# Project Report: Pong Game for FPGA

Yahia Kilany, Mahmoud Refaie, Seifeldin Elwan
CSCE 2301: Digital Design I
Dr. Dina Mahmoud
Department of Computer Science and Engineering, The American University in Cairo

December 8, 2024

**Overview**

The design implements a digital Pong game system using an FPGA. The system interfaces with a VGA monitor to display the game, incorporates player controls via buttons, and generates audio for sound effects. It includes score tracking displayed on a 7-segment display. Key modules were designed for managing VGA signals, paddle and ball movement, collision detection, scoring, and sound generation.

**Modules and Their Functions**

1. **Top Module**
   - **Purpose:** Serves as the main integrator for all components.
   - **Inputs:**
     - *`clk_100MHz`*: System clock from the Basys 3 FPGA board.
     - *`Reset`*: Reset signal.
     - *`up1`, `down1`, `up2`, `down2`*: Player paddle control inputs.
   - **Outputs:**
     - *`hsync`, `vsync`*: VGA synchronization signals.
     - *`rgb`*: VGA pixel colors.
     - *`seg`, `an`*: Signals for 7-segment display.
     - `audio`: PWM signal for audio output.
2. **VGA Controller (*`vga_controller`*)**
   - Generates VGA timing signals (*`hsync`, `vsync`*) and coordinates *(`x`, `y`)* for rendering on the screen.
   - Outputs a pixel tick signal *(`p_tick`)* used for updating graphics.
3. **Paddle Logic *(`Paddle`)***
   - Tracks paddle positions and boundaries.
   - Detects when a paddle is active based on player inputs.
   - Outputs paddle boundaries and activation signals.
4. **Ball Logic (*`ball`*)**
   - Simulates the ball's movement and interactions with paddles and walls.
   - Detects collisions, scoring events, and outputs `ball_on` signal for rendering.

5. **Score Module (*`Score`*)**
    - Maintains the scores for Player 1 and Player 2.
    - Tracks unit digits for display on the 7-segment display.
6. **Score Text Renderer (*`Score_text`*)**
    - Converts score values into ASCII bits for displaying on the VGA screen.
7. **Color Multiplexer (*`color_mux`*)**
    - Determines the RGB color output based on active objects (paddles, ball, or text) and whether the VGA signal is on.
8. **7-Segment Display Controller (*`score_7seg`*)**
    - Drives the 7-segment display to show scores in decimal format.
    - Handles multiplexing for the 4-digit display.
9. **Debouncer (*`debouncer`*)**
    - Debounces player button inputs to ensure stable signals for paddle controls.
10. **Sound Module (*`sound`*)**
    - Generates audio signals for paddle hits, wall hits, and scoring events.
    - Uses PWM to output sound through the audio pin.

**Design Challenges**

1. **Synchronization Issues:**
    - Initially, the score kept incrementing erroneously due to incorrect logic. This issue was mitigated by ensuring that score calculation only occurred on the posedge of the scoring signals, synchronizing it properly with game events.
2. **Collision Detection:**
    - Designing the collision detection logic required trial and error to fine-tune the conditions that control ball movement. Adjustments were made iteratively to ensure accurate paddle and wall interactions.
3. **Sound Integration:**
    - Generating distinct sound effects for different game events while minimizing noise required tuning the PWM signal and managing reset synchronization.

**Future Enhancements**

- Add more customizable game settings such as ball speed and paddle size.
- Implement dynamic colors or themes for the game objects.

- Add a win screen when a player reaches a score of  9.

This design was developed with modularity and clarity in mind, allowing for straightforward debugging and scalability. Each module is designed to handle a specific task, such as controlling the paddles, managing colors, or tracking scores, etc.

Contributions:

**Yahia Kilany:** Developed **Paddle.v** (paddle control and display), **ball.v** (ball movement, collision), **Score.v** (score calculation), and **score_7seg.v** (7-segment display score handling).

**Seifeldin Elwan:** Developed **score_text.v** (display score on VGA screen) and ball scoring logic.

**Mahmoud Refaie:** Developed **sound.v** and authored the report.