

**Faculty of Engineering and Technology**

**Electrical and Computer Engineering Department**

**DIGITAL ELECTRONICS AND COMPUTER**

**ORGANIZATION LABORATORY**

**ENCS2110**

**Experiment No. 1**

**Combinational Logic Circuits**

Prepared by:

**Student Name Student NO.**

Partners:

**PARTNER1\_NAME Number**

**PARTNER2\_NAME Number**

Instructor:

Teaching assistant:

Section:

Date:

Abstract  
  
In this experiment, we will review our knowledge of digital systems in general, and in combinational logic circuits especially, such as AND, OR, NOT, NAND, NOR, and OR operations and their implementation and their Truth tables, and how to implement Boolean functions using NAND gate only or NOR gate only. In addition to minimization techniques by K-map for optimal description for any logic circuit. In the practical part, we will learn techniques of the solution to logic design problems practically so, we will construct these gates using NAND or NOR gates. Finally, we will construct an AOI gate with basic gates.   
  
At the end of this lab, my theoretical background will be proved practically so, I will be able to construct basic gates using NAND or NOR gates, in addition construct an AOI gate with basic gates.

# Table of Contents

[**Abstract I**](#_Toc132547568)

[**Table of Contents II**](#_Toc132547569)

[**Table of Figures III**](#_Toc132547570)

[**List of Tables IV**](#_Toc132547571)

[**1. Theory 1**](#_Toc132547572)

[**1.1 Basic logic gates: 1**](#_Toc132547573)

[**1.1.1 AND GATE: 1**](#_Toc132547574)

[**1.2 Universal Logic Gates: 1**](#_Toc132547575)

[**1.2.1 NAND GATE: 2**](#_Toc132547576)

[**2. Procedure and Discussion: 3**](#_Toc132547577)

[**2.1 The characteristic of NOR gate: 3**](#_Toc132547578)

[**2.1.1 Construct NOR gate: 3**](#_Toc132547579)

[**2.1.2 Construct Buffer gate from NOR gate: 4**](#_Toc132547580)

[**Conclusion: 6**](#_Toc132547581)

[**References: 7**](#_Toc132547582)

# Table of Figures

[Figure ‎1.1: Two-input AND Symbol 1](#_Toc132548651)

[Figure ‎1.2: Two-input NAND Symbol 2](#_Toc132548652)

[Figure ‎2.1: IT-3002 NOR Gate Block 3](#_Toc132548653)

[Figure ‎2.2: Build buffer using NOR gate 4](#_Toc132548654)

# List of Tables

[Table ‎1.1: Two-input AND Truth Table 1](#_Toc132548601)

[Table ‎1.2: Two-input NAND Truth Table 2](#_Toc132548602)

[Table ‎2.1: Results of NOR gate implementation 3](#_Toc132548603)

[Table ‎2.2: Results of Buffer gate implementation using NOR gate 4](#_Toc132548604)

# Theory

## 1.1 Basic logic gates:

"Basic Logic Gates are the fundamental logic gates using which universal logic gates and other logic gates are constructed. These gates are associative and commutative in nature. AND, OR, and NOT in the famous examples of basic logic gates." [1].

### 1.1.1 AND GATE:

It is the basic gate that performs the logical multiplication which known by AND function. The output of AND gate will be HIGH when it's all inputs are HIGH and LOW otherwise [1]. Figure 1.1 shows a symbol of 2 input AND gate, and Table 1.1 shows the truth table of AND function:

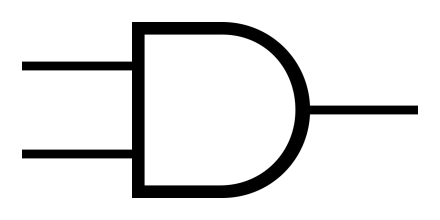


Figure ..1: Two-input AND Symbol. (Source: Wikipedia, online)

Table ..1: Two-input AND Truth Table

|  |  |  |
| --- | --- | --- |
| A | B | Output |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## 1.2 Universal Logic Gates:

"Universal logic gates are those capable of implementing any Boolean function without requiring any other type of gate. We called these gates Universal gates because Universal gates are not associative in nature, but they are commutative in nature. There are two universal logic gates NAND gate and NOR gates." [1].

