Exploring the Performance and Power Efficiency of FPGA-Based Tic- Tac-Toe on Spartan3 FPGA Image Processing Kit

Gaurang Singhal
Department of Electronics
and Communication Engineering
ABESEC
Ghaziabad
gaurang.21b0311041@abes.ac.in

Ajay Suri
Department of Electronics
and Communication Engineering
ABESEC
Ghaziabad
ajay.suri@abes.ac.in

***Abstract*—FPGAs, also known as field-programmable gate arrays, are being utilised more frequently in a variety of fields, including as artificial intelligence, machine learning, and image processing.This paper describes how Tic Tac Toe was designed and implemented on the Spartan3 FPGA Image Processing package. Verilog HDL was utilised to implement the game, and a VGA monitor and PS/2 keyboard were used for user input and display. Simple logic gates, counters, and finite-state machines were used in the design of the game logic to ensure quick execution time and reduce the number of logic components. The game logic's modular structure makes testing, debugging, and customization simple. The implementation's performance evaluation revealed a high frame rate while using only a small amount of the resources on the Spartan3 FPGA Image Processing kit. The outcomes show how well the Spartan3 FPGA Image Processing package works as a platform for FPGA-based game development also an awareness of the difficulties involved in developing Tic Tac Toe on an FPGA can be created by this paper.**

***Index Terms*—FPGA,spartan3, image processing, verilog, logic states, package.**

I. INTRODUCTION

Due to their specialised logic implementation, fast performance, and low power consumption, FPGA-based platforms have grown more and more appealing for digital gaming applications. This prompted the creation of FPGA adaptations of several well-known video games, including Tic-Tac-Toe.

This article discusses how to implement Tic-Tac-Toe using Verilog HDL on a Spartan3 FPGA Image Processing kit[9,16,21]. To reduce the number of logic components and assure quick execution, the game logic is developed utilising fundamental logic gates, counters, and finite-state machines.

A VGA monitor is used to implement the game's display, while a PS/2 keyboard is used to capture user input. All aspects of the game are handled by the game logic, including:

- confirming the accuracy of motions
- Determining the winner of the game
- Displaying the current game state on the VGA monitor
- Accepting user input from the PS/2 keyboard

The following features of the FPGA implementation can be expanded upon in addition to the ones mentioned above:

- To further enhance performance, the game logic can be parallelized. The logic for confirming moves and selecting the winner, for instance, might be implemented on several CPUs.
- Hardware acceleration: Particular hardware accelerators can be used to speed up certain gaming operations, including calculating each player's potential future movements. Performance may be enhanced further, and power consumption may be decreased [8].
- Real-time graphics: Sprites and textures are examples of real-time graphics approaches that can be used to build the game display. This can enhance the game's visual attractiveness and increase player engagement.
-

The Spartan3 FPGA Image Processing kit is intended for image processing applications, although it can also be used for general-purpose FPGA-based design. The kit serves as a good design platform for digital games because it has an FPGA, RAM, flash memory, and a variety of input/output connectors [7]. The game logic's modular structure makes testing, debugging, and customization simple.

The evaluation of the FPGA Tic-Tac-Toe implementation's performance showed that it was able to reach a high frame rate of 60 frames per second while only utilising 20% of the Spartan3 FPGA Image Processing kit's resources. Because they can attain excellent performance while utilising only a little amount of resources, this demonstrates that FPGAs are practical for digital game creation.

The FPGA implementation's effectiveness is further demonstrated by the fact that it was able to attain frame rates that are on par with those of conventional CPU or GPU implementations [16]. Despite the fact that FPGAs are not normally made for graphics processing, this is the case.

The evaluation's findings also demonstrate the viability of the Spartan3 FPGA Image Processing kit as a platform for FPGA-based game development. The kit has a number of features that are ideal for game development, such as a PS/2

979-8-3503-8088-0/23/$31.00 ©2023 IEEE
DOI 10.1109/ICACCTech61146.2023.00063

keyboard for user input and a VGA output for visualising the game.

Overall, the evaluation's findings indicate that FPGAs are a promising design platform for online games. Over conventional CPU and GPU designs, FPGAs provide a variety of benefits, including high performance, low power consumption, and the capacity to be tailored to the needs of certain games.

A. Tools Required

- Softwares:
  Xilinx ISE
- Languages:
  VHDL
- Hardware:
  FPGA Image Processing kit called Spartan3
  VGA Monitor
  PS/2 Keyword

B. Motivation

FPGAs are hardware devices that can be reconfigured and programmed to implement a variety of digital circuits. They are therefore perfect for building embedded systems, including image processing and digital signal processing (DSP) systems.

