IT 208 Digital Logic Art Project

Arnela Sokolić

November 2024

1 Abstract

This project explores the concept of digital art through the use of logic circuits in Logisim. By applying basic digital logic principles, we created visually engaging patterns and animations using LEDs and simple logic gates. This project demonstrates how fundamental elements of digital logic, such as AND and NOT gates, can be creatively combined.

2 Overview

In this project, we combined digital logic with creativity to make six unique designs using LEDs in Logisim. We created a flower, a mandala, a heart, a robot, the text "CO 208" and one simple shape. Each design uses simple logic gates, mostly AND and NOT gates, to control the LED patterns. Overall, this project shows how simple logic gates can be used creatively to make artistic patterns with digital circuits.For the simple shape and one version of the heart design, we went a step further by defining inputs and outputs, along with a truth table.

The clock signal is essential for adding movement and dynamic effects to our designs. For instance, in the mandala and robot designs, the clock alternates between high and low states, toggling different parts of the design on and off in a rhythmic sequence. By adjusting the clock frequency, we can control the speed of these transitions, making the animations appear slower or faster.

3 Metodology

For each design, LEDs are arranged on a grid, and logic gates are used to control their on-off states. We employed AND and NOT gates, as well as clock signals, to create dynamic lighting effects. Each design is explained in detail, including the logic used and the intended visual effects. The designs were created and tested in Logisim, a tool for digital circuit simulation.

4 Design

This section goes through each of the five designs We created in Logisim: a flower, a mandala, a heart, a robot and the text ”CO 208”. Each design was built by arranging LEDs in specific patterns and connecting them to basic logic gates to achieve different visual effects.

4.1 Flower

The flower design uses LEDs arranged in a circular pattern with additional LEDs to represent petals and leaves. We created two versions to show different lighting effects.

In the first version of the flower design, when the clock is switched on, one set of LEDs lights up, and when it switches again, a different set of LEDs lights up. This alternating effect is achieved using AND and NOT gates: some LEDs are connected to AND gates, while others go through NOT gates, causing them to light up in turns.

The clock signal toggles between high (1) and low (0). One set of LEDs (Set A) is directly connected to the clock signal, meaning that when the clock is high, these LEDs light up. In contrast, the second set of LEDs (Set B) is connected through a NOT gate. When the clock is high, the NOT gate outputs low, turning off Set B’s LEDs.

When the clock signal goes low, the NOT gate outputs high, illuminating Set B’s LEDs. This configuration allows for a seamless alternating lighting effect, where one set of LEDs is illuminated while the other remains off, creating a dynamic visual display.

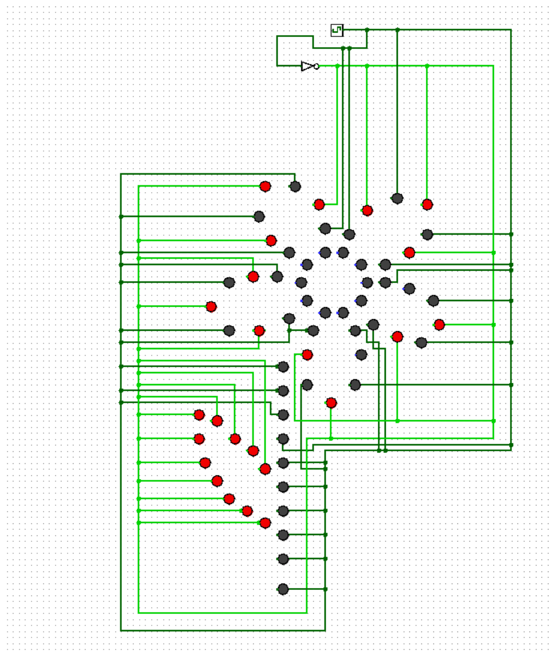

Figure 1: First version of Flower Design in Logisim

In the second version of the flower design, we aimed to create a pattern where multiple LEDs light up at the same time but in different areas of the design. To achieve this effect, we utilized AND gates to control multiple sets of LEDs simultaneously, allowing certain LEDs to be active together while others remain off.

Specific patterns of LEDs are activated in different areas. For example, one AND gate control LEDs in the petals, while another controls LEDs in the center of the flower. When the clock signal is high, and the control inputs for both AND gates are also high, both sets of LEDs illuminate at the same time.

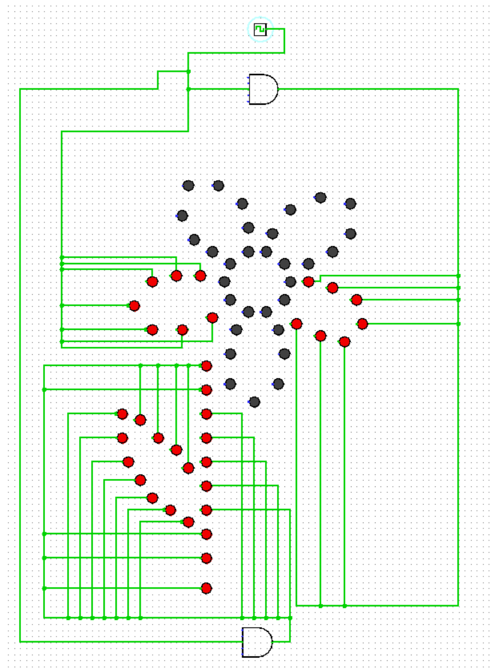

Figure 2: Second version of Flower Design in Logisim

4.2 Mandala Design

In our "Mandala" design, we arranged LEDs in a symmetric, circular pattern that resembles a traditional mandala. At the center, there's a core circle of LEDs that creates a focal point, surrounded by several branches that extend outward to smaller circles, forming a layered, radiating look.

For the Central Circle, all LEDs light up together, creating a bright central focus. We used AND gates to control this section, allowing for precise and consistent illumination.

