BSUCA12 Single Cycle Processor

Must be submitted electronically by Midnight on Friday, April 13.

In this part of the project, you will build and test the complete, but non-pipelined, BSUCA12 processor using the components you have been building during the course of the semester. As part of this project, you will demonstrate that your design works correctly in simulation and show how fast it can execute a set of provided test programs.

The processor has three inputs:

- ✓ a 1-bit input called `clk`
- ✓ a 1-bit input called `reset`
- ✓ a 16-bit input called `instruction`

The processor must be reset before running a new program. Upon reset, default values will be loaded into the registers and the control unit outputs will be all zeros. In addition, the fetch unit should load the PC with the address of the first instruction to execute (0x0000). Each instruction should execute in one clock cycle. The frequency of your BSUCA12 will depend on the way you implemented it and you should be able to calculate this value from the delays given in each of the designed units (remember that each of the components designed had a delay value associated with it, use similar values when you design the rest of the data path and the controller). The fetch unit adder should use another instance of the adder you have already designed. Testing of your design will be done on the maximum value of frequency that you calculated.

The processor has five outputs:

- ✓ a 1-bit output called `destination_sel`
- ✓ a 16-bit output called `write_data`
- ✓ a 16-bit output called `destination_address`
- ✓ a 16-bit output called `destination_reg`
- ✓ a 16-bit output called `instruction_address`

The single bit output "`destination_sel`" is used to identify the destination as specified by the instruction being executed: ('1' for register and '0' for memory). All other outputs are self explanatory. The instruction memory is located in the testbench. It reads instructions from a file that is created from an assembler program. These instructions will be fetched when a new address is supplied by your data path in sequence. The testbench should provide a print out of the instruction entering the data path and the output values of the data path after the execution of a particular instruction. High level module must be named `BSUCA12` and file name must be `BSUCA12.v`