Portland State University

SoC Design With FPGAs

ECE540

Tunnel Vision

Erik Rhodes Bhavana Dhulipala Rohan Deshpande Nikhil Patil

March 20, 2014

1 Introduction

Tunnel Vision is a racing game that can be played on the **Xilinx Nexys3 FPGA** board and be displayed on a VGA monitor.

1.1 Gameplay

The player tries to avoid hitting the walls as it travels down the tunnel by moving the vehicle left and right. The space in between the walls steadily decreases until the player hits a wall or obstacle. The score is based on the amount of time the vehicle remains "alive", and is displayed on the 7-segment display.

1.2 Controls

The player can move his vehicle by using the left and right pushbuttons on the Nexys3. When the game is over, hitting the middle button will reset the course. The top button starts the game and the bottom pushbutton pauses it. Different icons and speeds can be selected by toggling the switches on the board.

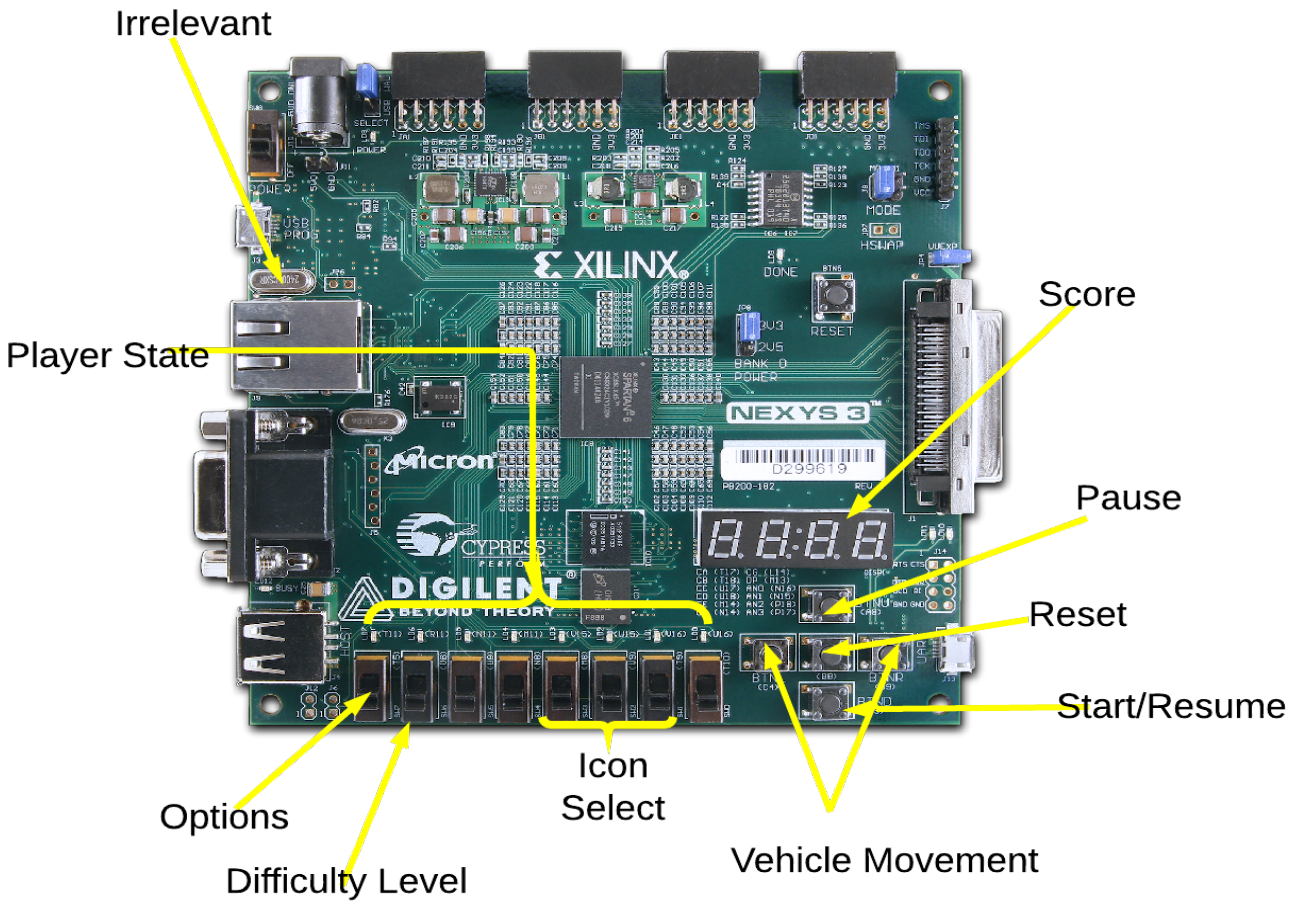

Figure 1: Player Controls

1.3 Features

Tunnel Vision features both starting and ending screens. The courses are generated randomly through a pseudo-random number generator. Additionally, the LEDs are lit with certain patterns depending on the action the player is taking. If the player selects the harder difficulty, the score is incremented at a faster rate and with a multiplier, awarding them a higher score for the same distance traveled.

