|  |  |  |
| --- | --- | --- |
| **American University of Sharjah**  **College of Engineering**  Dept of Computer Science & Engg  P. O. Box 26666  Sharjah, UAE | A picture containing logo  Description automatically generated | **Lab Instructor:** Eng. Donthi Sankalpa  **Office:** ESB-1036C  **Phone**: 971-6-5154826  **e-mail**: [dsankalpa@aus.edu](mailto:dsankalpa@aus.edu)  **Semester**: Fall 2023 |

Lab #3 Simplification of Boolean Functions using Boolean Algebra and Universal gates

**Objectives**

* Use of switches as inputs and LEDs as outputs.
* Understand and test operations of basic logic gates.
* Convert logic designs to universal gates designs (all NAND or all NOR).
* Study the representation of functions using truth tables, logic diagrams and Boolean algebra.
* Simplify and modify Boolean logic functions using postulates and theorems of Boolean algebra including De Morgan laws.
* Implement SOPs using standard logic gates.

**Lab Equipment and Circuit Components**

* 1. **Equipment** 
     + ELVIS III Board
  2. **Circuit Components**
     + 7400 NAND
     + 7402 NOR
     + 7404 inverter
     + 7408 AND
     + 7432 OR

**Boolean Theorems**

The following table describes the various postulates and theorems that can be applied on complex Boolean expressions to simplify them so that we use the most minimum amount of gates for our final output.

