***On my honor, I pledge that I have not violated the provisions of the NJIT Student Honor Code***

STUDENT TEAM No\_\_**2**\_\_

Name Signature

…Arsh Bhamla………… \_\_\_**ARSH BHAMLA~~\_\_\_~~**\_ …Pranay KC………….. \_**\_PRANAY RAJ KC\_\_\_\_\_** ..Shreedhar Rajendran. **\_\_Shreedhar Rajendran—**

..Christopher Karkenny. \_\_**Christopher Karkenny**\_

Experiment performed on date \_02/ 28 / 2025\_\_

Report submitted on date \_\_\_\_\_\_\_\_\_

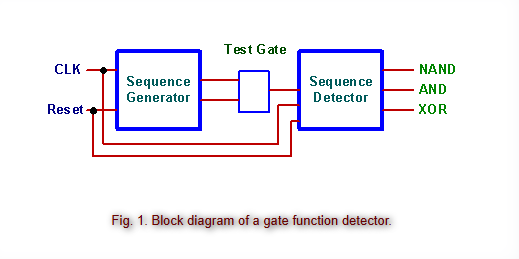
Returned for corrections on date \_\_\_\_\_\_\_\_\_ Grade \_\_\_\_\_\_

Returned after corrections on date \_\_\_\_\_\_\_\_\_ Grade \_\_\_\_\_\_

**Introduction**

In this lab project, we will develop a chip function detector to accurately identify an unknown two-input, one-output gate. We will generate a two-bit input sequence (00,01,10,11) and monitor the outputs to deduce the gate’s function. The possible functions (NAND,AND,XOR) are each characterized by unique truth tables and by analyzing the output, we can determine the gates function. The sequence generator that is built with flip-flops will create a 2-bit counter producing input values sequentially, one per clock cycle. The sequence detector will store the output values from the test gate. As the sequence generator cycles through all input combinations the detector accumulates the outputs. After completing the input sequence the detector compares the stored outputs against the expected outputs for NAND, AND, and XOR. The circuit will activate an LED for the detected function. We will use a reset input to synchronize input sequences and clear stored outputs to ensure accuracy. This lab project applies theoretical knowledge of digital logic design and will allow us to gain hands-on experience with sequential circuits.

