iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71372

Design and Implementation of an Embedded Image Processing System on Zynq ZedBoard: A VLSI Perspective

Anushka[1], Kanika Jindal[2], Ashutosh Kumar Singh[3]
Department of Electronics and Communication Engineering, Noida Institute of Engineering and Technology,
Greater Noida, India

Abstract: The increasing demand for real-time image processing in embedded systems has driven the integration of programmable logic with processing systems. This paper presents the design and implementation of an embedded image processing system on the Xilinx Zynq-7000 ZedBoard, targeting efficient computation through hardware-software co-design. Using the dual-core ARM Cortex-A9 and the programmable logic (FPGA), computationally intensive image processing algorithms such as filtering, edge detection, and thresholding are offloaded to custom hardware accelerators designed in VHDL/Verilog. The system is evaluated in terms of resource utilization, latency, and power consumption, offering a comprehensive VLSI-level insight into performance versus flexibility trade-offs. The results demonstrate the potential of Zynq-based systems in real-time vision applications, emphasizing design methodologies relevant to modern VLSI implementations.

Keywords: VLSI Design, Real-Time Processing, Xilinx Vivado, Hardware Acceleration, Verilog

I. INTRODUCTION

Image processing plays a crucial role in modern embedded applications, including object detection, medical imaging, surveillance, industrial automation, and autonomous navigation. These applications demand real-time performance, low power consumption, and compact hardware footprints—requirements that traditional software-based systems often struggle to meet. The computational complexity of even basic image processing tasks, such as filtering, edge detection, or morphological operations, requires high-throughput architectures capable of parallel execution and deterministic timing.

Field-Programmable Gate Arrays (FPGAs) offer a compelling solution by enabling custom hardware acceleration tailored to specific image processing algorithms. However, standalone FPGAs often lack the flexibility needed for system-level integration and software interaction. This challenge is effectively addressed by the Xilinx Zynq-7000 SoC platform, which combines a dual-core ARM Cortex-A9 processor with reconfigurable FPGA logic on a single chip. The Zynq ZedBoard allows designers to partition image processing tasks between software and hardware, optimizing both performance and efficiency.

This paper presents the design and implementation of an embedded image processing system using the Zynq ZedBoard, with a focus on leveraging hardware acceleration to meet the performance requirements of real-time image analysis. Core image processing algorithms—such as grayscale conversion, convolution-based filtering, and edge detection—are implemented using a hardware-software co-design approach. The programmable logic fabric is used to create custom processing pipelines, while the processing system manages high-level control and I/O operations.

.

II. LITERATURE REVIEW

Image processing on hardware platforms has garnered significant attention in recent years due to the increasing demand for real-time and power-efficient processing in embedded systems. The limitations of general-purpose processors in handling compute-intensive tasks have led to the widespread adoption of hardware accelerators such as FPGAs, which provide high throughput, parallelism, and reconfigurability. Several researchers have explored FPGA-based implementations of fundamental image processing algorithms. In [1], a Sobel edge detection filter was implemented on a Spartan-6 FPGA, demonstrating significant improvements in processing speed compared to its software counterpart. Similarly, [2] presented a hardware design for Gaussian filtering, showing reduced latency and efficient resource utilization on the Virtex-5 platform. However, these standalone FPGA implementations often lack flexibility and system integration features required in real-world embedded applications.

The introduction of hybrid SoC platforms such as the Xilinx Zynq-7000 series has enabled a new design paradigm where both hardware and software components coexist on a single chip. In [3], the authors utilized the Zynq ZedBoard to accelerate image thresholding and histogram equalization using a hardware-software co-design methodology. Their work highlighted the advantages of partitioning tasks between the Processing System (PS) and the Programmable Logic (PL), achieving better performance and modularity.

Further work in [4] explored a real-time object tracking system implemented on the Zynq platform. The researchers used custom IP cores for background subtraction and motion estimation, interfaced via the AXI bus. This demonstrated the suitability of Zynq devices for vision-based applications requiring high frame rates and low latency. Additionally, in [5], a convolutional image filtering application was implemented using High-Level Synthesis (HLS), showing how design productivity can be improved without sacrificing performance.

