Portland State University

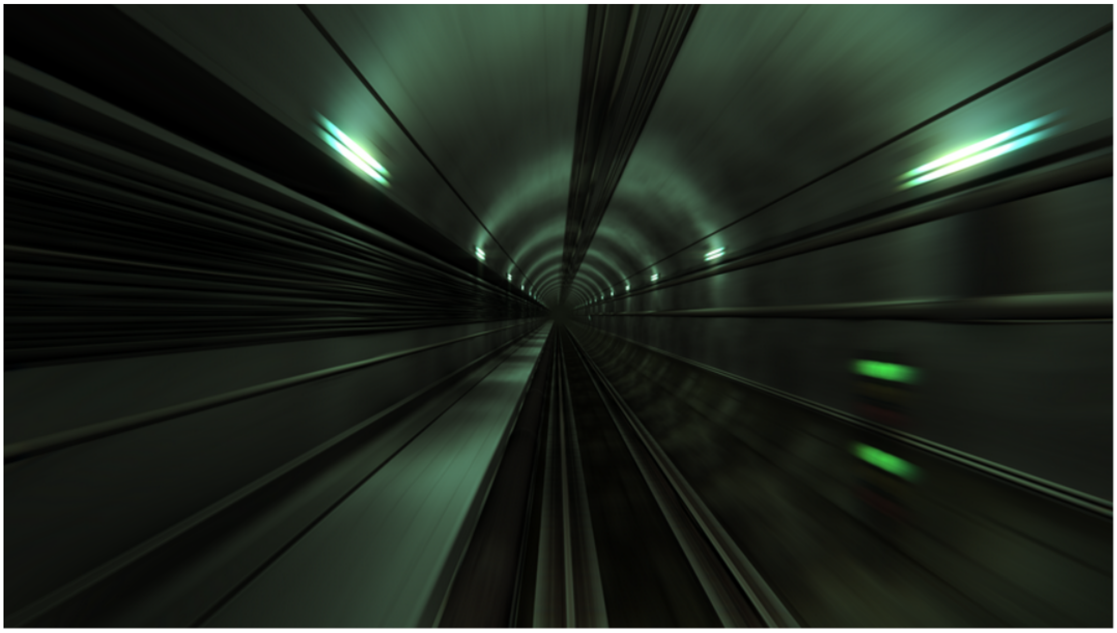
SoC Design With FPGAs

ECE540

Tunnel Vision

Erik Rhodes Bhavana Dhulipala Rohan Deshpande Nikhil Patil

March 7, 2014



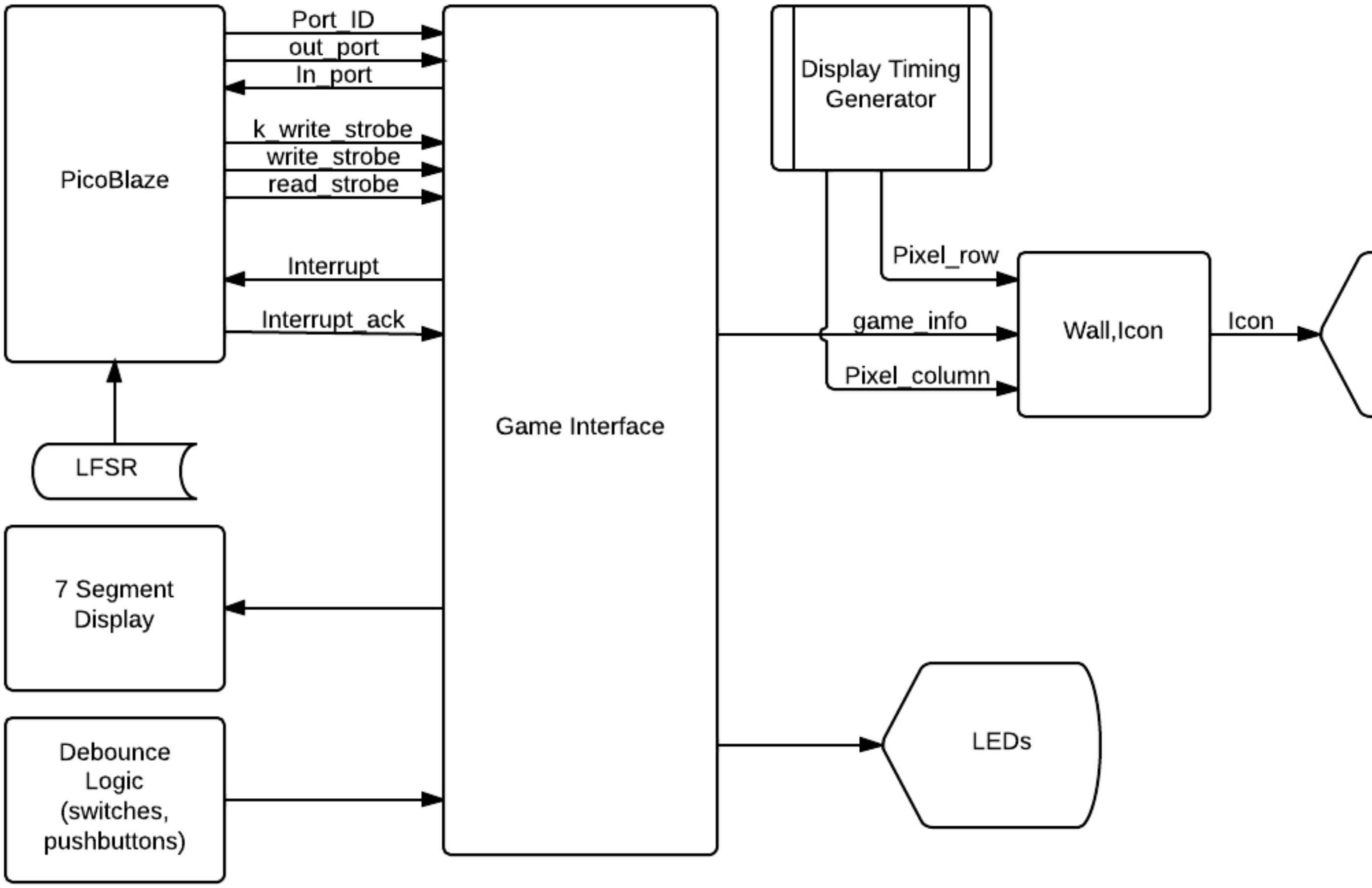


Figure 1: Gameplay Block Diagram

1 Introduction

What is tunnel vision, in general how does it play?

2 Software Implementation

