ELE307 Spring2009 LAB6

What's new in this lab:

VHDL	Multi-dimensional array, constant, loop with local variables.
EDA	State Machine Viewer and RTL viewer.
FPGA	PLL, built-in memory array.
RTL	Complex timing signal generation, memory interface.

Exercise#1

- Download lab6_1.qar from ftp://ele.uri.edu/outgoing/jcl/306/lab6_1.qar.
- Restore, and compile the project. Program the DE2 board.
- This exercise gives you a VGA circuit (vga.vhd) that assumes a resolution of 320x240 with 2-bit for color for each pixel. Lines 72 and 73 of vga.vhd describe how the Video RAM address is generated from the horizontal and vertical timing. Be aware that this vga.vhd is different from that in lab5. Lines 107 to 109 show how the 2-bit for the color of each pixel are mapped to the R, G and B. Remember that each color has 10 bits.
- In this exercise, the bouncing ball is made out of four pixels. The colors of each pixel are controlled by switches separately as illustrated below.

SW(17), SW(16)	SW(15), SW(14)
SW(13), SW(12)	SW(11), SW(10)

- The "ball" is moving at a speed controlled by Switches 3 downto 0. The ball only bounces off from the boundaries at right and at left.
- Q1: What is the shortest time interval (fastest speed) between updating the ball on screen?

Exercise#2

- Download lab6_2.qar from ftp://ele.uri.edu/outgoing/jcl/306/lab6_2.qar.
- Restore, and compile the project. Program the DE2 board.
- This exercise uses the same VGA.vhd as in previous exercise. However, this exercise's bouncing ball is a 3x3 pattern pre-defined as a constant (lines 22 to 27) in lab6_2.vhd. The current pattern is a circle, however, with such low resolution it may appear not as a circle visually.
- The color of the ball (all 3x3) is controlled by switches 17 downto 16. Switches 3 downto 0 are still used to control the ball speed.
- In lab6_2.vhd, states "erase1" to "erase6" use variables i and j forming a two-dimensional loop to erase the entire 3x3 pixels. Similarly, states "draw1" to "draw6" uses variables I and j to draw the entire 3x3 pixel conditioned by the pre-defined pattern (line 213). This is a demonstration of how to access an element in a two-dimensional array.
- Q2: Go to "State Machine Viewer". Click on the "Encoding" tab at the lower part of the window to see how the states are encoded in "1-hot" code. Use the RTL viewer to see how each state is represented by one signal (or net).
- Q3: Also locate the two variables i and j in the RTL. How the two variables are realized in hardware? How the two variables are used in the state transition? (see the State Machine Viewer under "transitions")

Assignment (Racquetball)

- Create a simple racquetball game. Imagine the screen as a room with walls on three size. Finish the assignment in the following ten incremental steps: (10% grade each step)
- STEP 1: Clear the entire screen after "reset". The memory contents will not be erased by a simple reset signal. You must write "background color" to all visible locations in Video RAM.
- STEP 2: Draw walls on three sides. You may make the line thicker by increase the pixel width. The room size should be no less than 160x120.
- STEP 3: The ball is a 4x4 pattern of your own design.
- STEP 4: The ball is moving across the room and bounces off walls at the speed controlled by four switches and thus having 16 speed settings.
- STEP 5: The paddles must be 4-pixel wide and at least 16-pixel and no more than 32-pixel long. The paddle is four pixels from the boundary.
- STEP 6: Use two keys on the keyboard to control the movement of the paddle.
- STEP 7: Make ball bounces off paddle.
- STEP 8: Keep track of the misses by displaying the number on seven segment displays.
- STEP 9: Add "Game Over" feature. Assuming that a player has only three lives. After three misses, the game ends and the entire screen is freeze. Hit the "enter" on the keyboard will restart the game. Hitting the "reset" to restart game does not count.
- STEP 10: Add "Pause" feature. The "space" key on the keyboard will pause the game. Hitting the "space" key will resume the game play.

Lab 6 Report:

Title page: Lab title, section number, your name, student ID and email address.

The body of your report should include a one paragraph introduction. Descriptions of your experiences from the exercise parts which associated with the items required below. The following required items should be embedded with the text explanation not aggregated at the end of your report. Approaches and thinking behind your solution to the assignment. Lesson learned from this lab.

From the Exercise #1:

1. Answer to Q1.

From the Exercise #2:

2. Answers to Q2 and Q3.

From the Assignment:

3. Algorithm or flowchart of your Racquetball circuit.
4. VHDL listing.

Also, send the followings to TA:

1. Lab6_3.qar or Racquetball.qar (assignment)