CMPE243-S01 LOGIC DESIGN

PROJECT REPORT

Project No:1

Design of a Mushroom Light Intensity Controller Using Basic Logic Circuits

Submitted by:
Ebru Hattapoğlu (CE)
Azra Nur Canbey (CE)
Lea Hijazi (CE)

Project Supervisor:
Dr. Seyed Sajad Ahmadpour,

Project Mentors:
Assoc.Prof. Nima Jafari Navimipour
Assoc.Prof. Taner Arsan
Assist. Prof. Ilktan Ar

Faculty of Engineering and Natural Sciences
Kadir Has University

Fall 2024

TABLE OF CONTENTS

1 INTRODUCTION

Light affects how plants grow and produce. It powers photosynthesis, which helps plants grow and release oxygen. Without enough light, plants will grow poorly and crops become smaller.

Since different colors of light have different effects on plants, our project will consist of 3 colors to help mushrooms grow when there is no sunlight [1]. Mushrooms don't use photosynthesis but still need light to grow properly during the fruiting stage. That is why we are developing the circuit based on these following colors and their roles:

- Red light: helps develop specific pigments and compounds.
- Full spectrum: light supports overall growth and health.
- White light: ensures balanced growth by providing a complete spectrum.

This project focuses on designing a Mushroom Light Intensity Controller using logic circuits to manage these lights and help mushrooms grow better at night [2].

2 APPLICATION FOR MUSHROOMS

Mushrooms are important in food, health, and the environment. Their growth depends on a specific management, including light control [3].

2.1 Mushrooms as Food

Mushrooms have high nutritional benefits, they are widely consumed because they are beneficial for human and animal bodies. They are rich in protein, vitamins, and minerals, and they add flavor to dishes.

2.2 Mushroom as Medicine

Many mushrooms have medicinal benefits because they contain compounds that boost immunity, fight inflammation, and support our health.

2.3 Environmental Benefits

Mushrooms contribute to the ecosystem, they decompose organic matter and enrich the soil. They also clean pollutants in natüre.

3 LOGIC CIRCUIT DESIGN ANALYSES

This part explains the steps followed while designing the circuit. First of all, the truth table was used to determine which input combinations of red, full spectrum and white lights work. This truth table formed the basic logic of the circuit and showed which colored lights work on which inputs. In addition, the Karnaugh map of each light color was created based on the data in the truth table. The given minterm values were transferred to this table and visualized. Then, grouping was done for each light color on this map and simple equations were obtained for each light.

3.1 TRUTH TABLE

In the truth table, it is found which light is active in which inputs A, B, C, D. The cases in which each light is active are marked as "1" in the relevant column of the table. This table is a basic table for testing both the logical design of the circuit and the accuracy of simulations.

Table 3.1.1 The truth table shows whether the red, full spectrum, and white light outputs are active (1) or passive (0) for all inputs

A	B	C	D	RED	FULL SPECTRUM	WHITE
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	0	1	1	1	1	0
0	1	0	0	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

3.2 KARNAUGH MAP (K-MAP)

The Karnaugh map showed the grouping of minterms to obtain simplified Boolean expressions for the colors red, full spectrum and white. In this way, the circuit became simpler and it was seen more clearly in which combinations the lights were on. For example, by grouping the "1"s in the K-map for red light, the expression F = $C + BD' + AD'$ was found and it was seen more clearly in which cases the red light was on. Similarly, in full spectrum, minterms were grouped and F = $CD + BC + A$ was obtained and simplified. In the same way, minterms were grouped for white light and F = $BD + BC + A$ was found. In this way, as in other colors, it was determined more clearly in which inputs the light was on or not. As a result, thanks to the Karnaugh map, input combinations were easily grouped, and simplified logic expressions were obtained [4]. In this way, it was seen more clearly in which situations the lights were on.

