Names: Jack Butler, Holland McQuaid, Christian Swords

CSCE 312

Lab 6 Project

This project was to design and implement a processor that could execute any program written in the instructure set architecture that we designed. In this project we implemented the 8 different instructions. These included irmov, which would move immediate values to a register, rrmov, which would move the value of one register to another, st, which would write the content of a register into an address, ld which would read the content of RAM at an address and load that into a register, OPL which would perform an operation such as add, subtract, multiply, and, or. We also implemented an unconditional jump jmp, a conditional jump jXX, and halt instruction which would end the program. We would then take that instructure set architecture and implement it into a Logisim .circ file. We broke up the CPU into different stages. We included 5 stages. The stages included Fetch, Decode, Execute, Memory, and Write back/PC update. The Fetch stage’s job was to fetch the instructions that were given to it from the ROM and allow those instructions to be accessed by the rest of the CPU. The decode stage was to take in the icode and the register values and would write or read to those registers. The execute stage was to take in icode and ifun and depending on those values would compute the valE to be used by the rest of the CPU. The memory stage was to take in the icode value and read or write to the memory unit (RAM). The write back stage was to write data when necessary to the correct location in the register. Then the PC update stage would just tell the fetch unit where to look for the next instruction. The processor and instructure set architecture was set to read 6 bytes for each instruction to make the timing synchronization simpler.

The project was an overall group effort and we all helped each other on each part whenever needed.