LAB #4: Packages and I/O in VHDL

This homework is due in the homework locker at 5pm on Friday, September 24, 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. Write and test a set of functions for performing arithmetic left and right shifts on vectors of type std_logic_vector. Put these functions into a package. Put the package into a library.

2. Implement an ideal memory model in VHDL. There are two special inputs to this memory: load and dump. When the load input is asserted, the memory is loaded from a file named mem.ram. The load wire should be asserted at reset, deasserted after a short delay, and never asserted again during the simulation (in other words, this can be shorted to the reset wire). When the dump input is asserted, the memory dumps its contents to a file named results.ram and halts the simulation.

   There are two input buses: an address bus (Address) and a data input bus (WriteData). There is also a data output bus (MemData). The width of each bus should be set by a generic. For testing, set the address and data width to 8-bits (i.e., a 256 byte memory). When the generic is set to a larger data width (i.e., 16 or 32 bit), you may assume only aligned memory accesses. In other words, for a 16-bit wide memory, the low order bit of the address should always be 0, and for 32-bit wide memory, the two lowest order bits should be 0. Use an assertion statement to check for unaligned memory accesses.

   There are three 1-bit control inputs MemRead, MemWrite, and the Clk signal. Whenever the Address changes while MemRead is high, the data at that address is read and placed on the MemData bus. If MemWrite is high, then the data on the WriteData bus is written to the address on the Address bus on the falling edge of the Clk signal.