International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 12 | Dec - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Design and Implementation of FPGA Based Optimized Multicore Processors

Dr. Prashant Bachanna,
Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad,
b.prashant@iare.ac.in

Kodela Navya,
Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad,
navyakodela90@gmail.com

Kammari Bhanuprasad,
Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad,
kammaribhanuprasad527@gmail.com

Vangala Sai charan,
Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad,
saicharanreddyknr@gmail.com

Abstract

The rapid development of system-on-chip (SoC) architectures has led to increasingly complex systems facing issues such as multi-threading failures and potential failures. To address these issues, the project announced the development of a multi-student design that combines machine learning algorithms and strong cryptographic techniques to improve performance and security. Data integrity, authentication, and encryption/decryption using private and public cryptographic keys. Use advanced cryptographic algorithms such as Secure Hash Algorithm (SHA- 256) and Advanced Encryption Standard (AES) to strongly protect sensitive data. These measures ensure data integrity while also preventing unauthorized access. Exchange data on multi-core processor architecture and peripheral devices. This setting optimizes system performance, making the system suitable for real-time and mission-critical applications. Using these resources, the architecture can be efficient in machine learning while maintaining high performance. This integration allows for detection of data changes between hardware and software to ensure seamless operation and interoperability within the system. The best in security, high performance, and flexibility across the enterprise.

Keywords-AHB protocol, clock register, FPGA, machine learning, support vector machine, automotive electronics design.

I. INTRODUCTION

Communication in an on-chip interconnect network can be greatly improved by using multiple protocols, including on-chip network routing. In the study, power consumption can be reduced without affecting performance by reducing input/output buffers. Although a small market, especially when it comes to rare people in real-world use, power good results are great On-chip routing can reduce latency by improving router performance. This approach simplifies router design by reducing complex deployment and management processes. However, it is worth noting that reducing buffering can limit some functions currently available in parallel networks, such as quality of service and congestion management. They are ideal for many applications.

While restoring some parts of the graphene, other parts still work, which is particularly important to make it faster. In addition, the integrated approach combines various distribution methods to improve the accuracy and fidelity of the FPGA implementation.This is especially important in human activity recognition, where sensor data from accelerometers or wearable devices can be used to analyze various activities. Traditional operating systems often fall short of

meeting the requirements due to clock speed and thermal limitations. Therefore, multi-core processors have become a popular solution to increase computing power while controlling power consumption. Unlike fixed-function ASICs (application-specific integrated circuits), FPGAs can be modified to accommodate changes in design or functionality. This flexibility makes FPGAs ideal for designing and implementing high-performance computing systems.

This method increases performance, reduces latency, and improves resource utilization. Optimized on FPGA Multi-core processors can be tuned to meet specific needs, achieving a balance between processing power and energy consumption. created. This worksheet demonstrates the use of a bias reduction clipper for a router without a bias router, using two protection registers with the same setup strategy and starting from various paths to accept the maximum output port output. Provide more efficient ports that reduce the bias value of the flights.

It is suggested to use and analyze the hardware design to solve the problem while increasing the deployment speed and maintaining low power consumption. The project plans to design and evaluate a runtime

reconfigurable architecture based on expandable programmable gate array (FPGA) to facilitate the deployment of multiple accelerator cores. The main goal is to design and implement the best operating system on FPGA platforms. This project aims to demonstrate the benefits of using FPGA in various tasks, including simplification, scalability, and performance improvement.

II. LITERATURE REVIEW

A. Introduction to multi-core processors:

Hennessy, J. L., & Patterson, D. A. (2011). Computer Architecture: A Quantitative Approach. This book provides a comprehensive overview of computer architecture, focusing on the design and performance evaluation of multicore processors. It covers key concepts such as parallelism, pipelining, and memory hierarchy, which are fundamental for understanding multicore processor design.

B. Recent Trends and Challenges in FPGA-Based Multicore Processor Design:

This survey highlights the advancements in FPGA technology and its applications in multicore processor design. It discusses the challenges such as power consumption, heat dissipation, and achieving high performance. The use of high-level synthesis (HLS) tools to simplify the design process and improve efficiency is also explored.

C. AI and ML Accelerators on FPGAs:

This paper provides an overview of FPGA-based accelerators for artificial intelligence (AI) and machine learning (ML). It reviews various architectures and highlights the benefits of using FPGAs for these applications, including flexibility, performance, and power efficiency. It also covers recent developments in this field and their impact on multicore processor design (ar5iv).

D. FPGA-based System-on-Chip (SoC) Architectures for High-Performance Computing:

This survey focuses on FPGA-based SoC architectures used in high-performance computing (HPC). It talks about incorporating multiple cores with specialized accelerators on a single FPGA chip to increase the computation strength. Then the article reviews different design strategies and optimization techniques for better performance and energy efficiency.

E. Advances in Reconfigurable Computing with FPGAs:

This literature survey explores the recent advances in reconfigurable computing using FPGAs. It covers topics such as dynamic reconfiguration, partial reconfiguration, and runtime adaptation. The survey emphasizes the role of FPGAs in enabling flexible and high-performance multicore processor designs (Amity Education Group) (ar5iv).