**Procedure**

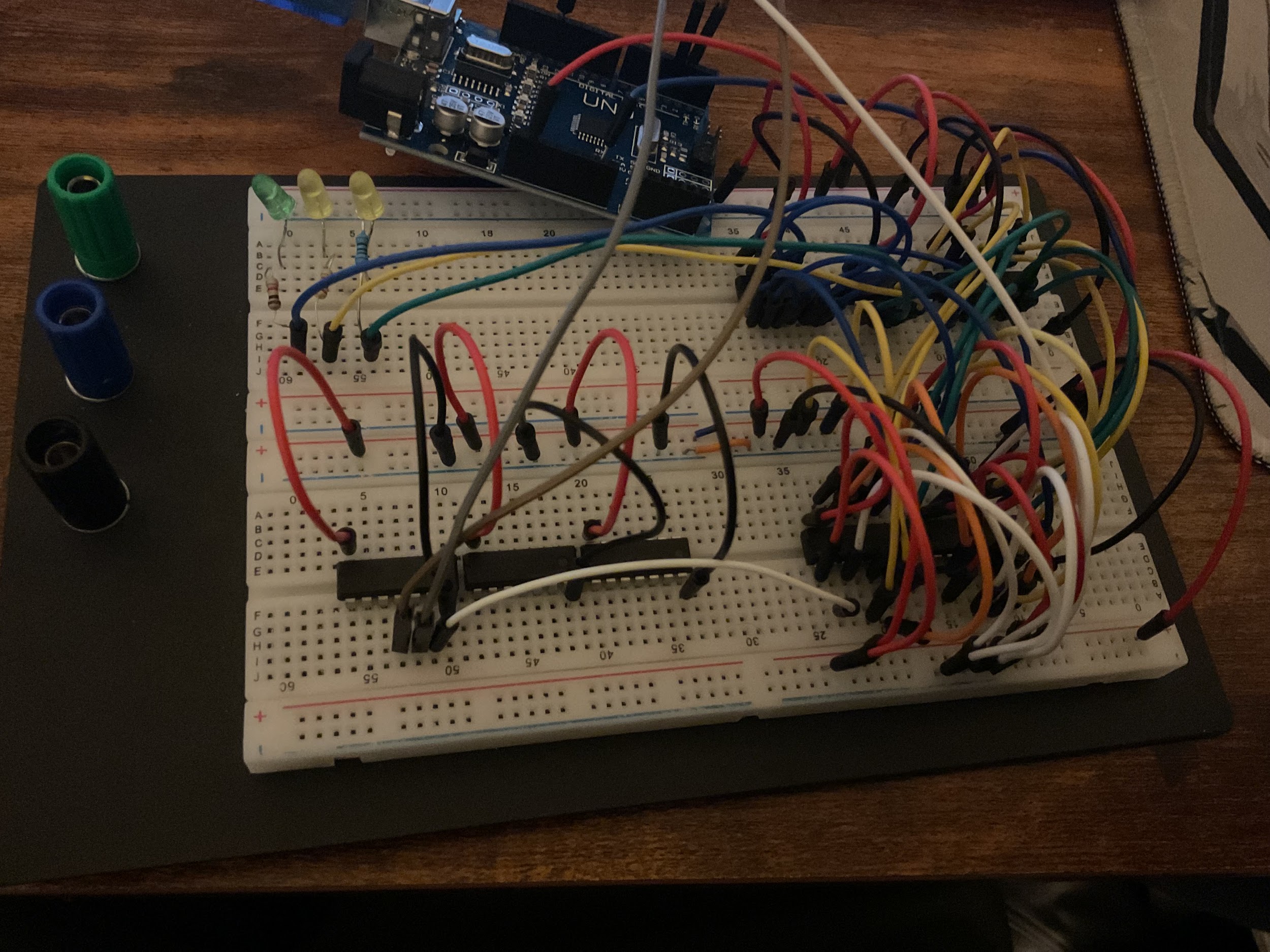
****

*Fig 1. Block diagram of a gate function detector from* [*ECE 294 lab manual.*](https://ecelabs.njit.edu/ece294/lab5.php)

We set up the sequence generator using flip-flops to make a 2-bit counter, which churned out the input sequence 00-01-10-11 one value per clock cycle. We connected the output of the generator to the input of the mystery gate and then crafted the sequence detector to capture the output generated by the gate for each input combination. The input sequences are synced with a reset input to make sure the sequence generator started at 00 and wiped any stored outputs in the sequence detector clean. As the generator cycles through inputs, the detector stored the corresponding output values. We compared the stored outputs with what would be expected for NAND, AND, and XOR gates with a LED.