Red Light

Minterms: 4, 7, 15, 2, 3, 6, 12, 8, 11, 14, 10

Table 3.2.1 The Karnaugh map shows the grouping of minterms to obtain simplified Boolean expressions for red light

AB/CD	00	01	11	10
00	0	0	1	1
01	1	0	1	1
11	1	0	1	1
10	1	0	1	1

$$F(A,B,C,D) = C + BD' + AD'$$

Full Spectrum

Minterms: 14, 6, 12, 13, 10, 15, 11, 3, 7, 8, 9

Table 3.2.2 The Karnaugh map shows the grouping of minterms to obtain simplified Boolean expressions for full spectrum

AB/CD	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	1	1	1	1
10	1	1	1	1

$$F(A,B,C,D) = CD + BC + A$$

White Light

Minterms: 6, 12, 8, 14, 5, 13, 9, 7, 10, 11, 15

Table 3.2.3 The Karnaugh map shows the grouping of minterms to obtain simplified Boolean expressions for white light

AB/CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	1	1	1	1
10	1	1	1	1

$$F(A,B,C,D) = BD + BC + A$$

3.1.1 Grouping

In the Karnaugh map, grouping was done by bringing together cells with the value "1". In other words, cells with 0 on the K-map are ignored and only cells with 1s are grouped to create more simplified logic expressions. In order to simplify the circuit, cells that were close to each other were grouped. These cells were grouped in multiples of 2, such as 1,2,4,8. Also, creating large groups made the circuit more understandable. In addition, during grouping, only the places with 1 were looked at and grouped accordingly. Finally, in Table 3.2.1 and Table 3.2.2 and Table 3.2.3, grouping operations are indicated using different colors.

3.1.2 Simplification

Thanks to simplification, the "1" values marked in Karnaugh maps were transformed into simpler and more efficient Boolean expressions. This process was very important to make the circuit more efficient. The aim of simplification was to use fewer logic gates while creating the circuit and thus make the circuit more efficient. Thanks to this method, larger groups were created by grouping adjacent "1" cells in the Karnaugh map. Thanks to this grouping, simplified Boolean expressions containing fewer terms were obtained. In other words, this process eliminated unnecessary terms while designing the circuit and kept only the necessary ones, making the logic simpler. As a result, thanks to this simplification, an expression corresponding to 11 minterms in the truth table was reduced to only three terms. For example, red light with minterms of 4, 7, 15, 2, 3, 6, 12, 8, 11, 14, 10 has been transformed into the equation $F(A,B,C,D) = C + BD' + AD'$ thanks to simplification and has become more understandable. In the same way, when the full spectrum minterms are 14, 6, 12, 13, 10, 15, 11, 3, 7, 8, 9, the equation $F(A,B,C,D) = CD + BC + A$ has been obtained through simplification. Similarly, white light with minterms of 6, 12, 8, 14, 5, 13, 9, 7, 10, 11, 15 has been transformed into the equation $F(A,B,C,D) = BD + BC + A$ through simplification [5].

4 SIMULATIONS OF THE LOGIC CIRCUIT

In this part, simulations made using Logism are shown in order to check the accuracy and operation of the designed logic circuits. Simulations were made separately for each light color in red, full spectrum and white using Logism. According to the simulations, the accuracy of simplified Boolean expressions was also tested, and it was observed that they gave correct results. As a result, it was shown that each Boolean expression was simulated correctly, and the equations created with the minterms used in this direction were also correct. [4]

4.1 The First Simulation on Logism

At this part, the red light circuit was tested independently. By using a simplified Boolean expression for the red light, it was ensured that it was active only at certain input combinations. In the circuit designed with basic logic gates (AND, OR, NOT) on Logisim, it was observed that the red light circuit was passive outside the specified minterms.

- Boolean expression of red light = $F(A,B,C,D) = C + BD' + AD'$
- Minterms = 4, 7, 15, 2, 3, 6, 12, 8, 11, 14, 10

During the simulation, it was verified that the red light was on only at these minterms when each input combination was applied. This phase proved that the theoretical design of the red light circuit was correct and that the simplified expression worked as expected.

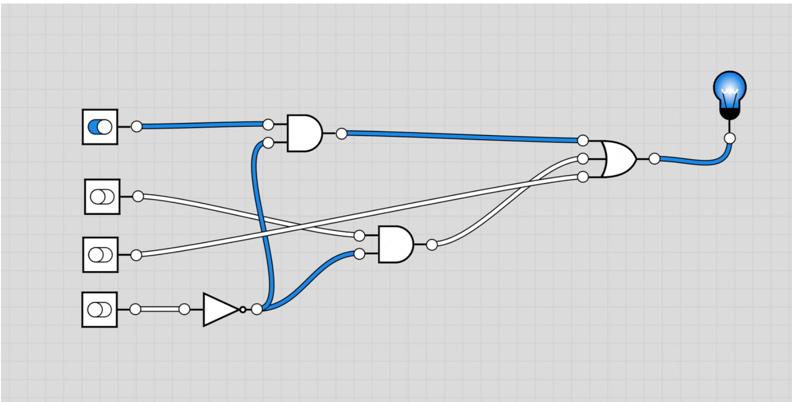

Figure 4.1.1 Shows the independent testing of the red light circuit on Logisim

4.2 The Second Simulation on Logism

In the second simulation, the full spectrum light circuit was tested independently. In this stage, a simplified Boolean expression was used to check whether the full spectrum light was active only in the input combinations assigned to it.

