CSCE 312 Final Project – ISA Design

Individual Report

Section 506

Julian Immanuel Tiu*

Akash Kundu

Cole Boggus

Robert Paraliticci

The parts I did alone included activity 1 and the instruction decoder. For activity 1, the task was to design the instruction set architecture. For the design, I decided that since there are only 4 operations, namely read, load, store, and add, there only needs to be 2 bits to represent the op-code. I also decided that there are going to be 6 registers, therefore to represent 6 registers there needs to be 3 bits to represent each register codes. For the actual CPU design, I was tasked to design the instruction decoder. The components of the instruction decoder include three decoders – one for the op codes, another for source register codes and one for the second register.

Since this was a team based assignment, the advantage is that the CPU design could be split in different components that each member can work on, and each components can be put together at the end and work together as one single unit. A disadvantage would be that there could be miscommunications regarding input lines that may cause the registers to hold wrong values. The design of the architecture required that the different members were to all make their design fit with the register file. The design of the components all worked around one component, for which if one component was off then the whole design could fail.

The most superior design was the design of the register file. The register file

contained 6 registers for which the results from the ALU addition were written back regardless of the registers specified by the instruction code. Registers 1, 2, and 5 were all designed as counters that hold memory addresses for the RAM to either load values to other registers or store values from a register. For the registers that hold memory addresses, each time they are being accessed the values in those registers are incremented by 1. The way the design could be improved is by not setting the register values to a set value, because the location in the RAM could be a different size for a different case. Each increment of either memory-holding registers may not correspond to the correct value in memory. If the sizes of the specified matrices were not 2x2, then the current design of the register files may fail.