Milestone 6

Wednesday, February 21, 2024

Group Work [TIME]

- Finalized the paper.

Monday, February 19, 2024

Group Work [1620 min]

- Finalized everything but RelPrime and branches
- Elenaor and I fixed all branches. At this point, the processor works correctly.
- We probably need to finish implementing SO.

Stage Implementation:

☒ Stage1
☒ Stage2
☒ Stage3
☒ Stage4
☒ Stage5

Tuesday, February 13, 2024

Individual Work [50 min]

- Continue work on `asm2bin.py`. It now outputs in hex. Each instruction begins with a "0x" and ends with a newline. These are easy to change if it isn't desirable.

Group Work [50 min]

- Dan and Elenaor are working on manually converting a lot of RelPrime to machine code to verify my assembler.
- I refactored asm2bin. It outputs hex with labels converted to addresses now. IT HASN'T BEEN VERIFIED TO WORK YET!!!

Monday, February 12, 2024

Group Feedback [30 min]

- Explain SR better.
- Explain LLI better.
- The RTL table is good!
- We should probably reorder the design doc so it's easier to read.

Group Work [45 min]

- I worked on making the design doc prettier.
- We communicated with Jada about how we feel that she needs to give us realistic deadlines.

Met with Williamson [6 min]

- Make the assembler do the right stuff.
- Make the assembler output hex values, not binary.

Group Work [180 min]

- SR was broken. I fixed it.
- Jada got Stage2 working.
- Elenaor created `StagedDatapath.v`. It hypothetically should work.
- Next steps: I need to make the assembler do the assembler stuff. Dan will manually assemble RelPrime to make sure I'm not being an idiot.

Sunday, February 11, 2024

Individual Work [90 min]

- I still don't have a working memory module. I'm extremely frustrated at this point because it's excuse after excuse from Jada on why it's not getting done.
- I told Dan and Jada that if their stages aren't done by 1:00 PM, I'll write them myself.
- I finished implementing all test benches except Stage2.
- asm2bin now takes a file. Its usage is `python asm2bin.py target destination`.

Individual Work [180 min]

- I heard back from Dan and Jada. Dan's working on their stuff and I don't see a problem.
- I went ahead and implemented Stage1.
- I asked Jada to write the test bench for Stage2 because she knows more about Lab7 than me.
- I am confused why Stage5 is broken. It looks like the ALUoutput register is always 0. Also, the SR register is wrong. I spent ~2.5 hrs on this so I'm going to take a break.
- There was a bunch wrong with the ALU. I think it works now, though.

Individual Work [90 min]

- I think I figured out what's going on with Stage5. I think that the problem is that I can't feed SRout back into the input for the ALUsrcB MUX. I think the best way to fix this is to have the datapath itself deal with it.
- I just plugged SRout back into the ALUsrcB MUX and it looks like it works. I'm worried that the MUX expects a 16-bit input and SRout is only 2 bits, though.

- I'm trying to be understanding, but it still looks like Dan hasn't actually run the tests, just compiled the code.
- Dan said they have other work to do so I'm gonna just do all their stages.

Individual Work [60 min]

- Continuing work on Stage4.
- It's giving an output of "z" and I'm not sure why.
- Lo and behold, `Register.v` is broken too.
- I got Stage4 working. I honestly don't know what I did. I just started changing values around until it worked. I think it was grabbing a reg from somewhere else.

Saturday, February 10, 2024

Individual Work [210 min]

- I'm hoping I'll have time to work on asm2bin.py but I'm not sure if I will.
- I pinged Dan and Jada again because they haven't said if they're okay to work on their parts.
- I'm assuming that they're capable to do their parts because they're pretty small so I'll continue my parts.
- I finished creating test benches 1, 3, and 4 today.
- Creating test benches
  - ☒ Stage1
    - Both the Register and Mux files had different capitalization from their test benches. I sincerely doubt either of them were tested before their designer said they were complete.
  - ☐ Stage2
    - I can't do this yet because Jada hasn't finished memory.
  - ☒ Stage3
  - ☒ Stage4
  - ☒ Stage5

Milestone 5

Thursday February 8, 2024

Meeting with Dr. Williamson [20 min]

- We didn't follow the integration plan and we should have.
- Update the integration plan for what we *actually did* and what we can do going forward.
- Can't we just test components individually?
  - The internal wires can't be written to.
- The actual `Datapath.v` file should only have 4-ish things in the file if we did it correctly (1 per step in the integration).