- Boolean expression of full spectrum = $F(A,B,C,D) = CD + BC + A$
- Minterms = 14, 6, 12, 13, 10, 15, 11, 3, 7, 8, 9

The full spectrum light circuit was active only for these input combinations and inactive for all other combinations. This simulation verified that the theoretical design of the full spectrum light worked correctly and that the simplified expression was implemented appropriately.

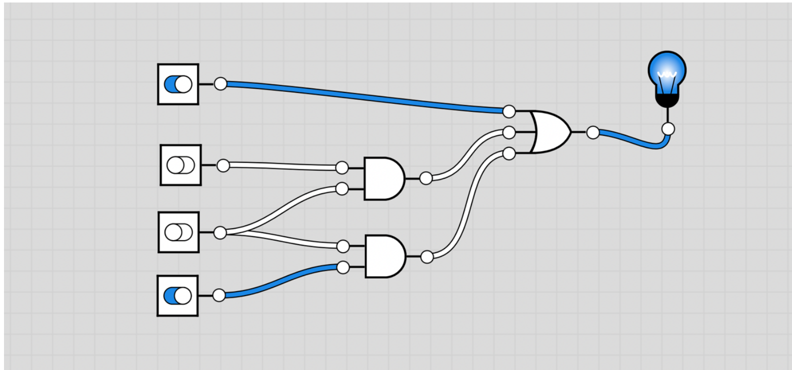

Figure 4.2.1 Shows the independent testing of the full spectrum light circuit on Logisim

4.3 The Third Simulation on Logism

In the third simulation, the white light circuit was tested independently. Again, using a simplified Boolean expression, it was verified that the white light only works for certain input combinations.

- Boolean expression of full spectrum = $F(A, B, C, D) = BD + BC + A$
- Minterms = 6, 12, 8, 14, 5, 13, 9, 7, 10, 11, 15

In this simulation, it was observed that the white light was active in the input combinations assigned to it and remained inactive in other cases. This simulation confirmed that the theoretical design of the white light worked correctly and that the simplified expression was implemented appropriately.

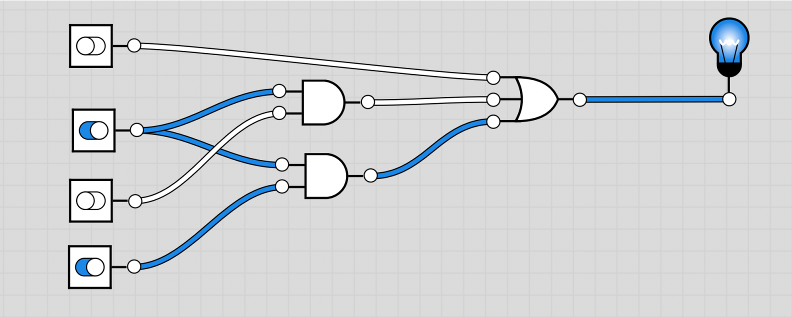

Figure 4.3.1 Shows the independent testing of the white light circuit on Logisim

4.4 Comparison of Components Used in Simulations

Table 4.4.1 Compares the type and number of logic gates used in each simulation

	First Simulation	Second Simulation	Third Simulation
COMPONENTS	2 AND Gates 1 OR Gates 1 NOT Gates	2 AND Gates 1 OR Gates	2 AND Gates 1 OR Gates

This table, Table 4.4.1, shows the type and number of gates we used to create the circuit in each simulation. In the first simulation, we used both AND, OR, and NOT gates for the red light, while in the other simulations, fewer gates were needed. This shows that the circuit became simpler through simplification. Specifically, by using minterms and simplified Boolean expressions, we achieved the same function with fewer gates in the full-spectrum and white-light circuits.

