Dice Game

Hannah Smith, Nicolas Gomez,
Justin Phuc Le Nguyen

Verilog-Based Dice Game

- Overview
    - Introducing a digital rendition of a classic dice game implemented in Verilog
- Game Concept
    - Description: Players engage in a virtual two-player dice rolling contest
    - Objective: Roll dice, calculate scores, and enjoy a competitive gaming experience
- Implementation
    - Description: Utilizing Verilog, a hardware description language
    - Purpose: To simulate game logic, randomization, and visual display

Dice Game Overview

- Dice simulation
  - Four dice represented on-screen
  - Each player rolls two dice, resulting in unique values
- Score Tracking
  - Players score displayed alongside the dice
  - Scores update based on the total values of rolled dice
- Player Turns
  - Rolling trigger generates random values for each dice
- Score Calculation
  - Compare the sum of values from both players’ dice rolls
  - Scores update based on the total values of rolled dice

Dice Game Logic

- Inputs
  - clk: Clock signal, rst: Reset signal, roll_trigger: Trigger signal for dice rolling
- Outputs
  - die1 to die4: Outputs representing the values of four dice
  - player1score: Output for players 1’s score
  - player2score: Output for players 2’s score
- Score Calculation
  - Compares total values of dice pairs to determine score increments
  - Increases the score for the respective player based on the comparison
- Increment Control
  - Manages increment_done flag to ensure scores are updated only once per roll_trigger

Linear Feedback Shift Register (LFSR)

- Initialization
  - Resets the counter and sets initial values for random outputs on reset signals
- Roll Trigger
  - Incrementing counter on roll_trigger signal
  - Utilizes counter values for pseudo-random number generation
- Logic
  - Implements feedback logic for each output using counter values
  - Generates pseudo-random numbers using modular arithmetic

Seven Segment Display Module

- Resets the display and timers on reset signal
- Display Control
  - Utilizes digit_select to cycle through different displays
  - Maps input values to corresponding seven-segment representations
- Segment Output
  - Maps input and values to corresponding seven-segment representations
- Clock and Timer
  - Tracks and controls digit selection and display timing

VGA Display

- Input Mapping
  - Maps 4-bit input values to corresponding 7-segment display codes
- Display Data Generation
  - Combines the four 4-bit inputs into a 16-bit display data for VGA
- Visual Feedback
  - VGA display showcases dice values and scores
  - Real-time representation of the game state

Conclusion

- Game Logic and Functionality
  - Utilized verilog to simulate a two-player dice game
  - Modular design with components for dice simulation, score tracking, and VGA display
- Technical Implementation
  - Efficient utilization of clock cycles and trigger signals
  - VGA output for visually engaging gameplay representation
- Component Highlights
  - Seven-segment display for visual representation of dice values
  - LFSR generating pseudo-random numbers
  - VGA display module presenting dice and scores in real-time