

A Strategical Approach for Implementing Digital Games on FPGA

Sk.Irfan Ali¹, P.Shyam², Sk.Dilkush^c, M. Baby^{3,1}, P.Kalyani^{4,2}, S.Suresh^{a,b,c,3}, G.Teja^{a,b,c,4}, S.Jayasri^{a,b,c,5}

¹Assistant Professor, Department of ECE, RGUKT, Nuzvid, AP and 521202, India ² Assistant Professor, Department of ECE, RGUKT, Nuzvid, AP and 521202, India ³Assistant Professor, Department of MME, RGUKT, Nuzvid, AP and 521202, India

^{1,2,3,4,5B}-Tech Students, Department of ECE, RGUKT, Nuzvid, AP and 521202, India

* Corresponding author: Sk.Irfan Ali: Tel.: +8985580754; fax: +0-000-000-0000 .

E-mail address: irfanalimj@rguktn.ac.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Field Programmable Gate Arrays (FPGAs) are providing potential solutions to the most of the technological advancements. Starting from the basic modeling of RISC and CISC processors to the complicated designs of processors for data analytics, the FPGAs are being used. Parallelism and Pipelining are the features that are enabling for its application is almost all the domains of Electronics and Communications Industry. In view of such applications, getting expertise in implementation of Digital System Designs on FPGA will become inevitable for Electronics and Communication Engineering Graduate. Learning from Gaming has proved to be successful from the very past. Inspired from the Project based learning pedagogy [15], we have come up with a novel idea of making a student learn FPGA in the process of designing and implementing a single player and two player games. The entire problem formulation of designing these games is divided into tasks and student will be learning these tasks initially. Later, integrating such tasks will evolve into games. As a part of these five games, namely Snake game, Spaceman game, Pacman game, Tic-Tac-Toe and Battle tanker games were successfully implemented on the NEXYS4 DDR FPGA Board. This implementation is done using Xilinx Vivado Software.

Keywords: Snake game; Pacman game; Tic-Tac-Toe game; Spaceman game; Battle tanker game; NEXYS4DDR.

1. Introduction

Field Programmable Gate Arrays (FPGAs) are semi-conductor integrated circuits that are re programmable and reconfigurable. This contain configurable logic blocks along with configurable interconnects. The greatest advantage of this device is the possibility of interface between the processor and the FPGA. These types of interfaces provide fast transmission of the data [1]. This work is oriented towards learning FPGA in the process of designing and implementing gaming. The key modules includes to interface the peripherals like VGA monitor, keyboard, push buttons, switches and audio device of FPGA and to control them.

One of the commonly used screen devices is the Video Graphics Array (VGA). As part of this structure, VGAs are used to display any information, images and figures on the screen [2]. The controlling portion of the input involves a USB keyboard interface. Players use the keyboard to control the game process and execute block movement (Pushbuttons can also be used instead) [3].

Audio interfacing is used to generate music and sound for the game. The successful implementation of these games provides a comprehensive understanding of the design issues over FPGA.

Problem Formulation:

The main task is to learn FPGA through designing games. This is achieved by designing five games over FPGA.

Hardware Design:

FPGA Platform

Designing for FPGA's is conceptually comparable to embedded processors because of their re-

programmability. Simulation tools can be used to check and debug an application's functionality before a physical unit is effectively programmed, and tools are easily accessible from FPGA suppliers for in-system debugging. ARTIX-7 SOC incorporating a general-purpose dual-core ARM processor and a single chip FPGA as shown in Fig.1. This heterogeneous platform combines flexibility in software and hardware effectiveness, enabling developers to distinguish their products through increased performance and energy efficiency. A critical element for successful hardware design is to allow the logic. The custom hardware acceleration must compensate for the overhead interaction.

2. DEVELOPMENT OF A GAME

VGA Interfacing

'Working Display System' is the most significant thing to create games. Video Graphical Array (VGA) is used by most individuals to show Games. So this VGA must be interfaced with FPGA. VGA is a standard for analog video with a 15-pin D-sub connector as shown in Fig 2. VGA has five primary signal pins: one red, green, and blue, and two sync pins. A line is demarcated by horizontal sync. Vertical synchronization marks a screen, also known as a frame as shown in Fig 3. Using 14 FPGA signals, the NEXYS4 board creates a 4-bit VGA port per color and the two standard Horizontal Sync and VS Vertical Sync signals. 4096 distinct colors can be represented using this circuit, one for each distinctive 12-bit pattern. A video controller circuit must be created in the FPGA to drive the sync and color signals with the correct timing [5].

Keyboard Interfacing

Many keyboards connect via a USB (Universal Serial Bus) connector cable to the laptop. The cable must bring energy to the keyboard, regardless of the type of connector used, and must carry signals from the keyboard to the board. A keyboard consists of a key matrix and an embedded microcontroller that monitors (i.e. scans) the key activity and sends the scan code (as shown in Fig4) accordingly, where the scan code is nothing more than the unique code assigned to each key that is sent when a key is pressed [6].

