Digital System Design Lab

A MINI PROJECT REPORT

ON

IMPLEMENTATION OF TIC-TAC-TOE USING FPGA

Submitted By

HARSHITHA KARNAM	**(4NM17EC052)**
JANE RASHMI FERNANDES	**(4NM17EC054)**
JATIN GANJOO	**(4NM17EC055)**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified)
Accredited with 'A' Grade by NAAC

April 2019

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belagavi)
(ISO 9001:2015 Certified)
Nitte – 574110, Karkala, Udupi District, Karnataka, India

Department of ***Electronics and Communication Engineering***

CERTIFICATE

This is to certify that **HARSHITHA KARNAM (4NM17EC052) , JANE RASHMI FERNANDES (4NM17EC054)** and **JATIN GANJOO (4NM17EC055)** bonafide students of N.M.A.M. Institute of Technology, Nitte has submitted the report for the project entitled "**IMPLEMENTATION OF TIC-TAC-TOE USING FPGA"** in partial fulfillment of the requirements for the Digital System Design Laboratory during the year 2018-2019.

Name of the Reviewer **Signature with date**

Project Evaluation

Name of the Examiners **Signature with date**

1. _____ _____

2. _____ _____

CHAPTER 1

INTRODUCTION

Tic-tac-toe, noughts and crosses, or Xs and Os is a paper-and-pencil game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.

Figure 1.1 illustrates a sample game play where first player wins the game.

Figure 1.1 Example of Tic-tac-toe game

Players soon discover that the best play from both parties leads to a draw. Hence, tic-tac-toe is most often played by young children.

Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the searching of game trees. It is straightforward to write a computer program to play tic-tac-toe perfectly or to enumerate the 765 essentially different positions (the state space complexity) or the 26,830 possible games up to rotations and reflections (the game tree complexity) on this space.

The game can be generalized to an m,n,k-game in which two players alternate placing stones of their own colour on an $m \times n$ board, with the goal of getting k of their own color in a row. Tic-tac-toe is the (3 3 3) game. Harary's generalized tic-tac-toe is an even broader generalization of tic-tac-toe. It can also be generalized as a n^d game. Tic-tac-toe is the game where n equals 3 and d equals 2. If played properly, the game will end in a draw, making tic-tac-toe a futile game.

Therefore, keeping in mind the rules of the game we are trying to implement tic-tac-toe using Verilog on a FPGA and provide an interface such that the player can play a fair game with the computer.[1]

