Thu mar 7, Sieberz, 9:30p-12a

Alex & Genevieve met to work on the paper design and to better clarify direction for the project. Dan couldn’t make it, so we’ll work more with him tomorrow! We made google docs (and filled them) to start organizing our notes for the paper design.

Fri, Mar 8, EV

11a-12:30p

Alex and Dan discussed the pipeline and worked on the paper design.

1:45 pm - 3:30

Genevieve and Alex continued work on the paper design, raising questions about the more subtle differences between our old datapath implementation and what we’ll need for out pipelined design.

4-5 pm

GAD all met to finalize paper design & get approval

Sun Mar 10 8-9 pm

Alex and Genevieve broke down design for ADD, AND, LDR, STR to establish which signals are set to what exactly and how they are set -- also started planning control word

Wed, Mar 13, 7p-12a (-1am Dan)

Determined design for BR. - all

**GIT**ting our design shared as a team online.

Designed Control ROM and wrote some great test code. - Genevieve

Started inputting paper pipeline into computer. - dan and alex

Thurs, Pi, 7p-1am (-2am Dan)

Dan and Alex nearly finished inputting the cpu.

Wed, Mar 27, 6:30p-11p

DAG all worked together on CP2 -- adapted the cache, memory block, and the two-port memory to accommodate 128b memory lines.

Thu, Mar 28

11a-12:30p

DAG all worked on CP2 -- commenting/understanding testcode, cleaning up the design, adding the load signals

7p-12a

DAG all worked on CP2 -- debugging memory, Mentor Graphics, and other issues.

Fri, Mar 29, 9:30a-12pm

Alex worked on adapting cache to support 570ns delay.

We all met in the afternoon, also met with V for handin.

Thursday April 4, 7-8:30p

Worked all together to understand what cp3 entails, started hammering out data forwarding ideas.

**Friday April 5, 3-4:30p**

Discussed plan for cp3 and met with V for questioning

Sun Apr 7, 3 - 9p

GImplemented LEA & SHF

GPartially implemented STB, LDB (control word changes, added double byte component to WB) -- need to add write\_h/l logic, need to add zext for LDB

-- may need to change control word -- technically, write\_h/l is only important for stb, everything else will set both low

-- worked on fixing a branch bug

Mon, Apr 8, 9p-12a

Alex worked on fixing the branch bug. Fixed it. Made another bug in doing so. Need to fix that now.

Wed, Apr 10, 10:30-1am

Alex worked on fixing the new branch bug. By the time he leaves, he’ll have nailed it.

CAUSE HE DA BEASTEST

Genevieve “fixed” stb, but we’ll see.

Indeed as was fortold, Alexander fixed the terrible, horrible, no good, very bad memory reads with branching bug. Also made some handy components for clipping to the clk inverse and a DFF word latch.

Thurs, Apr 11, 8p - 3/5am

Everyone fixed bugs. Implemented some instructions. Made the initial design for forwarding and the arbiter. Wrestled with Git and with Mentor Graphic’s impeccable ability to fail in every fiber of its being.

Friday, Apr 12, 11am - 5p

Everyone fixed bugs, finished instructions, and made some stuff work. Put arbiter into the CPU.

**Thursday, Apr 18**

**5-6pm**

Genevieve fixed STI and made other cool changes to fix cp3.

We also had a meeting with Victoria.

7:30-10:30pm

Alex made the detailed paper design for 8-way pseudo-LRU for L2 cache. He also designed a 32-bit synchronized counter for use in performance counters.

**Friday, Apr 19 2p-6pm**

Met with Victoria. Talked about BTB/BHT stuff.

Genevieve fixed JK flip flop and signal problems that might for real fix STI.

Saturday, George Takei’s birthday, 1p-7p, 8:30-11:59:59p

Alex and Genevieve worked on a lot of data forwarding stuff. Made most of it work. Branch flushing seems to work. need to forward from WB to MEM.

Sunday, April 21, 1:30 - 11:30p

Genevieve and Dan worked on a lot of various load/store issues, some jsr issues --

Found a bug in forwarding:

When you have two LDRs then an ADD using the two results, we stall and wait for the second operand to leave the mem stage and go into the WB stage. We are able to get that second operand from the WB stage, but the first operand (which was in the WB stage before the stall finished) is long gone by the time we are ready to use it.

Alex jumped in and helped fixed it with Dan. MP3-cp3 works perfectly. Started inputting paper design (completed) for L2.

Next steps: forward from WB to mem. Make L2. Make BTB + BHT.

Monday, April 22, 8- ...

Alex and Genevieve enhanced data forwarding by adding it to the mem stage and fixing R7 forwarding for special instructions.

**Tuesday, April 23, 4:50p - 1:30a**

MEET WITH V

Made most of the datapath for the L2 and the BTB.

Wednesday April 24 7p - 10p

Genevieve put BTB/BHT signals and logic into MEM

Still need clipped clock for BTB and to actually drop BTB into IF

committed and pushed changes

11p-4a

Alex and Dan worked on the L2 and fixing bugs in the competition code. Could not find the point in competition code where we start being wrong (correct up to about 3,000,000 ns?). Alex finished L2, but it’s a bit buggy. Competition code breaks earlier with it. Its writeback isn’t working properly???

THursday 1p - 6am fri? Everyone

**MET WITH V -- Dan & Alex (& G via speakerphone!)**

LDI bug, trying to make final run w/o btb

Unfortunately, me sitting in front of the computer another two hours led only to frustration and confusion: if.taken goes high occassionally -- i hate that misnomer with every fiber in my being, so i took the three seconds to change it to ReBranch. ReBranch fluctuates because sometimes memstage decides the predicted pc doesn’t match the branch addr because the predicted pc sometimes goes to zero (because the ReBranch sometimes goes high... see my frustration? haven’t figured out what starts the circle).

I’m changing predicted pc to p\_target just to see.... it may end some frustration.

That didn’t work. It made our first branch loop over and over again. I sense there may be an issue with the generated NOPs of IF -- those go in like an instruction, so they don’t look like generated NOPs... problem???

Another Q: reset value for BHT is 10 for everything... so why does it not initialize to that?

Why does PC = 10 propogate while it is still waiting for instructions to return from ICache? (mp1fivefactorial)

Leaving much too late. planning to be up around 10? to completely restart paper design.

April 26, Friday 10am-4:20p Everyone

Sped up L1 cache reads to shorten BTB critical path on TRAP to fit within 50ns. Made all test code other than final and competition code work with BTB (and still with L2). Found where issues originate in BTB. Did not get to fix it or do performance counters. But we’re done! :D