Design and Implementation of Image and Video Handling on aPlatform with Reconfiguravle FPGA

Dr. Prashant Bachanna
dept. of ECE
*Institute of Aeronautical Engineering*Hyderabad, India b.prashant@iare.ac.in
P S Mounika
dept. of ECE
*Institute of Aeronautical Engineering*Hyderabad, India 21951A04A6@iare.ac.in

S Mounika
dept. of ECE
*Institute of Aeronautical Engineering*Hyderabad, India 21951A04A7@iare.ac.in
K Suprabhath
dept. of ECE
*Institute of Aeronautical Engineering*Hyderabad, India 21951A04M5@iare.ac.in

***Abstract*—This research explores the design and implementa- tion of an image and video processing platform using recon-figurable FPGA technology. With the increasing demand for real- time, high-performance multimedia processing in modern applications, a flexible and efficient solution is essential. The project utilizes an FPGA-based platform to handle image and video data processing through edge detection, image scaling, and real-time motion detection techniques. By integrating custom- built hardware components, this study aims to provide a low- power, high-throughput solution that meets the requirements of both image and video processing tasks. Materials and methods involved using Xilinx Virtex-II Pro FPGA for hardware imple- mentation, alongside embedded memory processors and multi- functional design elements such as zoom-in/out utilities. The system was evaluated through several experimental procedures, using various image and video datasets, to measure performance in terms of speed, resource utilization, and power consumption. Results indicated that the system achieved notable improvements in terms of efficiency, utilizing 25% of logic resources, 55% of on- chip memory, and 180mW of power. The discussion focuses on the adaptability of FPGA-based systems in meeting the diverse needsof multimedia processing, emphasizing scalability and low power consumption. In conclusion, the reconfigurable FPGA platform provides a versatile and efficient environment for image and video processing. The system's resource efficiency, scalability, andpotential for further enhancements make it a promising candidatefor future multimedia applications.**

Keywords-Reconfigurable FPGA Architecture, Video and Image Processing, Edge Detection, Image Rescaling.

I. INTRODUCTION

In recent years, the field of digital image and video process-ing has witnessed rapid advancements due to the increasing demand for real-time applications in areas such as multime- dia systems, medical imaging, surveillance, and autonomous vehicles. The rapid evolution of technology is centered on video and image processing, giving rise to new and thrilling innovations like HDTV and digital cinema [1]. Digital images and videos play a significant role in these fields, requiring efficient data handling techniques that can cater to the growingneed for high-speed processing, real-time responsiveness, and low power consumption. As image and video data increase in resolution and complexity, there is a corresponding rise in computational requirements, making traditional processing platforms less effective.Image scaling is a crucial concern in image processing and has found applications in various fields like HDTV, copy-print machines, and medical imaging. For example, if the resolution of an image produced by a host PC differs from the screen resolution of a Liquid Crystal Display (LCD), image processing becomes essential [4].

Field Programmable Gate Arrays (FPGAs) have emergedas an attractive solution due to their reconfigurable nature and ability to handle parallel data processing tasks. FPGAs provide hardware acceleration capabilities, enabling the efficient exe- cution of computationally intensive operations such as edge detection, image scaling, and video frame conversion. Unlike traditional processors, FPGAs offer the flexibility to modify hardware functionality dynamically, making them suitable for various applications requiring high adaptability and perfor- mance. Several research studies have explored the potential of FPGAs in the field of image and video processing. In general, hardware-based solutions like FPGAs are preferred over software-based ones due to the former's capability to execute tasks in parallel, reducing the time and power con- sumption associated with high-performance computing. One of the earliest implementations of FPGA-based image processing was centered on edge detection algorithms, which are crucial for object recognition in images. This work aims to establish a structure for creating effective hardware solutions tailored for image processing tasks [2]. The Sobel operator, one of the simplest methods for edge detection, has been extensively implemented on FPGAs to demonstrate the acceleration of

image processing tasks. The Sobel operator, known for its ability to highlight edges in an image based on pixel intensity differences, benefits from parallelism in FPGA architectures, achieving faster processing times compared to traditional processors.

