LAB #3: Modeling Structure in VHDL

This homework is due in the homework locker at 5pm on Thursday, September 16, 1999. NO LATE HOMEWORK WILL BE ACCEPTED.

In this homework, you will complete the following problems using the VHDL simulator within VeriBest's DesignView. For each problem, turn in your VHDL code and a waveform(s) showing your code working. Be sure to comment your code and annotate your waveforms for full credit.

1. In the last lab, you implemented a 32 bit ALU with support for the following operations: add, sub, and, or, and complement. The ALU also produced an output signal that is asserted when the ALU output is 0. For your 32 bit ALU from the last lab, you were told to use two processes: one to describe the ALU and one to test it. Separate it into two entity/architectures putting the ALU process into one and the testbench into another. The testbench should include the ALU as a component.

2. Construct a 1 bit ALU that supports the same operations as the 32 bit ALU. This ALU should be built using basic gates such as AND gates, OR gates, XOR gates, MUX's, etc. You should also build a behavioral model of each basic gate that you need. Use generics for the gate delay.

3. Using your 1 bit ALU as a component, construct a 32 bit ALU. Don't forget to add a component to compare the 32 bit output with 0. This design should be an alternative architecture for the ALU from the first problem (i.e., it should use the same entity). Use a configuration to select which architecture to simulate (i.e., the behavioral one or the structural one). Use the same testbench to test both ALUs to verify that they have the same behavior.

4. Construct a 1-bit register again only using basic gates (you may assume that a D flipflop and a tri-state buffer are basic gates).

5. Using the 1-bit register as a component, construct a 32-bit register.

6. Using the 32-bit register as a component, construct a register file with 8 registers. Again compare the behavior with the register file from the last lab.