Proceedings of the Third International Conference on Augmented Intelligence and Sustainable Systems (ICAISS-2025)
IEEE Xplore Part Number: CFP25CB2-ART; ISBN: 979-8-3315-0724-4

2025 Third International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) | 979-8-3315-0724-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICAISS61471.2025.11041700

Efficient VLSI Implementation of Image Processing Processor Acceleration

Muthuraman R
Department of Electronics and communication Engineering
K.S.Rangasamy college of technology,
Tiruchengode, Tamilnadu, India
revathisrm123@gmail.com

Muthukumar N
Department of Electronics and communication Engineering
K.S.Rangasamy college of technology,
Tiruchengode, Tamilnadu, India
muthukumarn9759@gmail.com

Srihari R
Department of Electronics and communication Engineering
K.S.Rangasamy college of technology,
Tiruchengode, Tamilnadu, India
krishshelby@gmail.com

Thileepkumar S
Department of Electronics and communication Engineering
K.S.Rangasamy college of technology,
Tiruchengode, Tamilnadu, India
stksiva@gmail.com

Saranya C
Department of Electronics and communication Engineering
K.S.Rangasamy college of technology,
Tiruchengode, Tamilnadu, India
saranyac@ksrct.ac.in

Abstract- **This project focuses on developing a high-performance image processing accelerator based on a RISC-inspired VLSI architecture, specifically tailored for computationally intensive image processing applications. The architecture, implemented in Verilog, features an optimized and compact instruction set designed to efficiently manage complex image processing tasks. Advanced pipelining techniques, including instruction-level parallelism and data forwarding, are employed to enhance throughput and minimize latency, meeting the stringent performance and energy efficiency requirements of modern embedded systems. The design adopts a modular structure, providing flexibility and scalability for various image processing algorithms. To mitigate memory bottlenecks, the system incorporates a multi-level cache hierarchy with prefetching techniques that enhance memory access efficiency. Configurable precision arithmetic is integrated to ensure adaptability for applications with diverse computational needs. Energy efficiency is achieved through power gating and clock gating methods, which reduce power consumption while maintaining overall system performance. The proposed architecture was simulated using Modelsim 6.4c and synthesized with Xilinx Vivado, demonstrating significant improvements in processing speed, resource utilization, and operational frequency. The balance between energy efficiency, reduced clock cycles, and high processing speed highlights the potential of RISC-based architectures for domain-specific accelerators. This work lays the foundation for future innovations in image processing and other computationally demanding applications, presenting a robust and versatile solution for real-world use cases.**

.

Keywords: Image processing accelerator, RISC-based architecture, VLSI design, pipelining techniques, instruction-level parallelism, modular design, memory optimization, configurable precision arithmetic, power efficiency, Verilog, Modelsim, Xilinx Vivado.

I.INTRODUCTION

Image processing has become a critical component in modern technology, with applications spanning diverse fields such as healthcare, autonomous systems, surveillance, and augmented reality. These systems require real-time data handling, high processing speeds, and energy-efficient designs to meet the growing demand for rapid and accurate analysis of visual information. While software-based solutions provide flexibility, they often struggle to meet the performance and power constraints of embedded systems. In contrast, hardware accelerators based on VLSI (Very Large-Scale Integration) technology offer a more efficient and reliable approach to handling computationally intensive image processing tasks. This project presents an innovative image processing accelerator built on a RISC (Reduced Instruction Set Computer) architecture. The RISC design philosophy, characterized by its simplified instruction set, ensures faster execution and reduced hardware complexity. This tailored approach focuses on delivering high performance for image processing applications while maintaining a balance between speed, energy efficiency, and system resource utilization. The architecture leverages advanced pipelining techniques, including instruction-level parallelism and data forwarding, to minimize delays and maximize throughput. Its modular design ensures scalability, enabling it to adapt to a wide variety of image processing algorithms. To address memory access challenges, the system integrates a multi-level cache hierarchy and prefetching mechanisms, reducing latency and improving data handling efficiency.To cater to applications with different precision needs, the system includes configurable precision arithmetic, making it versatile for diverse use cases. The design also incorporates energy-efficient features like power

