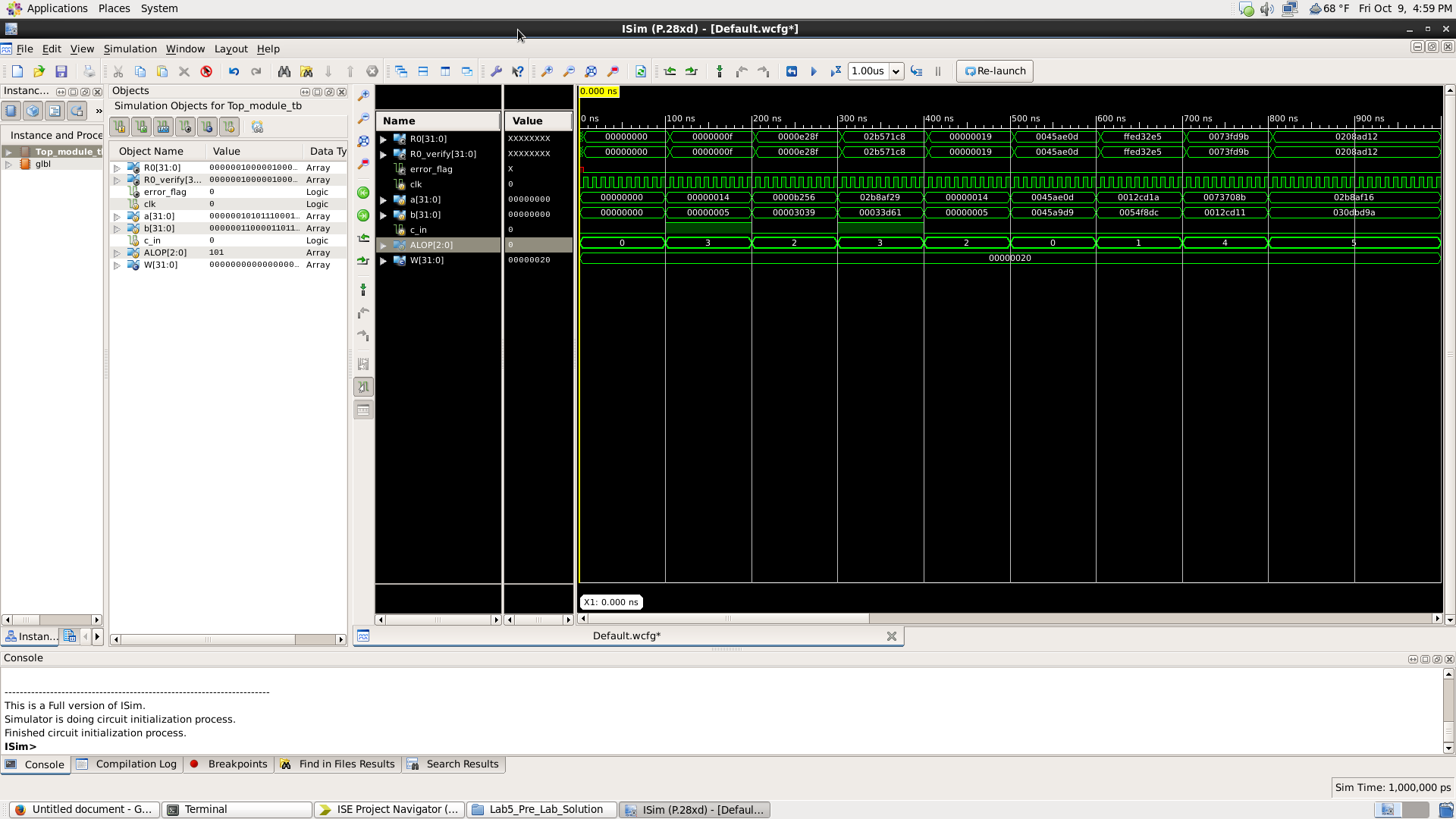
