Alp Tekin

22403597

EE102-02

10.12.2025

Bilkent University

Department of Electrical and Electronics Engineering

Lab 7 Report: Finite State Machine

Purpose

The aim of this experiment is to design a finite state machine (FSM) on breadboard by using some components like ICs, LEDs and buttons.

Methodology

The FSM involves 2 states, 2 inputs and 1 output. The FSM design was initially built using state transition diagram, state transition table and output table. After that, the design was implemented using essential logic gates, buttons and LEDs on breadboard. Finally, the FSM behavior is tested by applying various input sequences to verify the corresponding LED states.

Design Specifications

The FSM design represents a door locking mechanism. The two states “locked” and “unlocked”, show the current door status of the system. The corresponding state is assigned as shown:

- state: UNLOCKED: “1”
- state: LOCKED: “0”

Two push-buttons are used as inputs (X_1, X_0). Each input bit is connected to a white LED to provide visual feedback for the input signals. The door locking mechanism operates based on the current state:

- **LOCKED State:** This state is displayed by the Red LED. The door remains locked for inputs “00”, “01”, and “10”. Only the correct password input **“11”** triggers a transition to the UNLOCKED state. Once correct password is entered the Red LED turns off.
- **UNLOCKED State:** This state is displayed by the Green LED. When unlocked, the system uses its memory feature where inputs “00” and “11” keep the door open. To re-lock the door, a 'touch-to-lock' logic is used. Thus, inputs **“**01” and “10” return the system to the LOCKED state. Once only a button is pressed the Green LED turns off.

Furthermore, the state transition diagram is shown in Figure 1, while the state transition table and output table can be seen in Figure 2.

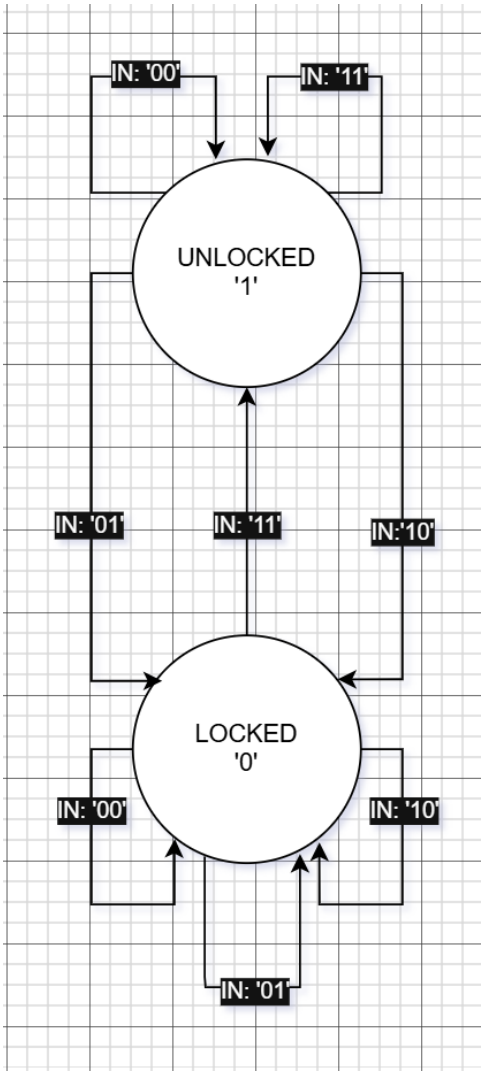

Figure 1: The State Transition Diagram

Q (Present State)	X_1	X_0	Q^+	Output
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**0**	**0**
0	**1**	**0**	**0**	**0**
0	**1**	**1**	**1**	**1**
1	**0**	**0**	**1**	**1**
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**0**	**0**
1	**1**	**1**	**1**	**1**

Figure 2: State Transition Table

Using the Karnaugh Map of the table, the logic for D input is found as $D = (X_1.X_0) + (Q.\overline{X_1}.\overline{X_0})$ and the output of the system is equal to Q^+. Since the output depends only on the states, this system is a Moore Machine. The design includes 4 different gates: a NOT gate (SN74HC04N), a 2-input AND gate (SN74HC08N), a 2-input OR gate (SN74HC32N) and a Positive-Edge D Flip Flop (SN74HC74N). The clock input of the D Flip Flop is driven by a function generator set to a 1 Hz frequency to allow visual observation. The state is stored using a D Flip Flop while the combinational logic is implemented using the logic gates mentioned above. White LEDs are used to indicate the status of 2-bit inputs; the green LED represents the UNLOCKED state (Q = 1) while red LED represents LOCKED state (Q = 0).

