Lebanese American University

COE322 – Logic Design Lab

Final Project – Logic-Controlled Board

Mahmoud Al Ashkar 202302853

Hussein Itani 202300351

Mahdi Zein Al Dine 202300041

14/05/2025

Table of Content

Table of Content 2
Table of Figures 3
Abstract 4
Introduction 5
Equipment 6
Analysis 7
Paper Design 8
Quartus Design and Analysis 9
  Finite State Machine 9
  Display Unit 9
  Sequence Unit 10
  Timers and Clocks 15
Breadboard Design and Analysis 16
Financial Study 19
Delay Calculation 20
Power Consumption Analysis 21
Challenges 22
Design Advantages 23
Conclusion 24

Table of Figures

Figure 1: Diagram summarizing system functionality 8
Figure 2: Final design of the finite state machine 9
Figure 3: Display Unit design 9
Figure 4: Sequence Unit design 10
Figure 5: Enable Memory design 11
Figure 6: Mapping Unit design 11
Figure 7: Counter W Mode design 12
Figure 8: Mapping Memory design 12
Figure 9: Memory Register design 13
Figure 10: Mapping Stage Flag design 14
Figure 11: Sequence Detector design 14
Figure 12: 100kHz Clock using 555 timer on LTSPice 15
Figure 13: Breadboard implementation 16
Figure 14: Circuit organization 17
Figure 15: Sequence detector 17
Figure 16: Display unit 18
Figure 17: Soldering Technique 18

Abstract

The objective of this project is to design and implement a Logic-Controlled Board, which utilizes digital logic circuits, memory elements, and finite state machines (FSM) to dynamically control the activation of four colored lamps (red, green, blue, and yellow) with four switches. The last switch turned off determines the lamp activation sequence. Users can switch caps and change lamps position, yet the correct color-to-switch mapping will still be maintained. The system creates an illusion leading the audience to believe that the switches are intelligently connected to the lamps. Additionally, this project includes an advanced feature where removing a switch cap deactivates it. This system will be designed and simulated virtually on "Quartus II" software and implemented physically while trying to minimize the number of components used.

Introduction

This logic-controlled board integrates digital logic design concepts to create an interactive and dynamic system for controlling the activation of four lamps based on user interaction with four switches. This system demonstrates the practical application of finite state machines (FSM) and memory elements to track the status of switches and determine the sequence of lamp activation. The system operates in two states: a normal state, where the sequence of lamp activation changes dynamically based on the last switch turned off, and a locked state. Where the switches are directly mapped to their corresponding lamps. In addition to its basic functionalities, this system includes features such as disabling a switch by removing its cap. Additionally, a 7-segment display is implemented in performance mode to indicate in which mode the board is working. This project bridges theoretical knowledge with real-world engineering applications.

Equipment

The components and equipment used in this project:

- Toggle switches (×4)
- Colored caps (×4)
- Colored LEDs (×4)
- Circuit board (×1)
- wires
- Voltage source (×1)
- 720Ω resistors (×2)
- 0.01μF capacitor (×1)
- 10μF capacitors (×1)
- 7-segment display (×1)
- NE555N IC chip (×1)
- 74LS82 IC chip (×4)
- 74LS175 IC chip (×1)
- 74LS74 IC chip (×3)
- 74LS73 IC chip (×8)
- 74LS153 IC chip (×4)
- 74LS157 IC chip (×1)
- 74LS48 IC chip (×1)
- 74LS04 IC chip (×5)
- 74LS08 IC chip (×8)
- 74LS11 IC chip (×4)
- 74LS32 IC chip (×3)
- 74LS02 IC chip (×3)
- 74LS86 IC chip (×2)
- Soldering Iron and iron wire (×1)
- “Quartus II” software
- “LTSpice” software

Analysis

The logic-controlled board system utilizes a combination of digital circuits, memory elements and finite state machines to dynamically control the activation of the four lamps based on the interaction with the four switches. The system has input 4 switches and a reset timer signal as inputs, and 4 outputs (to turn on the lamps) with a 7-segment display. There are two states: normal state (switch 2 is off when the system is turned on, it enters normal FSM flow), locked state (switch 2 is on when the system is turned on, each switch is directly mapped to its corresponding LED (1 to 1, 2 to 2, etc.). So, a multiplexer is needed with the selector (depending on switch 2) to decide which set of inputs to use (locked or normal state). The lamp activation sequence is based on the last switch turned off, and every switch will stay mapped to a lamp until a reset condition. Therefore, we need a sequence unit, consisting of a mapping unit which also contains a sequence detector to indicate the mode we will be working in, and a mapping memory to store the mapping of switches to its corresponding lamps. A display unit is needed to display in which mode we are working on the 7-segment display. Additionally, a clock will be designed using a 555 timer.