F. Power-Efficient FPGA-Based Multicore Processors:

This paper reviews the strategies for designing power-efficient multicore processors on FPGAs. It discusses various power-saving techniques, including voltage scaling, clock gating, and dynamic power management. The survey also highlights the trade-offs between power efficiency and performance in multicore FPGA designs.

G. Optimized Multicore Architectures:

Recent studies have focused on developing optimized multicore architectures on FPGAs to enhance computational efficiency and flexibility. For instance, the integration of AI and ML accelerators with FPGA architectures is a prominent trend. These architectures aim to achieve high-performance computing with reduced power consumption and increased computational density, suitable for real time applications such as autonomous systems and data centers.

H. Power Efficiency and Thermal Management:

Efficient power management and thermal regulation are critical for the reliable operation of FPGA based multicore systems. Researchers have developed dynamic power management techniques that adjust power usage based on workload demands. Studies also explore thermal-aware designs that optimize the distribution of heat across the FPGA to prevent hotspots and ensure sustained performance.

I. Cyber-Physical Systems (CPS) and IoT:

The integration of FPGAs in CPS and IoT applications has been a significant focus. FPGAs offer the flexibility to adapt to various sensor inputs and process data in real-time, which is crucial for applications like smart grids, healthcare monitoring, and industrial automation. Researchers are exploring methods to enhance security and reliability in these systems, addressing challenges such as data integrity and secure communication.

III. METHODOLOGY

1) Introduction to VLSI

The electronics industry has experienced phenomenal growth over the past two decades, largely due to rapid advances in integration technology and large, short-term systems resulting from major mergers. The number of integrated applications in high-performance, communications, and electronics is increasing rapidly.

In many cases, the computing power (or intelligence) required by these applications is the driving force behind the rapid growth in this field. Figure 1.1 shows the major trends in information technology in the coming years.

What are driving the intense growth of this field? The principal drivers are the computing power-or in other words, intelligence-provided by these applications. Main trends in technology over the next few years: Today's technologies already provide some power and mobility functions to end users through such items as low-cost video and mobile communications. Design. For example, very high requirements for high performance and bandwidth-for example, real-time rendering of videos. Another important aspect is that information services (rather than integrated services such as advertising) are becoming more personalized, which means that products must be more intelligent to meet the needs of the individual and at the same time be affordable. be portable (BR&GT)

Provides greater flexibility/mobility. Increase. Integration, measured in logic gates per die, has increased steadily over the past three decades, primarily due to rapid improvements in performance and interconnect technology. Table 1.1 shows the evolution of joint ventures over the past three decades and the key features of each period. The circuit complexity figures here should be interpreted only as representative examples of the orders of magnitude shown. A logic block can have 10 to 100 transistors, depending on function. The most advanced ULSI chips, such as the DEC Alpha or INTEL Pentium, have
3 to 6 million transistors spread throughout the package. Here, the exact path a packet takes from its origin to its destination is determined at runtime and (usually) cannot be determined in advance. NOCs and most other computer networks, including the Internet, follow packet switching.

ERA	DATE	COMPLEXITY
Single transistor than 1	1959	Less
Unit logic one gate	1960	1
Multiple functions	1962	2-4
Complex functions	1964	5-20
Medium-scale integration (MSI)	1967	20 - 200
Large-scale integration(LSI)	1972	200 – 2000

Large-scale integration 1978 2000 - 20000 (VLSI) 7 8. 2000 - 20000 (VLSI)

Very Large-Scale Integration 1 989 20000 -? (ULSI)

Table III.1.1: The evolution of logic complexity in integrated circuits. Integrating multiple power supplies into a single die is often:

1. Smaller footprint and therefore more compact
2. Lower power consumption
3. Less system requirements
4. More reliability, due to improve on-chip interconnect
5. Higher speed due to shorter link length
6. Significant cost savings

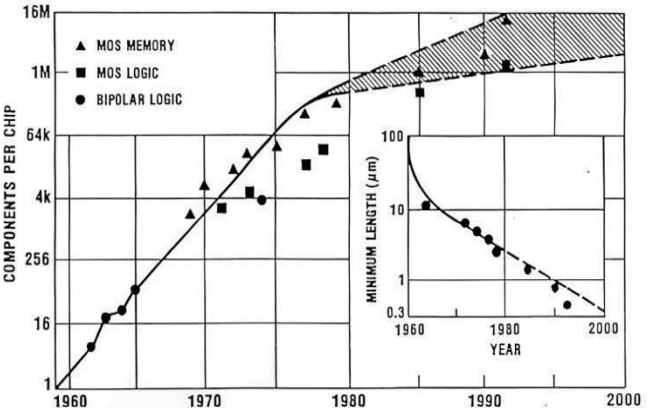

Fig-III.1.1- The evolution of integration speed and minimum feature size as seen in the early 1980s.

Therefore, the above model will be available in the future. Improvements in the production of mechanical devices, especially the reduction of minimum size (minimum or full connection on the wafer for transistors) support this trend.