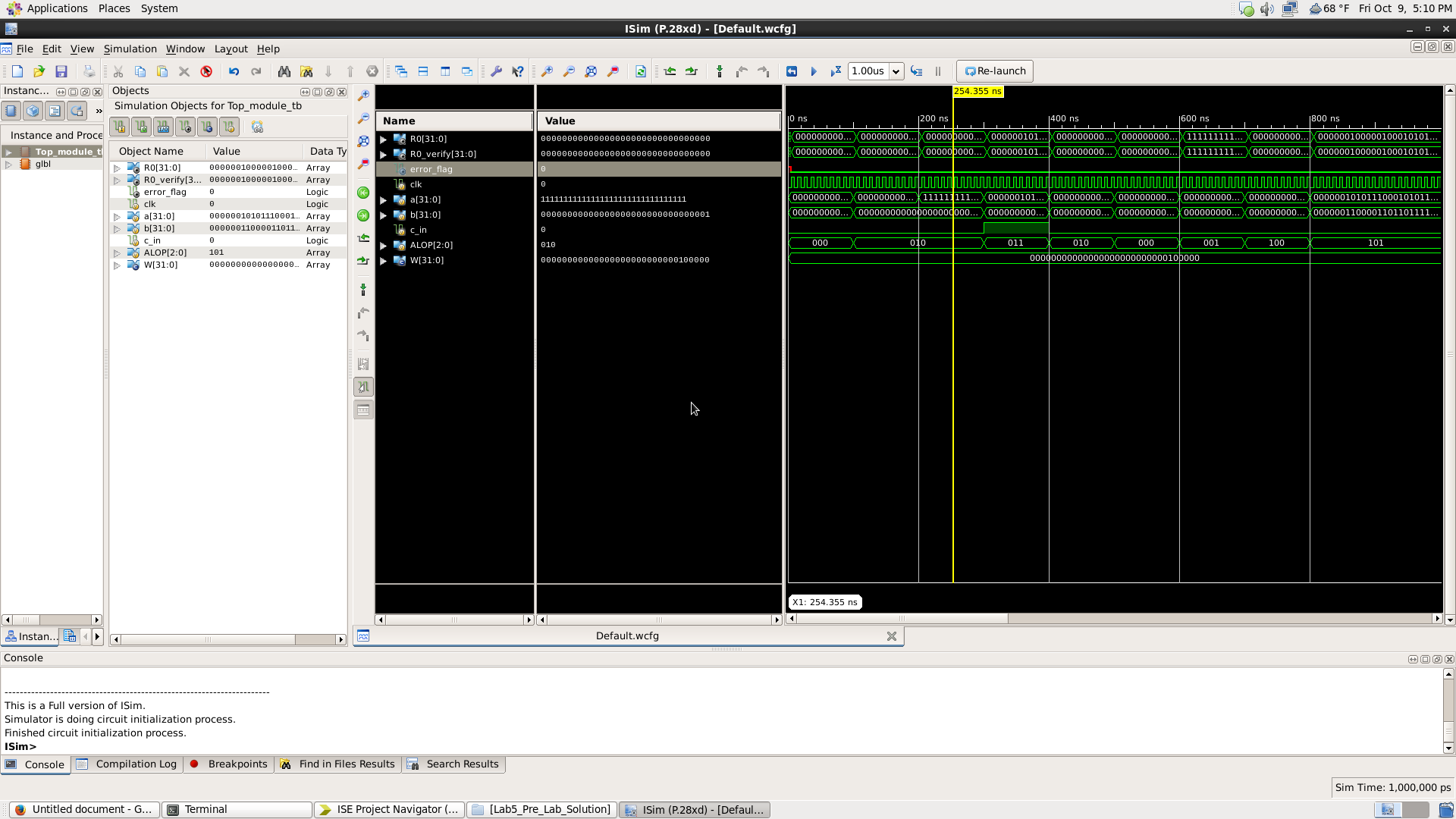
Computer Organization Lab 5 Write up

