Jared Wheeler
Josh Langley
November 7, 2016

Lab 4 Report

CSCE 312 - 503

For Problem 1 a, Jared Wheeler wrote this alone. The design is based on a mixture of the twenty back-and-forth sequential LED design described in the manual, and modern car blinkers that are more practical. There are two switches that function like a blinker switch in a vehicle, one that prompts the light to shift left and one that shifts right. This input is then fed into a crafted counter like our TA showed us in lab; it uses a multiplexer and a register that logisim has prebuilt into it. There are two inputs fed into the counter; the input switch itself, and a safeguard in the form of a XOR gate. This is so that the lights cannot be going simultaneously, just like a real vehicle. Two counters are on display in this circuit, one that blinks left and one that blinks right. The one that is leftward is using a subtracter, which functions as the opposite of the right blinker's adder. Before starting the simulation, be sure to enable the adder and subtractor or the circuit will not be activated.

Upon a high clock cycle, given that the inputs are one, the register will up the count (or subtract it, depending on the input) and each high cycle, it will increase by one. This will continue until the input is 15 or 0. In this case the circuit will reset the count and continue the blinker.

In part b, a ROM replaces the decoder. The ROM will convert the appropriate binary digits and convert them to a Hexadecimal number that will translate to a corresponding LED. Refer to the following truth table for more information.

Binary	Hex
0.0.0.1	0.0.0.1
0.0.1.0	0.0.0.2
0.0.1.1	0.0.0.4
0.1.0.0	0.0.0.8
0.1.0.1	0.0.1.0
0.1.1.0	0.0.2.0
0.1.1.1	0.0.4.0
1000	0.0.8.0
1001	0.1.0.0
1010	0.2.0.0
1011	0.4.0.0
1100	0.8.0.0
1101	1000
1110	2000
1111	4000

For problem 3, Jared Wheeler wrote this alone. It is divided into 9 part. Part one is the single bit adder that will help handle the mechanics in the other circuits. Part two is the 8 bit adder. It is designed with an input of 8 bits for an input A and another 8 bits for an input B. These two inputs are then put through eight sequential 1 bit adders that were designed in part one. The last part of this design is the subtractor, which is similar to the 8 bit adder, but invokes two's complement. This is done by adding a NOT gate after input B and then adding an input of one to the input A.

The next designs are for the bit shifters. There are 3 left-right shifters included, and they come in a variety of 2, 4, and 1. This is done using a crafted multiplexer to be able to handle three inputs, because logisim's multiplexer wouldn't do for this design. The multiplexer is wired to a register. The bit shifter is based on the design covered in Homework 3 of the lecture. These three shifter are used in the barrel shifter, which allows the user to pick which shifter to use at will. The bits that are shifted are chosen using an input option of three bits, each one representing 1, 2, and 4. Toggle one of them to see the circuit in action. The input 8 bit option is simply for which bits will actually be shifted left or right. Lastly, there is a one bit switch that will determine whether the bits are shifted left or right. 0 is for left shifts, 1 is for right shifts. These inputs go through a decoder.

The magnitude comparator has two parts, the first is a one bit comparator and the second is an eight bit comparator. Like with the adder, the eight bit version is designed with eight of the one bit comparators in sequence. It takes a simple input of 010 which is mostly unimportant in the design other than initializing the program. Input A and Input B are what is really compared, so when testing, just use those.