Branches and Outer Circles are connected to this central core, with each branch ending in a smaller LED circle. These "petals" are also controlled by AND gates, which allow us to selectively light up sections of the mandala for a more intricate, petal-like effect.

Using a clock signal, we added an alternating pattern that makes the LEDs light up in a rhythmic sequence. This pulse-like effect brings the mandala to life, as the branches and outer circles light up in a soothing, wave-like pattern. This design highlights how logic gates and timing signals can create visually complex and dynamic designs from simple components.

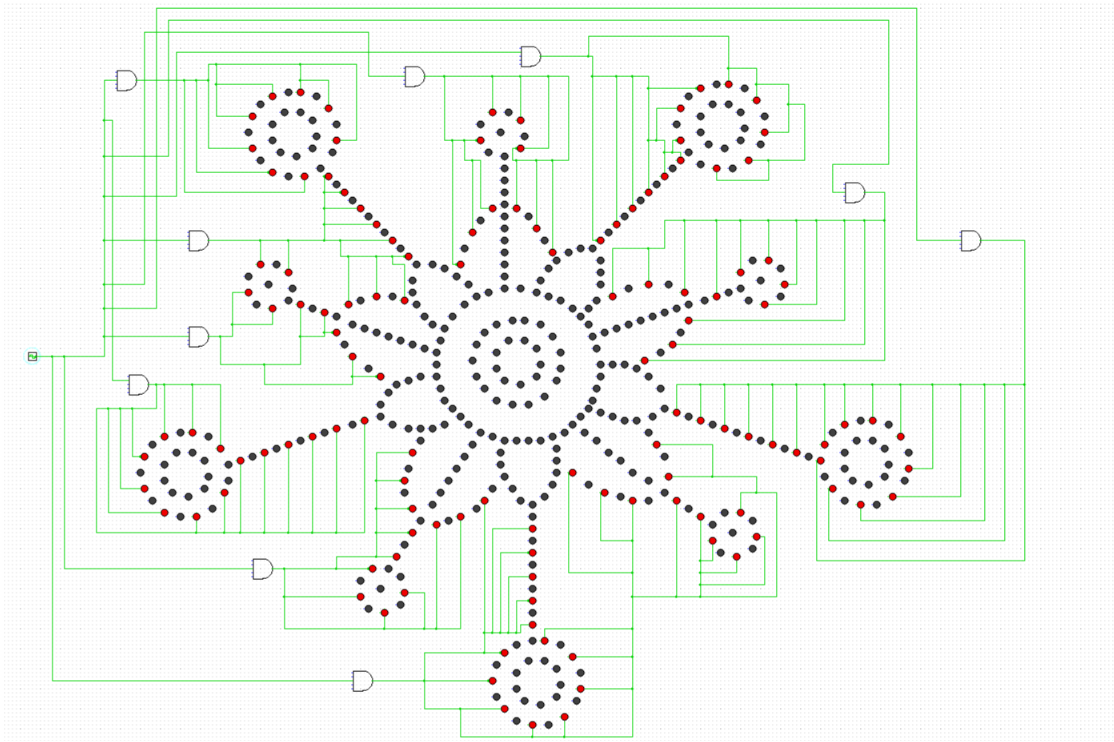

Figure 3: Mandala Design in Logisim

4.3 Heart Design

The heart design has three versions, each showcasing unique lighting effects using AND and NOT gates, similar to the flower design.

In the first version of the heart design, we implemented a pattern where some LEDs light up while others remain off. This alternating effect is controlled by a clock signal and achieved using AND and NOT gates. When the clock is high, one section of the heart lights up, while the opposing section remains dark. Conversely, when the clock switches low, the NOT gate allows the opposite section to illuminate, creating a visually striking effect.

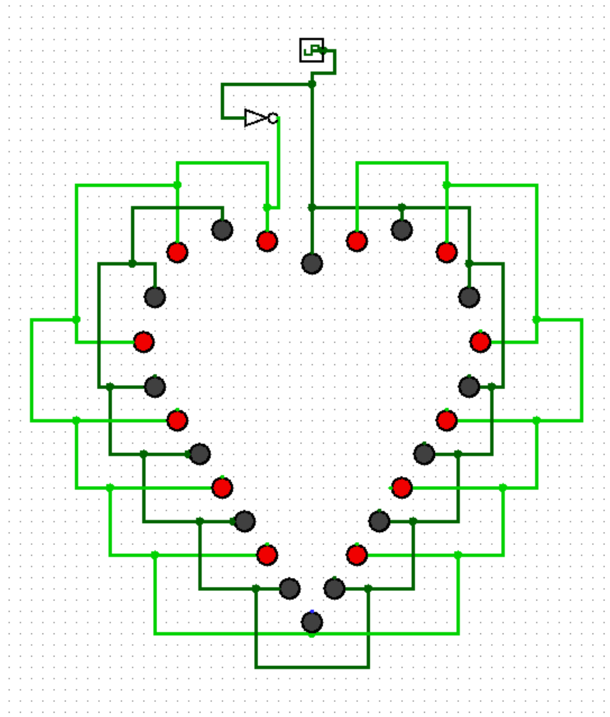

Figure 4: First version of Heart Design in Logisim

The second version of the heart design creates a more harmonious lighting effect, where both the upper and lower parts of the heart are illuminated simultaneously. This is achieved by utilizing AND gates, allowing specific sets of LEDs in both sections to light up together when the clock signal is high, resulting in a balanced and aesthetically pleasing display.