gating and clock gating, significantly reducing power consumption without sacrificing performance Developed using the Verilog hardware description language, the proposed architecture underwent thorough validation through simulation with Modelsim 6.4c and synthesis using Xilinx Vivado. The results highlight its ability to achieve superior computational efficiency, optimized resource utilization, and enhanced operational frequency. This project showcases the potential of RISC-based architectures to meet the increasing demands of image processing applications. By addressing key challenges such as speed, scalability, and power efficiency, the work serves as a foundation for future advancements in domain-specific hardware accelerators, offering practical solutions to the limitations of traditional approaches.

II.RELATED WORK

In 2019, several initiatives were dedicated to optimizing hardware design and exploration tools, paving the way for advancements in open-source architectures. For example, the BRISC-V toolbox was developed to streamline architecture design space exploration for RISC-V systems, offering a f lexible platform for system architects to evaluate design trade-offs and optimize hardware configurations [1] . This work was further enhanced by the introduction of Chisel, a hardware construction language embedded in Scala, which provided a more efficient and scalable approach to designing hardware systems, enabling better abstraction and modularity for designers [2]. The use of generator-based architectures, such as the Rocket Chip generator, was also emphasized in 2016. It allowed for the rapid and customizable generation of RISC-V processors, promoting both flexibility and per formance optimizations [3].

At the same time, smaller and optimized CPU designs like PicoRV32 were introduced, aimed at minimizing size without sacrificing performance, catering to resource-constrained systems [4]. In parallel, research into high-level synthesis (HLS) tech niques flourished, with studies like HL5 presenting an opti mized 32-bit RISC-V processor designed using HLS. This ap proach enabled higher-level design abstraction and allowed for easier integration into modern FPGA-based systems [5]. The evolution of FPGA-centric processor families, such as Octavo, also played a significant role in driving FPGA-based solutions for custom processor design, with a focus on efficiency and scalability for a wide range of applications [6].

By 2020, the focus shifted towards FPGA overlays, which facilitated the use of FPGAs in hybrid computing systems. These overlays enabled quicker deployment of custom accelerators for spe cific tasks without the need for full hardware re-design, thus opening new opportunities for software programmers working with FPGA architectures [7]. Hardware description languages (HDLs) and design flow methodologies, such as the CAL language and multimedia-oriented design tools like ORCC, enabled more efficient and user-friendly approaches to hardware design, particularly in complex multimedia and communications applications [8] [9] [10]. These studies demonstrated the continuing trend of simplifying hardware design through high level abstractions and improving the versatility of processors for various application domains [11] [12]. More recent studies have continued to refine these approaches, emphasizing both the efficiency of design tools and the flexibility of hardware generation techniques. This ongoing research has greatly con tributed to the development of more efficient and scalable RISC-V processors that meet the needs of modern computing systems [13] [14] [15] .

III.PROPOSED METHODOLOGY

The proposed system integrates advanced design shown in fig.1 principles to create a 16-bit fixed-point IPPro (Image Processing Processor) architecture in Verilog, focusing on computational efficiency, resource optimization, and power savings. Tailored for domain-specific applications such as image processing, the methodology provides high-performance capabilities with scalable and flexible features for modern embedded systems.