Results

First the FSM was built on breadboard using necessary components as shown in Figure 3.

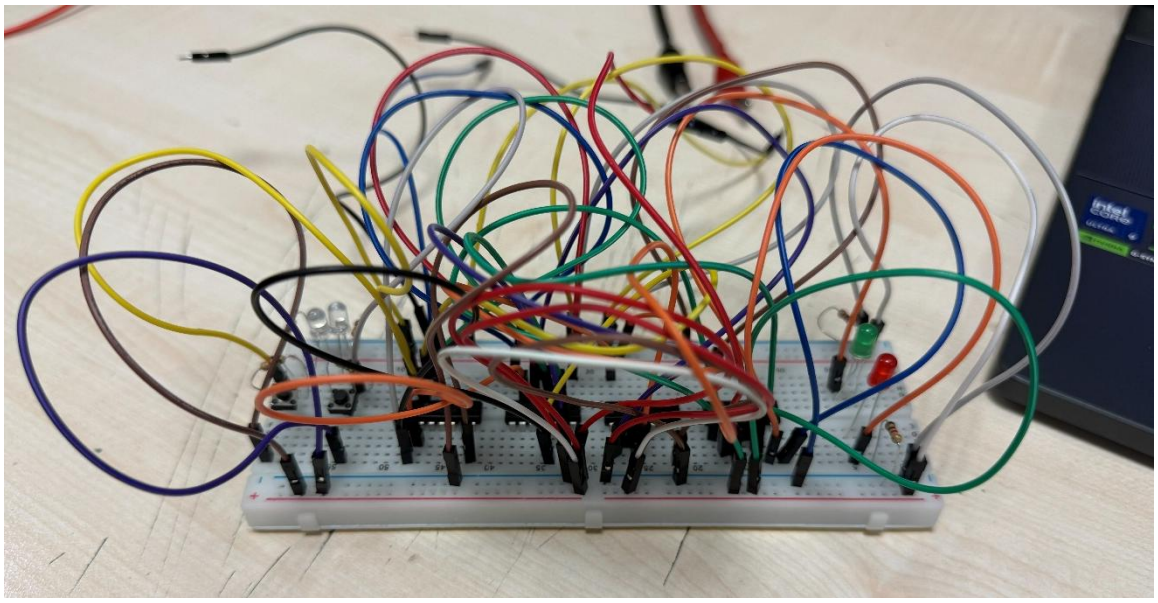

Figure 3: Breadboard Setup

Then, to verify that the system is working properly, FSM was tested using different 2-bit inputs via buttons. The inputs and the corresponding state of the door can be observed in Figure 4-10.

