Jordan Ansari Work Log

Milestone 1:

Sat, April 10, 2021: Met with team [30 min]

We agreed upon an accumulator processor, as it would be unique from MIPS but still usable and interesting to create. We also started working on the document for milestone 1. Most of the time was spent on the things which did not require much deliberation (such as necessary instructions and registers).

Sun, April 11, 2021: Met with team [45 min]

We continued to work on our design document, coming to a few important decisions- first, we realized we would need more registers than a single accumulator for our design to work. We agreed to add more accumulator registers, as well as temporary, stored, and argument registers. The rest of the night was spent adding to our design document- tomorrow we intend to work on the actual example code and its machine translation.

Mon, April 12, 2021: Met with team [45 min]

I worked on writing the machine code for Euclid's Algorithm. While writing it, I realized we'd need a few more procedures (set, setc), as well as more registers ($0, $v1-v2). We added these to accommodate these realizations, and the rest of the team worked on fine-tuning and coming up with opcodes to transform the assembly code into the machine language translation.

Milestone 2:

Mon, April 19, 2021: Met with Ian [70 min]

Ian and I worked on the Milestone 2 for a while. We contemplated how the RTL specifications would work, and I ended up creating the table while Ian looked at the rest of the milestone. After a fair amount of deliberation, we ended up with a table with which we were satisfied. While making it, we noticed a few things: firstly, beq and bne would not work with the current setup, as the instruction format is simply opcode and address. Therefore, we decided to change how the branch statements and slt work- beq branches to the given address if the current Cool Register is equal to 0, bne branches if the current Cool Register is not equal to 0. Additionally, slt now becomes -1, 0, or 1, depending on if the current CR is less than, equal to, or greater than the given register. This allows for branching while maintaining the 16-bit opcode. We also realized the opcodes were 1 bit too long- they were 6 bits, but only needed to be 5. Overall, I'd say so far this milestone has helped me understand the RTL concepts quite a bit better. In terms of the group, I wish we could communicate a bit better- it feels as though we may struggle if we don't learn to assign and split up tasks.

Tue, April 20, 2021: Met with Team [60 min]

I showed up to the group meeting quite a bit after everyone else, but still did a fair amount of work. Hunter had been working on the assembler the whole time, and made rather quick

progress, which is encouraging (he's nearly done). I worked on writing a recursive code snippet (multiplication), and also made some minor changes according to the feedback from the first meeting with Dr. Williamson. Overall, I feel more content with the progress so far- hopefully, we will continue the good work tomorrow and can wrap up the milestone cleanly. I will probably also do the optional lab tomorrow, as no one in our group has so far, which seems like a problem which needs resolving.

Wed, April 21, 2021: In-class work [160 min]

We spent the lab today doing work on our design document. I stayed in the classroom for about 40 extra minutes, wrapping up and making tweaks to mistakes I found. We wrapped up our RTL diagram and consulted Dr. Williamson to find any errors. We decided to store the cool registers in their own register file to make accessing and editing them easier, which is nice. I'm starting to grasp the concepts much better now, which is good. Hunter kept working on the assembler, and I believe he's close. We also realized our addressing was incorrect- since the instructions are 16 bit, we only need to increment the address by 2 per instruction, not 4. Overall, I feel pretty good about where we're at.

Milestone 3:

Mon, April 26, 2021: Memory Lab [100 min]

This morning I completed lab 7. There's not much to say about this- I feel more confident working with memory and Xilinx in general, and I think I completed it to the best of my ability. I believe Ian is also working on the lab- I think doing it separately then comparing notes will help us come to independent conclusions and agree upon the best method.

Tue, April 27, 2021: Group Meeting [90 min]

Tonight, our group met to pursue Milestone 3 and create a plan. Logan had come up with a datapath, so we based most of our work off of that. I made some edits to the design document, but most of this was spent planning. We discussed if we would implement pipelining and whether or not we would be multi-cycle- we agreed upon multi-cycle and decided to implement pipelining if we had time. I think it would be feasible for us to do, so hopefully it gets done.

Wed, April 28, 2021: In-class work [120 minutes]

Today we worked on the design document and came up with test cases for the components. Hunter and I spent most of the class working on the design document, while Logan and Ian worked on creating the components in Xilinx. I also demonstrated lab 7 to Dr. Williamson, and he explained how the memory addresses will have to be 10 bits due to the constraints on our processor. I'm glad Logan and Ian made decent progress on the components, as it feels as though our project is really getting off the ground now.

Milestone 4:

Mon, May 3, 2021: Group meeting [100 minutes]

Today was more of a conversation than work- we spent most of the time clearing up misconceptions and planning the future tasks. Logan and Ian have been working on creating the datapath in Xilinx, while Hunter has worked on the test cases. I'm a bit confused on how memory is intended to work, as Dr. Williamson mentioned we can only use 10 bits for an address instead of 16- does this mean we must store everything in 1024 addresses? It's a bit worrying, but we've made quite a bit of progress in other areas, so I think we'll be alright.

Tue, May 4, 2021: Independent work [80 minutes]

I worked on determining control signals for each instruction based on our given datapath. However, in doing so, I came across a few discrepancies, and so created my own using Logan's advice that we would then compare and make edits to based on my suggestions. Truthfully, the hardware is not my strong suit, so I found it quite difficult, and imagine my datapath has more problems than his design. However, it will hopefully allow us to find any errors that may come up and make finishing the project easier. The next two weeks are looking project-heavy in all 4 of my classes, so it'll probably be a lot of work.

Milestone 5:

Tue, May 11, 2021: Group meeting [130 minutes]

This meeting, I worked with Logan on the control signals and converting them to multicycle by creating a state machine. We also discussed how memory would work, and I showed them the results from lab 7 and how we would likely need to connect everything. Our datapath is nearing completion, which is exciting.

Wed, May 12, 2021: In-class work [100 minutes]

Today, after taking the Pipeline make-up problem, we worked further on completing the datapath. I helped Logan fine-tune the control signals, while Hunter and Ian worked on testing the components. I'm feeling pretty confident about our progress, but I'm apprehensive about actually running Euler's algorithm- specifically the recursion, as while storing information in the stack is not hard in theory, I worry something will go wrong. Truthfully, I did less work this week than I would have liked, but it felt as though my groupmates were on top of everything already.

Milestone 6:

Mon, May 17, 2021: Met with group [90 minutes]

Today we continued marching closer to the completed datapath. We discussed memory implementation and finished writing test cases for all the subsystems and discussed how to implement I/O- it doesn't seem like it'd be too bad. We hopefully will not have much more to do- the datapath just needs to be assembled.

Wed, May 19, 2021: In-Class work [90 minutes]

In class today, we finalized essentially everything. We know that single instructions execute correctly, so all that's left is to import multiple instructions from memory and run Euclid's algorithm. I'm hoping that goes well- so far, everything has gone essentially without a hitch. If that streak continues, we should be able to wrap up the project smoothly. I didn't have much to do today, so I instead worked on creating our presentation. I created essentially all the slides we would need and added information to about half of them.

Thu, May 20, 2021: Met with group [420 minutes]

Today was the final meeting. I fixed Euler's algorithm to work with our current instruction set- which caused me to realize we needed to add *slt* back to our set. Logan was able to quickly add this to the control, which avoided a major headache. After everything was assembled, we put the code into a .coe file and added the block memory to the datapath. However, we ran into major trouble with the instructions being decoded- the opcode was not updating and we suspected it was due to a timing problem. After debugging for hours on end, rewriting the program three times, fixing errors in the control state, and finding errors in how the registers were stored, we *finally* got it working.