**Project Report**

**On**

Implementation of VGA on Pong Game

**BY**

**Toshit C  
(2019A3PS0462H)**

**S Ayaz Hussain  
(2019B5A81108H)**

**Prateek Nuti  
(2019AAPS1228H)**

# 

**SUBMITTED IN Complete FULFILLMENT OF THE REQUIREMENTS**

**OF**

**EEE F348: FPGA Based System Design Lab**

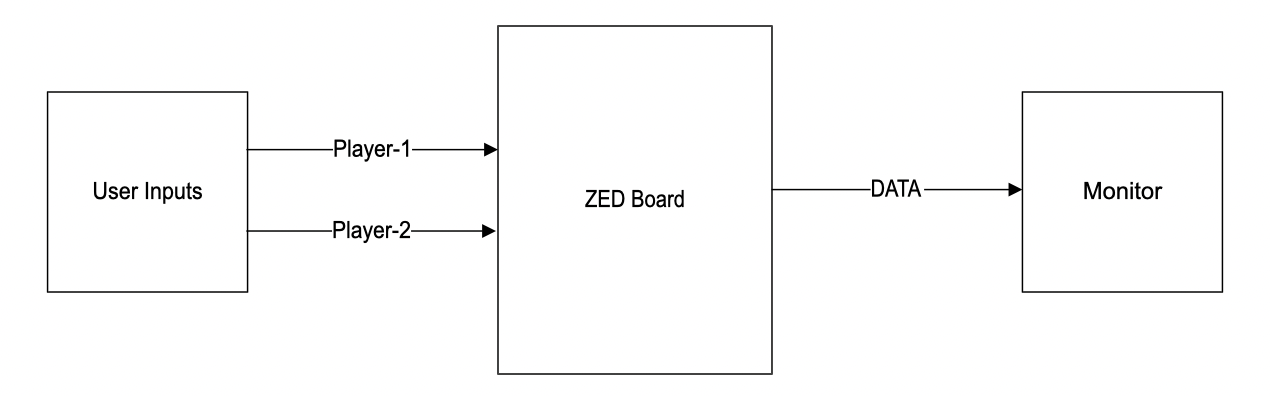


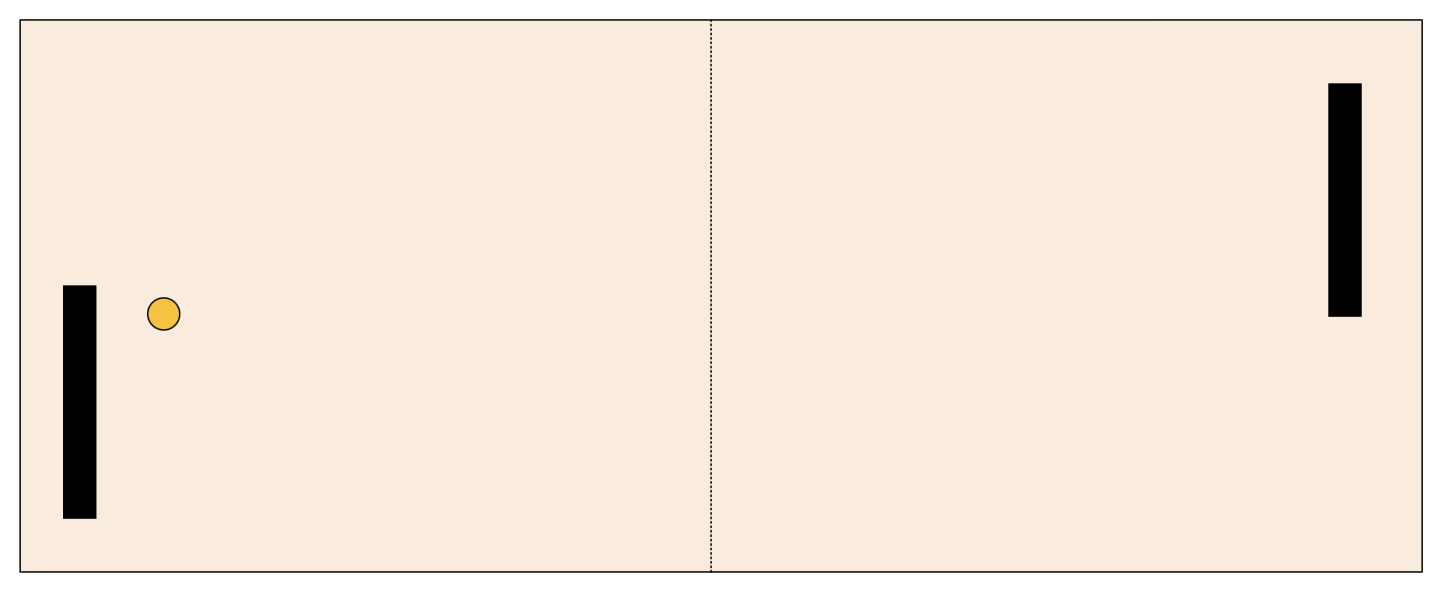
**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)**