From a VLSI perspective, these works emphasize the importance of optimizing hardware for area, speed, and power. Techniques such as pipelining, parallelism, and hardware reuse have been commonly employed to meet strict timing and resource constraints. The works reviewed confirm that Zynq-based platforms offer a balanced trade-off between the flexibility of software and the performance of dedicated hardware, making them ideal for embedded image processing systems.

Despite these advancements, there remains a gap in systematic design methodologies that integrate multiple image processing stages into a unified pipeline optimized for both performance and VLSI constraints. This paper builds upon previous research by presenting a complete embedded image processing system implemented on the Zynq ZedBoard, focusing on modularity, performance optimization, and resource-efficient hardware design.

III.SYSTEM OVERVIEW AND ARCHITECTURE

The system architecture consists of the following key components:

1) Processing System (PS): ARM Cortex-A9 dual-core processor handles high-level control tasks, image loading, and interfacing with external peripherals (e.g., HDMI camera or SD card).

2) Programmable Logic (PL): Implements custom hardware accelerators for image processing tasks using VHDL/Verilog or High-Level Synthesis (HLS).

3) AXI Interconnect: AXI4-Lite and AXI4-Stream protocols are used to interface between the PS and PL for data transfer and control.

The design and implementation of the embedded image processing system on the Zynq ZedBoard follow a hardware-software co-design methodology that strategically distributes computational tasks between the Processing System (PS) and the Programmable Logic (PL). The system is designed to accelerate essential image processing operations by offloading time-critical tasks to the FPGA fabric while maintaining system control and data management through the ARM Cortex-A9 processor embedded in the Zynq SoC. The core of the implemented system is an image processing pipeline, which includes several fundamental stages: grayscale conversion, edge detection, image filtering, and binary thresholding. Each stage is implemented as an independent hardware module within the programmable logic, enabling pipelined and parallel execution that significantly improves performance compared to a software-only implementation. In the first stage, the input color image is converted to grayscale using a weighted sum formula, which reflects the human eye’s sensitivity to different wavelengths. This transformation reduces the complexity of subsequent processing steps by working with a single luminance channel instead of three separate color channels.

Following grayscale conversion, the edge detection module applies the Sobel operator, a widely used gradient-based method for identifying the edges within an image. The Sobel filter calculates the gradient magnitude in both horizontal and vertical directions using convolution kernels, emphasizing regions with high spatial frequency—typically corresponding to object boundaries. The hardware implementation of the Sobel filter is carefully designed to support streaming data with minimal latency by using line buffers and pipelined multipliers and adders.

To enhance image quality and reduce noise before edge detection, an optional filtering stage is introduced. This can include a Gaussian blur, which smoothens the image by attenuating high-frequency noise while preserving structural information. The Gaussian filter is implemented in hardware using a convolution window, where each pixel’s new value is calculated based on its neighbors using a weighted kernel. Efficient implementation requires careful optimization to minimize logic utilization and latency.

Finally, a thresholding module converts the filtered grayscale or edge-detected image into a binary image. In this stage, each pixel is compared to a predefined threshold value, and its intensity is mapped to either a black or white pixel. This stage is essential in applications such as object detection, segmentation, and pattern recognition, where binary images are often used for decision-making.

The interaction between the software and hardware components is facilitated by AXI4 interfaces, allowing high-speed data transfer between the PS and PL. The PS is responsible for loading image data, configuring hardware accelerators, and retrieving the processed output. Control software written in C/C++ using the Xilinx Vivado toolchain manages these tasks and orchestrates the operation of the pipeline. Throughout the development, Vivado Design Suite is used for hardware description, synthesis, and implementation. Custom IP cores are designed using either Verilog , and integrated into the block design using Vivado IP Integrator. Each module is tested individually through simulation before being deployed on the Zynq ZedBoard. The system is evaluated using test images, with performance metrics including processing time, resource utilization, and power consumption, offering a comprehensive insight into the system's VLSI-level efficiency.

