**CECS 341 - Lab 2**

**“ALU and Register File”**

**Due date: 02/05/19**

Student Name: Chaz Del Prato

Student ID: 017844266

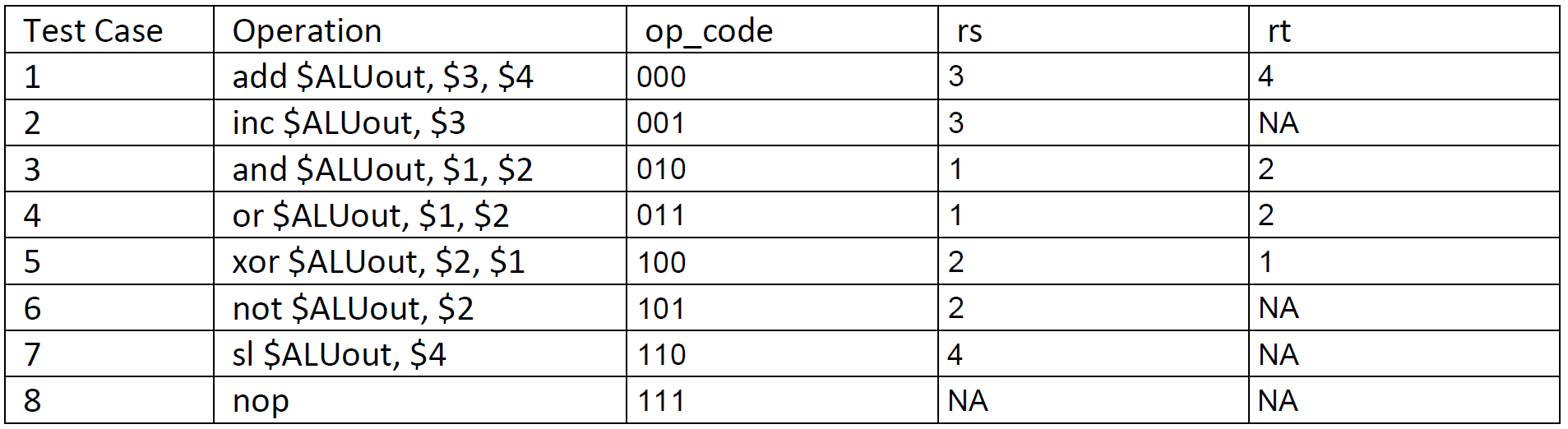
I certify that this submission is my original work

Chaz Del Prato

Lab Report: Lab Assignment 2 - “ALU and Register File”

1. **Goal:** The goal of the lab assignment is to model part of the CPU data path so that it will be able to execute ALU operations on data contained in the register file.
2. **Steps:**
   1. Step 1: Read over the lab instructions.
   2. Step 2: Copy the code for the register and ALUOut files. Also make a copy of the ALU file from the previous lab.
   3. Step 3: Copy the skeleton code for the test bench and the design files.
   4. Step 4: Fill out Table 1 with the correct values.
   5. Step 5: Complete the skeleton code for the design file by determining the relative inputs and outputs for each module.
   6. Step 6: Compete the skeleton code for the test bench file by references the completed Table 1.
   7. Step 7: Check each case to make sure the program is working correctly.
3. **Results:** The result of the program starts with outputting each register and its contents. The first check we can do is to make sure the first 5 registers are initialized to the given values in the test bench and then to see if each register has some contents in the them. For the test cases, it will display rs and rt (if applicable) in the srca and srcb wires. Then it will display the operation that the ALU is going to preform in assembly code. It will next output the output of the operation. Finally, the program it self will preform a self test to make sure the case passed. I would check each test case by hand writing and checking to see if the operation is correct. Using number 1 as an example, I would write the inputs that are the decimal 500 and 1000, the ALU command is to add the two numbers, I would add them and then check the result.

**Table 1**



1. **Conclusion:** I learned how to create a part of the CPU data path by using a register file to receive inputs for the ALU and to have that ALU perform operations. The most challenging part of the experiment was to fill in the test bench skeleton code. I needed to read almost every line to understand exactly what the program was expecting to be filled in.