**HYDERABAD CAMPUS (January 2023**)

# Abstract

* The Pong game on the ZedBoard is displayed using a VGA interface and written in Verilog.
* We use a finite state machine (FSM) to keep track of the game state.
* The player controls the state of the game with push buttons on the zed board, which helps the players to adjust the bar they are using to push the ball.
* Each player uses 2 input signals for up and down.
* So we will need 4 input push buttons in all. An LED or monitor can be used to display the game.
* The scores of players are displayed on the screen.





**Objective:**

To learn about graphics at the hardware level and get a feel for the power of FPGAs.

To learn how screens work by playing Pong.

**Hardware or Software required:**

Hardware: ZED board, LCD screen

Software: Vivado

# Contents

[**1. Abstract 2**](#_heading=)

[**2. Contents 3**](#_heading=)

[**3. Introduction 4**](#_heading=)

[Game Controls: 4](#_heading=h.l8f231fds9yl)

[**4. Detailed description 5**](#_heading=)

[**5. Implementation 6**](#_heading=)

[a) vga.v: 6](#_heading=h.aufsymjh9nso)

[b) clk\_divider.v: 6](#_heading=h.10m9frxobpn9)

[**6. Simulation and Results 7**](#_heading=h.uvw19xhfwuxm)

[VGA Display Images: 8](#_heading=h.ly6pzh5nh4l5)

[Schematic : 9](#_heading=h.3vecp0gzs9ya)

[Timing Constraints : 10](#_heading=h.42hpk9gc6li9)

[Hardware Utilization : 10](#_heading=h.o98t1vpov4vj)

[Power Consumption : 10](#_heading=h.g5dx2e6z1u8b)

[**7. Conclusion 11**](#_heading=)

[**8. Demonstration Video 11**](#_heading=h.4hpwxw7260ym)

[**9. References 11**](#_heading=h.1yej00v2s2ll)

# Introduction

VGA (video graphics array) is a video display standard that was first used in IBM PCs in the late 1980s and is now extensively supported by PC graphics hardware and displays.

Game Design:

The primary modules used in the design of the pong game are: top.v and clkdivider.v, here the top module of the verilog code consists of the VGA module as well which is responsible for the display of the game on the LCD Screen.

The clock division is performed so that the motion of the ping pong ball is at an optimal speed, along with the motion of the paddles and is observable by the human eye, for which the code is attached below.

Certain lines of the module will let you configure gameplay parameters. The parameters are mentioned in the code within the top module preceded by the keyword ‘parameter’. Accordingly, the other values in the code get adjusted for the gameplay.

## Game Controls:

The whole game requires inputs from the user using the Zedboard push buttons

and switches. The F22 switch is used for reset functionality throughout the game. In the start screen, the top button is used to move to the above option and bottom is used to move to the below option, and the center button is used to select the highlighted options and start the game. In single-player mode, the top button is used to move the players’ bat up, and the bottom button to move down. In dual-player mode, the top and left buttons are used to move the left player, whereas the right and bottom buttons are for the right side player.