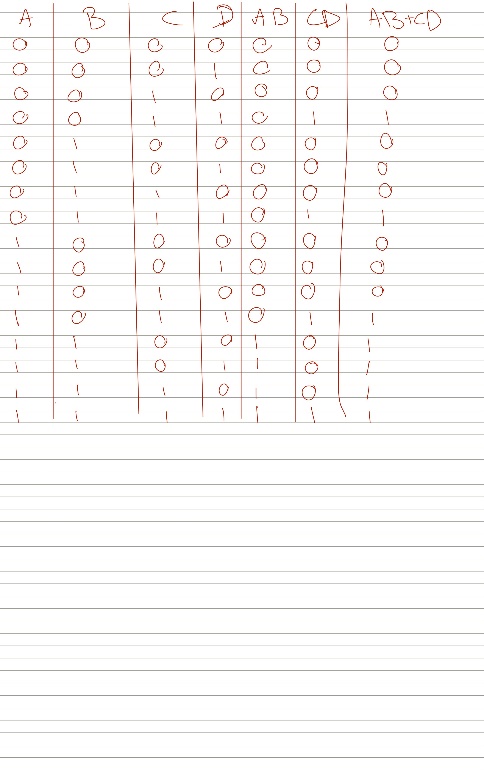
**Table

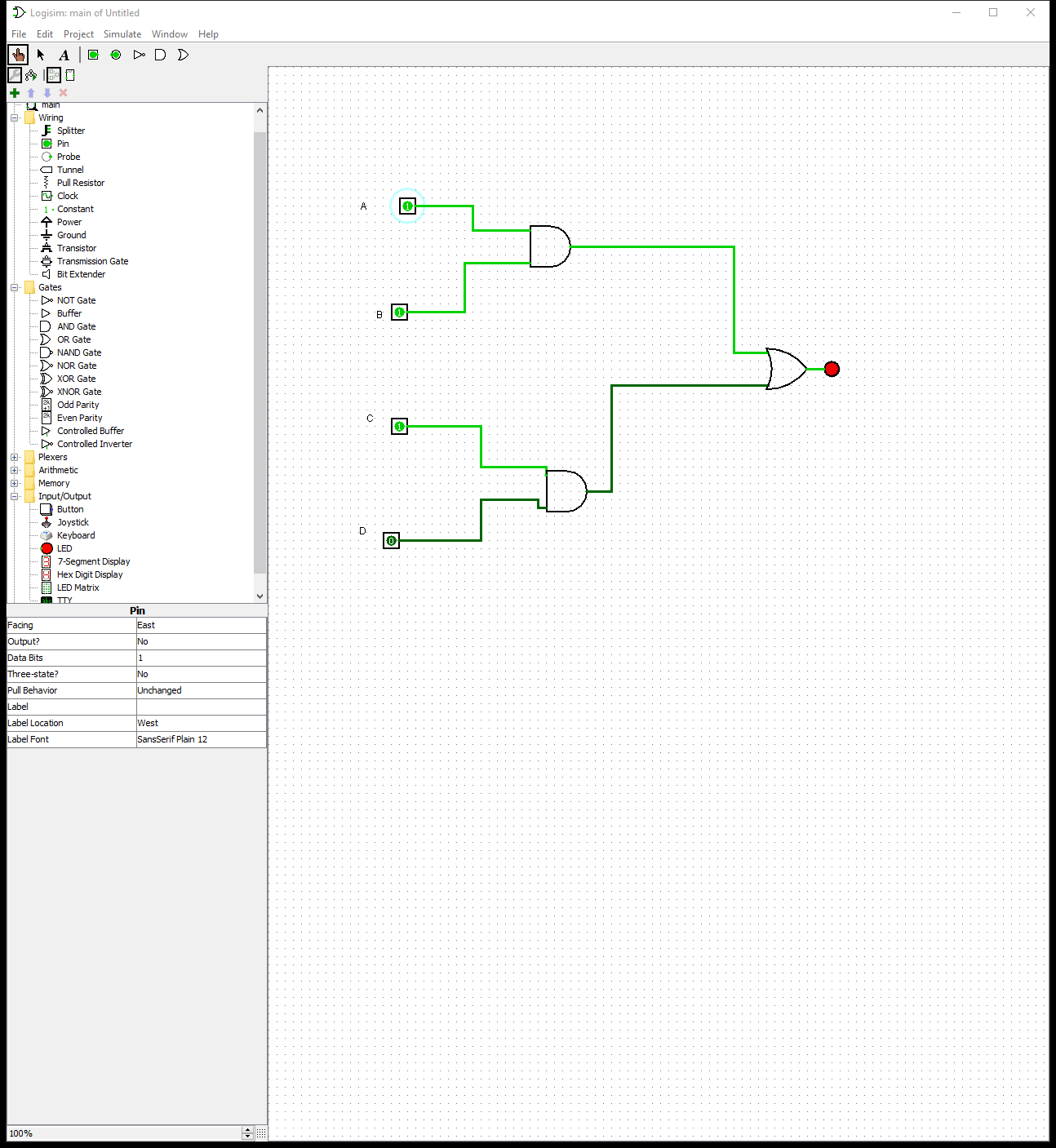
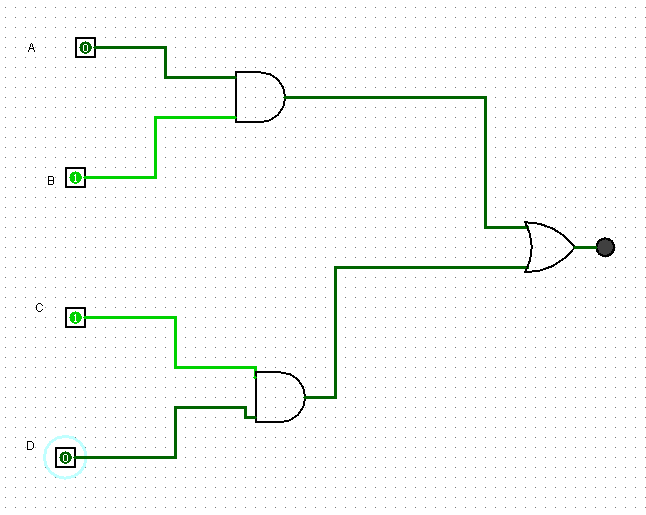
Description automatically generated**