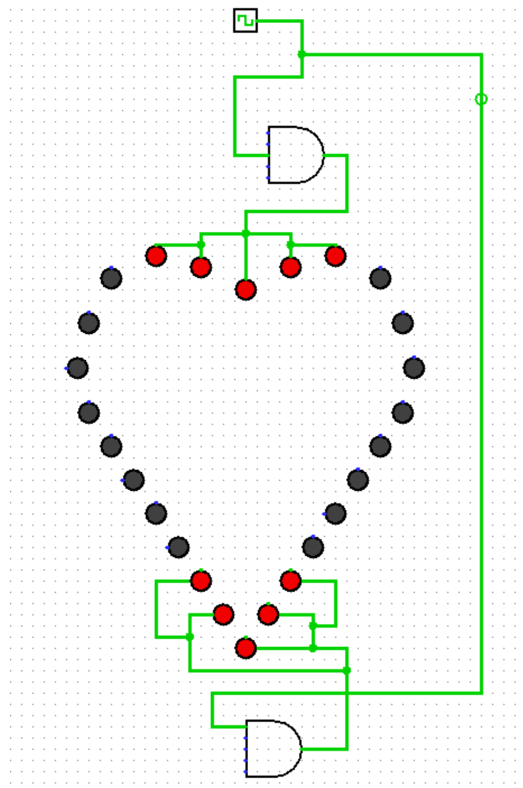

Figure 5: Second version of Heart Design in Logisim

In the third version of the heart design, we organized our 26 LEDs by naming them individually as LED 1, LED 2, LED 3, and so on, up to LED 26. This clear labeling made it easier to determine the specific pattern we wanted for each LED. We also labeled the clock and the inverter in our circuit, which helped us manage the timing and inversion of signals. For this design, we decided which LEDs should be on when the clock is 1 and which LEDs should be on when the clock is 0.

To ensure accuracy, we documented this decision process on paper, where we outlined each LED's state (on or off) based on whether the clock signal is high (1) or low (0). This careful planning allowed us to create a dynamic and well-organized heart pattern that changes based on the clock's state.

Selection: LED	
Facing	North
On Color	#f00000
Off Color	#404040
Active On High?	No
Label	LED1
Label Location	Center
Label Font	SansSerif Plain 12
Label Color	#000000

Figure 6: LED Properties

In this third version of the heart design, we first placed each LED carefully to form a clear heart shape. Once the LEDs were positioned, we organized and labeled them individually (e.g., LED1, LED2, etc.), which made it easier to control the lighting pattern and manage the circuit logic effectively.

Based on our design decisions documented on paper, we determined when each LED should be on or off, depending on whether the clock signal is high (1) or low (0). This planning also guided our configuration of each LED's ”Active on High” setting. For each LED, if it should be active when the clock signal is 1, we set ”Active on High” to Yes; otherwise, if it should activate when the clock signal is 0, we set it to No.

This configuration allows the LEDs to create a dynamic and visually pleasing heart pattern that alternates based on the clock signal's state. We modified the LEDs so that when the input (clock) is 1, the LEDs are blue, and when the input is 0, the LEDs that were blue now display as yellow.

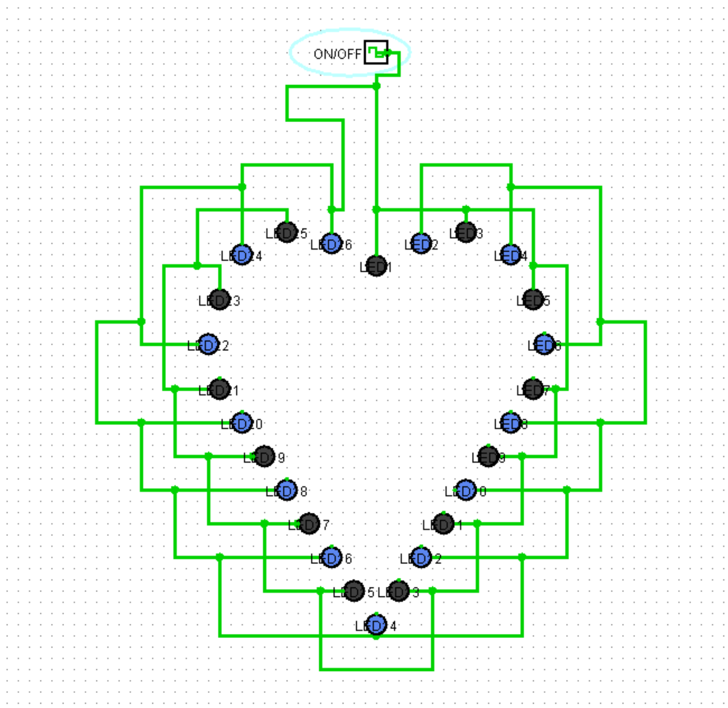

Figure 7: Heart when the clock value is 1

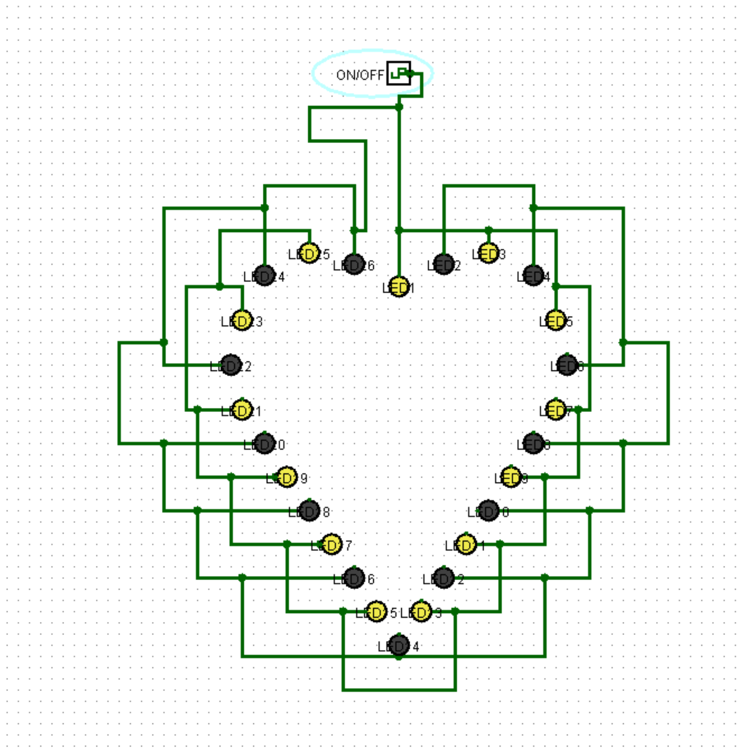

Figure 8: Heart when the clock value is 0

Selection: LED	
Facing	North
On Color	#f0eb4c
Off Color	#404040
Active On High?	No
Label	LED1
Label Location	Center
Label Font	SansSerif Plain 12
Label Color	#000000

Figure 9: Changing the colour of LEDs

This is how we changed the colour of the LED. F0EB4C is a HEX code for yellow colour.