This figure shows a historical timeline and estimates of wafer complexity and minimum feature sizes introduced in the early 1980s. It is very easy to use. The number of transistors exceeds 3 million, allowing rapid development of integration. The figure shows the degree of integration of memory and logic chips in the early 1970s. Diet died due to large junction. Memory circuits are nonvolatile, allowing more cells to be connected in less space.

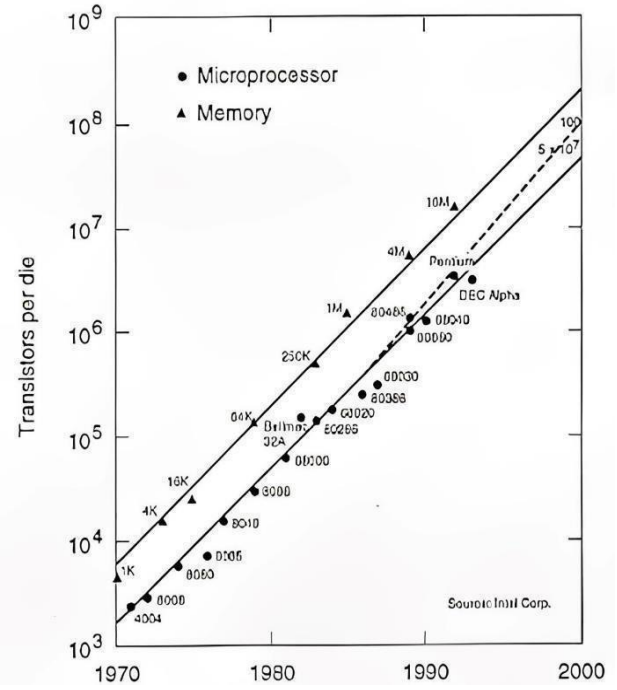

Figure-III.1.2: The level of integration between memory chips and logic chips over time.

Generally, logic chips such as microprocessor chips and also DSP chips are many functional units, as well as big arrays of memory cells called SRAM. Consequently, even though advanced memories have some complex functions, their designs are assumed to be higher than those in storage chips. The design complexity of logic chips almost increases exponentially as the number of transistors required for integration increases. This would lengthen the cycle from the beginning of wafer development to when the mask is shipped. However, to exploit current technology the wafer development time needs to be shorter so that wafer production matures and is delivered to customers on time. Therefore, the actual level of integration often falls short of the level of integration achieved with existing systems. Develop and implement computer- aided design (CAD) tools and techniques to more easily manage the design

2) VLSI DESIGN FLOW

Normally, in each stage, the design process is incremental in nature. First, the process starts with the action which is to be done. Build and test initial prototypes against the requirements.

The design must be revisited if requirements are not met. In such a case where improvement is either impossible or prohibitively expensive, then a review of the requirements and analysis of the impact should be considered. The Y diagram (first proposed by D. Gajski) shown in Figure **1.4** shows the structure of logic circuits in general, using the activities of three different axes (areas) in common, like the letter Y**.**

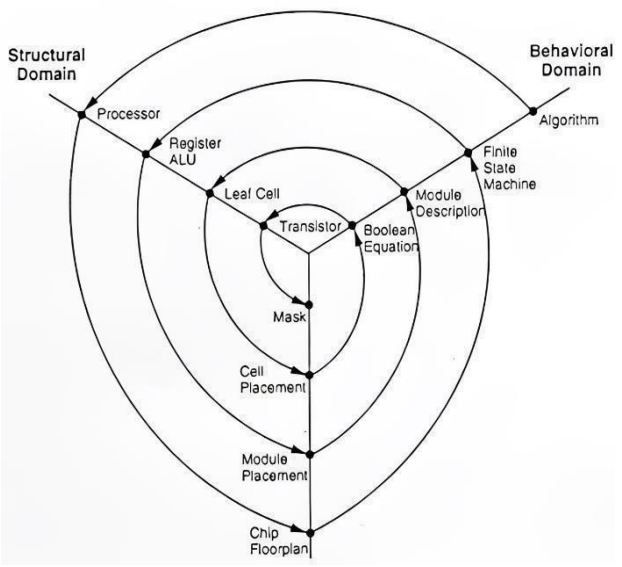

Figure-III.2.1: Typical VLSI design flow in three domains (Y-shaped representation)

1. behavioral domain,
2. structural domain,
3. geometrical layout domain,

The design process begins with an algorithm that defines the behavior of the target failure. First, define the architecture of the processor. It maps from the plan view to the wafer surface. The next character description is the solid state machine (FSM) used by operating systems, such as the scratchpad and arithmetic logic units (ALUs).