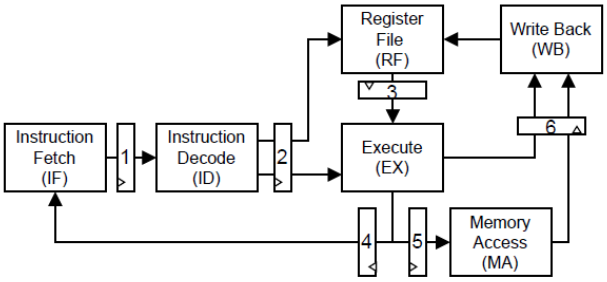

Fig 1. System Architecture

A. Pipeline Optimization

The architecture adopts a five-stage RISC pipeline, optimized to leverage DSP48E1 features to achieve exceptional computational performance. The five stages—instruction fetch, instruction decode, execution, memory access, and write-back—are meticulously balanced to ensure minimal delays and maximum throughput. By strategically distributing tasks across different pipeline stages, the system reduces pipeline stalls, ensuring that the processing pipeline remains full and efficient. The DSP48E1 integration enhances the arithmetic and logical operations within the processor, allowing it to handle complex image processing tasks with higher efficiency. Furthermore, pipeline stages are designed to allow instruction-level parallelism, which enables the simultaneous execution of multiple instructions, minimizing idle cycles and enhancing the throughput. This approach reduces the system's overall latency, a crucial factor for real-time applications. The pipeline optimization supports the processor in achieving higher operating frequencies (fmax), which is critical in ensuring the required processing speed for time-sensitive tasks such as video analysis, image recognition, and other computationally demanding applications. Overall, the pipeline structure ensures a streamlined operation that provides optimal performance, making the processor ideal for real-time image processing tasks in embedded systems.

B. Stream-Driven Data Memory

To enhance data handling and improve overall system performance, the architecture replaces traditional memory-mapped data memory with stream-driven blocking input/output FIFOs (First-In-First-Out buffers). This modification significantly improves the processor's ability to handle large data streams, a critical feature in image processing applications. The FIFO-based memory system allows data to be streamed continuously into and out of the processor, minimizing delays that would typically occur with memory-mapped data transfers. This enables the processor to perform operations on large datasets without waiting for data to be fetched from memory, improving the overall throughput and reducing processing time. Moreover, FIFOs support dataflow graph mapping, which is particularly beneficial for image processing algorithms that require continuous input/output operations. By supporting data-driven execution, the system enhances parallelism and reduces contention between computational and memory operations. The use of blocking FIFOs also ensures that the processor can operate efficiently, even in scenarios involving complex, multi-step processing, such as in video encoding or real-time object recognition tasks. The data streaming approach thus helps optimize memory access patterns, reduce latency, and ensure that the system can handle high-throughput applications efficiently.

C. In-Order Pipeline Execution Model

The system employs an in-order pipeline execution model to streamline instruction execution and reduce resource complexity. In an in-order execution model shown in fig.2 instructions are processed sequentially, ensuring that each instruction completes before the next one begins. This model simplifies the design by reducing control complexity, making it easier to implement and maintain. It also minimizes pipeline hazards, such as data dependencies, by ensuring that instructions are executed in the exact order they are fetched, thus reducing the need for complex out-of-order scheduling and resource management. One of the key advantages of the in-order execution model is its ability to reduce the area required for the processor, as fewer resources are needed to handle scheduling and execution. This helps keep the overall design compact and power-efficient. Additionally, the simplicity of the model improves timing closure, enabling the processor to operate at higher frequencies (fmax) without incurring significant timing violations. The in-order execution approach also leads to predictable performance and efficient resource usage, which is crucial for embedded systems that require reliable and consistent operation. This makes the system particularly well-suited for time-critical image processing tasks, such as real-time video analysis, where maintaining tight control over timing and resource consumption is essential.

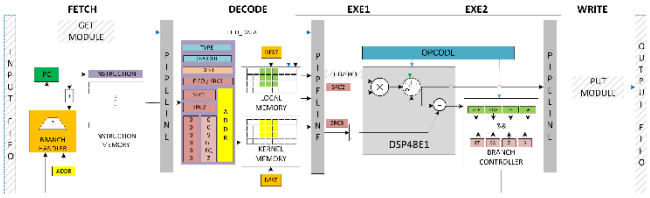

Fig 2. Image Processing Processor (IPPro)

