**THE UNIVERSITY OF AZAD JAMMU AND KASHMIR**



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

|  |  |
| --- | --- |
| COURSE TITLE | CA&LD |
| COURSE CODE | CS-1205 |
| TITLE | Half Adder Implementation |
| STUDENT NAME | Tayyab Hanif |
| STUDENT ROLL NUMBER | 2024-SE-11 |
| INSTRUCTOR NAME | Engr. Sidra Rafique |
| Lab No: | 05 |

**Lab 5: Half Adder Implementation**

**Objective**

To design and verify the working of a Half Adder circuit using basic logic gates. The lab covers both the simple and extended (complex) half adder versions.

**Apparatus Required**

Breadboard

IC 7408 (AND Gate)

IC 7432 (OR Gate)

IC 7486 (XOR Gate)

IC 7404 (NOT Gate)

Power Supply (5V DC)

Connecting Wires

LEDs or Logic Probes

Theory

A Half Adder is a combinational logic circuit used to add two binary digits. It produces two outputs:

- Sum (S)

- Carry (C)

It adds two single-bit binary numbers A and B.

1. **Simple Half Adder**

**Boolean Expressions:**

Sum (S) = A ⊕ B

Carry (C) = A · B

**2. Complex (Extended) Half Adder**

XOR Implementation using Basic Gates:

A ⊕ B = (A · ¬B) + (¬A · B)

So:

Sum (S) = (A · ¬B) + (¬A · B)

Carry (C) = A · B

**Part A – Simple Half Adder**

**Circuit Description**

Connect inputs A and B to a XOR gate to obtain the Sum output.

Connect inputs A and B to an AND gate to obtain the Carry output.

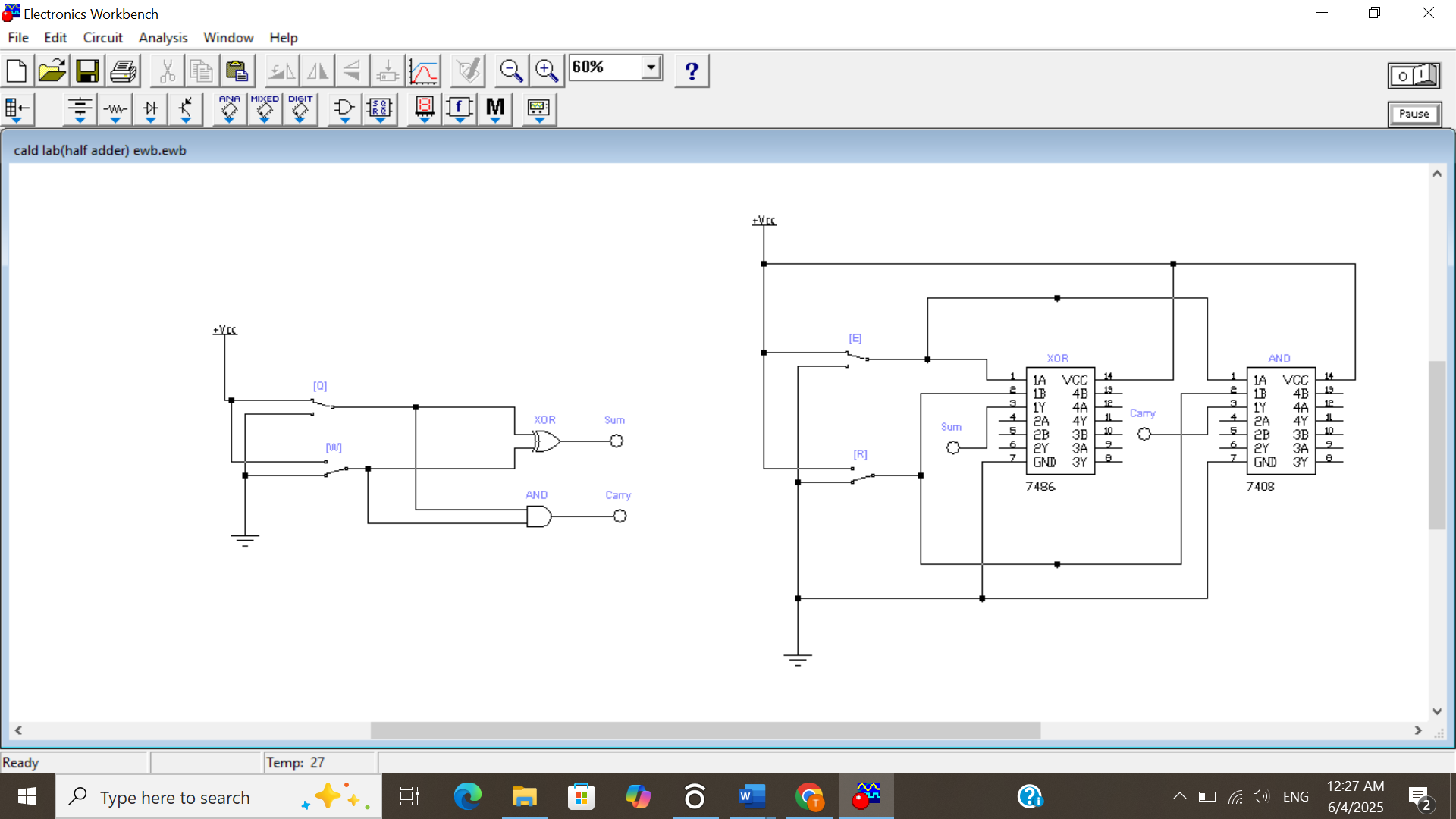
**Truth Table – Simple Half Adder**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | Sum (A ⊕ B) | Carry (A · B) |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