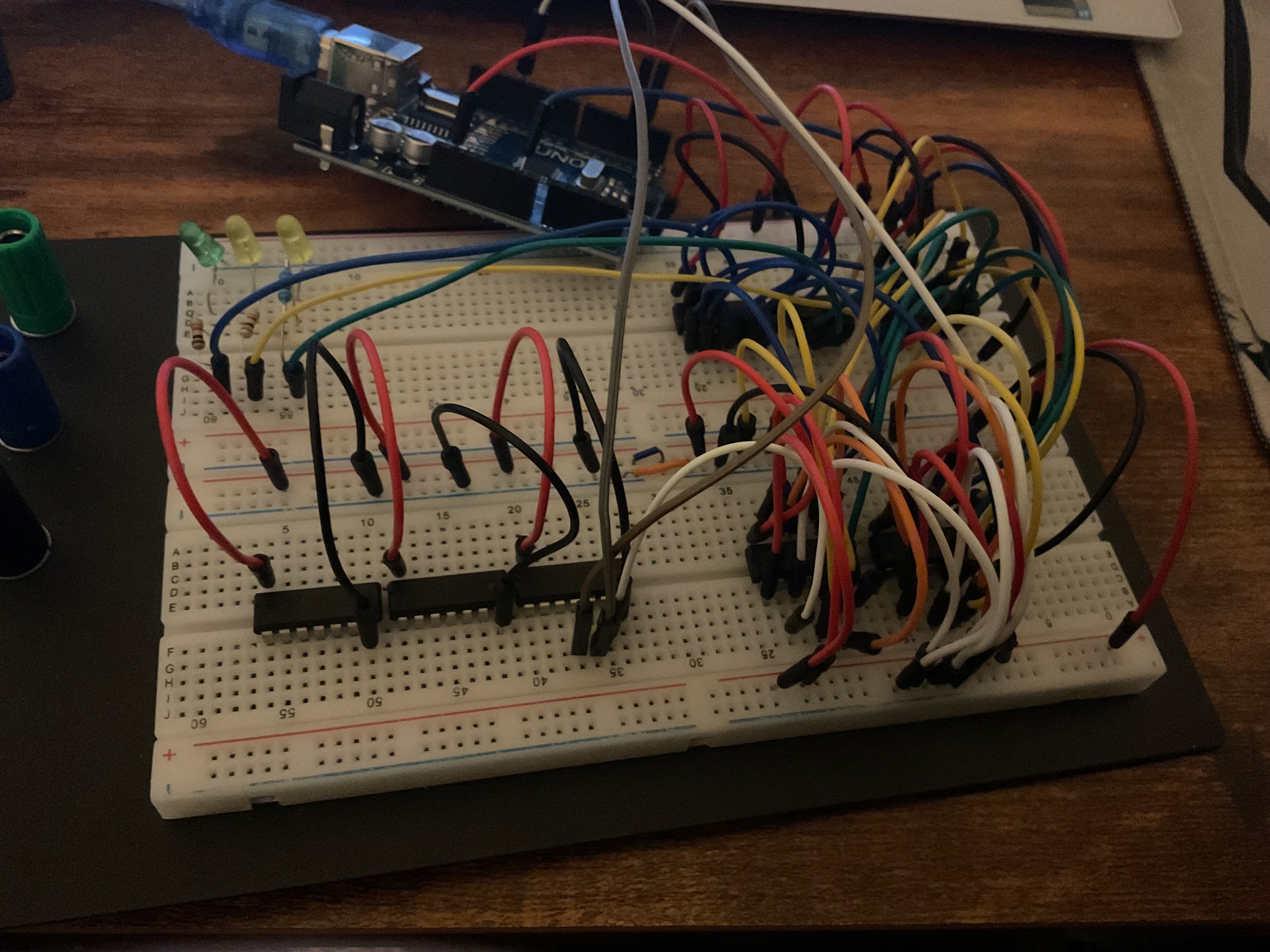
**Data and Calculations**

AND Gate



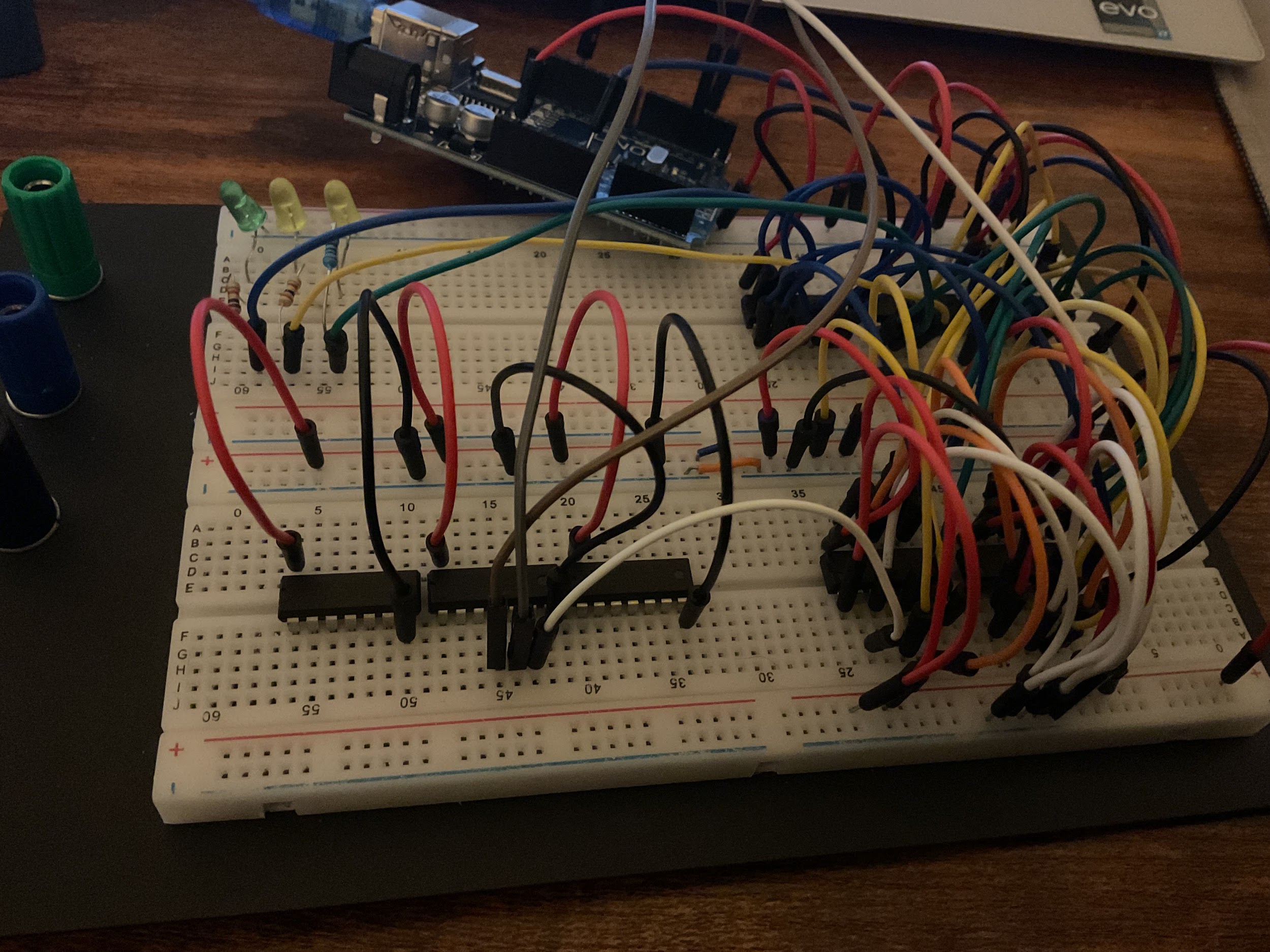
| **Clock** | **Flip Flop 1** | **Flip Flop 2** | **Flip Flop 3** | **Flip Flop 4** | **AND Output** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 1 |