5 IMPLEMENTATION OF THE LOGIC CIRCUIT

The goal of this project was to design and implement a circuit that controls a 7-segment display and an LED system (red, white, and full-spectrum) using logic gates and Karnaugh maps. The implementation was done in two parts: the first based on a logic schema, and the second using Karnaugh maps.

Components Used:

- 4 AND gate ICs
- 2 OR gate ICs
- 2 NOT gate ICs
- 7-segment display
- Breadboard and connecting wires.

Table 5.1 Comparision of Implemenntation in terms of Cost Efficiency

	WIRES	COMPONENTS	COST(WIRES+ COMPONENTS)
First Implementation	49	2 AND Gates 1 OR Gates 1 NOT Gates (In total, 4 Gates)	2.4$+0,85$=3,25$
Second Implementation	52	2 AND Gates 1 OR Gates 1 NOT Gates (In total, 4 Gates)	2.4$+0,88$=3,28$

As you can see in the Table 5.1, first implementation and second implementations were both have same cost as the number of wires and gates are the same. First implementation slightly more cost-efficient, due to number of wires that were used but there is a small difference that is not enough to be considered.

5.1 First Implementation

The first implementation was centered about controlling a 7-segment display with the use of 2 AND gates, 1 OR gate and a NOT gate. The inputs (A, B, C, D) were connected according to the given logic diagram and equations. The logic was structured to represent the tens digit and each segment of the display was controlled accordingly.

The ICs were placed in the right side of the breadboard. Inputs were connected to the logic gates, and outputs were connected to the 7-segment display.First the Tens has been connected and the empty ports of the Tens side of the segmentetion (left-side) were connected to the ground. The A, B, C ,D outputs were connected to the other side of the segmentation part All of the ICs were provided with 5V and ground connections to avoid any possible operational problems.

While putting it together, the numerous wires made it hard to follow connections which resulted in some initial errors displayed in the output.

The primary issues were identified and resolved during the first stages of the tests. Incorrect connection orders of wires were most likely the reason why mistakes were displayed. Improvement of performance was noticed after diagnosis and thorough examination of all the links. The circuit functioned accurately with the display showing the correct tens digit as required.

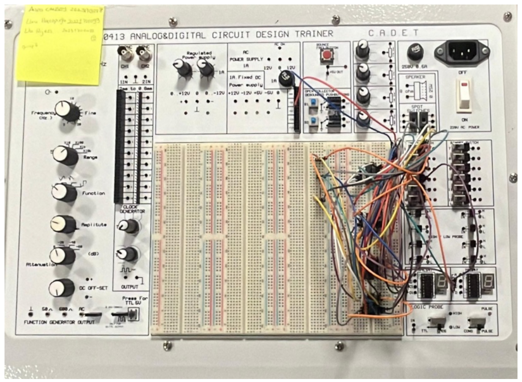

Image 5.1 The Photo of the First Implementation

5.2 Second Implementation

The second implementation used a same number of ICs but followed a different design approach based on Karnaugh maps. Karnaugh maps were used to simplify the logic equations for the LEDs (red, white, and full spectrum). The logic was implemented to control the LEDs according to specific input conditions.

ICs were placed on the left side of the breadboard and for the LEDs outputs, output section were used from the breadboard, with connections made according to the simplified logic obtained from Karnaugh maps. All of the ICs were provided with 5V and ground connections to avoid any possible operational problems. The inputs were routed in a systematic manner so as to enhance clarity during the testings.

The complexity of the circuit increased due to the numerous connections. As the wires got tangled and made it wired wrong, it took five tries to get the circuit configuration correct.

After multiple iterations, the connections were corrected. The LEDs (red, white, and full-spectrum) successfully lit up according to the given input numbers.

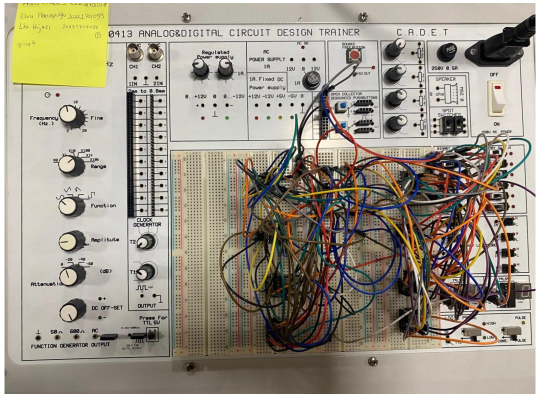

Image 5.2 The Photo of the Second Implementation

5.3 Results

Both implementations were successfully completed. The first implementation correctly displayed the tens digit on the 7-segment display, while the second controlled the LED system as per the input logic derived from Karnaugh maps.

