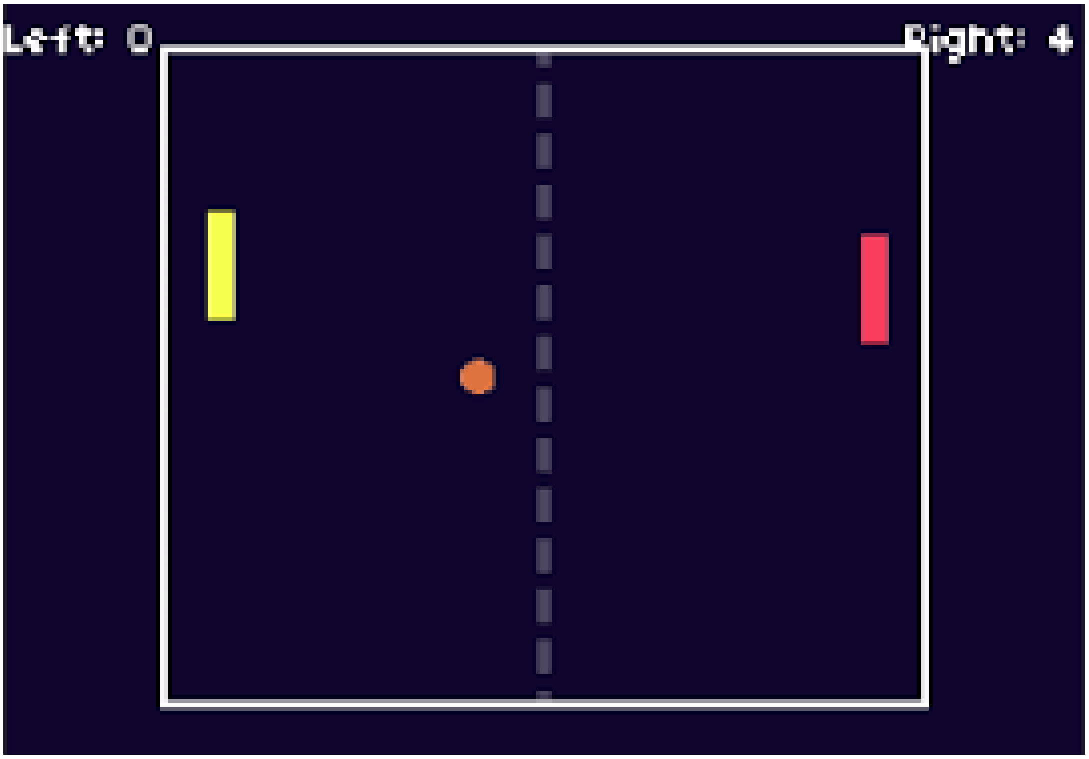
Pong Game Implementation on FPGA



Sara Hossam
Farida Ibrahim
Karim Abolghar

Abstract

This project tackles the well known game ‘pong’, and challenges us to recreate a version of the game using the theoretical material we learn throughout the semester. Using our knowledge of Verilog code, and new researched information, we create this game using a Basys3 board and a VGA output. This project is broken down into tasks that help us reach a final common goal of creating this game. After the completion of each task we work collectively to reflect our understanding as code and using hardware to output the final project.

Objective

To design, implement, and test the classic arcade game Pong on the Basys 3 FPGA board using Verilog HDL, providing a hands-on understanding of digital logic design and FPGA programming. The purpose of this project is to deepen your understanding of how digital circuits can be used to create interactive applications. [1]

[1] pong (2).pdf

Introduction

Pong is one of the earliest video games, where two paddles compete to bounce a ball back and forth. Pong was significant as an early video game as it helped to establish the video game industry.[2] Field-Programmable Gate Arrays (FPGAs) and Hardware Description Languages (HDLs) like Verilog play a crucial role in digital design for several reasons like their flexibility and customization. They can be programmed to perform different tasks based on one's objective. The purpose of this project is to help us learn the usage of such technology and their real world application through a practical point of view. The intertwining of this kind of hardware and software coordinates fully functional designs that can serve a purpose and provide solutions to real life problems.

