

EE431 February, 2010 Midterm Material up to end of Assignment 5

Date: Monday, February 8, 2010

Time = 1.00 hours

Text Books, Notes and Computer Files Only

Preamble

This exam is based on a verilog module called "exam_test_bench_2010". It is provided for you on the I drive. Instructions on how to retrieve it are given later. The port list (Verilog 2001 extension format) for this module is given below.

```
module exam_test_bench_2010 (
input clk, clear,
input [7:0] seed,
input [7:0] exam_number,
output reg [7:0] stimulus,
output reg [7:0] response,
output reg counter_full_bar,
output reg [15:0] accumulator_output
); // verilog 2001 extention
```

It has four inputs and four outputs.

In this exam you will be simulating the circuit built from prototype "exam_test_bench_2010". You will have to create a vector waveform file (i.e. a .vwf file) and make waveforms for the four inputs. Input "seed" will be an 8 bit constant given in the answer sheet. The input "exam_number" will also be a constant. It will be either be set to 8'd00 or set to the exam number (e.g. if your exam is exam number 3 then "exam_number" is set to 8'd3). Inputs "clear" and "clk" are waveforms that will be described in the questions.

The signals "stimulus" and "response" are declared outputs for the sole purpose of debugging. You may or may not want to use them. Their values are not reported in this exam.

You are to report the value of "accumulator_output" at the time when the output called " counter_full_bar" is low.

You will be designing circuits in this exam. The circuits you design are instantiated inside "exam_test_bench_2010" and will affect its outputs. The module "exam_test_bench_2010" contains instantiations of two different prototypes that you will be asked to modify. These prototypes are instantiated with

```
student_circuit cct_1(.clk(clk),
                    .clear(clear),
                    .cct_input(stimulus),
                    .cct_output(cct_output) );
```

```
student_scrambler scmblr_1( .clk(clk),
                           .clear(clear),
                           .exam_number(exam_number),
                           .scrambler_input(cct_output),
                           .scrambler_output(response) );
```

If you open the prototypes for “student_circuit” and “student_scrambler” you will notice that some of the inputs are not used. This is not a problem. The compiler just ignores any unused inputs.

Questions

(1)

1. Make a folder named “Feb_2010_midterm”.

Copy the verilog files “exam_test_bench_2010.v”, “student_circuit.v” and “student_scrambler.v” from directory I drive (I:) → classes → EE431 to the new folder.

Make a Quartus project called “exam_test_bench_2010” and make “Feb_2010_midterm” the working directory.

Make file ‘exam_test_bench_2010.v’ the top entity.

Choose the Cyclone II FPGA family.

Then do the following:

- (a) Compile “exam_test_bench_2010.v”.

NOTE: The compiler will issue some warnings. They should be the usual warnings of undefined clock, capacitance not assigned to pins, etc. Read them to make sure they can be safely ignored.

- (b) Open a waveform/vector file.
- (c) Make the simulation time 300 μ s.
- (d) Import all the input and output pins into the .vwf file.
- (e) Make the clock, which is “clk”, a square clock with period 1 μ s. The clock should start low and make a transition to high at 0.5 μ s.
- (f) Make input “clear” a pulse that is high from 0 to $6.25 \pm 0.1 \mu$ (i.e. the transition from high to low can be anywhere between 6.15 and 6.35 μ s.) and low for the remainder of the simulation.
- (g) Make input “seed”, the constant 8'HFF for the entire time of the simulation.
- (h) Make input “exam_number” the constant 8'd0 for the entire time of the simulation.
- (i) Perform a simulation.

- (j) Observe “accumulator_output” when “counter_full_bar” is a steady 1'b0. If your circuit is working properly the output should be as indicated on the answer sheet.
NOTE: counter_full_bar is the output of a combinational logic circuit so may contain glitches. When it is 1'b0 during a glitch it is not truly low.
- (k) Change the input “exam_number” to a constant equal your exam number. If your exam number is 3, then make the constant 8'd3. Simulate the circuit again and report the value of “accumulator_output” during the time “counter_full_bar” is a steady 1'b0.
- (2)
2. Change the prototype “student_scrambler” as per the instructions on the back of the answer sheet.
 - (a) Compile “exam_test_bench_2010.v”.
 - (b) In the waveform editor, change “exam_number” to 8'd0 and leave all other inputs unchanged.
 - (c) Simulate to check the validity of your circuit. The value of “accumulator_output” when “counter_full_bar” is a steady 1'b0 is given in the answer sheet.
 - (d) Change input “exam_number” to the number of your exam and re-simulate. The value of “accumulator_output” when “counter_full_bar” is a steady 1'b0 is given in the answer sheet.
 - (e) Change input “seed” to the number shown on your answer sheet. Change input “exam_number” to 8'd0. Re-simulate then record the value of “accumulator_output” when “counter_full_bar” is a steady 1'b0 on your answer sheet.
 - (f) Change input “exam_number” to the number of your exam and re-simulate. Record the value of “accumulator_output” when “counter_full_bar” is a steady 1'b0 on your answer sheet.
- (1)
3. In this question you will change the value of input “seed” and re-simulate. The value to be used for “seed” is the determined from the Verilog HDL given below. Use the value of “calc_seed” at the time when “q1==8'd79”.
After changing the value of “seed” in the vector waveform file, re-simulate and record the value of “accumulator_output” when “counter_full_bar” is a steady 1'b0 on your answer sheet.

```
output [7:0] calc_seed
reg [7:0] counter, q1, q2, q3, q4, calc_seed;
always @ (posedge clk)
    counter = counter+8'd1;
always @ (posedge clk)
begin
```

```

q1 = counter;
q2 = q2 ^ q2;
q3 = q1 | q2;
q4 = q3 + 8'd1;
calc_seed = q4 + 3'b010;
end
    
```