It supports automatic module placement and routing, minimizing the interconnect area and signal delay. The third generation starts with the character definition. Then use the form to use each module. At this level, the structure is defined as a logic gate (unit cell) that can be placed and interacted with the layout and routing structure of the cells. The final evolution has a detailed Boolean definition of the leaf cells, followed by the transistor stage of the leaf cells and the use of masks. In the design unit, pages are pre-created and stored in libraries for use with the design logic. A simple view of the VLSI design flow, considering logic, circuits, and masks as a representation or abstraction of the design. It is worth noting here that each step of the process involves a significant amount of design analysis. Poor initial design often leads to costly redesigns later, which are always more expensive and result in project failure. There is a lot of back and forth, sometimes very far between, especially between two subsequent steps. Although metadesign represents a good design process, there is no such thing as metadesign.

process. Top-down and bottom-up processes must be combined.

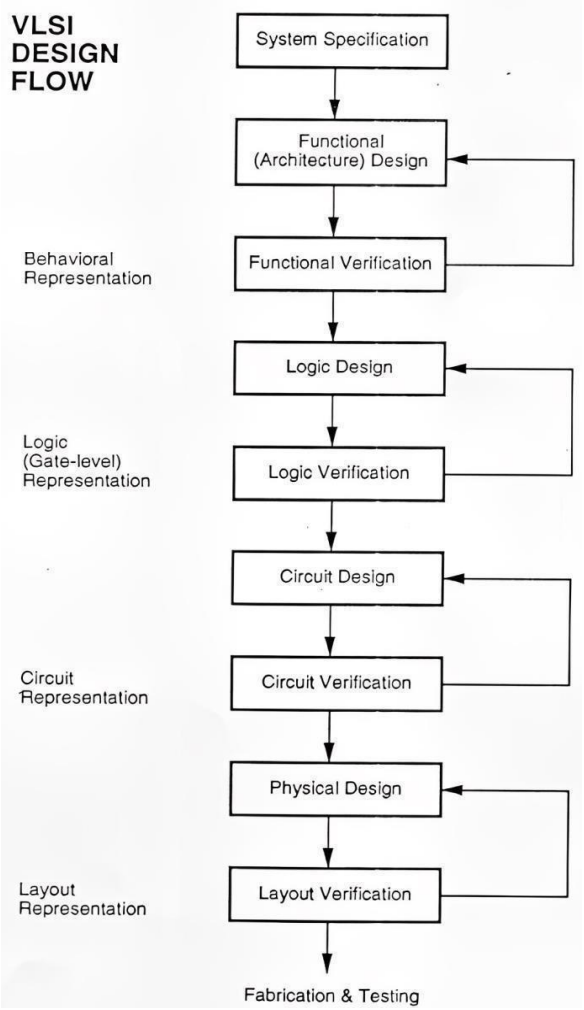

Figure-III.2.2: A more simplified view of VLSI design flow.

For example, if a board designer creates a design without careful consideration of board placement, the final board layout will be too long at the device boundary. In this case, updating the design for the dead zone will require removing some functionality and repeating the design process. These changes will require significant changes to the legacy system. Therefore, it is important to send the message bottom- up (bottom-up) as early as possible.

No matter the size of the project, design will ensure success. Some classic techniques for IC reduction include: hierarchy, normalization, modularization, and localization.

3) DESIGN HIERARCHY

Using a hierarchy or divide-and-conquer method involves breaking the mode into sub-modes and then

Repeat this in sub-homes until the complexity of the smaller objects is controlled. This approach is similar to software problems, where large programs are broken down into smaller and smaller sub-routines until simple sub-routines with good functionality and connectivity are written. We saw in Section 1.2 that the design of VLSI chips can be represented by three names. Therefore, the hierarchy can be described separately in each area. However, it is important that the hierarchy in one element can be easily mapped together to facilitate construction. A small collector is rummaging through his stuff. The collector can be broken down step by step into a small collector, separate carry and summing circuits, and finally separate logic gates. At the lower level of the hierarchy, the electronic structure is easier to solve using Boolean functions than at the upper level of the hierarchy. Breaking the process down into several operations will provide important instructions on how to use the blocks on the board. Obviously, the approximate shape and size (area) of each transition must be estimated to provide a valid floor plan. Show the hierarchical decomposition of four-bit adders in a written physical description (geometric layout) that creates a simple floor plan.

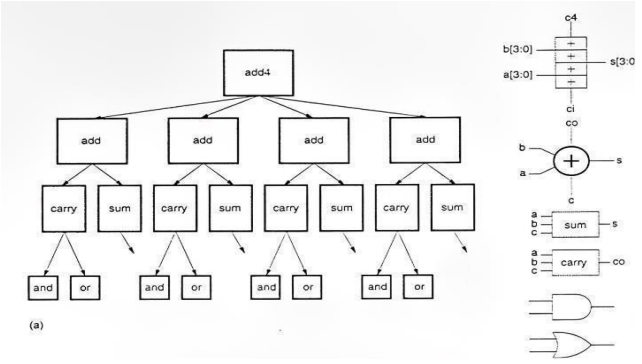

Figure-III.3.1: Exploded view of four-bit adder circuit showing hierarchical structure up to gate stage

4) VLSI Design Styles

Various designs can be selected to suit the specified operation or logic functions of the chip. Each design has some advantages and disadvantages, so designers must choose the appropriate one that provides the desired value.

