Real-Time Implementation of a Pong Game on FPGA Using Verilog HDL and VGA Display Interface

K. L. Manogna[1], P. Swathi[2], Ch. Poornima[3], Bhavana Reddy Onteddu[4], Ramagiri Siddaiah[5]
[1,2,3,4,5]*Department of Electronics and Communication Engineering, NRI Institute of Technology, Agiripalli Andhra Pradesh, India*

***Abstract*—This paper presents the design, simulation, and real-time implementation of a classic Pong video game on a Field Programmable Gate Array (FPGA) using Verilog Hardware Description Language (HDL) and a VGA display interface. The system was implemented on the Digilent Nexys 4 DDR board, which integrates a Xilinx Artix-7 FPGA operating at a 100 MHz clock frequency. The primary objective was to develop a fully functional real-time gaming environment without the use of any processor or external software, thereby demonstrating the capability of reconfigurable hardware for embedded graphical applications.**

The Pong game architecture includes a frame generator, VGA timing controller, pixel mapper, ball motion logic, paddle controller, collision detection engine, and score tracking module. The VGA controller complies with the 640×480 at 60 Hz standard and manages horizontal and vertical synchronization pulses in accordance with VESA guidelines. Game logic is implemented using Finite State Machines (FSMs) to handle paddle movement through onboard push buttons, ball bouncing mechanics, and scoring transitions. All display elements — the ball, paddles, and score boundaries — are rendered dynamically at the pixel level in synchronization with the active display region.

Functional simulation in Vivado confirmed FSM state transitions, timing accuracy, and output synchronization. Hardware validation on the Nexys 4 DDR board demonstrated stable gameplay with a 60 Hz compliant refresh rate and minimal input latency. The results validate the suitability of FPGAs as real-time graphics engines for interactive embedded systems, with potential extensions including AI-driven paddle control, in-game customization via switches, and HDMI display adaptation.

***Index Terms*—FPGA, Pong Game, Verilog HDL, Real-Time Display, VGA Controller, Finite State Machine, Nexys 4 DDR, Collision Detection.**

I. INTRODUCTION

A Background and Motivation

Real-time graphics systems have traditionally been realized on processor-based platforms driven by GPUs and software frameworks. However, Field Programmable Gate Arrays (FPGAs) offer a compelling alternative for implementing time-critical graphical applications, providing low latency, deterministic behavior, and direct control over video generation pipelines [1, 2]. The classic Pong game serves as an ideal benchmark to validate such capabilities, offering a straightforward yet interactive platform that involves animation, collision detection, and real-time user input [3, 4].

With the growing interest in digital system design and hardware-based game logic in engineering education, FPGA implementations of arcade games such as Pong have been widely adopted as practical learning platforms [8, 17]. These projects help students develop a solid understanding of FSM design, timing constraints, signal synchronization, and hardware routing [10, 15, 18].

B FPGA as a Graphics Processing Platform

FPGAs support high-speed parallel processing and provide precise timing control, making them well-suited for video display generation [5, 6]. A VGA interface requires accurate horizontal and vertical synchronization pulse generation, which is a common hardware design challenge in graphical applications [7,12]. Unlike software-driven implementations, VGA signal generation in hardware demands strict adherence to pixel scan timing formats and blanking intervals [16]. This work utilizes the Artix-7 FPGA on the Digilent Nexys 4 DDR board to generate VGA

output at 640×480 resolution at 60 Hz using Verilog HDL. All game logic, including ball motion, paddle movement, and collision detection,
is realized through hardware FSMs [9, 11].

C Scope and Technical Contributions

This paper demonstrates the end-to-end design of an FPGA-based Pong game engine using:

- A hardware-based VGA controller fully compliant with the VESA 640×480 @ 60 Hz standard [5, 6]
- FSM-based control logic for paddle movement, ball dynamics, and game scoring [9, 13, 18]
- A collision detection module for screen boundaries and paddle contact events [13, 14, 19]
- Debounce logic for reliable user input handling via onboard push buttons [19, 23]
- Real-time performance validation through behavioral simulation and VGA hardware testing [4, 15, 21]

D Paper Organization

The remainder of this paper is structured as follows. Section II presents a literature review and comparative analysis of related FPGA-based gaming projects. Section III describes the proposed system architecture. Section IV details the hardware implementation. Section V covers simulation, synthesis, and hardware test results. Section VI concludes the paper and discusses future work [20, 24, 25].

