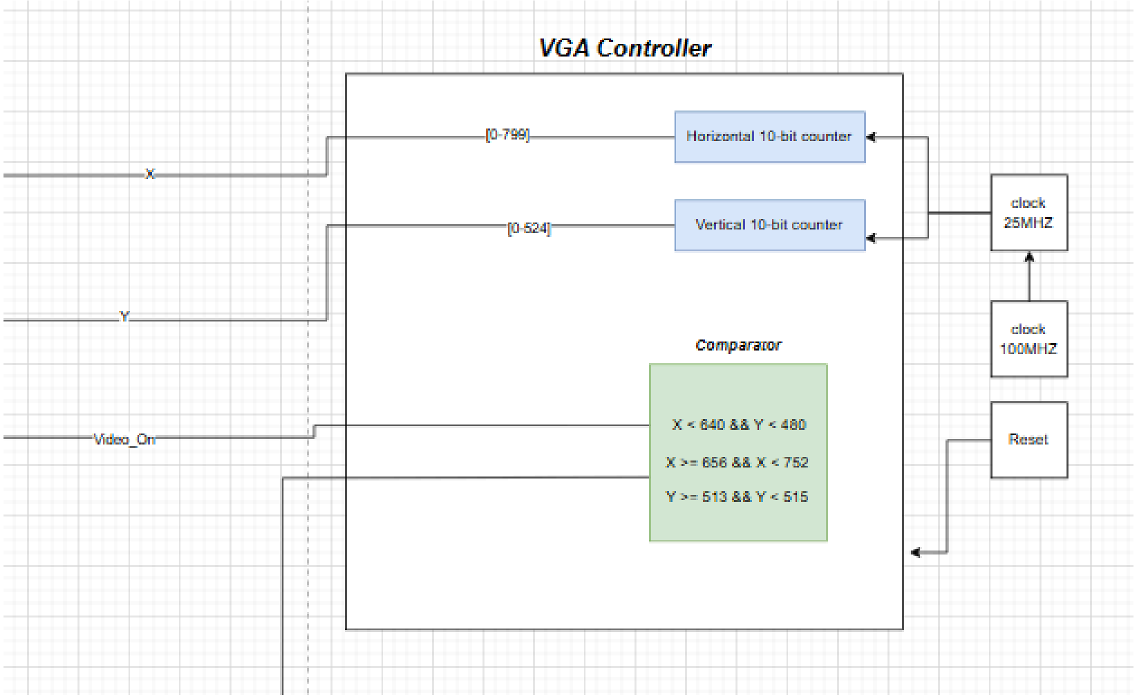
PONG GAME - Project report

GitHub repository: https://github.com/mahmoudhossamws/Pong_Game
Logic diagram:
https://drive.google.com/file/d/1XH2eaCGtwWvvsEmzvkRFTHJGWZ0WUjuV/view?usp=sharing

Design Approach:
the project consists of two main components: the VGA controller and the pixel generation circuit.