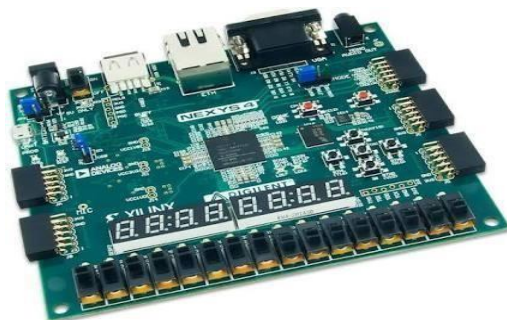


Fig. 1. NEXYS4 DDR Board [7]

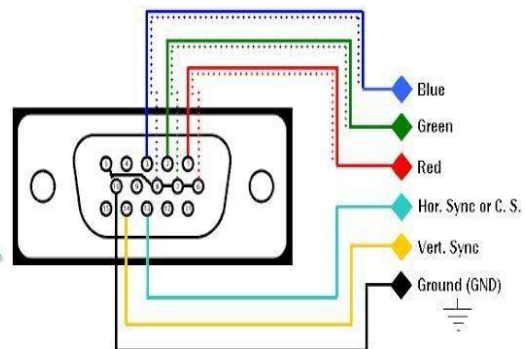
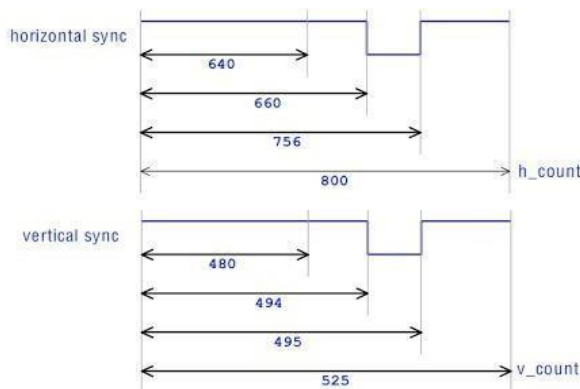


Fig. 2.VGA Connect or ports [8]



ESC 76	F1 05	F2 06	F3 04	F4 0C	F5 03	F6 0B	F7 83	F8 0A	F9 01	F10 09	F11 78	F12 07	
~ 0E	1! 16	2@ 1E	3# 26	4\$ 25	5% 2E	6^ 36	7& 3D	8* 3E	9(46	0) 45	-= 4E	+ 55	BackSpace ← 66
TAB 0D	Q 15	W 1D	E 24	R 2D	T 2C	Y 35	U 3C	I 43	O 44	P 4D	[54] 5B	\ 5D
Caps Lock 58	A 1C	S 1B	D 23	F 2B	G 34	H 33	J 3B	K 42	L 4B	:: 4C	'' 52	Enter ↵ 5A	
Shift 12	Z 1Z	X 22	C 21	V 2A	B 32	N 31	M 3A	,< 41	>. /? 49	? /? 4A	↑ 59	Shift E0 14	
Ctrl 14	Alt 11	Space 29						Alt E0 11	Ctrl E0 14				

Fig. 3. VGA Scanning Signals [9] chart [10]

Fig. 4. Keyboard scan code

Pixel generation circuits The 3-bit RGB signal for the VGA port is generated by the pixel generation circuit. The internal control and information signals indicate the display content, and the present pixel coordinates are provided by the Pixel_x(HS) and Pixel_y(VS) signals from the VGA sync circuit. This circuit of generation of pixels is split into three classifications.

- Object mapped scheme
- Bit mapped scheme
- Tile mapped scheme
-

Various items and letters are presented on the screen using these three methods [5].

Audio Interfacing:

Sound comprises of audible air pressure variability. Microphones transform air pressure variations into different voltages. This variable voltage is to be transformed into a sequence of numbers representing its amplitude, to represent noise digitally. This process is known as analog to digital conversion. The quality of the digital sound representation is determined by the sampling rate and resolution. The higher the sampling rate, the more accurately the sampled signal represents the original signal. There are two methods for audio interfacing.

The first method:

- In order to create a bitmap for the sampled values in the form of number specification format for one particular sound by using an Audacity software (Audacity is a famous multilingual open source audio editor and recording software for recording and editing sounds) to convert sound clip into unsigned 8 bit raw music.
- Bin2h converter is used to convert the 8 bit raw audio file into C header. Then a C++ code was written to get the raw music values into Verilog HDL.
- Finally these sampled values of the sound signal are converted into the PWM signal.

