N0: 7402**

**Logic Gate: Truth Table for NOR gate:**

|  |  |  |
| --- | --- | --- |
| A | B | Q=A+B |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

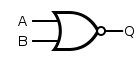
****

Fig: NOR Gate

**Discussion:**

In this experiment we worked with IC’s and verified the truth table of various gates. As IC is a very small component we got troubled to find the notch of the IC which indicates the start pin of the IC. When we inserted the IC on breadboard it was not working at first because there was problem in connection. During the experiment at we also faced some technical difficulties. The breadboard we took was not working properly. On the logic indicator sector an indicator light and on the logic source sector some switches were not functioning accurately. But we figured them out and completed the experiment.