# Detailed description

The Zedboard (FPGA) has 8 bit I/O for the display, but since we need this to represent the 3 colors RGB, we allocate 2 bits for each colour, i.e., Red, Blue, Green.

The input controls for both the players are present on the FPGA board which can be used to control the paddles’ vertical motion, thereby allowing the ball to stay within the boundaries mentioned in the **640 x 480** image resolution, where the screen utilization is only **640x480 pixels**.

We use a clock divider module to reduce the internal clock frequency of the Zedboard from 100 MHz to 50 MHz, where the clock is being divided by 2. This is done to make the speed of the objects in motion optimal. The 2 player controlled paddles and the ball remain in motion where the background (Pong table) remains static in Green.

We also use the debounce.v file to ensure that the use of buttons on the FPGA board undergoes a smooth transition. So, it is necessary to debounce the buttons in our design. The test bench for the same is attached below along with the simulation results in the Vivado program.

Here, since we do not use a CPU for the simulation of the game, a finite state machine (FSM) is utilized. The initial state of the paddle is in the center of the allotted screen size which is the Reset state in the code, and for every click on the button there is a movement of the paddle by 1 pixel in the vertical direction for the respective player. The reset signal ensures the state machine begins in a known state.

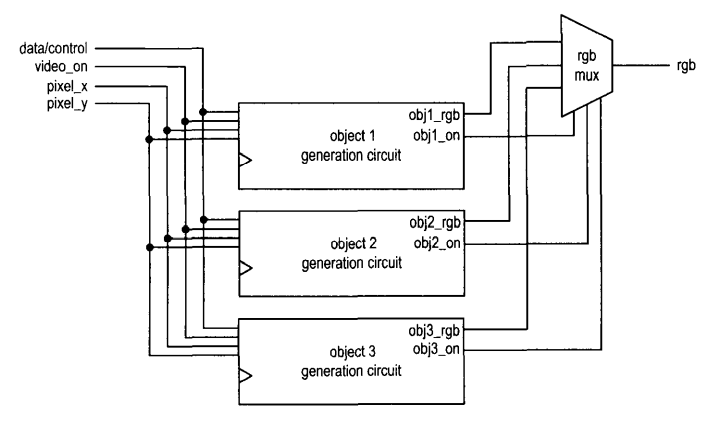
In play, the player compares the center of the paddle to the top of the ball. If the ball is below, it moves the paddle down as far as possible (without going beyond the bottom of the screen). Otherwise, it compares the middle of the paddle to the bottom of the ball, and if the ball is above, it moves up.

If the middle of the paddle is level with any part of the ball, we don’t move the paddle; this avoids jerky paddle movement.

The game starts with a start screen with two options, one that selects Single player

mode and the other that selects Dual player mode. For navigating those options, two buttons, up and down, are used. The center button is used to choose that option. The single-player mode screen has a computer-based player that plays the right bat. The player will play the left-side bat. If the player wins the game, “WON” will be printed on the screen. Or if the player loses the game, then “LOST” will be displayed. The player can still choose the two options if he wants to continue to play another game. In dual-player mode, two players will be assigned two bats to play. If the left player wins the game, “1 WON” will be displayed; else, “2 WON” will be displayed on the VGA screen. This exit screen also contains two options to play if he wants further. The current scores of the two players will be displayed on the game screen.

# Implementation



We have written the Pong Verilog code using Xilinx Vivado and implemented the design on ZedBoard. As for the design of the algorithm we have created 3 modules:

## vga.v:

This is the top module of our design. Here, the paddle sizes are taken into account and the motion of the paddles is adjusted such that the paddle does not exceed the limits of the screen resolution provided. Similarly, the collision of the ball with the paddle is also mentioned (for each of the right and left walls).

The debounce module with parameters is set to provide a 50 MHz clock frequency. This can be adjusted according to the player to increase or decrease the frequency of the game. The same applies for the size of the ball, paddle.