Paper Design

The diagram below shows how the system functions in every state

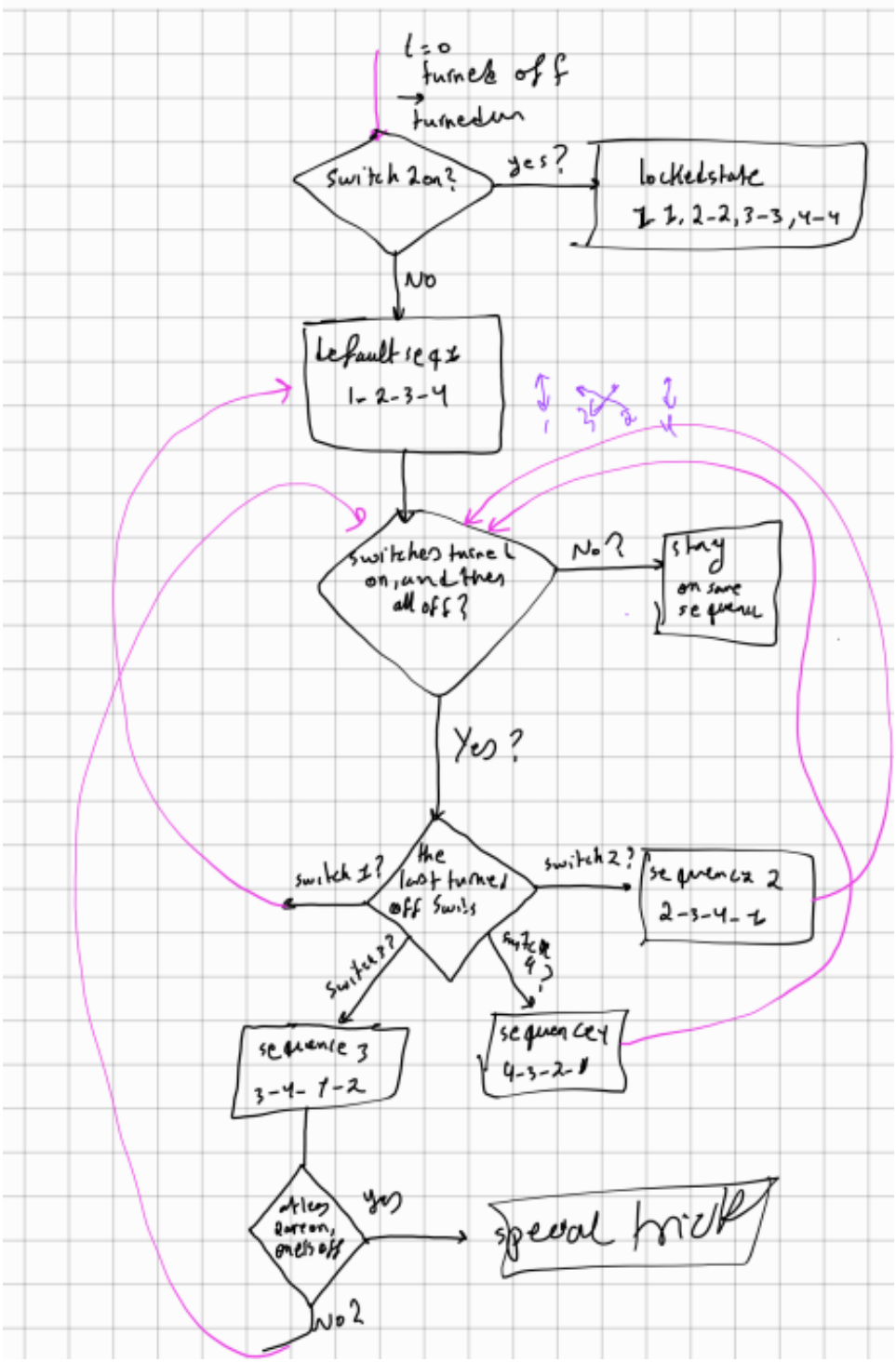

Figure 1: Diagram summarizing system functionality

Quartus Design and Analysis

The different parts of the system are constructed in different blocks and integrated into the final design on “Quartus II” software.

Finite State Machine

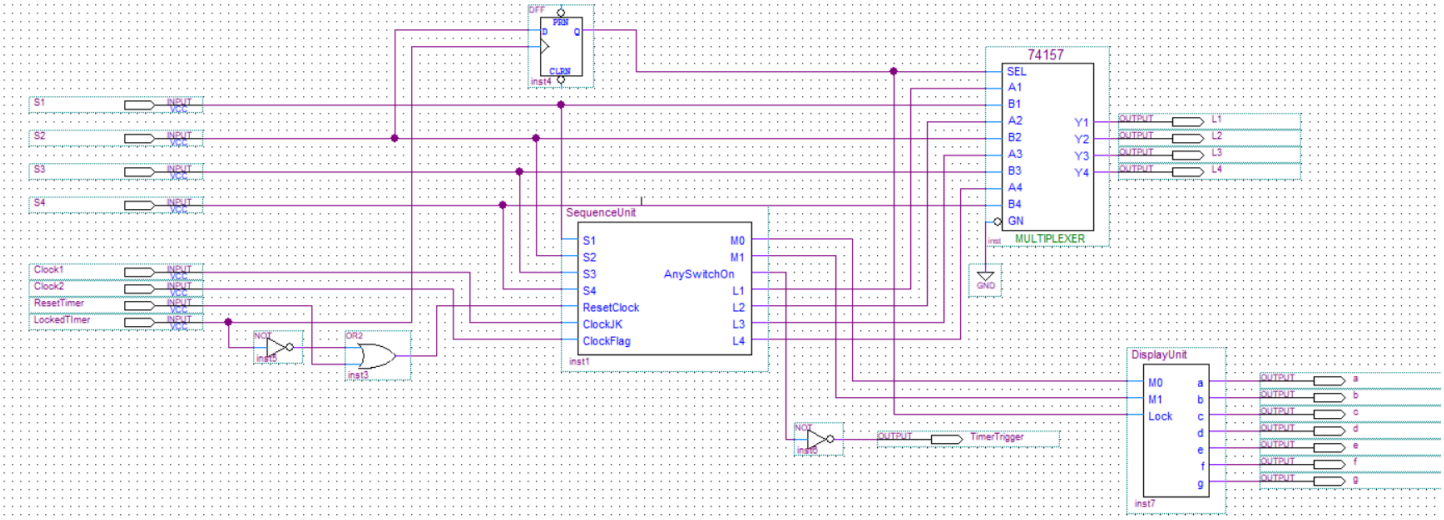

Figure 2: Final design of the finite state machine

This is the finite state machine consisting of the switches, clocks and timers as inputs, and four outputs to activate the lamps with a 7-segment display. A Display Unit is added to display on the 7-segment the mode we are working in. A D flip-flop using the locked timer, a clock and switch 2 as the input D, is the selector of a multiplexer that chooses which set of input is used (locked or normal mode). The system includes also a Sequence Unit block, where sequence detection and mapping of switches to LEDs happen.

Display Unit

The display unit includes a BCD to 7-segment display decoder with an incrementer because the modes start from 1 not from 0, with a lock bit that turns off the decoder and visualize “L” on the 7-segment display.

