EE314 TERM PROJECT FINAL REPORT

Fake Quidditch

Yavuz TAÇKIN

Electrical and Electronics Engineering Department

Middle East Technical University

Ankara, Turkey

e226161@metu.edu.tr

Erdoğan Mert DEMİRYÜREK

Electrical and Electronics Engineering Department

Middle East Technical University

Ankara, Turkey

e216623@metu.edu.tr

***Abstract*—This report is about the EE314 Digital Circuits Laboratory term project namely a fake quidditch game design.**

Keywords*—*Verilog, FPGA, VGA display, Modelsim

I. INTRODUCTION

FPGA is an integrated circuit that can be programmed in the field by the customer. Hardware description languages are used to implement projects. VGA displays can be controlled by Verilog. In this project, we designed a fake quidditch game which can be played by 4 players. This game is displayed on a monitor with 640x480 resolution by VGA connection.

II. DESIGN AND OPERATION OF SUBSYSTEMS

A. Player

There are 4 playable objects in this project. Each team has 2 players, one can move vertical and the other moves horizontal. These players are rigid bodies and collide when they hit another rigid body according to physics laws. They collide when distance between centers of the ball and player is equal to the sum of the radiuses. The movement system is achieved by shifting the objects in x or y direction 1 pixel at a time when the clock enable is active. When they reach the frame of the playable game area, they stop moving and maintain their position. Furthermore, when the bludger collides with a player, it puts the player in the idle state for 10 seconds and blocks player movement. The player control is achieved by the buttons and switches on the FPGA board.

player1 = ((x-490)*(x-490)+(y-yson1)*(y-yson1) <400)

Table 1: Defining Player

Player 1 (Red Vertical)	
If switch3 is off and button 3 is not pressed	Stops
If switch3 is off and button 3 is pressed	Goes Down
If switch3 is on and button 3 is not pressed	Goes Up
If switch3 is on and button 3 is pressed	Stops

Table 2: Moves of Player 1

Player 2 (Red Horizontal)	
If switch2 is off and button 2 is not pressed	Stops
If switch2 is off and button 2 is pressed	Goes left
If switch2 is on and button 2 is not pressed	Goes right
If switch2 is on and button 2 is pressed	Stops

Table 3: Moves of Player 2

Player 3 (Blue Vertical)	
If switch1 is off and button 1 is not pressed	Stops
If switch1 is off and button 1 is pressed	Goes Up
If switch1 is on and button 1 is not pressed	Goes Down
If switch1 is on and button 1 is pressed	Stops

Table 4: Moves of Player 3

Player 4 (Blue Horizontal)	
If switch0 is off and button 0 is not pressed	Stops
If switch0 is off and button 0 is pressed	Goes Left
If switch0 is on and button 0 is not pressed	Goes Right
If switch0 is on and button 0 is pressed	Stops

Table 5: Moves of Player 4

B. Ball

At the beginning of the game, the ball moves in only horizontal direction. When it hits to the frame, collision occurs, and it bounces to the opposite horizontal direction if none of the players touch the ball yet. If a player touches the ball, ball moves according to the physics laws. Bouncing angle is arranged according to ball's and player's coming positions. We divide the players into pieces to arrange the balls bouncing angle. For rest of the game if ball hits to the frame it bounces with an angle with +45 or -45 according to its coming position. Also, if ball enters to a goal then it reset itself to starting position and same procedures will continue.

```
ball = ((x-x_ball)*(x-x_ball)+(y-y_ball)*(y-y_ball)
<100)
```

Table 6 : Defining ball

```
((ballx-330)*(ballx-330)+(bally-60)*(bally-60) <400) ||
((ballx-225)*(ballx-225)+(bally-90)*(bally-90) <400)
```

Table 7 : Goal Check for one Goal

C. VGA Driver Module

VGA monitor - FPGA connection is provided by a 15-pin D-sub connector. The main pins we are interested in are pins numbered by 1, 2, 3, 13 and 14.

