Joy Stockwell

Friday 9/28

7:30 to 9:30

We worked out basic functionality. We decided on accumulator architecture and hashed out the very basics of the language. We wrote out pseudocode for the algorithm we'll implement and brainstorm commands that we'll need sooner or later. We planned to reconvene on Saturday afternoon.

Saturday 9/29

8:00 to 9:30

I wrote the assembly code for the outer function of the Euclidean algorithm and commented it heavily with the assumptions I made so we could review thsoe assumptions during our meeting.

12:15 to 1:45

We went through my code discussed the issues I came across. We discussed conventions that I used and whether to keep them. We figured out what instructions were of what type. We did documentation. We decided that Matthew would write code snippets, Maura would continue documentation, and I would continue working on assembly. We planned to reconvene Sunday afternoon.

Sunday 9/30

4:00 to 5:00

We discussed problems Matthew encountered while writing snippets. We proposed another accumulator to make up for them. I left early because I'm not feeling well.

Monday 10/1

10:45 - 11:15

We came to office hours to discuss problems we've encountered. We spent most of the time discussing options for the accumulators.

7:00 to 7:45

We decided to go with a mix of our two solutions because it gives us a little novelty and a little familiarity at the same time: two accumulators and the ability to switch between them. We decided to keep sub because making the second accumulator negative before adding it would be annoying. We decided to use signed integers so it would be consistent with the (presumably signed) integers in the accumulators. We fleshed out commands and calling conventions and delegated tasks.

7:45 to 9:15

I wrote gcd(a, b) in assembly and updated relPrime to reflect our recent changes.

Tuesday

We decided not to meet because of employer events.

Wednesday

We met at the office during the class time and agreed to reconvene after the chaos of the carrier fair died down. We agreed to write RTL and discuss at our next meeting.

Thursday 10/4
9:00 to 9:15
I wrote steps for the last 6 commands.

Friday 10/5
9:00 to 9:45
We read off what we had written during the carrier fair. We discussed RTL and drew tentative block diagrams. We agreed to start meeting at specified times; we felt we'd been too vague with our starts and were losing time that way. We plan to discuss whether we have A & B regs, at, or all three, along with how we want to do I/O and our shopping list. We plan to meet again at 9:30 to 10:30 in the morning and 7:30 to 9:00 on Saturday night.

Saturday 10/6
9:30 to 10:30
We discussed our to do list, made state diagrams by ourselves, and discussed two of five state diagrams together. We planned to reconvene at 7:30. I wrote on the board as the others read off their commands to me. We kept a list of the control bits we needed. I was a little worried that we weren't far enough ahead to keep from falling behind in later stages.
5:00 to 5:20
I searched in the book for helpful information, but wasn't very successful.
7:30 to 9:00
We discussed the remaining command types, began typing them up, and started a tentative diagram of our hardware. It developed that my gcd and relPrime documents had never been translated into machine code. I thought this was very odd; I specifically remember asking Ben to do it and getting confirmation that he would. He did it tonight. I double-checked that everything from the current milestone was under way; it seemed to be. We decided to replace our flagbit with a decoration.

Sunday 10/7
1:30 to 2:00
Since I had been the one writing on the board for the earlier meetings, I didn't have a chance to type up my RTL. I did that now. I made a note to discuss the RTL of load and store with my group because we forgot about it last time. I tried to copy spek for most of the procedure, but we had a value that I didn't understand there; I made a note of that, too.

4:00 to 5:00
We tied up loose ends in our RTL descriptions and tables. We decided to make primary memory the first 2^8 addresses and to say you had to use shelly to access the rest. We're not sure what testing is supposed to look like. We discussed todos for the rest of the milestone. We decided to do interrupts for I/O and we got the

Status as of 4:37:
We have an RTL description of each individual instruction. Maura is putting together a chart with the rtl for jump, arithmetic, compare, swap, load, store, and stack instructions.

We have a list of generic components under "shopping list". It contains input, output, and control signals. We specify the number of bits in the control signals on the next page. The input and control signals are followed by explanations of what they do. We have descriptions of what RTL component each item implements.
Matthew is working on a bubble graph that will be our test to verify that our RTL works.
We will wait until the end to double check our state diagram and add a list of changes since last time.
Maura is keeping our design journal up to date.
We are each responsible for our own personal journals.

