EE3921 Digital Systems Design

Section 031, Fall 2021

Professor: Kerry R. Widder, Ph.D.

Lab Exam

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* **Date: Friday, Nov. 5, 2021**
* **Duration: 1 hour 50 minutes**
* This exam is open book and open notes.
* Laptops may be used for code development.
* Laptops may not be connected to any network at any point during the exam. **You will need to obtain the component code I am providing before you disconnect.**
* You may not receive help from any other person.
* You may use code examples distributed during lecture. You may use code that you wrote as part of previous lab assignments.
* You may use any of the reference materials, user's guides, and datasheets provided on the official course website (must be downloaded to your computer prior to the start of the exam).
* You must demo (i.e., check-out) the working program(s) to your Professor.
* When you complete your exam, logon to the internet and print your source code to OneNote.
* To earn maximum credit, your programs must:
  + exhibit good programming techniques
  + be fully compliant with the specifications and requirements
* Partial credit may be awarded for programs that exhibit minor bugs, but are otherwise fully-functional.

|  |  |
| --- | --- |
| Score | /100 |

1. Design a digital system using the DE10-Lite board to have the following functionality:

* Generate milli-second timing and display it on the 7-segment displays HEX3 through HEX0, i.e., have a display of X.YYY, where X is the number of seconds, and YYY is the number of milli-seconds.
* If SW(0) is a ‘1’, pause the time (be sure to pause the time counting, not just freeze the display with the timer still counting). If SW(0) is a ‘0’, then allow the timing to be active.
* If one of the KEY pushbuttons (pick one) is pressed, reset the time to 0.0. After it is released, if SW(0) is ‘0’, the count should immediately start going, otherwise hold it at 0.0 until SW(0) becomes ‘0’.
* **When the seconds digit of the count is odd, generate the sequence of lights shown on the next page using the HEX5 and HEX4 seven segment displays of the DE10-Lite (Note: in the patterns shown, a filled-in segment is ‘on’, a blank one is ‘off’).**
* For the seven-segment sequence, use the supplied count component to time the movement of the segments through the sequence.
* The seven-segment sequence should only be moving while the seconds digit of the count is odd. During all other times, the segment pattern should remain where it ended up when the seconds digit changed to be even.
* If SW(9) is a ‘0’, run the pattern ‘backward’, I.e., move up the page, otherwise run the pattern forward, i.e., move down the page. In both cases, wrap around and start over when the other end is reached.

Your design will be evaluated based on how well it meets the specifications listed above. It will also be evaluated based on how you implemented your solution.

* If you create your own design, you can achieve a maximum of 80 points
* If you use the components I provided in your design, you can achieve a maximum of 90 points
* If you use the components I provided in your design, and you also use a ROM to store the data for the seven-segment display segments of the moving pattern, you can achieve a maximum of 100 points

Be sure to follow good code formatting practices (header info, indenting, appropriate comments, etc.). Compile and download your code to the DE10-Lite board and demo the correct functionality to the professor.

When you are done, connect to the internet, print your code to OneNote (code includes VHDL code you wrote, .mif file).

. .

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Full custom | Components | ROM |

|  |  |  |  |
| --- | --- | --- | --- |
| Check off |  | Implementation |  |
| Timing |  | Milli\_timer1 use |  |
| Correct pattern |  | Count\_en3 use |  |
| Repeats |  | ROM use |  |
| Read switches |  |  |  |
| Pause |  |  |  |
| Reset |  | Submit files |  |

1

