//
Seongwoo Choi
scho29@ucsc.edu
CMPE - 012
#1368039
M/W 2:00PM - 4:00PM
//

Title:
Building an ALU

Purpose:
To build the logic necessary to perform any of the LC-3 operation instructions (ADD, AND, and NOT), but only for 4-bit inputs and a 4-bit output.

Procedure:
The methodology of using MultiMedia Logic software to build a basic storage using LC-3, and the D-Latch. First, scheme through the lab 4 document and then work on lab 3 since this is the extended version of the lab 3. Second, create a schematic page and modify the lab 3 and then add few more logic operations. Download the pdf that was provided from the website.

Algorithm and Other Data
In the beginning of this lab, the first step to do was figuring out how many switches and buttons, and latches were needed. In total, there are eight switches and two buttons. Two buttons were used for clock and reset functions. There are in total four registers. Each of them has different outputs and inputs, but those inputs are from previous registers. For instance, if the register 0 has 4 outputs, those outputs become inputs for the next register, which is register 1. Since, in this lab there are two addresses named A.0, and A.1, it was necessary to use a 2:4 multiplexer that can combine those two addresses and outputs them into R.0, R.1, R.2, and R.3. Each of these outputs goes into AND gate of each register on Register0.

For Latch A and B, there is one logic gate that reads three inputs; clock, LA_Read, and R/W. Those three inputs were connected to AND gate and outputs an LA-WE sender (Write Enable). There are four receivers that receive the output of LA-WE.

At the end of the designing, the inputs of registers were replaced by Input Selector Logic outputs, which are Data0, Data1, Data2, and Data3.

What went wrong or what were the challenges?

For this specific assignment, the challenge was minimal because this was the extended version of the laboratory assignment 3. However, this does not mean that this was the easy assignment. Especially, OPCODE assignment was very challenging since it was hard to figure how those logic gates can be used as ALU. For instance, for ADD logic gates, it was necessary to look up how full 4-bit ADDER functions work. The inputs of the ADD function were LA.0, LB.0, and ground. Those three are used and the outputs are ADD.0 and C.0 (Carry out). On the next logic gate, C.0 replaces ground, and the LA.1 and LB.1 were used.

Conclusion:

In comparing to the previous labs, this lab was very challenging because this made me to spend a lot of time to work on it. It was a time consuming procedure to figure out how ALU works and how the interface works eventually. In general, the lab 4 was the most challenging assignment and at the same time, it showed me how to represent ALU and I learned how to design the opcode.

Extra:

Since this lab only allows to use 4-bit input, there will be overflow if the logic gate cannot represent number that goes over by 4-bit inputs. For instance, 1110+1101 cannot be represented. So there is an overflow in the logic gate.