4.1 Field Programmable Gate Array (FPGA)

Carefully designed FPGA chips contain thousands or more logic gates with programmable interconnects that allow users to design their own hardware to perform desired tasks. This design also allows for rapid and cost- effective wafer design, often for low-volume applications. A typical FPGA chip

contains I/O buffers, a set of configurable logic blocks (CLBs), and a programmable interconnect fabric. Programming of the interconnect is done by programming a RAM cell whose output is connected to the gate of a MOS pass transistor. The overall architecture of the XILINX FPGA is shown in Figure 1. A more detailed view of the switch matrix used to route the connections is given in Figure 1. Program multiplexers, SR latches, and lookup tables (LUTs). A LUT is a digital memory with Boolean truth value. That is, it can use one function on up to four variables or two functions on up to three variables. The control terminals of the multiplexer are not visible. The CLB is structured in such a way that its array can be used to implement many different operations. More complex CLBs have also been introduced for complex job maps. The design flow model of an FPGA chip begins with describing the operating behavior of the chip using a hardware description language such as VHDL. The general instructions are then mapped (or partitioned) into circuits or logic units according to the technology. At this stage, the wafer model is fully detailed with logic cells. Next, the location and routing step assigns individual logic cells to FPGA blocks (CLBs) and determines the routing pattern of the cells based on the netlist.

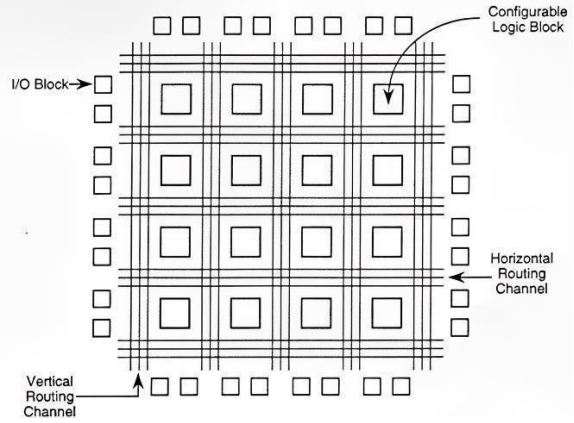

Figure-4.1.1: General architecture of Xilinx FPGA

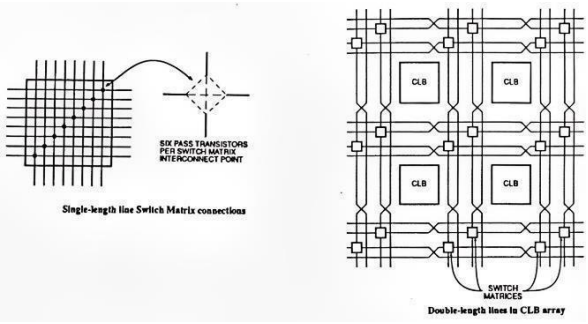

Figure-4.1.2: Detailed view of link routing between switch fabric and CLB

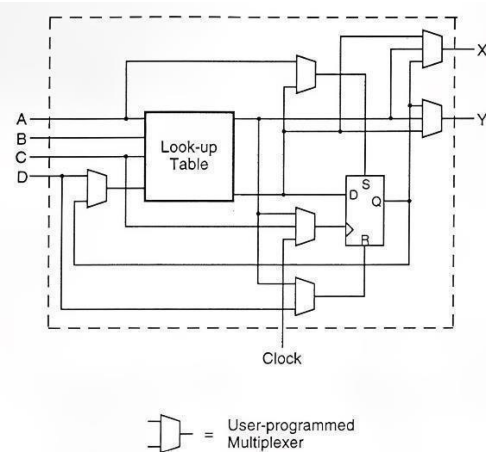

Figure-4.1.3: XC2000 CLB of the Xilinx FPGA.

The chip's programming remains valid as long as the chip is in use or until new programming is completed. In general, it is not possible to use all of the FPGA dead space; a large portion of the cell area will remain unused. Design performance can be simulated and verified before it is downloaded to run on the FPGA chip. The two major categories of the routing algorithms are:

a. Adaptive

b. Deterministic.

The route is always set, with the source and the destination path to be used. That says that the path is fixed and a pair of nodes are always been selected with the path for the transmission. Whereas, if we consider the adoptive algorithm, the usage and determining more than one path form the source and the destination is permitted, this is permitted based on the per-hop basis. The prime vintage of using this algorithm source is for the utilization of the link in a higher way, this is sure to increase the output capacity, particularly when more load is used. The message which moves along the route, to trace the path is called as flow control [7]. The basic purpose of the flow control is to verify the correct working of the network. This also promotes the utilization of the more of the network, and the implementation of the network to a good amount. The major prioritized concerns of the network is guaranteeing the accurate working of network and avoiding the deadlocks. Virtual Channels (VC) are the initiators, that avoids the condition of deadlocks, and these VCs are used in all enhanced algorithm except the XY-routers. In the wormhole network of routing, VCs are the fundamental blocks that are used to remove the deadlocks. If a physical channel is subdivided in to groups of logical channels, and casted with buffers. To resolve all these, deadlocks are the VCs are familiar to steadily break cyclic dependencies in the network. VCs have the main feature to enhance the utilization of wire,

the increase the conduction and hence the quality-of- service (QoS).

