EE 3755 Homework 1 Due: 7 February 2002

Important: When logging in for the first time select "Common Desktop Environment" after entering your user name and password. (If you're reading this after selecting the wrong option, log out and then, from the log in dialog select Options, Session, Common Desktop Environment.)

Solve this problem by modifying a copy of `http://www.ece.lsu.edu/ee3755/2002/hw01.html` *which can also be found in* `/home/classes/ee3755/com/v/hw01.v`. *See* `http://www.ece.lsu.edu/ee3755/proc.html` *for instructions on running the simulator. Alternate instructions can be found in Lesson 7 of the ModelSim Tutorial, linked to the references web page,* `http://www.ece.lsu.edu/ee3755/ref.html`. *The links are clickable when this assignment is viewed with Acrobat Reader. The ModelSim tutorial and other documentation can also be accessed from the Help menu on the ModelSim GUI (started by the command vsim -gui).*

Problem 0: Copy the homework template, `/home/classes/ee3755/com/v/hw01.v`, into a subdirectory named `hw` in your class account. Simulate the welcome module in the homework template. It should print a "Hello, World!" message.

Problem 1: Complete module `decode_2_to_4` so that it implements a 2-to-4 decoder. The first four ports on the module are 1-bit outputs, the fifth port is a 2-bit input. The module should operate as a standard decoder: The first 1-bit output should be 1 and the other outputs should be zero when the input is zero; the second output should be 1 and the others zero when the input is 1, etc.

- The module must be in explicit structural form. Do not use assign statements or any operators.
- Be sure to declare the inputs and outputs (they are omitted from the template). Note that if the inputs and outputs are declared incorrectly the module may compile without an error but an error message will be given by the simulator when the design is loaded.
- Use the testbench `test_decoder` to test the module.

Problem 2: Complete module `atoi_implicit` so that it converts a hexadecimal number in a two-character ASCII string to an integer. The input port `s` holds the ascii string, with bits `7:0` holding the LSD and bits `15:8` holding the MSD. The string can contain the digits 0-9 (remember, they are in ASCII) and the letters a-f, assume no other characters, including A-F, will be present. The output should be the value of the hex string in binary.

- The module should use implicit structural code, meaning the `assign` keyword and operators can be used.
- As always, try to avoid using expensive operations, such as multiplication, division, and remainder.
- The ASCII value of "0" is 48 and the ASCII value of "a" is 97.
- Use the testbench `test_atoi` to test this module (and the one for the next problem).

Hint: Look at the binary representations of the ASCII characters "0" and "a". The problem can be solved by adding two lines of Verilog.

Problem 3: Complete module `atoi_explicit` so that it does the same thing as the previous problem, but uses explicit structural code only. That is, `assign` and expressions should not be used except for bit select, part select, and concatenation.

- In addition to the primitive gates the solution can use the following modules (defined in the template): `add4`, `sub4`, `add8`, `sub8`.
- Use the testbench `test_atoi` to test this module (and the one for the previous problem).

  This problem can also be solved by adding just two lines of Verilog.