The project focuses on implementing the classic Pong game on the FPGA using Verilog. By implementing Pong we are able to demonstrate the use of the fundamental principles such as clock dividers, finite state machines and signal processing to create a project that uses these building blocks for more complex ideas. Moreover, the Basys 3 board provides more than enough features to ensure the correct functioning of our game and controller. It provides the correct pins and hardware required for the game to run with the desired features and expected output. This project aims to strengthen our theoretical knowledge and also reinforce our creativity and the potential of the FPGA.

[2] Pong | Video Game, Arcade, Atari | Britannica

Project Overview

The project implements the game of Pong in Verilog, it divides the game into modules that handle game logic, FPGA hardware and user interaction. Some of the primary modules include:

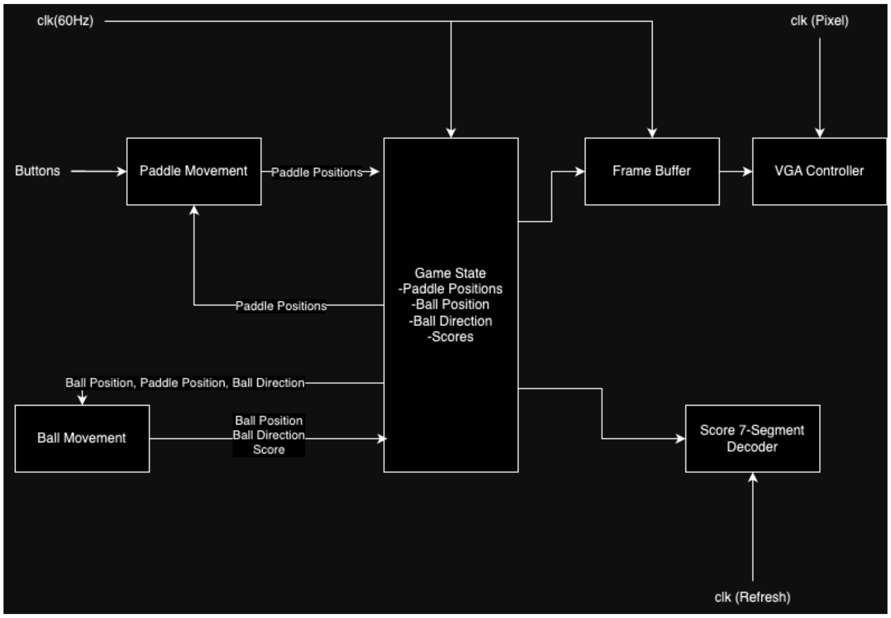
- ➔ Paddle Control: Use switches or buttons on the Basys 3 board to control the movement of two paddles.
- ➔ Ball Movement: Program the logic to control the ball's movement across the screen. The ball should bounce off paddles and screen edges .
- ➔ Collision Detection: Implement collision detection for the ball and paddles, as well as screen top and bottom boundaries.
- ➔ Score Keeping: Keep track of the score for each player. Display the scores on the Basys 3's 7-segment displays.
- ➔ Display Output: Use the VGA output of the Basys 3 to display the game on a monitor, where the paddles, ball, and score will be visualized.[3]

Game Goal

Players play a tennis-esque game controlling two paddles to bounce a ball back and forth trying to direct the ball past the other player's paddle. A player gains a score point when the opponent fails to return the ball.

[3] pong (2).pdf

Block Diagram



Module Descriptions

- **BallMovement:** Controls the movement, position and bounce logic of the ball.
- **PaddleMovement:** Handles the movement of the paddles using the user input through the FPGA push buttons.
- **GameState:** Manages the start, playing and end states of the game.
- **VGA Controller:** Used to generate the VGA signals for the game to display.
- **ScoreController:** Tracks score of both players and the winner.

Implementation:

Hardware and Tools Used

- Basys 3 FPGA Board
- Verilog HDL
- VGA Display
- 7-Segment Display

Verilog Code Overview

The code consists of different modules that all work to perform a specific functionality of the entire program. They are all interdependent and are used to complete each other's purpose.

BallMovement: The purpose of this module is to handle the ball's position while the game is running. It calculates the position based on the current velocity and updates the ball's direction due to collisions. Key features of this module include:

a. Responsible for the ball bouncing off of the paddles
b. Adjusts the direction of the ball and position after it hits a player or the wall.
c. Detects scoring conditions for either player.

PaddleMovement: Manages the user’s inputs and provides the movement of the paddle as feedback. The module supports the following features:

d. Reading input signals from the FPGAs push buttons to determine the direction of the paddle’s movement.
e. Updating the paddle’s location and ensuring they remain within the screen boundaries.
f. Synchronizes the paddle with the game’s clock.

