Section 1: Team

at_your_own_risc
Patrick De Leo 217220203
Arnav Gupta 219973452

Section 2: Verilator Waveform

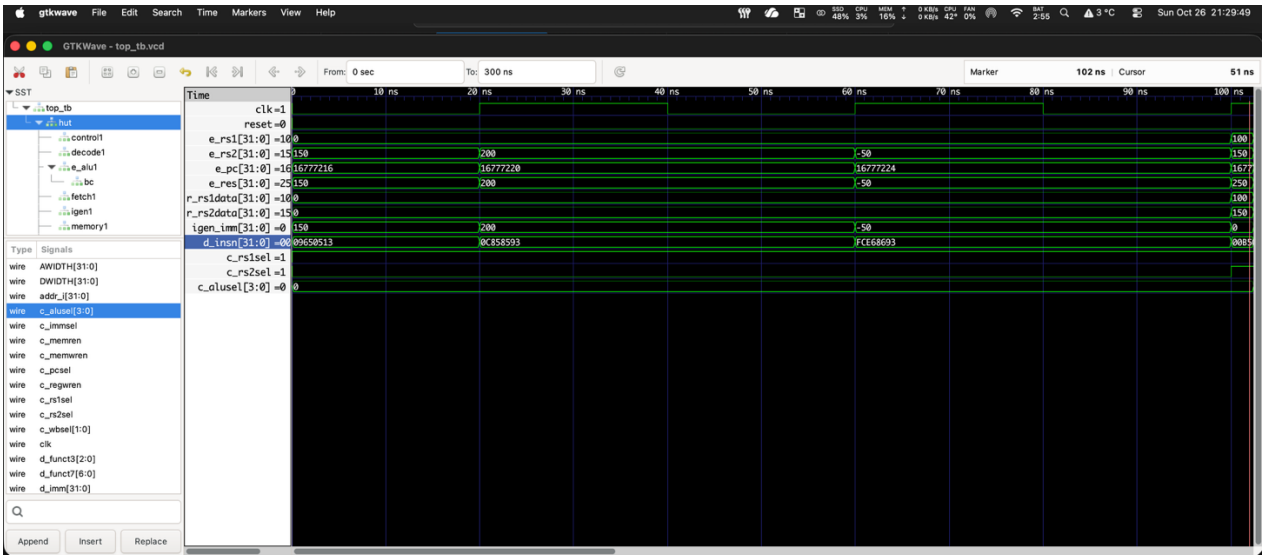

Figure 1 - Test 1, 2, and 3: addi x10, x10, 150, addi x11, x11, 200, addi x13, x13, -50