IV. RESULTS AND DISCUSSIONS

The embedded image processing system implemented on the Zynq ZedBoard demonstrated a significant improvement in performance and efficiency, leveraging the hybrid hardware/software architecture of the Zynq platform. By offloading computation-heavy tasks to the FPGA, the system achieved real-time image processing with reduced power consumption and efficient resource utilization, which is crucial for embedded vision applications. In terms of processing time, the hardware-accelerated system showed a marked reduction in latency compared to a purely software-based approach. The FPGA implementation of key image processing tasks, such as grayscale conversion, Sobel edge detection, and thresholding, significantly sped up the processing pipeline. This allowed the system to handle images in real-time, providing faster results compared to traditional software-only implementations. The overall reduction in processing time was substantial, making the system viable for time-sensitive applications, such as live video analysis or real-time object detection.

The resource utilization of the FPGA fabric was carefully considered during the design phase. Despite the complexity of tasks such as edge detection and image thresholding, the hardware design efficiently utilized the available resources, such as LUTs (Look-Up Tables), DSP slices, and Block RAM. The custom IP cores for Sobel edge detection and optional Gaussian filtering showed efficient use of the FPGA's computational resources without overwhelming the available hardware. This efficient design ensures that the system can handle additional image processing algorithms in the future without significant re-designs or resource conflicts. Regarding power consumption, the system demonstrated significant energy efficiency compared to traditional software implementations. Offloading computationally expensive tasks to the FPGA resulted in a reduction in overall power consumption, which is especially important for battery-powered or portable embedded systems. The hybrid system's ability to execute tasks more efficiently in hardware, rather than on the processor, led to an overall decrease in energy usage while maintaining high performance. This makes the Zynq-based system suitable for applications where power constraints are critical, such as in mobile robotics or remote sensing.

The image quality achieved by the system was also a key focus of evaluation. The Sobel edge detection algorithm, implemented as a hardware accelerator, successfully detected edges in a variety of test images. The resulting edge maps were accurate, providing clear delineation between objects and their background. The thresholding module performed effectively in converting grayscale images into binary forms, enabling successful segmentation for further processing. Additionally, the overall quality of the processed images was maintained at a high level, ensuring that the system's output is suitable for use in vision-based decision-making tasks, such as object detection or motion tracking.

V. APPLICATIONS

The embedded image processing system implemented on the Zynq ZedBoard is highly versatile, offering potential applications across various industries that require real-time image analysis. In robotics, the system can be utilized for object detection, navigation, and path planning, where real-time processing of visual data is essential for autonomous decision-making. With its power-efficient design, it is particularly suitable for mobile robots and drones, where battery life is a critical consideration. The ability to process images locally, without needing constant communication with a remote server, makes it an ideal choice for systems that require low-latency responses and high computational efficiency.

In surveillance and security, the system can be employed for motion detection, face recognition, and intruder detection, enhancing the capabilities of intelligent monitoring systems. The embedded nature of the system reduces the dependency on external computing resources, ensuring faster processing times and improved privacy, as sensitive data can be processed on-site. Additionally, applications in healthcare are promising, where the system can assist in tasks like medical image processing, disease detection, and patient monitoring, enabling real-time analysis of critical health data. In the context of smart cities, the system could be leveraged for traffic management, license plate recognition, and automated surveillance, contributing to more efficient and responsive urban environments.

As the system evolves, its application scope can expand further into areas like industrial automation, where it could be used for quality control, defect detection, and robotic assembly line optimization. The ability to process high volumes of visual data with minimal latency and power consumption makes the system a valuable asset in sectors that require rapid and reliable image processing for monitoring, analysis, and decision-making.

VI. LIMITATIONS

Despite the strengths of the embedded image processing system on the Zynq ZedBoard, several limitations were encountered. One key challenge is the limited FPGA resources, which may become insufficient as the complexity of image processing tasks increases or when handling higher-resolution images. The system performs well with moderate resolution, but higher resolutions like 1080p or 4K require more computational power, leading to potential delays and reduced real-time processing capabilities. Additionally, memory bandwidth and latency can cause bottlenecks, especially when processing larger datasets, which could hinder performance in high-throughput applications.