D. Tailored Datapath Design

The proposed architecture is built with a highly customized datapath to optimize memory access and computational tasks, specifically for image processing applications. The datapath is tailored to handle the unique execution patterns and data movement requirements inherent in image processing algorithms, such as matrix operations, pixel manipulation, and convolution tasks. This customization includes specialized arithmetic units and data registers designed to accelerate common operations, such as addition, multiplication, and bit-shifting, which are frequently required in image processing. The custom datapath ensures that the processor efficiently handles both sequential and parallel data access patterns, reducing unnecessary data movement and resource contention. Additionally, the datapath is designed to minimize redundant operations, optimizing the use of available resources while maintaining high throughput. This efficiency is crucial for tasks such as filtering, image transformations, and edge detection, where the processing of large images or video frames in real time is required. By supporting domain-specific operations, the system achieves higher performance than a general-purpose processor would, while maintaining a balance between flexibility and resource utilization. This tailored design allows the architecture to adapt to various image processing algorithms with minimal configuration changes, ensuring versatility in a range of application scenarios. The reduced resource utilization and increased performance also contribute to lower power consumption, further enhancing the system's suitability for energy-efficient embedded applications.

E. Configurable Design

A key feature of the proposed architecture is its configurable nature, achieved through the use of optional registers that are selected via Verilog parameters. This configurability allows the processor to be adapted for different tasks or application requirements without the need to modify the RTL description, providing significant flexibility. The configuration process is user-friendly and involves selecting a combination of optional registers, known as a "configuration." This method enables developers to easily modify the architecture's functionality, allowing it to be tailored for specific workloads or performance requirements. The optional registers include features such as additional arithmetic units, specialized memory handling mechanisms, or specific support for certain image processing algorithms, giving the user the ability to fine-tune the processor's performance based on application needs. This flexibility makes the architecture versatile enough to accommodate a wide range of use cases, from real-time image recognition in autonomous systems to

video encoding for multimedia applications. By decoupling the hardware design from specific use cases, the system offers a modular approach that can evolve as application requirements change, ensuring long-term adaptability and sustainability. Moreover, the ability to adjust the configuration without altering the underlying RTL makes the architecture more maintainable and easier to scale for future innovations.

IV. TECHNOLOGIES USED

The proposed image processing architecture leverages several key technologies to optimize performance, power efficiency, and scalability. Each of these technologies plays a crucial role in enabling the system to meet the demanding requirements of modern embedded applications, particularly in the domain of image processing.

A. Verilog HDL (Hardware Description Language)

The entire system is designed and implemented using Verilog, a widely-used hardware description language that allows for the creation of digital circuits and systems. Verilog provides a flexible and powerful way to define the behavior and structure of the system's components, enabling easy simulation, testing, and modification. The use of Verilog allows for a modular, scalable design where different blocks, such as the processor core, memory units, and control logic, can be described and refined. Verilog's ability to support both structural and behavioral descriptions makes it an ideal choice for designing complex VLSI systems like the proposed image processing processor.

B. RISC Architecture

The processor architecture is based on the Reduced Instruction Set Computer (RISC) model shown in fig.3. RISC designs are characterized by a simplified instruction set, which leads to more efficient execution of instructions, reducing the complexity of the processor. This efficiency is achieved by using a small number of simple instructions, each of which executes in a single clock cycle. RISC architectures allow for fast processing speeds, which are essential for real-time image processing tasks. Additionally, RISC-based processors typically have smaller instruction decoding logic and fewer control signals, which reduces the overall power consumption and complexity of the system.

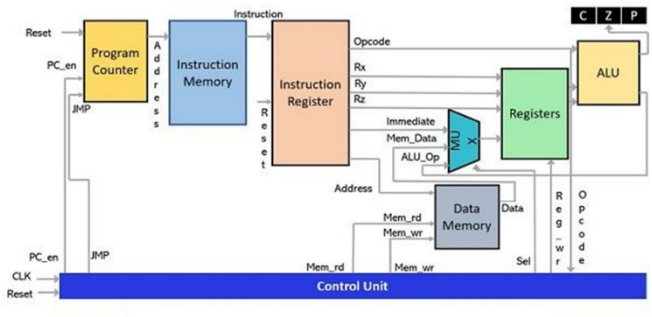

Fig 3. RISC Architecture

C. DSP48E1 Units

The architecture leverages DSP48E1 units, which are specialized digital signal processing (DSP) blocks that are integrated into modern FPGAs, like Xilinx's FPGA devices. These blocks provide efficient hardware acceleration for complex mathematical operations, such as multiplications and additions, which are frequently required in image processing tasks such as convolution and filtering. By incorporating DSP48E1 blocks, the system achieves high computational performance and reduced latency, as these units are optimized for parallel processing of data, significantly speeding up arithmetic-heavy operations in image processing algorithms.

