

**The University of Azad Jammu and Kashmir,**

**Muzaffarabad**

Department of Software Engineering

|  |  |
| --- | --- |
| Name | Saqlain Mushtaq |
| Course Name | Computer Architecture and Logic Design |
| Submitted to | Engr. Sidra Rafique |
| Semester | 2nd |
| Session | 2024-2028 |
| Roll No | 2024-SE-35 |
| Lab No | 05 |

**Half Adder Not**

**Half Adder simple circuit**

**Procedure:**

1. **Objective:**I wanted to make a half adder that adds two 1-bit inputs (A and B) and gives me the sum and carry outputs using a simple circuit.
2. **Understanding the Half Adder:**I know the half adder takes inputs A and B and produces:

* Sum = A XOR B
* Carry = A AND B

Since this is a simple circuit, I directly used XOR and AND gates for these outputs.

1. **Building the Circuit**:

* I connected inputs A and B directly to an XOR gate to get the Sum output.
* Then I connected inputs A and B to an AND gate to get the Carry output.

1. **Testing the Circuit:**I tested all possible inputs:

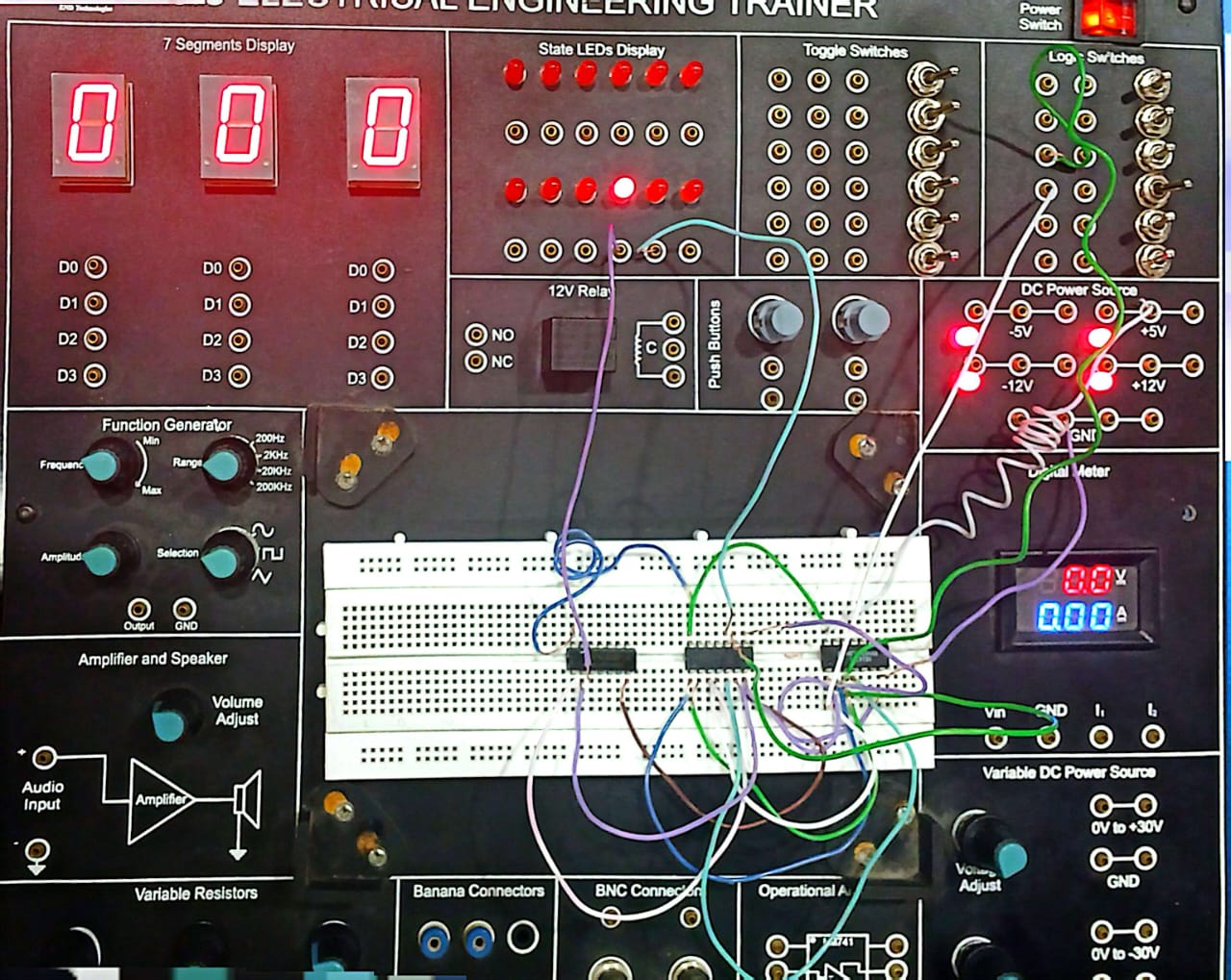
* A=0, B=0 → Sum=0, Carry=0
* A=0, B=1 → Sum=1, Carry=0
* A=1, B=0 → Sum=1, Carry=0
* A=1, B=1 → Sum=0, Carry=1

