ECE 385

Final Project Proposal

mswei2, dylanjb5

Nick Lu

1. Idea and Overview

We propose to design and implement a basic, working version of Tetris on the FPGA. Our implementation will include all the basic rules of Tetris (stacking blocks, clearing rows, speed increasing). Additionally, if time permits, we plan to add audio with the game, support for the FPGA joystick and button inputs, and additional levels once a score is increased (advancing levels will increase speed at which blocks drop). We will utilize RAM, video display, and keyboard input with our design. Our design will use a NIOS II CPU for interfacing with the keyboard, similar to how it was used in lab 6.2 as well as RAM and VGA output like how it was used in labs 7.1 and 7.2. For software code, we believe it is easier to have the Tetris game coded in C, so most of the game logic will be present in software while most of the low-level side (user input, outputting to monitor) will be handled by the FPGA or hardware. Also with C, we get access to some of its libraries which will make stuff like generating random blocks easier to implement. In the end, we wish to demonstrate accurate gameplay of Tetris in which blocks can be generated, stacked, and erased as well as accurate score tracking as per normal Tetris gameplay.

1. Block Diagram

Diagram

Description automatically generated

1. List of Features

**Baseline Features**

1. Rows successfully cleared
2. Score can be tracked accurately
3. Blocks can be generated and placed properly
4. Gameplay and actions of block correct (rotations, moved)
5. Can detect if player loses (reach the top)
6. Score display on Hex displays

**Additional Features (if time)**

1. Add audio for placing blocks, background music, and clearing rows
2. Add support for the FPGA controller and joystick
3. Add different levels (once score reached, next level initiated and speed increases)
4. Putting score on screen

1. Expected Difficulty

The expected difficulty for this Tetris emulation should be in its base state at least a 5. This difficulty is justified since the given examples on the slides listed Tetris at this number assuming accurate gameplay. This would entail scorekeeping and basic movement features as described in the baseline section. If we were to include audio, FPGA controller, and/or different levels this would increase our difficulty by one point for each of these. Thus, the total maximum count if we were to achieve total functionality would be 8 points.

1. Proposed Timeline

Week 1: Figure out Platform designer, VGA output, I/O input, and which modules are needed/potentially needed to create the design.

Week 2: Generate blocks successfully on screen. If extra time, can try to get blocks moving left and right from user input and down when generated.

Week 3: Get blocks moving and rotating properly (no overlapping, can detect if another block is reached, etc.). Accurately clear rows and increase/keep track of score (should go together).

Week 4: Debug any bus detected in previous week and/or finish what was not completed by the deadline. If time, try to add and implement the additional features, but keep a backup for the working final project if successful.