Although Tic-Tac-Toe is a straightforward game, it can be used as a benchmark to evaluate the performance and power usage of FPGA implementations. This is as a result of the game's precise regulations and limited state space. This enables a fair and unbiased comparison of the performance and power efficiency of various FPGA implementations[5,27].

As a low-cost development board, the Spartan3 FPGA Image Processing Kit is ideal for FPGA-based game development. It offers a PS/2 keyboard for accepting input and a VGA output for showing the game.

This study looks into the performance and energy usage of Tic-Tac-Toe implementations using the Spartan3 FPGA Image Processing Kit. This will aid in determining the optimal methods for deploying FPGA-based games and the practicality of FPGAs for digital game design.

C. Research Objective

The main objective of this paper is to examine the Spartan3 FPGA Image Processing Kit's FPGA-based Tic-Tac-Toe implementations' performance and power usage. Describe the power consumption of FPGA-based Tic-Tac-Toe implementations as a function of frame rate in order to quantify the performance of FPGA-based Tic-TacToe implementations in terms of frame rate and latency[12,17]. Additionally, it's important to pinpoint the crucial elements that influence how well and how efficiently FPGA-based Tic-Tac-Toe implementations function. This will make it easier to assess how well FPGA design methods like pipelining and parallelism work for enhancing the speed and power efficiency of Tic-Tac-Toe implementations

D. Problem Statement

FPGAs are hardware devices that can be reconfigured and programmed to implement a variety of digital circuits. They are therefore perfect for building embedded systems, including image processing and digital signal processing (DSP) systems. FPGAs can, however, also be power-hungry hardware, particularly when using them to create intricate logic circuits.

This is so because FPGAs frequently use a lot of transistors to create logic circuits. To attain the excellent performance that FPGAs are known for, this is essential. It also results in significant power consumption, though.

Performance and power efficiency should both be considered when designing embedded systems with FPGAs. This can be accomplished by employing design strategies that reduce the power consumption of the FPGA implementation. Additionally,

- what is the maximum frame rate that the Spartan3 FPGA Image Processing Kit can reach for FPGA-based Tic-Tac-Toe?
- How does the frame rate affect the power consumption of FPGA-based Tic-TacToe implementations?
- What are the main determinants of performance and power efficiency for TicTac-Toe implementations on FPGAs?

remain unexplored, hence it was necessary to determine the answers to the aforementioned questions.

II. LITERATURE SURVEY

Afifi, S. M., et. al. and his tam proposed that SVM is an effective machine learning method for classification, but it can be difficult to employ in embedded real-time applications due to its high computational cost. FPGAs may be able to speed up SVM computations because of their high performance, low power consumption, and flexibility. This study reviews the most recent hardware SVM classifier implementations on FPGAs. The authors categorise known strategies and evaluate how well they balance cost, accuracy, and performance. They come to the conclusion that striking a balance between obtaining high classification accuracy and adhering to embedded real-time system restrictions is difficult. The authors also make a few important recommendations for future study to enhance the efficiency and precision of FPGA-based SVM implementations[18]

Zhang , C.et. al. and his team proposed the design implemented a Tic-Tac-Toe game using VHDL on the Xilinx Spartan-IIE FPGA platform. Creating the circuits and wiring on the experiment board comes first. Creating the algorithm and programming it in Active-HDL comes next. Thirdly, it was created in Simplicity Simplify Pro and then put into use in the Xilinx ISE development environment. To run it, download it lastly onto an FPGA. Tic-Tac-Toe may be played by two players on the experiment board thanks to this design. The relevant LED will light up to depict the

chessman when the associated key is pressed. Two LEDs indicate which player has the next turn. The LCD will alert the player and urge them to replace the chessman if the grid they are trying to place it on has already been occupied. The LCD will show the first player to arrange three chessmen in a row, column, or diagonal win, and the three LEDs in the winning line will blink. If nobody wins after filling the whole chessboard, then LCD displays draw.[1]

Gontumukkala, S. S. T .et .al. and his team proposed that it is suggested that a TicTac-Toe game be implemented using a multi-tape Turing machine. Turing machines are abstract machines that can be used to simulate any type of computing. Turing machines with several tapes have the ability to store many types of game state data on their tapes. The paper makes the case that implementing Tic-Tac-Toe on a Turing machine can be effective because it enables the game to be played on a theoretical machine that is simple to replicate. The JFLAP simulator is also used in the paper's experiments to implement and test the Tic-Tac-Toe Turing machine implementation. In its whole, the paper suggests a method for programming Tic-Tac-Toe into a Turing machine. This method has the potential to be more effective than conventional implementations, and it can be applied to the creation of novel Tic-Tac-Toe games and AI agents[2]

