**CECS 341 - Lab 6**

**“MIPS Instruction Fetch Stage”**

**Due date: 04/09/2019**

Student Name: Chaz Del Prato

Student ID: 017844266

I certify that this submission is my original work

Chaz Del Prato

Lab Report: Lab Assignment 6 - “MIPS Instruction Fetch Stage”

1. **Goal:** The goal of this lab is to understand one of the 5 cycles of the MIPS single processor. We will create the Instruction Fetch stage of the processor that fetches machine code from the IMEM and the Program Counter value to increment and the updating the Program Counter to the next logical step.
2. **Steps:**
   1. Step 1: Read over the entire lab and understand the procedure
   2. Step 2: Copy the code for the register module (flopr.v)
   3. Step 3: Copy the code for the 32-bit adder
   4. Step 4: Copy the code for the Instruction Memory
   5. Step 5: Copy the code for the MUX that was previously used on a different lab
   6. Step 6: Copy the code for the test bench
   7. Step 7: Copy the skeleton code for the IF Stage module
   8. Step 8: Understand how the ALU and control signals work to make the Instruction Fetch Stage occur.
   9. Step 9: Complete the skeleton code for the IF Stage module by filling in the correct inputs and outputs for each module inside.
   10. Step 10: Check the results with the results given in the lab instructions
3. **Results:** In each test case the number of the test case, PCSRC, PCBranch, PC, Instr, PCNext, and the reset flag will be outputted. For example, Test Case number 0 will have PCSRC at zero because that will just increment the Program Counter as usual. The contents of the PCBranch are randomly generated. PC is 4’h because each instruction is 4 bytes long and has been incremented from 0 because the PCSRC is 0. The contents of Instr is the 32-bit machine code from the Instruction Memory at the address of the PC. Then, PCNext has the address of the next logical spot in memory, such as adding another 4 bytes to the PC. Finally the reset flag is set to zero because nothing is getting reset.
4. **Conclusion:** I learned how the Instruction Fetch Stage of the MIPS single processor works with the rest of the processor. The challenges I had from this lab was understanding how the ALU and signals worked to make the Instruction Fetch Stage work correctly and understanding which were the correct inputs and outputs for each module.