

Task Breakdown and Schedule

Week 1

The majority of week one was spent thinking about the project, and becoming more comfortable with assembly and building subroutines. We wanted to make sure that we had a firm grasp on the language before starting to reduce the overall workload.

During this time we developed a general schedule (which we veered off greatly), as well as a GitHub repo to share code to.

Week 2

During week two we got together and analyzed the manual. We built a document that outlined all of the masks, values, etc for each opcode and any special considerations that we would have to make. We also tried to find opcodes that were similar in nature so that we could divide the work into blocks.

Week two was finished with writing a basic masking function, and some quick testing of NOP and DATA.

Week 3

Week three was where the bulk of the work happened. We spent the first half of the week pair programming a library of subroutines. Each day we met and decided on subroutines that would be useful for us moving forward.

Halfway through the week, we discovered that we had abstracted the problem out enough that we could define an operation with through a structured language. This discovery led us to develop what we are calling **Structured Disassembler Language**. Once we had laid out the basic structure of **SDL** we went on to develop the "master" subroutine, *sub_process_opcode*. This parsed our definitions and output the strings to memory.

The last part of the week was spent writing up the definitions for the opcodes. Within a day we went from having done 1 opcode to 88. We managed to finish up some IO work leaving only address start and end input for week 4. The vast majority of the documentation was finished this week as well.

Week 4

Week four didn't leave much left to be done. We wrote a method that could take input and convert it to hex, and then finished out the remaining commenting and documentation tasks that were left. The

final stage was doing some testing to verify the correctness of the opcode output.

Division of Labor

The entire project from programming to documentation was done together as a pair. The work was split exactly 50/50 by this method.