EE431 Final Exam 2007

Date: April, 2007
In class for first 45 minutes.
In Computer Laboratory for 2 hours 15 minutes
Computer Laboratories: Delta, Gamma and
Material Allowed: Text Books, Notes, Worked examples,
Verilog and hex files
NOT ALLOWED: Cell phones, Access to email and the web.

(8) 1. This question is based on assignment 11. In preparation create a new folder called 'Modelsim_final'.

(a) Move file stack_test.v from 'I drive (I:) →classes → EE431' to folder 'Modelsim_final'. Move the stack module you designed in assignment 11 to folder 'Modelsim_final'. Then make a new Modelsim project called 'stack_test' and add your stack module as well as file stack_test.v to the project.

(b) An instantiation of a stack is made in *stack_test.v* just above the endmodule Replace or modify this instantiation to instantiate your stack.

(c) Test bench stack_test.v contains the stimuli to test your stack. One of the stimuli is the *push* signal, which is a signal that is generally 1'b0 except for the occasional 1 μs pulse. Your are to modify *stack_test.v* to add one more pulse to the *push* signal. That pulse is to occur between 50 μs and 51 μs.

(d) Compile and simulate.

(e) Observe "accumulator_output" when "counter_full_bar" is truly 1'b0. This will be near the end of the simulation. If your circuit is working properly the output should be **16'H1C03**.

(f) Change statement where *seed* is set to 8'HFF for the duration of the simulation so that *seed* is set to the value shown on your answer sheet for the entire simulation.

(g) Recompile and simulate.

(h) Report "accumulator_output" when "counter_full_bar" is truly 1'b0.

(5) 2. Questions 2, 3 and 4 are related. Question 3 builds on question 2 and question 4 builds on question 3. In this question you will be connecting your micro to the test bench and modifying the computational unit.

(a) Make a folder named "final_ques_2" and inside that folder make a second folder named "exam_test_bench".

(b) Copy the verilog file "exam_test_bench.v" from directory I drive (I:) → classes → EE431 to the "exam_test_bench" folder.

(c) Copy all the verilog files in your microprocessor to the "exam_test_bench" folder.

(d) Make a new Quartus II project called "exam_test_bench" and make the "exam_test_bench" folder the working directory.

(e) Modify the "exam_test_bench" module by:

i. Replacing the line near the top that reads
reg [7:0] output_from_circuit_under_test;
with
wire [7:0] output_from_circuit_under_test;

ii. Replacing the two-line always block located just above the endmodule statement, which reads

```
always @ *
output_from_circuit_under_test = input_for_circuit_under_test;
```

with two statements:

A. The first is an instantiation of your entire microprocessor, in which "clk" is connected to "clk"
"clear" is connected to "reset"
"input_for_circuit_under_test[3:0]" is connected to "i_pins"
and "o_reg" is connected to "output_from_circuit_under_test[3:0]".

B. The second is an assign statement that directly connects
"input_for_circuit_under_test[7:4]" to "output_from_circuit_under_test[7:4]".

iii. Modify your computation unit to change the four instructions
mullo x0,y0;
mullo x1,y0;
mullo x0,y1;
mullo x1,y1;
to no-operation instructions.

(f) Copy the Intel format hex file "final_Q2.hex" from directory I drive (I:) → classes → EE431 to new folder "exam_test_bench". This Hex file is to be used to initialize the program memory ROM. It can be integrated into your microprocessor by renaming it to be the name that you used.

(g) Compile "exam_test_bench.v".

(h) Open a waveform/vector file.