Advanced Exploration 1: Boolean Algebras

MATH2603: Discrete Mathematics

Overview: In this Advanced Exploration, you'll have a chance to explore the connections between mathematical logic, boolean algebras, and computer circuits/architecture by building an addition circuit for a new type of computer!

Instructions: First, read Chapter 14 in the textbook (Boolean Algebras). This material may seem very familiar; much of it is equivalent to propositional logic that we discuss in Chapter 1. Next, complete the exercises below. Your solutions should be complete, clear, and correct. Your solutions should also be accompanied by explanations written in complete sentences that justify your work.

Instructions for submitting your work, and information on how the EMRN rubric will be applied to evaluate this exploration, are at the end of the assignment.

Background: Almost all modern computers store information in *bits* (short for binary digit); a bit is a digital version of a boolean varaible, so we will use x or y to refer to bits. Just like a boolean variable, a bit can have values of 0 or 1.

Computers have digital logic gates capable of performing the logical operations of `AND`, `OR`, and `NOT`. These gates, and associated notation, are discussed in the text. You can also find the symbols themselves on Wikipedia. By combining these logic gates in an appropriate way, one can make a *digital adder*: a digital circuit that takes as input two bits x and y, and outputs the sum $x + y$ (using binary addition) in the form of two bits s (for sum) and c (for carry). An input/output table for a digital adder is shown below. A digital circuit that performs this operation, along with additional details, can be found in the §14.4 of the text.

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The Problem: Casper is an undergraduate at UArk and has been doing cutting edge research computer architectures based on the ternary number system: these computers use ternary digits which can have three values (0, 1, or 2).

1. Casper wants to make an adder for his ternary number system, but he's a bit stuck. Help him out by **making an input/output table for a ternary adder**. Just like the binary adder above, a ternary adder should have two inputs (X and Y) and two outputs (Carry and Sum). This time, there should be three possible values for each input.

2. Casper's research is woefully underfunded, so they wants to build their ternary adder using binary-encoded signals so that two bits are used for each ternary digit. Let the ternary digit X be encoded in the binary digits x_1x_0 so that $00 = (0)_3$, $01 = (1)_3$, and $10 = (2)_3$. Similarly encode Y as bits y_1y_0 and S as s_1s_0. Since the carry digit only takes values 0 and 1, we can encode C using a single bit (which you may also call C).

Design a binary circuit that implements the ternary adder that Casper needs to continue their research. A good first step would be to make an input/output table using binary (as opposed to the ternary table above).

Submitting your work: Your work must be neatly typed up using a system that supports mathematical notation. For example, you can use MS Word and its equation editor; or you can write your work in a Jupyter notebook using Markdown and LaTeX. Once it is written up, the work must be saved as a PDF file and then uploaded as a PDF to the area on Blackboard where the original assignment is located. Remember that the work is not actually submitted until you upload the file and click the Submit button. Grading and feedback will take place entirely on Blackboard. The following are not allowed: Submissions outside Blackboard (for example through email); files that are not in PDF form; and work that contains any handwriting, though you may *draw a diagram* neatly by hand, scan it, and include it in your submission.

Evaluation: Like all Advanced Explorations, your work will be evaluated using the EMRN rubric. Please see the statement of this rubric in the syllabus for an explanation of how it is used. When applied to this Advanced Exploration, the following criteria help to assign the grade:

- **E:** The solution consists of a clear, correct, and complete solution that includes several paragraphs explaining the binary circuit which implements the ternary adder. The solution contains no major errors (computation, logic, syntax, or semantic); it is also exceptionally clear and the writeup is professional in its look and style. The solution would be at home in a professional lecture or publication.

- **M:** The solution consists of a clear, correct, and complete solution that includes several paragraphs explaining the binary circuit which implements the ternary adder. The solution contains no major errors (computation, logic, syntax, or semantic) and is neatly and professionally written up.

- **R:** The solution contains at least one, but not several, major errors (computation, logic, syntax, and/or semantic) that require revision. An "R" may also be given for writeups that do not expend sufficient effort to produce a good-looking writeup.

- **N:** The solution has several significant errors; or the solution is missing large portions of the solution; or the solution is for a significantly altered version of the problem; or the submission is excessively cluttered, messy, difficult to read, or handwritten.