Ethan Hutton Work Log

Milestone 1

Wednesday, September 27th

- Met with Wesley and Ethan in class to read over milestone 1. Discussed which processor we want to build and which instructions to use. Tried to get everything to fit into a 4-it opcode to allow space for other things like registers and immediates. Discussed trying to have instructions act like other processors such as accumulator and/or mem-to-mem [30 minutes]

Sunday, October 1st

- Discussed with the whole team and discussed changing processors. After some debate and discussion we ended up settling on memory to memory since it looks to be the most efficient. Designed our own green sheet and got the majority of the milestone done. [~ 3 hours]
    - I described the different instruction types, talked about our addressing mode and calling conventions. Instruction types are long but necessary for memory to memory. We're using direct addressing since it's simplest and most efficient to just pull directly from memory for memory to memory. The only calling convention we have is to push the return address on the stack before jumping. This is so we can pop it off the stack and return to where we need to after.
- Our current priority is to just add in the few missing points and meet up on wednesday before break to assign some tasks if we want to work over break

Monday, October 2nd

- Finished up what I was assigned to do and formatted what we had into a nicer document [~20 minutes].

Tuesday, October 3rd

- Met with team to discuss next milestone and pushed journal and design doc [20 minutes]

Tasks assigned for the next milestone:

1. Take an instruction type and create pseudocode to break it up into steps
2. Take the pseudocode for the instruction type and create a component out of it. [est 2 hrs]

Milestone 2

Wednesday, October 4th

- Reviewed milestone 1 with the team and Dr. Stamm. Discussed ways of returning and passing arguments to a function. Met with the team again to make sure we know who is doing which instruction type for milestone 2 and to make sure that milestone 1 is complete. [90 minutes]

Saturday, October 7th

- Made a draft of sorts for the instructions I was assigned, J and L type. Also created some components that will hopefully be added to. Going to wait for the other instruction types to start testing them. [~1 hour]
    - J type is a little janky. I think splitting it up into two separate instructions might be better. Having it all as one right now feels odd having if else statements
    - The components I made don't seem very well thought out so I hope there are corrections made to them. I think some make sense like the adder having two 16 bit inputs and a 16 bit output, but the memory is a little goofy. My reasoning is that we have a max of 56 bit instructions. Therefore we will at most load 3 16 bit addresses to output. We also only write a max of 16 bits unless we write more than one thing at a time. Register is basically the example copied, but adjusted to our processor

Monday, October 9th

- Met with the group to get most of the milestone done. We all took our rtls and tweaked them into a table that looks like it works. We still need to test them and we will have to break that up. We ended up getting rid of any garbage bits that an instruction didn't need and now it will only use as many as the instruction calls for. We decided on some of the testing conventions and made our components. [~2 hours]
    - Did the J and L type as well as the ALU Component

Tuesday, October 10th

- Made a rough draft of a test rtl and discussed it with Wesley. Also made the patch notes (changes to assembly and machine language instructions) [~20 minutes]

Wednesday, October 11th

- Met with the team and finally tested the rtls. Adjusted part of the design document since we switched s1 and s2 in our instruction types. Assigned tasks for next milestone [~30 minutes]
    - I did the tests for the j and ja types

Tasks assigned for next milestone:

1. Draw out the datapath
2. Write out the components and their tests

Milestone 3

Thursday, October 12th

- Tweaked some things after our meeting with Stamm. Made sure everyone knew what they were doing for the next milestone [~30 minutes]

Friday, October 13th

- Started creating a datapath with Wesley. We are using muxes to choose registers and it looks quite scuffed. We went through and did all the rtls, though so it should hopefully just need some modification. [~60 minutes]

Sunday, October 15th

- Redid the datapath. Had the idea to just have the memory write to each register, but it will only write if the flag to that register is on. Makes it look a lot nicer and makes more sense. Now we'll have to work on describing the control signals. [~45 minutes]

Monday, October 16th

- Met with the group and did the final redesign of the datapath (hopefully). Got rid of an extra memory block and just made it so that you could choose from memory addresses or the PC. One interesting thing that hopefully works is that we will read a memory address into register A and then a cycle or two later we will read from the address of A into register A so now register A will have the value of A. This is the only thing that I'm concerned won't work as intended, but if it doesn't, then we can just create another memory block and more registers to write to instead of overwriting the old one. It should work, though… Right? Right?? [~40 minutes]
- Added control signals. They look pretty alike, but most of them are write flags that, well, write. I'm hoping that they are pretty self-explanatory. [~20 minutes]
- Started on the iterative plan. Wondering if it is better to build it step by step like building a datapath from rtls or in blocks that we eventually connect. The milestone says subsystems which implies blocks to me. I'll go with that. [~20 minutes]