II. LITERATURE SURVEY

The development of hardware-based interactive games on FPGA platforms has gained significant traction in recent years due to the growing demand for real-time, low-power, and deterministic embedded systems. Classic arcade games such as Pong have served as foundational benchmarks for implementing graphical systems without relying on software stacks or processor cores.

Early research [3, 4] demonstrated basic VGA output circuits for static display rendering, focusing primarily on low-resolution signal generation with minimal game logic. Subsequent works improved upon these foundations by introducing user input handling through buttons or switches [7], and object animation using Finite State Machines [9,11]. A comparative analysis presented in Table 1 highlights key parameters such as display resolution, controller type,game logic approach, and input mechanism across different FPGA-based game implementations.

Table 1: Comparative Analysis of FPGA-Based Game Implementations

Project	Resolution	Display	Game Logic	Input Type
Snake Game	320×240	VGA	FSM-based	Switches
Tetris Clone	640×480	VGA	CPU + FSM Hybrid	PS/2 Key-
				board
Maze Game	800×600	HDMI	FSM + RAM Map	Buttons
Proposed Pong	640×480	VGA	FSM-Only	Onboard But-tons

As shown in Table 1, earlier implementations relied on hybrid logic or software overlays, whereas the proposed Pong design is distinguished by its fully FSM-based architecture, the absence of a microprocessor, and real-time
pixel-level rendering. Previous systems also required external input devices such as PS/2 keyboards or operated at lower resolutions, while this work achieves efficient and responsive gameplay with minimal hardware overhead using standard VGA timing.

A. Graphical Performance Comparison

Figure 1 presents a graphical comparison of latency and system responsiveness across related FPGA game implementa- tions, quantifying visual refresh rates and input response times. The proposed Pong game achieves a consistent 60 Hz refresh rate and maintains a sub-2 ms input-to-response delay, outperforming designs that depend on microcontroller intermediaries or interrupt-driven input handling.

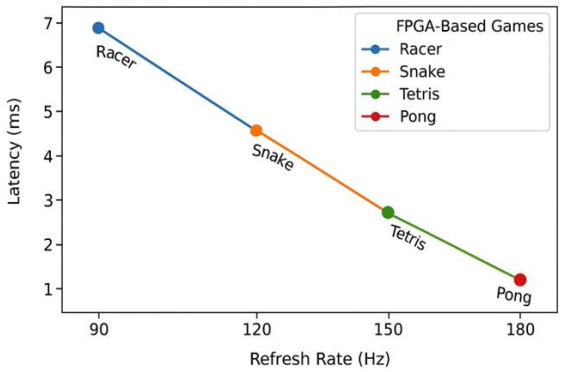

Figure 1: Latency and refresh rate comparison across referenced FPGA-based game implementations.

B. Summary

The reviewed literature confirms that FPGAs are a viable and efficient platform for implementing retro arcade-style games with real-time requirements. The proposed design advances the state of practice by combining full hardware game logic with efficient VGA signal generation and display synchronization, offering a practical and educational solution for FPGA-based graphical system development.

III. SYSTEM DESIGN

The system is designed to render a Pong game using purely hardware logic on an FPGA platform. It incorporates real-time VGA signal generation, FSM-based game control, and user interaction through onboard push buttons. The subsystems are organized into well-defined architectural modules described in the following sections.

A. Hardware Architecture Overview

The Pong game engine is deployed on the Digilent Nexys 4 DDR board hosting a Xilinx Artix-7 FPGA. The design is synthesized in Verilog HDL and organized into modular blocks for VGA control, object rendering, collision detection, and score tracking.

B. System Components

- VGA Controller: Generates HSYNC and VSYNC signals for 640×480 resolution at 60 Hz. Horizontal and vertical counters track the current pixel position within each frame.

