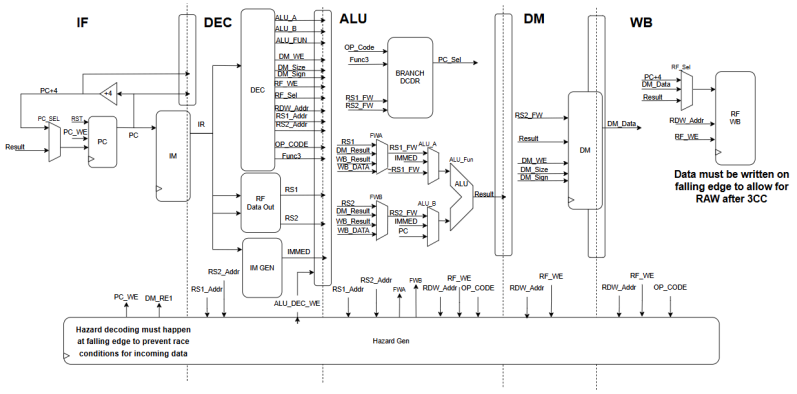
CPE 333 Lab 3: Pipeline with Hazard Handling

Contributions:

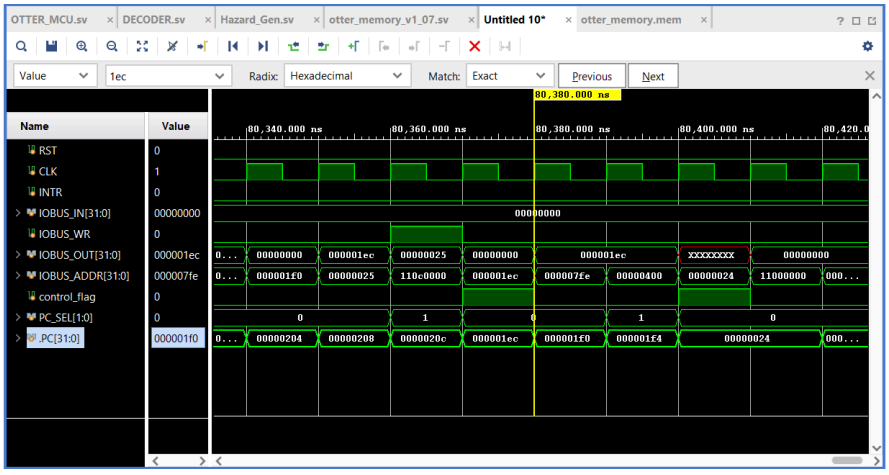
- Wyatt Tack: Wrote new verilog code, designed hazard generator
- Justin Rosu: Helped design data hazard block in hazard control
- Brayden Daly: Helped debug memory module
- Tyler Hamilton: Helped design/implement control hazards portion of hazard gen

Design:

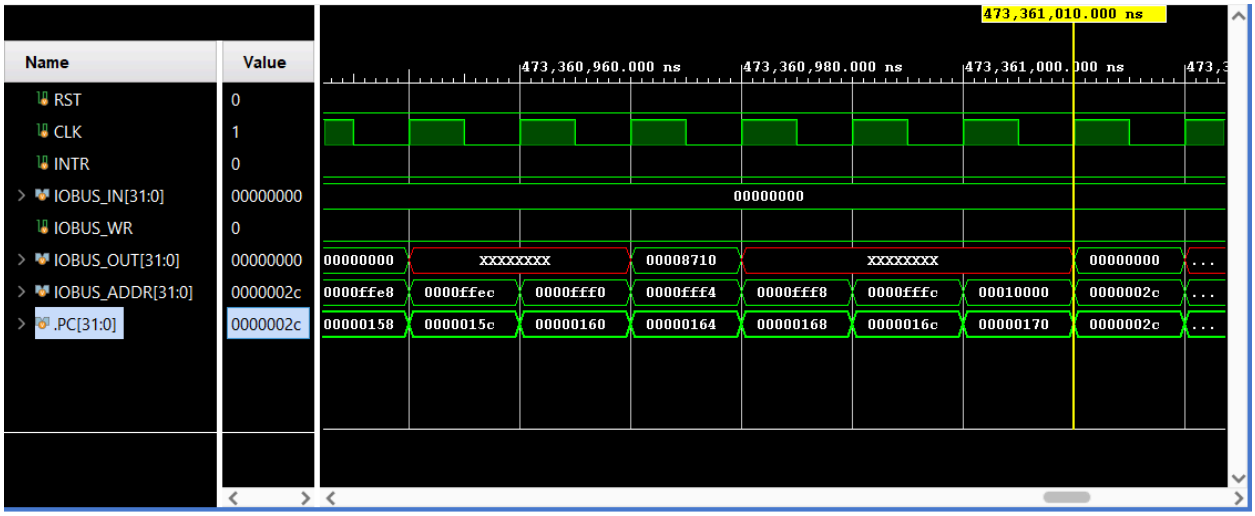


Our design built off the first implementation of the pipeline, with the main addition being the hazard generator, which despite its name is meant to solve both data and control hazards. Data hazards are monitored through a constant view of register addresses at each sequential step, while control hazards are monitored through the watch of the PC select mux.

Vivado Simulation:



Above shown is the simulation for TEST_ALL, the benchmark verification of the RISC-V CPU family. The multicycle CPU design previously reported on finished this program in 32,979 clock cycles, as our improved design finished it in a mere **8,038 clock cycles**, over 4 times as fast as previous versions.

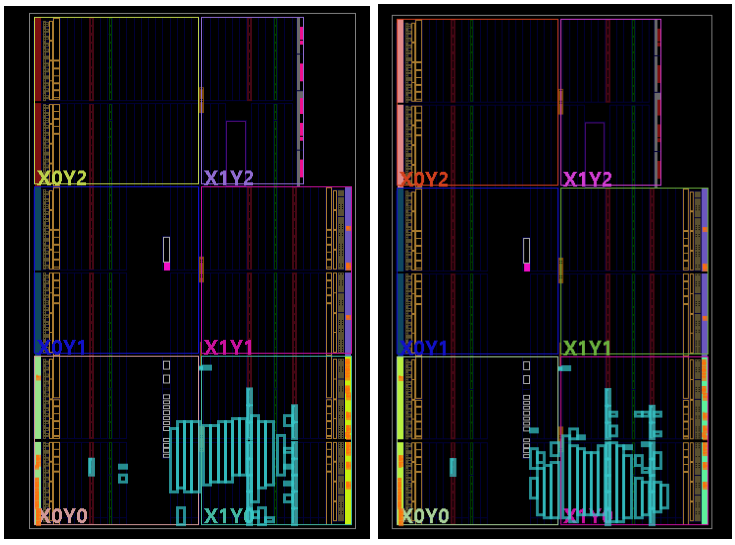


<u>50x50MM: 47,336,101 CCs</u> (from beginning of instruction 0x02c: "j end")

Further shown is a screenshot of the pipelined otter executing the 50x50 matrix multiplication algorithm demonstrated in lab 1, in which the multi cycle CPU took 64,748,903 clock cycles to complete, as which the sped up pipelined processor only took 47,336,101 clock cycles.

<u>Area and Power:</u>

The multi-cycle CPU was, in its design report, estimated to take 0.102W of power on average, with a 39% dynamic and 61% static distribution. In devices, 7% of LUTs, 1% of FF, and 44% I/O was used on the old design. Our new design was verified with a lower power rating of 0.095W with 34% and 66% dynamic/static spread. Our design increased usage, with 8% of LUTs, yet only 1% of FF, and 43% I/O. Below is the implementation of the old CPU(left) and new CPU(right):



<u>Multi-Cycle (right) and Pipelined (left) CPUs</u>