Final Project Report

By

Sarthak Dubey , Chia-Kai Yeh

Prof. David Zaretsky

EECS 355 : ASIC and FPGA Design

Winter 2016

# OUTLINE TITLE

## Introduction

1. Objective
2. Design
3. Methodology
4. Tank Parameter
5. Height & Width of Tank
6. Bullet Parameter
7. Main Program
8. Tank Movement
9. Bullet Movement
10. Game Over and Draw Cases
11. Board Implementation and Peripherals
12. Synthesis Results

Conclusion

Sarthak Dubey , Chia-Kai Yeh

Prof. David Zaretsky

EECS 355 : ASIC and FPGA Design

Winter 2016

# Final Project Report

**Objective:**

The objective of this project is to a game called “tank duel” that could be displayed on the VGA monitor and controlled by two players by one PS2 keyboard.

**Design:**

The tanks are realized as square boxes on the VGA monitor. They keep continuously moving back and forth on a horizontal line on the screen. Each tank can move with three different speeds. The direction of the tanks can be controlled using left and right keys. Both tanks shoot one bullet at a time; if it hits the opponent’s tank, it scores one point. A tank can shoot the second bullet only when the previous bullet goes out of the screen or hits the opponent. The score of both players is displayed on the LEDs. Whoever first scores 3 points wins. When a player wins, only the winner’s tank is displayed and the winner message is displayed in the LCD screen. The game resets on pressing the reset button.

**Methodology:**

**Tank Parameter:**

The tank parameter description is handled by the file tank\_parameter.vhd. Consisting of the tank and bullet size descriptions, this file is used in the main program to direct the VGA display regarding the tank and the bullet at any particular point.

**Main Program:**

The main Program entity tank.vhd consists of the following processes that control the entire behavior of the game and its individual components.

* The first part of the process (Control Tank 1) controls the direction and the speed of the movement of the tank 1.
* The second part controlling the Tank 2 is similar to the “Control Tank 1” and only contains the highlighted part in correspondence with Tank 2 signals and variables.



The next part of the program defines the bullet inside the same game PROCESS as 2 similar files named Tank 1 and Tank 2 bullet. This sequence implements the active cases for the bullet display like:-

* If the bullet exists or the bullet is triggered – display bullet
* Hide/delete the active bullet when it reaches the coordinate of tank 2.
* If no command to shoot & bullet does not exist, do not show bullet.
* Update bullet position if it’s being displayed.



The last part handles the situation of the FSM when the bullet hits the Tank. If both bullet coordinate for bullet #first\_number and Tank #second\_number coincide, tank explosion process is started and calculation for the points and game over bit is issued. The code also checks for rare cases due to a draw situation and then resets the game.

The second process handles the Keyboard presses and corresponds to the key declaration.



Keyboard instruction Set :-

|  |  |  |
| --- | --- | --- |
| Function | P1 | P2 |
| Low speed | 1 | Numpad 1 |
| Intermediate speed | 2 | Numpad 2 |
| Top speed | 3 | Numpad 3 |
| Reverse | Tab | Numpad . |
| Shoot | Space | Numpad 0 |
| Reset | Backspace | |

**Components Used:**

**VGA\_top\_level:** This is the top level entity that structurally connects the VGA, LED and LCD components that display the output.

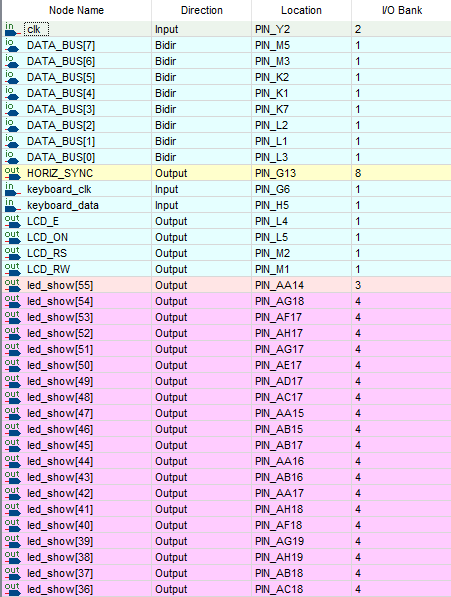
**lcd:** The lcd component maps de2lcd component and displays the winner message based on programvalues.

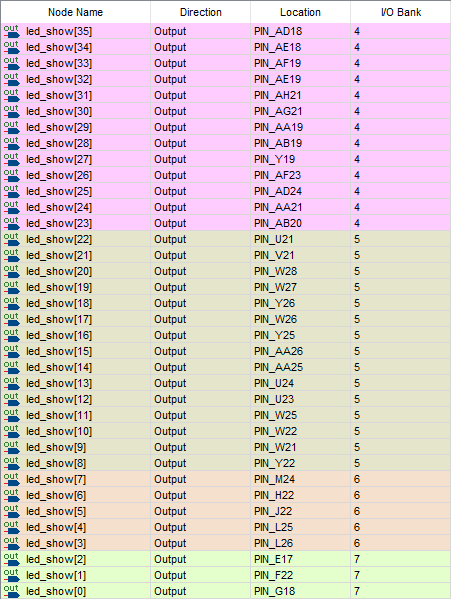
**leddcd:** This component decodes the hexadecimal score into seven-segment display for both the tanks.

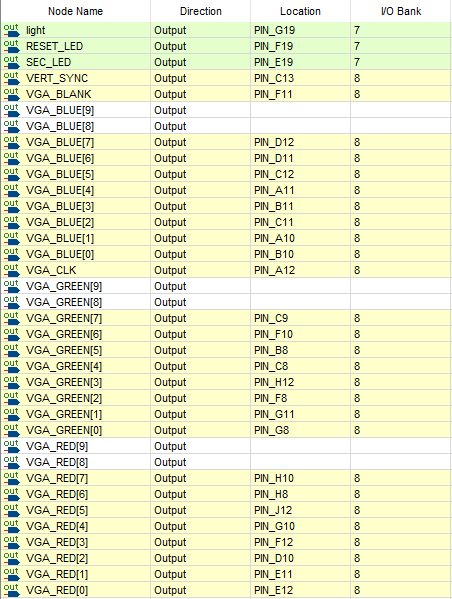
**pixelGenerator** : The provided pixelGenerator component was modified to map ps2 component to read keyboard press values, decode them accordingly and make changes in the pixel values sent to the VGA control. To identify keyboard values, scan\_readyo, scan\_code and hist1 signals were compared. As the clock speed is too high for human eye perception, counters were used to reduce the speed at which the bullet moves.

Other components such as oneshot, colorROM, ps2, keyboard, VGA\_SYNC and leddcd that were provided were not modified.

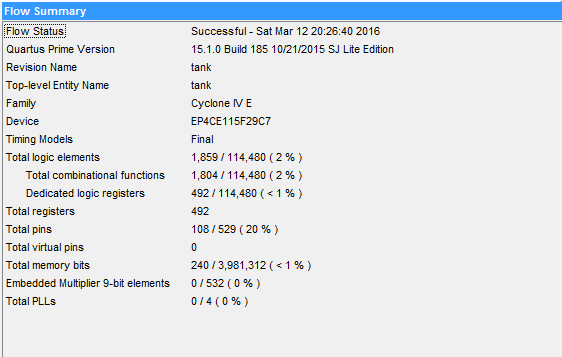
**Board Implementation, Peripherals:**







**Synthesis results, including memory, clocks, and resource utilization:**



**Conclusion:**

---------------------------------------------------------------------------------------------------------------------