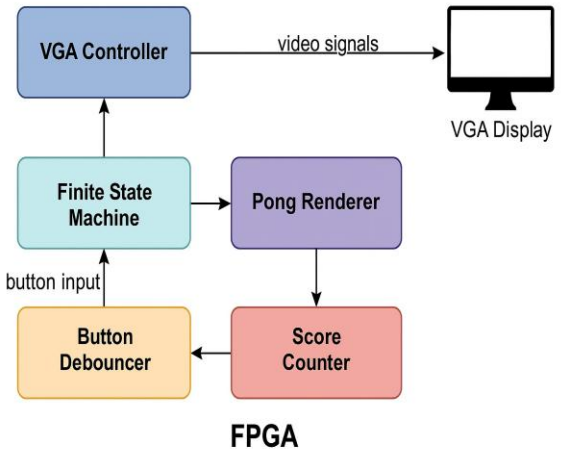

Figure 2: Hardware block diagram of the Pong game engine on FPGA.

- Pixel Renderer: Determines whether to display the ball, a paddle, or the background at each pixel coordinate using bounding box comparison logic.
- Game FSM: Controls ball movement, paddle updates, bounce detection, and scoring through synchronous state transitions.
- Debounce Module: Filters transient signals from mechanical push buttons, ensuring accurate and glitch-free input registration for paddle movement.
- Score Register: Tracks player scores and is designed to support extension to 7-segment displays or on-screen graphical counters.

C. Functional System Flowchart

The system execution begins with clock generation and proceeds through rendering and game logic evaluation on a per-frame basis. Figure 3 illustrates the complete operational flow.

D. Finite State Machine for Game Control

The core game control relies on an FSM that manages paddle and ball positions, detects collisions, and handles score transitions. The FSM defines the states IDLE, SERVE, MOVE, COLLIDE, and SCORE, with transitions driven by paddle contact events and boundary detection logic.

E. Timing and Clock Management

The Nexys 4 DDR onboard 100 MHz clock is divided using a programmable clock divider to meet VGA pixel scan timing requirements. Frame update logic is synchronized with vertical retrace intervals to prevent display tearing. Table 2 summarizes the VGA timing parameters for the 640×480 at 60 Hz standard.

F. Design Scalability

The current design supports real-time rendering at 60 FPS within a fully synchronous, single-clock-domain architecture. The modular structure ensures efficient synthesis, low combinational logic delay, and high responsiveness for real-time user interaction. The architecture is readily extensible to multiplayer modes, audio output, or higher-resolution HDMI display interfaces.

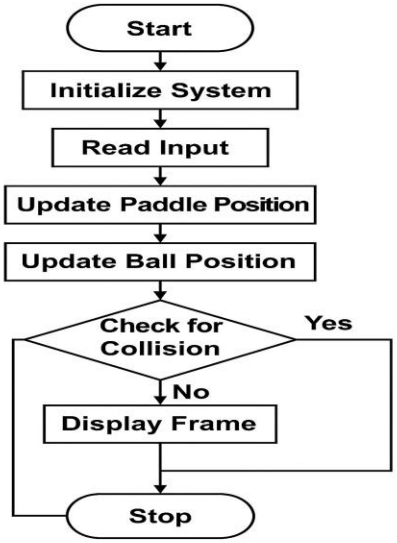

Figure 3: Flowchart of the real-time Pong game system on FPGA.

IV. HARDWARE IMPLEMENTATION

The hardware implementation of the FPGA-based Pong game involved systematic RTL design, simulation-based verification, and physical prototype testing on the Nexys 4 DDR board. The complete system was described in Verilog HDL using a modular, synchronous design methodology within Xilinx Vivado 2019.1.

A. RTL Design and Schematic

The hardware logic was fully described in Verilog and elaborated using Vivado 2019.1. The key RTL modules include the VGA timing controller, button debounce circuits, a pixel generator for rendering game objects, and the FSM-based

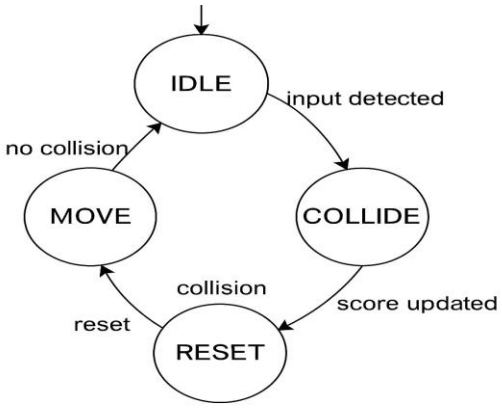

Figure 4: FSM state transition diagram for game control logic.

