Group 12 Project

Madiba Hudson-Quansah & Ronelle Cudjoe

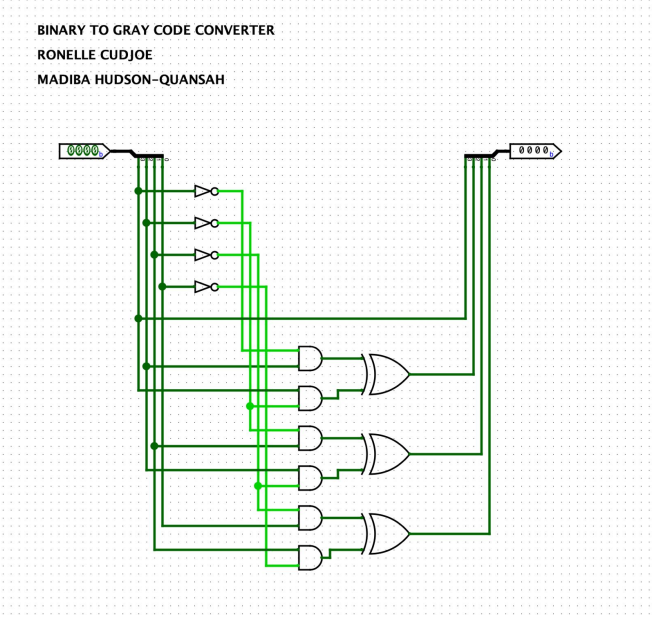
Question 1

Binary to Gray Code Converter
Identify the bit pattern changes and derive an expression with justification.

Solution: The bit pattern changes from binary to gray code ensure that only one-bit changes between consecutive values. The most significant bit of the gray code (G_3G_3) is the same as the binary most significant bit (B_3B_3), minimizing errors. Each subsequent gray bit is derived by XORing adjacent binary bits:

$$G_3 = B_3$$
$$G_2 = B_3 \oplus B_2$$
$$G_1 = B_2 \oplus B_1$$
$$G_0 = B_1 \oplus B_0$$

The XOR operation ensures only one-bit transitions, providing a reliable and efficient method for encoding, particularly in applications where minimal changes reduce errors during transitions.



Question 2

Gray Code to Binary Converter
Identify the bit pattern changes and derive an expression with justification.

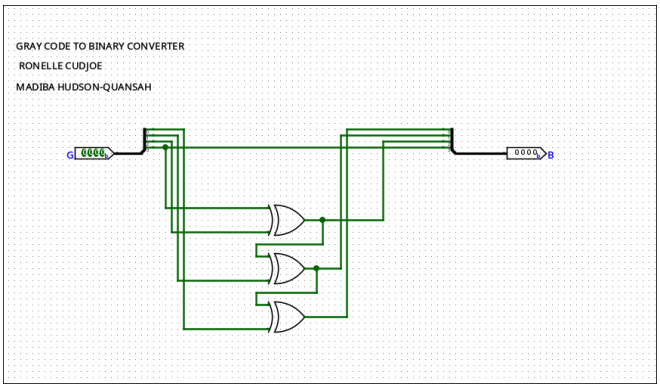
Solution: From our observation of the binary to gray code truth table these things are apparent:

- The most significant bit (MSB) in the binary representation stays the same in the gray code representation.
- Subsequent bits are derived by XORing the current gray code bit with the preceding binary bit, i.e, $B_1 = B_2 \oplus G_1$

This results in the expressions

$$B_3 = G_3$$
$$B_2 = B_3 \oplus G_2$$
$$B_1 = B_2 \oplus G_1$$
$$B_0 = B_1 \oplus G_1$$

Where B and G are the bits of the binary and gray code number representation respectively.



Question 3

BCD to 7-segment Decoder.

Solution:

1.

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

2.

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

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

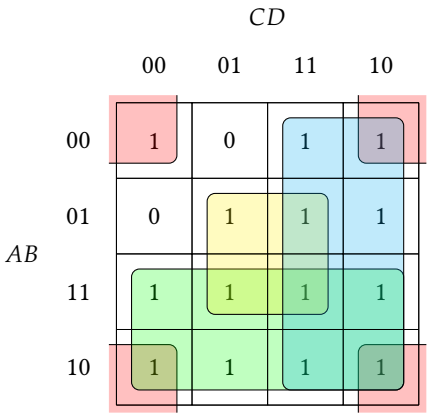
3.

