Project Status Report #10
David Gronlund
Group A3
4/27/19

This week I worked on getting the infrastructure working to test out the core on the Ultra96 board as well as working on the design of the vector floating point unit.

The bulk of my efforts were in trying to get PetaLinux working on the Ultra96 board, which ultimately became a joint effort with another capstone group who was using the same board. The issue came down to trying to use an ILA over JTAG when trying to also run Linux on the board at the same time. Otherwise getting a PYNQ image on the board was pretty easy since the correct image could just be downloaded from Avnet. However when using this default image, as soon as JTAG connected to the FPGA, the whole Linux image hung and never recovered. I went in over XSDB (another Xilinx tool for looking at the internal debugging logic) and found that all the processors had faulted and returned to their reset vectors. This was a little confusing since the JTAG should just be an observer at this point, but I had some suspicions about the DAP (the internal debugging logic of the ARM processor) being driven to execute some erroneous operations but Xilinx Hardware Server. My suspicions were confirmed by some forum posts about how JTAG would cause the ARM cores to fault if they happened to be powered down while in their idle state to save power.

Those same forum posts suggested the fix was to change a PetaLinux setting and rebuild the entire image, which kicked off the next four days of first using the wrong version of Ubuntu (18.04 is evil), then building in a virtual machine (finished after more than 12 hours), and the issue still remained. I thought another setting change could potentially disable CPU idle, so I reimaged a machine with Ubuntu 16.04 and while the rebuild was quicker this time (only 2 hours), it still did not work. At this point I was prepared to give up on Linux and the ILA at the same time, but I found an obscure forum post (it was not even a Xilinx forum) that discussed how CPU idle modes could be disabled in Linux at runtime. By running four echo commands (one for each core) I was finally able to prevent the CPU's from entering idle and the ILA cores were now accessible with Linux running at the same time. Honestly the only major lesson I got out of that experience was to just not trust Ubuntu 18.04.

With what time I had to spend while the images were building, I worked on integrating the floating point logic I had pipelined earlier into a single module, and then writing the logic to do conversions, memory accesses, and potentially striped writes to the register file due to how the vector entries get stored.

```
#!/bin/sh -e
echo 1 > /sys/devices/system/cpu/cpu0/cpuidle/state1/disable
echo 1 > /sys/devices/system/cpu/cpu1/cpuidle/state1/disable
echo 1 > /sys/devices/system/cpu/cpu2/cpuidle/state1/disable
echo 1 > /sys/devices/system/cpu/cpu3/cpuidle/state1/disable
exit 0
```

(The single most consequential shell script I have ever written)