The vga module is also used here as a relay to send the FSM output received by the top module to the display. Here, 2 bits have been allotted for each color as in the Zedboard.

## clk\_divider.v:

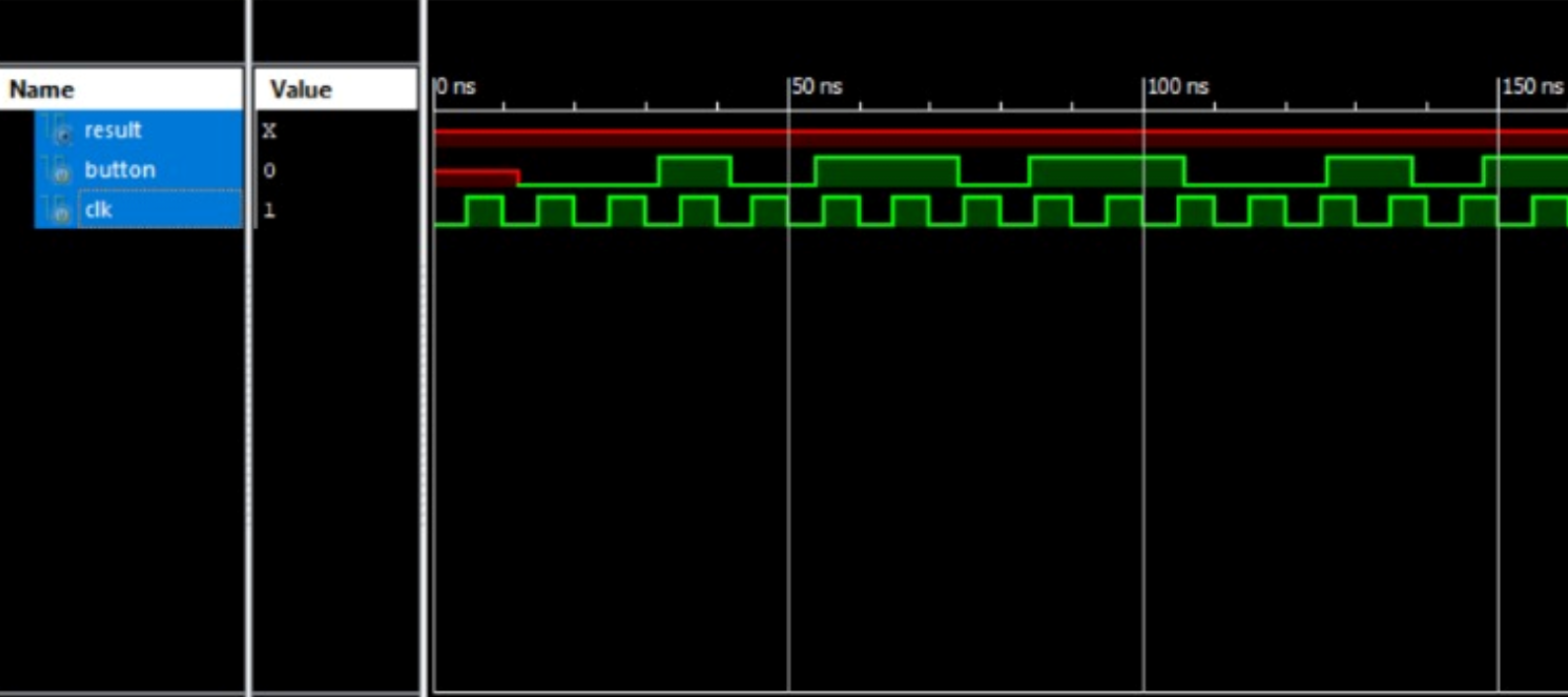
This module is used to divide the Y9 clock frequency by 2. The default frequency is 100 MHz, but in our code we toggle the clock whenever r\_nxt is equal to 2, so that for every second iteration there is a shift in the clock cycle leading to doubling of the time period.