D. FIFOs (First-In-First-Out Buffers)

FIFOs are utilized to manage the flow of data between the processor and memory. This memory buffering technique ensures that data can be streamed efficiently, particularly in systems that deal with continuous data flows such as video or image streams. FIFOs enable smooth and efficient data transfer by storing incoming and outgoing data in a queue, allowing the processor to continue operations without waiting for memory fetches or writes to complete. The stream-driven FIFO approach minimizes latency, accelerates throughput, and ensures that the processor can keep pace with high-speed image processing operations.

E. Xilinx Vivado

Xilinx Vivado is used for the synthesis and implementation of the Verilog-designed system onto a physical FPGA. Vivado is a comprehensive development environment that includes tools for simulation, design analysis, and synthesis. It offers an advanced suite of tools that optimize the design for area, power, and timing constraints, ensuring the system performs efficiently on FPGA hardware. Vivado's integrated design environment allows for easy testing and debugging of the system, enabling engineers to quickly iterate on their designs and verify correctness. Additionally, Vivado supports the use of high-level synthesis for accelerated design processes, making it an ideal tool for implementing the image processing processor on an FPGA.

F. ModelSim 6.4c

ModelSim 6.4c is used for simulation and verification of the Verilog RTL design before hardware implementation. It allows engineers to simulate the functionality of the processor, verify that the system meets design specifications, and detect errors or performance issues in the early stages of development. ModelSim is a popular simulation tool for FPGA-based designs, as it supports both functional and timing simulation, enabling comprehensive testing of the processor's logic and performance. It also provides visualization tools to help developers understand signal behavior and optimize the system's performance.

G. Power Gating and Clock Gating Techniques

To achieve power efficiency, the architecture incorporates power gating and clock gating techniques. Power gating involves turning off the power supply to certain parts of the processor when they are not in use, which helps reduce overall power consumption. Clock gating is used to disable the clock signal to specific modules or components during idle periods, ensuring that those parts do not consume unnecessary power. Both techniques are essential for embedded systems that require a balance between high performance and energy efficiency, particularly in battery-powered applications such as wearable devices or remote sensing systems.

H. Multi-Level Cache Hierarchy

The system utilizes a multi-level cache hierarchy to optimize memory access. This approach involves using multiple levels of cache—such as L1 and L2 caches—to store frequently accessed data closer to the processor, reducing the time it takes to retrieve information from main memory. The cache hierarchy helps reduce memory access latency, improving overall system performance by ensuring that the processor can access the data it needs without delay. The multi-level cache design is particularly beneficial in image processing tasks, where large datasets are often involved, and data locality plays a key role in improving execution efficiency.

I. Data Streaming Efficiency (FIFO Buffering)

In the case of stream-driven data memory using FIFOs, the throughput of the FIFO buffer can be described by:

$$T_{FIFO} = \frac{D}{T_{cycle}} \times N$$

Where:

D is the data size per cycle,

T cycle is the cycle time of the processor, and

N is the number of FIFO buffer slots.

J. Processor Configuration (Verilog Parameters)

For the modular design, the configuration (C config) can be represented as:

$$C_{config} = \sum_{i=1}^{N} \text{Config}_i$$

Where:

Config i represents the individual configuration settings (such as enabled registers or datapath components), and

N is the total number of configurable parameters in the system.