4.4 Robot LEDron

In this design, we created a digital representation of a robot LEDron using LEDs and logic gates in Logisim, combining structural elements with dynamic lighting effects to bring the figure to life. The main outline of the robot is crafted with green lines that define its head, body, arms, and legs, giving the robot a recognizable frame that remains consistently illuminated. This structural outline is always visible, symbolizing the stable framework of the robot.

We decided to make a robot because it's such a fun, creative way to bring a digital circuit to life. Plus, building a robot lets us experiment with logic gates and timing in a way that feels like storytelling. With each LED and each connection, we're adding another layer of character, making it feel more than just a circuit.

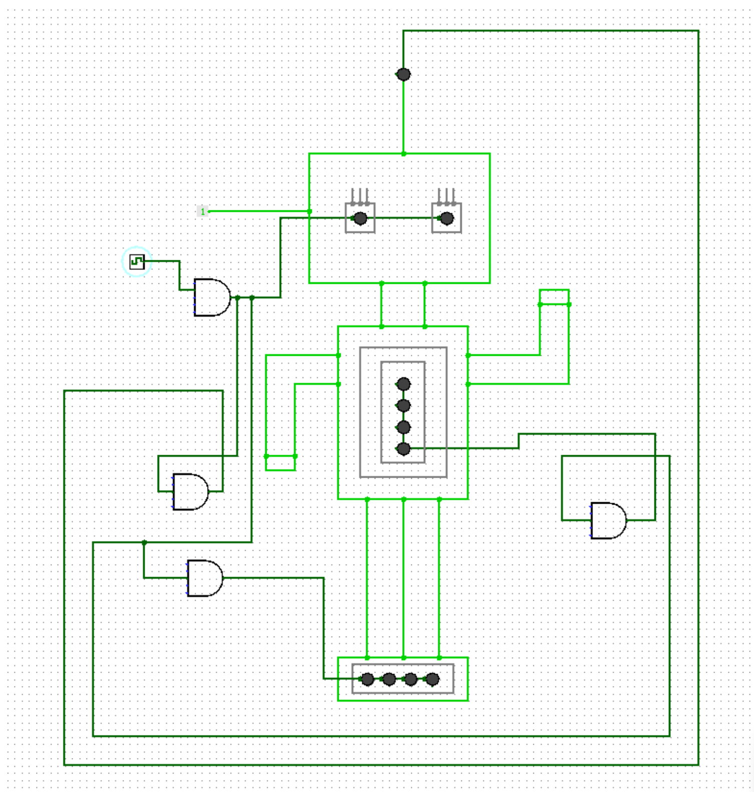

Figure 10: LEDron when the clock is off

To add more detail, we placed LEDs in specific areas to represent key features of the robot. For instance, two LEDs were used for the robot's eyes, connected through AND gates to the clock signal. This setup means the eyes light up only when the clock is on, adding an animated effect to the robot's face. Similarly, a series of LEDs were placed on the torso to mimic clothing or a chest panel. These LEDs are also tied to the clock signal, giving them a blinking effect that activates with each clock pulse, enhancing the visual complexity of the design.

The logic behind the design relies heavily on the use of AND gates.We used AND gates to light up LEDs at the same time but different locations. By connecting the LEDs to AND gates and routing the clock signal through them, we established a set of conditions under which each LED would turn on. This ensures that when the clock is active, the LEDs light up in certain parts of the robot, while the rest of the structure remains consistently green.

The clock acts as a central control, adding life to the design by making specific LEDs blink in response to its pulses. When the clock is off, only the green outline of the robot is illuminated, giving a basic, static representation. But when the clock is turned on, the LEDs representing the eyes and clothing activate, creating the appearance of a lively, animated robot

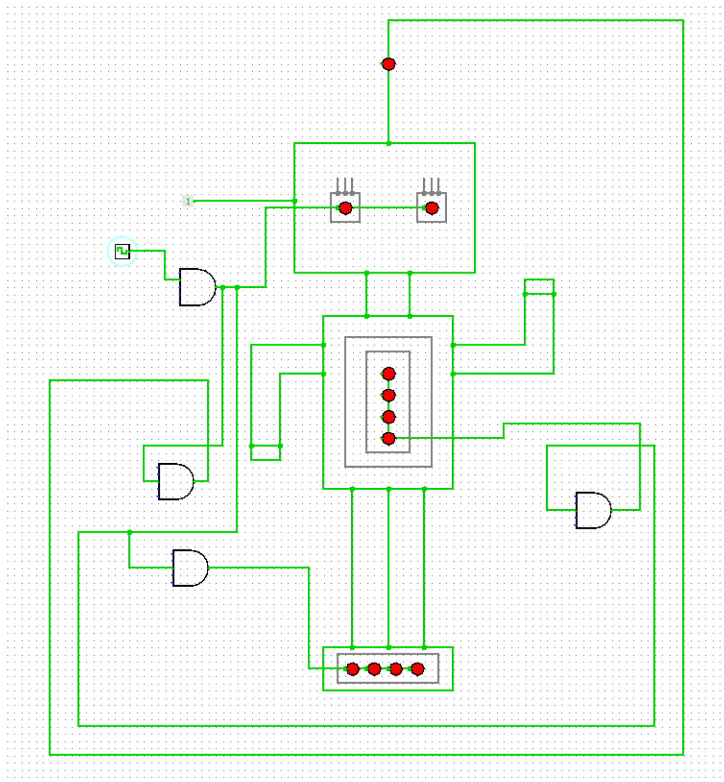

Figure 11: LEDron when the clock is on

4.5 CO 208 Design

For our “CO 208” design, we wanted something simple and clean that makes the course name and number easy to read on an LED grid. “CO” stands for “Computer Organization,” and “208” is the course code. We used AND gates to control which LEDs light up, so we could turn on specific sections at the same time without them blending into each other. This keeps ”CO” and ”208” clearly separated and readable.