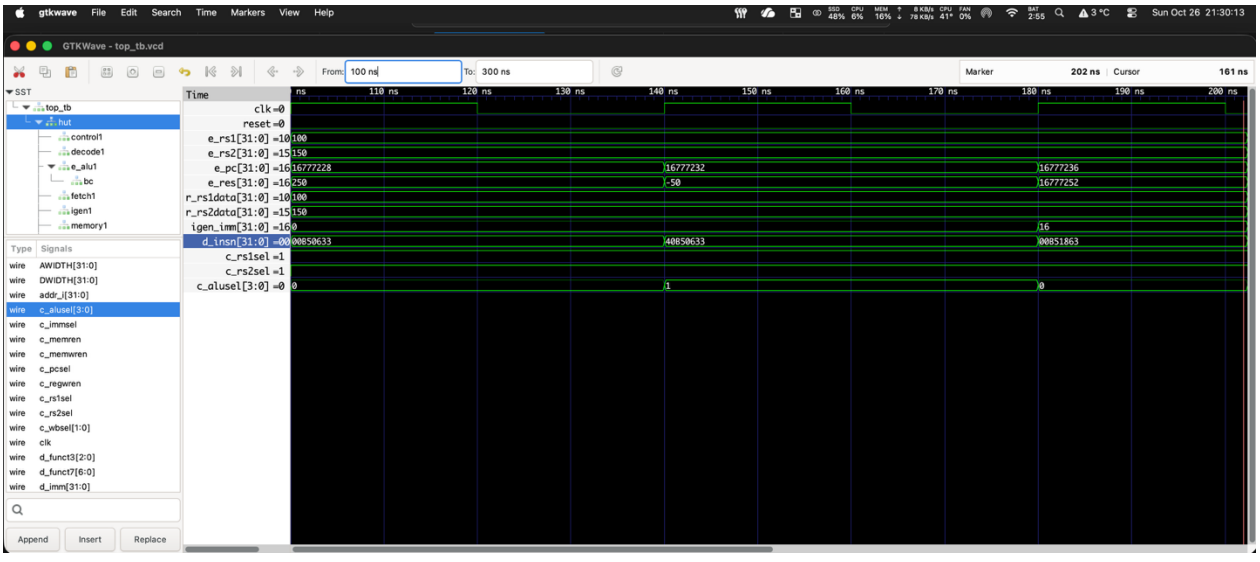

Figure 2 - Test 4, 5, and 6: add x12, x10, x11, sub x12, x11, x10, bne x10, x11, 16 Note: x10 = 100 and x11 = 150

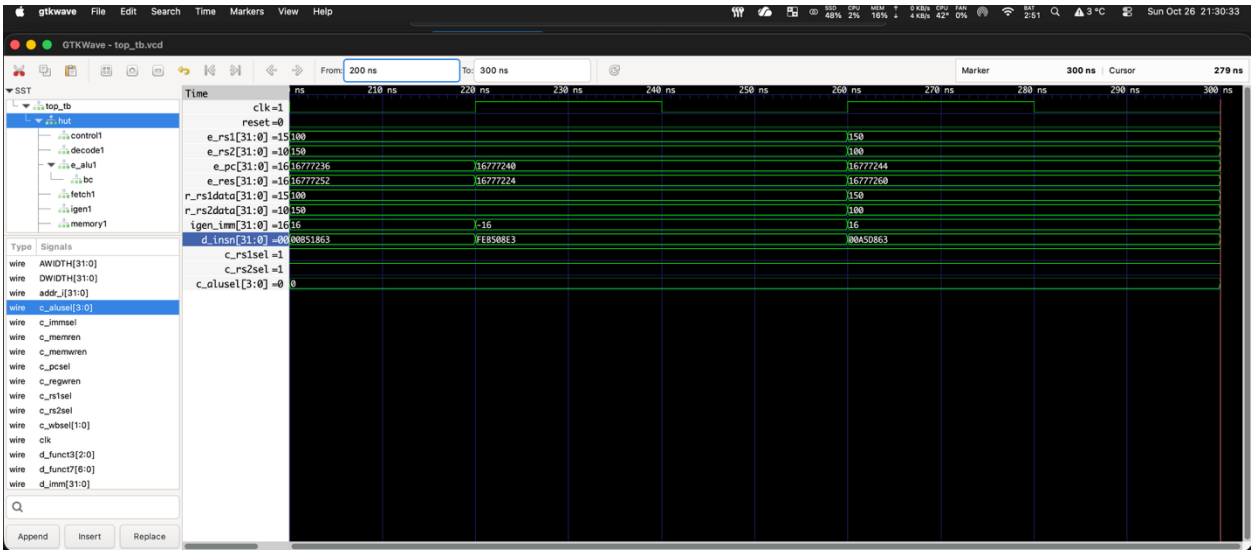

Figure 3 - Test 7 and 8: beq x10, x11, -16, bge x11, x10, 16 Note: x10 = 100 and x11 = 150

