Christian Meinzen (meinzecp)
CM – 401

Friday 10/4/19

- Discussed for one hour at 4:00 – 5:00 PM
- Decided to make processor with Stack architecture
- Discussed possible calling conventions and addressing modes
- Created three ISA for pushing/popping, jumps/branches, and arithmetic/stack manipulation.

Sunday 10/6/19

- Met in O259 from 5:30 – 8:45 PM
- Most of the time was used to discuss the Stack architecture and make sure all team members are on the same page
- Will and I worked on framing some calling conventions and designed the assembly code for relPrime and GCD
- Used the idea that memory would store return addresses and argument values when making a procedure call
- Tori designed instruction set that we could use to get around our ISA and the 16-bit limitations.

Monday 10/7/19

- Met with Sid from 1:20 PM – 2:10 PM to discuss advice and clarification on addressing modes. Realize that using the stack for return addresses will be better than using memory.
- Met with group from 3:30 – 6:45 PM to redesign assembly code and instruction set.
- Removed ‘jal’ and added ‘dup’ and ‘flip’ instructions for better stack manipulation
- Rewrote code for relPrime and GCD in our new instruction set using polished calling conventions where the return address and arguments are placed within the stack, and the return value is placed in memory.
- Side notes:
    - It feels as though without pseudocode, our code will be complicated to efficiently program
    - While everyone has contributed in some way to the project, I feel as though I have created most of the assembly code. So I am worried that my teammates might not understand how we are implementing our design through code.

Teach them.