GameState: The implementation of the finite state machine that controls the game. It handles functions such as:

g. Determining current state of the game (start, play, end)
h. Monitors and keeps track of the score to ensure the end state is reached.
i. Handles the ability to reset the game to restart.

VGA Controller: This module allows us to display the game’s output onto the monitor through VGA signals. The module’s most important features are:

j. Generating vertical and horizontal synchronization signals.
k. Mapping the game elements to the screen (paddle, ball, score)
l. Maintaining the correct refresh rate throughout the game’s runtime.

FrameBuffer: The purpose of this module is to maintain the current frame’s pixel data for the VGA output. Key features of this module:

m. Stores information about the game objects.

n. Updates the display buffer in sync with the VGA controller.

ScoreController: This module’s main function is to track and display the scores of each player. Other functions include:

o. Increments the score when a player wins a rally.

p. Triggers the end state when the limit for the score is met.

q. Displays the score on the FPGAs 7-segment display as well as the monitor.

ClockGeneration: Provides the clock signal for different parts of the system. It features functions like:

r. Generates a slower version of the FPGAs clock for the game logic.

s. Synchronizes various modules together to ensure a smooth operation.

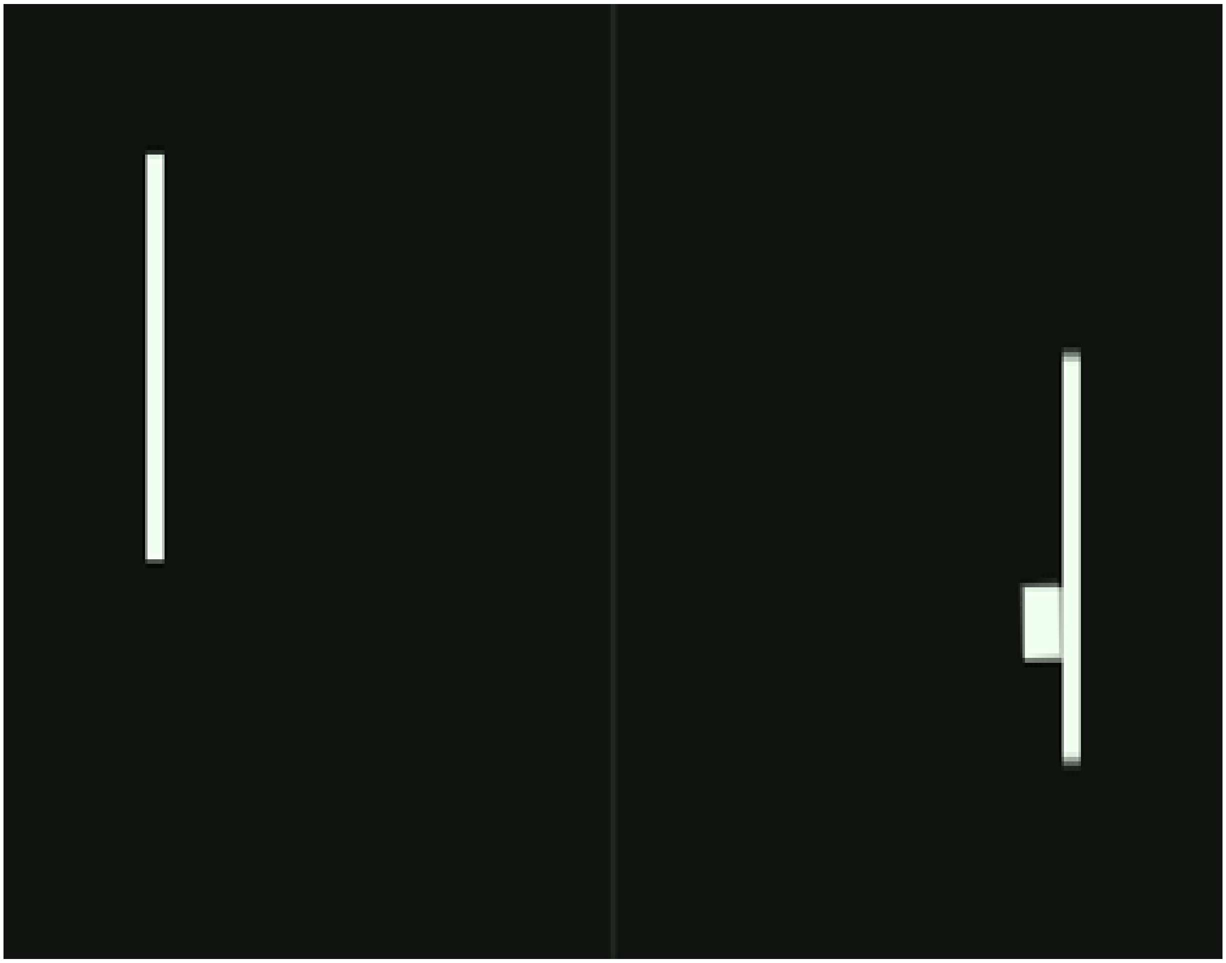
t. Provides separate clocks for game logic and VGA synchronization.