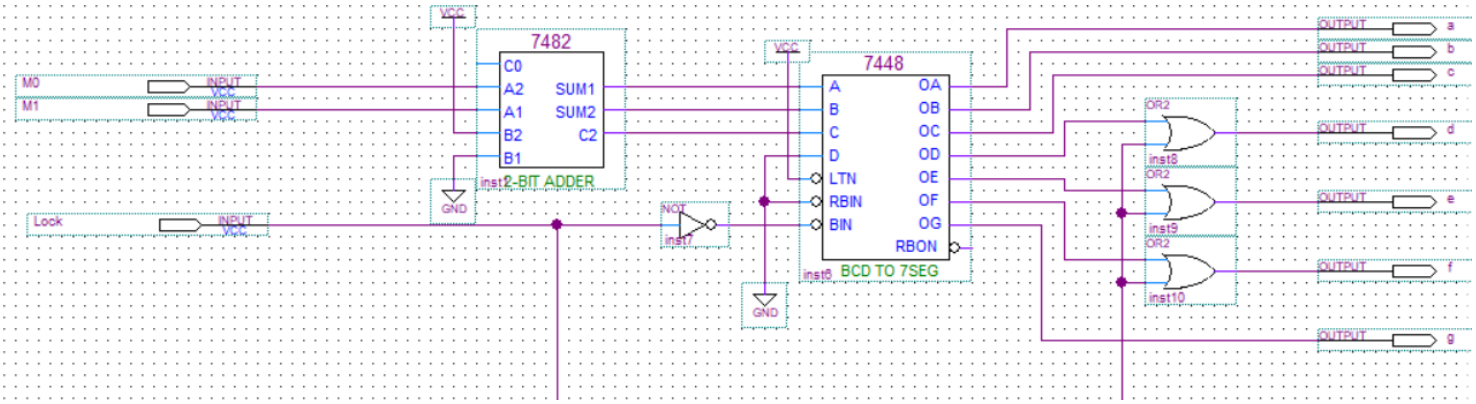

Figure 3: Display Unit design

Sequence Unit

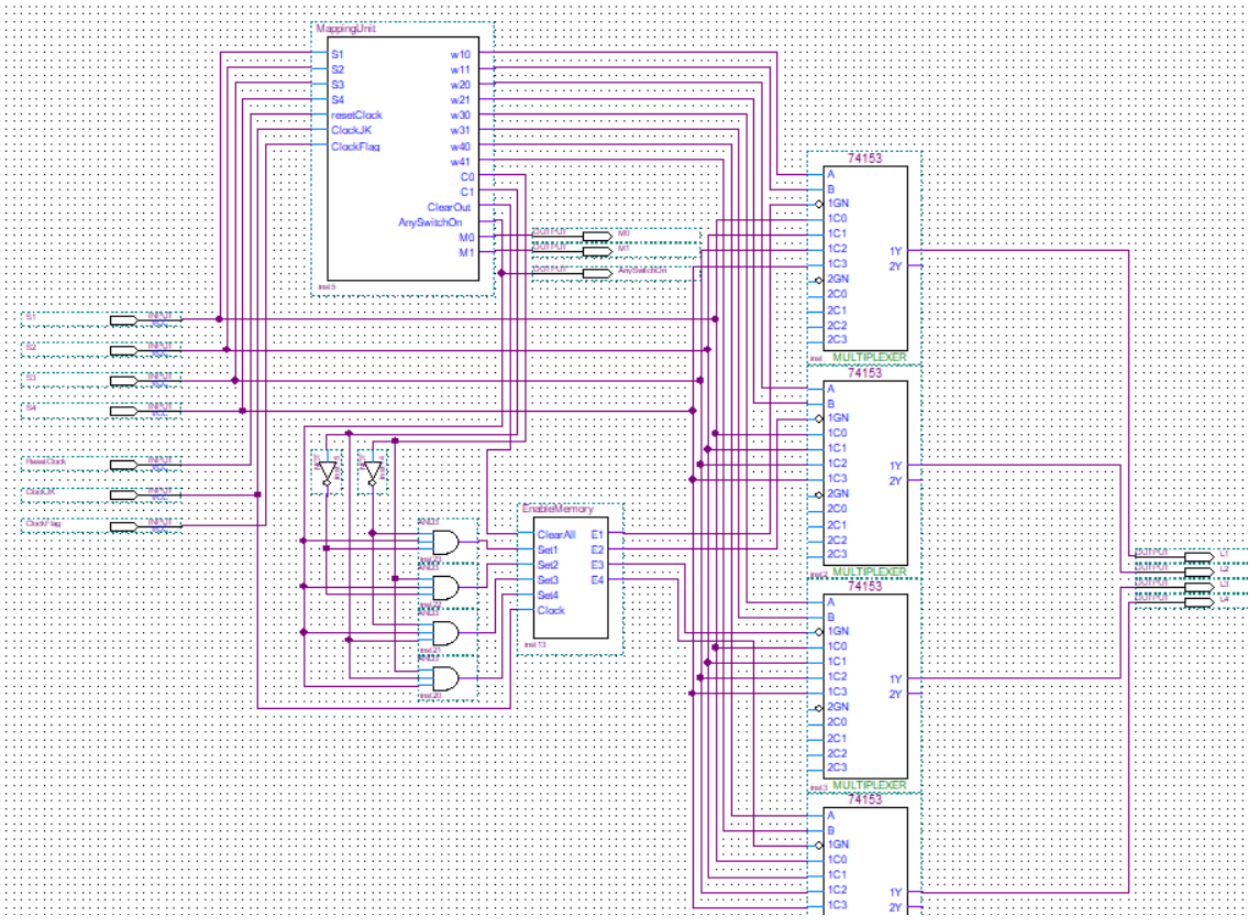

Figure 4: Sequence Unit design

The sequence unit includes a Mapping Unit, where the mapping between the switches and the LEDs is stored, and four multiplexers that decide which switch controls the LED, with an Enable Memory to prevent all the multiplexers from turning when a switch turns on.

The design of the Enable Memory:

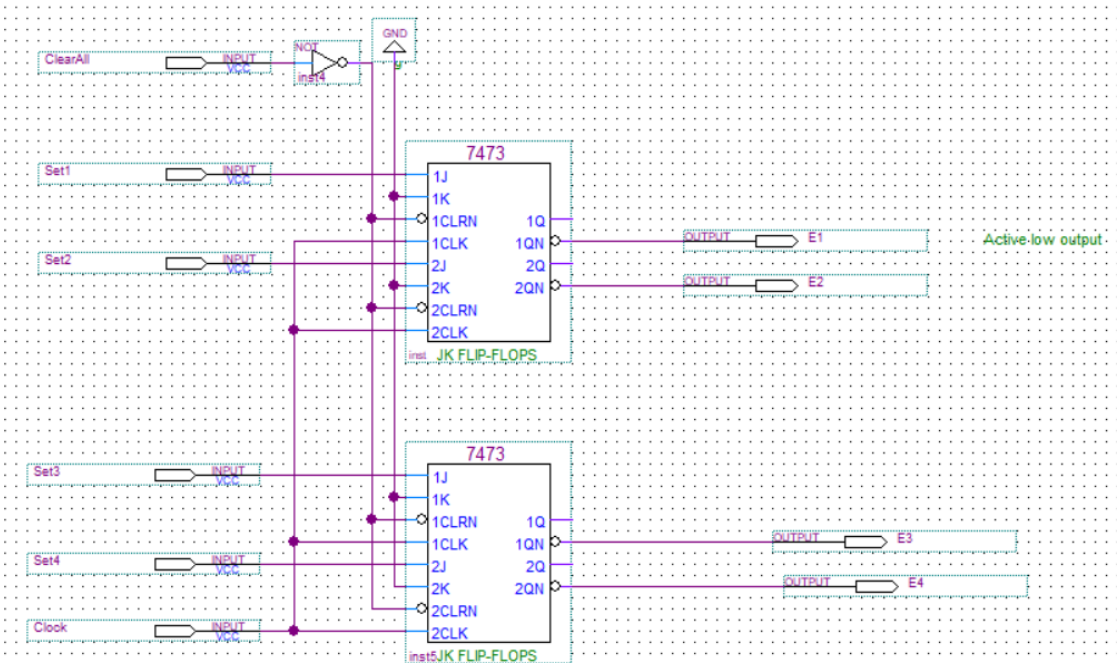

Figure 5: Enable Memory design

The design of the Mapping Unit:

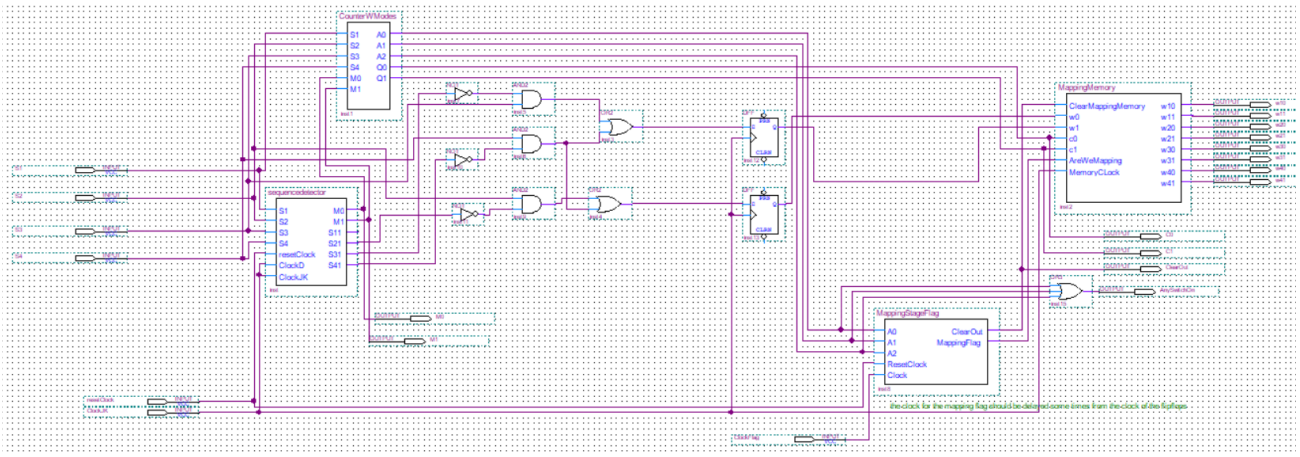

Figure 6: Mapping Unit design

The Mapping Unit includes a Counter W Mode that adds the number switches turned on subtracted by one (switch 1 is 00).

The design of the Counter W Mode:

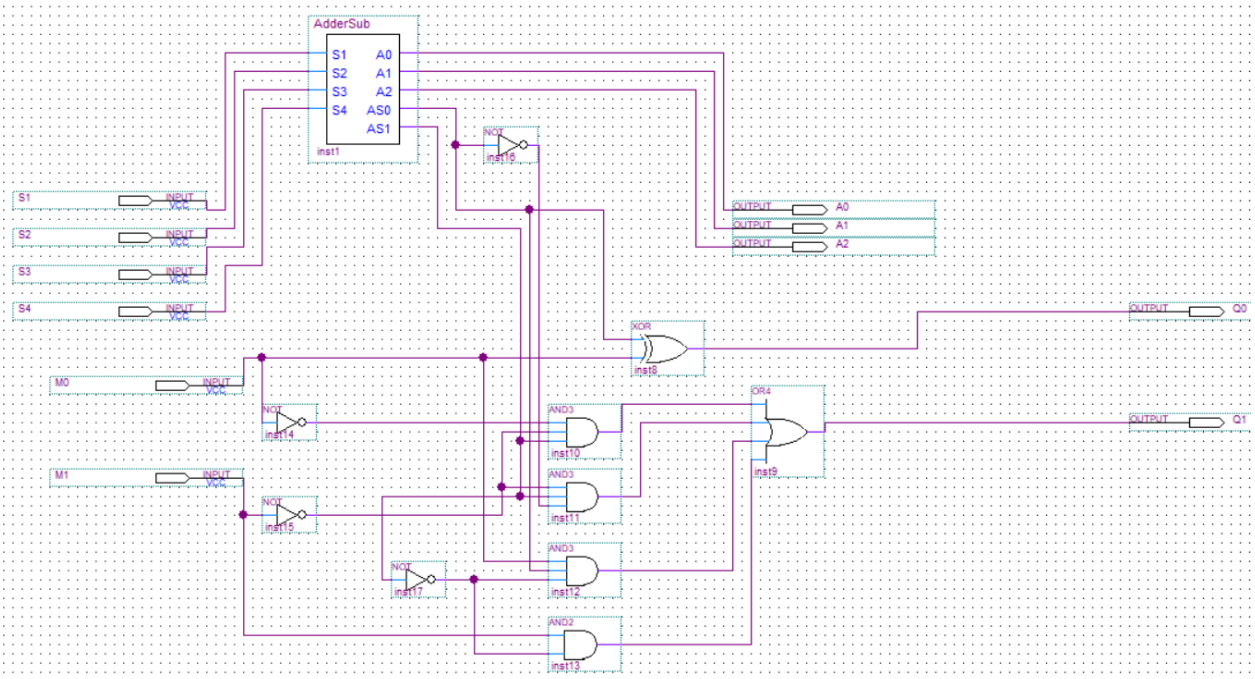

Figure 7: Counter W Mode design

The design of the Mapping Memory

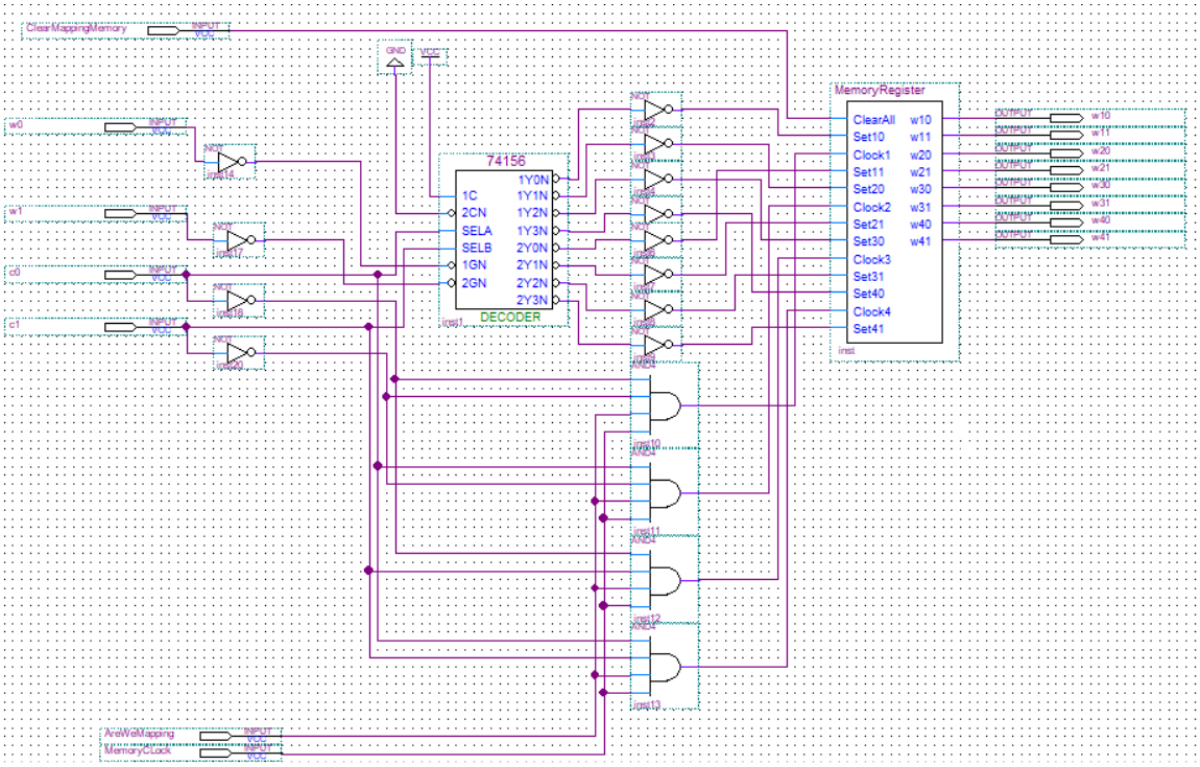

Figure 8: Mapping Memory design

The design of the Memory Register inside the Mapping Memory:

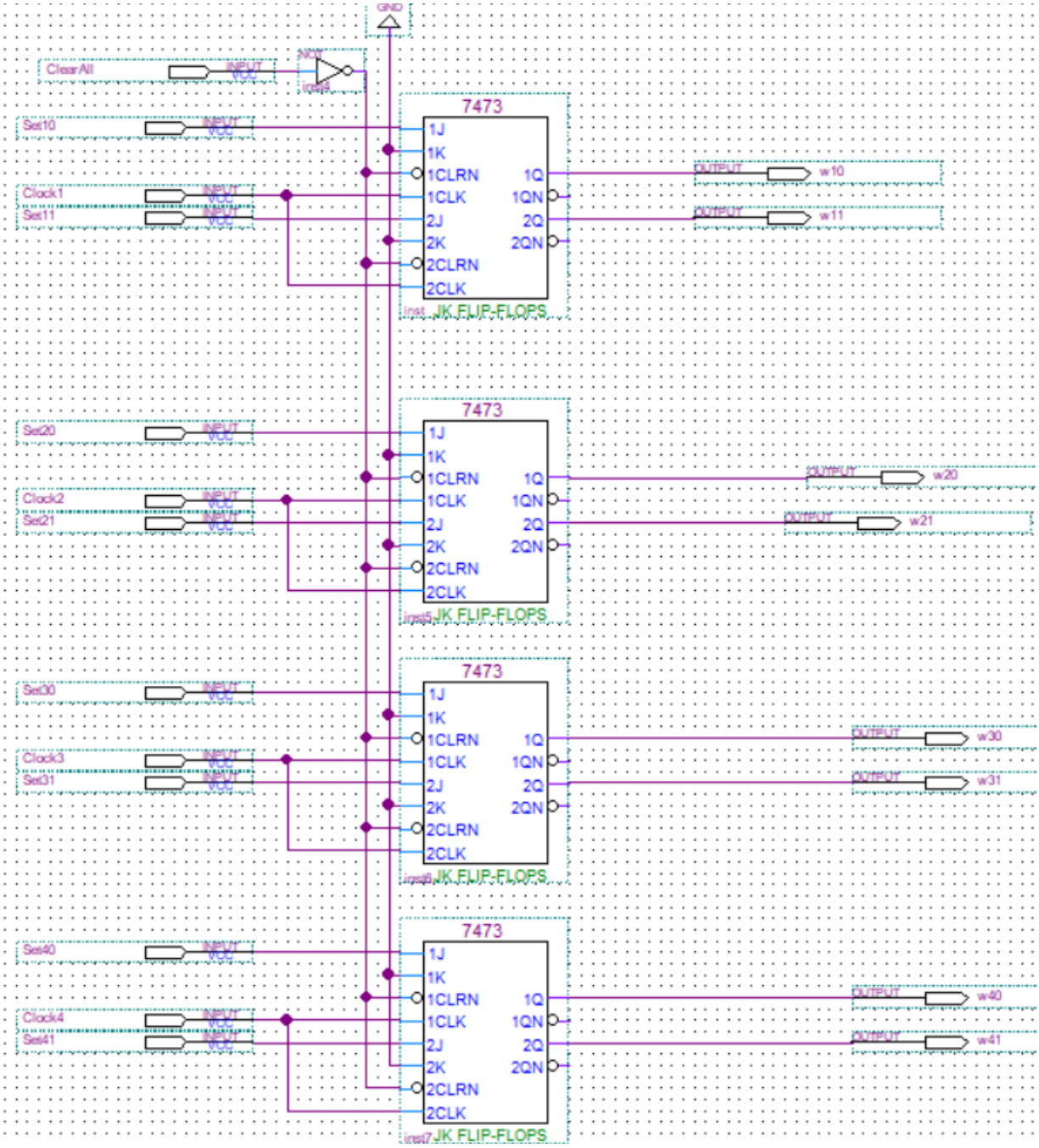

Figure 9: Memory Register design

The memory register includes 8 JK flip-flops (4ICs) to store the mapping of every switch to its corresponding LED.