CHAPTER 2

BLOCK DIAGRAM

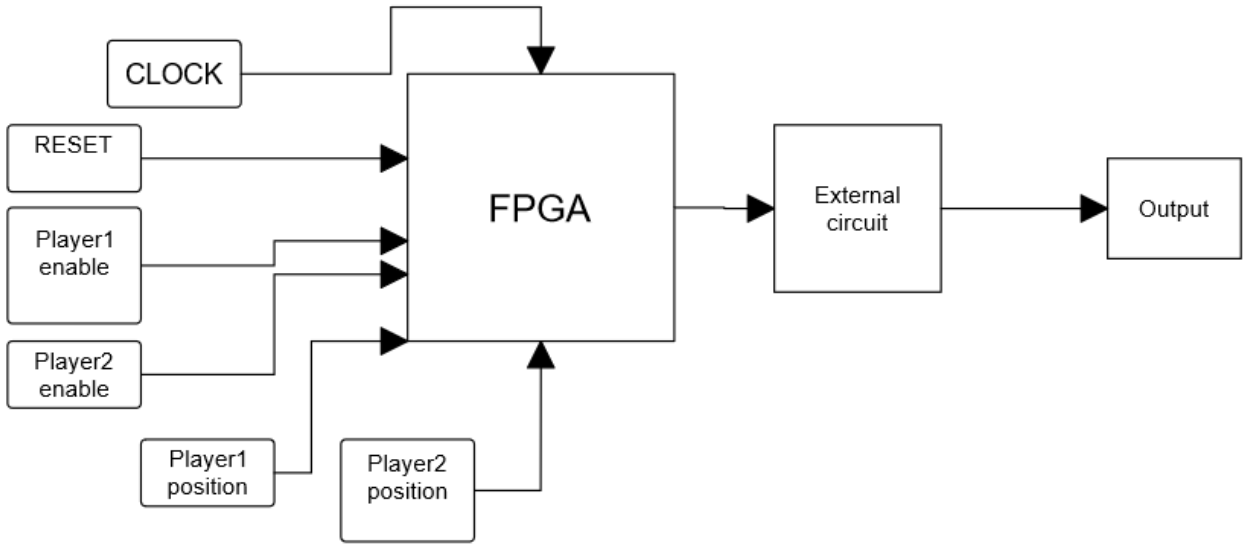

Figure 2-1 Block diagram of the design

Reset is used to start the game and to ensure that every position is off. Then player 1 enable is switched on allowing player 1 to make a move ,then player 2 enable is switched on to allowing player 2 to make a move. Dip switches are used to select a position for player 1 and player 2.When the game is finished, reset is made 1 to restart the game .

CHAPTER 3

CIRCUIT DESIGN

3.1 CIRCUIT DESCRIPTION

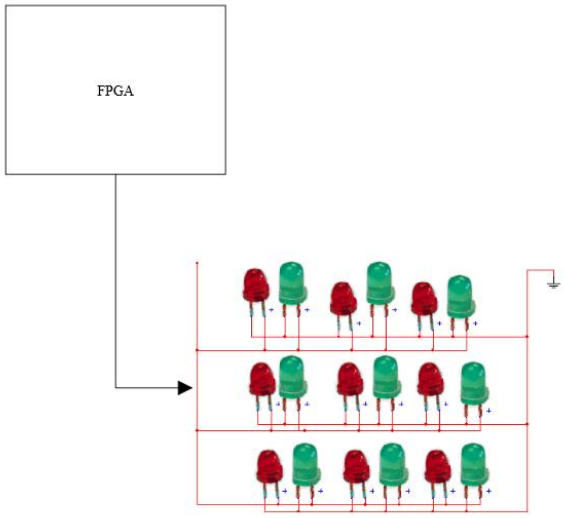

Figure 3.1 Circuit diagram

The synthesizable code written using Verilog is tested extensively with the help of a simulation tool. If it checks and confirms that all the expected functions are carried out satisfactorily then the code is dumped into the FPGA hardware which is connected to an external circuit which consists of a dip switch which is used to select the position of the player and a set of LEDs arranged in the form off a 3x3 matrix in order to display the game.

3.2 COMPONENT DESCRIPTION

Table 3.1 Components Required

S.No	Components used	Quantity
1	FPGA	1
2	Dip Switch	2
3	LEDs	20
4	Flat Ribbon Cable Connecters	4
5	9V Adaptor	1
6	JTAG Programmer	1

3.2.1 FPGA

Figure 3.2.1 FPGA model

A Field-Programmable Gate Array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term “field-programmable”. The FPGA configuration is generally specified using a Hardware Description Language (HDL), similar to that used for an Application Specific Integrated Circuit (ASIC). Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. Many FPGAs can be reprogrammed to implement different logic functions, allowing flexible reconfigurable computing as performed in computer software.

3.2.2 Dip Switch

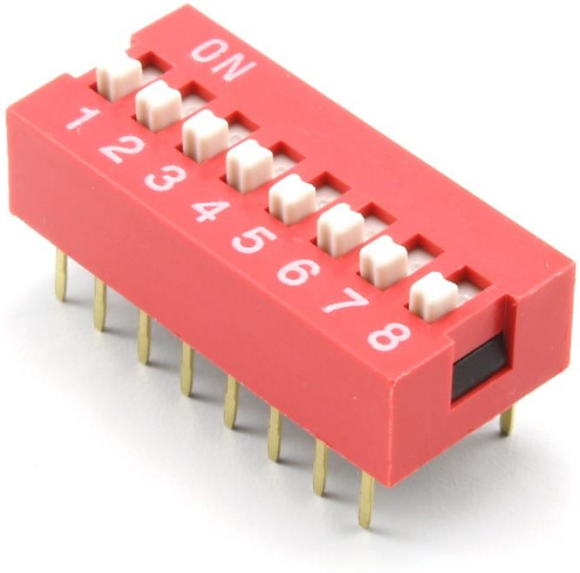

Figure3.2.2 A dip switch

A DIP switch is a manual electric switch that is packaged with others in a group in a standard Dual In-Line Package (DIP). The term may refer to each individual switch, or to the unit as a whole. This type of switch is designed to be used on a printed circuit board along with other electronic components and is commonly used to customize the behaviour of an electronic device for specific situations. DIP switches are an alternative to jumper blocks. Their main advantages are that they are quicker to change and there are no parts to lose.[2]