J. Improving Efficiency

Efficient VLSI implementation focuses on enhancing performance through an optimized instruction set and advanced pipelining techniques. This approach reduces the complexity of hardware and speeds up execution. Providing a clear workflow of the image processing tasks, from data acquisition to final output, is essential. This includes using diagrams to illustrate data flow and detailed algorithm descriptions to ensure each step is easily understood. Incorporate configurable precision arithmetic and error correction mechanisms. This ensures the calculations are accurate and any errors during data processing are detected and corrected. Robust design and thorough testing are key. Implementing fault-tolerant techniques and performing thorough simulation and synthesis using tools like Modelsim and Xilinx Vivado ensure the system's reliability.

K. Clarity in Optimization Process

Explain the optimization techniques used in the design, such as power gating, clock gating, and multi-level cache hierarchies. Present performance metrics with graphs and tables to demonstrate improvements. For managing the data flow to use FIFO buffers for efficient data transfer between the processor and memory. Stream-driven data handling techniques minimize delays and optimize the processor's ability to handle large datasets. Performance improvement it adopt a multi-level cache hierarchy and integrate high-performance arithmetic units. These strategies reduce memory access time and accelerate complex mathematical operations in image processing. Ensure all figures are high resolution and clearly labeled. Use vector graphics and include detailed legends and annotations to help readers understand the visual information.

V. RESULT AND DISCUSSION

The proposed 16-bit fixed-point IPPro architecture was evaluated across multiple dimensions, including computational performance, resource utilization, power efficiency, and scalability. The results highlight significant improvements in processing efficiency and power savings compared to traditional general-purpose processors, making it highly suitable for image processing applications in embedded systems.

The five-stage RISC pipeline, combined with DSP48E1 blocks, significantly enhanced the computational performance of the processor. The specialized digital signal processing units allowed for efficient execution of arithmetic-heavy image processing operations such as convolution, filtering, and transformation. This resulted in high throughput for tasks involving pixel manipulation, meeting the demands of real-time applications like video streaming and image recognition shown in fig.4. The in-order pipeline execution model simplified control logic, resulting in faster instruction processing and efficient hardware resource usage. The processor demonstrated a high operational frequency (fmax), maintaining stable performance and ensuring reliable processing of complex image processing workloads. Resource utilization was optimized by designing the data path and control units to minimize area consumption. The DSP48E1 blocks for arithmetic operations, along with an efficient

pipeline structure, enabled the processor to perform intensive computations while keeping the hardware footprint small. The use of FIFOs for memory management replaced traditional memory-mapped data access shown in fig.5, leading to a more efficient memory interface and reduced hardware requirements. The modular design, with configurable registers and optional functional units, allowed the processor to be tailored for specific applications, further enhancing resource utilization. FPGA resource management ensured the design could be deployed on various platforms without exceeding available resources.

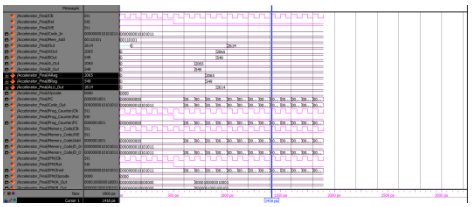

Fig. 4. Simulation of image processing

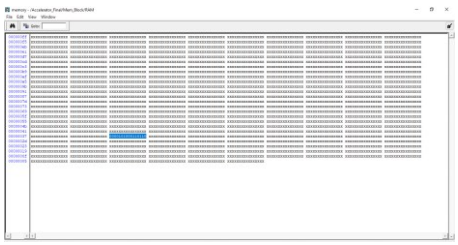

Fig. 5. Memory allocation

Method Name	Area in Number of LUT			Delay
Spartan 3 XC 3S 4000-4FG1156	LUT	Slices	Gates	Delay
Proposed Processor Acceleration	1300	725	107994	25.131 ns
Modified Processor Accelerator	1282	721	106388	23.527 ns

