- Ben Gothard CSSE232 Project Journal

-- Entry #1: 10.1.2018

After talking with my group about which processor design would be best to implement, we unanimously decided to go with a hybrid design that incorporated Matthews idea of explicitly stating in the assembly code which processor to store information into and my idea of implicitly deciding a "default" accumulator register in which to store the value. We then discussed whether we wanted different opcodes for commands that did operations on the 2 accumulators versus using a "funct" flag. I felt that using a funct code was a better idea since it prevented duplication of similar operations and kept the instruction set from getting large. I wanted to use the funct code since we also came up with the idea of having the funct code be the first bit in the machine code so we could easily tell whether it was doing an accumulator operation or an immediate operation. I then worked on a rough assembler to convert our instruction set assembly line-by-line (entered manually by us) to convert the instruction to machine code.

-- Entry #3: 10.3.2018

We met to discuss who is doing what for the next milestone. We broke up all of our instructions into 4 even 6 instruction sections so each of us could do some of the RTL. By some luck, each of us seemed to get a different "group" of instructions, like how I got all of the jump instructions. Since almost all jumps behave like jimm when the flag bit is set, they were all pretty similar in layout.

-- Entry #3: 10.5.2018

We met for about 45 minutes and talked over our RTL and started working on grouping our RTL steps into parallel sections and figuring out which instructions have similar blocks to combine.

-- Entry #4: 10.6.2018

We planned out a state diagram for each "group" of instructions with each person who wrote the RTL doing most of the planning with the rest giving input and insight. I had the jump RTL and diagram. I started working on making Euclid's algorithm into a table and started working on an assembler/compiler for our instruction set to make the process go faster.

-- Entry #5: 10.9.2018

We met the day of Milestone 2's due date to make sure everything was together. While converting our Euclid's algorithm into machine code, I noticed a few bugs in our assembly code, but since I had written a compiler, I checked with the group on bug fixes to the assembly and then re-ran the program to output the correct machine code. I added and made our assembly fragments a little more organized as well.