Now, back to the content of the Mapping Unit, the design of Mapping Stage Flag:

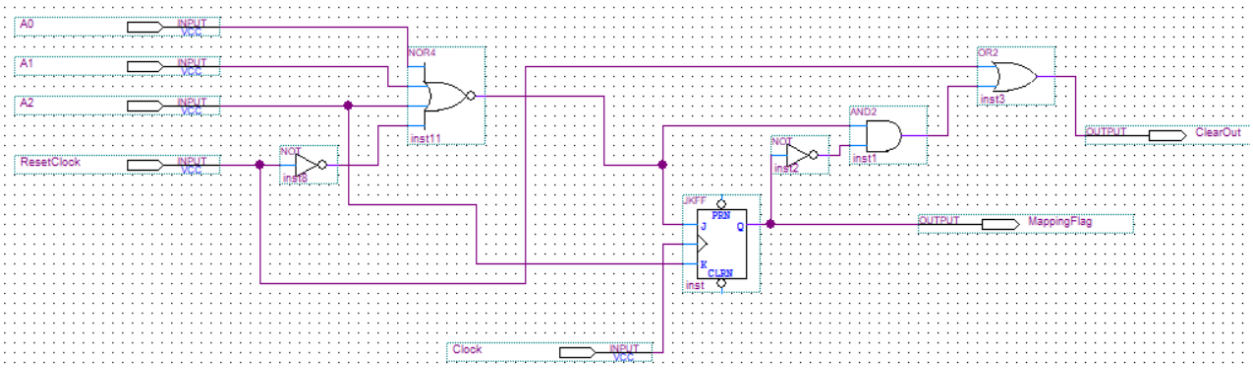

Figure 10: Mapping Stage Flag design

The mapping stage flag indicates if a switch is mapped.

The design of the Sequence Detector

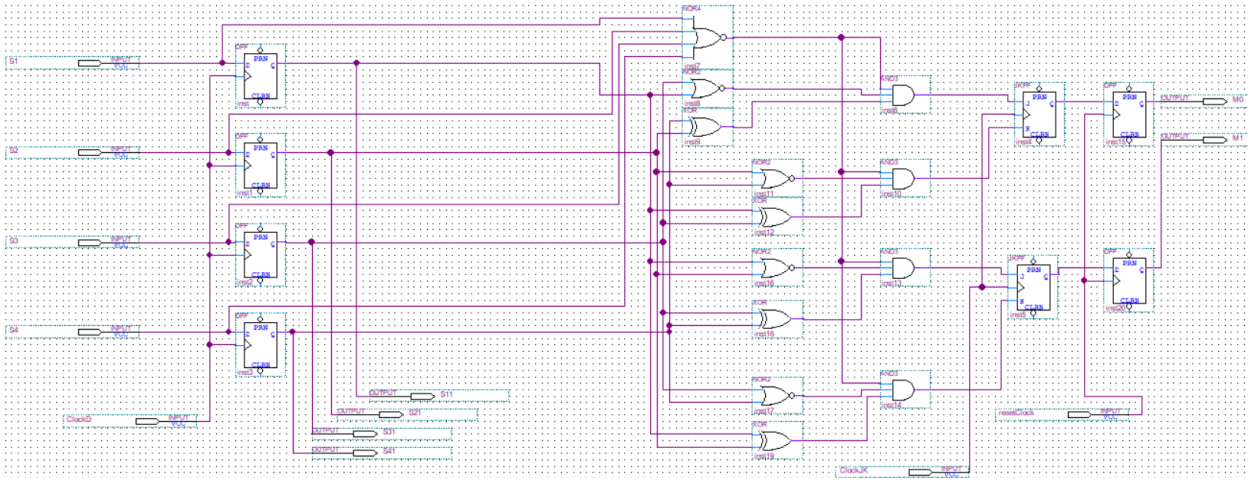

Figure 11: Sequence Detector design

The sequence detector in the system detects which mode we are working in.

So, in the mapping unit, the Sequence Detector indicates the activation sequence, the Mapping Stage Flag determines whether the switch is mapped, the Counter W Mode counts the number of turned-on switches, the Mapping Memory stores the mapping of the LEDs to the switches. Now the output of the Sequence Unit will be connected to the multiplexer for the LEDs to be turned on by the corresponding switch

Timers and Clocks

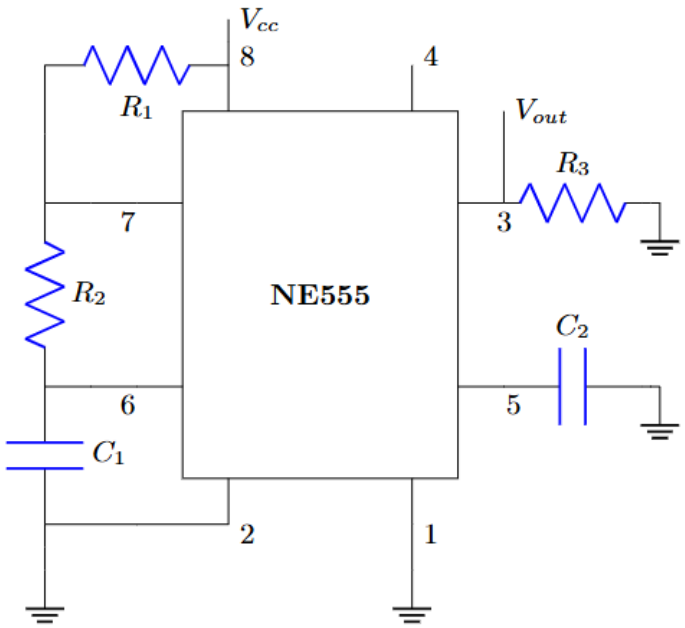

Figure 12: LaTeXed100kHz Clock using 555 timer

The frequency is calculated by:

$$f = \frac{1.44}{(R_1 + 2R_2)C_1}$$

$$f = \frac{1.44}{(720 + 2 \times 360)10^8} = 100kHz$$

However, two clocks are needed in our design, with one having much greater frequency than the other, so we used a frequency divider using 3 D flip-flops to divide the frequency by 8, so, the second clock has a frequency of 12.5kHz.

The locked timer is designed using an RC circuit connected to the ON/OFF switch of the power supply, when it turns on.

The reset timer will turn on when all the LEDs are off. To design it we input the four switches into a NOR gate, and we connect its output to a transistor connected to an RC circuit, which charges the capacitor.

