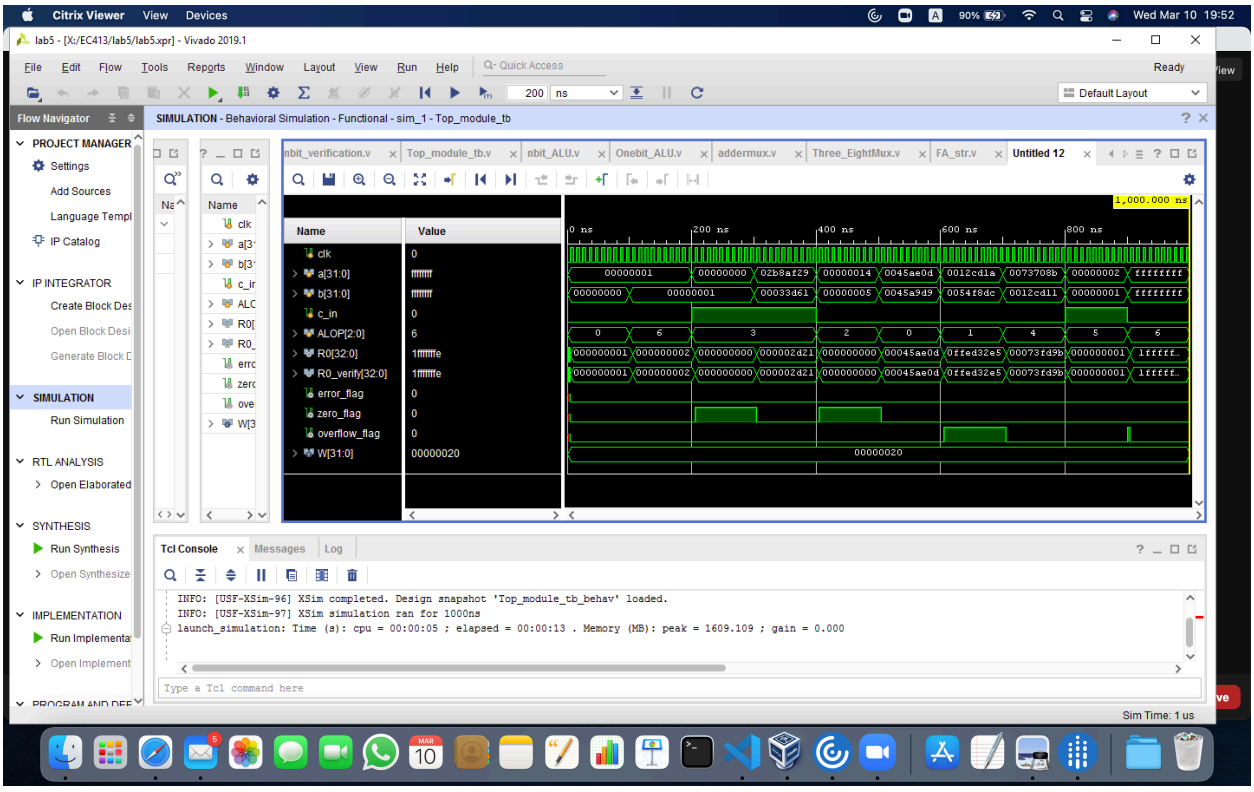
12/3/21
EC413
Ivan Isakov

Lab5

In my project I had 6 modules and other submodules. I had a Top_Module, which consisted of an nbit_ALU, a Register that received the output of the nbit_ALU, verification logic for the nbit_ALU and a Register that received the output of the verification logic. The nbit_ALU parametrised the Onebit_ALU a certain number of times, specified by the parameter W. The verification logic compared the values of the registers and raised an error flag if there was an issue. The Onebit_ALU consisted of: mov, not, add, sub, or, and modules. It called a MUX to decide which outputs to use, depending on the ALOP. It also had an addermux module to give an output of c_out from the adder/subtractor, if the adder was called.

The hierarchy went like this:

- Top_Module
- nbit_ALU
- Onebit_ALU
- Adder
- Subtractor
- Mov
- Not
- And
- Or
- MUXes



This waveform contains the test cases for all ALOP codes from 0-5. These cases had values in big and small ranges.

The error flag is down all the time which means the ALU has the same answer as the verification logic.

The zero_flag is up when the answer is 0.

The c_out is stored in the registers as a MSB.