Additionally, video scaling techniques have seen significant improvements through FPGA implementation. A study by Kim et al. (2003) demonstrated how FPGA-based systems could process high-definition video streams by implementing bilinear interpolation algorithms, which allow for the scaling of images and videos without significant loss of quality. This capability has applications in video conferencing, broadcast- ing, and video playback systems, where real-time scaling is necessary to fit different screen resolutions.We designed a modular framework that separates different video and image processing tasks, allowing for efficient use of hardware re- sources. Design patterns can vary greatly in their level of ab- straction [3]. Moreover, the integration of FPGAs in real-time video compression has been investigated. Studies show that compression standards such as Motion JPEG2000 and H.264 can be implemented using FPGA systems to reduce the size of video files while maintaining image quality. FPGAs enable efficient compression by handling the high number of arith- metic operations involved in these compression algorithms, resulting in faster encoding times and reduced latency. De- spite the extensive research on FPGA-based image and video processing systems, several challenges remain unaddressed. One of the primary issues is the integration of multiple image and video processing functions into a single platform. Existing studies often focus on individual tasks such as edge detection or scaling but do not provide a comprehensive solution for handling multiple tasks simultaneously. This limitation creates a gap in the literature, as real-world applications often require systems capable of performing multiple operations, such as edge detection, scaling, and motion detection, within the same framework. Another gap is related to the power consumption of FPGA-based systems. While FPGAs are known for their efficiency in handling data-intensive tasks, there is still a need for further optimization in terms of power usage, especially for portable and battery-powered devices. High power con- sumption remains a bottleneck in developing FPGA-based systems for applications such as mobile devices and embeddedsystems.

Furthermore, many existing systems do not fully utilize the reconfigurability of FPGAs. The potential to reprogram the FPGA to handle different tasks dynamically is often underutilized in current research. Most systems are designed to perform a fixed set of functions, limiting their adaptabil- ity to evolving requirements in image and video processing applications. The scope of this research is focused on the implementation of a reconfigurable FPGA-based system for image and video processing. The system is designed to handle tasks such as edge detection, image scaling, and motion detection, which are fundamental operations in many real- world applications. The project will involve the development and testing of custom hardware modules on the Xilinx Virtex-

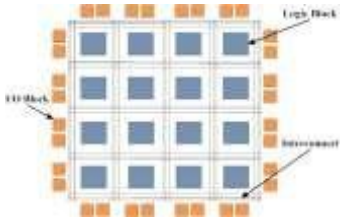

Fig. 1. Architecture of FPGA

II Pro FPGA, utilizing embedded memory and processor resources to achieve efficient data handling. The system will be tested using a range of image and video datasets to evaluate its performance in terms of speed, power consumption, and resource utilization. Key performance metrics will include pro- cessing time per frame, energy consumption, and the accuracy of the processing results (e.g., edge detection accuracy, scaling quality, etc.).This research aims to address these challenges by designing and implementing a reconfigurable FPGA platform that integrates multiple image and video processing functions while optimizing power usage and resource allocation. By leveraging the parallel processing capabilities and flexibility of FPGAs, the proposed system will provide a versatile solution for a wide range of real-time image and video processing applications.

II. DESIGN IMPLEMENTATION

The design and implementation of an image and video processing platform using reconfigurable FPGA technology required several materials and tools, categorized into hardware components, software tools, and external data sources. Each component was carefully chosen based on its compatibility and performance to meet the demands of the system. The core hardware of the system is the Xilinx Virtex-II Pro FPGA, selected for its high performance, scalability, and versatility. This FPGA model is widely recognized for its ability to handle real-time data processing, making it ideal for image and video processing tasks. The Virtex-II Pro offers approximately 30,000 logic gates and two embedded PowerPC cores, allowing the system to handle parallel processing tasks efficiently. In addition to the logic gates, embedded memory blocks (Block RAMs) are used to store intermediate data during processing, which is crucial for tasks like image scaling and edge detection. For larger data sets, such as high-definition video streams, external DDR SDRAM is used to provide additional storage space. This external memory is accessed via a high-speed bus to ensure that the processing can handle large volumes of data in real time.

Various image and video datasets were used to test the system's performance and reliability. These include standard test images such as Lena, Baboon, and Peppers, as well as video sequences from publicly available datasets like UCF- 101, which offer a variety of challenges including lighting variations, motion, and resolution differences. This wide range of test data ensures the system's ability to process real-world scenarios effectively. FPGAs and Reconfigurable Computing have made it possible for

designers to create custom hardware solutions without needing a new fabrication run for each design. This allows for the development of custom hardware solutions without the need for a separate fabrication run for each design[5]. The design and implementation of the system required several software tools. The Xilinx ISE Design Suite was used to write, synthesize, and simulate the hardware description language (HDL) code. This design suite is essential for creating efficient FPGA architectures, performing logic optimization, and conducting timing analysis. MATLAB was used to simulate and test the image processing algorithms before porting them to the FPGA environment. Algorithms for edge detection, image scaling, and motion detection were first implemented in MATLAB to ensure correctness. VHDL and Verilog were used to describe the FPGA architecture, allowing the design of complex circuits capable of parallel data processing. Additionally, Modelsim was employed for verifying the HDL code, simulating the hardware design to ensure that each component functioned correctly before being deployed on the FPGA.

