**Assignment 2 Project Report**

**Design Options & Approaches**

The assembler was implemented in Python and can assemble. One interesting difference between our architechture and the standard LC-2200 architecture is the use of opcode to select an ALU function. We can forgo ALUFunc because almost every instruction uses the ALU for one type of operation. For the exceptions (Branches, which need to check equality and add), there is a special altFunc signal which indicates to the ALU to use the secondary function instead of the first.

**Challenges**

When creating the processor, one of the most significant challenges was debugging. Compilation was very time consuming, so we wanted to make sure we got the most information we could between changes. To do this, I implemented a timer similar to that in assignment 1 in order to slow down the clock pulse to 1 tick per second. On each tick, we made the HEX display show the current value of the bus and made the LEDR show the current microstate we were on. We then assembled custom ASM files to debug certain instructions. This helped tremendously when tracking down where things went wrong.

**Contribution: 50% (Created processor in Verilog)**