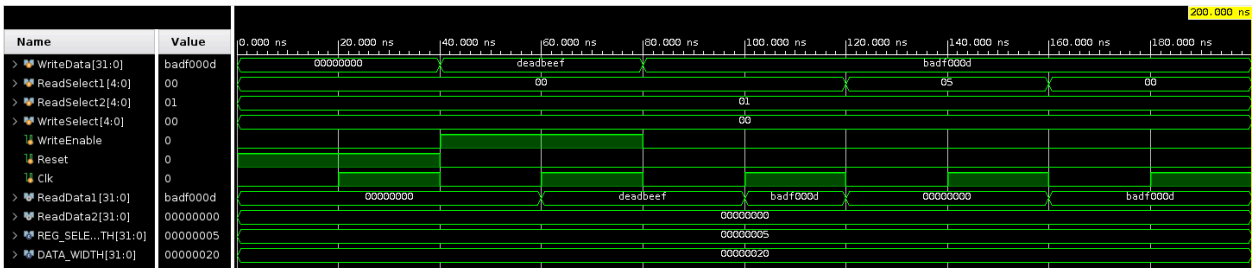
Leonardo Mattos Martins
U25267206
Lab 5 Write Up

The purpose of this lab was to build, debug, and test a 3-stage 32-bit pipelined datapath with a hierarchical design. The datapath consisted of: 3 stages, an ALU, a mux, and a register file. For a given instruction *InstrIn*, the first stage would decode the instruction and select the appropriate registers to use from the register file. The second stage would receive the value stored in the aforementioned registers and pass them to the ALU, where the ALU executes the operation described in the instruction. The final stage saves the result of the operation into the register file and outputs to *out*. Each stage takes a clock cycle to execute, taking a total of 3 clock cycles for an output. Due to the design of the pipeline, each subsequent instruction will be output 1 clock cycle after the other.

In this lab report, I will describe how the datapath (register file, stages, ALU) were implemented and tested.

Register File (register_file.v)
The register file was implemented using nbit_register_file.v from Prelab5 as a guide. It was tested in the register_file_tb.v testbench, which uses similar logic to nbit_register_file_test.v from the provided Prelab5 folder, where a range of values are written to and read from the register file. The waveform corresponding to this can be found below.



Stages (S1_Register.v, S2_Register.v, S3_Register.v, pipeline.v)
pipeline.v is the top/main file responsible for pipelined datapath, and it is tested using the pipeline_tb.v testbench. Pipeline_tb.v tests a set of different ALU operations on different registers for both I-type and R-type instructions. In this testbench values for registers and called more than once to ensure that the result was saved in the register file. The timing diagrams which correspond to the sample set of instructions provided in Pipeline_test.v in the Lab5 folder.

ALU (nbit_ALU.v)

The ALU was implemented behaviorally using ALU_behavioral.v from Lab4 as a guide. As it is implemented behaviorally and ALU_behavioral.v was used as verification logic in the previous lab, there was no need to test it.