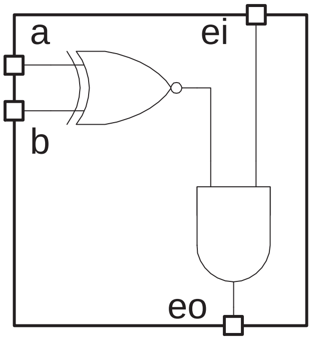
EE 3755 Homework 1 Due: 24 September 2001

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/ee3755/2001f/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 1: 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.

The logic diagram below is a 1-bit slice of a circuit that is used to determine whether two integers are equal. One bit of one integer is put on input `a` and one bit of the other integer is put on input `b`. (If the number has ten bits then ten slices would be needed.) Ports `ei` and `eo` work something like carry in and carry out in a binary full adder. Input `ei` is logic 1 if the higher bits of the two numbers are equal. Output `eo` is logic 1 if the higher bits are equal and `a` and `b` are equal.



The homework template contains module definitions for solutions to the problems below. It also contains a testbench, called test, that can be used to test the modules. This should be the module that you simulate when testing the other modules.

Problem 2: Complete module `eq_slice_es` (in the homework template) so that it is an explicit structural description of the 1-bit slice hardware illustrated above. Remember that this module only handles one bit of `a` and `b`.

Problem 3: Complete module `eq_slice` so that it is an implicit structural description of the 1-bit slice hardware illustrated above. Remember that this module only handles one bit of `a` and `b`.

Problem 4: Module `equality` is used to compare two four-bit numbers. It has two four-bit inputs, `a` and `b`, and a one-bit output `eq`. Complete the module so that output `eq` is 1 iff `a` and `b` are equal. The module must instantiate four copies of `eq_slice`, from the problem above.