Our jobs for next time:
Ben is translating the Euclidean algorithm into hex (which is late from the last deadline).
Maura is finishing RTL charts and a table version of the state diagram.
Matthew is doing lab 7.
I am checking the hex for Euclid's algorithm when Ben is done. I am checking that each step moves data from a reg to a reg when Maura is done. I am doing lab 8 (may ask for help) and going to office hours to ask about testing and the rising clock edge issue.

Monday 10/8
1:20 to 1:35
I came to office hours to ask questions.
Note: I told Ben and Maura to email/dm me when they had their jobs done so I could check them, but they didn't by Tuesday morning. Maura's boyfriend was in the hospital.

Tuesday 10/9
9:00 to 9:45
Matthew and I started writing tests. Ben came and told me that he had stopped working on the assembly to machine code chart because he had a question about the assembly code. I told him to @ me in the groupme next time and we worked the problem out quickly (I'd forgotten a jimm inst). Matthew and I continued testing and Ben continued with the assembly to machine code. (I think Maura was with her boyfriend.)
We plan to reconvene tomorrow during the class time after the meeting.

1:25 to 2:30
I checked that everything but the journals for the milestone was pushed. (I didn't want to open my teammates' journals.) I made my journal into a pdf and pushed. I started lab8.

Monday 10/15
12:40 to 2:30 (There was ~15 minutes of dead time after I was done with the reading but before I could clarify the lab instructions.)
I went to the CS lab to do lab 8 and read about kernels in Apendix A of the book. I got the lab signed off in class.
2:30 to 3:20

We asked clarifying questions about the milestone and split up the work. I was put in charge of making adders and hand drawing the datapath. I updated the documentation of our registers to reflect the fact that we no longer had a register file. I cleaned up files in our git repo.

Tuesday 10/16
2:30 to 3:50
I researched adders and tentatively decided that we would not have to make any because we could just use the + in verilog. I researched interrupts.

Wednesday 10/17
10:00 to 11:00
I double-checked the milestone checklist and worked on the diagram. I realized that we hadn't discussed integration, and planned to bring this up with my team members. I also noticed that our design journal was out of date and that we had labs due whose status I didn't know, so I planned to make sure that was taken care of at the meeting.

1:35 to 3:20
We wrapped up our individual work (testing, working on journals, reading the next milestone, double-checking, etc.). We said that we would need very little work on integration for this milestone because we are doing multicycle. I drew a zoomed-in diagram of our registers and updated our rtl to reflect the removal of the A and B registers. I worked on fixing up the RTL.

Thursday 10/18
2:30 to 3:20
We planned to get the remaining labs signed off. We went over the next milestone as a group. I continued fixing the RTL.

Friday 10/19
2:30 to 3:20
We went to the meeting and reviewed our notes from it.

Saturday 10/20
12:30 to 2:20
We discussed an integration plan and I wrote it down. I made a public todo list from my notes. I started doing lab5 again. Matthew and I discussed the issues we'd run into with interrupts and I wrote them in a questions email. We delegated tasks. I have to admit that I'm a little concerned; I think I'm the only one feeling time pressure.

7:45 to 9:00
I started looking into the first step of integration testing. I didn't make progress, because I couldn't figure out how to put two interracting modules in the same verilog file. I read about verilog, though, and looked at the code of the test benches for completed projects. I noticed our ALU was missing one.

jobs as of 10/20 at end of afternoon meeting:
Matthew: update state machine to cover all cases
Maura: build control unit
Joy: main responsibility for kernel
Ben: help Joy with kernel and make truth table

Sunday 10/21
4:00 to 5:25
I came to meet Ben, but he was late so I continued researching verilog. He came after ten minutes. We sucessfully tested PC and its mux! It was working. I agreed to keep working on the pc block and Ben agreed to work on updating our RTL.

Monday 10/22
2:00 to 2:30
I worked on integration testing and brought questions to Sidd.
2:30 to 3:20
We discussed what sort of extra features we wanted for our final project. We decided to support interupts, checking for one at the beginning of each instruction rather than at every cycle. We felt that we would not be implementing enough exceptions to make it worth it to have a checking procedure that ran at every cycle. We also decided to put the project on an FPGA board during the final week. We decided we would have automatic output and tentatively decided to support arithmetic overflow exceptions.
7:30 to 8:50
We worked out the hardware that we would need to support interrupts. We decided who would do what hardware implementation. I took responsibility for the pc and sp blocks. We agreed to meet up from 7:30 to 9:00 again tomorrow.

