Meeting 1:
September 28, 2018
I worked with Joy and Matthew to start creating the basis of our instruction-set architecture and began coming up with lists of instructions along with various conventions and necessary functionality. Decided to base it on accumulator, but quickly realized the need to add some aspects of stack.

Meeting 2:
September 29, 2018
Continued working with Joy and Matthew to try and work out problems with our existing design choices, including problems caused by backing up values to the stack. Solved by creating new commands along with small register file.

Meeting 3:
September 30, 2018
Worked with Ben, Joy, and Matthew to flesh out more details of the first milestone. Debated with Matthew and Ben how to best fix our existing problems with our architecture, which concluded with deciding on a two-accumulator architecture. Could not decide whether or not to make it an implicitly or explicitly used system of accumulators; implicit had the cool factor, while explicit was simpler but the syntax was not consistent with the rest of our code and seemed similar to Load-Store.

Meeting 4:
October 1, 2018
Worked with Ben, Joy, and Matthew to make more decisions and revisions based on our feedback from Sid. Decided on a compromise between Sid, Matthew, and Ben's idea with a swap accumulator, which works as a backup for the main accumulator value and the main is implicitly called. Combined our two instruction types into one for simplicity when assembling, and Ben wrote a basic Python script to interpret the assembly into machine code. I came up with a flag bit to choose whether a command will be done to the accumulators or an accumulator and a stack, which fixed a previous issue with our current approach, which would not work for "and"ing or "or"ing things.