ECE 4435 Learning Activity 7 Report

Spencer Hernandez - **ECE 4435** - April 28, 2023

I. Intro

The purpose of this lab was to add the implementation for an XOR operation to our RISC processor.

II. Verification

The screenshot below in Figures 1, 2, & 3 are why I believe my design meets the necessary requirements.

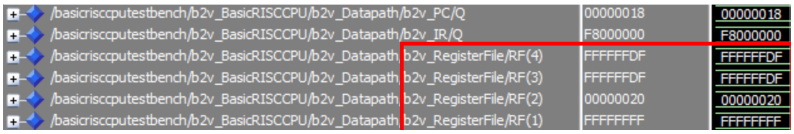

Fig. 1. CPU Verification

This verification was generated by altering the java simulator machine code for the add.asm file, which simulates the and operation being executed instead of the add operation. Because the and instruction and the xor instruction are very similar in how they are each called, altering the machine code that is generated from the and operation allows us to properly test the xor operation. In Figure 1, -1 is stored in RF(1) and 32 is stored in RF(2), and the result from RF(1) xor RF(2) is stored in RF(3) and RF(4).

Fig. 2. ALU Verification

Figure 2 verifies that the ALU is able to successfully perform the xor operation. When xor_op is high, C is loaded with A_Q xor B, which is -1 xor 32 (the same as in Figure 1).

```
VSIM 23> run 15000 ns
# ** Note: Number of errors = 0
#    Time: 12 us  Iteration: 0  Instance: /testbench
```

Fig. 3. Control Signals Logic Verification

Figure 3 verifies that the Control Signals for xor are properly generated. As one can see, there are no errors with the Control Signals after the xor_op verification was added to the test bench.