3.2.3 LED

Figure 3.2.3 LEDs

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. This effect is called electroluminescence. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.[2]

3.2.4 Flat Ribbon Cable Connectors

Figure 3.2.4 Flat ribbon cables

It is a cable with many conducting wires running parallel to each other on the same flat plane. Hence the cable is wide and flat. It incorporates multiple signal-ground pairs and facilitates error-free connections. To take advantage of the ribbon cable, a mating connector must be incorporated into the target system.[2]

3.2.5 9V Adaptor

Figure 3.2.5 9V adaptor

A 9V adaptor converts 230V from AC mains to 9V power supply for the FPGA board.[2] This is used to power up the circuit for the hardware experimentation.

3.2.6 JTAG Programmer

Figure 3.2.6 JTAG Programmer

This is a Pb-Free (RoHS compliant) USB compatible cable for in-circuit configuration and programming of all Xilinx devices. Platform Cable USB is certified by the USB Implementors Forum (USB-IF). It is compatible with Full 10

Speed (USB 1.1) and Hi-Speed (USB 2.0) ports, addressing the needs of new PCs that have eliminated legacy IEEE 1284 parallel ports.[2]

Key features:

- True plug-and-play
- Bus-powered USB device (no power supply required)
- Selectable target clock frequency up to 24 MHz
- Compatible with Full-Speed and Hi-Speed USB ports

3.2.7 Software

Xilinx ISE (**I**ntegrated **S**ynthesis **E**nvironment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesise ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

Xilinx ISE is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the Model sim logic simulator is used for system-level testing. Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and ChipScope Pro. The primary user interface of the ISE is the Project Navigator, which includes the design hierarchy (Sources), a source code editor (Workplace), an output console (Transcript), and a process tree (Processes).

The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a tree structure. For single-chip designs there may be one main module, with other modules included by the main module, similar to the main()subroutine in C++ programs. Design constraints are specified in modules, which include pin configuration and mapping.

The Processes hierarchy describes the operations that the ISE will perform on the currently active module. The hierarchy includes compilation functions, their dependency functions, and other utilities. The window also denotes issues or errors that arise with each function.

The Transcript window provides status of currently running operations, and informs engineers on design issues. Such issues may be filtered to show Warnings, Errors or both.[2]

CHAPTER 4

WORKING PRINCIPLE

In this game, two players play the Tic Tac Toe game. When the player1/ player2 plays the game, a 2-bit value is stored into one of the nine positions in the 3x3 grid like Xs/ Os in the real paper-and-pencil version. 2'b00 is stored into a position when neither the player1 or player2 plays in that position. Similarly, 2'b01 (X) is the value to be stored when the player1 plays in the position and 2'b10 (O) is the value to be saved when the player2 plays in the position. The players play the game by setting the corresponding position value in the dip switch. Red/ Bleu LED is lit in a position when the position is played by the either of the players respectively.

Consider the 3x3 grid table as shown in table 4.1 below as the order of the positions being played:

Table 4.1 3x3 Grid table

1	2	3
4	5	6
7	8	9

The winner of the game is declared when one of the player successfully places three similar (01-Xs) or (10-Os) values in the following row pairs: (1,2,3) (4,5,6) (7,8,9) (1,4,7) (2,5,8) (3,6,9) (1,5,9) (3,5,7).

The winner detecting circuit is designed to find the winner when the above winning rule is matched. To detect an illegal move, a comparator is needed to check if the current position was already played by either the computer or player. Moreover, “No space” detector is to check if all the positions are played and no winner is found.[1]

