Section 1: Team

at_your_own_risc
Patrick De Leo 217220203
Arnav Gupta 219973452

Section 2: Verilator Waveform

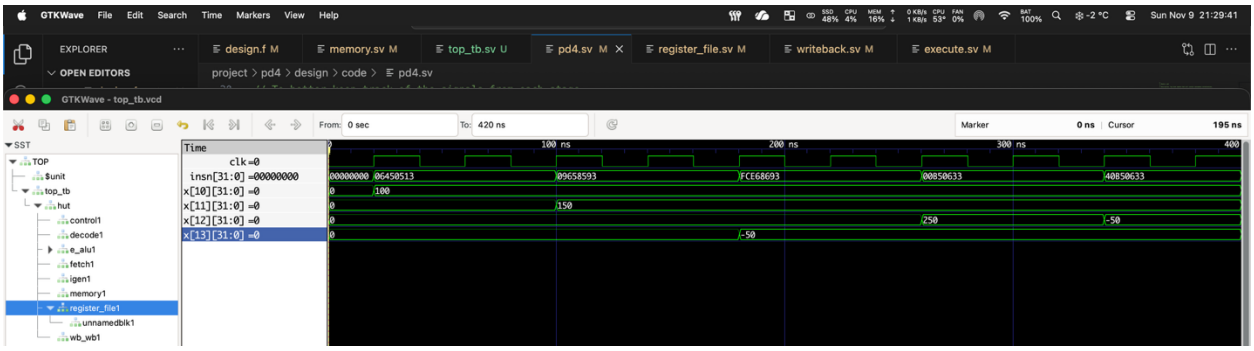

Figure 1 - Our own testbench showing the register file contents after each instruction

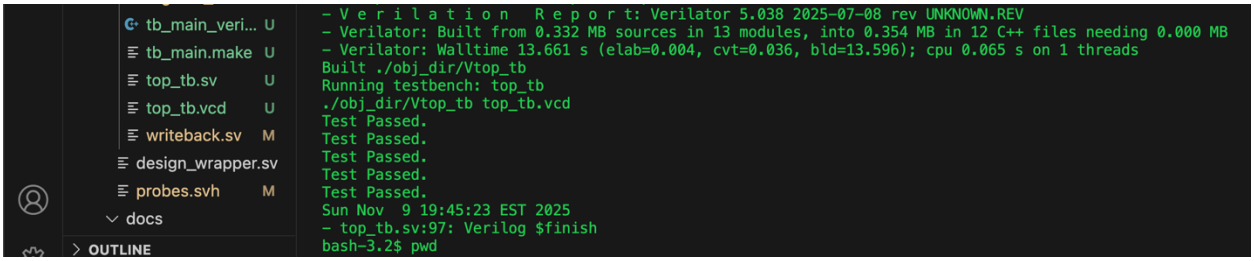

Figure 2 - Showing our test cases pass

Section 3: Reflection

There were a few challenges during this project deliverable:

- Determining size_encoded signal value
- Memory reads during non-memory access instructions
- Masking the memory address to handle “wrapping around”
- Handling non-word aligned wrap around memory accesses

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 signal 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.”

Handling memory accesses at addresses above the physical memory space was the most challenging part, particularly for non-word aligned accesses. The initial mask for word-aligned wrap around was simply, mask lower bits the fall within the maximum address available. So, for a maximum address of 0x01100000 which is offset to 0x00100000 we need to mask the lower bits which would be 0x000fffff. This can be interpreted as maximum byte - 1.
As of writing this, we were unable to figure out how to handle non-word aligned wrap around memory access. We tried masking each address of the little endian word assembly but this yielded the same failures.