In my project I had 6 main modules, that included other submodules as well. My top module was called the Top\_module. This had my nbit\_alu, the Register the nbit\_alu outputs to, Verification for the n\_bit, and the register that takes in the verification output. The nbit\_alu parameterized the onebit\_alu W amount of times. The verification module compared the values in the registers to make sure it was correct. In the Onebit\_alu, it contained the mov, not, add, sub, or, and the AND modules. It also called the mux which was able to select which of the outputs to use. I also had an add\_mux which would give an output of c\_out from the adder/subtractor, only if the adder was called.

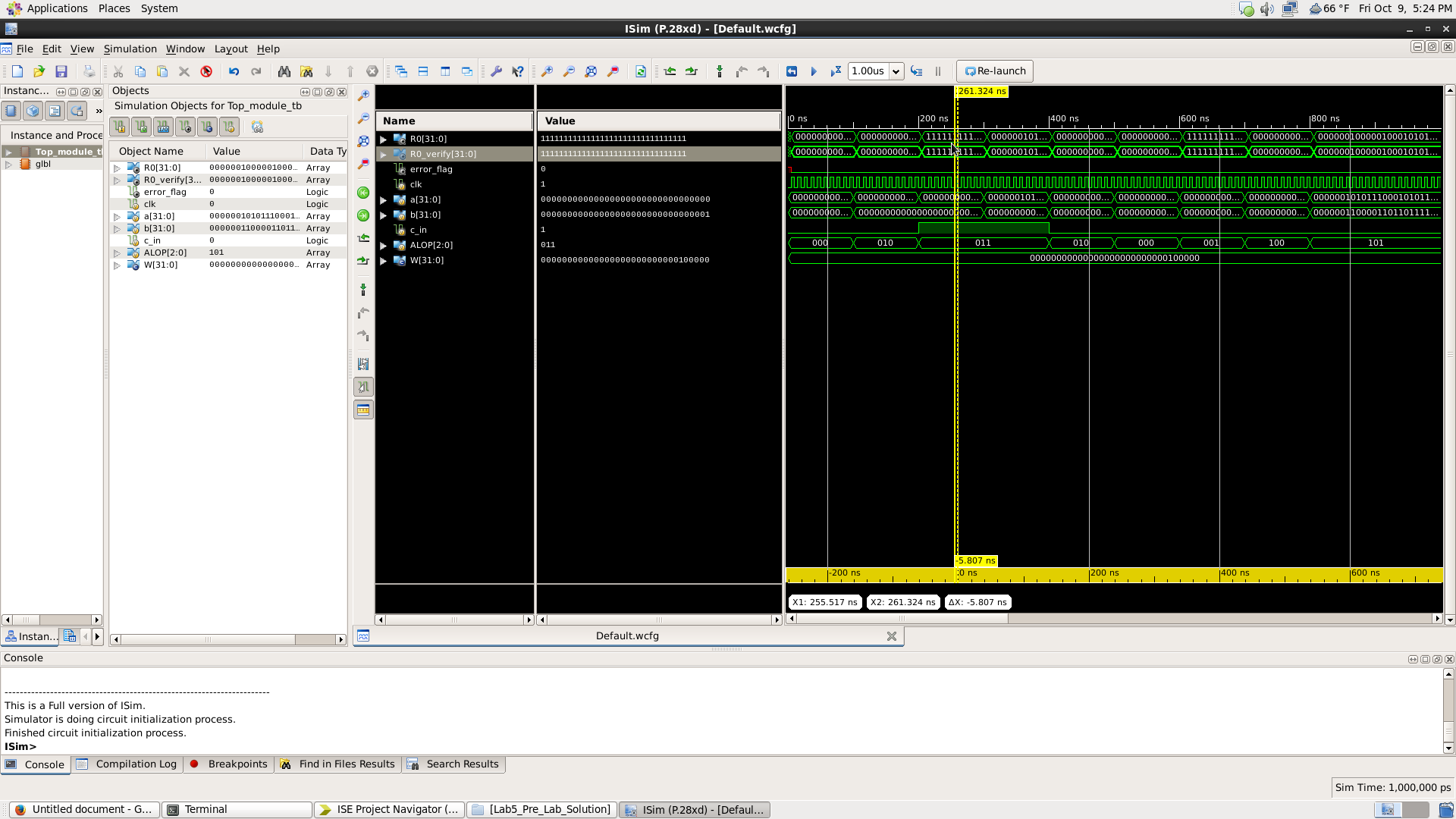
The Hierarchy I had for the ALU was Top\_module, ALU\_nbit, alu\_onebit, the muxes, and the adder, subtractor, mov, AND, and or modules.



This waveform contains the test cases for all the ALOP codes from 0-5. These cases had values ranging from 20 and 5 to values such as 45657897 and 212321.



This waveform shows the case where 32’hffff\_ffff was added to 1. As you can see the result is correct, which is verified by the error flag, which is 0.



This waveform shows what happens when you do 0 - 1. As you can see from the picture it becomes 32’hffff\_ffff.