Pin Number	Purpose of the pin
1	Red output
2	Green output
3	Blue output
13	Horizontal sync
14	Vertical sync

Table 8: Pin numbers and their purposes

Horizontal synchronization sets the boundaries of the horizontal limits and vertical synchronization sets the boundaries of the vertical limits. The VGA display system also contains front and back porchs. This time is used to return to the start of the line of the screen. In our VGA driver module, we defined x and y variables. They are kept between the frame of the display and show the position of each pixel. To create ball, we defined vertical and horizontal positions of the center which varies according to the balls movement and it comes from the ball submodule, and radius of the ball which is 10 pixels. VGA code for ball is like "assign ball = ((x-x_ball)*(x-x_ball)+(y-y_ball)*(y-y_ball) <100) ? 1 : 0;". Then, we decide the color of the ball and it displayed in the monitor. To display the goals, a circle is created for a constant center and outer and inner radius. Only the region between inner and outer region is displayed on the monitor. We arranged the outer radius for 25 pixels and inner radius for 20 pixels. To display the players, a circle is created for a varying center and constant radius which is 20 pixels. Creating the bludger's display is same with the ball. There is a frame at the monitor which represents the border of the field. Also, there is a timer and score board and 4 little counters. Main counter is count downs from 180, numbers between 180 and 0 can be seen at the monitor. Scoreboard displays the number of goals for each team. 4 little counters count from 9 to 0 in order to show how many seconds remains for disabled players because of bludger.

D. Bludger

Bludger is randomly bouncing from the frames and players. When bludger hits a player, it is disabled for 10 seconds and player cannot move.

E. Goal Checker

In this game there are 3 goalpost rings for each team. When the center of the ball enters the blue team's goal, red team scores. When the center of the ball enters the red team's goal, blue team scores. After one team scored, number of goals on the scoreboard of that team increases by 1 which is initially 0. Furthermore ball, players and bludger returns to their initial positions.

F. Countdown Timer

Every game lasts 3 minutes. This subsystem counts down from 180 with 1 Hz. When the timer reaches 0, the game resets. The number system in this part is designed using seven segment digit display. In order to count down from 180, we used a 3-bit BCD counter. Least significant bit of the counter is initially

0 and it decreases by 1 at every 1 second. When LSB of the counter which is corresponds to unit's digit reaches 0, next state becomes 9 and middle bit which corresponds to the tens digit decreases by 1 which is initially 8. Therefore, tens digit of the counter decreases by 1 for 10 seconds. When tens and unit's digits are equal to zero, next state of them are equal to 9 and most significant bit which is hundreds digit decreases by 1 which is initially 1.

G. Clock Enable

Clock enable is an essential subsystem of this project. The FPGA works with 50 MHz clock. VGA, BCD counter and moving objects which works with different clock rates, we designed three clock enablers. For VGA we created a 25MHz clock and it toggles at every positive edge of main clock. The countdown timer decreases at 1 second and in order to use it, 50 MHz clock should be enabled once in every 50 000 000 clock hits. To arrange the speed of the moving objects we used a clock enable that enabled once in every 500000 clock hits.

III. STATE DIAGRAM

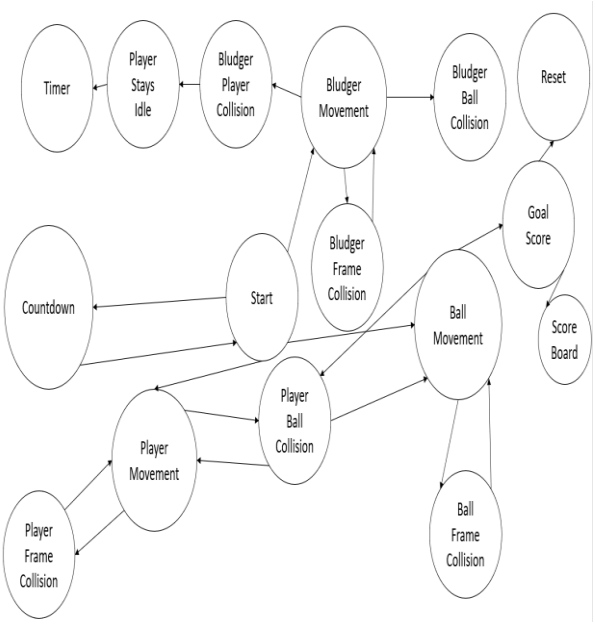

Figure 1: State diagram of the fake quidditch project.