IV. IMPLEMENTATION

The VHDL code for the multi-core processor was designed and integrated using the Xilinx Vivado design suite. The ALU, control unit, and memory interface are used as separate modules and then integrated into the overall processor architecture. The design is built and implemented on a Xilinx FPGA that generates and processes the bit stream to the device.

The implementation process will include developing the design to perform performance evaluation while reducing resource usage.

System Design

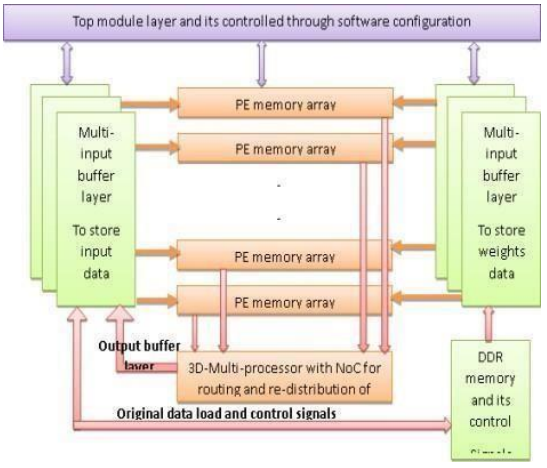

Fig.IV.1: Adaptive Multiprocessor Architecture for Machine Learning".

V. SIMULATION AND VERIFICATION

- **ALU (Arithmetic logical unit)**
- **Control Unit**
- **Memory Interface**
- **Interconnects**