**Lab Assignment**

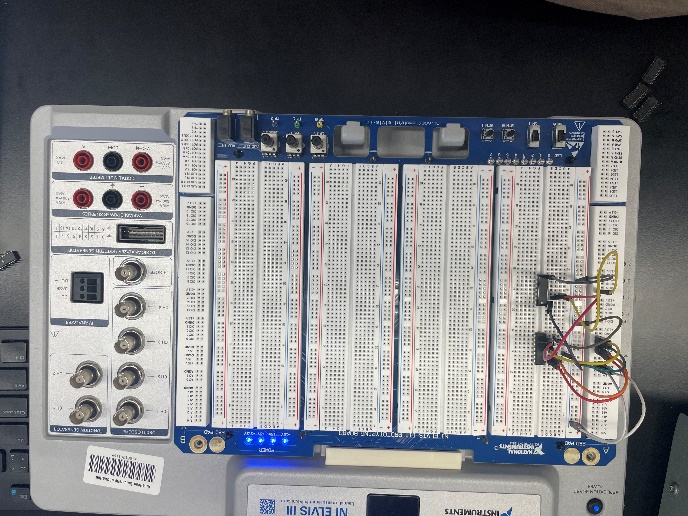
**Exercise 1**

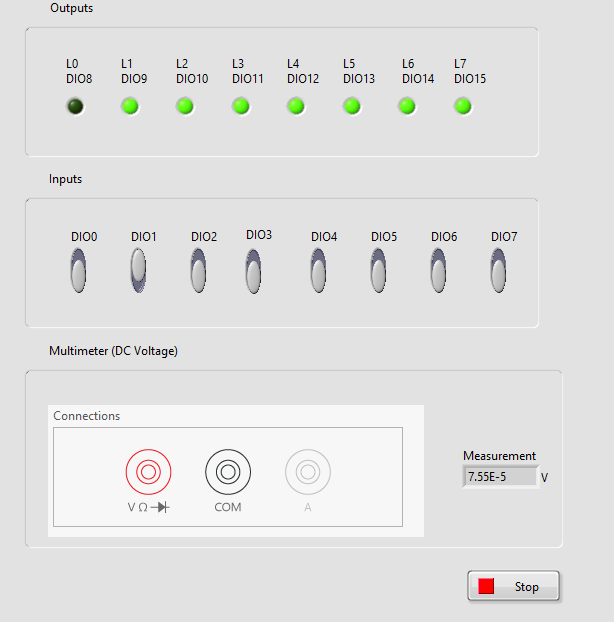
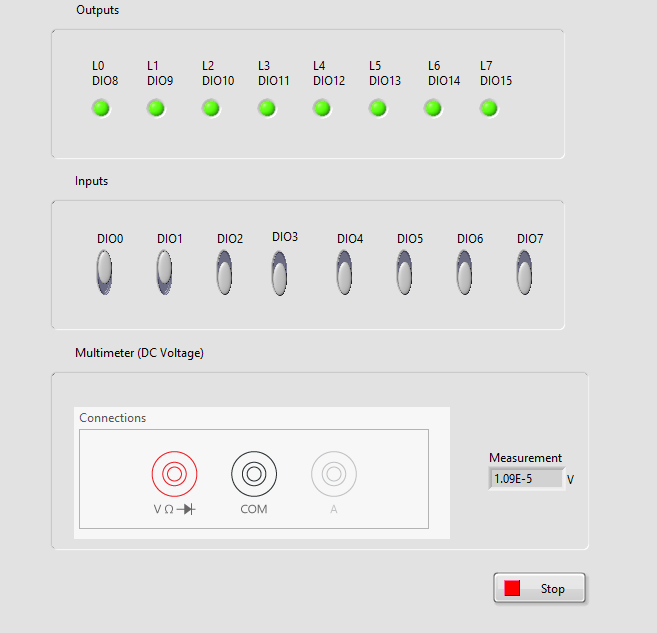
Simplify the expression and demonstrate the following:

