

Project 1 Assignment

## Preface

Before starting this project, please be sure that you have completed all of the following activities.

- Read Chapters 2 and 3 of the textbook and view the associated online lectures to gain an understanding of number systems, codes, Boolean logic, and combinational logic circuits.
- Review the course syllabus and be sure that you understand grading and submission policies.
- Review the course schedule to understand the due dates for this and other assignments.
- Review the Graduate Honor System at <https://graduateschool.vt.edu/academics/expectations/graduate-honor-system.html>. Review the Graduate Honor System Constitution, especially Articles I (Sections 1, 2, and 3), V, VI, VII, VIII, and IX.
- This project should be completely your own work. You are welcome to discuss high-level aspects of the project with others and you are encouraged to the Project discussion area on the class website for this. Any questions concerning design details must be directed to the instructor or graduate teaching assistant.
- You will need Logisim to complete this project assignment. Further information about LogiSim is at <http://www.cburch.com/logisim/>. If you have not done so already, download and install Logisim 2.7.1 from <http://sourceforge.net/projects/circuit/>.

## 1. Introduction

The objective of this project is to reinforce your understanding of binary codes, combinational logic design, and logic simulation. You must:

- 1) design a combinational logic circuit that displays the hexadecimal value of a gray code input according to the specifications given below;
- 2) debug and test your design by simulating it using the Logisim simulator; and
- 3) document your work in a short report.

## 2. Gray Codes

Consider a system where a value is changed by being incremented or decremented by one. The value is encoded by  $n$  binary signals. As a specific example, consider a value, represented with 4 bits, being incremented from 3 to 4. In a traditional weighted binary encoding, 3 is represented as 0011 and 4 is represented as 0100. For the change from 3 to 4, three bits must change. Since the time of the transitions in the actual signals will always be different if examined at a sufficiently fine scale, the value will not change instantaneously from 3 to 4. As an example, the transition could occur as follows, where the transitions to value 7 and then value 5 are transient in nature.

0011 (3)  $\rightarrow$  0111 (7)  $\rightarrow$  0101 (5)  $\rightarrow$  0100 (4)

The physical reality of such signal transitions can create problems for applications including mechanical encoders and asynchronous (clock-free) systems. This problem can be overcome using Gray codes, which are non-weighted codes that can be used to represent values. Gray codes have the special property that any two adjacent values differ in just one bit. For example, the standard four-bit Gray code for 3 is 0010 and the code for 4 is 0110. These two codes differ in just one bit, the second bit from the left. So, only a single signal needs to change from 0 to 1 (or 1 to 0 for other values) to represent an adjacent value. You can read more about Gray codes at [http://en.wikipedia.org/wiki/Gray\\_code](http://en.wikipedia.org/wiki/Gray_code).

For this project we consider a special type of Gray code called a Balanced Gray code. In a Balanced Gray code, the number of transitions for each bit position is the same when counting through the values. For

example, a four-bit Balanced Gray code can be used to count from 0 to 15 (hexadecimal F). There are 16 transitions as the count goes from 0 to 1 to 2 and so on to 15 and then back to 0. For a Balanced Gray code, there are four bit transitions for each of the four bit positions during the 16 total transitions. This property is useful in some applications.

Table I below shows the encoding of hexadecimal values 0 through F using a 4-bit Balanced Gray code.

Table I. Hexadecimal Values and Associated 4-bit Balanced Gray Code and Binary Code

Hexadecimal Value	Balanced Gray Code ( $X_3X_2X_1X_0$ )	Binary Code ( $Y_3Y_2Y_1Y_0$ )
0	0 0 0 0	0 0 0 0
1	1 0 0 0	0 0 0 1
2	1 1 0 0	0 0 1 0
3	1 1 0 1	0 0 1 1
4	1 1 1 1	0 1 0 0
5	1 1 1 0	0 1 0 1
6	1 0 1 0	0 1 1 0
7	0 0 1 0	0 1 1 1
8	0 1 1 0	1 0 0 0
9	0 1 0 0	1 0 0 1
A	0 1 0 1	1 0 1 0
B	0 1 1 1	1 0 1 1
C	0 0 1 1	1 1 0 0
D	1 0 1 1	1 1 0 1
E	1 0 0 1	1 1 1 0
F	0 0 0 1	1 1 1 1

### 3. Design Specification

You are to design a combinational logic circuit that accepts a four-bit Balanced Gray code ( $X_3X_2X_1X_0$ ) as its input and creates a four-bit output ( $Y_3Y_2Y_1Y_0$ ) that uses standard binary encoding to represent the same hexadecimal value. In other words, the circuit translates between the Balanced Gray code input and the binary code output as indicated in Table I. Figure 1 provides a block diagram of the function. You do *not* need to minimize the logic function or associated circuit, but you may choose to do so.

