Lab 4

VGA Display – THE FROGGER

Design and implement a digital circuit capable of displaying predefined patterns on the screen of a VGA monitor, and provide the basic components for “The Frogger” game. Your circuit must generate all control and data signals driving the VGA output of the NEXYS3 board. Please refer to the lecture slides for information about the meaning and timing of VGA control signals.

Here are the objects you need to create to implement the game


**Task 1: Display the background**  
(Points: 20%)

Display the background (in **Blue**, **Cyan**, **Black**, and **yellow** color). You can adjust the width and (x, y) coordinates of the lines and rectangles in order to obtain the image shown in the figure below.
Task 2: Extend Task 1 by displaying four Magenta Bricks (Points: 10%)

Display the Bricks on top of the water zone. You can win the game when the Frog reaches any of these Bricks.

Task 3: Display and create movement of the “logs” (Points: 15%)

Extend Task 2 by displaying the “logs” and adding the capability of their movement, as shown in the figure below using white horizontal arrows. The upper row of logs should move horizontally to the right in a wrap-around manner, and re-enter from the other side when they touch the boundary. The lower row of logs should have the exact same movement pattern, but in the opposite direction, i.e., to the left.
Task 4: Display and create movement of the “cars”  
(Points: 15%)  
Extend Task 3 by displaying the “cars” and adding the capability of their movement as shown by white horizontal arrows in the street zone in the figure below. The cars should also move in a wrap-around manner, i.e., toward left for the upper row of the cars and toward right for the lower row of the cars.

![Diagram of cars with white horizontal arrows](image)

Task 5: Introduce “The Frog”  
(Points: 25%)  
Extend Task 4 by introducing “The Frog”. The movement of the Frog should be controlled by four push buttons (BTNU for Up, BTND for Down, BTNL for Left, and BTNR for Right). The Frog should move in the direction represented by the button pressed. The goal of the Frog is to reach any of the magenta bricks located at the top without colliding with any of the cars and without falling into the water. The frog is allowed to take a ride on any of the logs.
In the water zone, the Frog should be able to move Right or Left while riding on any of the logs, which may result in the Frog falling into the water in case the button is pressed too many times. The Frog should be free to move Right or Left once in any of the two safe zones. The Frog will have three lives to reach to any of the magenta Bricks. Each time the Frog collides with a car or falls into the water, it should start its journey from the beginning, with one life being consumed.

**Task 6: Displaying score and number of lives left on VGA display**  
(Points: 15%)

Extend Task 5 by displaying the total score. Every forward move on the street should result in 10 points. Each time the Frog successfully rides on any of the logs, 20 points should be added to the total score. The count on the VGA display should increment and display the total score on the Leftmost magenta Brick.
The Frog should have three lives initially. Every collision with a car or fall into the water should result in the loss of one life. When the Frog while riding a log hits any of the boundary walls, it should result in loss of one life. The count of remaining lives should also be updated on the VGA display. After three such incidents, the game will be over.

When the Frog reaches any of the Bricks, this event should pause the game, and YOU WON should be displayed on the VGA display. Otherwise, if the Frog encounters three collisions or falls during the game, GAME OVER should be displayed on your VGA display.

Final view after winning 😊
**Bonus Task: (Points: 25%)**

Extend the game with the following capabilities.

1. Any loss of life should result in the change of Frog color from green to red. The dead red frog should flash three times a second before disappearing.
2. Increase the speed of cars and wood logs to twice the original speed using a switch.
3. Display the time elapsed since the beginning of the game, with a different precision, depending on the position of switches.
   - Precision: 1 second, format: minutes:seconds, e.g., 3:04
   - Precision: 0.1 second, format: minutes:seconds.tenths-of-a-second, e.g., 3:04.1
   - Precision: 0.01 second, format: minutes:seconds.hundredths-of-a-second, e.g., 3:04.06.

**Deliverables:**

1. VHDL source codes for all modules required by the game.
2. UCF file.
3. A short report describing
   A. List of tasks fully implemented
   B. List of tasks attempted but not completed (please describe shortly what is missing)
   C. List of tasks not attempted
   D. List of any deviations from the original specification.
   E. Resource utilization
      a. Number of CLB slices
      b. Number of LUTs
      c. Number of flip-flops
      d. Number of DSP units
      e. Number of BRAMs
   E. Timing properties
      a. Minimum clock period after synthesis [ns]
      b. Maximum clock frequency after synthesis [MHz]
      c. Minimum clock period after implementation [ns]
      d. Maximum clock frequency after implementation [MHz]
   F. Difficulties encountered and lessons learned.
### Important Dates

<table>
<thead>
<tr>
<th>Event</th>
<th>Monday Section</th>
<th>Wednesday Section</th>
<th>Thursday Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hands-on Sessions and Introductions to the Experiment</td>
<td>03/23/2015, 03/30/2015</td>
<td>03/18/2015, 03/25/2015</td>
<td>03/19/2015, 03/26/2015</td>
</tr>
<tr>
<td>Demonstration and Deliverables Due for Schedule A</td>
<td>04/06/2015</td>
<td>04/01/2015</td>
<td>04/02/2015</td>
</tr>
<tr>
<td>Demonstration and Deliverables Due for Schedule B</td>
<td>04/13/2015</td>
<td>04/08/2015</td>
<td>04/09/2015</td>
</tr>
</tbody>
</table>

(-8 point penalty for not attempting Lab 5 or Lab 6; no penalty for Lab 4)