1. Build the Truth Table. 
2. Simulate the simplified equation circuit. AB + CD



1. Implement the circuit and verify your implementation.





1. Compare the number of gates used to implement the expression before the simplification and the number of gates used in this lab after the simplification. Before simplification there was 13 gates and after is 3 gates
2. Compare the number of ICs used to implement the expression before the simplification and the number of ICs used in this lab after the simplification. Before we need 5 and now, we need 2

**Exercise 2**

Design a logic circuit to produce HIGH output only if the input, represented by a 3-bit binary number, is greater than five or less than two. (8 inputs)

1. Build the Truth Table

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

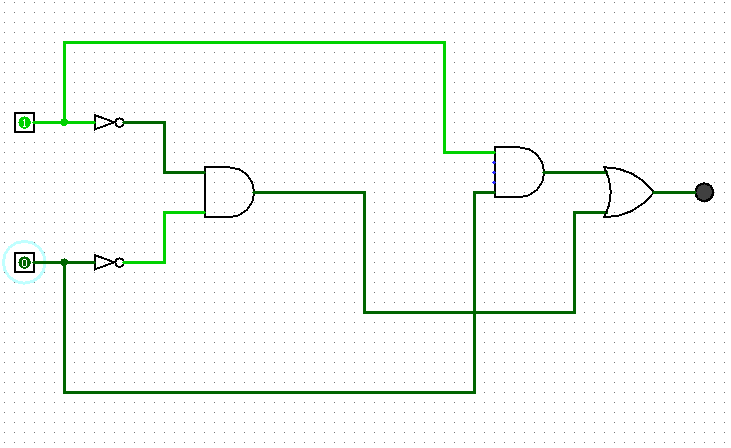
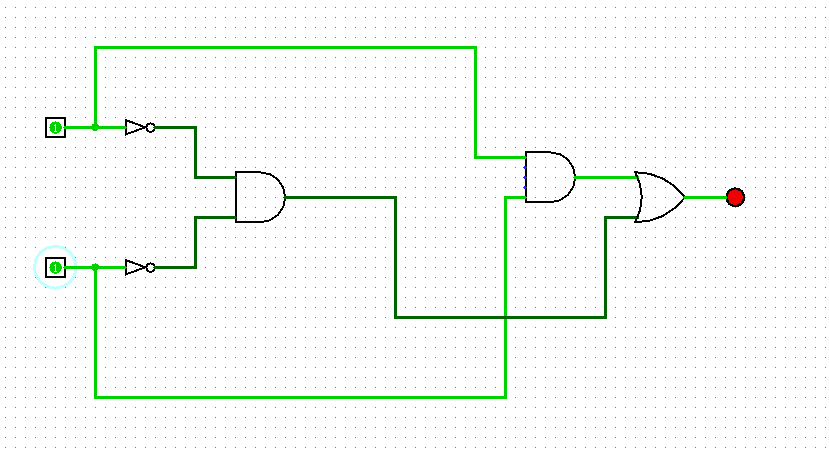
1. Using theorems of Boolean algebra, simplify the equation.

