Project 1

Due August 15, 2025 at 9:00 PM

This project description is subject to change at any time for clarification. For this project you will be working alone.

Desired Outcomes

- Exposure to using circuit simulator (Logisim Evolution)
- Exposure to gray codes
- Exposure to seven segment displays
- An understanding of how to develop simple combinatorial logic circuits

Project Description

You will be using a digital logic simulator called Logisim Evolution Version 3.9.0 available at https://github.com/logisim-evolution/logisim-evolution/releases for this project. For this project you can only use components under the Wiring and Gates sections. You will be given a starter circuit that provides a testing main circuit. Submit a README.md with your circuit file compressed together in a tgz (tar gzip) file. Your README file must have your name, SID number, a brief description of what circuits work/don't work, and a list of sources you used for designing of your circuit (you do not need to list the book or lecture notes it is assumed these have been used). Any allowed use of Generative AI must be properly documented.

1. Use Logisim Evolution to design a circuit to do a three-bit unsigned comparison of two values. Use X2, X1, X0 and Y2, Y1, Y0 as inputs. Outputs should be E and N. E is true if X = Y and N is true if X < Y. Use the following naming convention:

```
Inputs:   X2 X1 X0
          Y2 Y1 Y0
Outputs:     E  N
Circuit: Compare
```

2. Use Logisim Evolution to design a circuit to convert a 4-bit binary value into a 4-bit Gray code. In a Gray code successive values differ in only one bit. Your Gray code will be based upon the last digit of you SID. Use the following naming convention:

```
Inputs:   B3 B2 B1 B0
Output:   G3 G2 G1 G0
Circuit: Gray
```

Binary $B_3B_2B_1B_0$	**0** $G_3G_2G_1G_0$	**1** $G_3G_2G_1G_0$	**2** $G_3G_2G_1G_0$	**3** $G_3G_2G_1G_0$	**4** $G_3G_2G_1G_0$	**5** $G_3G_2G_1G_0$	**6** $G_3G_2G_1G_0$	**7** $G_3G_2G_1G_0$	**8** $G_3G_2G_1G_0$	**9** $G_3G_2G_1G_0$
0000	1111	1101	1100	0100	0101	0111	0110	0010	0011	0001
0001	1110	1111	1101	1100	0100	0101	0111	0110	0010	0011
0010	1010	1110	1111	1101	1100	0100	0101	0111	0110	0010
0011	1011	1010	1110	1111	1101	1100	0100	0101	0111	0110
0100	1001	1011	1010	1110	1111	1101	1100	0100	0101	0111
0101	1000	1001	1011	1010	1110	1111	1101	1100	0100	0101
0110	0000	1000	1001	1011	1010	1110	1111	1101	1100	0100
0111	0001	0000	1000	1001	1011	1010	1110	1111	1101	1100
1000	0011	0001	0000	1000	1001	1011	1010	1110	1111	1101
1001	0010	0011	0001	0000	1000	1001	1011	1010	1110	1111
1010	0110	0010	0011	0001	0000	1000	1001	1011	1010	1110
1011	0111	0110	0010	0011	0001	0000	1000	1001	1011	1010
1100	0101	0111	0110	0010	0011	0001	0000	1000	1001	1011
1101	0100	0101	0111	0110	0010	0011	0001	0000	1000	1001
1110	1100	0100	0101	0111	0110	0010	0011	0001	0000	1000
1111	1101	1100	0100	0101	0111	0110	0010	0011	0001	0000

3. Use Logisim Evolution to design a circuit to convert a five-segment counting rod like encoding to a seven-segment display output. The counting rod numbering looks as follows:

Digit	Image	Vert Lines	Horz Lines
0		0	0
1		1	0
2		2	0
3		3	0
4		4	0
5		0	1
6		1	1
7		2	1
8		3	1
9		4	1

Make sure to get all correct encodings of vertical and horizontal lines to the seven-segment display. For example V W X as true and W X Y as true are both valid encodings for 3, as are V W Y and V X Y.

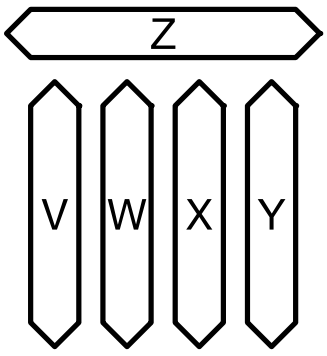

Figure 1. Five Segment Display

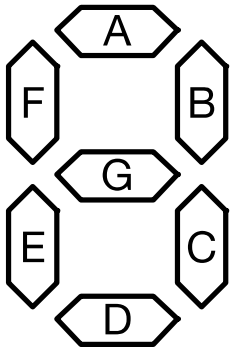

Figure 2. Seven Segment Display

```
Inputs:   V W X Y Z
Output:   A B C D E F G
Circuit: FiveSeven
```

You can unzip the given tgz file with utilities on your local machine, or if you upload the file to the CSIF, you can unzip it with the command:
`tar -xzvf proj1given.tgz`

You **must** submit the circuit file, README.md file, and `.git` directory in a tgz archive. You can tar gzip a directory with the command:
`tar -zcvf archive-name.tgz directory-name`

You should avoid using existing circuits as a primer that are currently available on the Internet. You **MUST** specify in your README.md file any sources of circuits that you have viewed to

help you complete this project. You **MUST** properly document **ALL** uses of Generative AI following the guidelines outlined in the Generative AI Restrictions. All class projects will be submitted to MOSS like software to determine if students have excessively collaborated. Excessive collaboration, or failure to list external code sources will result in the matter being referred to Student Judicial Affairs.

Grading

The point breakdown can be seen in the table below. Make sure your circuit executes correctly in Logisim Evolution 3.8.0 as that is where it is expected to execute. You will make an interactive grading appointment to have your assignment graded. You must have a working webcam for the interactive grading appointment. Project submissions received 24hr prior to the due date/time will received 10% extra credit. The extra credit bonus will drop off at a rate of 0.5% per hour after that, with no additional credit being received for submissions within 4hr of the due date/time.

Points	Description
10	Has git repository with appropriate number of commits
10	Has README.md with proper documentation
15	Comparison circuit correctly outputs E on inputs
15	Comparison circuit correctly outputs N on inputs
20	Gray code circuit correctly encodes values according to SID
20	Five segment encoding correctly outputs seven segment values
10*	Student understands all circuits they have provided

* Students who are unable to demonstrate understanding of their circuit could receive negative points and resulting in score as low as zero overall regardless of functioning of circuit submitted.