

FPGA Tetris - A Classic Game Reimagined on DE1-SoC (Proposal)

Chuyi Jiang(cj2792) Xinzi Yu(xy2590)

CSEE W4840 Spring 2024

1 Introduction

The goal of this project is to reimagine the classic Tetris game on a hardware platform, specifically utilizing the DE1-SoC development kit from Terasic. This platform, equipped with an FPGA and essential peripherals for input/output operations, provides an ideal environment for implementing a high-performance, interactive game. The project will not leverage the SoC's ARM processor; instead, it will focus purely on the FPGA logic, ensuring compatibility with a wide range of FPGA boards. This proposal outlines the game's specifications, design considerations, and the development approach.

2 Project Objectives

The primary objective is to create a fully functional Tetris game with the following features:

- a) A standard set of Tetris blocks with behavior closely mirroring the classic game.
- b) Colorful gameplay, with each Tetris block having a distinct color.
- c) Randomly generated Tetris blocks with uniform distribution.
- d) A preview window displaying the next block.
- e) Game state information displayed on-screen, including score, lines cleared, and current level.
- f) Progressive scoring system and increasing difficulty levels.
- g) Detection of game over conditions and the ability to start a new game.
- h) User inputs via a PS/2 keyboard.
- i) VGA output for game display.

3 Hardware Requirements

The project will be developed on the DE1-SoC development kit, chosen for its comprehensive set of peripherals, including PS/2 and VGA connectors, crucial for the game's input and output, respectively. The kit's FPGA component will be utilized to implement the game's logic and rendering engine.

4 Project Components

The schematic of the project is shown in Figure.1.1, and each part is described as follows.

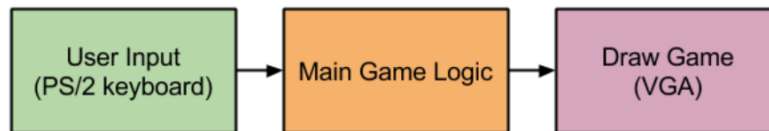


Figure.4.1 Schematic of the project

4.1 User Input

User inputs will be captured through the PS/2 connector, using a PS/2 controller to interpret keyboard commands. This will allow players to control game actions such as moving and rotating Tetris blocks, as well as starting new games.

4.2 Game Logic

The core of the game will be a finite state machine (FSM) that manages game states (Figure.1.2), including block generation, movement, line clearance, and score calculation, etc. The states are shown below.

Block Generator: Creates a new Tetris block, ensuring random selection with uniform distribution.

Movement and Collision Detection: Manages block movements and rotations, checking for collisions with existing blocks or the game boundary.

Line Clearance and Scoring: Identifies and clears complete lines, updates the score based on the number of lines cleared, and adjusts the game's difficulty level accordingly.

Game State Manager: Tracks and displays the current score, level, and lines cleared, along with the next block preview.

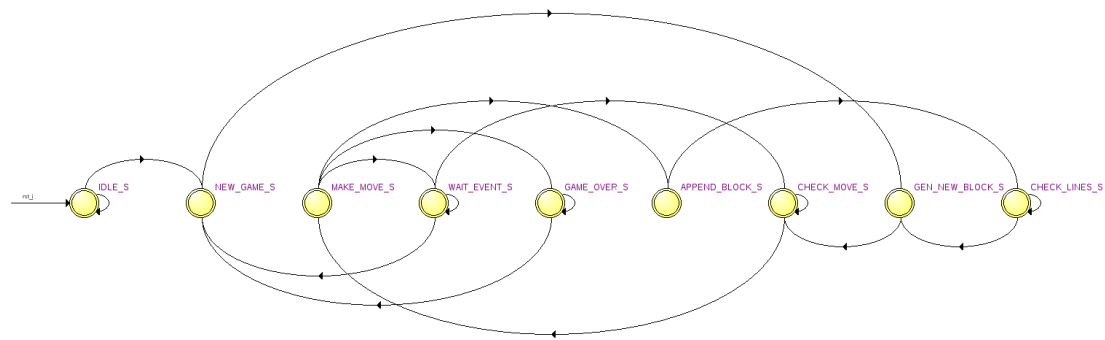


Figure.4.2 Schematic of the project

4.3 Display Output

The game's visual output will be handled via the VGA connector, employing a VGA signal controller to render the game state on a display. The resolution and graphical details will be designed to ensure clear and visually appealing gameplay.

5 Development Approach

The project will be divided into several phases:

1. **Design and Specification:** Finalizing the game's detailed specifications and designing the FSM and other core modules.
2. **Implementation:** Coding the game's logic in Verilog/SystemVerilog, focusing on modular design to facilitate future enhancements or modifications.
3. **Testing and Debugging:** Rigorous testing of individual modules and the integrated game system, using simulation tools and the development kit itself for real-world testing.
4. **Optimization and Refinement:** Fine-tuning the game's performance and visuals, including adjusting the speed, colors, and display resolution to enhance the user experience.