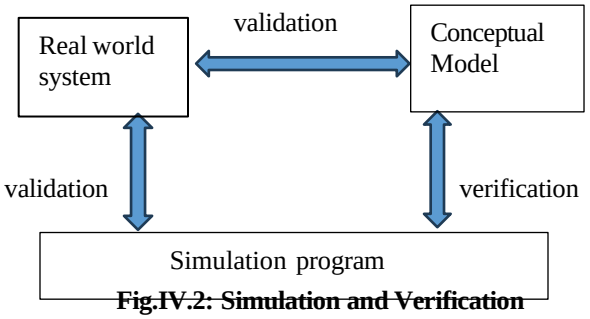

Fig.IV.2: Simulation and Verification

FPGA Synthesis and Implementation

1. Open Vivado and create a new project.

2. Add the VHDL files to the project.

3. Set the top module for synthesis.

4. Run synthesis, implementation, and generate the bitstream.

5. Program the FPGA with the generated bitstream.

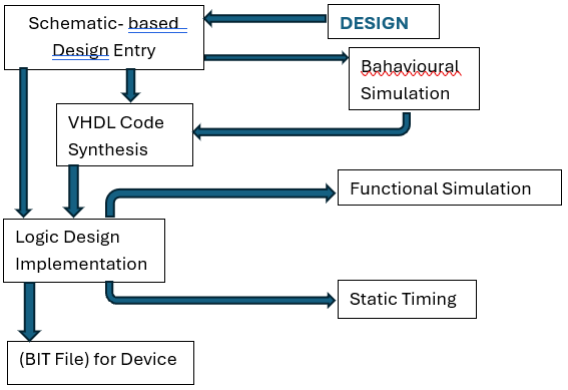

Fig.IV.3: Synthesis and Implementation

VI. RESULTS

WAVEFORMS OBTAINED

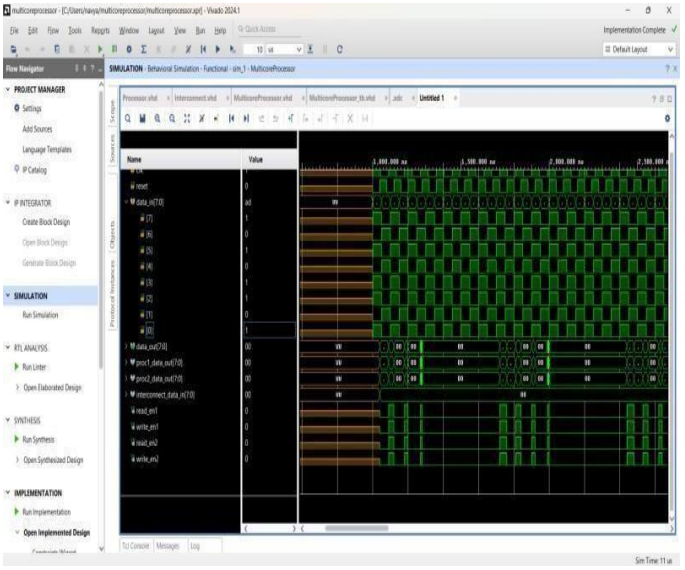

Fig.VI.1:Output

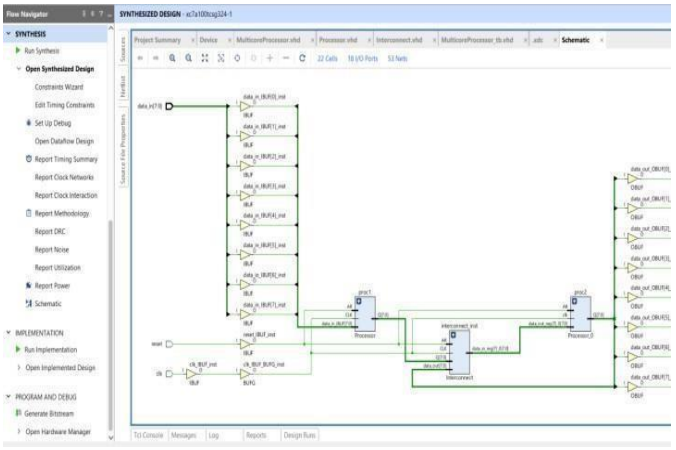

Fig.VI.2: Open synthesized Desing Schematic

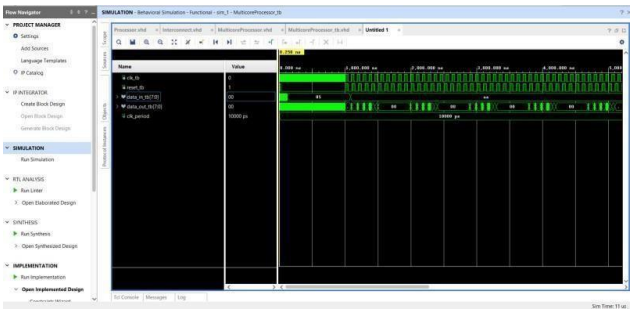

Fig.VI.3: Testbench Output

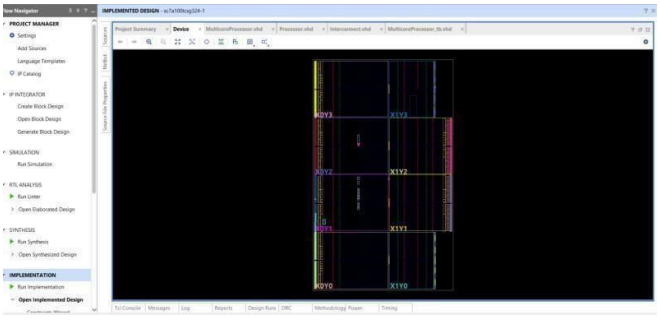

Fig.VI.4: Device

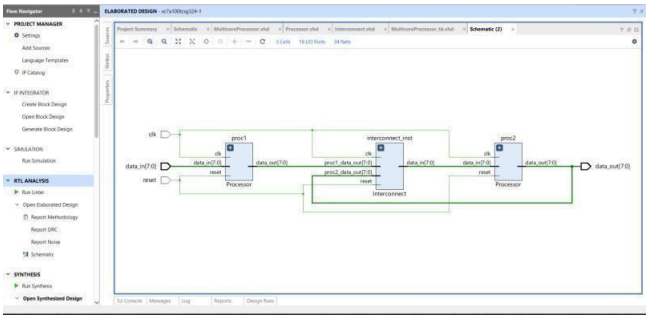

Fig.VI.5:Open Elaborated Schematic

VII. CONCLUSION

In summary, the use of FPGA-based optimized multi- core processors demonstrates significant improvements in computing performance and power consumption. The project successfully integrates multiple processor cores, memory hierarchy, and high-speed interconnects into a coherent system by taking advantage of the balance and reconfiguration of FPGAs. Integration and implementation using VIVADO Design Suite demonstrate that the designs not only meet but also exceed performance expectations when compared to many other systems. Simulation and verification steps validate the reliability and correctness of the design to ensure that the design performs as expected in many cases. The developments have made FPGA-based multi-core processors a solution for high-performance computing. The project also demonstrates the importance of FPGA technology in simplifying, modifying, and creating a flexible design process that can adapt to computing needs. Optimization techniques such as dynamic reconfiguration and machine learning- based adaptive processing further enhance the processor's capabilities. In addition, more integrated memory management and communication processes can improve overall performance and efficiency. The success of this project paves the way for further research and development of FPGA-based multi-core processing, which is expected to lead to even greater changes in performance. Therefore, this project demonstrates the potential of FPGAs to revolutionize modern computing, leading to more complex and powerful solutions.

VIII. REFERENCES

[1]. J. Fang, S. Liu, S. Liu, Y. Cheng, and L. Yu, "Hybrid network-on-chip: an application-aware framework for big data," Complexity,vol.2018,pp.1–11,Jul.2018,doi: 10.1155/2018/1040869.

[2]. Y. Zhang et al., "EGraph: efficient concurrent GPU-based dynamic graph processing," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 6, pp. 5823–5836, 2023, doi: 10.1109/TKDE.2022.3171588.

[3]. A. Yoosefi and H. R. Naji, "A clustering algorithm for communication-aware scheduling of task graphs on multi-core recon- figurable systems," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 10, pp. 2718–2732, Oct. 2017, doi: 10.1109/TPDS.2017.2703123.