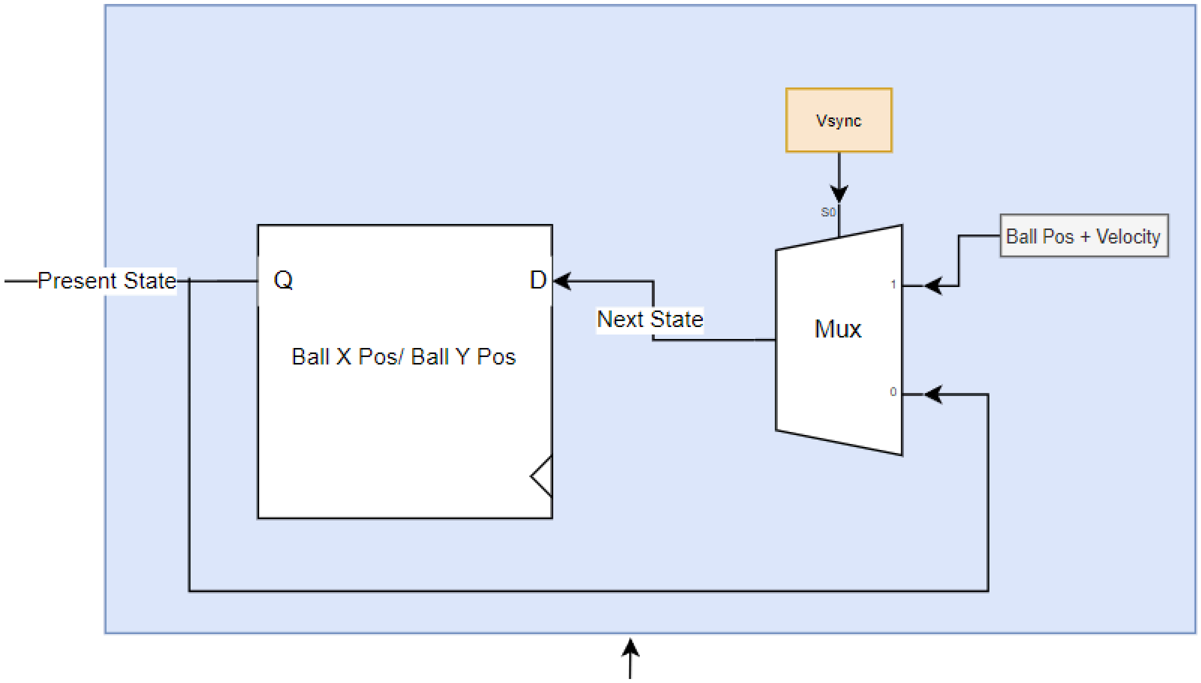
VGA controller:
As shown in the logic diagram Simply the VGA controller works by counting till the correct point in the screen to send the Vsync and Hsync signals to the screen. The VGA controller knows the right moment by comparing the count of x and y coordinates to the display dimensions.



Pixel Generation circuit:
The pixel generation circuit is responsible for generating the content displayed on the screen by determining which color should be displayed and sent to the RGB at every single moment. It uses parameters for the area where every graphical element exists in the display and compares the current count for x and y coordinates with this parameter to determine if this element is currently on. The pixel generation circuit consists of ball logic, paddle logic, collection detection, and score logic.

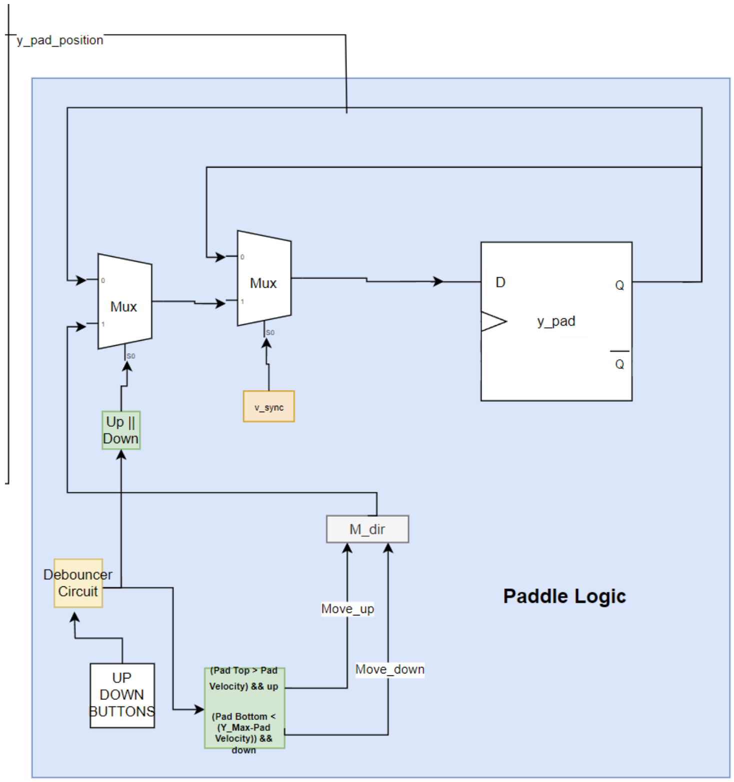
Ball logic:
The ball logic determines the ball's current position by either keeping it the same if there was no refresh tick or changing it to be adjusted with the ball velocity as shown in the diagram.

