Design of Game “Pong” Using VHDL

Mr.Sudhanshu Mohan Khare
Dept. of Electronics and Communication. LNCT
Bhopal, India
sudhanshumohankhare@yahoo.in

Mr.Nakul Nagpal
Dept. of Electronics and Telecommunication. JIT
Nagpur, India
nkn5603@rit.edu

Ms. Mayuri Chawla
Dept. of Electronics and Telecommunication. JIT
Nagpur, India
chawlamayuri23@gmail.com

Ms. Sharon Bhatnagar
Dept. of Electronics and Communication. LNCT
Bhopal, India
sharon_bhatnagar1989@rediffmail.com

***Abstract*—The game pong is a two player game involving two paddles and a bouncing ball. In order to win the game the player needs to strike the ball with the paddle whenever the ball bounces towards the user’s end. The computer controls the other paddle. The game is lost when a player misses the ball. For this game a user paddle was designed that is controlled by the up and down arrow keys of the keyboard while the computer paddle has artificial intelligence (AI) to judge the ball position when it approaches towards the computer paddle. Keyboard interfacing, VGA interface, and the game logic was implemented using VHDL. Quartus II tools are used for simulation and synthesis and the design is implemented in the FLEX 10 Network Processor Unit(NPU) which is embedded on the UP1[1] board. This board also has two seven segment displays, LEDs, pushbuttons, switches, PS/2 and VGA connections as peripherals.**

***Keywords*— VGA; VHDL; user paddle; ball; computer paddle**

I. INTRODUCTION

Purpose of this lab was to know about VGA (video graphics array) interface and keyboard interface better, initially VGA interface and keyboard interface were implemented but they were independent of each other, here the keyboard control are managing the movement of one of the object on VGA Display. For keyboard interface a state machine as in fig 1 is designed to have proper handshaking between keyboard and UP1 board as the keyboard clock operates at a different frequency, the keyboard sends data serially so when complete 1 byte of data is received a scan ready signal is enabled and while the state machine is reading it upload its status on the read signal as reading data so that the scan data available is not updated with the next data until the machine read it. The VGA interface use signals horizontal sync and vertical sync for synchronization and pixel row and pixel column counter to locate the pixel address, to design this unit process was used consisting of sequential statement to define sequence of operation. The state machine that define the game design consist of two state idle and game play as shown in figure 2 when game is reset it jump to the Idle restore the initial conditions and then jump to game play state which was looped to itself until game over.

II. DESIGN

The specification for this lab were as follows:

“The project will be to create the game *Pong*. For those of you who do not remember this 1970s game, it is basically a two dimensional version of table tennis. This project is to be coded entirely in VHDL. Schematic entry may be used to interface the keyboard and VGA sync modules previously created, or else this can be done structurally in VHDL”[3]. As per specification components required were VGA sync for VGA interface and keyboard interface for user controls. For the Keyboard interface a FSM was designed for detecting two keys; the up arrow and down arrow. To detect whether a key is pressed or released the corresponding make and break codes are monitored. The output of this FSM consists of Up and Down signals, these signals help in evaluating the Y position of user paddle. VGA SYNC [1] is used to generate the timing signals needed for the VGA video display; it generates horizontal sync and vertical sync and consist of two counters used as horizontal and vertical count that give the pixel row and column address. These pixel row and column indicate X and Y coordinate of the present pixel location.

The main file included the logic for bouncing ball,

user paddle and computer paddle control. The design of this file proceeded in number of steps, first it started with display of three static objects: the ball, a user paddle and a computer paddle. In the second step the ball was made to move in X and Y directions and also bounce against the wall. To realize this, the ball‟s X and Y position were controlled by signals X motion and Y motion which calculate the new X and Y position depending on the current location of the ball and also ball is made to reflects from the wall of the screen this was achieved by applying boundary conditions which altered the X motion and Y motion in turn evaluating the new X and Y position. The third step was the user paddle control here the X position of paddle is fixed just need to evaluate the Y position so Y motion component was used, this involved the keyboard interface to get control signal for the up and down movement of the paddle these signals were used to evaluate Y motion so the next Y position is calculated.

Fourth step involved building computer paddle AI,

computer paddle is similar to user paddle i.e. its X position is fixed and it performs movement in Y direction. Initially computer paddle was designed such that its Y position was same as that of ball as so the computer paddle was bouncing up and down. Further to provide the computer paddle with AI it was modified such that it judge the Y position of ball and accordingly make a move and that to only when ball approach it else computer paddle doesn‟t show any motion. Now the design was ready with user paddle, computer paddle and bouncing ball, the ball bounced against the walls but not the paddle and it was desired that if the user or computer paddle miss the ball the game should hang indicating that the user or computer lost depending upon who missed the ball, so last step is to mange ball movement, for this reference point for reflection of ball from left and right wall is shifted to facing edge of user and computer paddle and the Y axis limit is just the limit of paddle so if the ball‟s X position same as X position of user or computer paddle and its Y position is in the range of paddle Y position +/- half the height of paddle plus half the height of ball i.e. it checks for the complete paddle range it will be clear from figure 3. This was simple and step wise approach, at the end a reset was introduced that sets the ball position to center of the screen and for the paddle making their Y position match that of ball i.e. center of vertical edge of screen maintaining their X position same.