The Figure 4.1 shows the flow of the game

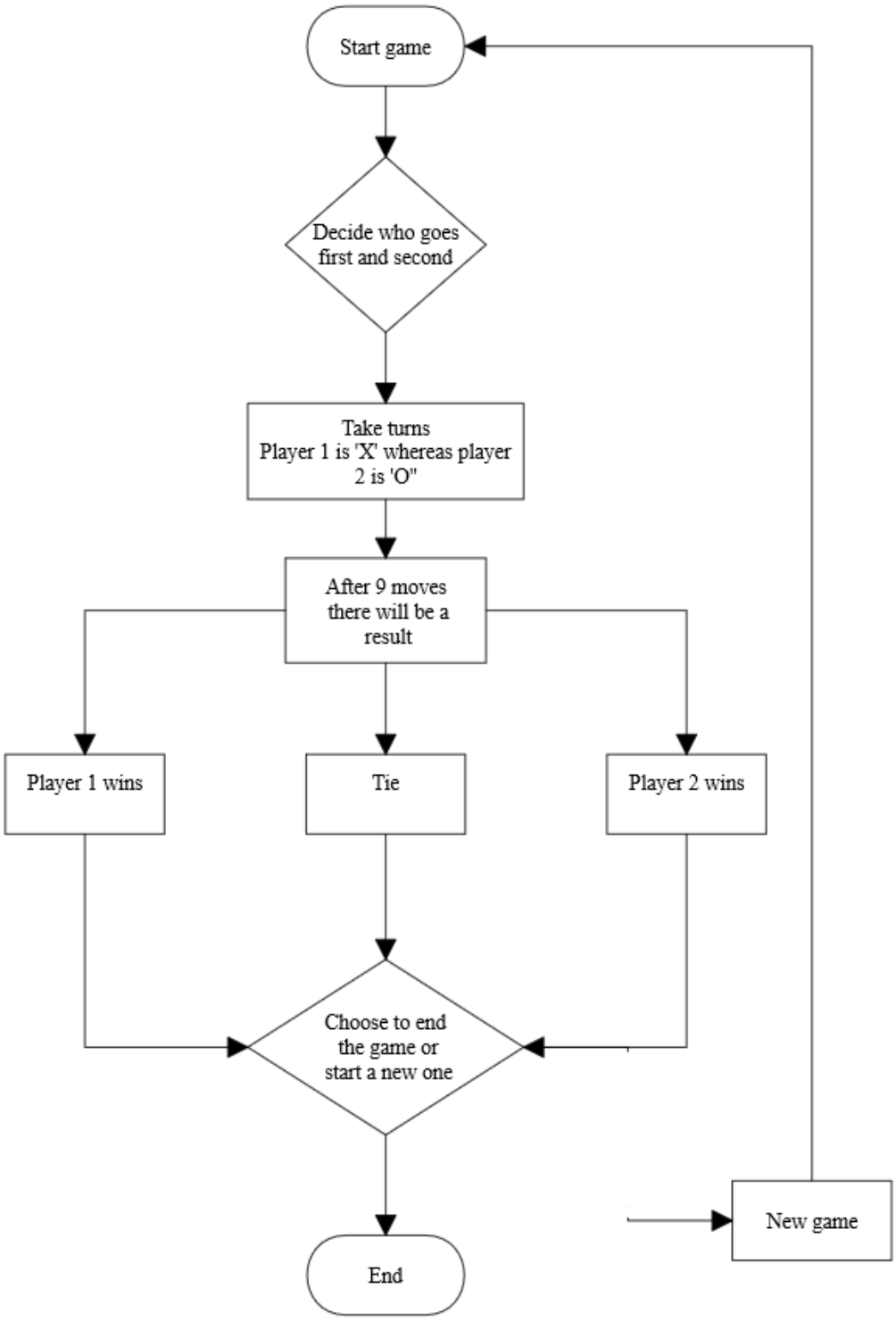

Figure 4.1 Flow diagram of the game

CHAPTER 5

RESULTS

Figure 5.1 shows the simulation of the code on a simulation software

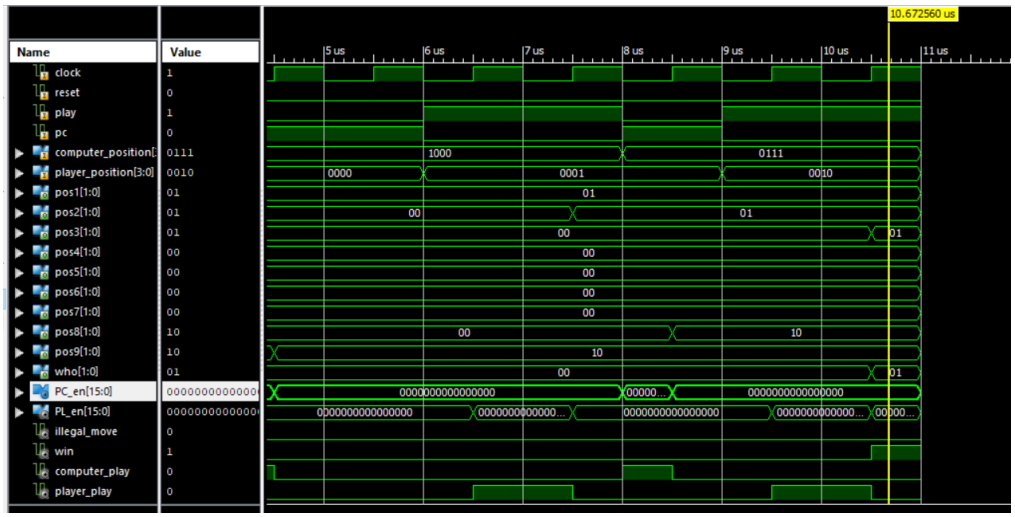

Figure 5.1 Simulation result