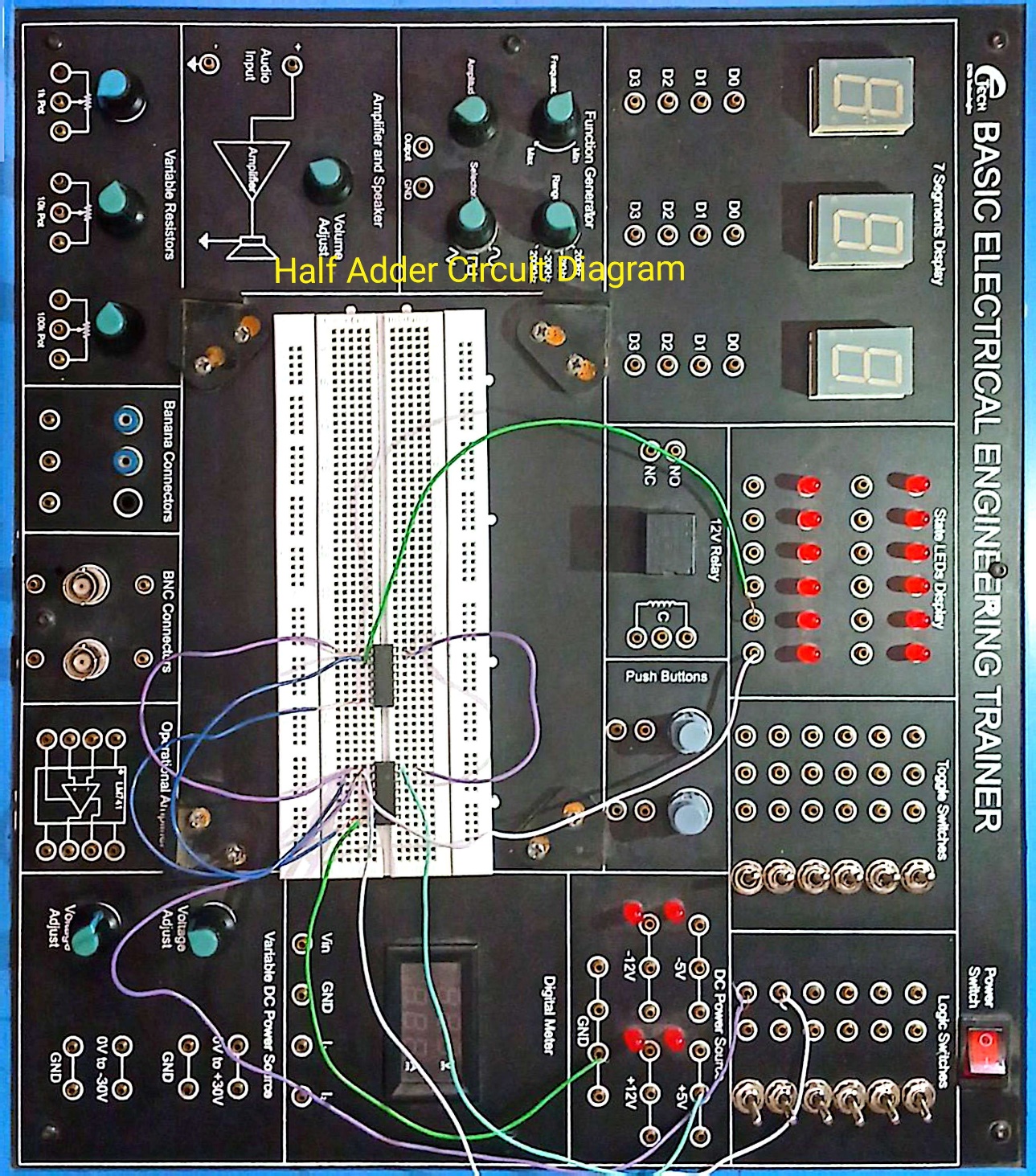
**Observation**

The Sum and Carry outputs match the expected results from the truth table. Thus, the simple half adder is verified.