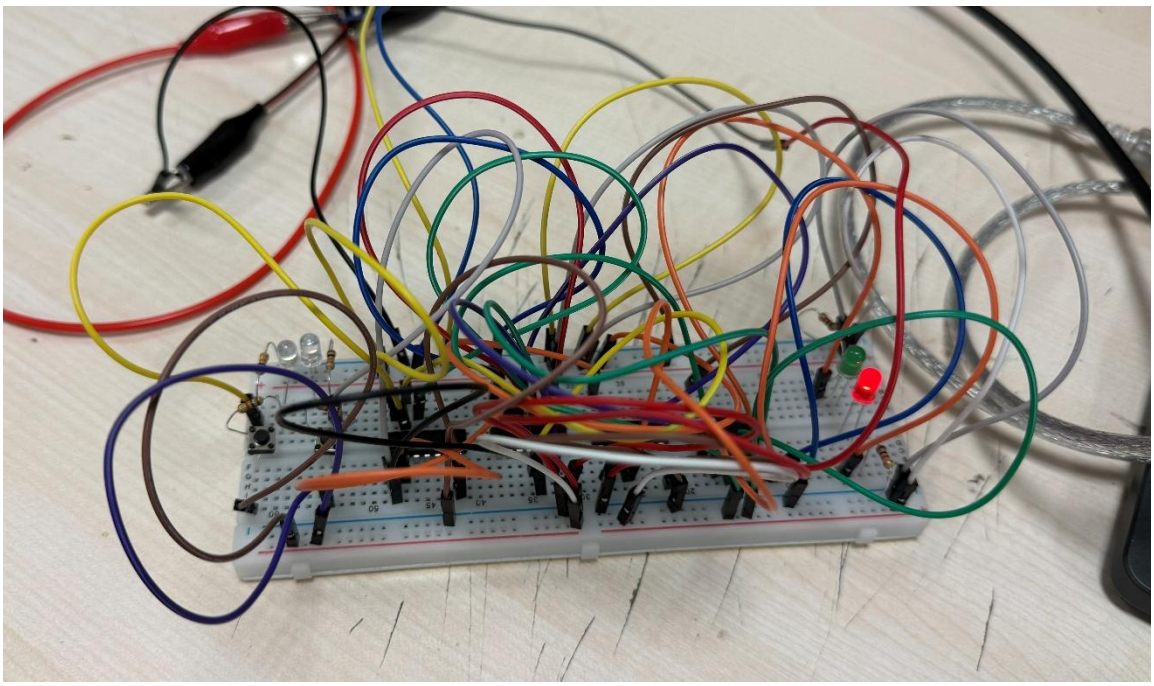

Figure 4: When Inputs X_1X_0: “00”, Q: “0” Output is 0 again so the LOCKED state remains

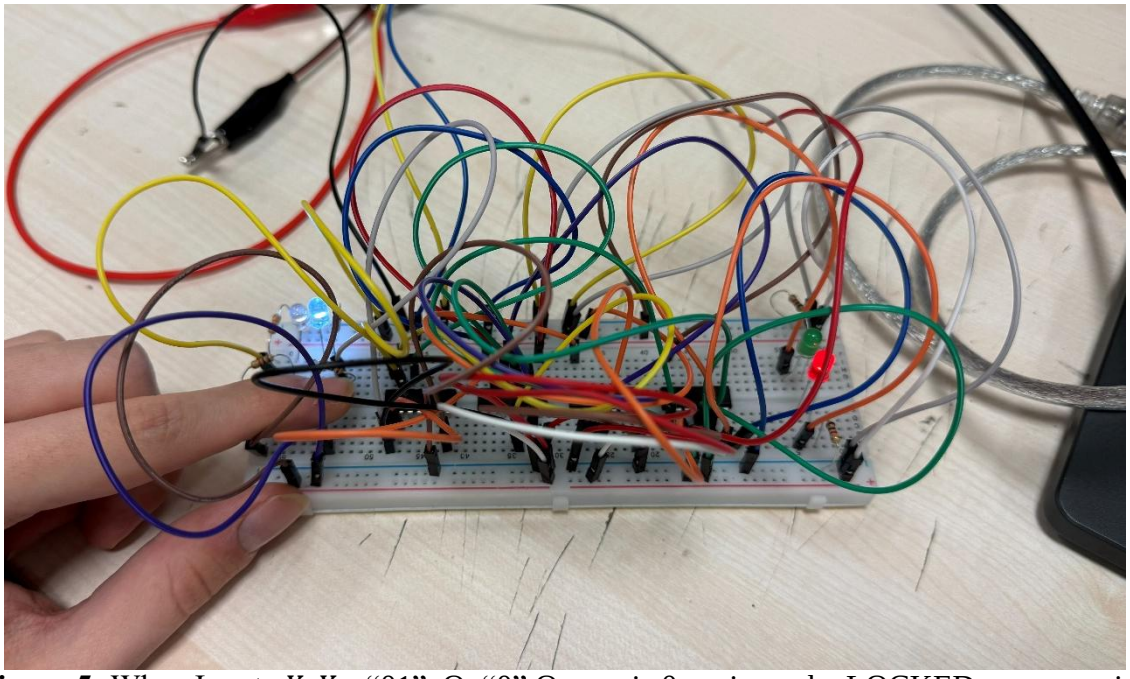

Figure 5: When Inputs X_1X_0: “01”, Q: “0” Output is 0 again so the LOCKED state remains