The second method:

Mat Lab script will make coe.m resample audio file (mp3 format) to 8KHz frequency, and convert them to

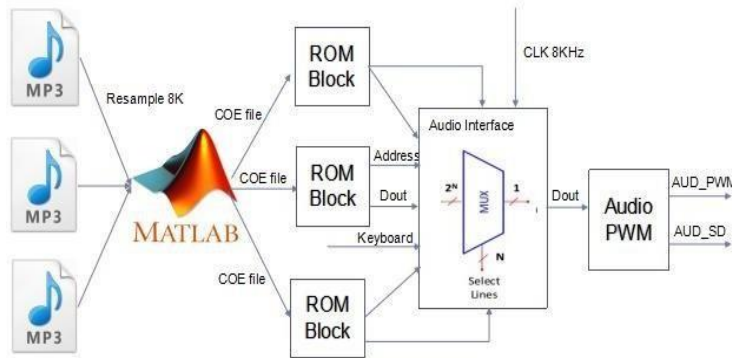


Fig.5. Audio interfacing block diagram [11]

.coe file respectively. These coe files are inserted the block ROM to generate Addr and 8-bit data (Dout). Addr and 8-bit Dout are passed as the input for Audio Interface module. The Audio Interface operates like a multiplexer with control signal coming from keyboard and output from main module.

Push Buttons

Depending on its status, basically a button sends one of the two values. Usually, when pushed, logic 0 and logic 1 will be returned if un-pressed. Sometimes when you press the button it will not modify its state and other times it will alter its state several times for one button to press this phenomenon is called bounce. And this occurs because when pressing or releasing a button, the shift from one state to another has some intermediate states (as shown in Fig.6) that are deemed a press or release.

Image Generation

The images were made in paint in RGB format. The idea behind making the images was use very few colors so that we can save the memory. Also fewer colors mean that less gradation at boundary between different colors. All the screens were made in paint in JPEG format. Mat Lab script was used generate the image.mif file. This file was later saved as image.coe file which contained the RGB values of image encoded in decimal values. Each pixel was encoded in 4 decimal values in decimal format.

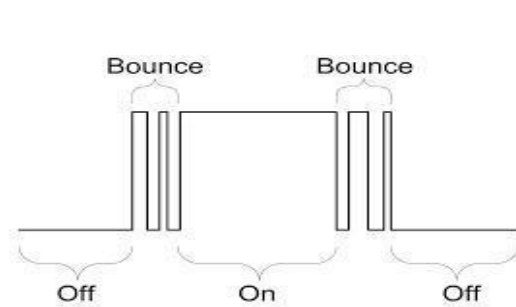


Fig. 6. De-Bouncing Diagram [12]
Generation [11]

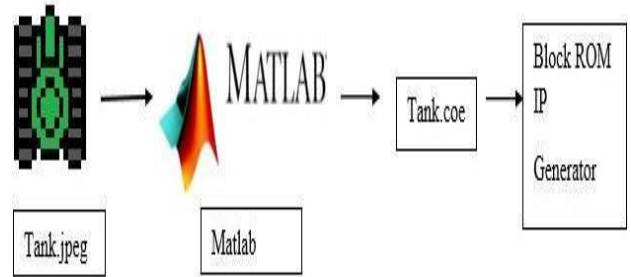


Fig. 7. Block diagram for Image

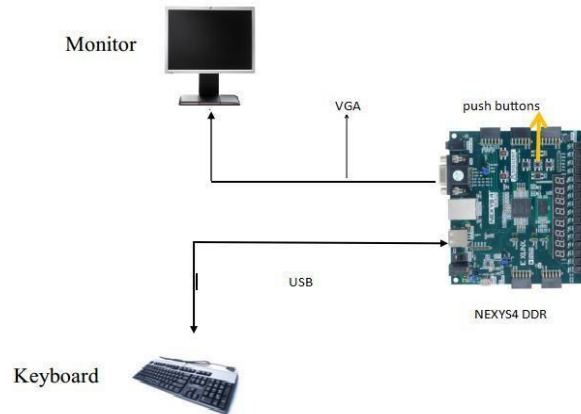


Fig. 8. Peripherals connected to FPGA used to develop a game [13]

3. CASE STUDY

Game-1: Snake Game

Snake is one of the most famous and oldest games in the game age, and now on the FPGA assessment kit we want to repeat the history of this game. And about the game, there is black side to the snake color yellow and black ground. The aim of the game is for the player to control a dot, square or object on a bordered plane where the snake begins as a dot and becomes a line that grows in size with the line itself as a main barrier. By altering its levels, we can play distinct levels of this game each level is distinct in the sense of obstacle situation. We can play this match at four distinct speeds of the snake. This game is becoming progressively popular due to easy rules and quick and vibrant games.

3.1.1 Design and Implementation Methodology:

The board assessment is based on strategic ideas such as pixel generation circuit mapping of objects, distinct speeds, distinct concentrations of barriers, VGA interfacing, keyboard interfacing and audio interfacing. In each and every game, score plays a major role, so we displayed the score by creating the segments on the screen like seven segment displays, and by using binary to BCD conversion logic we converted the binary number that comes from the counter (which counts score) to the BCD.