The major concern, however, was that of wires used within the circuits provided since it made the breadboard very cluttered. Organizational improvements such as color-coded wires would enhance future projects. One of the benefits of this project was the ability to work on building circuits with logic gates, implementing their designs with Karnaugh maps, and fixing breadboard mistakes.

That is, using digital simulation tools beforehand would prove beneficial in understanding better the circuit and therefore reducing the window for error during wiring.

6 DISCUSSION

The Mushroom Light Intensity Controller had two main configurations: the first design closely focused on 7-segment display controls while the second one concentrated on red, white and full color LEDs. In cost effectiveness, complexity and efficiency both configurations were also detailed.

The introduction of the truth table gave way to an easy logic circuit designing process. It made it possible to understand and follow which input combinations would turn on which lights and therefore, what the next steps would be. The transformation and determination of inputs A, B, C, D to outputs red light, full spectrum light and white light built the logic in an orderly logical and testable way. This step also enabled the user to test the firmness of the designed circuit to logical functions by ensuring that actual outputs were within the objectives of the project

The simplified Boolean expressions of the logic table were possible because of Karnaugh maps or K-maps. K-maps helped in ‘picking up’ and ‘dropping’ of the needed terms for any given output which made the number of term in each output considerably lower and therefore made possible a practical and effective circuit. Such examples include; the rution for red light logic in making less drastic terms therefore less gates were used in the implementation.

K-maps also helped to reduce the number of gates in these equations. These simplifications helped in reducing at the same time the number of gates used and worked toward the efficiency of the circuit as a whole. K-maps clarity was more useful at the stage of implementation since the possibility of making errors and creating unnecessary havoc in the circuit was greatly minimized.

Logisim being circuit simulation software was very useful in providing an intermediary experience between design on paper and construction in real life. Since doing the logic circuit on computer, the chances of making mistakes by braiding the wrong wires was eliminated. This red flag ensured that the circuit design was valid as well as reduced time wasting in careless fixing while putting the circuit together.

While implementing, both implementations faced challenges related to wiring complexity. With 101 wires used in total, the breadboard became cluttered, making troubleshooting difficult. This was particularly evident in the second implementation, where

due to the large number of connections it took five turns to wire the circuit correctly. Either the systematic wiring or using of colored cables would have helped eliminate the confusion experienced as well as enhance the efficiency of the assembly activity.

In spite of the configurations which were mostly warring towards the beginning, eventually both implementations passed. The first implementation managed to properly display the tens place of the number reading from the 7-segment display as it was intended. The secondary implementation was also successful in LED control; the red, white and the RGB lights activated as per the simplified layout based on the simplified Boolean expressions. [5]

7 CONCLUSION

The Mushroom Light Intensity Controller project applied digital logic design to improve a very important aspect of the agricultural systems. We controlled red, full spectrum, and white light intensities by implementing a combinational logic circuit, to ensure the optimal conditions for mushroom growth during nighttime in their fruiting stage. Since each light has a role, combining those 3 lights led to great outcomes, done by simplifying boolean equations using Karnaugh maps, simulating the circuit in Logisim, and testing it on a breadboard. This project demonstrated mushroom lighting needs and also improved our in logic design skills, problem-solving, and practical application. Additionally, this project showed us how digital logic systems can contribute in agricultural productivity and support other innovations in controlled environment agriculture.

REFERENCES

[1] A. Marulanda, J.I. Querejeta, M. García-Hernández, et al., "Influence of light on growth and productivity of mushrooms," Mycosphere, vol. 8, no. 1, pp. 1–9, 2017.

[2] S. S. Ahmdpour, *CMPE243 Project Slides: Design of a Mushroom Light Intensity Controller*, Kadir Has University, 2024.

[3] Y. Zhan, Y. Lu, X. Li, et al., "Effects of different light quality on the growth and development of edible mushrooms," Front. Microbiol., vol. 11, p. 1250, 2020.

[4] Jafari Navimpour, N., "CMPE243 'Logic Circuit Design' Lecture Slides," Kadir Has University, 2024.

[5] M. Mano, Digital Design, 2nd ed., Prentice Hall Int. Inc., 1991.