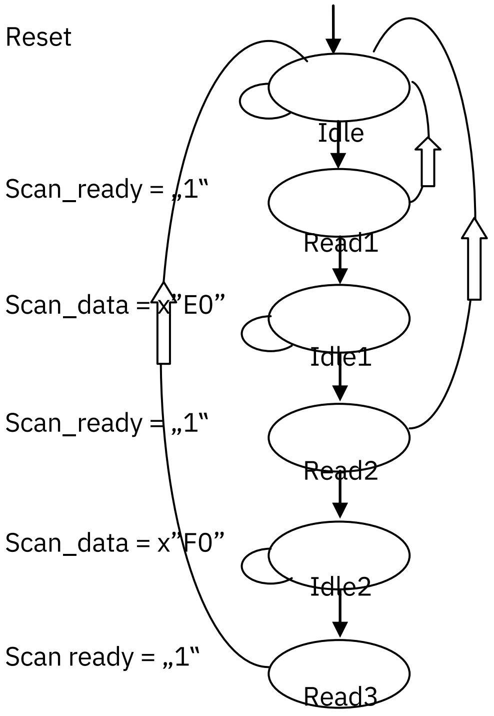

Figure 1

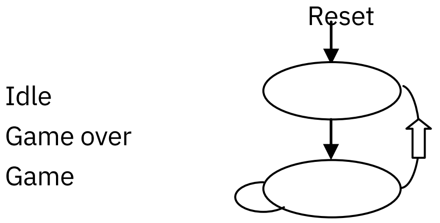

Figure 2

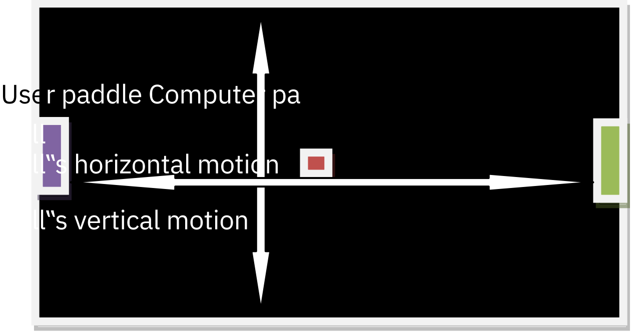

Figure 3

III. TEST PLAN

Testing done in this case doesn‟t include test bench or waveforms as there are no arithmetic or logical operation. The parameter needs to be tested here include movement of ball, user paddle and computer paddle, for which code was downloaded in FLEX 10k on UP1 board and making the required connection, keyboard and VGA these parameter can be observed. First for user paddle movement in Y direction that is controlled by keyboard, if Up arrow of keyboard is pressed the paddle move in upward direction and for Down arrow move downward and when key is released it stops immediately. If both keys are pressed simultaneously the paddle will move downward as down has priority over up. It was observed that the keyboard interface handshaking is proper because the paddle reacts properly when the key is pressed.

Second step of testing included the computer paddle AI. When the ball is approaching the computer paddle it will adjust its Y position to match the Y position of ball. The speed of the computer‟s paddle is set to move at 2 pixels at a time. The computer paddle compare the Y position of ball only when it is heading towards it and X position is greater than 320 pixel column. This distance can be increased or decreased, and was set to allow the computer adequate time to return the ball most of the time. As movement of ball is fixed 2 pixel diagonally also movement of paddle is fixed, so the time limit that is assigned to paddle for judgment is sufficient enough to chase ball. Testing result showed that in the given period of time the computer paddle can cover 80 to 85% of the wall edge that is what it was designed as it starts tracing the ball when it approaches and is in the vicinity of computer paddle so the time period in which the ball reach in front of that ball stop its movement. It was found that ball was working properly.

Finally reset is tested to sets the Y position of user and computer paddle and X-Y position of ball as desired.

IV. RESULT

FLEX 10k FPGA chip was configured and peripherals connected to the board i.e. keyboard and VGA display. It was observed that user paddle and computer paddle were displayed in center along there corresponding edge and the ball in the left towards the bottom showing game over this is because the ball is reset with X and Y position to form center of the screen and X and Y motion of ball was set to move along the straight line in third quadrant and user paddle did not respond initially indicating game over stage. After pressing reset the user paddle, computer paddle and ball take the initial position and ball start heading towards the user paddle. with the help of keyboard control user paddle is moved either upward or downward as per position of ball and if user mange to strike the ball it bounces back and head towards computer paddle , computer paddle adjust its position automatically comparing its Y position with that of ball and make decision accordingly and if it manage to strike the ball , the ball bounces back towards user paddle this continues till either user or computer paddle miss the ball and then the game pauses the user paddle, computer paddle and ball in their current positions and the ball change its color.

.

V. CONCLUSION

The Pong game was successfully designed as per the specifications that states “It should have two player one user controlled and one computer controlled and a rolling ball to form pong game which is 2 dimensional version of table tennis”. The user paddle and computer paddle are working properly also the ball is bouncing properly. Initially when the code was in testing phase there were some errors in the game, when the corner of ball hit the corner of paddle, ball passed and indicating game over but making some correction in the code this error was removed. When the FPGA is initially configured the game starts in the game over state this can be corrected by defining the reset condition properly. Due to time constraints was not worked out.

Further, some additional features can be added to the

REFERENCES

[1] James O. Hamblen, Michael D. Furman, “Rapid Prototyping of Digital Systems: A Tutorial Approach”, 2nd ed., Kluwer Academic Publishers, 2000.
[2] Stephen D. Brown, Zvonko G. Vranesic, “Fundamentals of Digital Logic with VHDL Design“, 2nd ed., McGraw-Hill Professional, 2005.
[3] Lab 06 handout.