We initially stored 40 parts of the snake so that the numbers will increase whenever the snake eats food. And player loses the match if the section of the snake touches the border and gains a point if the section of food is touched by the snake head.

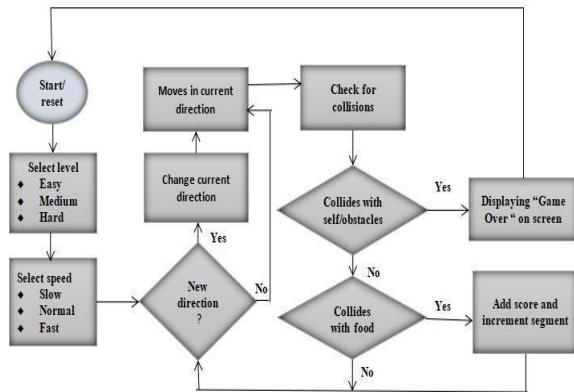


Fig. 9. Flow chart for Snake Game

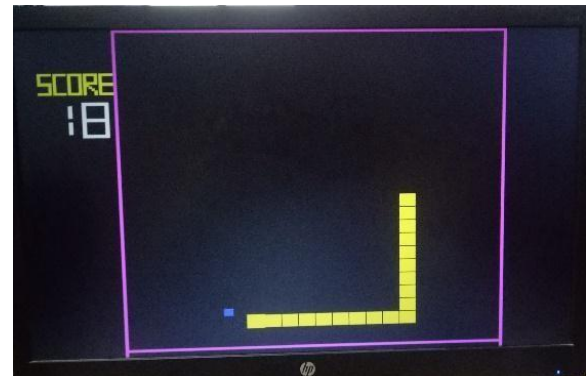


Fig. 10. Snake game display on VGA screen

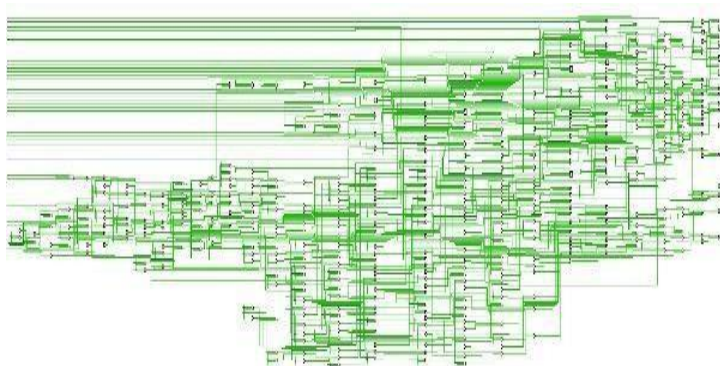


Fig. 11. RTL schematic of Snake game

Game-2: Spaceman Game

Space-Man is a game in which a spacecraft collects fuel tanks by avoiding mobile obstacles (clouds). Colliding with these clouds results in game-over. Collection of each fuel tank awards the player with one point. This game has two stages of 60 seconds each. The speed of clouds is high in the second stage when compared with that of the first one.

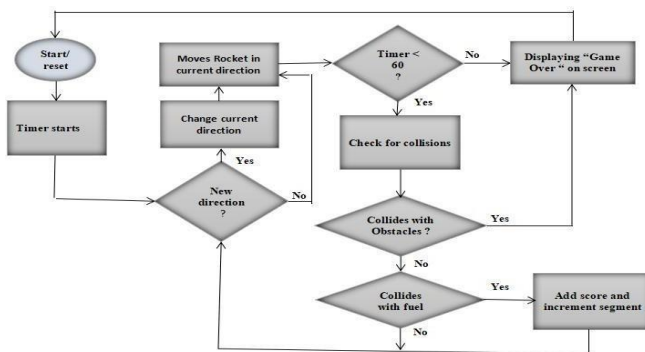


Fig. 12. Flow chart for Spaceman Game



Fig. 13. Spaceman game display on VGA

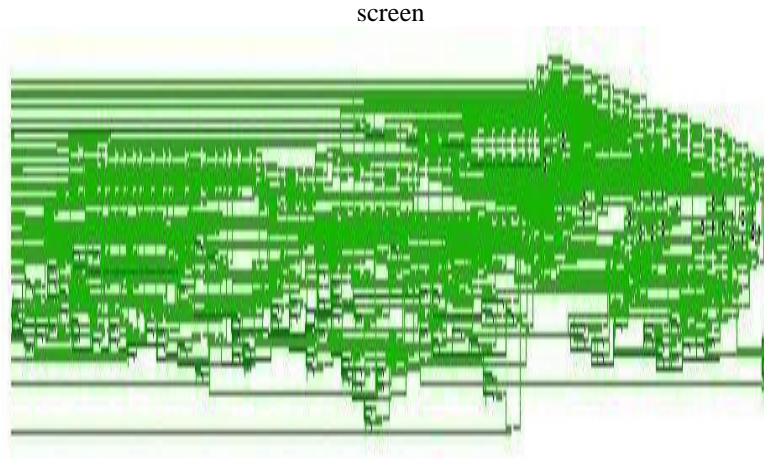


Fig. 14. RTL Schematic of Spaceman game

The goal is therefore to attempt to collect as many fuel tanks as possible, without colliding with the clouds, within the given time frame. The player who collects the highest number of fuel tanks emerges a winner.