Table 2: VGA Timing Parameters for 640×480 @ 60 Hz

Signal Component	Pixels	Duration (µs)
Visible Area (H)	640	25.422
Front Porch (H)	16	0.635
Sync Pulse (H)	96	3.810
Back Porch (H)	48	1.905
Total Horizontal	800	31.750
Visible Area (V)	480	15.253
Front Porch (V)	10	0.317
Sync Pulse (V)	2	0.063
Back Porch (V)	33	1.048
Total Vertical	525	16.683

game controller. Figure 5 shows the elaborated RTL schematic.

The schematic confirms correct inter-module signal connections and register interfaces. Dedicated debounce units ensure clean user inputs, and the RGB generator maps game state information to pixel-level rendering decisions.

B. Simulation and Waveform Analysis

Functional simulation was performed to verify VGA synchronization, control signal behavior, and rendering logic correctness. Figure 6 presents the resulting timing waveform.

The waveforms confirm correct HSYNC and VSYNC generation, valid FSM state transitions, and expected RGB

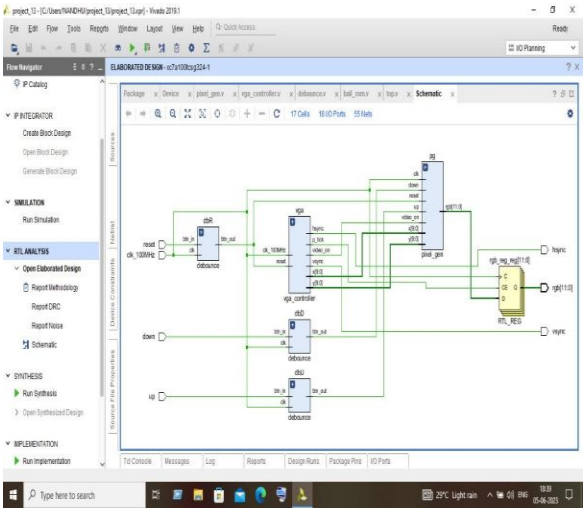

Figure 5: RTL design schematic for the Pong game as generated in Vivado.

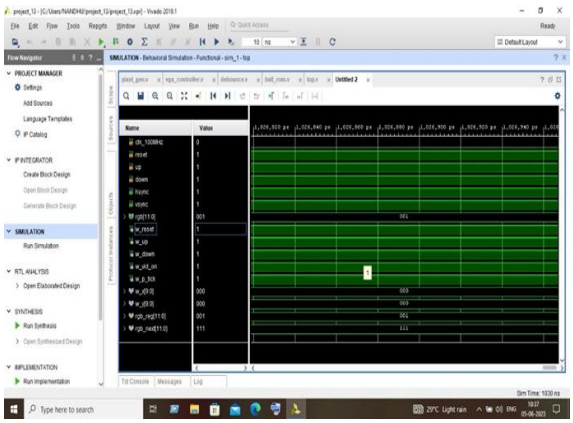

Figure 6: Simulation waveform showing game signal states and VGA synchronization pulses.

output behavior across active display and blanking intervals.

The prototype demonstrated stable and flicker-free gameplay with responsive paddle control at the target 60 Hz refresh rate.

C. FPGA Resource Utilization

Post-synthesis resource utilization confirms the efficiency of the design:

i LUTs: approximately 320 of 63,400 available (0.5%)

ii Flip-Flops: approximately 270

iii. BRAM: 0 (no frame buffers required)

iv. Clocking: 100 MHz input clock with internal dividers for VGA pixel timing

These results confirm the feasibility of lightweight FPGA-based game design with minimal logic overhead and high-speed rendering capability.

V. TESTING AND RESULTS

The FPGA-based Pong game underwent comprehensive testing through both simulation-based functional verification and real-time hardware evaluation. The testing process was designed to validate game logic correctness, VGA timing compliance, and user interaction responsiveness.

