Section 1: Team

at_your_own_risc
Patrick De Leo 217220203
Arnav Gupta 219973452

Section 2: Verilator Waveform

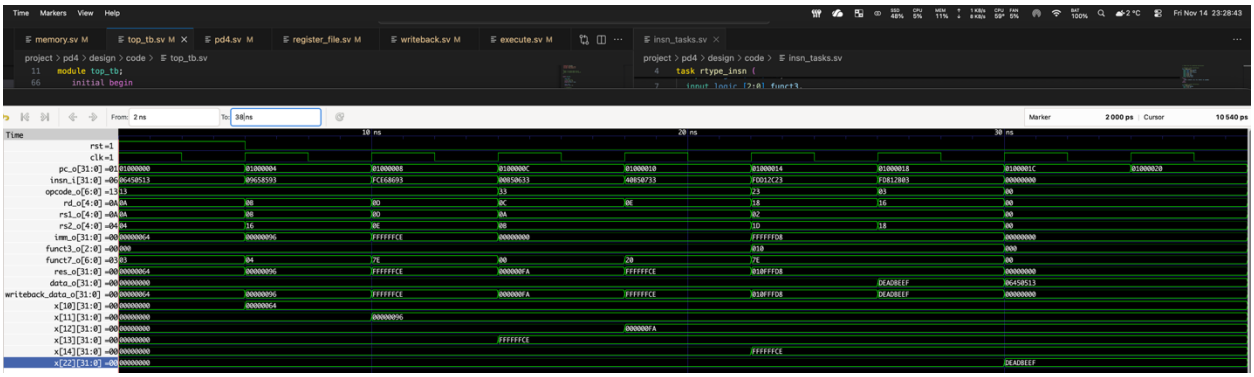

Figure 1 - Our own testbench showing the register file contents after each instruction

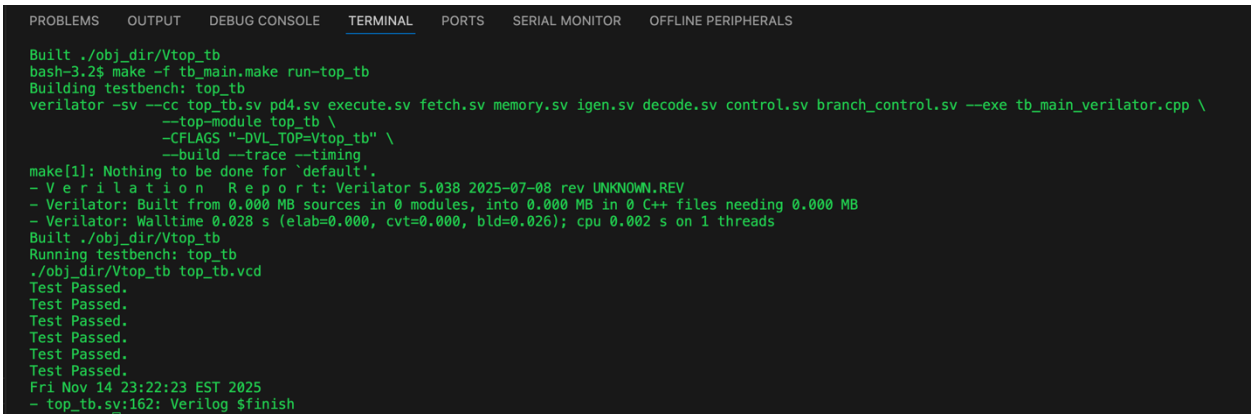

Figure 2 - Showing our test cases pass

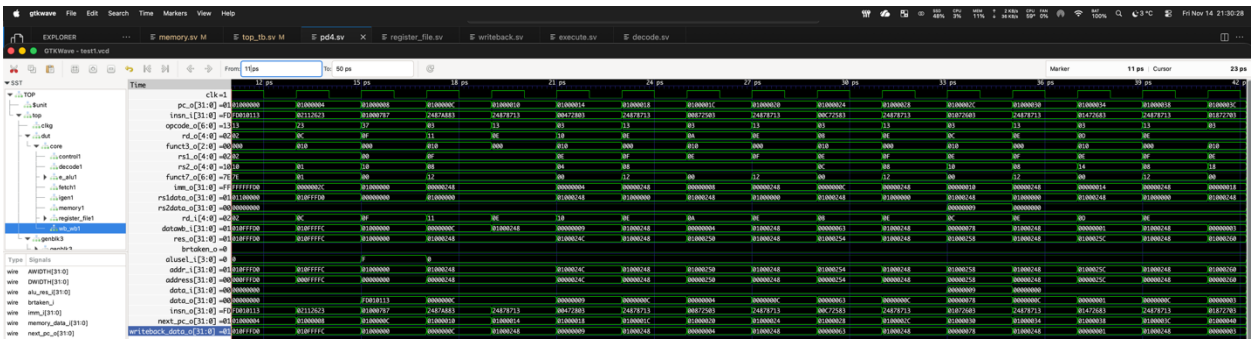

Figure 3 - TEST1 waveform showing the first 16 instructions

Section 3: Reflection

There were a few challenges during this project deliverable:

- Determining size_encoded signal value
- Memory reads during non-memory access instructions
- Data memory conditions

For the size_encoded signal value, we suspected this would be associated with the funct3 value of the memory access instruction corresponding to byte, half-word, and word. Given that 0x0 and 0x4 (byte encoding) and 0x1 and 0x5 (half word encoding) have the same 2-bit LSBs, we could bit extract 2-bit LSBs from the memory access instructions. The difficulty was in determining that the lower 2-bit LSBs of non-memory access instructions was used for the size_encoding signals for all other instructions. Thanks to some help from classmates, we were pointed in the right direction.

For memory read during non-memory access instructions, we had assumed that there would be no memory reading while this was occurring. However, the PD4 test cases showed this was not the case. This required changing the funct3 inputted into the memory module so that non-memory access instructions would still read a word. This is commented in PD4 “memory stage.”

After many days of failure reading from memory, and what felt like arbitrary failures, we decided to step through our memory module code. In doing so, we decided to remove any conditions on the address for data memory reads to see if this would resolve our “M Stage” mismatches. Although this did work, we are a little confused as to why it did not with the condition, given it is rather benign. The condition was:

else if (address + 32’d1 < BASE_ADDR + MEM_BYTES)

// Allow memory read

Which we expect to mean if the address provided is the last accessible byte allow reading data, if the address provided is above the physical memory space, do not read. It turns out that this rather benign condition was the source of most of our M Stage mismatches.