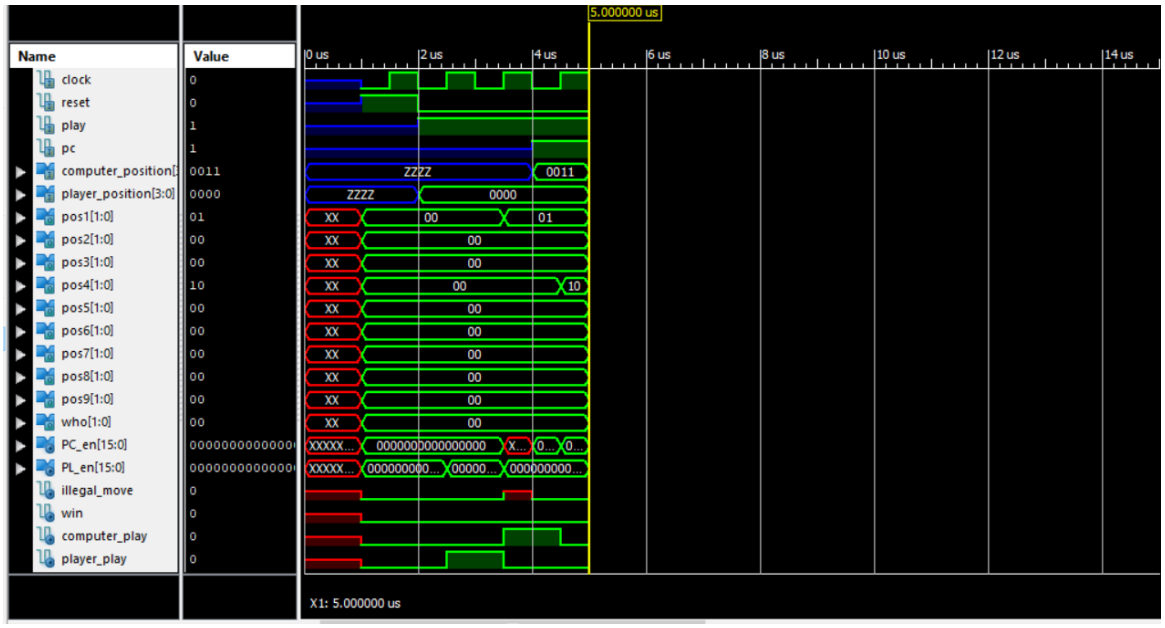

Figure 5.2 Simulation for the first two moves of the game

Figure 5.2 shows the simulation output when player 1 has played in position corresponding to binary input 0000 when play is made high and the player 2 has played in position corresponding to 0011 when pc is high