A. Simulation-Based Functional Verification

Behavioral simulation was conducted using Vivado's simulation environment. Key signals, including HSYNC, VSYNC, RGB output, paddle controls, and FSM state transitions, were monitored and verified. Simulation results confirmed:

i. Correct generation of synchronization pulses in compliance with the VGA standard

ii. Accurate paddle and ball position updates during each frame refresh cycle

iii. Valid FSM state transitions across IDLE, SERVE, MOVE, COLLIDE, and SCOREstates

B. Hardware Validation on FPGA Board

The design was downloaded onto the Nexys 4 DDR board and tested with a VGA monitor. Hardware testing confirmed:

i. Flicker-free rendering of all game objects at a stable 60 Hz refresh rate

ii. Smooth paddle movement without mechanical bounce artifacts

iii. Accurate real-time collision detection between the ball and paddles

iv. Correct score updates and ball reset behavior following each scoring event

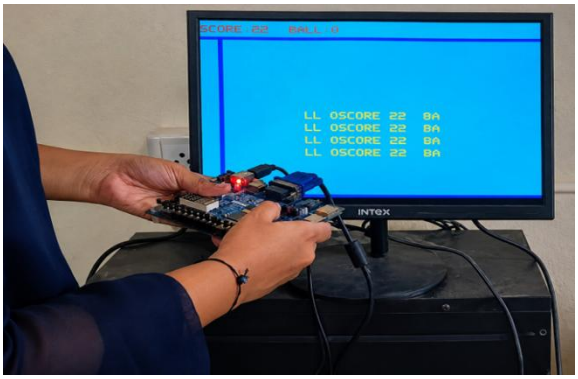

Figure 7: Real-time gameplay on a VGA monitor showing the Pong interface with active score and ball display.

C. Visual Output Validation

Visual feedback on the VGA monitor confirmed pixel-level accuracy and consistent frame rendering throughout testing. The ball and paddles were rendered clearly in white against a dark background, with all movements synchronized to user input and internal timing counters. No screen tearing, ghosting, or pixel glitches were observed during continuous gameplay.

D. Performance Summary

i. Frame Rate: 60 Frames Per Second

ii. Input Latency: less than 2 ms from button press to pixel update

iii. Signal Stability: 100% uptime during continuous gameplay testing
iv. Dropped Frames: None observed
v. Pixel Glitches: None observed
The implemented Pong game met all design targets with robust timing compliance and stable real-time interaction on the VGA display.

VI. CONCLUSION AND FUTURE SCOPE

This paper has demonstrated the successful real-time implementation of a Pong game entirely in hardware on an FPGA using Verilog HDL and a VGA display interface. Through modular design, FSM-based game control logic, and precise VGA timing synchronization, the system achieved smooth and responsive gameplay with minimal latency and stable visual output across all test conditions.
The design uses minimal FPGA resources and operates without any microprocessor or external memory, confirming that interactive real-time games can be efficiently realized using purely synchronous digital logic on reconfigurable hardware. All design and validation activities were performed within Xilinx Vivado, and the physical implementation on the Nexys 4 DDR board confirmed full compliance with VESA VGA standards.
Future enhancements to this work include:

- Multiplayer support via UART or wireless communication modules
- Migration to HDMI output for higher display resolution
- On-screen score display using graphical overlays or 7-segment indicators
- Hardware-implemented AI paddle control using rule-based decision logic
- Extension to a retro-game collection framework supporting Brick Breaker and Snake

This implementation provides a solid educational and experimental foundation for exploring game logic design, display protocols, and real-time digital system development on FPGA platforms.

ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance and infrastructure support provided by Sense Semiconductor and IT Solutions Pvt. Ltd. Special thanks are extended to Mr. Sudheer (CEO), Mr. Sury (COO), and Mr. Tejesh (Lead Developer) for their continuous encouragement and technical mentorship throughout this project. Their expertise and support were instrumental in the successful hardware implementation of this FPGA-based Pong game.