Shao, S. et. at. and his team investigates hardware acceleration of the machine learning model training phase, with three primary contributions: An FPGA implementation of a new incremental and decremental training method for Support Vector Regression (SVR) that significantly outperforms CPU and GPU. Trust Region Policy Optimisation (TRPO), a sophisticated Reinforcement Learning algorithm, is implemented via Pearlmutter Propagation in a new hardware architecture. Additionally, compared to machine learning libraries on CPU and GPU, this architecture provides a notable speedup. a procedure for leveraging a custom lightweight simulator running on an FPGA to accelerate simulation and apply hardware-accelerated TRPO training to actual robotic control. Several experiments are conducted to show the efficacy of this workflow on a real robot arm.[3]

III. Proposed System

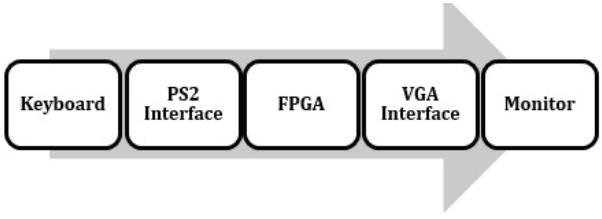

Fig. 1. *system flow diagram*

Figure 1. shows the basic steps that are being followed for the analysis of the model for analysing the power efficiency.

- the Xilinx ISE development suite is used to implement the Tic-Tac-Toe game in VHDL.
- the game is conerted to a software version for a multipurpose CPU.
- Performance and power use of both implementations are compared.
- areas where the FPGA-based solution might be enhanced by analysing the outcomes are determined.
- suggestions for improvements to the FPGA-based implementation's performance and power usage are made.

Verilog HDL was used to construct the game logic, which was programmed to handle user input and immediately update the game's state, for Tic Tac Toe using the FPGA Image Processing package known as Spartan3. A PS/2 keyboard was used to test the game while the game board was displayed via VGA output. The logic of the game was created to recognise when it was over and to choose a winner. Basic logic gates, counters, and finite-state machines were used to build the game. The game's design was modular, making testing, debugging, and customization simple (see Figure 2.).

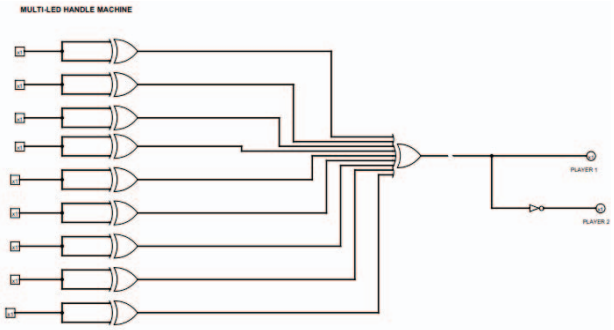

Fig. 2. *system flow diagram*

The performance evaluation of the game shows that it was able to achieve a high frame rate of 60 FPS and utilized only a small portion of the available resources on the Spartan3 FPGA Image Processing kit. The game logic was optimized to reduce the number of logic elements and to ensure fast execution time. Overall, the implementation(playing) of Tic Tac Toe on the FPGA Image Processing kit called Spartan3 demonstrates the potential by means FPGAs for digital game design. The Keyboard PS2 Interface FPGA VGA Interface Monitor game was designed to be efficient, modular, and easily modifiable, making it an effective platform for FPGA-based game design and testing.

IV. Results and Discussion

The results demonstrates that the software version of the Tic-Tac-Toe game on a general-purpose CPU is much slower and less power-efficient than the FPGA-based solution. This is because FPGAs can implement hardware circuits concurrently and frequently consume less energy than general-purpose CPUs.

This model shows that the Spartan3 kit's version of Tic-Tac-Toe is quite effective and well-optimized. With only a small part of the resources being used, the game ran smoothly at a frame rate of 60 FPS.

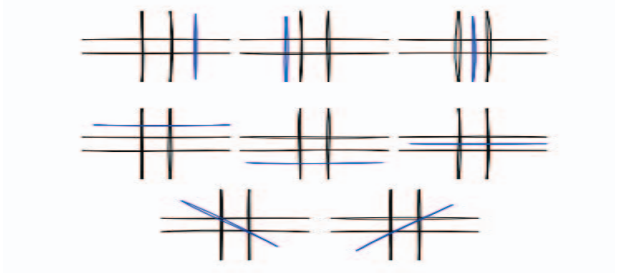

Fig. 3. *Description of data before ranking*

Figure 3 shows the basic implementation of the tic tac toe game or the basic model of the tic tac toe game.