Design and Implementation Methodology:

The board assessment is based on strategic ideas such as the mapped and bit-mapped pixel generation circuit object, VGA interfacing, keyboard interfacing and audio interfacing. Score plays a significant part in each and every match, so by generating the sections on the screen like seven segment displays, the score was displayed. And by using binary to BCD conversion logic, we converted the binary number that comes from the counter (which counts score) into the BCD. There are more opportunities for a player with more accessibility during the match to win the match.

Game-3: Pacman Game

Food is organized randomly in this match and we need to retrieve it within the time limit. But PAC-MAN shouldn't crash with any ghosts during the food collection as it leads to lose game play. If we collect one food, the score will increase to 1. Game lasts 60 seconds; the game will automatically be over later. Our objective is to retrieve all the food before the timer ends. The player's rating relies on the amount of ingredients colliding with the pacman and adding one point. The goal is therefore to attempt to collide with as many foods as possible.

3.3.1 Design and Implementation Methodology:

The board assessment is based on strategic ideas such as mapped and bit-mapped circuit object generation, VGA interface, keyboard interface and audio interface. Score plays a significant part in each and every match, so by generating the sections on the screen like seven segment displays, we displayed the score. And by using binary to BCD conversion logic, we converted the binary number that comes from the counter (which counts score) into the BCD. There are more opportunities for a player with more accessibility during the match to win thematch.

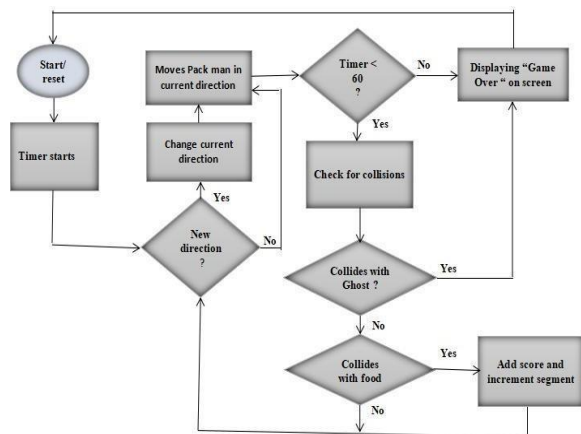


Fig. 15. Flow chart for Pacman Game



Fig. 16. Pacman game display on VGA screen

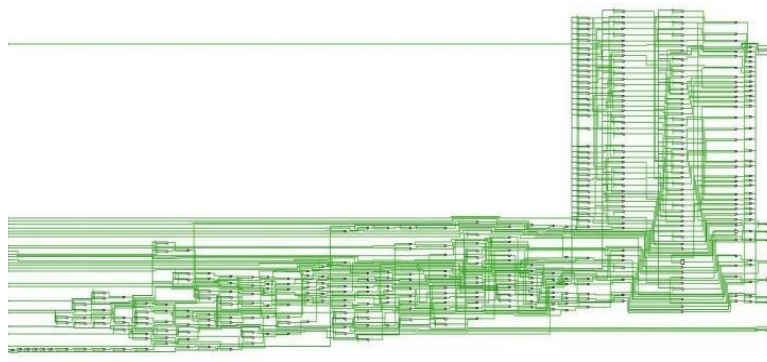


Fig. 17. RTL schematic for Pacman game

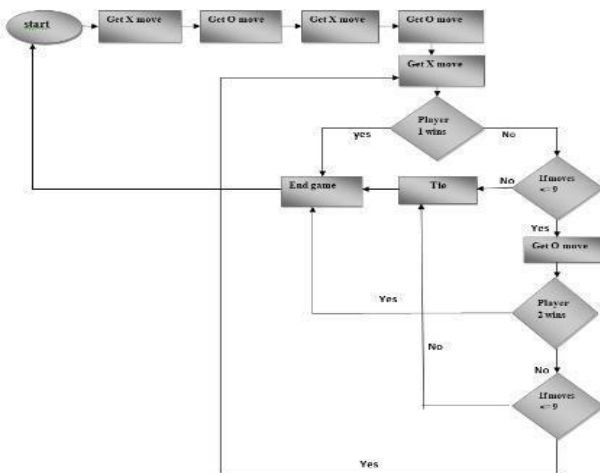
Game-4: TIC-TAC-TOE Game (Two Player Game)

Tic-tac-toe game for two players is a very popular game of paper and pencil in a 3x3 grid. The Player who in a diagonal, vertical or horizontal line makes the first three of their marks wins the match. 'X' is the player 1 value stored and 'O' is the player2 value stored. The two players play alternatively and store their respective symbols. For selection purposes, one block is developed. Only when a block push button interfacing is used to move the block up, down right and left positions will the player place their symbol [14].

