

Milestone 1

Developed Euclid's Algorithm and RelPrime (3 hours)

- Developed Euclid's Algorithm and 'reprime'. Developed test files for the code in assembly. Tested each module separately before bringing them together to test relatively prime. Developed a max-branch logic so that team could understand how much branching we would require worst-case before designing the ISA.
- Prepared assembly code as a preliminary model for instruction set architecture (ISA) development. This preparation was completed before the team meeting to facilitate immediate discussion on ISA.

Team Meeting on 1/2/2024 (Duration: 40 minutes):

- The team agreed on developing a load-store-based processor.
- We allocated 16 bits among registers, types, etc., and engaged in detailed discussions on optimizing and limiting the function and opcode to 4 bits, allowing for a maximum of 16 different instructions.

Team Meeting on 1/9/2024 (Duration: 2.5 hours):

- Began drafting Milestone 1, focusing on design philosophy and reviewing our ISA, including bit allocation for each type and instruction.
- Assigned opcodes and functions to each instruction and reviewed the process of converting these to machine language.

Milestone 2:

Meeting Scribe (Duration: 1 hour):

- Recorded notes during the meeting and implemented suggestions post-meeting.
- Revised the documentation to simplify language and correct errors discussed.

Reviewed RTLs (Duration: 30 minutes):

- Re-examined all implemented Register Transfer Languages (RTLs) and made minor adjustments.
- Tested RTLs.

Reviewed Design Components (Duration: 20 minutes):

- Reviewed essential design components necessary for Milestone 3 datapath designing.

Concise Alternative Logic (Duration: 1 hour):

- Examined logic for minimal branching, focusing on reducing byte offset and minimizing jumping. Currently trying to further minimize jumping and total instructions to implement relprime

Milestone 3

Meeting Scribe and Housekeeping Activities (Duration: 1 Hour)

- Note-taking during the meeting, capturing key points and actionable items. (30 Minutes)
- Post-meeting, engaged in a thorough review and implementation of the suggestions that emerged.
- Drafted and dispatched comprehensive housekeeping emails, incorporating all revisions prompted by the Milestone meeting.

Development of Python Scripts for RTL Modules (Duration: 3 Hours)

- Successfully created Python scripts tailored for eight distinct RTL (Register Transfer Level) modules.
- During the development process, identified and resolved a critical error in the Branch instruction, collaboratively rectified with colleague Harrison.
- Initiated the task of crafting a mini-interpreter aimed at simulating Mini-programs. This involved scripting encode-instructions linked to memory operations.
- Encountered a persistent issue in 'miniprogram1', but after consulting with Dr. Sher, the decision was made to focus on individual RTL testing for the interim.
- Documented my work to add to the Milestone 3 document, serving for the additional features in development of our project.

Development of Program Counter and Corresponding Testbench (Duration: 40 Minutes)

- Developed the Program Counter (PC) module and its associated testbench within Python, integrating \$display statements as a strategic alternative to waveform analysis.

- The testbench rigorously evaluated the PC module across various scenarios: reset functionality, precise incrementation by 2, and flexible address jumping capabilities.
- Additionally, the testbench was designed to assess undefined behaviors, particularly those occurring before the initial positive clock edge, confirming robustness and reliability of the PC module. All tests were successfully passed.