The clock signal ties it all together by lighting everything up in sync, making the display look smooth and unified. Overall, we aimed to show off “CO 208” in a way that’s clear, neat, and highlights what digital logic can do when applied creatively.

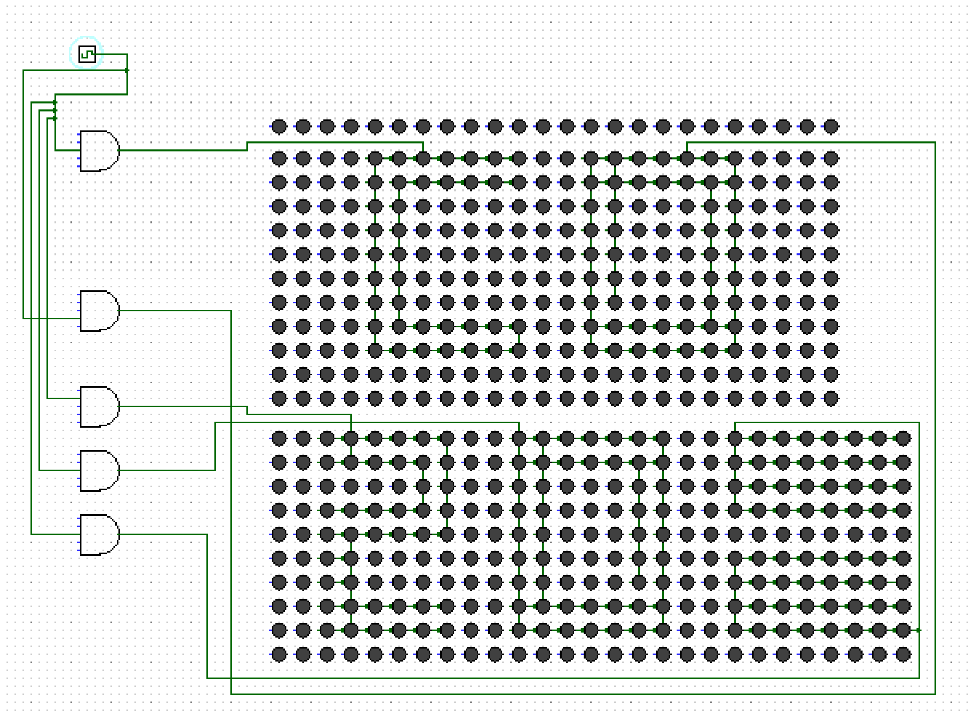

Figure 12: CO Design when clock is off

To create the “CO 208” display, we arranged LEDs on a grid to form each letter and number. We used AND gates to control which LEDs light up, allowing us to highlight specific parts of the display at the same time without overlap. The clock signal synchronizes everything, making the whole design light up together for a smooth, unified look. We chose this setup to keep “CO” and “208” clear and easy to read. The result is a clean, readable display that shows off what digital logic can do with simple components.

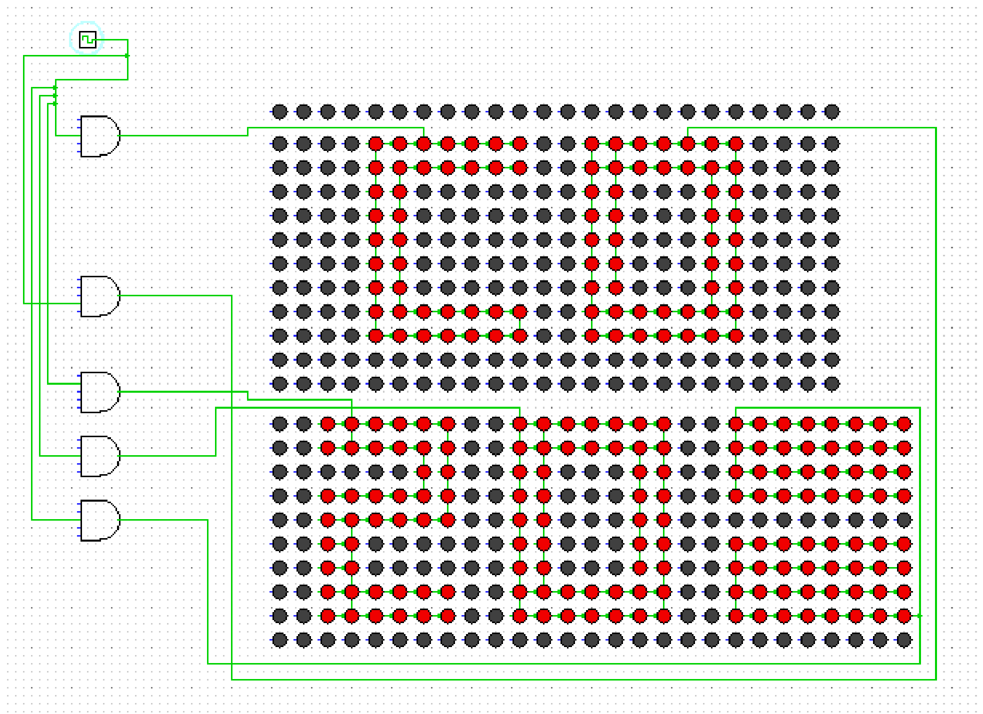

Figure 13: CO Design when clock is on

5 Simple Shape Design

We designed a simple LED-based shape in Logisim to explore how logic circuits can control LED patterns. The circuit includes a clock input that manages the on-off state of the LEDs, creating an alternating pattern. By using basic logic gates, we arranged LEDs in a shape that lights up in specific sequences depending on the clock's state. We created a subcircuit to define the behavior of each LED, with a truth table specifying which LEDs should be on or off for each clock cycle.