**Circuit Diagram**



**Implementation**



**Part B – Complex Half Adder Using Basic Gates**

**Boolean Expression for Sum**

S = (A · ¬B) + (¬A · B)

**Circuit Description**

Connect input A to a NOT gate to get ¬A.

Connect input B to a NOT gate to get ¬B.

Use AND gates to get A · ¬B and ¬A · B.

Feed these two outputs into an OR gate to get the Sum.

Use an AND gate with inputs A and B to get the Carry.

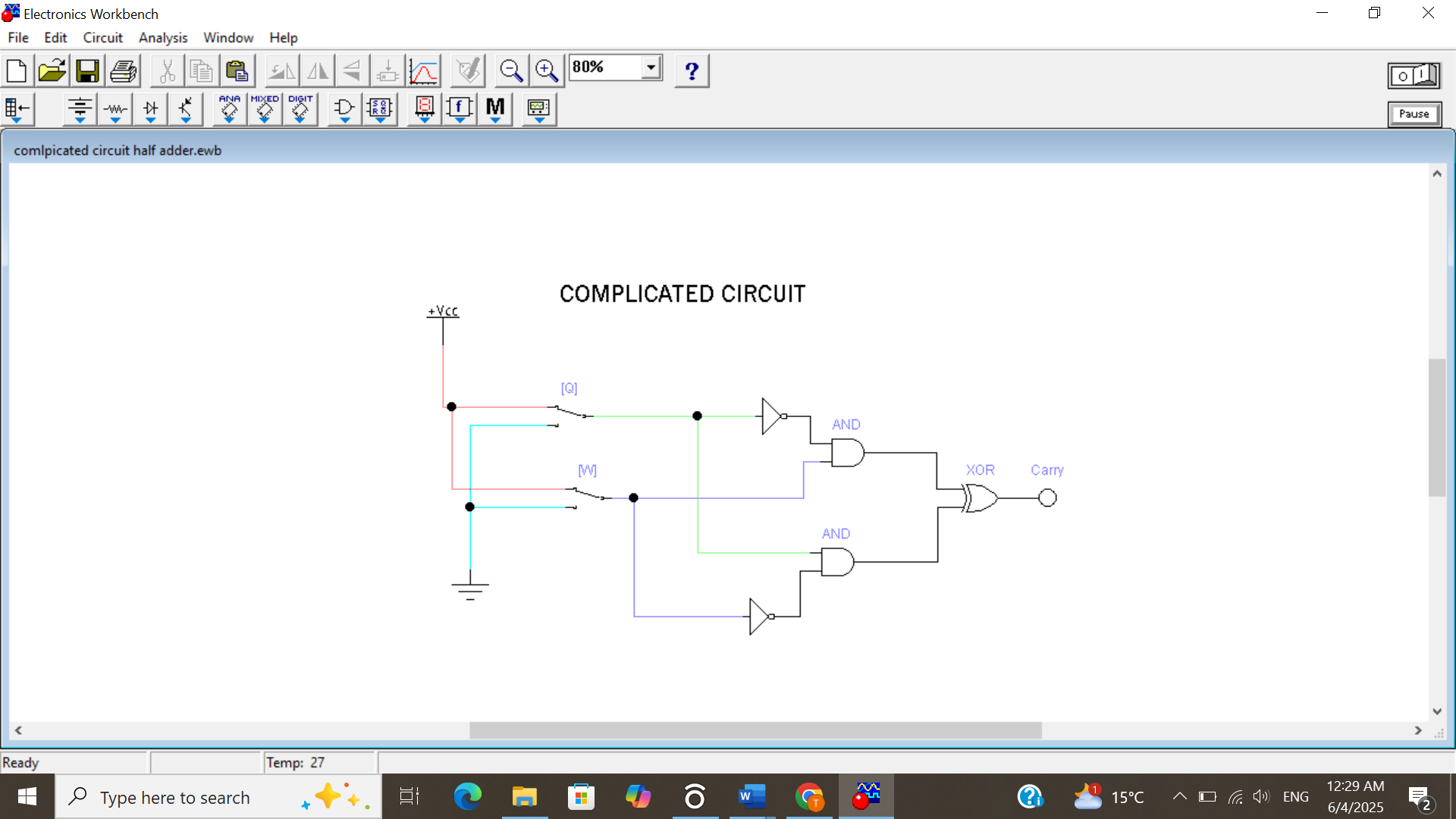
**Truth Table – Complex Half Adder**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | ¬A | ¬B | A·¬B | ¬A·B | Sum | Carry |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