Note that Table I is not a true truth table in that it is not ordered by input. You need to rearrange the rows in Table I to construct a standard truth table with inputs  $X_3X_2X_1X_0$  appearing in order from 0000, 0001, 0010, ..., 1111.

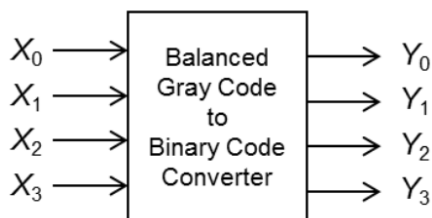


Figure 1. Block diagram of the converter function.

### 4. Modeling the Circuit in Logisim

Use the Pin device in Logisim's Wiring library to control the four inputs ( $X_3X_2X_1X_0$ ) to the combinational circuit. The Pin device is also available on Logisim's toolbar. Each pin can be interactively set to 0 or 1

using Logisim's Poke tool to test the circuit for different Balanced Gray code input values. If the proper connections are in place when Logisim is running, signals with logic level 1 appear in bright green and signals with logic level 0 are shown in dark green.

The circuit's four output bits should be used to control a hexadecimal display to show values 0 through F, inclusive. Use the Hex Digit Display device in Logisim's Input/Output library. It accepts a 4-bit binary encoded value as input and displays the hexadecimal digit corresponding to the binary-encoded input. Use the Splitter device in Logisim's Wiring library to interface the four individual single bits produced by the combinational circuit ( $Y_3Y_2Y_1Y_0$ ) to the four-bit wide input to the Hex Digit Display. The Hex Digit Display device has a second input to control the decimal (hexadecimal) point. The decimal point input can be left unconnected. Figure 2 shows a possible layout for the design. The associated Logisim circuit file is provided with this assignment.