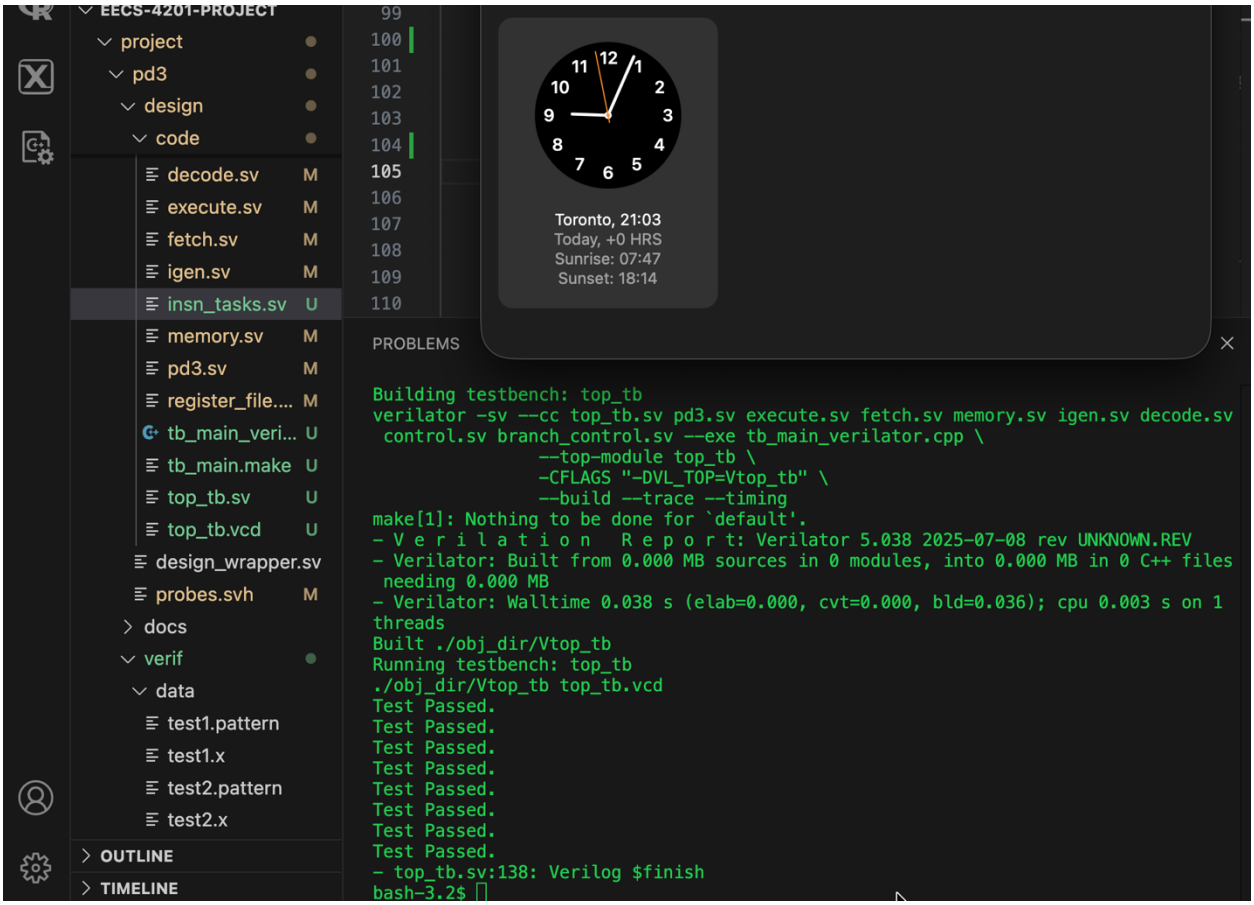

Figure 4 - Image showing all tests have passed.

Section 3: Reflection

There were a few challenges during this project deliverable:

The first was determining how to properly wire execute and the branch control. Relying on only the input signals provided, we were forced to instantiate branch control inside the pd3.sv because of the four operands necessary for a branch control instruction. Since the Asel mux and Bsel mux control the program counter (pc) or source operand 1 (rs1) and the source operand 2 (rs2) or the immediate value (imm), there was no way to handle branch instruction from within the execute model. However, from the TA's comments on Discord, we figured we were allowed to make necessary changes so long as we commented as to why. We added inputs for the opcode, alu select, and immediate value to the execute.sv file. This allowed us to do branch comparison with rs1 and rs2, while using the alu to compute the branch destination (pc + offset).

The second was how to instantiate the stack pointer register (x2 or sp). Calculating this value was simple, we checked the Makefile for MEM_DEPTH, converted it to a hexadecimal and added it to the BASE_ADDR. However, because of the reset signal at time 0, which required close analysis of the waveform, this caused our register file design to reset immediately, forcing the stack pointer to no longer contain the desired address (0x01100000). Therefore, we had to change our design to instantiate our desired values for the register file at reset.

After this, each error was simple to debug and only required changing the control signals within control.sv.

The only other issue we encountered was handling of branch instructions in our own test bench. Due to how we handle splitting the immediate value and how immediate values are handled in branch instructions; we were off by a factor of 4 on our branch offset calculations. We determined that this was doubling the bit shifting done by the immediate generator and therefore had to right shift the immediate value by 1 in our instruction task file. This allowed for our test bench to pass perfectly. To debug this required many hours because at times we felt that our immediate generator was the culprit and not our actual test bench.