***Lab 2***

***Second stage***

***Medical Physical Department***

***Digital Electronics***

**Lab2: Basic logic Gates (AND, OR, and NOT gates)**

**By**

***Asst. Prof. Dr. Mehdi Ebady Manaa***

# Basic logic Gates (AND, OR, and NOT gates)

## Objectives

1- To study and understand the 3 basic gates. 2- Implement the basic gate in EWB.

3- The study the specifications of every gate when connected it with one input constant and the other is variable.

### AND and NAND gates

This gate gives high output (1) if all the inputs are 1’s. otherwise the output will be low (0).

Its Boolean algebra representation is: **C=A.B**

And it’s truth table and schema as following:



|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

The NAND gate works opposite to the AND gate. Its Boolean algebra representation is: **C=(A.B)'** And it’s truth table and schema as following:



|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

### OR and NOR gates

This circuit will give high output (1) if any input is high (1).

Its Boolean algebra representation is: C=A+B

and it’s truth table and schema as following:



|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

The NOR gate works opposite to the OR gate. Its Boolean algebra representation is: **C=(A+B)'** And it’s truth table and schema as following:



|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

### NOT gate

This is the simplest gate it just inverts the input, if the input is high the output will be low and conversely.

So B=A’

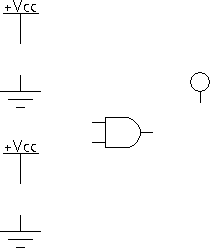


|  |  |
| --- | --- |
| **A** | **B** |
| 0 | 1 |
| 1 | 0 |

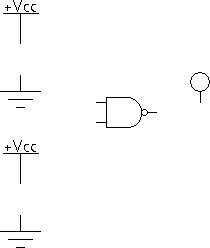
## Lab Tasks

### Task 1: The AND and NAND gates

In EWB, draw the following two circuits and fill the truth table below

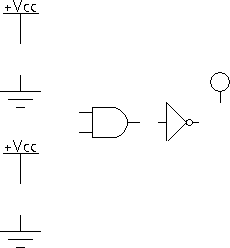


|  |  |  |  |
| --- | --- | --- | --- |
| A | B | A.B | (A.B)' |
| 0 | 0 |  |  |
| 0 | 1 |  |  |
| 1 | 0 |  |  |
| 1 | 1 |  |  |



### Task 2: The AND-NOT combination

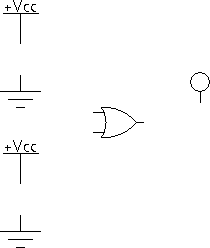
In EWB, draw the following circuit and fill the truth table



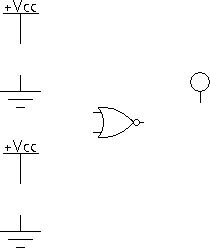
|  |  |  |
| --- | --- | --- |
| A | B | (A.B)’ |
| 0 | 0 |  |
| 0 | 1 |  |
| 1 | 0 |  |
| 1 | 1 |  |

### Task 3: The OR and NOR gates

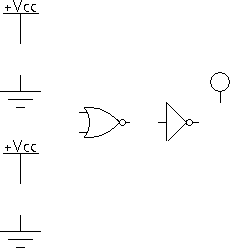
In EWB, draw the following two circuits and fill the truth table below



|  |  |  |  |
| --- | --- | --- | --- |
| A | B | A+B | (A+B)’ |
| 0 | 0 |  |  |
| 0 | 1 |  |  |
| 1 | 0 |  |  |
| 1 | 1 |  |  |



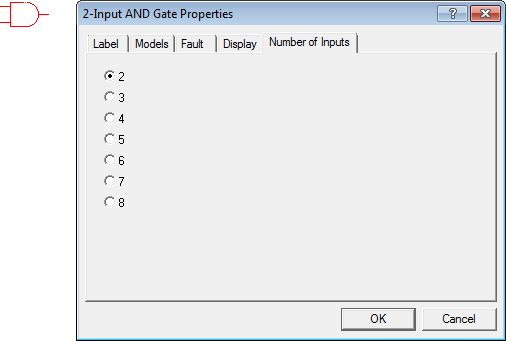
### Task 4: The NOR-NOT combination

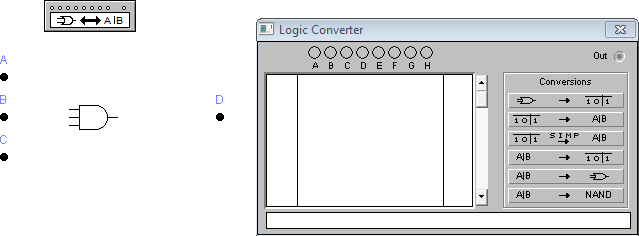


|  |  |  |
| --- | --- | --- |
| A | B | ((A+B)’)’ |
| 0 | 0 |  |
| 0 | 1 |  |
| 1 | 0 |  |
| 1 | 1 |  |

### Task 5: Finding the truth table of a three input gate using the logic converter

Repeat what you did in task 5 for a three-input AND gate. Show your connections in the circuit below. **Note**: you can obtain a three-input AND gate by drawing a regular two-input AND gate and then changing its *Number of Inputs* property as shown next.





|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **D** |
| 0 | 0 | 0 |  |
| 0 | 0 | 1 |  |
| 0 | 1 | 0 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 0 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |
| 1 | 1 | 1 |  |