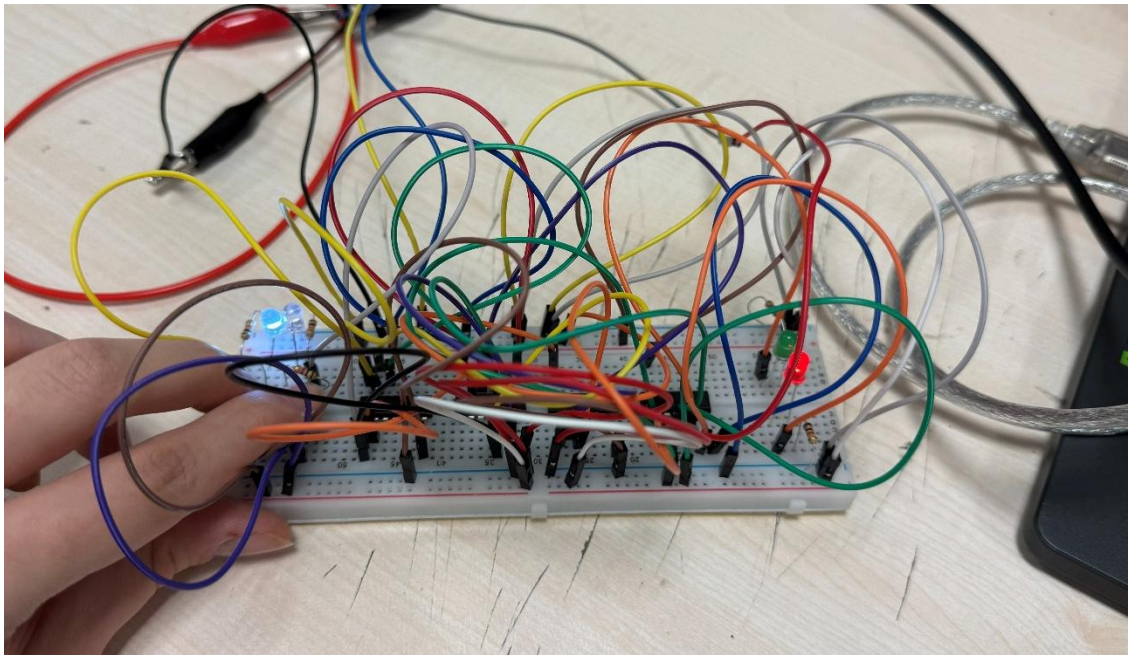

Figure 6: When Inputs X_1X_0: “10”, Q: “0” Output is 0 again so the LOCKED state remains