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

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

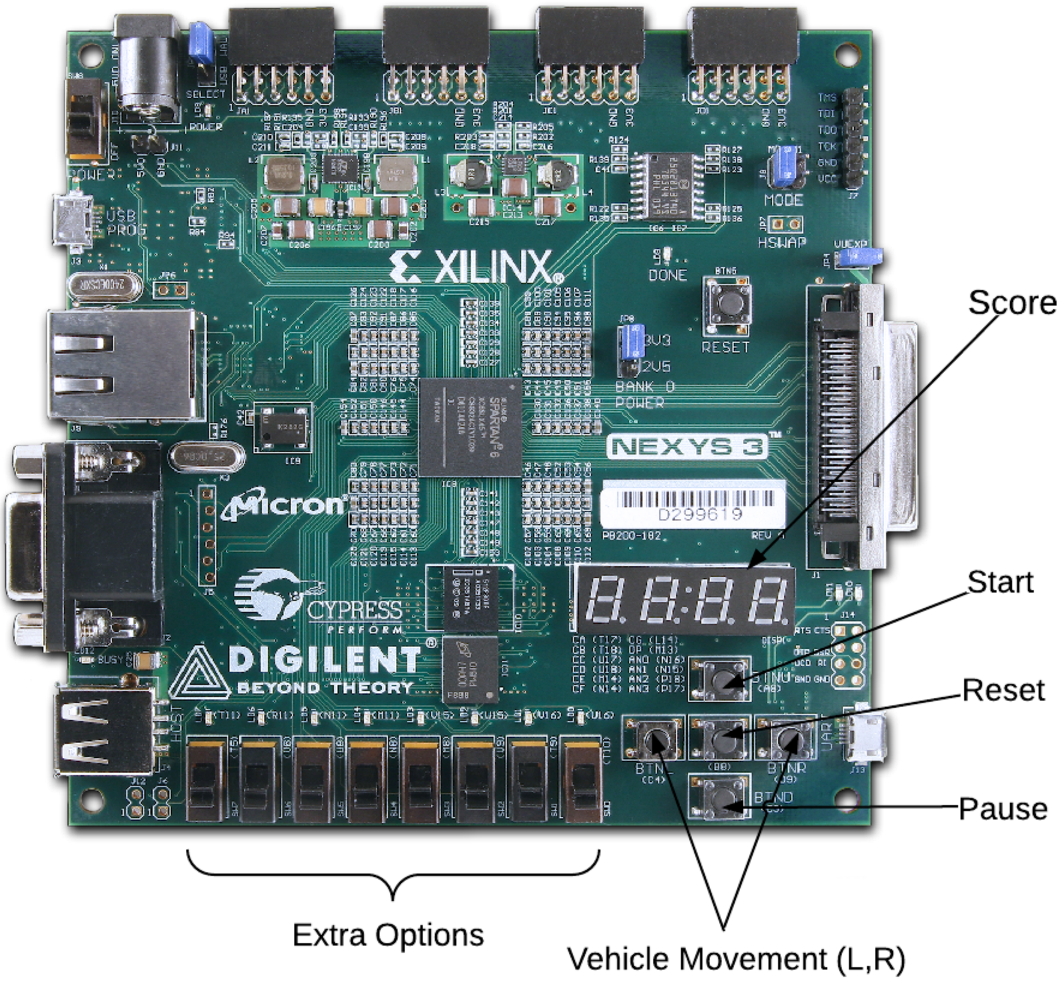


Figure 2: Bot Control Flowchart

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

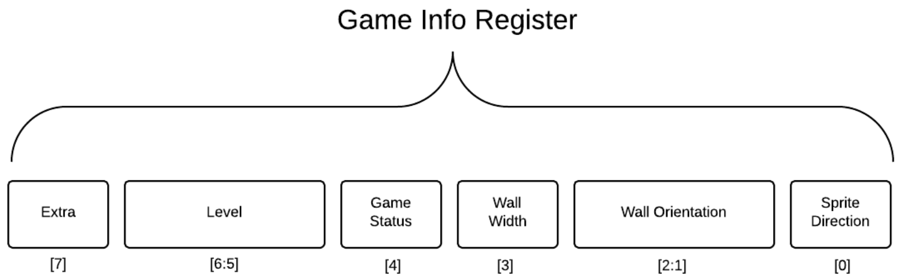


Figure 3: Allocation of bits in game_info register

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:**