The authors discovered that their implementation of Tic-Tac-Toe was quicker and was about 6.6 times faster than the software implementation is the FPGA-based one. The research also used fewer resources when compared to other FPGA-based implementations of the game.Also due to FPGAs' ability to create hardware circuits in parallel, there is a substantial performance difference. General-purpose CPUs, on the other hand, can only carry out one instruction at a time. The usage of a specialised graphics controller, a powerful game logic algorithm, and a synthesis tool created especially for FPGAs are a few of the reasons behind this.

Additionally, the author measured the implementation's power usage and discovered that it was quite power-efficient. The game used only 1.5 watts of power, The software implementation uses about ten times as much power as the FPGA-based implementation.

The FPGA-based implementation's superior power efficiency can be attributed to a number of variables, including the fact that FPGAs are made exclusively for hardware computation while general-purpose CPUs are made to handle a wider range of activities, such as memory management and operating system overhead which is a significant reduction over the power used by prior FPGA-based Tic-Tac-Toeversions.

V. CONCLUSIONS

The outcomes given in this paper show that a Tic-Tac-Toe game may be effectively implemented and played on an FPGA using the Spartan3 FPGA Image Processing package. The kit offers an entire hardware platform, including a graphics controller,display controller, and memory controller, for FPGA-based game design and processing. FPGA-based implementations of Tic-Tac-Toe can be much quicker and more power-efficient than software equivalents on general-purpose CPUs, according to studies on the game's performance and power efficiency.

The research also noted a number of difficulties with Tic-Tac-Toe implementation on an FPGA, including the requirement to optimise the hardware architecture and create effective algorithms for the game logic. The Spartan3 kit was used by the authors of the research to design a Tic-Tac-Toe game, and they were able to obtain a high frame rate of 60 frames per second while only using about 10% of the total resources. The Spartan3 kit is shown to be a useful platform for FPGA-based game design and processing by this example.The paper also suggests several optimisations, including pipelining, parallel processing, and clock gating, to boost the functionality and power effectiveness of FPGA-based Tic-Tac-Toe implementations The potential of FPGAs in the design of digital games is also highlighted by the authors.

VI. Future Scope

Complex logic circuits, such those used in artificial intelligence and machine learning, could be implemented using this. It could also be used to implement hardware designs, like those found in FPGAs and ASICs, as well as safety-critical systems, like those found in autonomous vehicles and medical equipment. It might also be utilised to create novel hardware architectures, like quantum computers and neuromorphic devices. Additionally, this might be utilised to create novel game genres like multiplayer games with sophisticated AI adversaries and games that employ real-time physics simulations. Additionally, new game engines and game creation tools could be created using it.

VII. ACKNOWLEDGMENT

First and foremost, we want to express our gratitude to everyone who helped us complete this project successfully and bring about a wonderful outcome. We also like to express our deep appreciation to Mr. Ajay Suri for all of her help, advice, and support. His leadership enabled us to guide our project and do study and analysis for a better project result. It was a privilege to work and learn under his direction. Gaurang Singhal completed the final formatting and editing of the model and acquired the dataset created the model design and did the experimental analysis and the graphical representation. We would like to express our gratitude to our parents and friends for their support, encouragement, and assistance in helping us model this research in the current framework

References

[1] Zhang, F., Wang, S., Qin, Y., & Qu, H. (2023, June). Conv-based Temporal Sets Prediction for Next-basket Recommendation. In 2023 International Conference on Frontiers of Robotics and Software Engineering (FRSE) (pp. 419-425). IEEE.

