TCES 330, Digital Systems Design
Midterm Exam
May 9, 2013

General Comments:
Follow the directions very carefully. You will be partly graded on how well you can follow these directions. This exam requires you to create and test a Quartus II project; this project is described below. When you are finished notify an instructor and we will upload your exam to Moodle.

Do NOT connect to the Internet during the exam. Do NOT use your cell phone in any fashion during this exam except for emergency purposes; keep your cell phone out of view. Do not consult any other individual except for an instructor. This is an **open book** exam. You may use your text books, your notes, my slides, the lab assignments, Altera tutorials, etc.

When you set up your project, consider the following:

- Pick the Cyclone II EP2C35F672C6 device.
- Assign unused pins properly and set output pin capacitance
- Pay close attention to the warnings you get when you compile.
- Use comments appropriately. Especially make sure your name appears in **every** Verilog module.

There are no 'trick' questions. This exam is a straight-forward test of your abilities to write Verilog HDL programs and to verify them.

Problem.

We want to design a state machine that acts as a modulo-5 counter. It should behave as follows. It is reset to 0 by the *Reset* input. It has one input, w_0, which controls its counting operation. If $w_0 = 0$, the count is incremented by 1. If $w_0 = 1$, the count is decremented by 1. All changes take place on the positive edge of a *Clock* input. Use toggle switch SW_0 for input w_0. Use toggle switch SW_1 as an active-low synchronous reset, and use the pushbutton KEY_1 as a manual clock. Display the decimal contents of the counter on the 7-segment display *HEX0*.

1. On a sheet of paper, draw the state transition diagram for this state machine. Label all states. Show all state transitions, including any transitions back to the same state. Show the output for each state. Put your name on this sheet of paper!
2. On your computer create a folder (directory) called MT1.
3. In this folder create a state machine module with the following signature

```
module Mod5FSM( Clock, Reset, W0, D );
  input Clock; // system clock (derived from KEY0 in this case)
  input Reset; // resets the mod 10 counter to 0
  input W0;    // determines count up or count down

  // declare D here!
```

4. In folder MT1 create a new project which will be used to implement the circuit on the DE2 board and create a top-level module that will interface your state machine to the switches and KEY0. Use LEDR0 and LEDR1 to indicate the state of SW0 and SW1.
5. You will need your Hex display decoder for the numbers 0-9 (or 0-F is OK, too). **Copy this decoder, whatever you call it into your MT1 folder.** Don't take the chance that I can't find it when I compile your project!
6. Compile your project, removing all warning that we normally remove. Also assign pins. If you are left with latch warning or hierarchy warnings, or something similar, fix these problems now.
7. Run the TimeQuest Timing Analyzer and create a *.sdf file that contains a clock. Remember KEY1 is our clock in this system. You can set the period to be 20 ns.

8. Recompile. You should have no warnings left.
9. Look at the Netlist Viewers/RTL View and verify that Quartus found a state machine. If you don't have a state machine, see if you can modify your code so that you do. Indicate on your sheet of paper whether or not the Quartus state machine matches yours.
10. Download to the DE2 board and verify your circuit.

When you are done with your project.

Zip the contents of the project folder into a zip file called **yourname_MT1.zip**. When I open your zip file I should see your folder called MT1. Raise your hand to summon an instructor and **then and only then** you will be allowed to connect to the Internet (Moodle) to upload your files. Then turn in the sheet of paper you used in Step 1 **with your name on it.**