Ball Logic

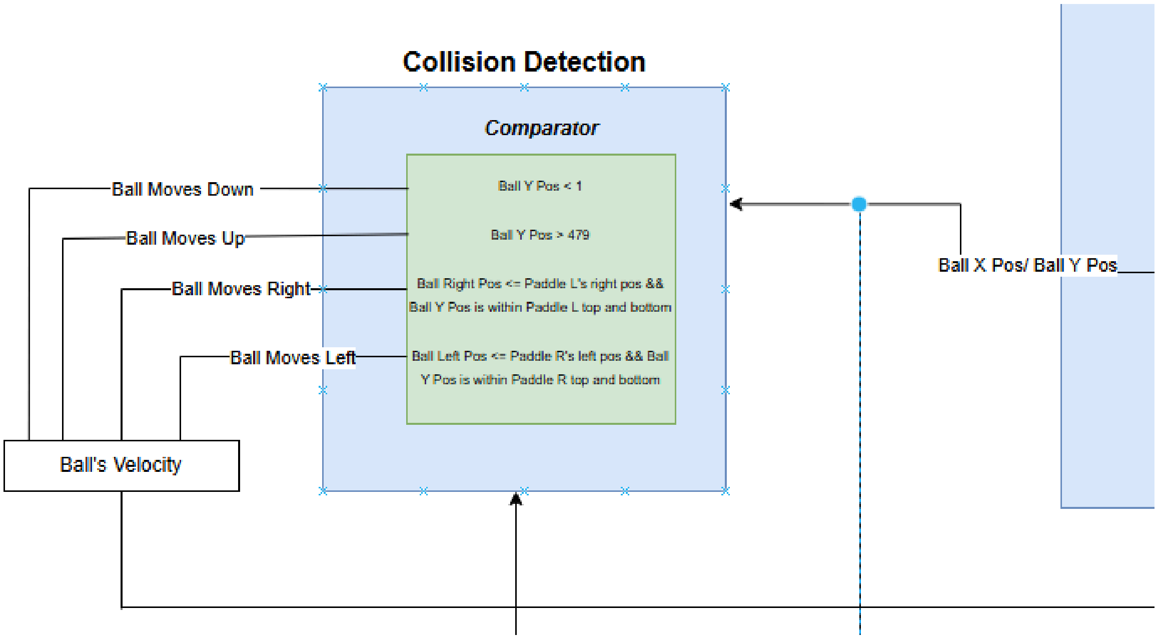


Paddle logic:

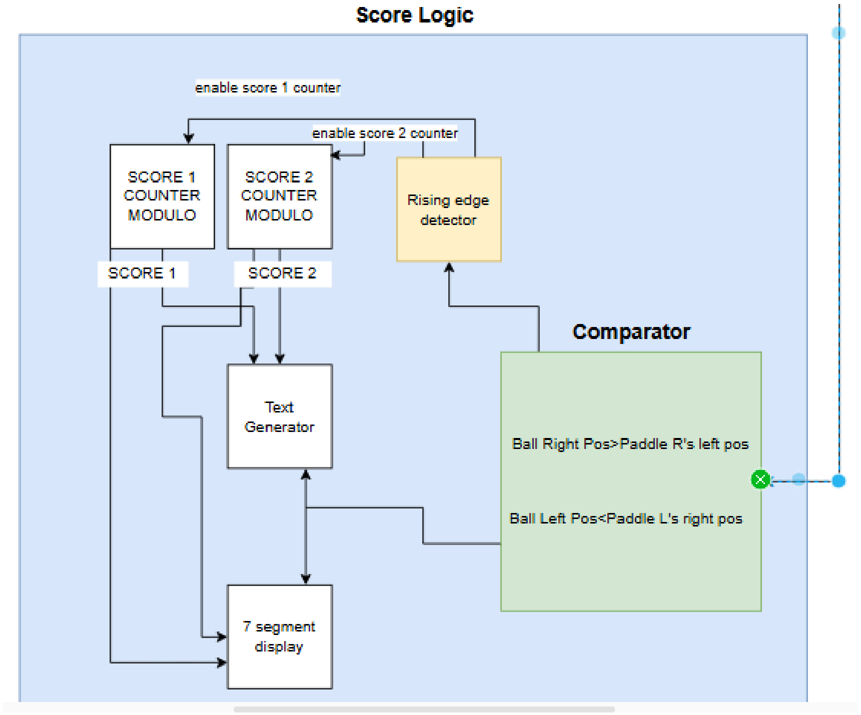
The paddle logic receives input from the push buttons to make the paddle move up or down, making sure that the paddle stops when it reaches the top or bottom of the screen.