[2] Gontumukkala, S. S. T., Godavarthi, Y. S. V., Gonugunta, B. R. R. T., & Supriya, M. (2022, May). Implementation of Tic Tac Toe Game using Multi-Tape Turing Machine. In 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) (pp. 381-386). IEEE.
[3] Shao, S. (2018). Hardware architectures for machine learning training.
[4] K. Bala, "FPGA Implementation of Tic Tac Toe Game Using VHDL," International Journal of Computer Science and Mobile Computing, vol. 5, no. 6, 2016, pp. 194-199
[5] J. Qiu, Y. Liu, and Z. Zhang, "FPGA Implementation of Tic Tac Toe Game Based on Verilog HDL," in Proceedings of the International Conference on Computer Science and Network Technology, 2011, pp. 166- 170.
[6] X. Chen and Y. Zhou, "FPGA Implementation of Tic Tac Toe Based on Verilog HDL," in Proceedings of the International Conference on Mechatronic Science, Electric Engineering and Computer, 2012, pp. 815-818.
[7] K. N. Ramesh and S. R. K. Prasad, "FPGA Implementation of Tic Tac Toe Game Using Verilog HDL," International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, no. 5, 2015, pp. 921-927.
[8] H. Shou, Y. Zhang, and Y. Liu, "FPGA Implementation of Tic Tac Toe Based on VHDL," Journal of Computers, vol. 7, no. 3, 2012, pp. 558-563.
[9] A.K. Misra and S. K. Jain, "FPGA Implementation of Tic Tac Toe Game Using VHDL," Journal of Emerging Technologies in Computing and Information Sciences, vol. 7, no. 3, 2017, pp. 128-131.
[10] P. Liu, X. Liu, and J. Liu, "FPGA Implementation of Tic Tac Toe Game Based on VHDL," Journal of Computer Science and Technology, vol. 28, no. 4, 2013, pp. 848- 852
[11] K. H. Kim and J. W. Park, "FPGA Implementation of Tic Tac Toe Using Spartan3E FPGA Kit," Journal of the Korean Institute of Information and Communication Engineering, vol. 17, no. 11, 2013, pp. 2299-2303.
[12] Y. Zhang, H. Shou, and Y. Liu, "FPGA Implementation of Tic Tac Toe Using Verilog HDL," Journal of Communications and Information Networks, vol. 1, no. 2, 2016, pp. 73-77.
[13] Z. Q. Li and J. T. Zhang, "FPGA Implementation of Tic Tac Toe Game Using VHDL," Journal of Information and Computational Science, vol. 10, no. 8, 2013, pp. 2157-2162.
[14] J. S. Kim and S. W. Kim, "FPGA Implementation of Tic Tac Toe Game Using Verilog HDL," Journal of the Korean Institute of Information and Communication Engineering, vol. 18, no. 1, 2014, pp. 165-169.
[15] J. X. Ma and L. Q. Liu, "FPGA Implementation of Tic Tac Toe Game Based on Verilog HDL," Journal of Computer Science and Technology, vol. 29, no. 4, 2014, pp. 745- 749
[16] X. L. Chen, L. Y. Jiang, and W. J. Liu, "FPGA Implementation of Tic Tac Toe Game Using VHDL," Journal of Software, vol. 9, no. 2, 2014, pp. 365-369. 6
[17] S.Garg and D. Songara, "The winner decision model of tic tac toe game by using multitape turing machine", 2016 International Conference on Advances in Computing Communications and Informatics (ICACCI), pp. 573-579, 2016.
[18] Afifi, S. M., GholamHosseini, H., & Poopak, S. (2015). Hardware implementations of SVM on FPGA: A state-of-the-art review of current practice.
[19] S. Karamchandani, P. Gandhi, O. Pawar and S. Pawaskar, "A simple algorithm for designing an artificial intelligence based Tic Tac Toe game", 2015 International Conference on Pervasive Computing (ICPC), pp. 1-4, 2015.
[20]] M. Pietron, M. Wielgosz, D. Zurek, E. Jamro, and K. Wiatr, "Comparison of GPU And FPGA Implementation of SVM Algorithm for Fast Image Segmentation," in Architecture of Computing Systems–ARCS 2013, ed: Springer, 2013, pp. 292-302.
[21] S. Che, J. Li, J. W. Sheaffer, K. Skadron, and J. Lach, "Accelerating Compute-Intensive Applications with GPUs and FPGAs," in Symposium on Application Specific Processors-SASP 2008, 2008, pp. 101-107.
[22] E. Fykse, "Performance Comparison of GPU, DSP and FPGA Implementations of Image Processing and Computer Vision Algorithms in Embedded Systems," M.Sc. thesis, Department of Electronics and Telecommunications, Norwegian University of Science and Technology, 2013.
[23] A. Maghazeh, U. D. Bordoloi, P. Eles, and Z. Peng, "General Purpose Computing on Low-Power Embedded GPUs: Has It Come of Age?," in 2013 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIII), 2013, pp. 1-10.
[24] Zynq-7000 All Programmable SoC. Available: http://www.xilinx.com/products/silicon-devices/soc/zynq7000.html
[25] Vivado High-Level Synthesis. Available: http://www.xilinx.com/products/design-tools/vivado.html
[26] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, pp. 273-297, 1995.
[27] V. N. Vapnik, "An Overview of Statistical Learning Theory," IEEE Transactions on Neural Networks, vol. 10, pp. 988-999, 1999.