(A’B’C’) + (AB)’C + (AB)C’ + ABC

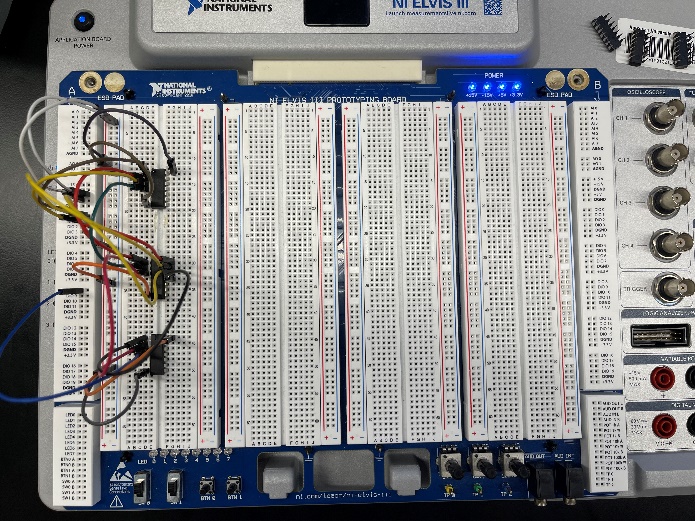
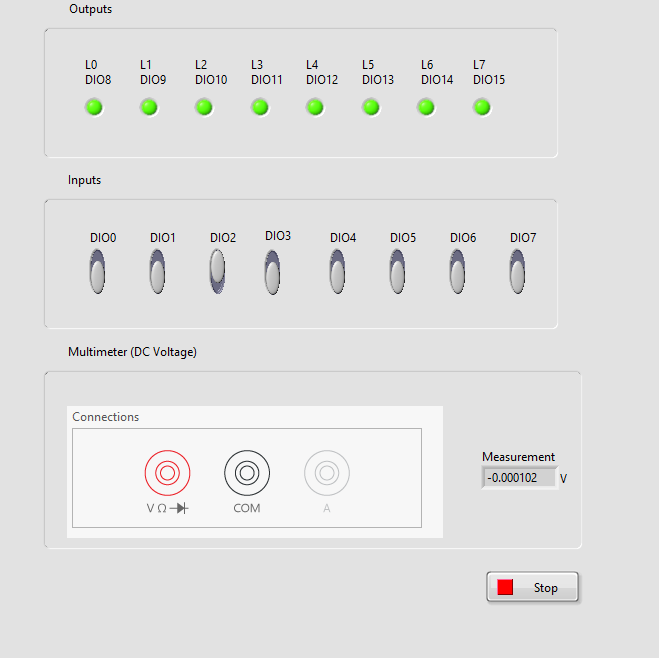
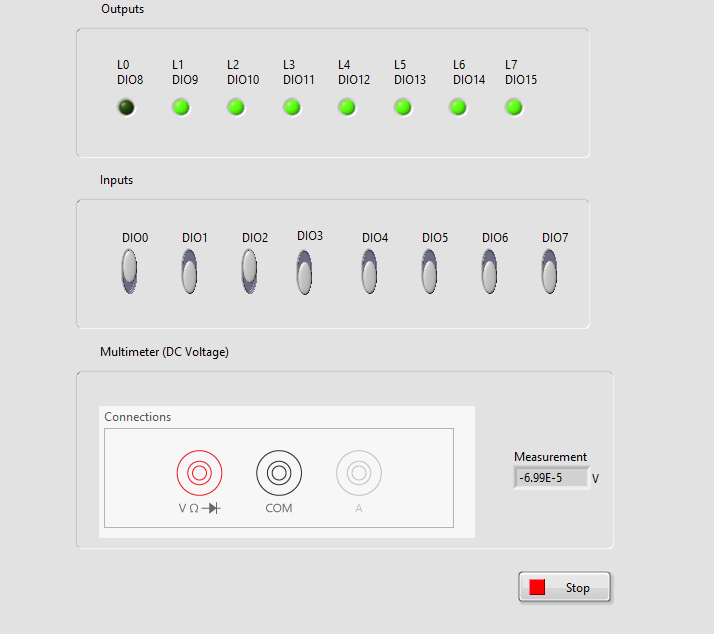
A’B’(C’ + C) + AB(C’ + C)

(A’.B’) + (A.B)

1. Simulate the simplified equation circuit.



1. Implement the circuit and verify your implementation.



**Exercise 3**

Given the following Boolean expression:

1. Draw the circuit for F and implement it using LogiSim, verify its operation by building the truth table.

A B F

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1. Convert F to all NOR design, show your design and implement it on ELVIS 3 board.
2. Convert F to all NAND design, show your design and implement it on ELVIS 3 board.