- We should probably just put the components in their own files.
- Williamson says to test it for a few hours fully integrated and see if it works. If it doesn't, just put it in its own files.
- Datapath should instantiate Control. Datapath should have CLK, reset, input, output.
- Once it's running, we need an execution time estimate: $executionTime = instructionCount \cdot cycleTime$.
- Use a test bench to count cycles.

Tuesday February 6, 2024

Individual Work [45 min]

- Continued working on a test plan for the datapath.
- I haven't had time to implement the datapath test bench yet. I'll probably do that after the exam tomorrow.
- I'd also like to get the assembler to handle files instead of just command line arguments.
- Hopefully, I can have it do pseudo instructions, too.

Individual Work [120 min]

- Spent most of the time debugging errors from other peoples' Verilog files.
- I reformatted Datapath.v to allow for tb_datapath.v to work by setting all the control bits as inputs instead of local variables.
- I need to talk to the group about verifying their code. I'm getting the feeling that some people are just writing stuff down and not verifying it.
- Tomorrow, I plan to implement tb_datapath.v unless another unforeseen situation happens.

Monday February 5, 2024

Individual Work [30 min]

- Took a 2nd look at Control.v and fixed the compile errors.
- It looks like the Control module was being declared in a combination of the old and new way to declare modules. I changed the module declaration to make it follow the specification for the new way.
- I'm not sure if the FSM works as advertized but I'll look at that later today.
- Jada informed us that she is only half way done with Lab7. We are waiting on her to finish the lab to implement memory. I am worried about going too far with control without Lab7 being done because memory might behave differently.

Individual Work [35 min]

- I'm working on "A system test plan to test your assembled datapath."
- I split the datapath into 5 sections:

- ☒ PC Register and IorD MUX
- ☒ Memory and MDR Register
- ☒ IR and CompCode Registers and Immediate Generator
- ☒ RegOp MUX and Reggie
- ☒ aluSrcA and aluSrcB MUXes, ALU, and ALU_output and SR Registers

- I'm confused for the input signals for items that are further along the datapath. I'm assuming that you can manually set the inputs to the modules without causing any errors.

Sunday February 4, 2024

Met with Elenaor [210 min]

- We're currently being held back because Jada hasn't finished Lab7 yet.
- Worked on fixing control and datapath.
- I'm confused about how to begin Datapath.
  - My idea is that datapath is kinda dumb i.e. it doesn't really do any logic, it just wires the components together. All the logic is done in the FSM.

Individual Work [90 min]

- It looks like Elenaor's implementation of Control.v has a lot of issues. I'm trying to debug it but it's hard because it's not documented very well.
- Elenaor got back to me and she will fix Control.
- I started converting the FSM to a draw.io diagram because I had trouble reading it.

Milestone 4

Tuesday, January 30, 2024

Finished Memory [20 min]

- Finished the testbench. It tests writing and reading.
- Haven't connected MDR to memory yet.

Monday, January 29, 2024

Finished Immediate Generator [35 min]

- Figuring out the test bench was the hardest part.
- Used a weird syntax that I found on Stack Overflow for the actual sign extension.

Met with Williamson [15 min]

- Helped me figure out some stuff with ModelSim.

- We fixed Quartus not including files as well.

Worked on Implementing Memory [40 min]

- This Stack Overflow page was good: https://stackoverflow.com/questions/7630797/better-way-of-coding-a-ram-in-verilog

Sunday, January 28, 2024

Worked on Immediate Generator [90 min]

- Realized the input for ImmGen was wrong on the design doc. I changed the input from [15:0] to [11:0].

Friday, January 26, 2024

Met with the group and decided what to do [30 min]

- Jada will do the registers.
- Dan will do SR and CompCode.
- Elenaor will do ALU and control.
- I will do Imm Gen and MEM.

Milestone 3

Tuesday, January 16, 2024

Things to do for M3:

- Have a short English description of how each RTL works.
- Put an example of a minimum function call on the calling conventions.
- Add instructions on how to use asm2bin to the design doc.
- Everything on course website.

Questions for Williamson:

- Is 2^10 bits enough for text memory?