The outputs were exactly what I expected.

1. **Conclusion:**By using the XOR and AND gates directly, I built a simple and efficient half adder. This helped me understand how these basic gates work together to perform addition.

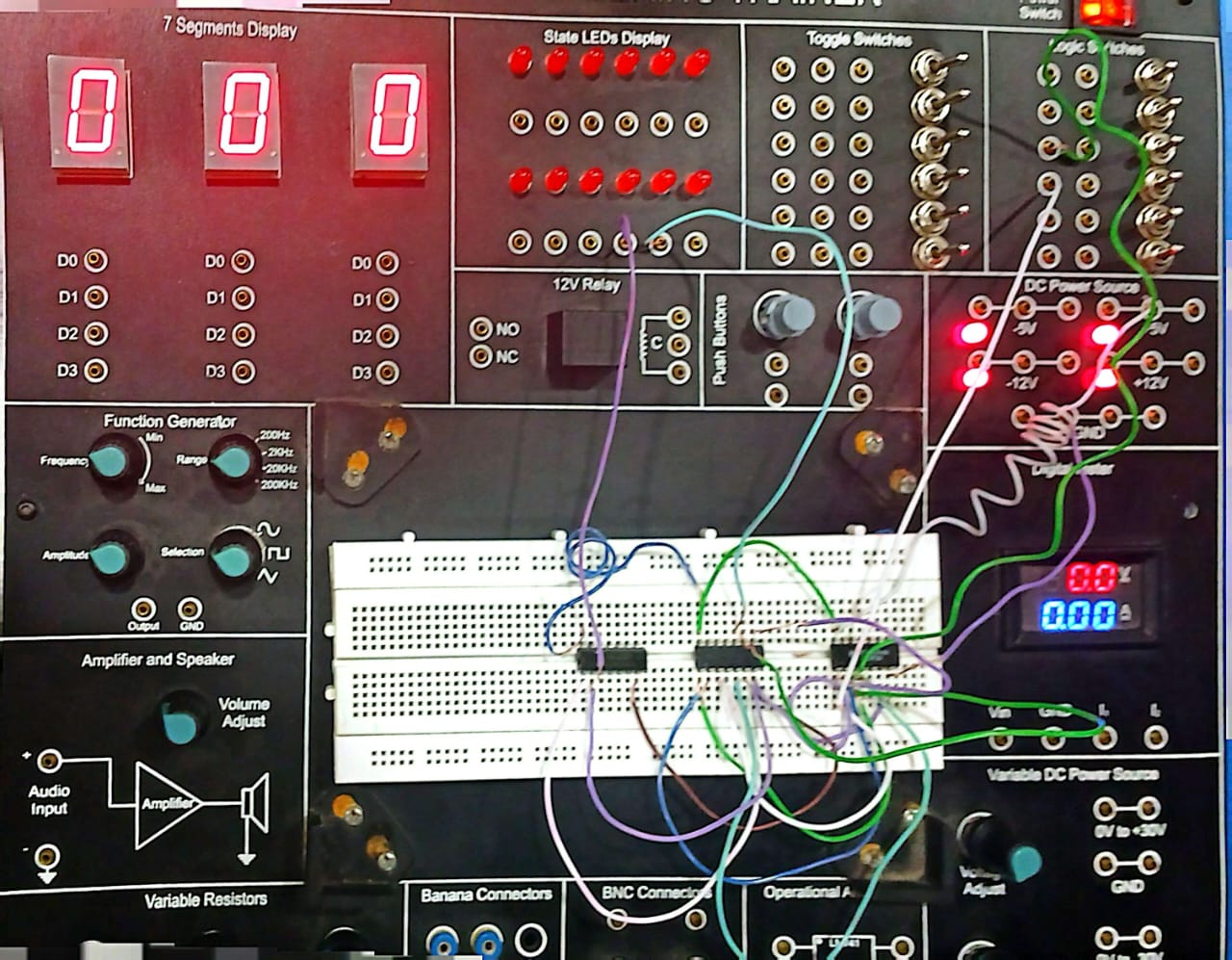
A B = Sum Carry

1 0 = 1 0



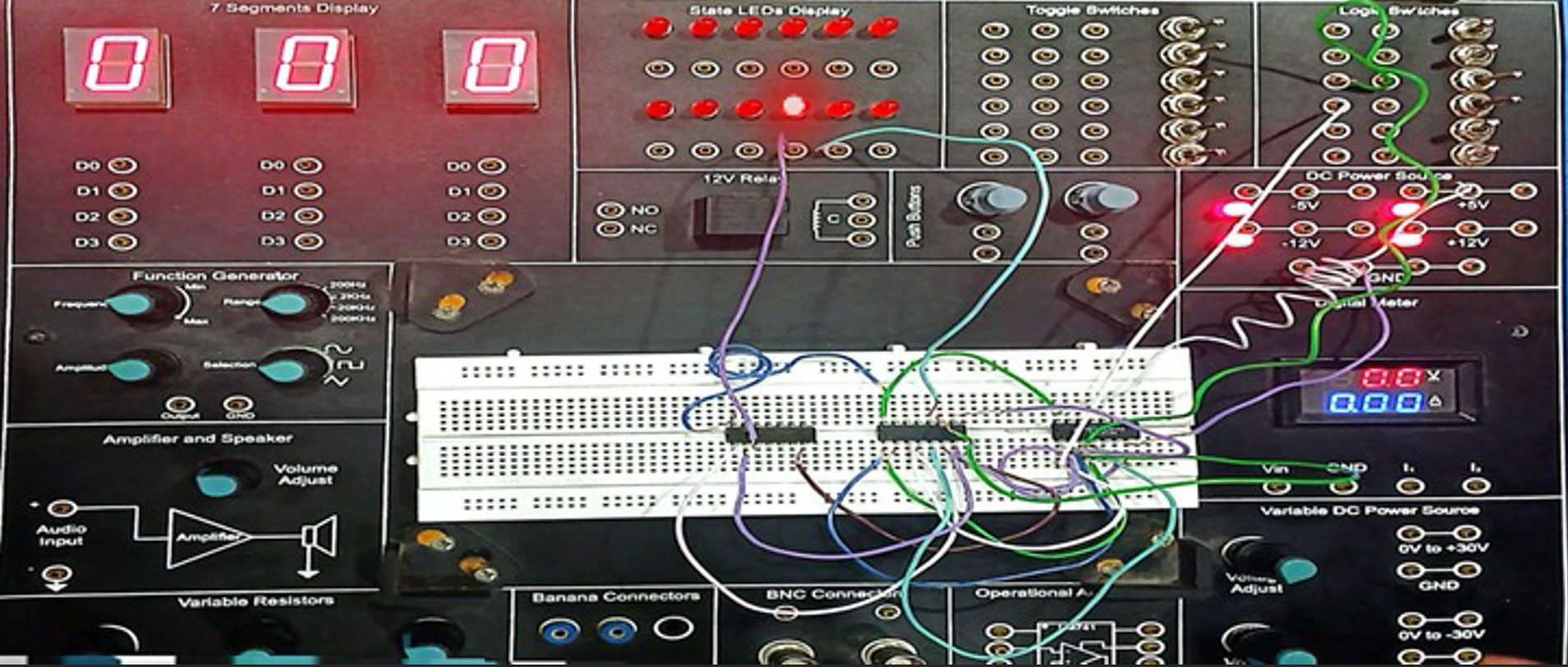
A B = Sum Carry

0 0 = 0 0



A B = Sum Carry

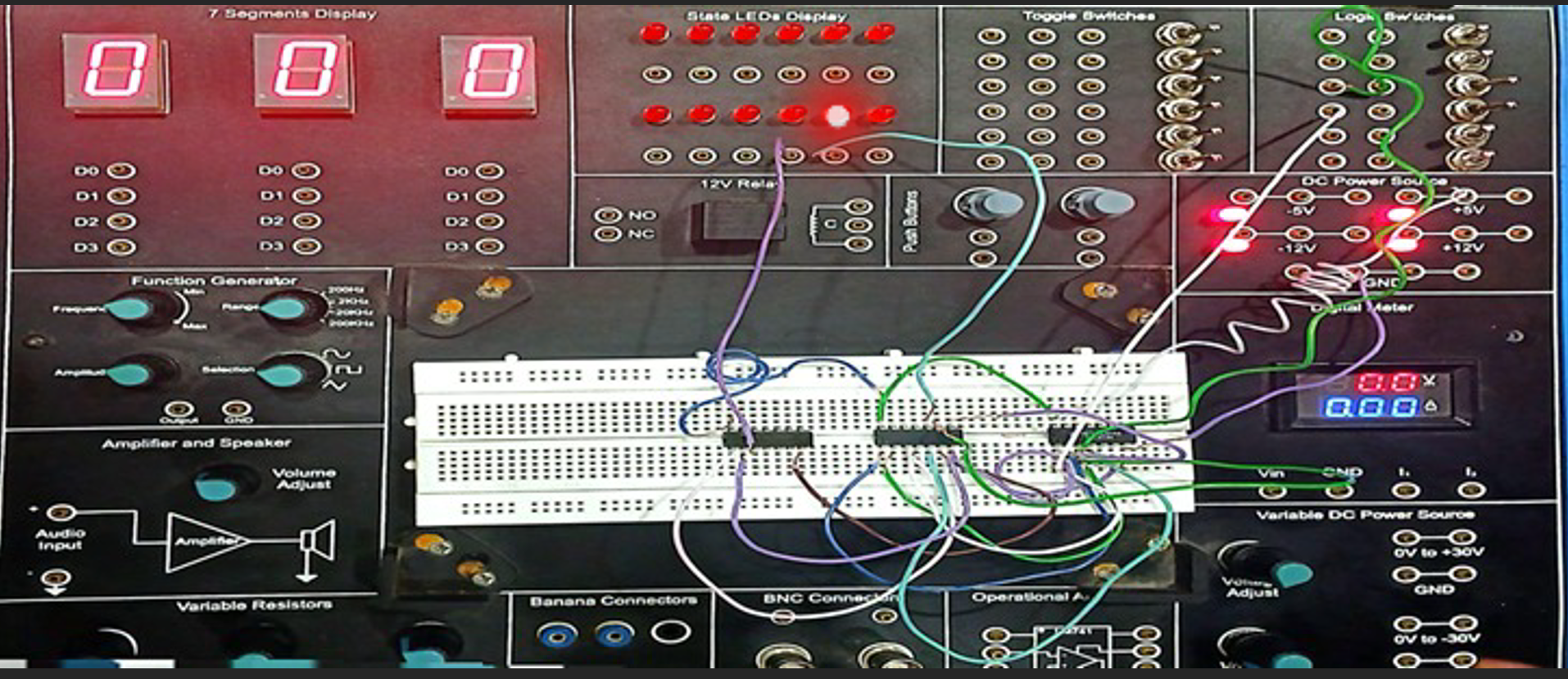
0 1 = 1 0



A B = Sum Carry

1 1 = 0 1





**Half Adder X-OR**

**Half Adder Complicated circuit**

**Procedure:**

**1.Objective:**I wanted to build a half adder that adds two inputs and outputs sum and carry, but only using NOT, AND, and OR gates because XOR gate wasn’t allowed.

**2.Understanding the Logic:**I know a half adder has two inputs (A and B) and two outputs:

* Sum = A XOR B
* Carry = A AND B

Since I can’t use XOR directly, I needed to express XOR using only NOT, AND, and OR gates.

**3.Building the Circuit:**

* I connected input A to a NOT gate to get A’, and input B to AND gate.
* Then I connected input A with AND gate and input B with NOT gate.
* Then I ANDed A’ with B, and A with B’.
* I ORed the results of those two AND gates to produce the sum output.
* For carry, I simply ANDed A and B directly.

**4. Testing the Circuit:**I tested the circuit for all possible inputs:

* For A=0, B=0 → Sum=0, Carry=0
* For A=0, B=1 → Sum=1, Carry=0
* For A=1, B=0 → Sum=1, Carry=0
* For A=1, B=1 → Sum=0, Carry=1

Everything worked exactly as expected.