# Simulation and Results

For testing the model, we have written a test bench taking the clk and input buttons and showing the results. The buttons are pressed every 2 sec signifying the movements of paddles. The testbench for the same is depicted in Fig 5.1. and the simulation result is shown in Fig 5.2.



*Figure 5.1 main*



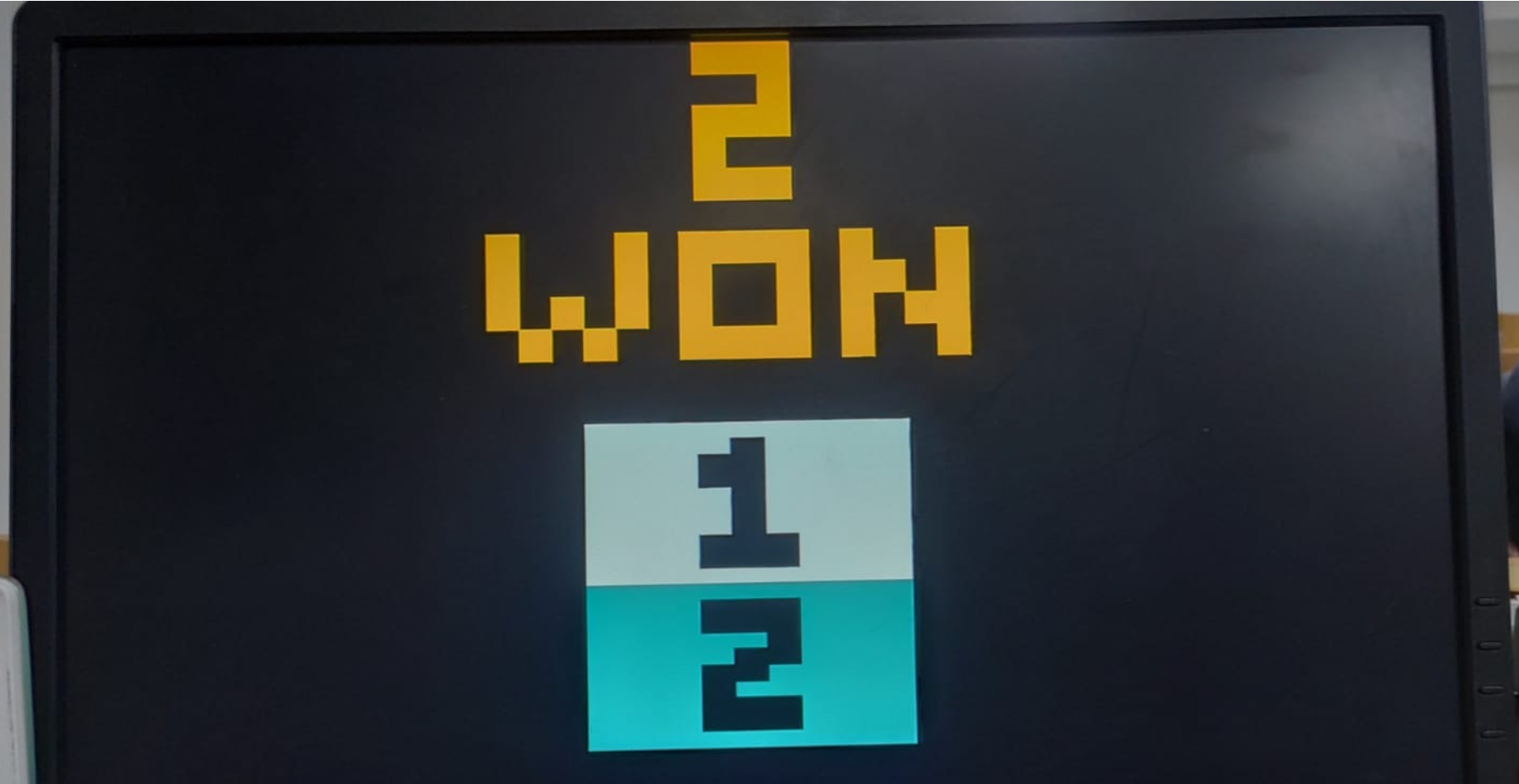
*Figure 5.2 Simulation Results*

## VGA Display Images:

Here attached a screenshot of the game.