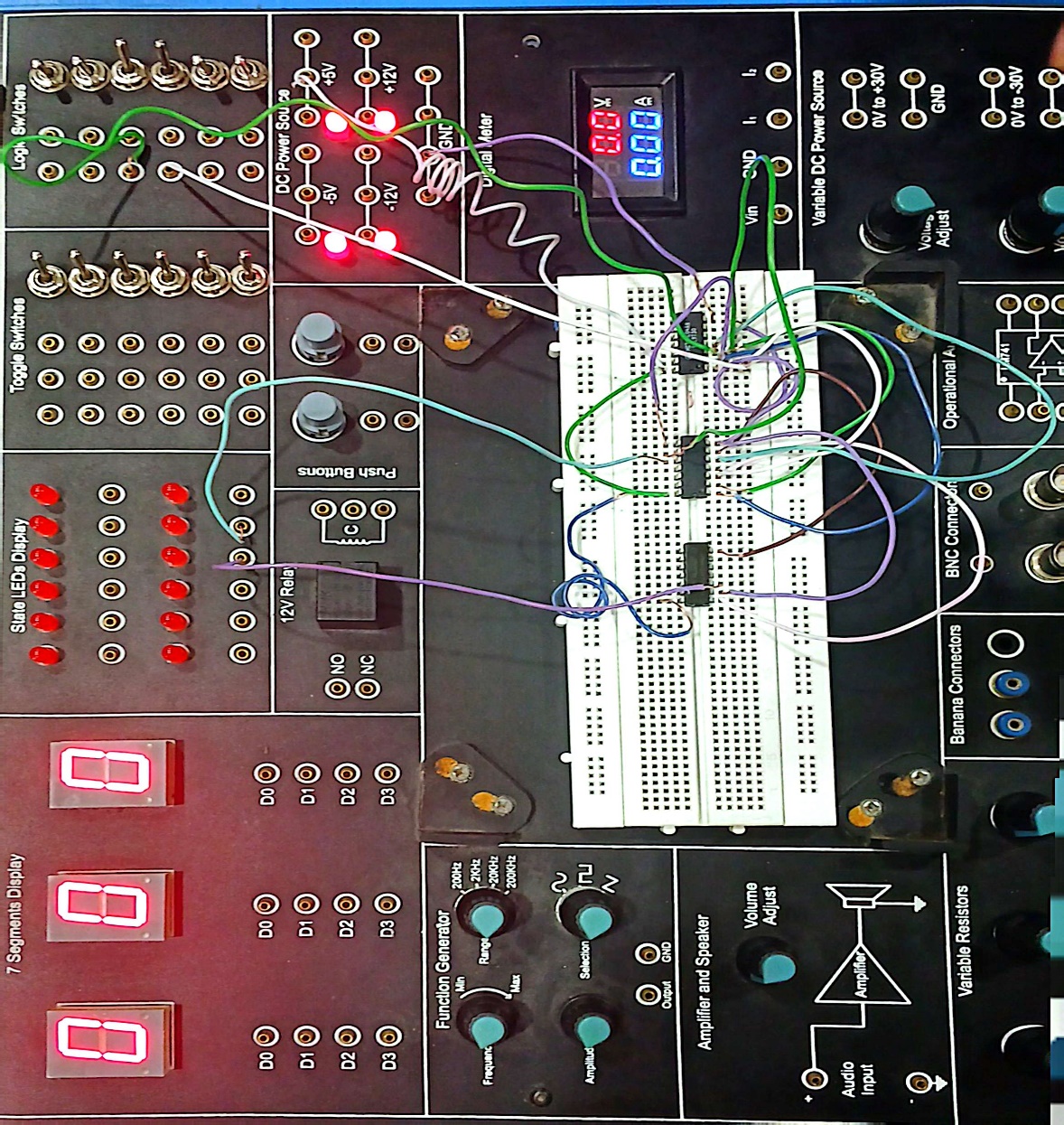
**Observation**

The outputs of the complex half adder match the expected outputs. The XOR function is successfully implemented using only AND, OR, and NOT gates.

**Circuit Diagram**



**Implementation**



**Conclusion**

Both simple and complex versions of the half adder were implemented and tested successfully. The results confirm that:

XOR gates simplify the design of half adders.

XOR functionality can also be replicated using basic gates, validating the theoretical logic expressions.