**5.Conclusion:**By building the half adder using NOT, AND, and OR gates, I learned how to create XOR functionality from basic gates. This helped me understand how complex logic circuits are made from simple parts.

A B = Sum Carry

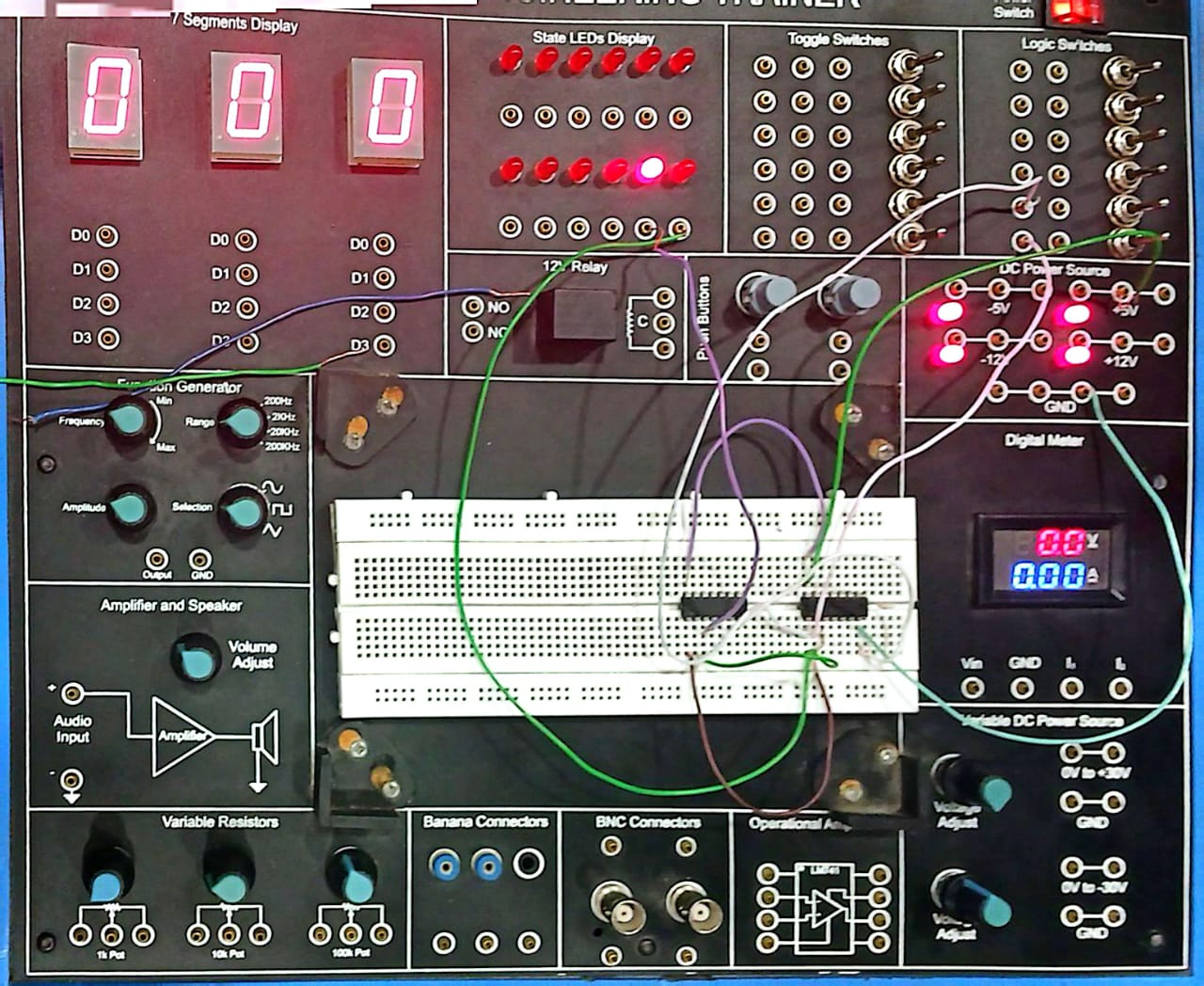
0 0 = 0 0

A close-up of a electrical trainer

AI-generated content may be incorrect.

A B = Sum Carry

1 1 = 0 1



A B = Sum Carry

0 1 = 1 0