IV. TIMING DIAGRAM

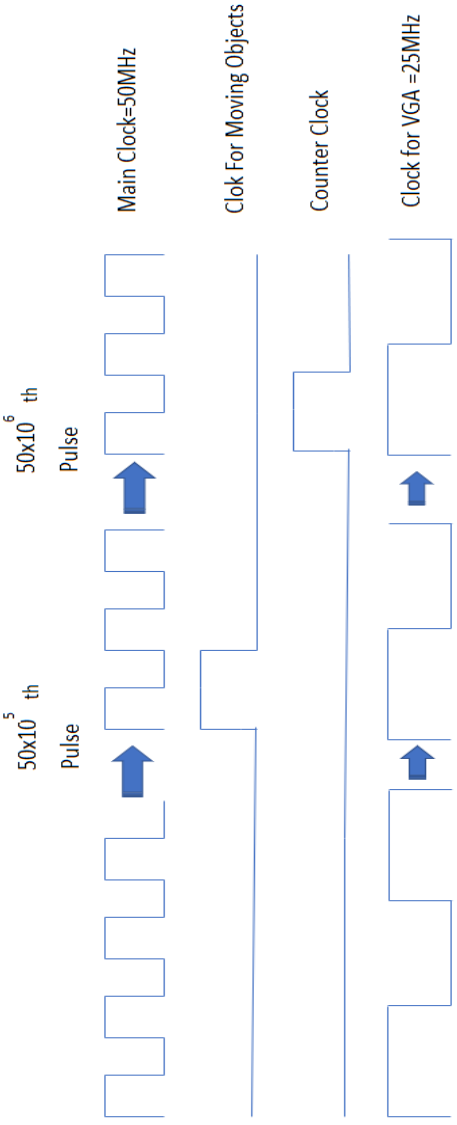

Figure 2: Timing diagram of the fake quidditch project.