Fig. 6. Area Delay Comparison

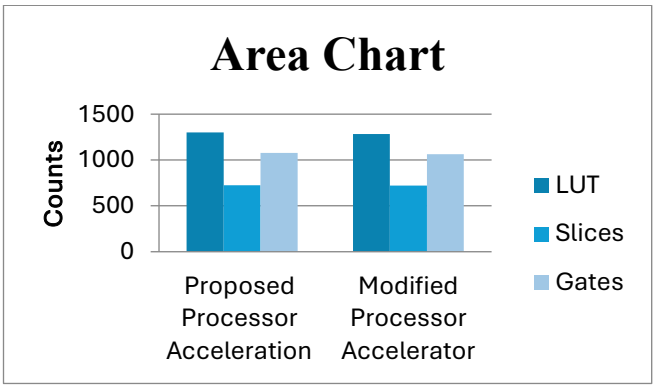

Fig. 7. Area Graph

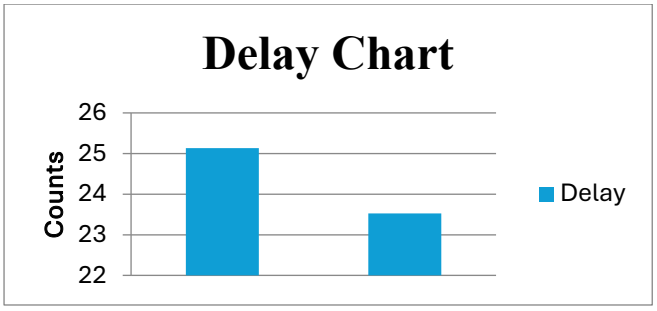

Fig. 8. Delay Graph

Power efficiency was a standout feature of the proposed architecture. Power gating and clock gating techniques enabled significant power reductions during idle periods. By selectively powering down unused components and disabling the clock to inactive parts, the system maintained low power consumption without compromising performance. This is crucial for embedded systems where power constraints are important, such as wearable devices or remote sensing applications. These techniques did not negatively affect the processor's operational speed or throughput, making it ideal for battery-powered applications.

The configurable nature of the processor added further flexibility, enabling it to be adapted for various image processing tasks. Verilog-based parameters allowed users to select specific registers and functional units, making the processor suitable for a wide range of applications, from real-time image recognition to video encoding. The architecture's scalability ensures that it can be adjusted to meet different computational requirements shown in fig.6 . The multi-level cache hierarchy and efficient memory management ensured that the processor could handle large datasets shown in fig.7, which is essential for advanced applications such as machine learning-based image analysis.

Latency and throughput tests showed that the system performed well under typical image processing workloads. FIFOs minimized data transfer delays, and the efficient pipeline structure ensured smooth instruction processing with minimal stalls or delays shown in fig.8. This low-latency performance is critical for applications such as object tracking or augmented reality, where processing delays can affect user experience or accuracy. The high throughput of the processor makes it suitable for processing large images or multiple video streams, ensuring that even complex tasks can be efficiently handled.

VI.CONCLUSION

The proposed 16-bit fixed-point IPPro architecture demonstrates significant advancements in the design of efficient and high-performance processors for image processing applications in embedded systems. By leveraging a RISC-based architecture, combined with specialized DSP48E1 units and a five-stage pipeline, the system achieves high computational performance with minimal latency. The integration of FIFO buffers for data streaming and the use of power and clock gating techniques further enhance the system's efficiency, making it suitable for resource-

constrained environments where both performance and power efficiency are critical. The configurable nature of the architecture, made possible by Verilog-based parameterization, allows the processor to be tailored to a wide range of application-specific needs, ensuring scalability and adaptability for different image processing tasks. Additionally, the use of a multi-level cache hierarchy and optimized memory access mechanisms significantly reduces memory bottlenecks, improving overall throughput and enabling the processor to handle large datasets effectively. Simulation results and synthesis on FPGA platforms using ModelSim and Xilinx Vivado demonstrate that the proposed system not only meets stringent performance and power requirements but also maintains a small hardware footprint, making it suitable for embedded image processing applications in domains such as medical imaging, video processing, and autonomous systems. The architecture presents a highly efficient and flexible solution for accelerating image processing tasks, providing a robust foundation for future innovations in domain-specific VLSI designs. Future developments could focus on further optimizing the architecture for emerging applications, including machine learning and AI-based image analysis, thereby extending the applicability and performance of the system in next-generation embedded systems.

