

Team 3v: Design Process Journal

**Members: Maura Coriale, Ben Gothard,
Matthew Lyons, Joy Stockwell**

Milestone 1:

Meeting 1: 1.5 hours

September 28, 2018

Members Present: Matthew, Joy, Maura

Overview: We began the process of creating our instruction set architecture by deciding on creating a mostly accumulator-based CPU with some facets of stack. We then began brainstorming our commands and instruction types, as well as deciding conventions for procedure calls, instruction layouts, etc.

We also began writing our first draft of the Euclid's method program using our instructions and current syntax so far, and Joy offered to write a complete second draft before our next meeting.

Major Decisions: Accumulator type (with elements of stack), instruction types, commands, and conventions.

Meeting 2: 1.5 hours

September 29, 2018

Members Present: Matthew, Joy, Maura

Overview: We picked up where we left off last night, starting with going over Joy's second draft of the Euclid's method in our own assembly. We then changed some of our calling conventions that we decided on yesterday, and discussed how comparisons and jump commands would work with our processor. We also worked to try and fix some problems we came across with our commands.

Major Decisions: "Frankenstein type" of accumulator with Stack, calling conventions with procedures,

Meeting 3: 1.75 hours

September 30, 2018

Members Present: Maura, Joy, Matthew, and Ben

Overview: We met again after we all worked on our own portion of Milestone 1. While working on code fragments, Matthew found a

problem with how we are using stack. Ben proposed a new way to approach the problem, including a possibility of 2 accumulator registers to handle problems with operations during function calls. Maura proposed a flag bit for modifying existing instructions. Currently, we are tackling the question on how much the compiler and assembler should do for us and how much should be implicitly in the language. Ex: OP Code for add and lw are the same, only func code is different, should we do something similar with our flag bit?

The Accumulator Problem: Since we probably want 2 accumulators minimum, how do we use them, explicitly by calling them by name in the code or implicitly building it into the hardware with multiplexers and having a flag of sorts that comes from the accumulator that tells the CPU whether a particular accumulator is empty and then makes the decision about where to put the value from there. We whiteboarded side-by-side several different problems (a simple addition problem and a summation in a for loop) to try and see some pros and cons of each solution. We continued adding to our list of questions to ask Sid on Monday during our meeting.

Major decisions: Needing multiple accumulator registers, need to change how we interact with data, Implicitly or explicitly moving values between accumulators

Meeting 4: 2.25 hours

October 1, 2018

Members Present: Maura, Ben, Matthew, Joy

Overview: Chose a two-accumulator with a backup accumulator as a compromise between Ben's and Matthew's ideas for the accumulator design. Flag bit for choosing whether to do functions between the accumulators or the stack. All values are signed. Continued making Frankenstein-based name puns (Instruction set = Clerval, Main accumulator's nickname is Mary, secondary's nickname is Shelley). Decided to reduce to one instruction set type to simplify the assembly process, and Ben started writing a basic assembler in Python. Worked through problems as we came across them in writing the assembly fragments, assembling them, and working out syntax and semantics.

Major Decisions: Calling conventions cont, main and backup accumulator, flag bit

Total for Milestone 1: 7 hours

Milestone 2:

Meeting 5: 1 hr

October 3, 2018

Members Present: Maura, Ben, Joy

Overview: Decided on meeting times for the next milestone, split up the RTL commands to write, and discussed some possibilities for I/O and opinions on single/multi-cycle/pipeline.

Major Decisions: NO PIPELINE

Meeting 6: 0.75 hr

October 5, 2018

Members Present: Maura, Joy, Matthew, Ben

Overview: Reviewed our individual RTL's, decided to meet again Saturday morning and evening to try and finish up Milestone 2.

Major Decisions: RTL conventions

Meeting 7:

October 6, 2018: 1.5 hr

Members Present: Maura, Joy, Matthew, Ben

Overview: Met in the morning to start working on State Diagrams, figure out our shopping list of computer components, etc. before our meeting in the evening, due to time conflicts on other days.

Major Decisions: Some conventions for RTL, wrote out State Diagrams

Meeting 8:

October 6, 2018 (pt. 2) 1.5 hr

Members Present: Maura, Joy, Matthew, Ben

Overview: Joy and I worked more on State Diagrams while Matthew and Ben began drawing a circuit diagram to start on the shopping list of parts needed and their states and stuff.

Notes: Want to left shifter hooked up to B, in both the stack and jump to function, going to be left-shifting immediate and storing in b

Major Decisions: Decided to change our flagbit to be part of the mnemonic, so for aadd with accumulators, for example, it would be aadd@, while an aadd with

Meeting 9: 1 hr

October 7, 2018

Members Present: Maura, Ben, Joy, Matthew

Overview: Worked individually mostly, but hashed out what was left to get done on Milestone 2.

Major Decisions: Interrupt-based IO

Meeting 10: 1 hr

October 9, 2018

Members Present: Ben, Joy, Matthew

Overview: Created tests for the RTL, fixed assembler to support flagbit decoration and added label support

Major Decisions: Decided to test the RTL by tracing through it and noting the status of each register before and after each multi-cycle chunk. Also added a no-operation instruction (noop) which is used by our assembler to implement loops.

Meeting 11:

October 11th, 2018

Members Present: Find out next time!

Overview: Find out next time!

Major Decisions: Find out next time!