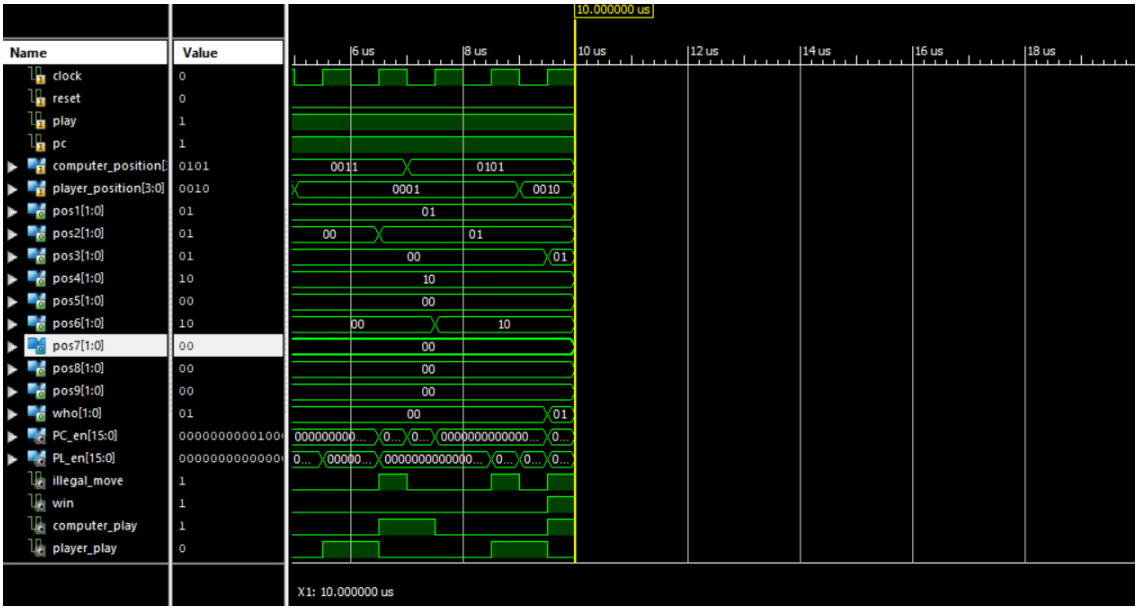

Figure 5.3 Simulation which shows the winner of the game

Figure 5.3 is the simulation output where the winner is declared. Here player 1 is the first to occupy three consicutive positions in the same line, hence player 1 is declared the winner by giving the variable who a binary value of 01.

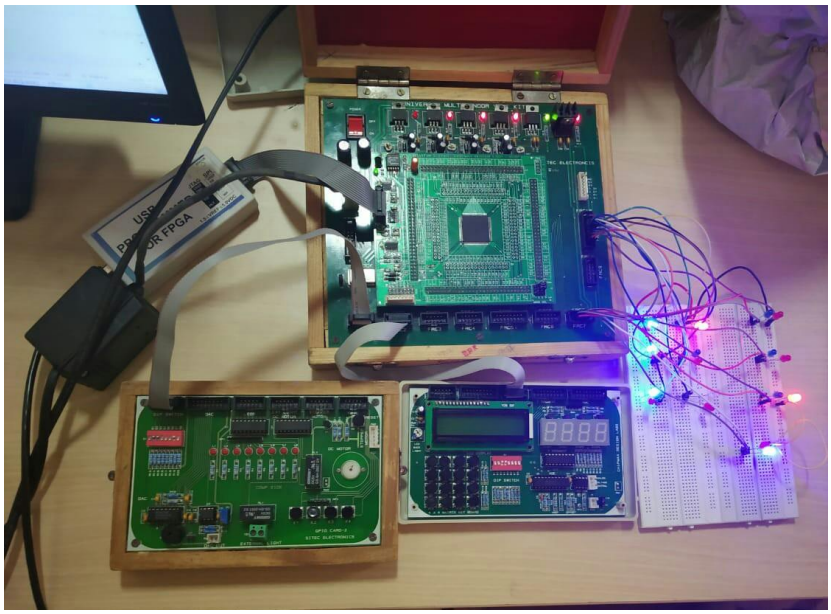

Figure 5.4 Hardware output

The code is dumped into the FPGA and the output is observed on the external circuit containing LEDs. These LEDs indicate the position in which a player has played. Figure 5.4 shows the output of the game on the hardware.

CHAPTER 6

CONCLUSION

Thus by using a FPGA kit to implement the code, the output can be observed on an external circuit containing different colored LEDs arranged in the form of a 3x3 grid.

The program can be further modified to provide an option for the player to play the game with the computer as well. By using a VGA interface and an image processing kit the game can be played in the computer.

REFERENCES

[1] https://www.fpga4student.com

[2] https://en.m.wikipedia.org>wiki>Website