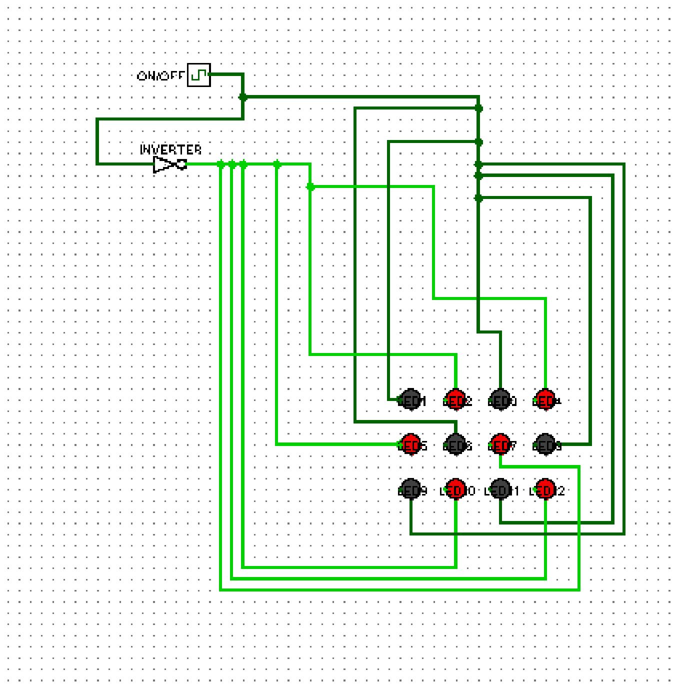

Figure 14: This is how the simple shape looks like when the clock value is 0

This close-up view of the LED circuit shows each LED (from LED1 to LED12) connected to the clock input. Alternating LEDs are linked to NOT gates (inverters), allowing them to turn on and off in sequence as the clock switches between high and low states. When the clock is high, LEDs without inverters turn on, and those with inverters stay off. When the clock is low, the roles reverse, creating a flashing effect.

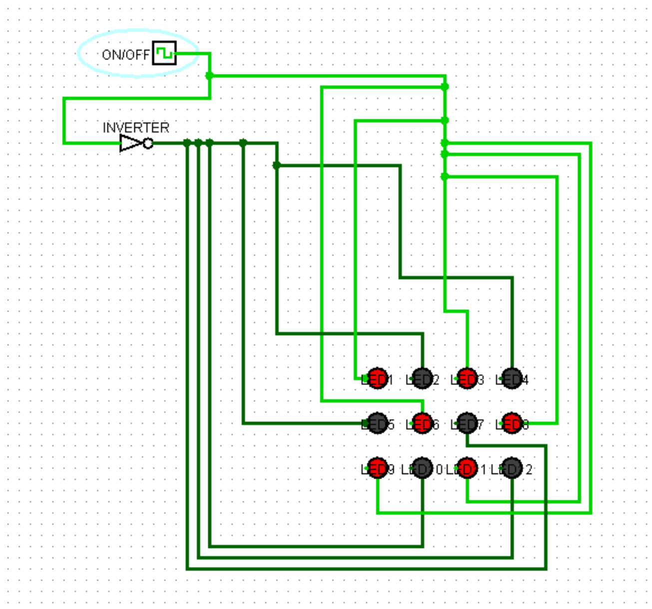

Figure 15: This is how the simple shape looks like when the clock value is 1

To control the LEDs, we created a subcircuit with a truth table that defines the behavior of each LED according to the clock signal. The truth table specifies that when the clock is high (1), one set of LEDs turns on, and when the clock is low (0), the alternate set of LEDs turns on.

Our input is clock and the outputs are LED1, LED2, LED3 so until LED12.