Breadboard Design and Analysis

The physical implementation of the circuit was divided into a breadboard implementation and soldered components on board.

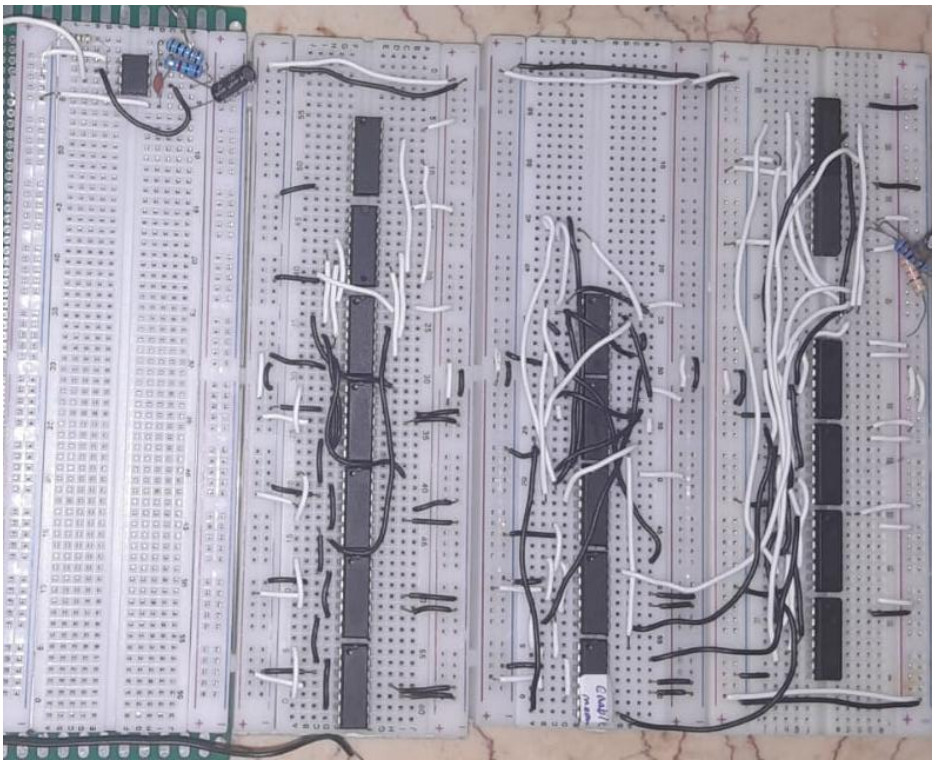

Figure 13: Breadboard implementation

The breadboard implementation covered 555 timer circuit (clock), the memory register of the mapping unit, final stage multiplexers and multiplexer enabler circuit.

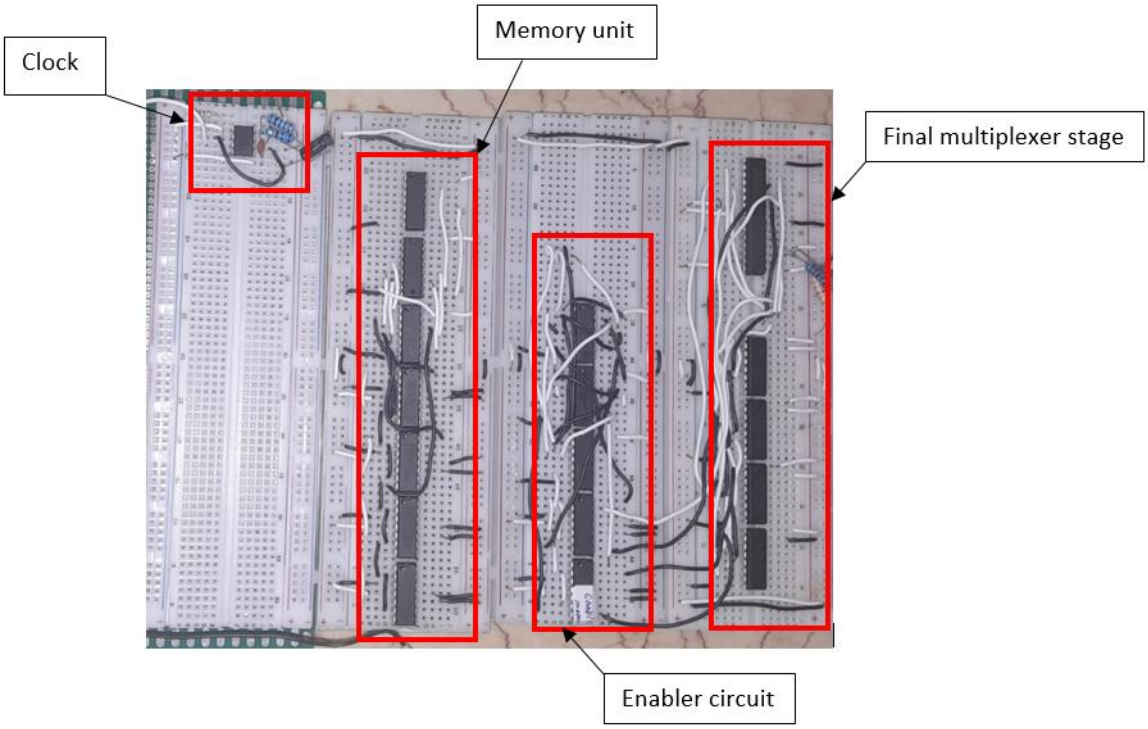

Figure 14: Circuit organization

The soldering covered the sequence detector and the display unit:

