Tulsi Manohar, Work Log

Team Lime, Section 1

MILESTONE 1:

Tuesday, December 19th, 2023.

- Discussed with the team during class time to get a sense of how we planned to take the project forward. Establishing smaller, achievable tasks and ensuring everyone is on the same page every step of the way.
- Set up communication channels and decide when we'd be available to meet. We also talked about the need to prioritize the project and make changes in individual schedules to accommodate work times where everyone would be able to attend rather than having smaller meetings and having to spend more time bringing the others to the same level of understanding.

Thursday, December 21st, 2023. @6pm

- Worked on instruction sets and bifurcated them into categories based on requirements. How we could optimize our work based on the limitations of the project requirements as compared to what we are comfortable with in RISC-V.
- Discussed and analyzed RISC-V instructions to understand what we needed vs what we could go without for our specific processor and language.
- We all worked on the descriptions and specific notes for instructions as required.
- Created a memory map, and list of registers and worked on space allocation.

Thursday, December 28th, 2023. @2am (Indian Standard Time)

- The meeting was planned and held an hour later, unfortunately, at that time I was asleep and I couldn't attend. However, I did work on understanding what was worked on by my fellow team members before I attended the next meeting to ensure that I could contribute to subsequent meetings effectively.

Tuesday, January 9th, 2024 @2pm

- We rechecked all past work to ensure our solutions make sense.
- We then spent a considerable amount of time working on translating our code from assembly language to machine code. The same was done for our example code fragments as well.
- I worked mainly on the relPrime, greater, gcd_end, and other instances where jalr and jal were called which needed work based off of addressing indexes.

MILESTONE 2:

Saturday, January 13th, 2024.

- We focused on coming up with component descriptions at first, analyzing how we would need to adapt risc-v conventions to match our work.
- We discussed input and output lists for each of the aspects such as ALU, PC, etc and listed them down. In particular I helped with ALU, Memory and Programmable registers.
- In terms of the ALU, one of the first things we did, as a team, was come up with all the operations it could carry out such as addition, shifting, subtraction and even XOR.
- Something that took us a little bit to figure out was how we could use the ALU Control component in a different way as compared to what we had in the given sketch because of the fewer number of bits we use to represent operations in our instruction format.
- Control Unit: helped list the many outputs as well as inputs through this component. At this point we noticed a lack of uniformity in terms of how we were naming our inputs and outputs so I spent some time going over it and fixing them to ensure uniformity.
- Immediate Generator, this one prompted some discussion but it wasn't too bad. Was fairly easy to figure once all of us had the same level of understanding in terms of its functionality. This is in regards to why we were passing in the whole instruction rather than just the bits that hold the immediate, we soon realized that the way we have everything set up with regards to the ALU control this would work itself out.
- Memory, during this meeting although separating data and instruction memory was proposed, we decided to stick to just one for the both of them.
- To ensure commonality between the way each of the instructions were addressed, we worked together to make RTLs for at least one instruction under each format.After that, we split up the remaining instructions and worked on them individually during our own time.
- I was specifically responsible for half of the 2RI instructions.

Sunday, January 14th, 2024.

- During this meeting we collectively worked on correcting or addressing problems each of us had faced while working out RTL instructions for individual instructions.
- We focused primarily on getting milestone 2 tasks out of the way.
- An important change we made was eradicating the need for ALU Control by including its tasks and functionality in ALU. Adjusting the inputs and outputs for the same accordingly.
- Yueqiao suggested we add in comments to help make the entire design document more easy to read and we all agreed, adding in simple two or three word notes to give just a quick description of what each of the input and output names for and how many/which bits they utilized for the task.
- WE then had a lengthy discussion regarding the pros and cons of separating data and instruction memories which finally led to us doing so. Engaged in discussion that helped allocate appropriate inputs and outputs for the same.
- Lastly we spent some time reflecting on mistakes committed in milestone 1 and working on creating fixes for the same. Luke volunteered to redo the philosophy while

Naziia and I agreed to work on creating code snippets to support our assembly and machine code. I worked mostly on the translation from assembly to machine code.

MILESTONE 3:

Saturday, January 20th, 2024.

- We worked on drawing the data path, drawing and redrawing it on the whiteboard, understanding how we could change or alter each iteration to become more efficient and or easier to use. This also included brainstorming the different components we wanted to include or not include. I was primarily in charge of this task, later I worked on drawing the final version which was submitted.

Sunday, January 21th, 2024.

- Naziia worked on simplifying single and multi-cycle descriptions and the rest of us helped review and check for errors.
- Revised the datapath, rectifying MDR connections. Added an Instruction Register to our multi-cycle model, detailing its functions and RTL symbols, while Luke handled IR testing. Streamlined the single-cycle RTL for the instruction set.
- Refined component behavior and RTL symbols for multi-cycle components, focusing on clock edge-triggered instructions and necessary I/O modifications. Removed clock functionality from the ALU and Control Unit after a second review.
- We then worked on multi-cycle RTLs for the instruction types while Naziia typed for us.

Tuesday, January 23rd, 2024.

- Naziia and Yueqiao constructed a multi-cycle instruction summary table, rectified RTL errors and adjusted the ALUOut operation to the first cycle on Yueqiao's recommendation. I wasn’t as involved during this meeting as I had an exam in the next class on the same day.