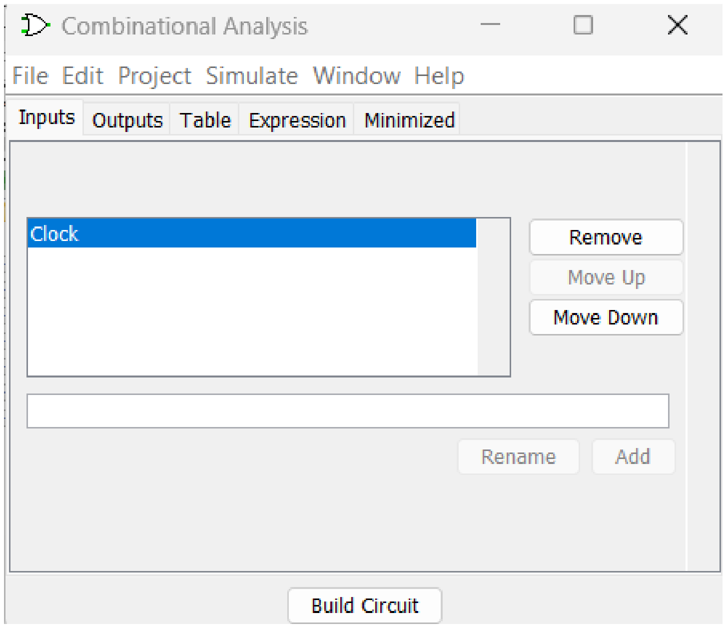

Figure 16: Clock is input

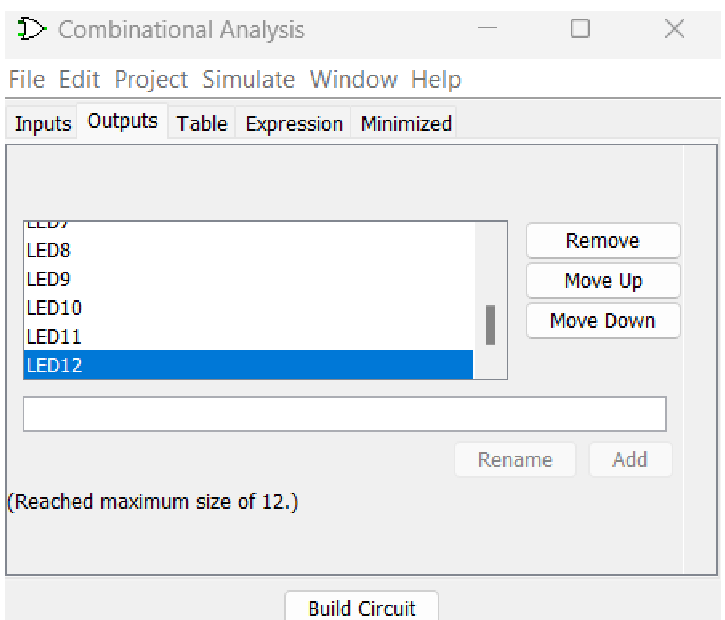

Figure 17: LEDs as the outputs

Clock	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8	LED9	LED10	LED11	LED12
0	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	0	1	0	1	0

Figure 18: This is how our truth table looks like

This truth table represents the behavior of a clock-controlled circuit with 12 LEDs (LED1 to LED12). The ”Clock” column indicates the two possible states of the clock signal: 0 and 1.

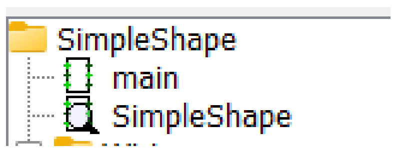

Figure 19: File structure

In the Figure 19. we can see the actual circuit layout in the ”SimpleShape” subcircuit. It has a single clock input connected to each of the 12 LEDs. Each LED has a different configuration depending on the presence of NOT gates in its connection path. LEDs with NOT gates will switch their state opposite to the clock input (on when clock is 0 and off when clock is 1). LEDs without NOT gates will match the clock signal (on when clock is 1 and off when clock is 0).