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

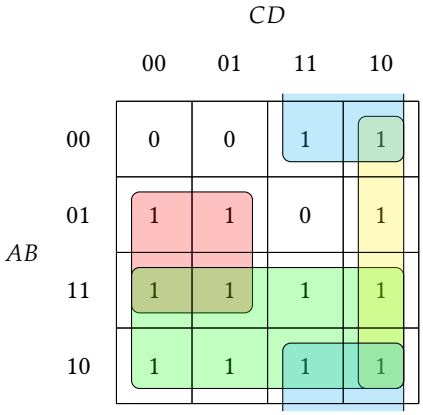
4.

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

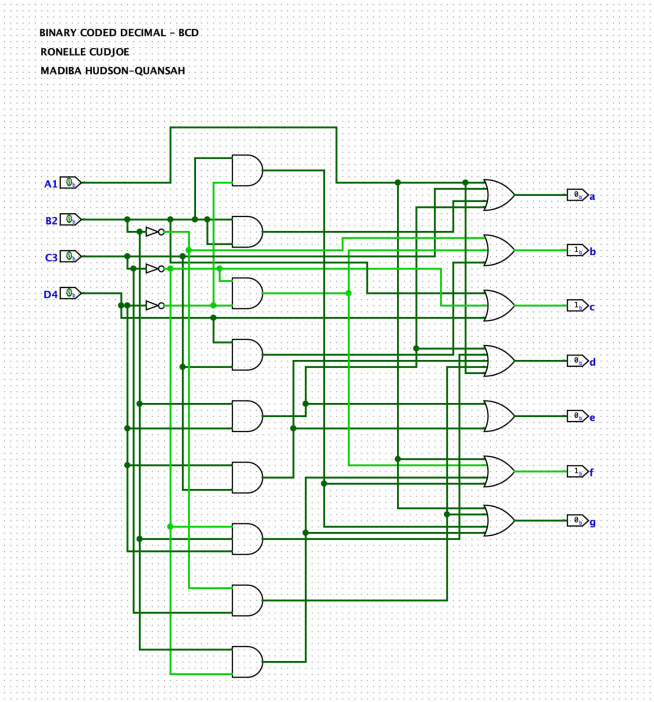
5.

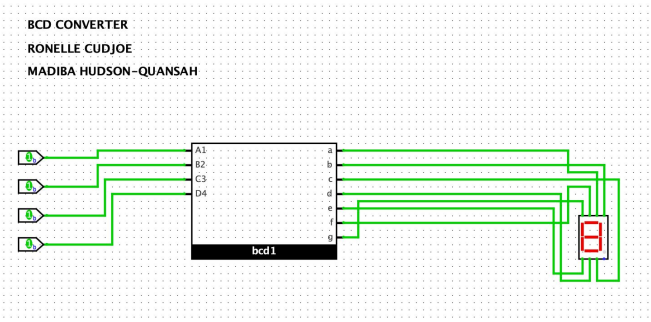

CD
00 01 11 10
AB
00 1 0 1 1
01 0 1 1 1
11 1 1 1 1
10 1 1 1 1

6.


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

7.