Pong: The top-module that gathers the entire “circuit” into one whole. It allows for the system to run correctly and integrates them. Some of the important features include:

u. Connect all inputs and outputs together.

v. Make sure interactions between the different modules are orchestrated.

PongConstraint: This is the constraint file that maps the different signals to their hardware pins. All the switches, VGA signals and push buttons are mapped using this file.



Results and Testing

VGA Output Display

For the output to be displayed successfully, paddle movement was achieved allowing the user to control the paddle using push buttons. A Ball bouncing back and forth with accurate rebound calculations was created to ensure gameplay. Additionally, we created an accurate collision detection to ensure a score is incremented when a collision between the paddle and the ball does not take place. The VGA output correctly aligns with gameplay elements, providing an entertaining and enjoyable experience for the user. The output includes two paddles, a scoring system, a ball, and the constraints of the walls. The game output displays all the elements that make up the game and help the user to feel engaged in the game due to the low delay between the board and the corresponding output on the monitor.

Score Keeping

The score was implemented, updating as either player meets the condition of bouncing the ball past the other paddle. It is reflected on both the 7-segment display and the monitor through the VGA.

Challenges and Solutions

During this project, there were a few complications that we met along the way and had to navigate our way around in order to present a complete final product. This project was our first time implementing a game on an FPGA, which is why we chose to use Verilog, as it was the language we were taught. The first problem that we encountered was having to understand how the game components should be represented and which logic should be responsible for controlling what parts of the game. In order to solve this problem, we had to break down the game into smaller modules and allocate responsibilities accordingly. This was not an easy task as some components, such as collision detection and VGA display control, were new to us and we had not covered their usage in depth. A clear example of a component we struggled implementing was the VGA controller. This component was unfamiliar to us and required a lot of research and troubleshooting to understand how to use it with the Basys 3 FPGA board to display the game correctly. The same can be said for the logic required to handle the ball's movement and bouncing, which we found to be more complex than what we had expected. We chose to utilize external resources, such as FPGA documentation and online forums, to help us understand how to implement these components and use them effectively in our final design. Another problem we faced was time management, as we had to meet various milestones while working continuously on the project. We tackled this problem by dividing our work into smaller tasks and assigning responsibilities among the group members, ensuring that everyone could work on different parts of the project simultaneously. This project was a new experience that taught us a great deal and helped us understand how to apply theoretical knowledge to real-world hardware through Verilog programming.

Conclusion

This project allowed us to use theoretical material that we have learnt during the semester in a real life application. Practical projects expose us to new challenges that we overcome and learn from, which is a different form of learning and growing as students compared to theoretical material. This project allowed us to take a well known game that established the gaming industry and recreate it with our own skills and understanding of the concepts. The final project is a recreation of the pong game, which eventually helped us understand Verilog programming and how a user can interact with an interface via a board. We learnt a new coding language and how to use a VGA to display our outputs, which was something not covered in our theoretical material. Learning from this project and gaining experience, there are a few improvements to be made for future extensions of this project. We would use two controllers or boards instead of one to enhance the gaming experience for the user, where two controllers would be connected to the VGA output. Additionally, we would add some more game features to make the game more enjoyable and entertaining for the user. In conclusion, this enjoyable project opened our eyes to the real world application of what we learn, and through hands-on experience, we were able to develop more skills and understand our material better.

References

Pong | Video Game, Arcade, Atari | Britannica
pong (2).pdf