CSE P567 Homework #2 Winter 2010

Due: Tuesday, Jan 19 – Problem 1 at the beginning of class, Problem 2 in Lab

1. Read the following pages of Chapter 15 of the book "**EDA for IC implementation, circuit design, and process technology"** edited by Louis Scheffer, Luciano Lavagno, Grant Martin. This chapter covers hardware description languages and new languages that are being used for verification. The section we will read focuses on Verilog and an overview of VHDL and System C. Write a short (< 1 page) summary that includes answers to the following questions:

a. What is the difference between using an HDL for modeling and synthesis?
b. What is the difference between wires and regs, and what problem do regs cause?
c. How is writing Verilog circuit descriptions different from writing parallel software programs?

As a UW student, you have access to this book online. Follow this link and then go to Chapter 15 via the table of contents. Note: If you have trouble accessing this chapter (see note on the publisher Web page), please send me email. I was able to access this via Windows, but via my Mac.
http://www.engnetbase.com.offcampus.lib.washington.edu/ejournals/books/book_summary/summary.asp?id=4676
Read the beginning of the chapter up to Sec. 15.3.2.1 (pages 15-1 – 15-10), Sec. 15.3.2.2 to 15.3.3.2 (pages 15-13 – 15-16).
(If you have problems with this link, try going in through the UW Library main page to click the off-line login link at the top right of the page. Then use the search to find the appropriate book.)

2. This problem is a design problem that you will use in Lab 3 – if you have it completed by the next class, then this Lab will go very quickly for you!
This design adds a frame counter to the video pipeline. This counter counts the frames using the "newFrame" signal, which is asserted for one clock cycle every frame, and displays the count using the 7-segment displays on the Altera board. The design has two parts:

a) A counter that starts at zero (on reset) and increments whenever the increment input is asserted. This count will be displayed using the 4-digit 7-segment display. We could keep this count in binary, but since we want to display it in decimal, we would have to use division. The alternative is to count in decimal, so we don't have to do any conversion. Your counter will be a 4-digit counter, where each digit counts to 10.

b) Decoders that take each digit and assert the appropriate 7 segment enable signals for the 7-segment displays.

We are giving you a template for this design that uses the following design hierarchy:

- frame_counter – root-level module that is instantiated by the cameraPipeline design
    - count10 – a one-digit decimal counter. Instantiated 4 times to make a 4-digit counter.
    - dec7seg – translates one 4-bit digit into the 7 enables for a 7-segment display. Instantiated 4 times, once for each digit. Note that the segment enables are asserted low. That is, the segment is lighted with the enable=0.

We are also giving you a test fixture (hw2_tf.v) that you should use to test your design. Here are the interfaces for the two modules you will be designing.

```
module count10(
  input clk,                 // System clock
  input reset,               // Synchronous reset
  input increment,           // Increment count when asserted
  output reg [3:0] count,    // Count output
  output carry               // Asserted when next digit should
increment
  );
```

```
 *           Segment numbering:
 *               ---0----
 *               |      |
 *               5      1
 *               |      |
 *               ---6----
 *               |      |
 *               4      2
 *               |      |
 *               ---3----

module dec7seg(
  input [3:0] in,              // Decimal digit (binary)
  output reg [6:0] segmentL    // Segment enables are assert LOW
  );
```

Note: The test fixture prints out a pictoral description of the 7-segment displays. To make it look right, you need to Edit Preferences (Tools menu) and change the main window text font to a fixed-point font like Terminal 9pt.