Design and Implementation Methodology:

The assessment of the board is based on two players ' strategic game. The ideas implemented are object mapped pixel generation circuits and tile mapped system, VGA interfacing, keyboard interfacing, push interfacing button, audio interfacing and interfacing switches. There are more opportunities for a player with more accessibility during the match to win the match.

Fig.18. Flow chart for TIC-TAC-TOE Game. Fig.19. TIC-TAC-TOE game player1 win condition displaying on VGA



screen

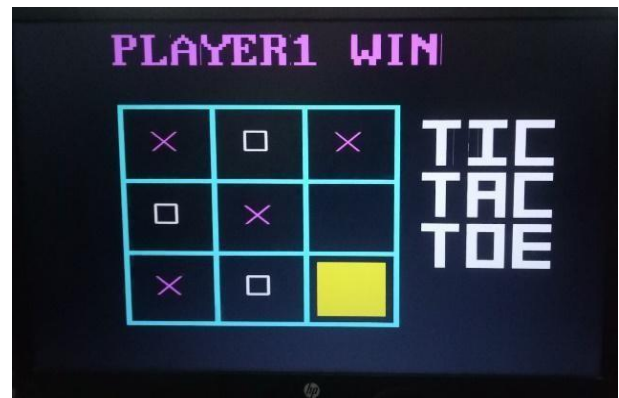


Fig. 20. TIC-TAC-TOE game player2 win condition displaying on VGA screen

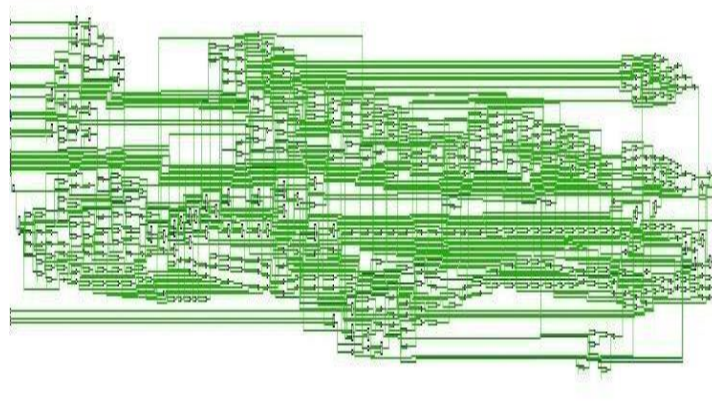


Fig.21. RTL schematic for two player Tic-Tac-toe game

Three Games in a Single Module:

Lastly, our primary objective is to create three games in one module. To do this, we need to instance three games in one module and take one output from the Game module as the final output at a specific stage of moment depending on the choice. We can play any match in this manner at any moment. We can select only one module at a time in this module in which three games are installed, and this selection process of games cannot be achieved using loops and if circumstances. Because we cannot install a module in loops, conditional operators and in blocks at all times in Verilog. But by using control signals, this can be achieved. Here the control signal works in such a manner that the corresponding control signal is enabled whenever a specific key is pushed on the keyboard and the respective outputs of the game module are taken as final inputs.

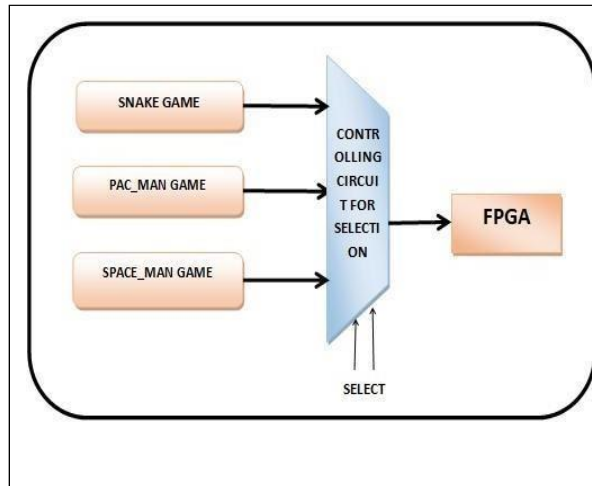


Fig.22. Instance of four games on single module games in a single module
Game-5: Battle Tanker

