## **Will Yelton, WORK LOG**

**MILESTONE 1 WORK:**

Tuesday, January 14, 2020

Met with team [4.5 hours]

* We decided on a very modified load-store design and planned out the whole thing including instructions, transferring it to machine code, and a little bit of discussion about physical architecture and efficiency.
* I worked on writing the code for Euclid’s algorithm and giving a lot of input on the architecture design.

Wednesday, January 15, 2020

Corrected some errors in the Euclid’s Algorithm Code [30 min]

* Changed li to lli
* Added the machine code for J and Jal
* Changed the immediates in the load and store word commands to load/store in the correct place on the stack for our architecture.

Met with team [15 min]

* Decided on Milestone 2 tasks
* Some miscellaneous fixes for the design document and instructions

My tasks for Milestone 2:

1. Work on RTL instructions (all of us will do this)
2. Start working on an assembler

**MILESTONE 2 WORK:**

Tuesday, January 20. 2020

Wrote a basic assembler [30 min]

Corrected some errors in the design document [30 min]

Met with team [2 hours]

* Wrote the RTL for all or our instructions
* Discussed some aspects of the datapath
  + Created some designs for some components (like register file)

Wednesday, January 21, 2020

Added error checking and some more features to the assembler [1 hour]

Met with team [30 min]

* Corrected some errors in the design document
* Finished up some work on Milestone 2 (components for RTL instructions)
* Fixed comments in Euclid’s algorithm code

Tasks for milestone 3:

* Create some hardware components
  + Not decided who is doing which components
* Finalize assembler
  + Make labels work properly
* Work on component tests

**MILESTONE 3 WORK:**

Monday, January 27, 2020

Created and started testing a register file in Xilinx [3 hours]

* Also created and tested many other smaller components like decoders and 16 bit muxes
* Most of the register file works, but there are still a few problems that I have to work out

Fixed some problems and inconsistencies in our design document from previous milestones and added a small I/O section [30 min]

Tuesday, January 28, 2020

Worked a little more on testing on the register file [30 min]

Met with team [1 hour]

* Designed and drew out our datapath
* Ran through our instructions to make sure that they would work with our datapath
* Assigned tasks to be worked on for the rest of milestone 3

Wednesday. January 29, 2020

Added some components to the design document and helped in adding sections about testing and how to build the components [30 min]

**MILESTONE 4 WORK:**

Monday, February 3, 2020

Fixed some errors in the register file and added more test cases [30 min]

Made an instruction register [15 min]

Wednesday, February 5, 2020

Met with team [45 min]

* Created the finite state machine for our control

Added more tests to some components [1 hour]

* Added tests to the instruction register
* Added test to the register file
* Added tests for some other muxes and the decoder
* Fixed some small errors in the decoder and register file

Created an integration test with the register file, instruction register, and memory [1:45 hours]

* Put some instructions into memory
* Read them into the instruction register
* Used some of the instruction to write to specific registers or to move data from one register to another
* Mocked control signals using inputs and setting them in the test file
* Also found and fixed some small issues in the components included in the test

**MILESTONE 5 WORK:**

Monday, February 10, 2020

Created an integration test with the ALU, zero extender, sign extender, and the shift left units. [1 hour]

Created the memory unit that can map the 16 bit address onto the 10 bit address in the distributed memory. [1 hour]

Tuesday, February 11, 2020

Created tests for the integration test. Including fixing some errors. [1 hour]

Wednesday, February 12, 2020

Fixed some small things about the memory unit and the integration test. [30 min]

Helped a little bit with the control unit design.