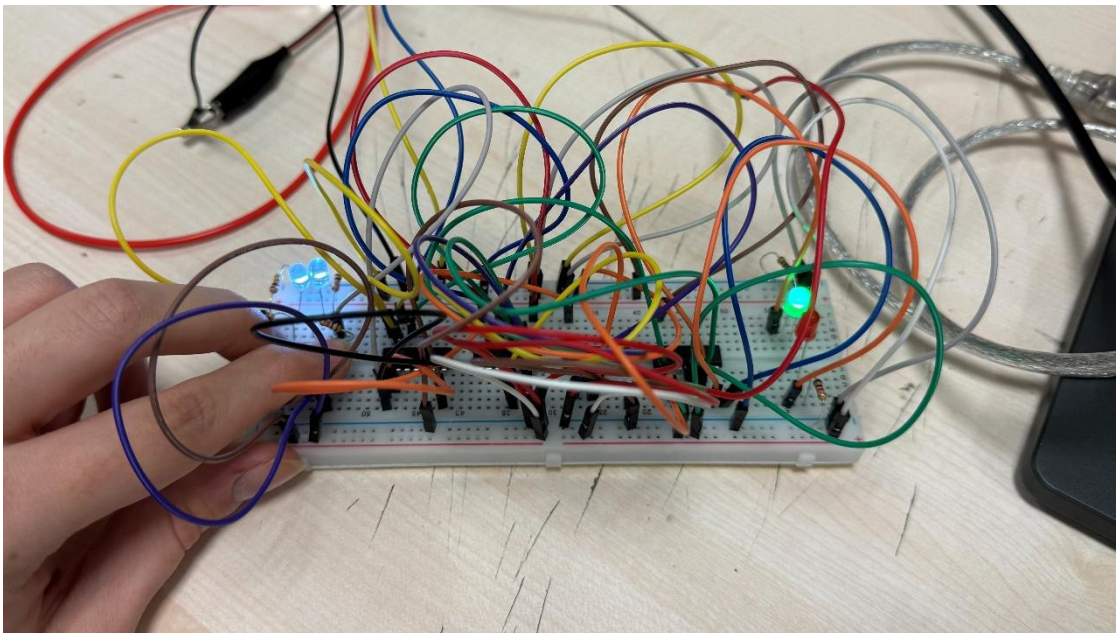

Figure 7: When Inputs X_1X_0: “11”, Q: “0” Output is 1 so the door UNLOCKED

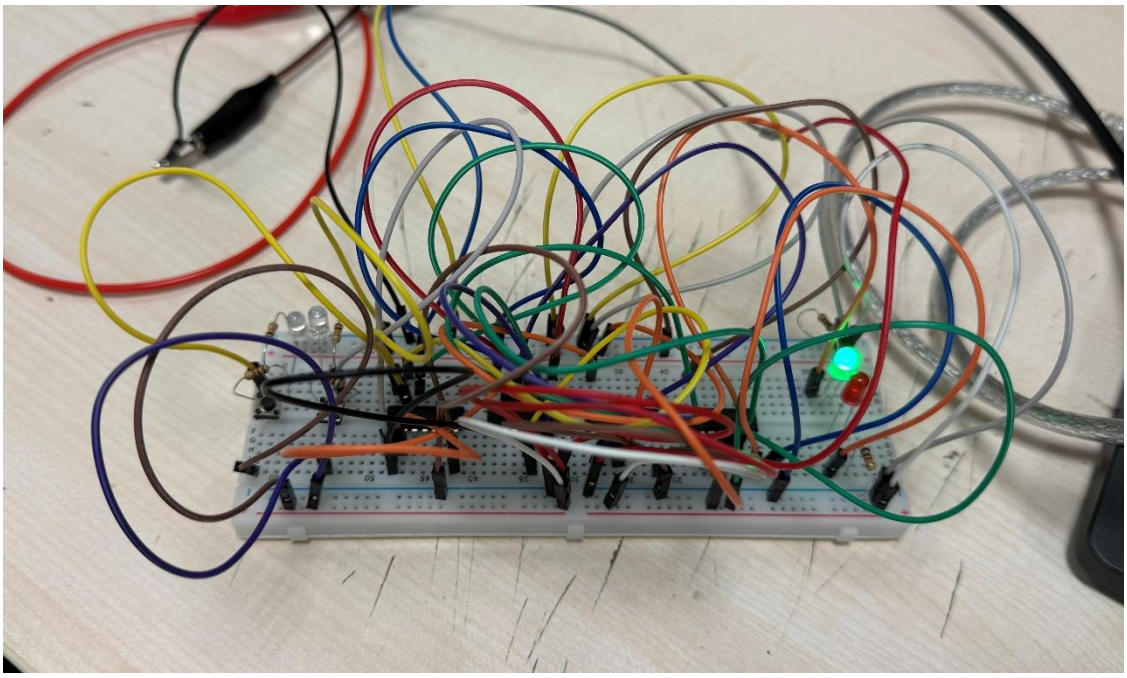

Figure 8: When Inputs X_1X_0: “00”, Q: “1” Output is 1 so the door state remains UNLOCKED