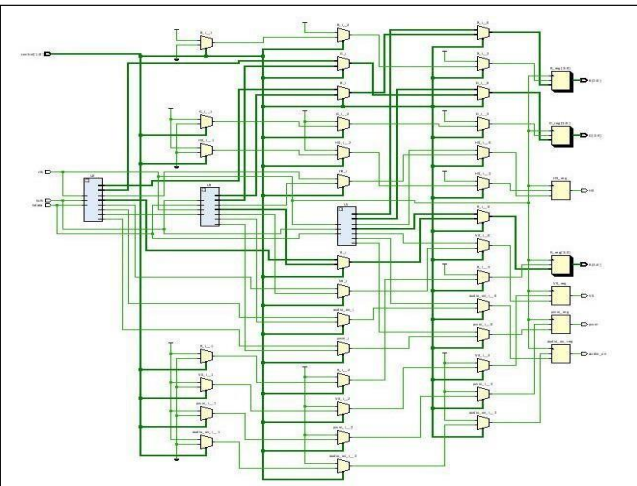


Fig. 23. RTL schematic for three single player

Battle Tank is a two player game in which players get to operate one tank each in the battle field. The objective of the players is to aim and shoot the opponent. The player who shoots his opponent first knocks his opponent down and eventually wins the round. There are totally three rounds in the game and the player wins the game on best of three criteria. The players operate their respective tanks in the battle field using the assigned keys on the keyboard interfaced with FPGA.

3.6.1 Design and Implementation Methodology:

The board assessment is based on strategic ideas such as image generation, VGA interfacing, keyboard interfacing and audio interfacing. The game of Battle Tanks consists of two tanks who are trying to shoot each other. The player 1 (green tank) tries to shoot player 2 (red tank) with a bullet. At the beginning of the game, the tanks are placed at different positions in map. ‘Space-bar’ and ‘Enter’ are the keys assigned to fire, for player 1 and player 2 respectively. The map consists of multiple brick walls which help the players to hide from each other or to dodge the shell. The player who succeeds in hitting the opponent wins the round.

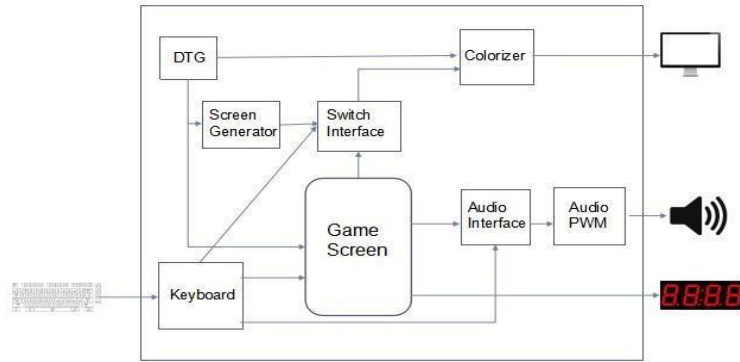


Fig.24. Block diagram for Battle tanker game [11]



Fig. 25.First screen for Battle Tanker game Tanker game

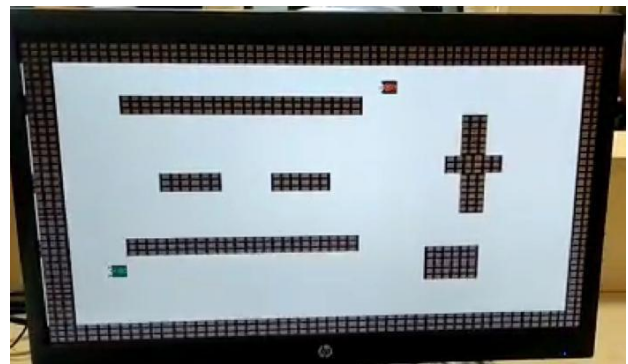


Fig. 26.Game screen for Battle

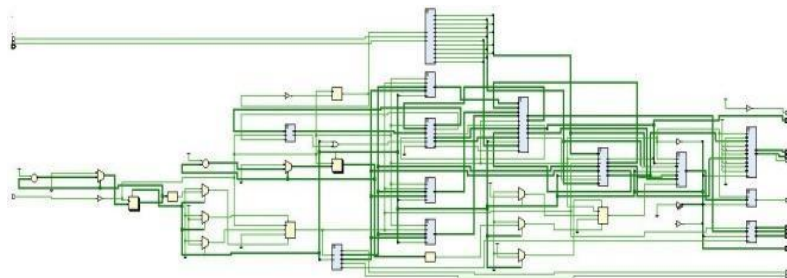


Fig.27. RTL Schematic for Battle Tanker game

The above block diagram shows the overall integration of all the modules used to design this project. It gives a general overview of the whole system. This project uses keyboard to control the movement and operation of the tanks. The Screen Generator regarding to first screen, and 2 game over screens take coe and output from DTG as the input. These coe files are generated by using a MATLAB script which converts a image into 12 bits coe. Screen interface take output from Screen Generator as a parameter input. The output of interface is driven into Colorizer module to display on a 640x480 resolution VGA display. Game Screen implements game logic battle which contains animation and score. The score is displayed on seven-segment led. The game also uses speaker interfaced

with FPGA for the audio output.