Tuesday 10/23
7:30 to 9:00
We started a few minutes late because Ben's practice ran over. Matthew helped me clear up the problem setting PC to PC + 2. We ran into another problem with getting the output of our pc_block to be the same as our pc register. We discussed our status and decided to meet after the test tomorrow to continue working on integration. I would write in the design process journal in the morning tomorrow as well. We decided to pseudo-code more tests and continue with integration.

Wednesday 10/24
12:00 to 12:45
I updated the design process journal because Maura spent all week working on the control unit. (She's working really hard--I think everybody is at this point!) I noticed our integration plan hadn't been updated and updated it.I had some extra time, so I looked at my test from yesterday and noticed that pcSrc wasn't changing when it should have been. I couldn't get it to change, even when I used <= instead of =.
3:15 to 4:15
We tied up loose ends. Ben helped me get pcBlock to work!

Friday 10/26
2:30 to 3:20
We planned our weekend meetings. I typed up our new integration plan. I made a page titled "unit tests" and asked my teammates to write about the unit tests they wrote there. I double-checked that we had plans to tackle everything on the milestone in time.

Sunday
3:30 to 3:45
I prepped for the meeting by neatening and typing up the end of my pc test.
4:00 to 4:45
I had a merge conflict and Ben helped me clear it up. We were supposed to start working on integration this meeting, but I started work on sp because he ran into an obstacle with the memory test.

Monday
1:10 to 2:15
I worked on the coprocessor. I defined and tested the logic for determining whether the cpu was in user or kernel mode. I began implementing its registers.
2:30 to 3:20
Ben and Matthew tackled the bug in memory, so we waited on integration. I added reset to sp and tested it. I pushed the finished sp_block and the coprocessor that I had begun.

Tuesday
8:30 to 9:00
I finished the parts of the coprocessor I could without consulting my group. I messaged Ben and asked if he could meet me before class today to work on integrating memory.
2:30 to 3:20
Memory still wasn't ready, so Matthew and I discussed I/O. Maura wasn't feeling well, but Matthew gave the impression that everything was still on track. For the most part, we were on the same page. We planned to do something like memory-mapped, then put in the coprocessor.

Wednesday
10:00 to 12:35
Ben was late, so we started 15 minutes late. We updated our individual components to get them in shape for integration. When I was done, I started integrating, while Ben pusued a few questions he and Matthew had had about the memory datapath last night. I wrote tests and found that pc was not working.

(This whole milestone, I've felt very nervous about our status, especially because a lot of work was happening late at night after I was asleep. On multiple occasions, I asked my teammates to reach out to me if they wanted help. They usually responded by saying (I thought) they were about just to finish their part of the integration. Often, they said "I'll finish tonight." I noticed that things like memory took much longer than they said they would, but I didn't feel comfortable questioning their judgement, even though it kept me from working on my part of integration.

Some examples: (1) I told Maura yesterday to reach out to me because she seemed to have a lot on her plate. She messaged last night asking if she'd be able to get help at some point in the future. I told her "definitely," but she hasn't elaborated and I don't know her status. We haven't integrated it. (2) Last night, Matthew started integrating pc, sp, and memory without telling me and found a problem with pc_block. (3) I felt that, since I didn't know about so much work going on, I had perhaps ended up being negligent. I asked Ben about it and he said I was fine, because he "did a lot of less important stuff" and put things off to work late at night.)

2:30 to 4:10
I looked at the problem with pc: it was not getting written to. I found that Matthew had started integration without telling me and so I asked him to help. We ran into trouble because files loaded from his external harddrive didn't end up in git. Ben continued working on memory. Matthew and I worked on his machine to fix the problem with pc. We ended up doing so by changing the register_component file, but we had trouble pushing. I started a redone version of sp,pc, and memory. Matthew said he would finish integration tonight. I said I wanted to keep going with pc, because it was my part in the project. I double-checked everything in the milestone and made sure everything we had was up to date in the repo.

Details about interrupts:
How do we want to enable writing to the coprocessor's cause reg? Can't we just always write overflow to it if there's overflow, interrupt to it if there's no overflow but interrupt? It will maybe sometimes say interrupt while tackling an overflow, but we only need to look at it at the beginning of an exception, and it should be accurate then.

To add interupts, add an interupt wire to the CPU. This wire feeds control. When this wire is a 1, the kernel gets the data from the 16-bit in bus. We need to make the transfer to the kernel happen. We also need to write the assembly code that is the kernel, but this is for a later milestone.