- (3) 4. Change prototype "student_circuit" found in file "student_circuit.v" to make the circuit shown in Figure 1.

After the modifications to prototype "student_circuit" have been completed, re-compile the top entity "exam_test_bench_2010.v" etcetera and report the result on the answer sheet.

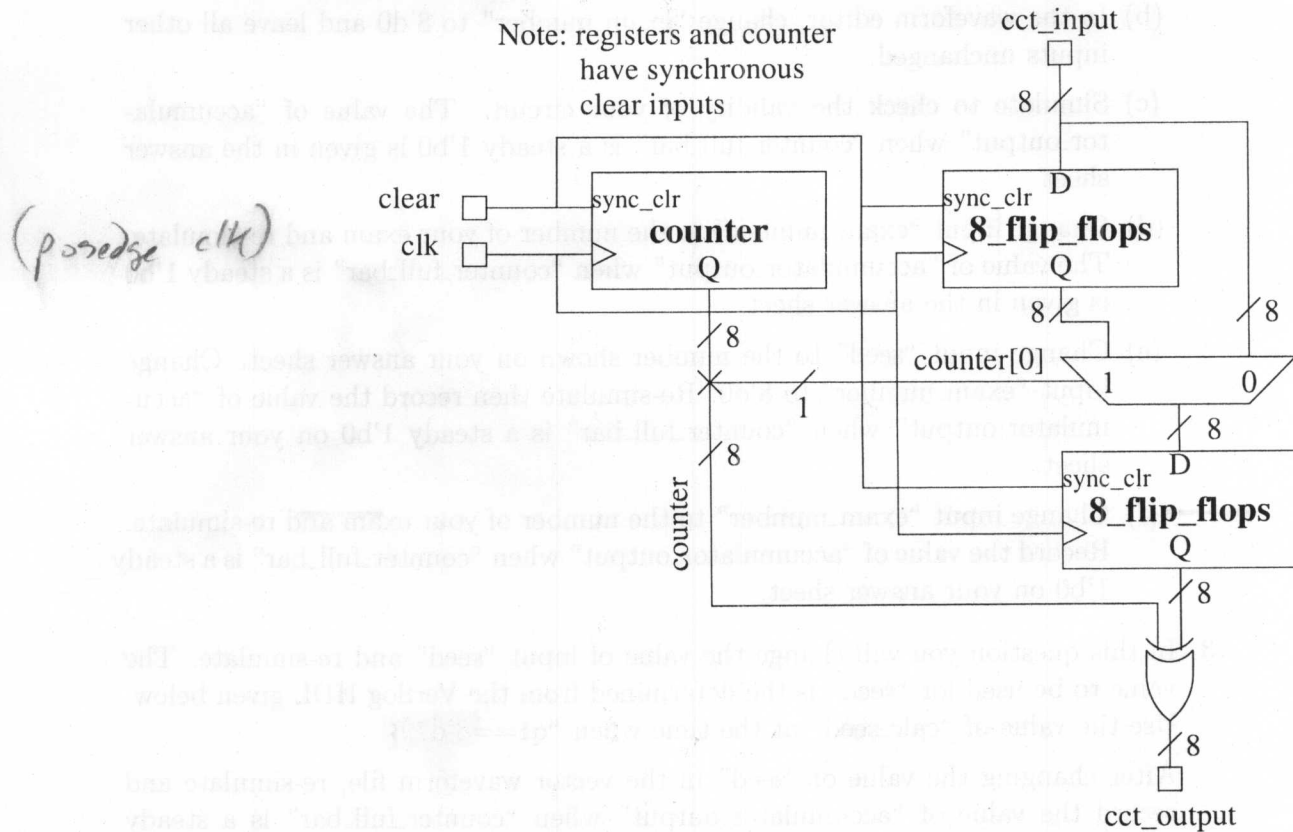


Figure 1: A block diagram of the circuit to be described in prototype "student_circuit".