Met with Elanor [75 min]

- Tried converting this note sheet to markdown but it didn't work out.
- When creating a minimum function call example, I realized that we have no way of LW and SW to the place SP points to. We need to create a new instruction.
- Created LMEM and SMEM. They load from and store to the place in memory that the memory address provided points to.
- Created a minimum function call example. Saved the text into design/MinimumFunctionCall.txt.

Individual work [15 min]

- Added instructions on how to use asm2bin to the design doc.

Wednesday, January 17, 2024

Met with team [30 min]

- Created a kanban board to hold each other accountable.
- Assigned each person some tasks.

Friday, January 19, 2024

Individual work [30 min]

- Set up Quartus
- Reordered log

Sunday, January 21, 2024

Met with Elanor [45 min]

- Confirmed the datapath
- Planned the next steps for the project
- Learned that I'm not allowed to talk during the M3 meeting

Tuesday, January 23, 2024

Met with the group [30 min]

- Discussed with the group about not getting the work done.
- I was ensured that it would get done by "well before 5:00."

Thursday, January 25, 2024

Met with Williamson [15 min]

- Integration plan:
  - Explain what each component does after 1,2,3, etc. cycles.
  - Write tests for each step.
- Addi:
  - We haven't done the actual addi yet.
  - Look back at RTLs and make sure they're actually doing what they're supposed to be.
- CMP
  - What is the compare actually doing in the last cycle?
  - We probably need a cycle where we're doing ALUOut.
- Overall:
  - Look back at RTLs
  - There is a .gitignore for Quartus in the milestone instructions.

Milestone 2

Sunday, January 14, 2024

Met with the team over Teams [60 min]

- We decided to make the CMP register only 2 bits. We did this because we were worried about the number of lines we could jump.
- I will look through the ASM and add the machine code.

Individual work [210 min]

- Finished ASM
- As I went through the doc, I realized that the calling conventions were done incorrectly. They were set up with multiple registers even though we only have one.
- I've decided we're doing big endian because it is the easiest to understand.
- I decided to make my life easier with this table:

Abbreviation	**Memory Location**
RA	0x0400
SP	0x0402
A0 - A6	0x0404 - 0x040F
S0 - S7	0x0410 - 0x041F
T0 - T15	0x0420 - 0x043F

- I also noticed that LUI and LLI were set up backwards, so I fixed them.
- Concern: Will having 2^{10} bits for program access be enough?

Monday, January 15, 2024

Individual work [70 min]

- Updated assembler to include BMEM. Right now, it adds an additional bit to the end of the immediate and I'm not sure why.
- Updated the table in previous day to be correct.
- I modified the memory map because it didn't have all the necessary components (missing storage locations for RA and SP)

Individual work [30 min]

- Read through the design doc to verify everything is there.
- Made the RTLs more compact.
- Removed the dynamic part from the memory map because it isn't needed.

Met with Williamson [10 min]

- Wants to have Generic Components contain EVERYTHING in RTLs
- Changed addressing modes to formats
- Wants an example function call in ASM from calling conventions

Milestone 1

Saturday, December 23, 2023

Met with the team over Teams [90 min]

- Decided to create a processor based on an accumulator.
- Processor will have 4 registers:
  - Input register for taking in the input.
  - Output register for outputting information.
  - "Reggie" the actual accumulator register.
  - Status Flag (SF) register for recording the status of comparisons.
- Created addressing modes. My only concern is that M and B types seem basically the same. Maybe we can combine them?
- Created a green sheet with the group and I converted it to markdown.

Monday, January 8, 2024

Met with Elanor to finish up the ASM and finalize the submission [80 min]

- Fixed the Green Sheet. Some opcodes were repeated, and we were missing one instruction.
- Looked over the Design Document to make sure there were no errors.
- Got rid of the Google Sheets version of the Green Sheet because it's easier to only maintain one file (.md) instead of both.

Friday, January 12, 2024

Met with the team over Teams [20 min]

- Decided to put the Green Sheet in Google Docs and to delete the MD one.
- Decided I should work on making a program to convert ASM to machine code.
- Decided to do absolute addressing for memory.

Created ASM2BIN.py [60 min]

- I decided to write it in Python even though I haven't used it in years so I had to keep looking up syntax stuff.
- Currently it's just command line but I want to have a GUI.
- CMP codes must be in binary. Everything else is decimal.
- Usage: python asm2bin.py