CSCE 313

Lab 1

Lighting up the DE2 Board

Due Date: 2/11, 11:59pm

Design Requirements

This lab will introduce you to the Altera platform tools for developing embedded systems for the DE2 board.

Part 1

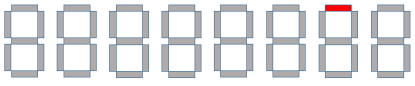
Follow the informal tutorial in the lecture slides to implement a system with the following characteristics:

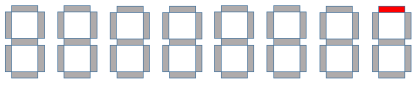
- One of the 24 LEDs on the board is always lit
- The lighted LED changes in a fixed periodic interval, which can be every 250 ms, 125 ms, or 50 ms, depending on the current mode of the system
- On each change, the lighted LED shifts to the left until it reaches LEDR17, at which point it shifts to the left until it reaches LEDG0, and then repeats
- Pushing KEY3 changes the period to 250 ms, KEY2 to 125 ms, and KEY1 to 50 ms
- The console (UART) should display the new period when the mode is changed

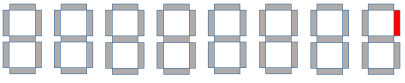
Part 2

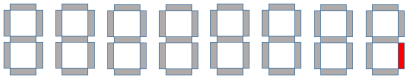
Start with Use the design from part 1 for the following new features on our own:

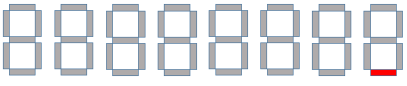
- Add eight 7-bit parallel I/O modules (in output mode) to your SOPC design corresponding to each of the DE2's seven-segment display
- Generate your NIOS system, then connect the resulting outputs to the HEX7 to HEX0 outputs in the top-level design
- Write code for your NIOS2 processor that will change the value of the seven-segment displays at the same period as the LEDs:
    - When the period is 250 ms, the 7-segment displays will light **with random configurations**
    - When the period is 125 ms, the 7-segment displays will follow a pattern where one segment is always lit, and the lit segment follows a clockwise circular path around the outermost ring of segments, for example:

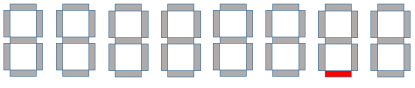
(1) 

(2) 

(3) 

(4) 

(5) 

(6) 

- When the period is 50 ms, the 7-segment displays will show a decimal count from 0 to 99,999,999 and repeat

Setting Up Output Ports for 7-Segment Displays

Please make sure you check the documentation on the 7-segment LEDs before starting this lab:

DE2-115: http://www.cse.sc.edu/~jbakos/313/DE2_115_UserManual.pdf

Specificially, note that these 7-segment displays are active low and the segments go from the least significant bit to the most significant bit. In other words, when bit 0 is LOW the top segment will illuminate, and when bit 6 is LOW the middle segment will illuminate.

1. In SOPC Builder, add a 7-bit output parallel I/O (PIO) for each of the eight displays.

2. Add the following to the nios_system port map (somewhere under the line "nios_system my_system("):

```
.out_port_from_the_HEX0(HEX0),
.out_port_from_the_HEX1(HEX1),
.out_port_from_the_HEX2(HEX2),
.out_port_from_the_HEX3(HEX3),
.out_port_from_the_HEX4(HEX4),
.out_port_from_the_HEX5(HEX5),
.out_port_from_the_HEX6(HEX6),
.out_port_from_the_HEX7(HEX7),
```

3. Delete the following lines from your Verilog code:

```
assign HEX0 = 7'h00;
assign HEX1 = 7'h00;
assign HEX2 = 7'h00;
assign HEX3 = 7'h00;
assign HEX4 = 7'h00;
assign HEX5 = 7'h00;
assign HEX6 = 7'h00;
assign HEX7 = 7'h00;
```

4. After doing this, the eight 7-segment HEX outputs will be connected from your SOPC design through the pins on the FPGA to the actual 7-segment displays. You can access them from your C code using, for example:

```
IOWR_ALTERA_AVALON_PIO_DATA(HEX0_BASE,3);
IOWR_ALTERA_AVALON_PIO_DATA(HEX1_BASE,4);
IOWR_ALTERA_AVALON_PIO_DATA(HEX2_BASE,5);
IOWR_ALTERA_AVALON_PIO_DATA(HEX3_BASE,6);
IOWR_ALTERA_AVALON_PIO_DATA(HEX4_BASE,12);
IOWR_ALTERA_AVALON_PIO_DATA(HEX5_BASE,56);
IOWR_ALTERA_AVALON_PIO_DATA(HEX6_BASE,87);
IOWR_ALTERA_AVALON_PIO_DATA(HEX7_BASE,127);
```

Project Submission

Each group must submit an archive of their complete project directory to Dropbox. Make sure you include your software.