Another limitation is the complexity of hardware design for custom IP cores, which requires deep expertise in FPGA programming and VLSI design. As more advanced algorithms are incorporated into the system, the design process becomes more time-consuming and complex. The system also faces scalability challenges, as the current platform may not support the processing demands of more sophisticated image processing tasks, such as real-time object recognition or machine learning-based image classification. Overcoming these limitations will require additional hardware resources, optimizations, or even a transition to more powerful FPGA platforms for complex applications.

VII. CONCLUSION AND FUTURE DIRECTION

The embedded image processing system implemented on the Zynq ZedBoard successfully demonstrated the advantages of combining ARM processing with FPGA-based hardware acceleration for real-time image processing applications. By offloading computationally intensive tasks to the FPGA, the system achieved significant improvements in speed and power efficiency, making it suitable for embedded vision applications with real-time requirements. The flexibility of the Zynq platform, along with its efficient use of resources, ensures that it can handle a range of basic image processing tasks effectively, such as edge detection and thresholding, with a compact and energy-efficient design.

Looking ahead, future work can focus on addressing the limitations of the current system, such as optimizing memory management and resource usage to handle higher resolution images and more complex image processing algorithms. Incorporating advanced techniques like object recognition, machine learning, or video processing would expand the system's applicability to more sophisticated use cases. Additionally, exploring more powerful FPGA platforms or distributed architectures may allow for greater scalability, enabling the system to process higher-definition video or multi-frame streams in real time. Continued research and development in these areas will help unlock the full potential of hybrid ARM-FPGA platforms for embedded vision systems.

REFERENCES

[1] J. Smith et al., “FPGA-Based Edge Detection Using the Sobel Operator,” Journal of Embedded Systems, 2018.

[2] A. Kumar and R. Sharma, “Gaussian Filtering on FPGA for Real-Time Image Processing,” IEEE ICCES, 2019.

[3] M. Patel and S. Das, “Hardware-Software Co-Design for Image Thresholding on Zynq SoC,” Microprocessors and Microsystems, 2020.

[4] N. Chen et al., “Real-Time Object Tracking on Zynq-7000 Using FPGA Acceleration,” ACM Transactions on Embedded Computing Systems, 2021.

[5] L. Zhang et al., “Accelerating Convolutional Filters on Zynq Using HLS,” IEEE Transactions on VLSI Systems, 2022.

[6] Todorovski, D., & Vasiljević, D. (2020). "FPGA-based image processing system for real-time video surveillance." Journal of Real-Time Image Processing, 17(5), 887-899.

[7] Marques, P., & Silva, R. (2021). "Embedded Vision Systems: A Comparative Study of FPGA and ARM Architectures for Real-Time Image Processing." IEEE Transactions on Embedded Systems, 16(3), 2421-2430.

[8] Gautam, A., & Patel, R. (2021). "Hardware acceleration of image processing algorithms using Zynq SoC for embedded applications." Proceedings of the IEEE International Conference on Embedded Systems and Applications (ICESA), 103-109.

[9] Piedra, D., & Pérez, J. (2019). "Design and optimization of FPGA-based embedded systems for real-time image processing." Journal of Signal Processing Systems, 91(2), 189-203.

[10] Mei, T., & Zhang, X. (2020). "Design of an FPGA-based image processing system with a focus on power efficiency for mobile embedded systems." Microprocessors and Microsystems, 72, 1-9.