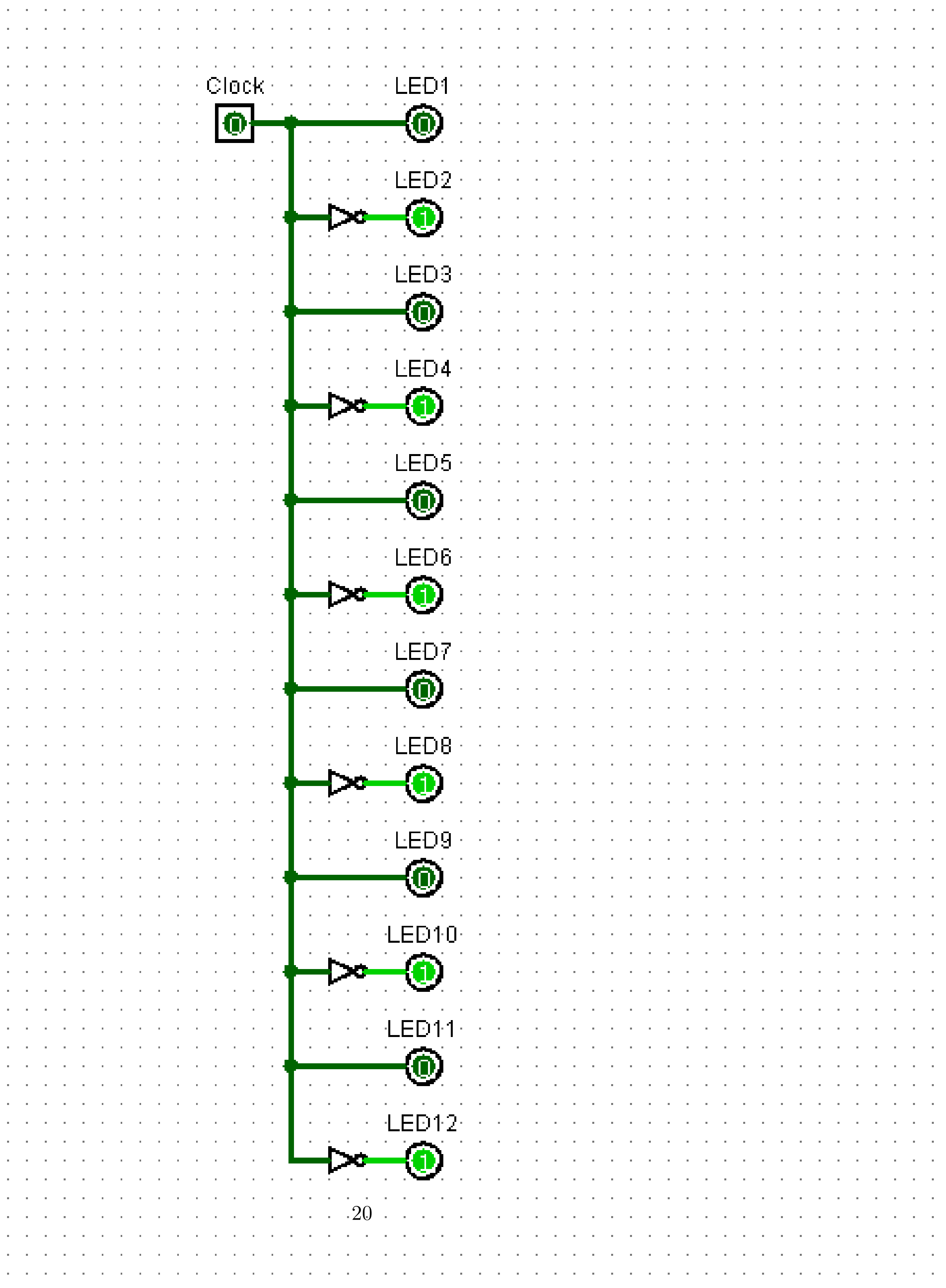

Figure 20: Simple Shape subcircuit

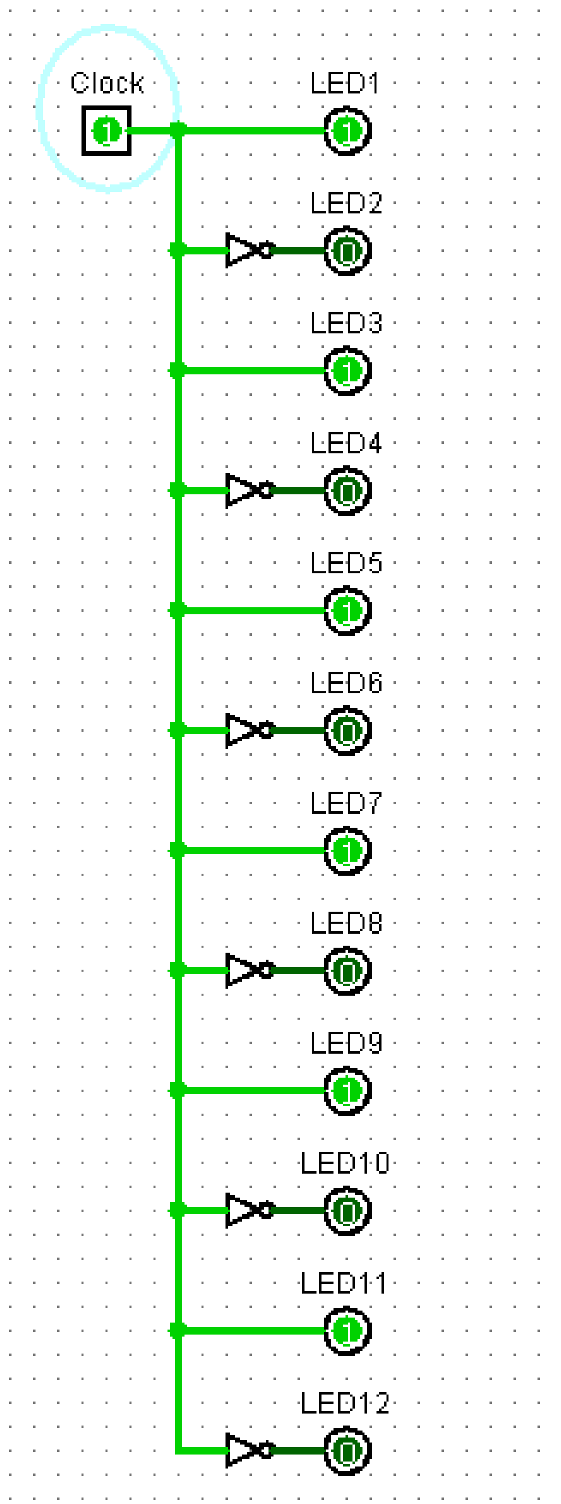

Figure 21: Simple Shape subcircuit

In Figure 20, we can see the actual circuit layout in the "SimpleShape" subcircuit with the clock set to 1. Each of the 12 LEDs responds to the clock input based on the presence of NOT gates in their connection paths. LEDs with NOT gates will display the opposite state of the clock input, meaning they are off when the clock is 1. LEDs without NOT gates will match the clock signal and are therefore on when the clock is 1.

6 Conclusion

This project was a great experience that allowed us to combine the technical aspects of digital logic with creative design. We explored how basic logic gates like AND, NOT, and clock signals can be used to create engaging visual patterns, and the results were more than expected.

The process started with brainstorming ideas and sketching out what we wanted to create. This gave us a chance to visualize the patterns we wanted to design and plan out the steps ahead of time. Once we had a clear concept in mind, we moved on to the actual construction of the circuit. We used logic gates to control the LEDs, ensuring that each part of the design worked as planned and contributed to the overall aesthetic.

One of the key things we learned was how to synchronize the patterns without them overlapping or interfering with each other. This required a lot of trial and error, as we adjusted the timing and gate configurations. But after experimenting with different setups, we were able to create clear, alternating patterns that enhanced the visual appeal of our designs.

By using basic circuits and gates, we were able to create something that was both technical and artistic, blending the two in a way that we had not imagined before. Overall, this project has been an excellent opportunity to apply what we have learned in digital logic while also experimenting with new ways to use it.

7 References

Digital Logic Gates. Available at: https://www.cburch.com/logisim/docs/2.3.0/libs/gates/basic.html

Introduction to Logisim. Available at: https://www.labs.cs.uregina.ca/201/Logisim/lecture.html

Introduction to Logisim. Available at: https://study.com/academy/lesson/introduction-to-logisim-setup-overview.html

Introduction to Logisim. Available at: https://www.scribd.com/document/523130223/Lab-1- Introduction-to-Logisim

The Guide to Being a Logisim User. Available at: https://www.cburch.com/logisim/docs/2.3.0/guide/index.html

Logisim Tips and Tricks. Available at: https://www.cs.cornell.edu/courses/cs3410/2019sp/logisim/tips$_{a}nd_{t}ricks.html$

Logic Circuits and Logisim. Available at: https://www.cs.hmc.edu/ cs5grad/MyCS/2013/Logisim.pdf

Simulation in Logisim. Available at: https://www.cs.williams.edu/ tom/courses/237/labs/Lab3.pdf