Agnay Srivastava (Srivasa) Worklog

Milestone 1

- Worked on part of the instruction set/format table [1hr]
- Reviewed example assembly programs for errors [1 hr]
- Worked on the machine translation table for relprime [1hr]
- Working on fibonacci to demonstrate recursive programs using our language [2.5hrs]

- Few to no design decisions as Ethan finished most of the work by himself.

- Work for M2 to be decided, as most of us were active on teams, we only had a single meeting to finalize the design document before the 5:00 p.m. Tuesday deadline.

Milestone 2

- Thursday, January 11:
  - Met with ethan to understand the processor hardware logic that he had implemented in turing complete as well as how to run the assembler in python that he had built. [1.5hrs]
  - Discussed workload split for milestone 2. [30 minutes]
  - Wenzi and I were to split the rtl insrtuction breakdown while Emma and Ethan were to work on jump + return and the rtl component list. [30 minutes]
- Saturday, January 13:
  - Wenzi and I met and discussed homework 13 and assigned the exact workload split for milestone 2. [3 hrs]
- Sunday, January 14:
  - Ethan commited the initial push for the quartus files and discussed the rough breakdown of how we would work through the verilog + unit testing on quartus on teams. [1hr]
  - I finished the first pass of the rtl instructions [2 hrs]
- Monday, January 15:
  - Made incremental changes to correct rtl instructions. [15 mins]
  - Met with Emma to finalize the component table + rtl instruction table for M2 [1hr]

Milestone 3

- Friday, January 19:
  - Cleaned up and fixed the Single cycle RTL into its 17 respective categories, each of which correspond to an ALU. [3hrs]
- Saturday, January 20:
  - Started working on the multicycle RTL along with Wenzi. Finished first pass. [1.5 hrs]
  - Made a few minor revisions to consolidate the single cycle RTL. [30 minutes].
- Monday, January 22:
  - Fixed L_ALU_11 and T_L_ALU_11. NAND/NOR/XNOR logic was slightly off. [3o minutes]
  - Met with team and we collectively decided to switch our focus to integration testing rather than unit testing due to its sheer simplicity.
  - Tasks for M4 were assigned - Wenzi and I are to work on the ALU implementation, whereas Ethan and Emma are working memory and the assembler respectively [1hr].
  - My work this week was slightly less rigorous as I had 4 exams this week, and I'll be sure to make up for it by milestone 4.

Milestone 4

- Tuesday, January 23:
  - Finished Implementing ALU_4
- Saturday January 27:
  - Started working on memory lab 7. Finished writing tests for block_memory_control and testing it. [3hrs]
- Sunday January 28:
  - Finished lab 7. Wrote tests for connected_memory and verified that reading and writing to memory worked + control bits were being correctly set. [3.5hrs]
- Monday January 29:

- Demoed lab 7 to Dr. Stamm. Met with Emma to understand new Datapath design + pending M4 work. [1.5hrs]
- Wrote two unit tests for memory modules, verifying whether both ports were function on desired clock edges. [2.5hrs]

- Tuesday January 30:
  - Updated Unit testing methodology for memory. [10 minutes]

Milestone 5

- Friday February 2:
  - Discussed unit testing strategy for control and logical sector. We discussed the tasks that had been assigned, what sections of our datapath were working, and which needed further verification. (Type 11 instructions were broken and needed a minor fix) [2.5hrs]
  - Started writing tests for L_Logical_Sector.
- Saturday, February 3:
  - Finished Writing tests for the same. These tests weren't exhaustive, but the strategy for these was to take all variations of an operation / result with the same register names, and change the assembly code supplied to our verilog module. Eg: for geq, we compare 10>=9, 10>=10 and 9>=10 all on x1 x2 and x3.
  - Tests were lenghty,helper functions had to be written, reused and cleaned up a lot of wenzi's tests for the control unit. [9-11 hours]
- Sunday, February 4:
  - Discussed strategy for testing the entire processor, and how assembly code would have to be written via teams.