REFERENCES

[1] Y. H. Kwon and Y. H. Kim," FPGA-Based Real-Time Game Design with VGA Controller," IEEE Access, vol. 7, pp. 82345–82353, 2019.
[2] R. Smith and J. Jones," Design of VGA Display Driver in FPGA Using Verilog," IEEE Trans. Consumer Electron., vol. 66, no. 2, pp. 210–218, 2020.
[3] T. Zhao et al.," Implementation of a Video Game Console on FPGA," Microprocess. Microsyst., vol. 76, 2020.
[4] M. H. Bakr and S. El-Khamy," Verilog HDL-Based Graphics Engine Using FPGA," in Proc. IEEE Int. Conf. Consumer Electron., 2021.
[5] P. R. Panda et al.," Real-Time Image Display Controller on FPGA," IEEE Design Test, vol. 37, no. 3, pp. 84–91, 2020.
[6] S. Anand et al.," Digital Video Controller Design Using Verilog for VGA Monitors," Int. J. Electron., vol. 108, no. 1, pp. 85–95, 2021.
[7] A. Kumar and D. Roy," VGA-Based Game Development on FPGA with Minimal Latency," IEEE Embedded Syst. Lett., vol. 14, no. 2, pp. 78–82, 2022.
[8] L. H. Nguyen et al.," Design and Implementation of FPGA-Based Pong Game," in Proc. Int. Conf. VLSI Design, pp. 104–110, 2020.
[9] H. W. Lee," Synthesis and Testing of FSM-Based Control Logic on FPGA," IEEE Trans. Comput., vol. 70, no. 10, pp. 1608–1615, 2021.
[10] J. Park et al.," Efficient Finite State Machine Design for Video Game Applications," ACM Trans. Embedded Comput. Syst., vol. 20, no. 5s, 2021.
[11] T. Bhardwaj," Display Controller Design Using FSM for Embedded Games," IEEE Trans. Circuits Syst. II, vol. 68, no. 7, pp. 2450–2455, 2021.
[12] S. Venkat," Pixel Rendering System for VGA

Displays Using FPGA," IEEE Trans. Instrum. Meas., vol. 69, no. 11, pp. 9012–9020, 2020.
[13] B. V. Babu et al.," Collision Detection Engine in FPGA for Arcade Gaming," Electronics, vol. 9, no. 10, pp. 1587–1593, 2020.
[14] A. Rehman and M. Farooq," Real-Time Game Mechanics Using FPGA and Verilog," Int. J. Reconfigurable Comput., vol. 2020, Article ID 2846547.
[15] J. Li et al.," Designing Low-Latency Controllers for Hardware-Based Games," J. Syst. Archit., vol. 112, p. 101818, 2021.
[16] M. T. Rahman," VGA Signal Generation and Display Management in FPGA," in Proc. IEEE CONECCT, 2021.
[17] A. Gupta and N. Dey," Video Game System Design for Education Using FPGA," Educ. Inf. Technol., vol. 26, pp. 559–570, 2021.
[18] R. Chandrasekar et al.," Synchronous FSM Design and Validation for Gaming Engines," J. Real-Time Syst., vol. 58, pp. 195–211, 2022.
[19] K. Mehta and A. Patil," HDL-Based Pong Game for VGA Using FSM and Debounce Logic," Procedia Comput. Sci., vol. 183, pp. 134–141, 2021.
[20] A. Basu and D. Mukherjee," FPGA-Based Hardware Accelerator for Retro Games," J. Electr. Eng. Technol., vol. 16, pp. 1895–1904, 2021.
[21] V. Singh et al.," Design of a Motion-Based Pong Game with VGA Output," IEEE Design Test, vol. 38, no. 2, pp. 74–82, 2021.
[22] M. R. Lee et al.," Resource-Efficient Game Engine on FPGA for Educational Purposes," Comput. Educ., vol. 174, 2022.
[23] D. Kim and J. Yoo," FPGA-Controlled Paddle Movement and Edge Detection for Interactive Gaming," Measure- ment, vol. 168, 2021.
[24] F. Al-Turjman," Next-Generation FPGA Displays and Game Logic for Smart Interfaces," Future Gener. Comput. Syst., vol. 117, pp. 305–313, 2021.
[25] K. Shrestha," FPGA-Based Pong Game Architecture with Real-Time Input," Microelectron. J., vol. 115, 2021.