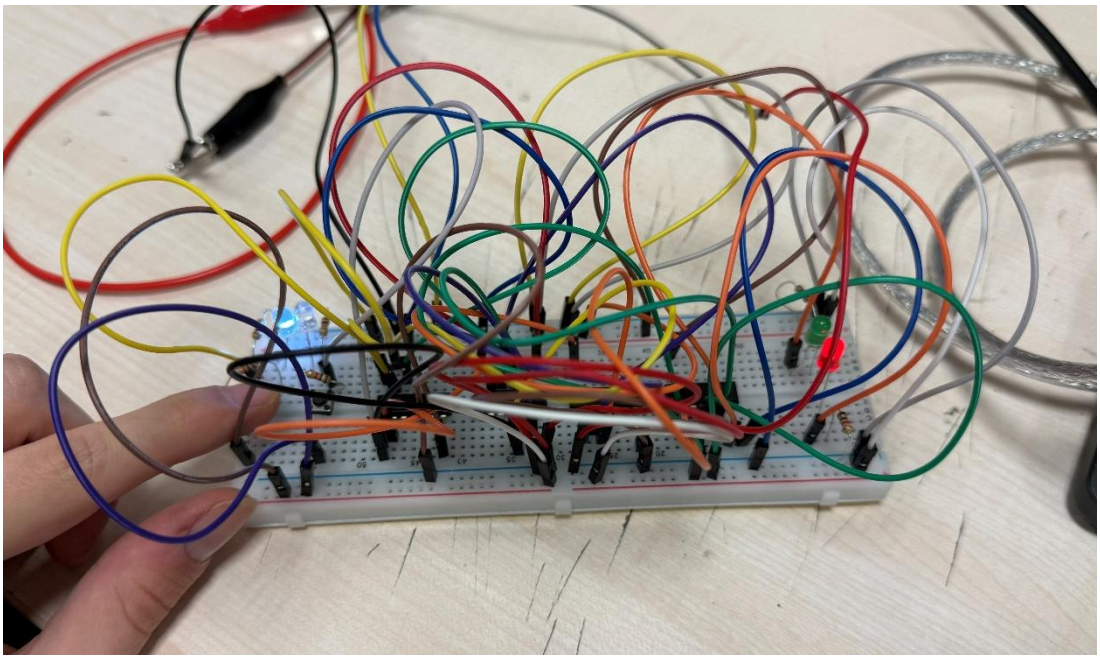

Figure 9: When Inputs X_1X_0: “10”, Q: “1” Output is 0 so the door LOCKED

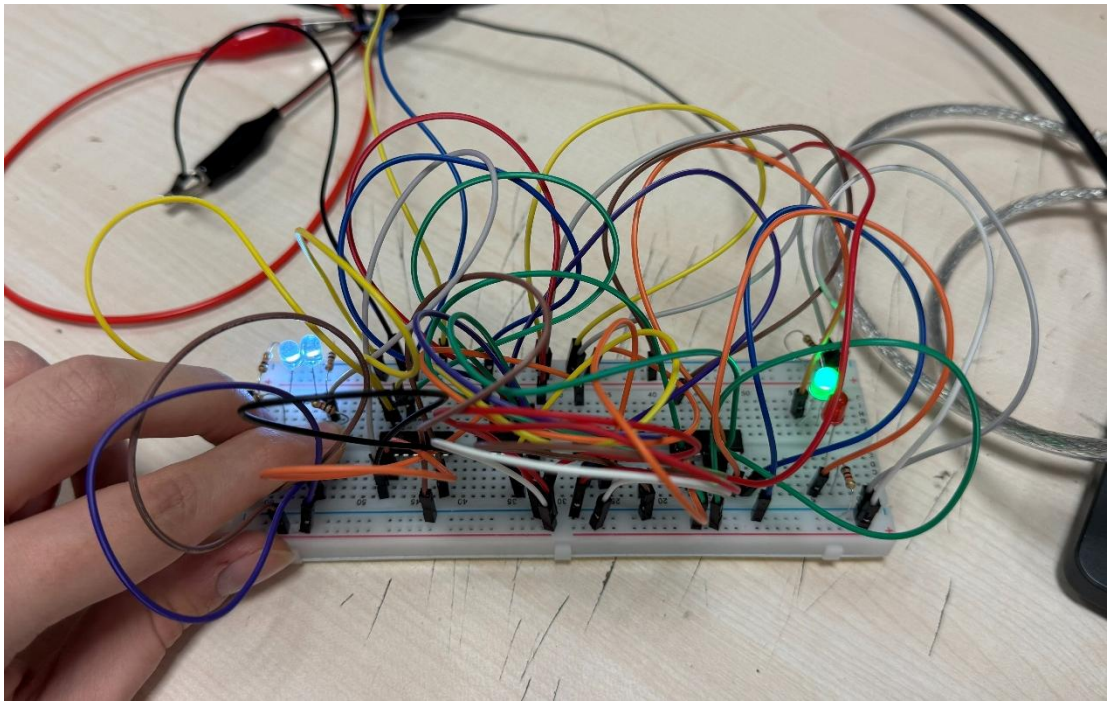

Figure 10: When Inputs X_1X_0: “11”, Q: “1” Output is 1 so the door remains UNLOCKED

Overall, the design behaved as expected and worked correctly. Thus, the finite state implementation was made successfully.

Conclusion

In this lab, an FSM design was constructed on breadboard using necessary components. The observed experimental results were consistent with expected ones so system is working properly. For the design a Moore Machine was selected, since the output depends only on the current state. However, implementing the logic on a breadboard was challenging. To implement the combinational logic, a lot of jumper cables needed to be used which made the breadboard look messy. Also, in the initial design the input “11” locks the door as well. Nevertheless, I observed that there were oscillations when the button “11” was pressed constantly. Since the password is also “11”, the state of the system was continuously changing which makes the system unstable. Overall, the lab was helpful for me to learn how to implement an FSM using a breadboard and logic ICs.

References

BTNcircuit.jpg

LEDcircuit.jpg

https://www.alldatasheet.com/datasheet-pdf/pdf/15661/PHILIPS/74HC74N.html

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/