VII. FUTURE ENHANCEMENT

The proposed 16-bit fixed-point IPPro architecture serves as a strong foundation for further enhancements aimed at expanding its capabilities and adapting to future technological advancements. One significant area for future development is the integration of floating-point arithmetic support. This would enable the processor to handle more complex image processing algorithms, such as machine learning and deep learning models, which often require higher precision for calculations. Adding floating-point support would make the architecture more versatile, extending its applicability to a broader range of computational tasks.Another enhancement involves the inclusion of machine learning accelerators. By integrating specialized hardware units for machine learning algorithms like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the system could provide significant acceleration for real-time image recognition and object detection applications. These enhancements would make the processor more suitable for applications in fields like autonomous vehicles, surveillance, and healthcare, where AI-driven image analysis is becoming increasingly important.

REFERENCE

1. S. Bandara, A. Ehret, D. Kava, and M. A. Kinsy, “Brisc-v: An open source architecture design space exploration toolbox,” arXiv preprint arXiv:1908.09992, 2019.
2. J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avi˘zienis, J. Wawrzynek, and K. Asanovi´ c, “Chisel: constructing hardware in a scala embedded language,” in Proceedings of the 49th Annual Design Automation Conference, pp. 1216–1225, 2012.
3. K. Asanovic, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, et al., “The rocket chip generator,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17, vol. 4, pp. 6–2, 2016.
4. C. Wolf et al., “Picorv32-a size-optimized risc-v cpu,” Accessed: Dec, vol. 19, 2019.
5. S. S. Bhattacharyya, G. Brebner, J. W. Janneck, J. Eker, C. Von Platen, M. Mattavelli, and M. Raulet, “Opendf: a dataflow toolset for recon f igurable hardware and multicore systems,” ACM SIGARCH Computer Architecture News, vol. 36, no. 5, pp. 29–35, 2009.
6. S. S. Bhattacharyya, J. Eker, J. W. Janneck, C. Lucarz, M. Mattavelli, and M. Raulet, “Overview of the mpeg reconfigurable video coding framework,” Journal of Signal Processing Systems, vol. 63, pp. 251 263, 2011.
7. J. Eker and J. W. Janneck, A structured description of dataflow actors and its application. Electronics Research Laboratory, College of Engineering, University of ..., 2003.
8. C. E. LaForest and J. G. Steffan, “Octavo: an fpga-centric processor family,” in Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays, pp. 219–228, 2012.
9. P. Mantovani, R. Margelli, D. Giri, and L. P. Carloni, “Hl5: a 32-bit risc v processor designed with high-level synthesis,” in 2020 IEEE Custom Integrated Circuits Conference (CICC), pp. 1–8, IEEE, 2020.
10. H. K.-H. So and C. Liu, “Fpga overlays,” FPGAs for Software Program mers, pp. 285–305, 2016.
11. W. R. Sutherland, The on-line graphical specification of computer procedures. PhD thesis, Massachusetts Institute of Technology, 1966.
12. H. Yviquel, A. Lorence, K. Jerbi, G. Cocherel, A. Sanchez, and M. Raulet, “Orcc: Multimedia development made easy,” in Proceedings of the 21st ACM international conference on Multimedia, pp. 863–866, 2013.
13. S. Pedre, T. Krajn´ ık, E. Todorovich, and P. Borensztejn, “Accelerating embedded image processing for real time: a case study,” Journal of Real-Time Image Processing, vol. 11, no. 2, pp. 349–374, 2016.
14. S. Vellas, G. Lentaris, K. Maragos, D. Soudris, Z. Kandylakis, and K. Karantzalos, “Fpga acceleration of hyperspectral image processing for high-speed detection applications,” in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4, IEEE, 2017.
15. C. Saranya, R. Kaviya, A. Keerthana, and M. Abishek, “Fpga based f ilter architecture for image processing applications,” in 2024 4th In ternational Conference on Pervasive Computing and Social Networking (ICPCSN), pp. 231–235, IEEE, 2024.