XOR Gate



| **Clock** | **Flip Flop 1** | **Flip Flop 2** | **Flip Flop 3** | **Flip Flop 4** | **XOR Output** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 |
| 4 | 1 | 0 | 1 | 0 | 1 |

NAND Gate



| **Clock** | **Flip Flop 1** | **Flip Flop 2** | **Flip Flop 3** | **Flip Flop 4** | **NAND Output** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 | 1 |

Code for the sequence generator used for the demo:



How it works:  
First, a sequence of dimensions 4x2 (4 rows and 2 columns) is created to store the sequence that will be generated, and the state is initiated as 0. The pins are also defined and the pin modes are declared.

In the void loop, pins a and b are written as per the sequence's first and second item values, which are displayed on the serial monitor. Then, since the D-flip flops are low asserted, the clock is set at 0 (after low asserted), delayed for 500 ms, set at 1(after low asserted), delayed for 500 ms, and then set as 0(after low asserted). The two delays of 500 ms add up to be 1 second representing a frequency of 1 Hz. Finally, the state is incremented by 1 and is MOD by 4, such that when the state becomes 4, it is set back to 1 to prevent Out-Of-Bounds error for the sequence.

**Discussion**

Our main goal was to figure out the function of an unknown two-input, one-output gate. To do this, we generated a two-input sequence and analyzed the outputs. We designed the sequence generator using flip-flops to create a 2-bit counter that produced the input sequence 00-01-10-11 in a loop so that way we could test every possible input combination. The output from the sequence generator was fed into the unknown gate and its output was captured by the sequence detector. The detector stored the outputs for each input combination and compared the expected values for NAND, AND, and XOR gates. We used a reset input to make sure the generator started at 00 and cleared the leftover values in the detector. We compared the sequence detector’s stored outputs with the expected outputs for the three possible gate functions using an LED.

**Conclusions**

This lab experiment provided us a comprehensive understanding and opportunity to construct a chip function detector. We successfully developed a circuit capable of identifying the function of an unknown two-input, one-output gate by generating and analyzing a systematic two-bit input sequence. Designing the sequence generator and sequence detector while making sure everything was properly synchronized helped us accurately determine if the gate was a NAND, AND, or XOR gate. We methodically tested each possible input combination and compared the outputs to the expected values. This hands-on experience really solidified our grasp of digital logic principles and showed us how crucial precise synchronization and thorough testing are in circuit design. The project’s success highlighted the importance of paying attention to details and systematically debugging to achieve accurate results in digital circuit design. This project definitely allowed us to deepen our theoretical knowledge and improve our practical skills in building and testing digital circuits.