4.Design Complexity:

The past findings show the usefulness of FPGAs in board games to enhance energy effectiveness. However, FPGA hardware design also has some disadvantages. Hardware description languages (HDL) such as verilog enable a preliminary version to be written in HDL code at a moment similar to C language growth. Writing HDL code ready for conversion into an efficient hardware implementation requires great control over the design of digital logic, concepts of computer architecture and parallel computing.

Table-01 . FPGA resources utilization for Snake game

Resource	Utilization	Available	Utilization %
LUT	461	63400	0.73
FF	197	126800	0.16
BRAM	0.50	135	0.37
IO	38	210	18.10
BUFG	2	32	6.25
MMCM	1	6	16.67

Table-02. FPGA resources utilization for Spaceman game

Resource	Utilization	Available	Utilization %
LUT	1810	63400	2.85
FF	669	126800	0.53
BRAM	36	135	26.67
IO	19	210	9.05
BUFG	4	32	12.50
MMCM	1	6	16.67

Table-03. FPGA resources utilization for Pacman game

Resource	Utilization	Available	Utilization %
LUT	2579	63400	4.07
FF	560	126800	0.44
BRAM	36	135	26.67
IO	19	210	9.05
BUFG	3	32	9.38
MMCM	1	6	16.67

Table-04. FPGA resources utilization for Tic-Tac-Toe game

Resource	Utilization	Available	Utilization %
LUT	461	63400	0.73
FF	197	126800	0.16
BRAM	0.50	135	0.37
IO	38	210	18.10
BUFG	2	32	6.25
MMCM	1	6	16.67

Table-05. FPGA resources utilization for Battle Tanker game

Resource	Utilization	Available	Utilization %
LUT	185	63400	0.29
LUTRAM	24	19000	0.13
FF	154	126800	0.12
BRAM	1	135	0.74
IO	59	210	28.10
BUFG	2	210	6.25

5. Conclusions:

We finally developed three single player games and one two player game on FPGA. For the purpose of controlling the games, we have made keyboard interfacing so that we can play the games by using the keys on the keyboard. We did VGA interfacing to display the games on the VGA monitor and also did audio interfacing for the music while playing the games. Here our main objective was to make student learn FPGA through most entertaining games in the history of the computer game.

Acknowledgements

The author is grateful to Prof.G.V.R.Srinivasa Rao, Director, Rajiv Gandhi University of Knowledge Technologies (RGUKT- Nuzvid Campus) for extending his support to carry out the experimentation at our ECE Department Laboratories, Mr.V.Gireesh, Application Engineer, Synopsys Company-Bangalore, and Ms.Ch.Ratna, Physical Design Engineer, LeadSoc Technologies Pvt Ltd.- Bangalore for extending their technical support to publish this work.

References:

1. Divyanash A Thakar, Rikin J Nayak, Jaimin kumar B Chavda, Jay R Patel, Mit P Patel, "Design of Image Display Controller Component for VGA Interfaced Monitor using Zed Board", 2nd International Conference on Current Research Trends in Engineering and Technology, 2018.
2. Shivam Sharama, D.S.Gangwar (2017), A Review on Design and Implementation of FPGA Based VGA Monitor and PS2 Keyboard Interface Technique, Imperial journal of inter-disciplinary Research (IJIR), 3(6),pp.1339-1348.
3. Huan Minh Vo, "Optimizing power consumption using Multi-bit Flip- Flop technique in Tetris game on FPGA," 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, 2017, pp.530-533.
4. J. Olivito, J. Resano and J. L. Briz, "Accelerating Board Games Through Hardware/Software Codesign," IEEE Transactions on Computational Intelligence and AI in Games, vol. 9, no. 4, pp. 393-401, Dec.2017.
5. Pong P.Chu "FPGA prototyping by Verilog examples," Xilinx SpartanTM-3 Version, Hoboken, New Jersey, John Wiley & Sons, 1959, pp.309-347.
6. Digilent Inc., "Nexys4 FPGA Board Reference Manual," http://www.digilentinc.com/Data/Products/NEXYS4/Nexys4_RM_VB_2.
7. www.google.com.
8. <https://images.app.goo.gl/Y8UMiTpejQNaFKGk7>.
9. <https://images.app.goo.gl/kEDX71D3ZXzDPZe67>.
10. <https://images.app.goo.gl/fPYMzwXC2A2MBcAZ8>.
11. https://github.com/danghai/Battle_Tanks_Game.
12. www.google.com.
13. www.google.com.
14. FPGA4 STUDENT, "Tic Tac Toe Game in Verilog and LogiSim". <https://www.fpga4student.com/2017/06/tic-tac-toe-game-in-verilog-and-logisim.html>.
15. A. Kumar, S. Fernando and R. C. Panicker, "Project-Based Learning in Embedded Systems Education Using anFPGA Platform," in IEEE Transactions on Education, vol.56, no.4, pp.407- 415, Nov.2013.