

Week 1

Hours worked: 6

-Worked on finalizing the type of architecture being implemented by weighing the pros and cons of each. The following is an excerpt from our design document that highlights our reasoning for choosing a stack implementation “ Our motivation for using the register-stack architecture is to create a stack that strikes an even balance between efficient subroutine calls and local variable storage. The stack will also enable operands and intermediate results to be more dynamic in nature. This architecture will enable us to retain simplicity while not compromising on overall efficiency.”.

-Was in charge of the design overview for stack based architecture, part of which is cited above.

- Contributed to defining the parameters needed to determine performance time. We went with short instruction syntax as we feel that this one area where a stack based architecture would shine.

-Weighed in on the discussion of using two stacks over any special registers. For the two stacks, one just holds the value of the return value and parameters whereas the other keeps track of other elements of the program. We made this decision to ensure that the entire process of pushing and popping would be more intuitive wherein order of operations would not be an issue.

-Was responsible for translating rel prime assembly code to machine code to each of the instructions.

-Was in charge of formulating the basic structure of the memory map. We used the bottom most segment for the globals so that we could translate from assembly code to machine code with ease.

- Helped decide the addressing modes being used for the instructions.

Week 2

Hours worked: 5

-Was in charge of formulating the relevant naming conventions

-Was in charge of formulating the component table required for milestone 2.

- Took part in discussion deciding the specifics of the data stack
- Took part in discussion that planned our workload for M3.