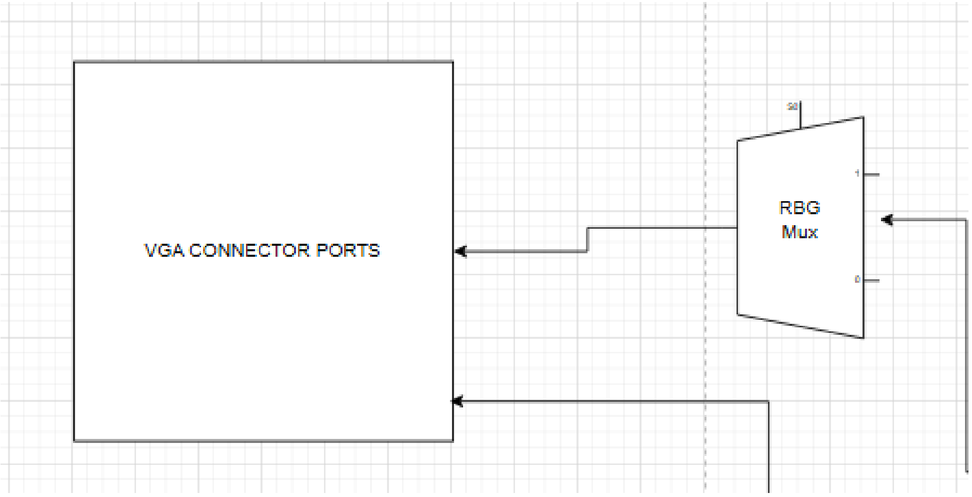
Collision detection:
It simply works by comparing the ball position with the paddle position or the top and bottom edges of the screen and negating the ball velocity when this happens.


Collision Detection
Comparator
Ball Y Pos < 1
Ball Y Pos > 479
Ball Right Pos <= Paddle L's right pos && Ball Y Pos is within Paddle L top and bottom
Ball Left Pos <= Paddle R's left pos && Ball Y Pos is within Paddle R top and bottom
Ball Moves Down
Ball Moves Up
Ball Moves Right
Ball Moves Left
Ball's Velocity
Ball X Pos/ Ball Y Pos

Score logic:
It works by comparing the position of the ball and whether it passed the paddles or not. A rising edge detector is used to ensure a single increment in the score per point for the player. Then the enables for the counters are activated and the score is increased. The score is then displayed on the screen and the 7-segment display in the FPGA.



Final stage:
Based on which graphical component is on the RGB colors it is chosen to be sent to the screen with Vsync and Hsync signals.



Modules:

top.v: connects the project parts.

pixel_gen.v: The pixel generation circuit for the project

vga_controller.v: Implements a VGA controller for the monitor.

ascii_rom.v: gets the representation of characters on the screen from their ASCII codes

Pong_text.v: Displays the score and words in the game

ball_rom.v: the representation of the ball on the screen

bcd_experiment1.v: 7 segment display managing

binarycounter.v: a counter used for the clock divider

clock_divider.v: A clock divider module

Rising_edge.v: a rising edge detector used for the debouncer

debouncer.v: Adjusts the inputs from the push buttons in the FPGA.

Const_pong1.xdc: the constraint file that connects the project with the FPGA

Challenges:

We had to work with the available material even though it was not ideal for our needs. For example, the standard resolution for such games' monitors is different from the ones we had in the lab. We overcame this problem by adjusting the monitor's display settings.
The ball logic and collision detection were tricky to implement correctly, especially when the ball was out of the screen. It was challenging to make it so that when the ball passed either the left or right edge of the screen, it would reappear at the center and travel in the opposite direction. This required careful handling of the position and velocity of the ball.