2 Implementation

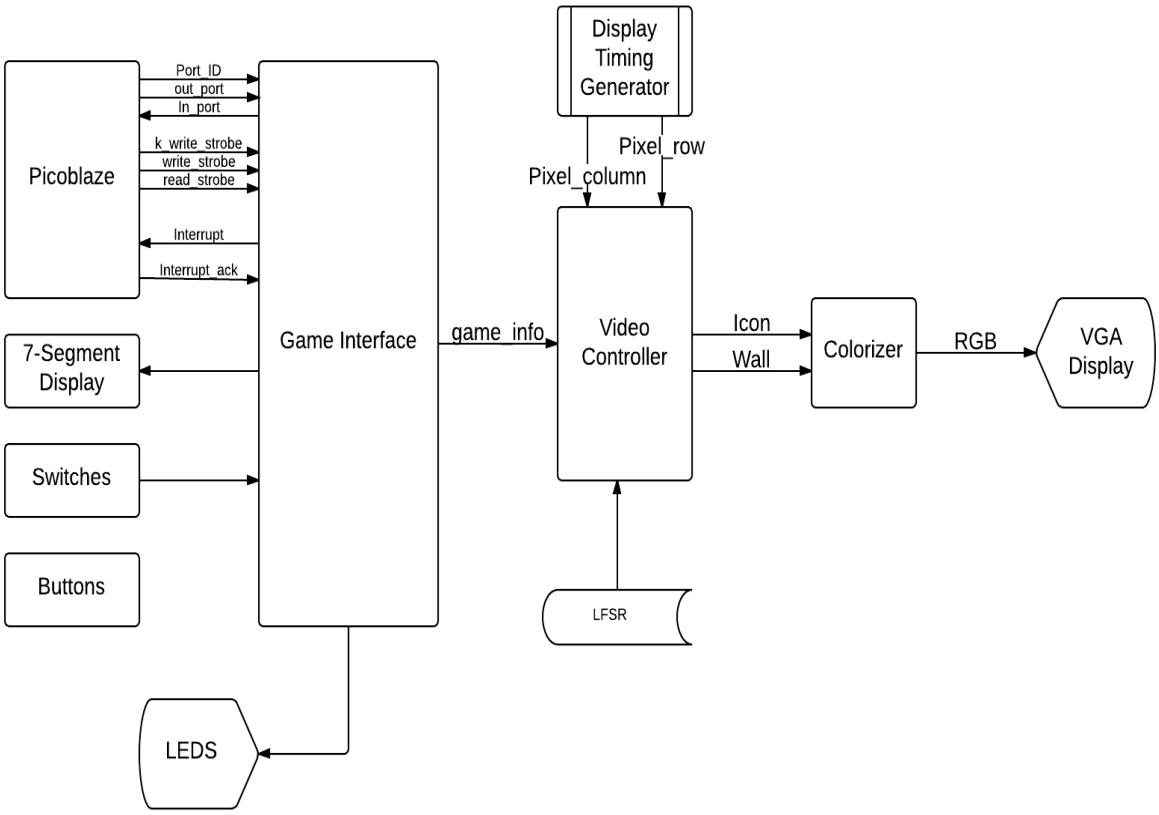

Figure 2: Gameplay Block Diagram

2.1 Picoblaze

Picoblaze assembly code was used to implement the algorithm controlling the vehicle's movement. Game logic, controls, score, levels, etc...
Insert various code here

Listing 1: Sequence used manage orientation counter

```
1  LOAD   s0 ,   LocX
2  FETCH  s1 ,   SP_OLD_LOCX      ;see if our current location is different
3  COMPARE s1 ,    s0             ;if it is, we must be moving forward on a black line
4  CALL   NZ,    clear_counter    ;we can clear the orientation counter at this point
```

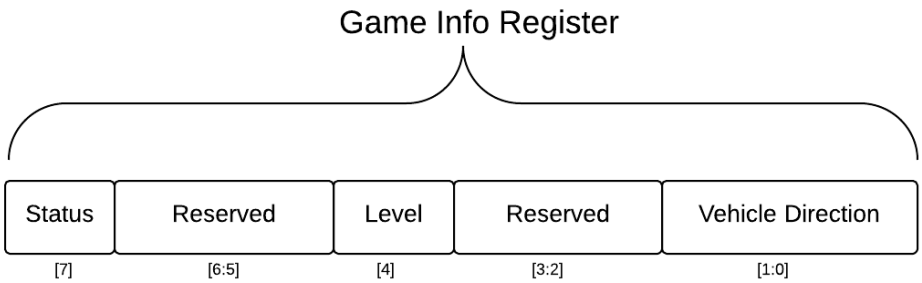

Figure 3: Allocation of bits in game_info register

3 Video Controller Implementation

The video controller module was designed... The icon, wall, and different backgrounds implementation

3.1 Colorizer

3.2 Icon

4 Conclusion

Length of time, github, results, etc.

4.1 Challenges

- Basically issues
- problems we had

4.2 Time Invested

	Erik Rhodes	Bhavana Dhulipala
bot_ctrl.psm	✓	
nexys_bot_if.v		✓
nexys3fpga.v	✓	✓
colorizer.v		✓
icon.v		✓

Table 1: Division of Tasks

4.3 Future Work

While our project completed all requirements and executed perfectly, there is still room for improvement. Future modifications would include:

- **Multiplayer Mode:**