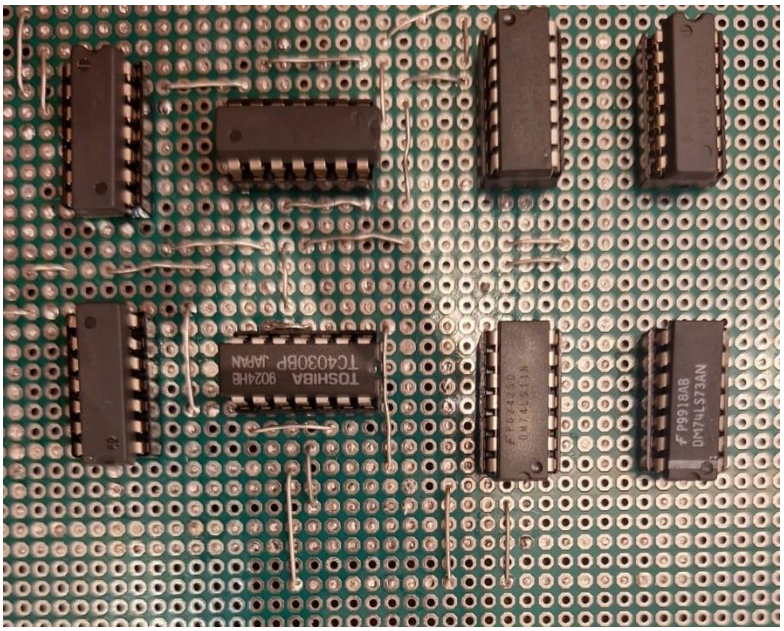

Figure 15: Sequence detector

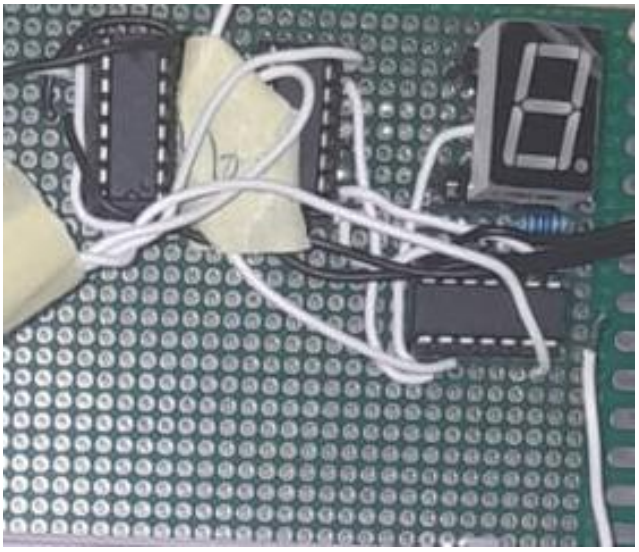

Figure 16: Display unit

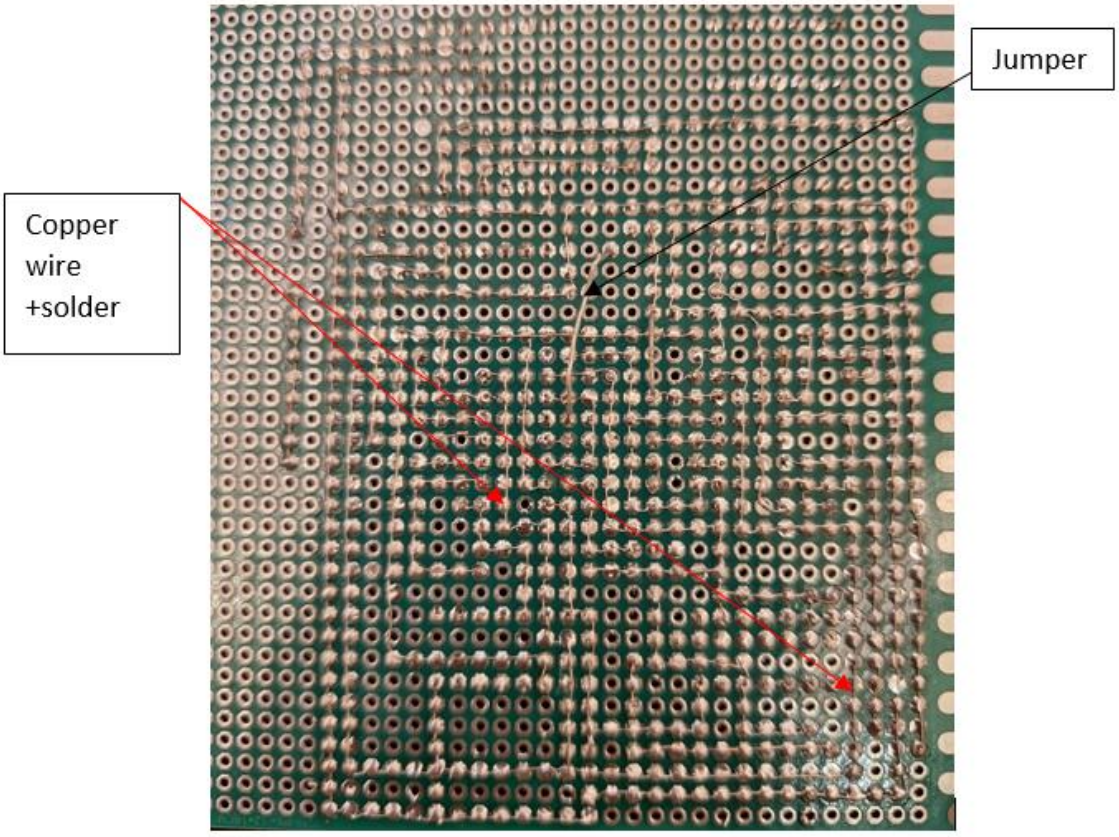

Figure 17: Soldering Technique

Financial Study

N/A

Delay Calculation

N/A

Power Consumption Analysis

N/A

Challenges

The first challenge faced was the delay propagation of the circuit previously discussed with clock speed; the problem was solved by choosing many clocks rather than setting 1 clock to the whole system. Additionally, when soldering the circuit, the soldered points kept disconnecting, the solution was to connect the pins with a very thin copper wire then solder over them, the other problem in soldering was the tight number of soldering holes, the solution was to add jumper metal lines.

The last challenge faced was replacing defective ICs. After soldering, some ICs were not working, so each time the solder had to be removed. Therefore, to facilitate repairing our circuit, the ICs were placed on sockets, and every socket was soldered, this way any IC can be replaced anytime.

Design Advantages

The system was designed in a structured way to facilitate error tracking and debugging as the final design looks simple and readable.

For example, if any problem appears in the display, we just inspect the display unit and when there is a problem in the LED order we check the mapping unit, the abstract design is very clear. Additionally, the display unit will display an “L” if the system has entered the locked state so that the user knows which state he is in. The circuit has no errors and works exactly as wanted.

Conclusion

This project demonstrates the practical application of digital logic circuits, finite state machines, and memory elements to dynamically control the activation of lamps based on user interaction with switches in the logic-controlled board. By successfully implementing dynamic switch-to-lamp behavior controlled by the last switch turned off and integrating advanced functionalities like the capless switch trick and the locked mode, this project illustrates the versatility and creativity that logic circuits can offer. Moreover, the structured approach to analyzing inputs and outputs, designing FSM states, simulating the system on a software deepen out understanding of digital systems.