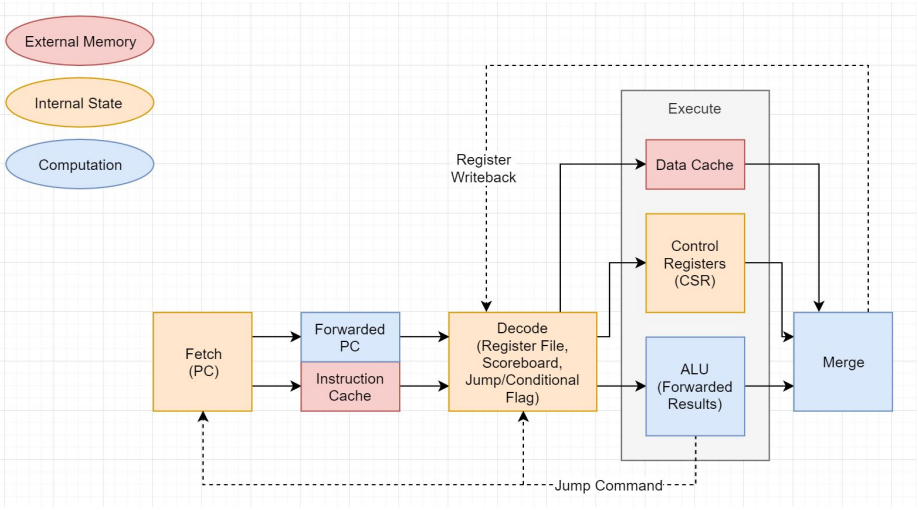
Project Status Report #1
David Gronlund
Group A3
2/16/19

This week I worked on building out the infrastructure for our vector RISC-V core as well as helping write some libraries for verifying our floating point implementation. I came up with a general architecture for our core, specifically one that allows us to do a single register write per clock cycle while still allowing for the ability to issue instructions to different execution units even if another execution unit is busy, for instance a floating point register operation followed by a memory access.



The floating point library I wrote was an extension of the one that Cyril started in Python, to run floating point operations alongside our verilog to make sure we generated the correct result. I continued this approach but in C, so that we could choose different rounding modes. The C code also lended itself easier to bitwise manipulations and we can invoke it in some verilog testbenches as well to streamline our verification process.

The project is currently on schedule, barring something catastrophic happening with the vector compiler. We are making good headway on implementing our floating point algorithms, with multiplication, addition, division, and square root all working in verilog, now we just have to pipeline them and map them to FPGA resources.

I am hoping to have the RISC-V integer core implemented by the end of next week, as well as having the C verification floating point verification program finished enough so that it could be invoked from a testbench to check the results of the testbench. Hopefully in two weeks I can then have the core passing the software tests so that we can work on integrating the floating point unit into it.