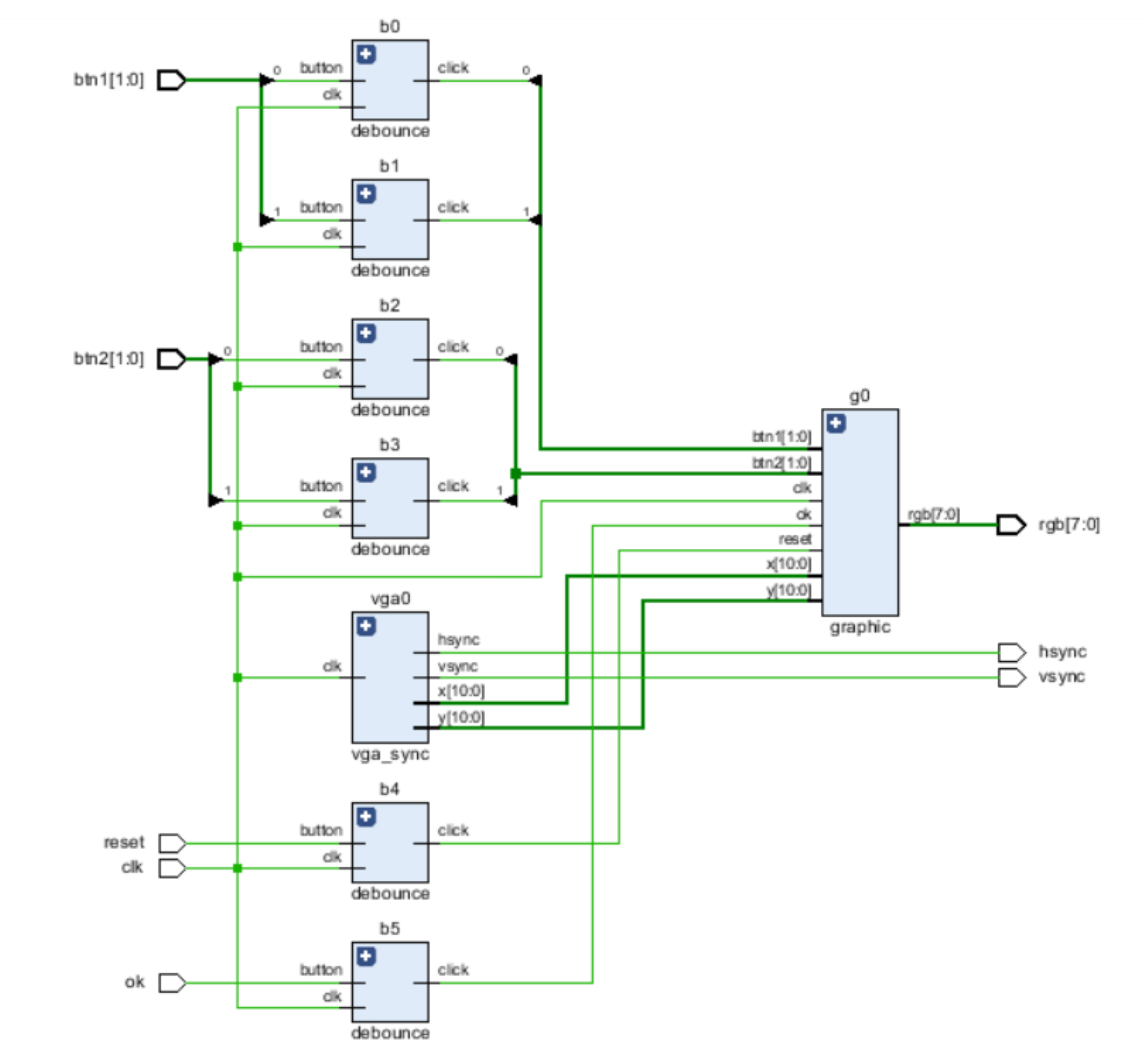


*Figure 5.3 VGA Display Images 1*



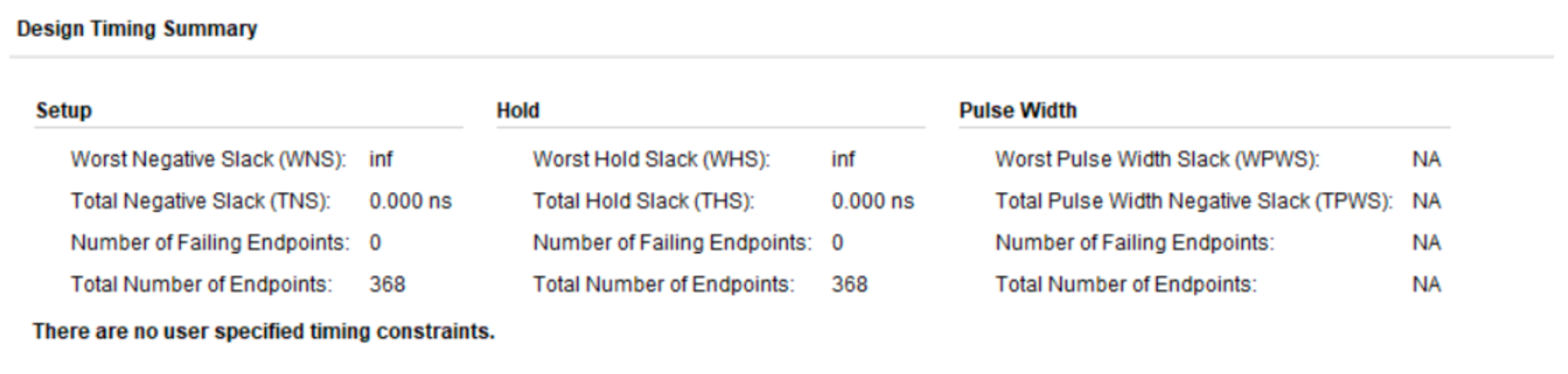
*Figure 5.4 VGA Display Images 2*

## Schematic :



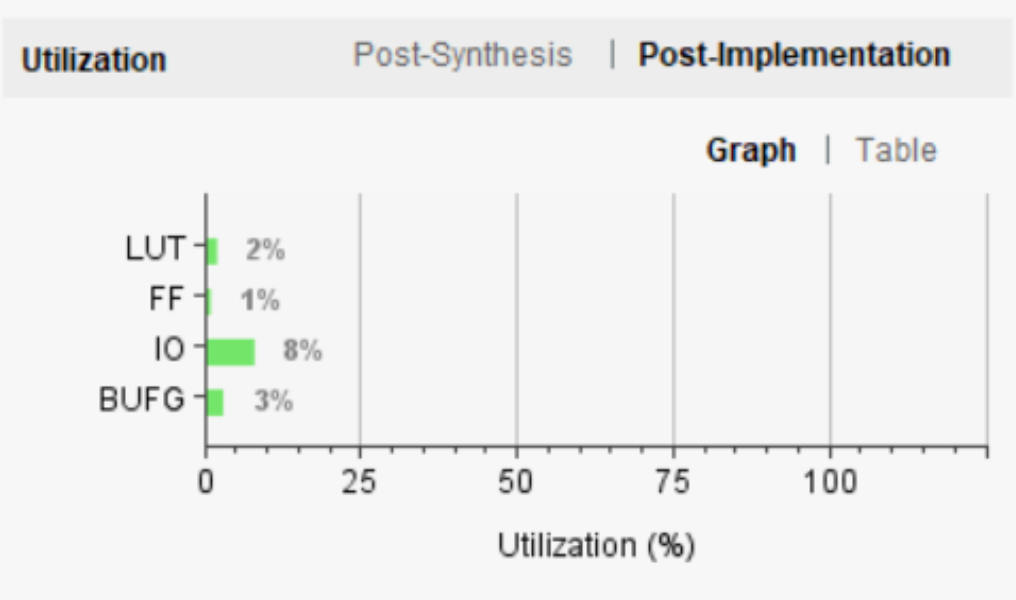
*Figure 5.5 Schematic of the Design*

## Timing Constraints :



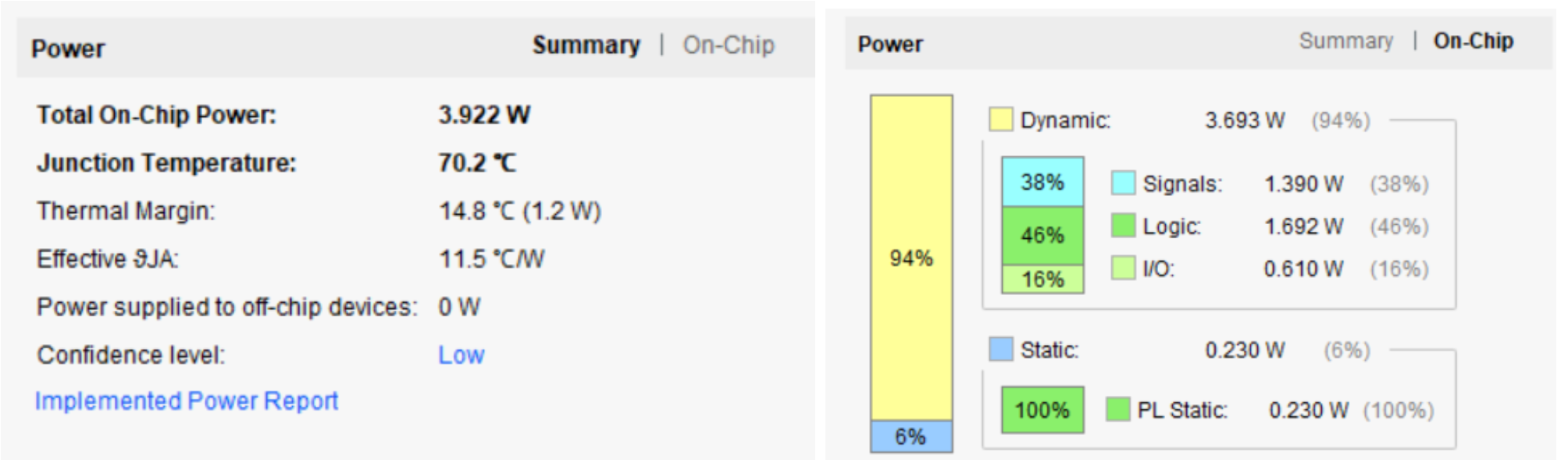
*Figure 5.6 Timing constraints*

## Hardware Utilization :



*Figure 5.6 Hardware Utilization*

## **Power Consumption :**



*Figure 5.7 Power consumption*

# Conclusion

The implementation of the Pong game has been accomplished with the aid of Xilinx Vivado and subsequently deployed on the Zedboard platform with the requisite interconnections.Throughout the duration of this project, we have acquired a significant amount of knowledge regarding VGA technology and its interfacing techniques in order to achieve a desired display output on a monitor. The commencement of the game is initiated upon the deactivation of the reset switch. The Zedboard facilitates gameplay for two individuals, designated as Player 1 and Player 2, through the utilization of corresponding switches.

# Demonstration Video

[Video Link](https://drive.google.com/drive/folders/1qeXG4w5HzEyNKjbmgsFXd-yffBpUbtsU?usp=share_link)

# References

1. [FPGA Prototyping with verilog with examples](https://faculty.kfupm.edu.sa/COE/aimane/coe405/FPGA%20Prototyping%20with%20Verilog%20examples.pdf)
2. <https://digitalsystemdesign.in/interfacing-vga-display-with-fpga/>
3. <https://www.beyond-circuits.com/wordpress/tutorial/>
4. <https://www.ece.ucdavis.edu/~bbaas/180/tutorials/vga/>