SNAKES AND LADDERS (A python-verilog project)

Submitted as part of Internal Evaluation (CIE) **Mini-project (AECE592)**

BACHELOR OF ENGINEERING

in

Electronics and Communication

Names of the students with USN:

Joanna Philip - 1MS22EC054

K Keerthy Vasan - 1MS22EC056

M. Naga Sarvani - 1MS22EC075

Niyaf Ahmed - 1MS22EC083

Faculty Coordinators

Dr. V ANANDI
Mrs. U.S.Pavitha
Department of E & C

Department of Electronics and Communication

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
Accredited by National Board of Accreditation & NAAC with 'A' Grade
MSR Nagar, MSRIT Post, Bangalore – 560054
2025

Contents

Abstract

This project aims to design a modern adaptation of the classic board game *Snakes and Ladders* using an FPGA-based approach. Combining the hardware logic capabilities of Verilog with the interactive graphical features of Pygame, the project demonstrates the integration of hardware and software for an engaging user experience. The primary goal is to create a single-player game that simulates dice rolls, player movement, and interactions with snakes and ladders using a finite state machine (FSM) model. The FSM ensures structured game progression by managing various states such as Idle, Dice Roll, Player Move, Ladder Climb, Snake Slide, and Win Condition. Python handles the graphical interface, reflecting real-time FSM outputs. This project highlights the practical applications of FPGA technology, emphasizing FSM design and Python integration.

Introduction

Snakes and Ladders is a timeless board game where players advance their pawns across a grid based on dice rolls. The game introduces elements of unpredictability with snakes (which send the player back) and ladders (which propel the player forward). The goal is to reach the final tile before any other player.

This project combines the classic gameplay mechanics with modern hardware and software technologies. Verilog, a hardware description language, is employed to implement the game's FSM, which dictates the flow of gameplay. The FSM's state transitions are based on player actions, such as dice rolls and board interactions. A Python script reads the FSM's outputs from a dynamically updated text file, ensuring that the game's graphical interface in Pygame reflects real-time changes.

By integrating the computational power of FPGAs with the flexibility of Python, this project bridges hardware and software. The result is an innovative gaming solution that not only replicates the traditional *Snakes and Ladders* experience but also offers insights into FPGA-based design and software-hardware communication.

Background and Objectives

Background

Field Programmable Gate Arrays (FPGAs) are powerful platforms for implementing real-time hardware logic. They allow developers to create custom hardware circuits that can perform specific tasks with high efficiency. Finite State Machines (FSMs) are widely used in FPGA design to model systems with distinct states and well-defined transitions based on input conditions. These qualities make FPGAs and FSMs ideal for applications like interactive games.

The project leverages these strengths by implementing the game's logic in Verilog and using Pygame for visualization. Verilog ensures precise and efficient state management, while Pygame provides a user-friendly interface for gameplay.

Objectives

The primary objective is to simulate a single-player version of *Snakes and Ladders* using an FSM implemented in Verilog. The secondary goals include:

1. Designing a robust FSM to handle all possible game scenarios.
2. Creating a seamless interface in Pygame to reflect game progress.

3. Demonstrating the practical application of FSM concepts in interactive systems.
4. Providing a foundation for extending the game to support multiplayer modes or additional features.

Methodology

FSM Design

The game's FSM is structured to model the entire gameplay process, from dice rolls to win conditions. The following states are defined:

- **Idle**: Waiting for the player to roll the dice.
- **Dice Roll**: Generating a random number to determine the player's movement.
- **Player Move**: Updating the player's position based on the dice roll.
- **Ladder Climb**: Moving the player to a higher position if they land at a ladder's base.
- **Snake Slide**: Sending the player to a lower position if they encounter a snake's head.
- **Win Condition**: Determining if the player has reached the final tile.

Verilog Implementation

The FSM is implemented in Verilog HDL. Each state is described with conditional logic to handle transitions based on inputs such as dice rolls and player positions. For example, if the dice roll results in a position containing a ladder's base, the FSM transitions to the Ladder Climb state. Outputs include the updated player position and game status.

Simulation and Verification

The FSM design is simulated and verified using Vivado, ensuring correct behavior under all scenarios. Test cases include:

1. Rolling the dice to move across the board.
2. Landing on a ladder or snake.
3. Reaching the final tile.

Python-Pygame Integration

A Python script reads FSM outputs from a text file that dynamically updates during gameplay. These outputs are used to update the graphical interface created in Pygame. The board, player position, and state changes are displayed in real-time, providing a visually engaging experience.

Results and Conclusion

Results

1. **Functional FSM**: The FSM successfully handles state transitions, ensuring smooth gameplay.
2. **Accurate Input Handling**: Dice rolls and board interactions are processed correctly, maintaining the game's integrity.
3. **Dynamic Visualization**: The Pygame interface effectively reflects real-time updates, enhancing player engagement.

Key achievements include:

- Smooth transitions between states, such as Idle to Dice Roll or Ladder Climb to Player Move.
- Accurate simulation of game rules, such as moving up a ladder or sliding down a snake.
- Real-time graphical updates synchronized with FSM outputs.

Conclusion

This project demonstrates the practical application of FSM concepts in designing interactive systems. By combining Verilog and Pygame, the system bridges the gap between hardware and software, showcasing the potential of FPGA-based designs in creating engaging applications. The project also enhances understanding of hardware description languages, simulation tools, and graphical interfaces.

Future Scope

The project provides a foundation for further development. Potential enhancements include:

1. **Multiplayer Support**: Expanding the FSM to handle multiple players, allowing for competitive gameplay.
2. **Improved Graphics**: Enhancing the Pygame interface with animations, sound effects, and a more polished design.
3. **Physical Deployment**: Implementing the design on an actual FPGA board, using physical buttons or switches for input.
4. **AI Integration**: Introducing an AI opponent to enable single-player challenges.

These enhancements would make the system more versatile and engaging, demonstrating the scalability of FPGA-based designs.