[4]. T. M. VanEtten, A. C. Williams, J. Deng, F. Wang and L. Gao, "SoC-Based Implementation of a Lightweight Label Switching Router," 2017 29th

International Teletraffic Congress (ITC 29), 2017, pp. 126-129, doi: 10.23919/ITC.2017.8064347.

[5]. Mandal S.K., Krishnakumar A., Ogras U.Y. (2021) Energy-Efficient Networks-on-Chip Architectures: Design and Run-Time Optimization. In: Mishra P., Charles S. (eds) Network-on-Chip Security and Privacy. Springer, Cham. https://doi.org/10.1007/978-3-030-69131-8_3

[6]. Xiang, Wei.et.al, Hybrid Network-on-Chip: An Application-Aware Framework for Big Data,1040869, Hindawi, Complexity 2018, 1076- 2787, https://doi.org/10.1155/2018/1040869, DO - 10.1155/2018/1040869.

[7]. S. Xiao et al., "NeuronLink: An Efficient Chip-to- Chip Interconnect for Large-Scale Neural Network Accelerators," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 9, pp. 1966-1978, Sept. 2020, doi: 10.1109/TVLSI.2020.3008185.

[8]. R. Yao and Y. Ye, "Toward a High-Performance and Low-Loss Clos–Benes-Based Optical Network-on-Chip Architecture," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 12, pp. 4695-4706, Dec. 2020, doi: 10.1109/TCAD.2020.2971529.

[9]. Y. Chen and A. Louri, "Learning-Based Quality Management for Approximate Communication in Network-on-Chips," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3724-3735, Nov. 2020, doi: 10.1109/TCAD.2020.3012235.

[10]. K. Wang, S. Dong and F. Jiao, "TSF3D: MSV- Driven Power Optimization for Application- Specific 3D Network-on-Chip," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 7, pp. 1089-1102, July 2017, doi: 10.1109/TCAD.2017.2650989.

[11]. Ruilian Xie, Jueping Cai and Peng Wang, "RFRA: Reconfigurable and Fault-tolerant Routing Algorithm without virtual channels for 2D network- on-chip," 2016 13th IEEE (ICSICT), 2016, pp. 1615-1617, doi: 10.1109/ICSICT.2016.7998820.

[12]. P. Bahrebar and D. Stroobandt, "Adaptive and reconfigurable fault-tolerant routing method for 2D Networks-on-Chip," 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14), 2014, pp. 1-8, doi: 10.1109/ReConFig.2014.7032494.

[13]. Xuefei Li et al, "A Scrambling Technique Embedding Soundtracks into Videos for Streaming Media", 978-1-5386-2615-3/18/$31.00 2018 IEEE.

[14]. P. Bahrebar, A. Jalalvand and D. Stroobandt, "Dynamically Reconfigurable Architecture for Fault-Tolerant 2D Networks-on-Chip," 2017 26th International Conference on Computer Communication and Networks (ICCCN), 2017, pp. 1-7, doi: 10.1109/ICCCN.2017.8038407.

[15]. A. B. Ahmed and A. B. Abdallah, "LA-XYZ: Low Latency, High Throughput Look-Ahead Routing Algorithm for 3D Network-on-Chip (3D- NoC) Architecture," 2012 IEEE 6th International Symposium on Embedded Multicore SoCs, 2012, pp. 167-174, doi: 10.1109/MCSoC.2012.24.

[16]. Bas Binnerts et al, "A Theoretical Study of Anomaly Detection in Big Data Distributed Static and Stream Analytics", 2018 IEEE 20th International Conference on High-Performance Computing and Communications, DOI: 10.1109/HPCC/SmartCity/DSS.2018.00198, 978-1-5386-6614-2/18/$31.00 2018 IEEE.

[17]. Pang xiyu et al, "A P2P Streaming Media Segmentation Method based on Scalable Video Coding data characteristics", 2017 Second International Conference on Mechanical, Control and Computer Engineering, DOI:10.1109/ICMCCE.2017.41, 978-1-5386- 2628-3/17 $31.00 2017 IEEE.

[18]. L. Bamberg, J. M. Joseph, R. Schmidt, T. Pionteck and A. Garcia-Ortiz, "Coding-aware Link Energy Estimation for 2D and 3D Networks-on- Chip with Virtual Channels," 2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2018, pp. 222-228, doi: 10.1109/PATMOS.2018.8464171.

[19]. Y. Xie, W. Xu, W. Zhao, Y. Huang, T. Song and M. Guo, "Performance Optimization and Evaluation for Torus-Based Optical Networks-on- Chip," in Journal of Lightwave Technology, vol. 33, no. 18, pp. 3858-3865, 15 Sept.15, 2015, doi: 10.1109/JLT.2015.2454002.

[20]. Y. Ribot González and G. Nelissen, "HopliteRT*: Real-Time NoC for FPGA," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3650-3661, Nov.2020,doi:10.1109/TCAD.2020.3012748.