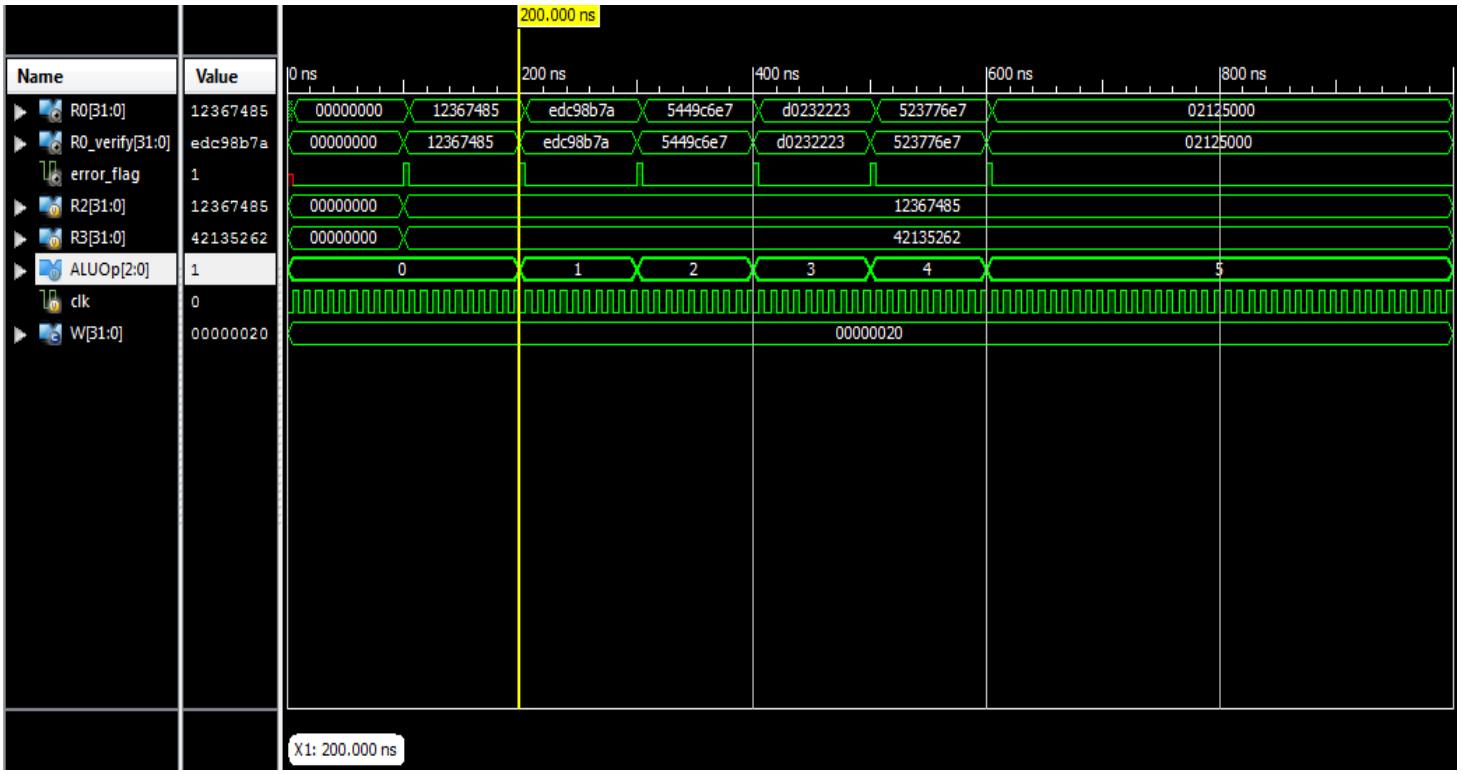
Jackson Hsu
Lab 5
10/9/15

Simulation Results and Waveforms



Above is a simulation and waveform of my TOP_MODULE.

I made individual modules for the 5 of the 6 operations of the ALU. The 5 operations being NOT, ADD, SUB, AND, and OR. The 6th operation, MOV, did not require an individual module as it could just be assigned from one to the other within the ALU module. I merely assigned output = input for the MOV module while keeping their bit sizes the same. The ALU executes the MOV operation at value of 3'b000 (0) and assigns R2 to R0.

For my NOT module, executed at ALUOp = 3'b001, (1) I used a for loop to loop through however many bits the inputs and outputs would have, controlled by parameter W.

For my AND and OR modules, executed at ALUOp = 3'b101 (5) and ALUOp = 3'b100 (4) respectively, I did the same as my NOT module (using a for loop), with their respective functions (and and or).

For my ADD and SUB modules, executed at ALUOp = 3'b010 (2) and ALUOp = 3'b011 (3) respectively, I also used a for loop. However the operation in the for loop was a full adder used in previous labs. The difference between the ADD and SUB operation was that while the ADD

operation took in R2, R3, and 1'b0, the SUB operation took in R2, R3_not, and 1'b1 (two's complement).

The Register module was the one used in PreLab5.

The ALU called the operation modules previously described and assigned them to output values based on the value of ALUOp in the TOP_MODULE.

The TOP_MODULE connects the ALU to the parameterized Register.

The hierarchy of my design was I created individual modules for almost all the ALU operations and connected them to the ALU. Then in my TOP_MODULE I connected the ALU and the register for the final output.