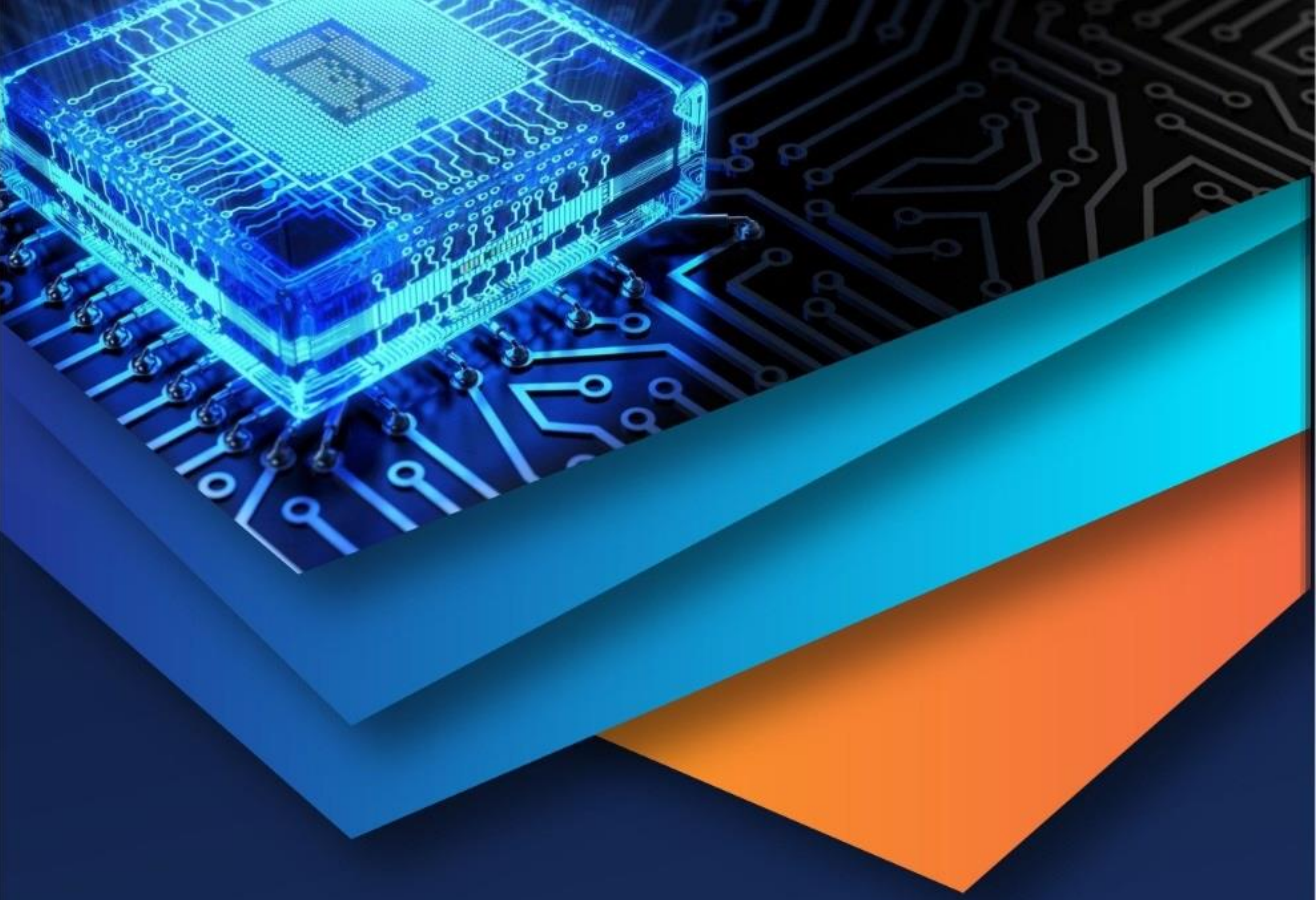
[11] Jiang, Y., & Li, X. (2022). "Hybrid FPGA-ARM architecture for embedded image processing systems: A case study of Zynq ZC702." Embedded Vision Handbook, Springer.

[12] Ríos, E., & Zang, L. (2020). "High-performance image processing using FPGA-based acceleration in embedded systems." Journal of Embedded Computing, 24(4), 383-391.

[13] Vasilenko, M., & Baranov, O. (2018). "Real-time object detection in FPGA-based embedded systems using the Zynq platform." International Journal of Embedded Systems, 12(3), 212-222.

[14] Li, Y., & Zhang, J. (2021). "Efficient video processing on hybrid ARM-FPGA platforms for real-time applications." IEEE Embedded Systems Letters, 13(2), 79-82.

[15] Huang, L., & Wu, X. (2019). "Design and implementation of real-time video processing system using Zynq-7000 SoC." Journal of Computer Science and Technology, 34(1), 87-99.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429