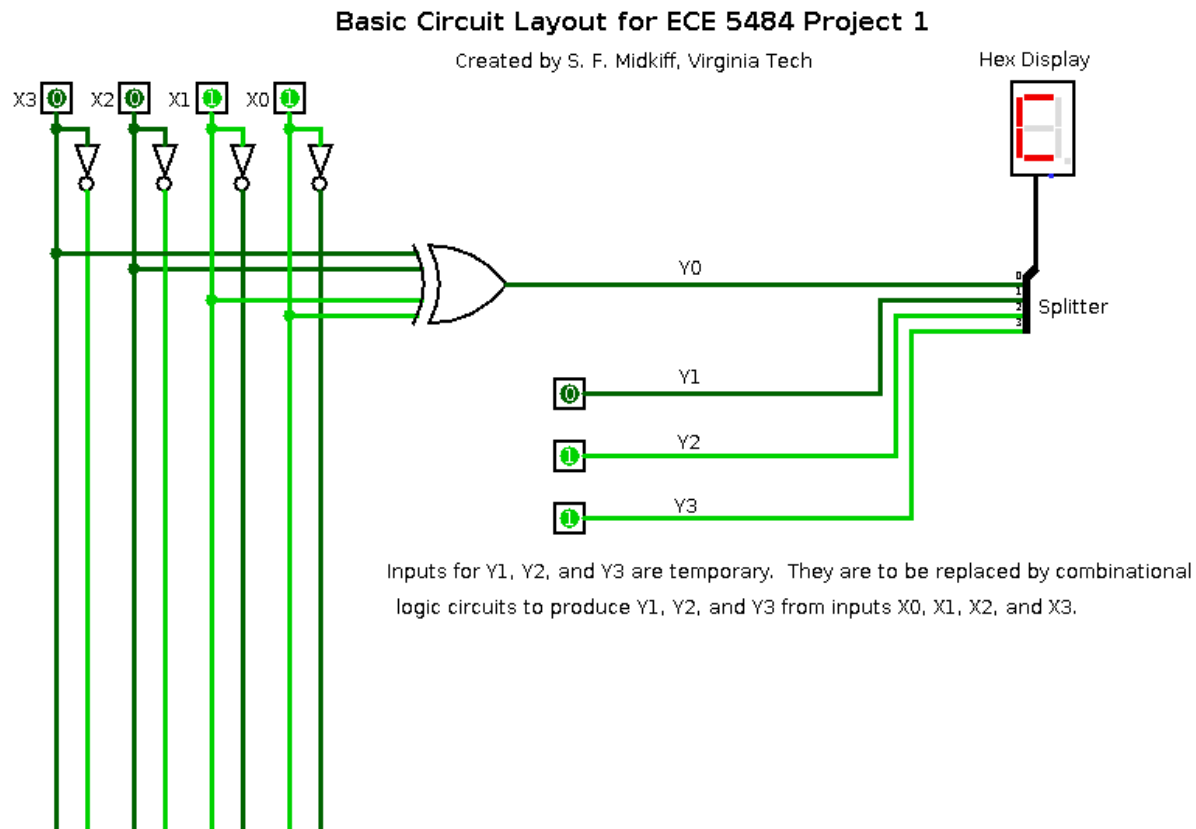


Figure 2. Possible circuit layout including logic to produce output  $Y_0$  (input is for Balanced Gray Code value 0011 which produces output 1100 or hexadecimal C).

The design in Figure 2 includes the combinational logic to produce output  $Y_0$ . By observation, we see that output  $Y_0$  is true if and only if there are an odd number of logic 1 inputs. Thus,  $Y_0$  is implemented by the exclusive-or (XOR) function, i.e.,  $Y_0 = X_3 \oplus X_2 \oplus X_1 \oplus X_0$ . For the Logisim XOR Gate, the Multiple-Input Behavior attribute needs to be set to “When an odd number are on.”

## 5. Simulation

After you create your design, use Logisim to simulate the code conversion circuit. You should test all 16 possible input combinations and verify that the correct values of  $Y_3$ ,  $Y_2$ ,  $Y_1$ , and  $Y_0$  are produced and that the correct hexadecimal value is displayed.

## 6. Submission

### 6.1. Report

You must document the design, simulation, and outcomes in a brief written report. Your report should contain the following items.

- At the top of the first page of your report, include: your name (as recorded by the university); your email address; and the assignment name (e.g., “ECE 5484, Project 1”). Do *not* include your Virginia Tech ID number or your social security number.
- The body of the report must contain the following sections. Use section numbers and headings to organize your report.

*Section 1 – Objectives:* Provide a brief summary of the design objectives and general approach to the design.

*Section 2 – Truth Table:* Provide a truth table with inputs  $X_3$ ,  $X_2$ ,  $X_1$ , and  $X_0$  and outputs  $Y_3$ ,  $Y_2$ ,  $Y_1$ , and  $Y_0$ . Each row should also be labeled with the corresponding hexadecimal value. The inputs must be in standard truth table order, from “0000” down to “1111.” Thus, the truth table will look similar to Table I above, but will be reordered.

*Section 3 – Logic Expressions:* Specify the Boolean logic expressions for  $Y_3$ ,  $Y_2$ ,  $Y_1$ , and  $Y_0$ . Show any work that led to the expressions. (You can just state the expression for  $Y_0$  given above, assuming you implement  $Y_0$  as shown in Figure 2.) The Boolean expressions shown in this section for the report should correspond exactly to what is shown in the circuit diagram of the next section.

*Section 4 – Circuit Design:* Include a schematic diagram of the logic circuit that you created using Logisim. Within Logisim, you can select Export Image from the File menu to produce an image file that can then be incorporated into your report. Uncheck the “Printer View” box when exporting the image from Logisim. Show the circuit with input  $X_3X_2X_1X_0 = 0100$  (hexadecimal value 9) applied. For full credit, the Logisim schematic must be neat and easy to read. The four input pins and the four output signals should be labeled, as shown in Figure 2. The diagram should also be labeled with a title, your name, and the date.

*Section 5 – Conclusions:* Briefly discuss the outcome of your design and any problems or aspects that do not work properly; what you learned by doing this project; and any experiences that were particularly good or bad. Also, specify the approximate number of hours that you devoted to the project. (The number of hours is just for the instructor to assess the suitability of this project assignment.)

### 6.2. Submission

Carefully follow these instructions when submitting your project.

- Create a single PDF file for your report. Name the PDF file *lastname\_firstname\_P1.pdf*, where *lastname* is your last or family name and *firstname* is your first or given name. Submit this file.
- Save your Logisim circuit with file name *lastname\_firstname\_P1.circ*. This file will be used to test your design for grading. Submit this file in addition to the report file.
- Submit your assignment in the Assignments area of the class website. You must submit both files by 11:55 p.m. on the due date.

## 7. Grading

The project will be evaluated based on the following criteria.

- Presentation (15 points)
  - Complete, clear, and well organized report
  - Mechanics (spelling, grammar, etc.)
- Technical Merit (35 points)
  - Discussion of design objectives

- Truth table
  - Logic expressions and their derivation
  - Circuit design
  - Conclusions
- Correct operation (50 points)
  - Correct operation as verified using the circuit simulation file