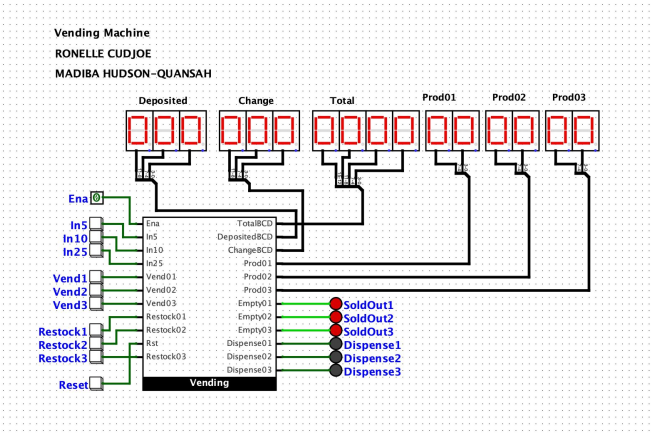

BINARY CODED DECIMAL – BCD
RONELLE CUDJOE
MADIBA HUDSON-QUANSAH
A1
B2
C3
D4
a
b
c
d
e
f
g



Question 4

Ashesi Vending Machine

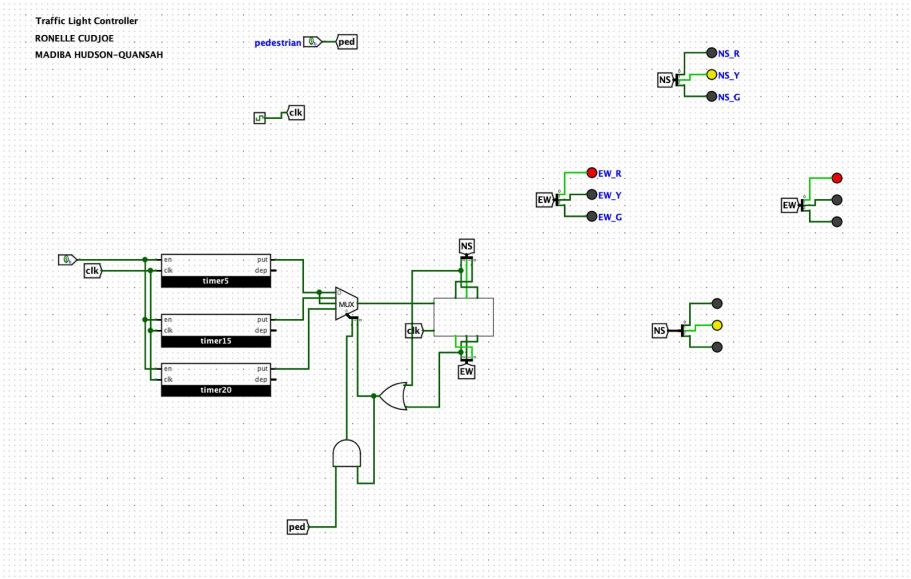
Solution:



Question 5

Traffic Light System

Solution:



Reflection Questions

Question 6

What challenges did you encounter during the design of the Binary to Gray Code Converter, and how did you overcome them?

Solution: The need to understand and implement the specific conversion logic between the two codes, proved challenging and overcoming it involved breaking the problem into manageable steps, using bitwise operations, and designing a modular system that can handle varying bit lengths.

Question 7

What considerations influenced the choice of error-checking mechanisms in the Gray Code to Binary Converter?

Solution: In verifying the Gray Code to Binary Converter, we used logisim's built-in test vector feature to validate the converter's functionality against the provided truth table.

Question 8

How did you define the mapping between BCD inputs and 7-segment display outputs in the BCD to 7-segment Decoder?

Solution: The mapping between BCD inputs and 7-segment display defines how a 4-bit binary input (representing digits 0-9) controls the segments of a 7-segment display. Each BCD input corresponds to a specific 7-bit output, where each bit controls one segment of the display (labeled a-g). For example, a BCD input of 0000 (decimal 0) turns on all segments except 'g', displaying the digit '0' on the 7-segment display.

Question 9

What strategies did you employ to simulate and validate the functionality of the Ashesi Vending Machine?

Solution: We broke down the system into subsystems like input, logic, and output, modelling each part separately and testing various input scenarios, including normal and edge cases to ensure the machine handles all possible user interactions correctly.

Question 10

How did you prioritize optimization efforts in each module to ensure efficient resource utilization?

Solution: We focused on minimizing the number of components, such as logic gates and memory elements, to reduce circuit complexity and resource usage. Simplifying control logic and improving decision-making speed to ensure efficient operation and fast response times.

Question 11

What insights did you gain from comparing different conversion algorithms in the Binary to Gray Code and Gray Code to Binary converters?

Solution:

We found that there were two main ways to convert between binary and gray code. The first was to represent each bit as an output of a boolean expression and find the SOP expression for each bit. The second was based on intuition and inspection of the truth table, which produced a XOR expression for each bit. The XOR expression was more intuitive and easier to derive, making it the preferred method for conversion.

Question 12

How did you address potential edge cases or corner scenarios in the design of each module?

Solution: To address edge cases, we implemented an in-built test vector that detects invalid inputs by checking the system's behavior against the truth table. This test logic simulates a variety of scenarios, ensuring that the system provides appropriate error handling and feedback under both normal and exceptional conditions. By testing edge cases and verifying correct responses, the system can handle unexpected inputs or failures while maintaining reliable operation.

Question 13

In what ways did this project enhance your understanding of digital circuit design principles and the practical applications of Logisim Evolution?

Solution: It has shown me the importance of simulation in digital circuit design, such as modeling a traffic light system, which helped me understand how timing, sequencing, and logic work in real-world applications. This project enhanced my understanding of digital circuit design by demonstrating how simulations allow for testing and refining designs before physical implementation, bridging theory with practical applications using Logisim Evolution.