8-bit CPU in Verilog

Jake Johnson

June 28th, 2016

1 Abstract

In this project, I implemented an 8-bit CPU in Verilog. Verilog is a hardware-descriptive language (HDL) so I had to work around some problems I could have solved with an ordinary programming language like C/C++. While building the CPU, I learned about finite state machines (FSM's), registers, synchronous processing, and memory.

2 Finite State Machine

My CPU has 4 states: one to set the operation code, one to set the first register id, one to set the second register id, and one to set the immediate value to be loaded to a register. The user pushes a button to traverse through the states, setting each value with 4 switches.

3 Registers

The registers are each given 8-bits and can be set using the set immediate value operation. Once an instruction is set (by setting the opCode, reg id's, and immediate value) that instruction is performed on the set registers.

4 Operations

I got all of the operations set up and able to be set and displayed on 7-segment displays, but I didn't get around to implementing them all. The one's that work are ADD, SUBTRACT, AND, OR, EXOR, LOAD, MOVE. The one's that are broken or haven't been implemented yet are INVERT, INCREMENT, SHIFT RIGHT ,SHIFT LEFT ,COMPARE, BRANCH IF EQUAL, BRANCH IF GREATER.

5 Display

Throughout each state, a display is given on 4 7-segment displays. They tell you which operation, register, or value you are setting as you traverse through the options with the switches. There is also a display button that can be pressed while traversing through the registers to see that register's value.

6 Conclusion

In all I wrote about 880 lines of Verilog code. I lost track of time, but I believe I put around 25 hours into the project, however I am not a very fast worker. I would expect an average student to complete this project in around 20 hours. The most difficult part was not being able to use arrays and for loops as freely

as I would have been able to in C/C++. Other than that, I learned the most about Verilog through this project and had the most fun with it, even over the reaction timer, although this project was much more difficult.