### 1.2.1 NAND GATE:

It is the abbreviation AND-NOT, its output will be LOW only if it's all inputs are LOW, and HIGH otherwise [1]. Figure 1.2 shows a symbol of 2 input NAND gate, and Table 1.2 shows the truth table of NAND function:

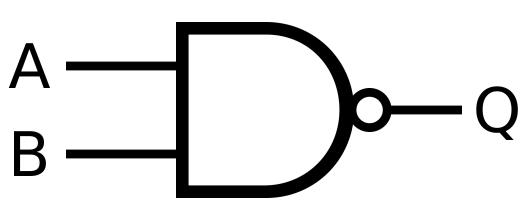


Figure .: Two-input NAND Symbol. (Source: Wikipedia, online)

Table .: Two-input NAND Truth Table

|  |  |  |
| --- | --- | --- |
| A | B | Output |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Procedure and Discussion:

## 2.1 The characteristic of NOR gate:

### 2.1.1 Construct NOR gate:

#### 2.1.1.1 Connection:

**We connected the circuit as it shown in the Figure 2.1, and discussed in the following points:**

****

Figure .: IT-3002 NOR Gate Block (Source: Lab Manual)

1. We Set the module IT-3002 and located block NOR gate as shown in Figure 2.1.
2. We Connected the +5V of module IT-3002 to the +5V output of the fixed power supply IT-3000 And do the same for ground (GND).
3. Then we selected the first gate U6 in Figure 2.1, and Connected inputs A and B with Data switch TTL level in power supply SW0 and SW1. Finally, we connected output F1 with Logic Indicator (LED) L0 in the power supply. The results are shown in Table 2.1.

#### 2.1.1.2 Results:

Table .: Results of NOR gate implementation

|  |  |  |
| --- | --- | --- |
| SW0(A) | SW1(B) | L0 (F1) |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

#### 2.1.1.3 Discussion:

As the results show, NOR function gives HIGH output only when all its inputs are LOW, and LOW otherwise. As the truth table shows if one of the switches has been fixed on the LOW state, the result will be the inverse of the other switch state, so this is a method to construct the NOT gate from NOR gate.

### 2.1.2 Construct Buffer gate from NOR gate:

#### 2.1.2.1 Connection

**We connected the circuit as it shown in the Figure 2.2, and discussed in the following points:**

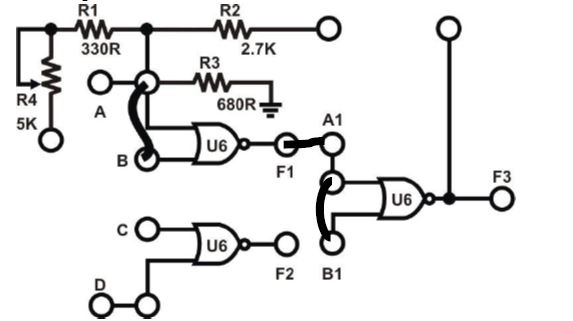
****

Figure .: Build buffer using NOR gate (Source: Lab Manual)

1. We Used two of U6 to construct a buffer as shown in Figure 2.2. we Inserted the connection clips between A and B, and connected F1 with A1 then we connected A1 with B1. We finally Connected input A to SW0 and output F3 to L1. The results are shown in Table 2.2.

#### 2.1.2.2 Results

Table .: Results of Buffer gate implementation using NOR gate

|  |  |
| --- | --- |
| SW0(A) | L1 (F3) |
| 0 | 0 |
| 1 | 1 |

#### 2.1.2.3 Discussion

As the results show, a buffer gate can be constructed from two NOR gates, by connecting single input to the two inputs of the NOR, so NOR gate will act as an inverter, then connect the previous result to the two inputs of NOR then the inverted result will be inverted again so it will give the original value.

# 

# Conclusion:

By completing this experiment all the objectives are obtained. Now, I can construct all digital basic gates using universal gates which are NAND and NOR, such as constructing AND from NOR using De Morgan low. NOT gate can be constructed in two ways, the first is to fix one of the inputs on 0 in the NOR case and on 1 in the NAND case. The second one is to connect single input to the two NAND/NOR input bins. If the last case is used twice in a cascading manner we will get a buffer gate. Finally, I known a new gate that I didn't know before which is the AOI gate, and how to build it using basic gates.

# References:

[1]: Manual for Digital Electronics and Computer Organization Lab, 2023, Birzeit University.

[2]: Wikipedia. [online image]

[Accessed on 16th April 2023]

<https://en.wikipedia.org/wiki/AND_gate>

[3]: Wikipedia. [online image]

[Accessed on 16th April 2023]

<https://en.wikipedia.org/wiki/NAND_gate>

# 