V. SIMULATION RESULTS

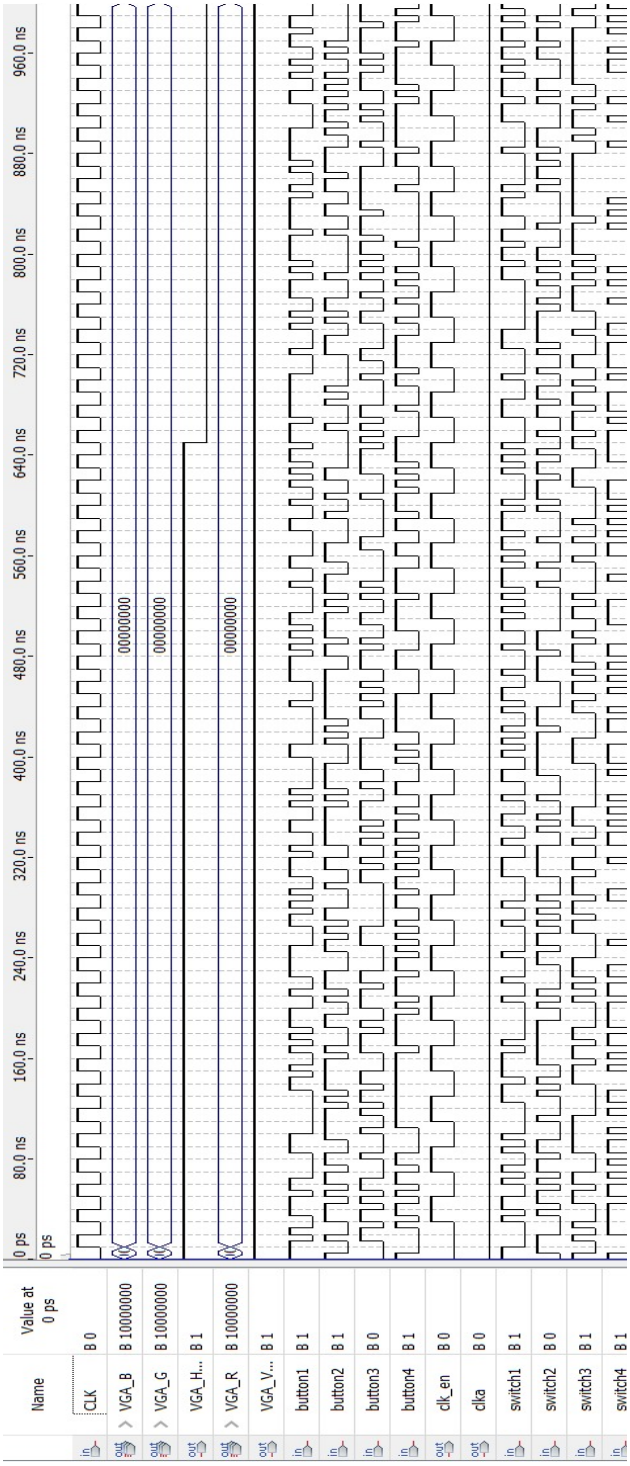

Figure 3: Simulation Results

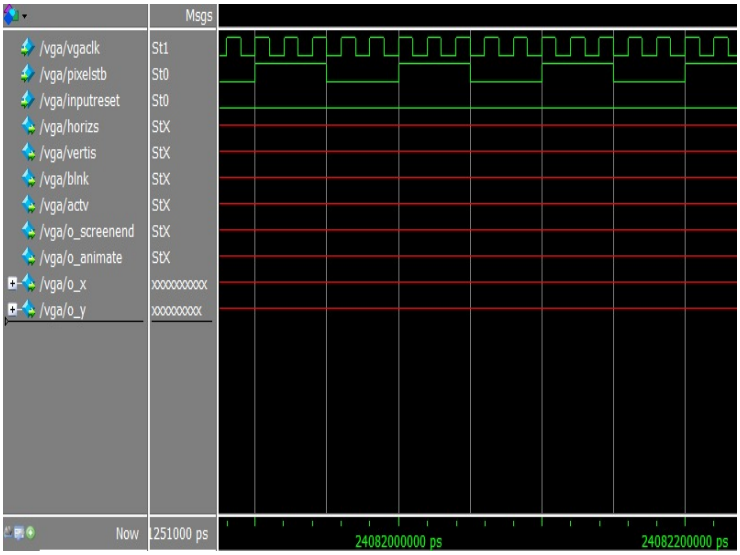

Figure 4: Simulation Results

VI. CONCLUSION

We designed a fake quidditch game in this term project and become familiar with FPGA and Verilog coding. This game can be played by 4 players. The ball's and players' collision systems are designed according to physical equations. There is a 3 minute counter for the game and each goal is equal to 1 point in this game. Working on this project was highly informative. Learning hardware design with VGA driver can be used in many other areas. Moreover, by using Quartus software, we learned further applications of our studies in digital electronics laboratory.

Working with VGA display was an exciting part of this project. At first, we struggled with driving it. After literature research and various compiles, we overcame this problem. By using clock enables, conditioning in Verilog and using different modules to design a top module our practical skills developed. Designing this project took so much time but in the end all the modules worked properly and in coordination. The FPGA implementation was as expected in the end. All in all, this project was highly informative and beneficial for us.

REFERENCES

[1] Chu, P. (2008). FPGA Prototyping By Verilog Examples. New Jersey: Wiley & Sons Publication

APPENDIX

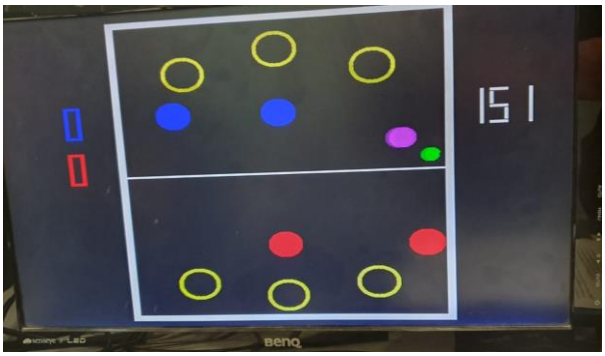

Figure 5: Fake Quidditch game displayed on VGA connected monitor

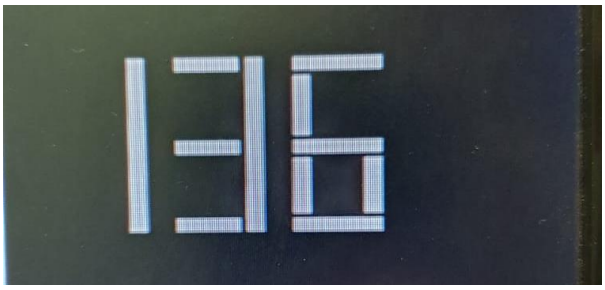

Figure 6: 3 minute countdown timer displayed on the monitor

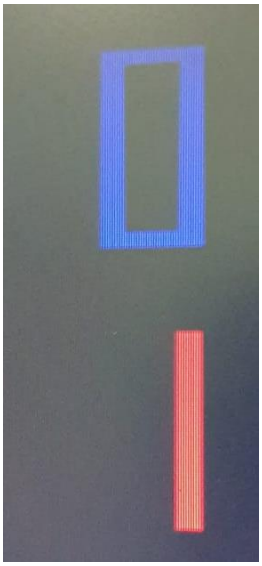

Figure 7: Scoreboard displayed on the VGA monitor