EEL 3705L, Digital Logic Design Lab, Spring 2011

Lab Assignment #6: Settable Digital Clock with Seconds Display

Version history:
v1.0, 3/3/11: Created the initial version of the assignment. –M. Frank

1. Document description:

This document describes the sixth full lab assignment (2nd lab assignment after the midterm) for the Spring 2011 semester of EEL 3705L, Digital Logic Design Laboratory; this assignment's lab periods will take place the week of Mon. Mar. 14-Thu. Mar. 18.

2. Assignment Synopsis:

In this assignment, students will create a modular sequential logic design using VHDL, to implement a simple digital clock application displaying hours, minutes, and seconds, and that includes an interface to allow the user to set the current time. This assignment is to be carried out as a structured design process. Students should design and test their circuits at home in the simulator, and then test their designs in lab on the boards.

3. Educational Objectives:

This assignment is intended to:

1. Give students practice at design & coding of modular sequential designs in VHDL using behavioral coding with process statements.
2. Exercise students' understanding of basic modular sequential elements such as counters.
3. Give students practice at programming, simulating, and debugging modular sequential HDL-based designs using the Quartus tool.
4. Continue exercising students' ability to go through a structured design process with all appropriate documentation.

4. Design Objective:

In this assignment, your task is to design and prototype sequential logic to implement a simple digital clock with a decimal hours/minutes/seconds display (HH MM SS). You may optionally include two more digits, for hundredths of seconds (HH MM SShh).

The prototype clock should be as accurate as possible given the limitations determined by the precision of the clock frequency of the DE2 board's clock oscillator – however, at minimum, your clock should not noticeably gain or lose seconds, compared to a reference clock (e.g., stopwatch), over the course of a 5-minute long test run.

The design should provide a simple means for the user to set the time, for example, buttons to advance the hours/minutes, and a button to reset the seconds to 0.

The design should be coded in a modular style, with emphasis on VHDL. At least the major core module(s) used to update the time must be written in sequential VHDL, in a behavioral style using a process statement, with statements to update state variables (not just combinational outputs this time). You are highly encouraged to code your entire

design, or at least, as many modules as you can, in VHDL instead of schematics (since you need to get used to programming in VHDL anyway – since schematic entry is often too slow and cumbersome to be the most practical approach for very large designs).

5. Pre-Lab Preparation:

Reminder: You MUST complete the pre-lab assignment BEFORE coming to lab, and turn in your pre-lab report printout at the start of lab.

In this and future lab assignments, please continue to follow a structured design process, similar to the one you went through in detail in earlier lab assignments:

1. Identify the design context.
2. Specify the system-level requirements.
3. Create the system-level design.
4. Specify component requirements.
5. Create a testing plan.
6. Create the detailed design of the components.
7. Perform simulation testing of components.
8. Construct a prototype (in lab).
9. Test the prototype (in lab).

5.1. Guiding Questions

The following questions are intended to help guide and stimulate your thought process as you work through this assignment. As you gain experience, eventually you will be able to go through this type of thought process on your own, without being prompted.

Question #1. Think about (and sketch) your overall system architecture. Your design will require a counter or counters of some sort. How many separate counters will you use in your design? What size should each of them be (in bits)?

Question #2. What will you use as your time reference? You could build your own ring oscillator, but that would not be very precise. There is a clock oscillator built into the DE2 board. What frequency does it run at? What is the name of its FPGA input pin? (Check the DE2 board user manual.)

Question #3. Since the built-in oscillator is much faster than 1 Hz (the rate at which you want to update your display – or 100 Hz if you're displaying hundredths of seconds), you will need a clock divider. Given the frequency of your reference oscillator, how many cycles of this oscillator will pass between display updates? So, how many bits large does your clock-divider counter need to be? However, the number of oscillator cycles per second is probably not a power of 2. Think about how you will make your clock divider produce output pulses (to trigger the seconds updating) after the appropriate number of cycles (just rounding to the nearest power of 2 and using the carry-out of a normal binary counter will not be precise enough).

Question #4. Similarly, think about how you will cause seconds to roll over (and minutes to increment) every 60 seconds, and how you will cause minutes to roll over

(and hours to increment) every 60 minutes, and how you will cause hours to roll over every 12 hours. **NOTE:** For consistency with ordinary wall clocks, the hours display must run from **1 to 12**, and **NOT** something else like 0 to 11 or 0 to 23.

Question #5. Think about how manual setting of time will be implemented. Specify the user interface for this in detail. What input devices on the DE2 board will you use for this purpose in your prototype? Do you require an extra input signal or signals to be fed into some (or all) of your counter modules to permit the time to be set? Specify the meaning of those inputs. Figure out how to use them to do what's required to allow the user to set the time in the way that's desired.

Question #6. By this point, you should have sketched out the basic structure and/or function of all your modules in block diagrams and/or pseudocode in your lab notebook. Now it's time to actually implement them in VHDL and/or schematics – although you should emphasize VHDL, as the problem statement specified.

Question #7. For simulation purposes, when simulating your system, you may find it convenient to simplify some parts of your design. So, for example, rather than simulating a big counter running for literally millions of cycles (which would take a very long time to simulate, if the simulation tool can even handle it), you may instead want to temporarily pretend, for simulation purposes, that seconds are much shorter relative to the board clock cycles, and adjust your design accordingly, so that you can at least verify the basic architecture of your design in the simulator. Once you know that it works, you can change the design back to the version meant for the real board prior to simulation testing.

Question #8. Think about design-for-test considerations, and plan ahead for prototype testing. Like in lab 5, you may want to tap out some extra signals for testing purposes, signals which will help you diagnose any problems in the design. For example, displaying raw counter values in binary on the LEDs may help you to narrow down whether any problems you see are in your display driver or your counter. There may not enough LEDs to display *all* your counter bits, but you could, for example, use a mux controlled by input switches to allow the user to select, at any given time, which of several different counters to pass through to the LED bank.

5.2. Pre-Lab Report Format

For this assignment, you may follow the generic pre-lab report format that is given in the *General Lab Requirements* document. This format is a little bit simpler than the one you had to use for lab #1. (Some of the larger projects you will do later in the semester will return to a more detailed format.) However, do be sure to include the simulation results, and the experiment design for the prototype testing.

6. In-Lab Procedure:

In lab this week, all you need to do is to put together and test your design, according to the experimental testing procedure that you already came up with in your pre-lab report. As you go along (at home and in lab), take detailed notes in your lab notebook (in

accordance with the Lab Report Requirements section of the *General Lab Requirements* document) that you will base your final report on. Be sure to take note of anything that happens during prototype testing that is different from what you expected, and take detailed notes that you will use for analysis and interpretation of your experimental results in your final lab report. If any design modifications turn out to be needed, go ahead and make them, and re-test the design until it is working. Before leaving the lab, demonstrate to the TA that you can set the time, and that your prototype keeps pace with the real time over a significant period (at least a minute), and show the TA your code. Remember to turn in your final lab report the following week.