EE 4702 **Homework 1** **Due: 12 February 2001**

Solve this problem by modifying a copy of `http://www.ee.lsu.edu/v/2001/hw01.html` *which can be found in* `/home/classes/ee4702/files/v/hw01.v`. *See* `http://www.ee.lsu.edu/v/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.ee.lsu.edu/v/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 1: Copy the homework template, `/home/classes/ee4702/files/v/hw01.v`, into a subdirectory named `hw` in your class account. Simulate the welcome module in the homework template. If it works, a message should tell you to proceed to problem 2.

Problem 2: In Homework 2 (yes, this is Homework 1) a priority encoder will be designed which has an n-bit input and an n-bit output. Let bit positions be numbered from $n-1$ to 0 and let bit zero be the least significant and the rightmost bit when written. Output bit i, $n-1 \geq i \geq 1$, shall be `1` if input bit i is `1` and if input bits $i-1, \ldots, 0$ are all `0`, otherwise output bit i is zero. Output bit 0 is `1` if input bit 0 is `1`, otherwise it is `0`. Therefore, at most one output bit is 1, corresponding to the first input bit that is `1`. Some examples: 0011 → 0001, 0110 → 0010, 0111 → 0001, and 0000 → 0000, where foo → bar indicates that output bar is expected for input foo.

The encoder will be constructed from n cells in the same way a ripple adder is constructed from binary full adder cells. These cells will be designed here, in Homework 1.

Complete module `priority_encoder_1_es` in the homework template so that it is a Verilog explicit structural description of the priority encoder cell. **Do not** rename the module or change any of its ports.

Problem 3: Complete module `priority_encoder_1_is` in the homework template so that it is a Verilog implicit structural description of the priority encoder cell. Do not rename the module or change any of its ports.

Problem 4: Complete module `priority_encoder_1_b` in the homework template so that it is a Verilog behavioral description of the priority encoder cell. Do not rename the module or change any of its ports.

Problem 5: Complete module `test_pe` in the homework template so that it tests the three modules designed above. The test should be by exhaustion. That is, apply all possible combinations of inputs to each module and verify the outputs. (Consider only `0` and `1` for inputs, but watch for `x` or `z` at the outputs, which would indicate an error.)

Module `test_pe` has four one-bit outputs. Output `done` should be set to `1` when the tests are complete. When `done` is `1` outputs `okay_b`, `okay_is`, and `okay_es` shall be set to `1` if the respective module works correctly or set to `0` if the respective module does not work correctly.

As before, do not rename the module or change any of its ports.