Tuesday, October 17th

- Created the iterative plan with Wesley. Ended up doing what looks like a combination of generic blocks and then those blocks combine to form the instructions. So the first 3 stages are blocks and the next 3 or 4 are building the datapath based on the rtls. We didn't really include anything about muxes even though they're important since although they're components, they aren't big components listed out, but they are used and implemented as we go. Same thing with flags. [~100 minutes]

Tasks assigned for next milestone:

1. Specify control units
2. Do one of the stages for the integration plan

Milestone 4

Thursday, October 19th

- Tweaked some things after talking with Stamm. Assigned who is doing what stage. [~30 minutes]

Sunday, October 22nd

- Made a finite state machine by hand. Tweaked the datapath a little bit and RTL a little bit. [~1 hour]
    - We adjusted the RTL so now JA takes one more cycle. We could have done it all in one, but it would have required more bits and adjustments to the datapath that it just seemed easier and clearer to make it another cycle. In the other case for SP, we just didn't have anything that made A = PC + 1 so we had to account for that in both the datapath and control/FSM.

Monday, October 23rd

- Met up with the team to see where we are at with the milestone. I started stage 3 and got some of it done. Hopefully the testing goes well. $6 is definitely the best option. [~90 minutes]

Tuesday, October 24th

- Actually finished stage 3 of the implementation. Now it should just be a matter of connecting them for stage 4 and adding in various muxes and flags for later stages. [~90 minutes]
    - I only tested 3 things. I don't know if more should be tested. It looks okay to me, but I wouldn't be surprised if there needs to be more. Also something that's a little weird is that if we want to store something in memory, we have to wait a cycle where it's all 0s and after that cycle it will output the correct value. Not really sure what that's about, but it works! I think.

Tasks assigned for the next milestone

1. Implementing control and testing it (I kind of just assigned this to myself, we did not really have time to talk about this).

Milestone 5

Wednesday, October 25th

- Tried to figure out some things with lab7. Checked it over with Stamm and realized we were all good. [~60 minutes]

Sunday, October 29th

- Spent some time looking over control to make sure it was okay. Made some small changes. I think the only thing we need to do is add a blank cycle for branch and maybe set other write bits to one whenever we write. We do that when we reset, but if we set it to one in a previous state we never set it back to 0 so I think we need to do that. [~60 minutes]

Monday, October 30th

- Made the rest of the testing for control. Everything looks to be in order. Had some problems connecting control to the actual datapath, but Garrett figured it out. I think he changed it to read on the negative clock edge, but changed it back. Now we set control bits on the positive edge and change states on the negative edge. [~60 minutes]

Tuesday, October 31st

- Okay, I guess Garrett has it all figured out because relPrime just works?! I thought we would need a bit more system testing and a blank cycle for the branch, but if he's getting the correct output then I can't be mad. We just spent most of the time trying to make sure that tests pass because he had to change the control a little bit, but it should look okay now

Wednesday, November 1st

- Now we're trying to move on to milestone 6 and some extra features. I'm looking into CPI and instructions while Garrett tries to redo the memory a bit. Currently it compiles way too slowly since we had to adjust it to how we're using it, but we're trying to make minimal changes to it in hopes that will work. [~2 hours]

Tasks assigned for the next milestone

1. Get the execution time for our processor
2. Work on the final design doc and presentation (assuming that everything keeps working)

Milestone 6

Friday, November 3rd

- Worked a lot on making the assembler work. I think it works. Hopefully. I added some of how it works to the design doc [~4 hours]

Monday, November 6th

- There were some errors found in the assembler I tried to fix them. We couldn't have negative numbers before, but now it should be able to handle them [~1 hour]

Tuesday, November 7th

- Added some input/output for the assembler. It's a bit weird because they're treated as labels in the assembler, but they're actually memory addresses. It should still work though. [~ 1 hour]

Wednesday, November 8th

- Updated the design doc for stuff about the Assembler. [~ 15 minutes]

Friday, November 10th - Sunday, November 12th

- Made sure everything worked. Created and developed a final presentation along with finishing up the design doc. Rehearsed our presentation a few times to make sure we were mostly on time.