

Solve this problem by modifying a copy of <http://www.ee.lsu.edu/v/2001/hw04.html> (or *.v*) which can also be found in `/home/classes/ee4702/files/v/hw04.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>. This page also has links to manuals for the synthesis program, Leonardo. 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:** Write a synthesizable Verilog behavioral description of a microwave oven controller in module `microwave_oven_controller` that passes the testbench in `test_oven`. The module is the same as the one assigned in Homework 3 with the following differences. There is a third input, `reset`. The oven should reset if `reset` is one at a positive edge of input `clk`. This is to be used for a power-on reset, it is not the front-panel reset button, and so the oven should reset regardless of what it is doing.

Input `key_code` should only be examined at positive `clk` edges. Input `key_code` will be set to a key's code as long as a key is pressed. Do not expect users to hold down keys for only  $\frac{1}{64}$  of a second. As before `key_code` will be `key_none` when no key is pressed.

The module must be synthesizable using the provided synthesis script (see below) and the synthesized hardware must pass the testbench.

Follow these steps:

(1) Write an oven module that passes the testbench (without synthesis). This can be based on your submission to Homework 3, a classmate's submission to Homework 3, or the solution to Homework 3 (when that is posted). Note that the testbench tests the module needed for this homework, which is slightly different than the one designed for Homework 3.

(2) Synthesize the module. This can be done in three ways:

- In Emacs: press S-F9 (shift f9) while a buffer with the oven module is active. Lines containing error, warning, and information messages will be highlighted. If mouse-2 (the middle button) is pressed while the pointer is over a highlighted message Emacs will jump to the corresponding line in the Verilog description.
- From a shell: type `syn.tcl hw04sol.v`.
- Using the GUI: start Leonardo by selecting "Leonardo" from the slide-up menu over the Emacs kitchen-sink icon on the CDE control panel at the bottom of the screen. Select the SCL05u technology target, under ASIC and Sample. Load the homework solution and press Run Flow. Additional steps are needed to generate Verilog output. Use the first two methods when Verilog output is needed. (The GUI can be used, but the scripts are easier.)

Make sure the module synthesizes (look for a "Synthesis Complete" message), correct any problems if it does not.

(3) Run the testbench on the synthesized module. To do this, load or restart the testbench into Modelsim **without** recompiling it. (The synthesis script should have compiled the synthesized module for you.) If this is done correctly Modelsim should print many lines that look like "Loading work.OR4T2," the names of the technology modules. Run the testbench and correct any errors.