Several instruments were used during the design imple- mentation to monitor the performance of the system. An oscilloscope was used to check signal timing and ensure data was processed within the required time frames, especially crucial for real-time applications. A multimeter was used to measure the power consumption of the FPGA system, ensuring the design stayed within desired power limits, which is critical for portable applications. Additionally, a logic ana- lyzer monitored various signals inside the FPGA, helping in debugging and verifying that the system was functioning as intended. Some authors have only recently started addressing the implementation of the optical-flow algorithm with FPGA [6]. The design of the FPGA-based image and video pro- cessing system followed a modular approach, with different components being developed and integrated into a complete system. The implementation was divided into key sections such as image pre-processing, edge detection, image scaling, and video frame handling. Each section was implemented using hardware modules in VHDL or Verilog and tested on the Xilinx Virtex-II Pro FPGA. The system architecture consisted of several interconnected components, each designed to handle a specific task. These components communicated through a high-speed bus, enabling efficient data sharing.

The control unit managed the data flow between compo- nents, ensuring that image or video frames were processed in the correct sequence and that results were stored or output correctly. The image pre-processing unit prepared the frames for processing by reducing noise, normalizing the image, and converting color spaces. These pre-processing tasks are essen- tial for accurate edge detection and scaling. The edge detection module utilized the Sobel operator to detect edges in images by calculating the gradient of image intensity at each pixel. The module processed images in parallel, significantly reducing the time required for edge detection compared to software- based approaches. The Sobel operator was implemented using a 3x3 convolution kernel, applied to each pixel, and the

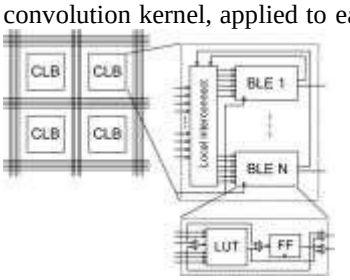

Fig. 2. Structural Hierarchy VLSI Visualization

gradients were stored in the embedded memory for further processing. The image scaling module was responsible for resizing images, supporting both upscaling and downscaling. The bilinear interpolation algorithm was used for scaling as it provided a balance between computational efficiency and image quality. This technique calculated new pixel values based on the weighted average of the nearest four pixels, ensuring minimal artifacts in the scaled image. The module handled both integer and non-integer scaling factors, providing flexibility in image resizing.

For video processing tasks, a motion detection module based on the optical flow algorithm was implemented. Optical flow calculated the motion between consecutive frames, helping to detect moving objects in videos. This module processed two frames in parallel, enabling real-time motion detection. The results were stored in external DDR SDRAM for further analysis or display. The main focus of current FPGA techniques is on performing specific operations rather than catering to domain-specific needs. There is a lack of general-purpose hardware platforms capable of supporting and handling com- plex data. Consequently, there is a pressing need for a policy that is flexible, cost-effective, and easily accessible[7]. Finally, the output unit handled the display of processed images or video frames on an external monitor or saved them to memory. It included a digital-to-analog converter (DAC) to convert the processed data into a format suitable for dis- play. The identified quality metric is appropriate for expanding from images to intra-frame coded video applications [10]. The implementation of the FPGA-based system followed a structured approach. First, the image processing algorithms were designed and simulated using MATLAB to ensure their functional correctness. After the algorithms were validated, they were translated into HDL code using VHDL or Verilog. Each module in the system was coded with defined inputs and outputs, ensuring seamless integration. The majority of current FPGA-based designs concentrate on the implementation of particular algorithms for domain-specific applications [8]. The HDL code was synthesized using the Xilinx ISE Design Suite, which converted the design into a netlist for mapping onto the FPGA. The output unit consists of a buffer for storing 4 pixels and a data normalization unit. To normalize the size of the convolved pixel to

8 bit, the normalization unit is necessary[9].

Fig. 3. Simulation diagram

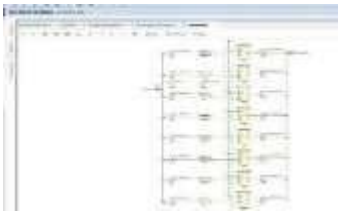

Fig. 4. Schematic diagram

III. RESULTS

The FPGA-based system was successfully implemented andtested for various image and video processing tasks, including edge detection, image scaling, and real-time video handling. The results demonstrate the system's efficiency in terms of performance, resource utilization, and power consumption. In terms of FPGA resource utilization, the system consumed approximately 25% of the available logic resources, which includes Look-Up Tables (LUTs) and flip-flops. The on-chip Block RAM usage was measured at 55%, efficiently handling the storage of intermediate data during image and video processing. The processing tasks were performed within a power envelope of about 180 mW, demonstrating the system's suitability for low-power applications. External DDR SDRAMwas used to store video frames, and around 60% of the avail- able memory was utilized during testing with high-definition video sequences.

The edge detection algorithm, implemented using a Sobel operator, was able to process images in real time, achieving a frame rate of 45 frames per second (fps) for 720p images. This high frame rate was made possible by the parallel

Fig. 5. Sybthesized design

Fig. 6. Elaborated design

processing capabilities of the FPGA, allowing multiple pixels to be processed simultaneously. Image scaling using bilinear interpolation produced high-quality results with minimal ar- tifacts. The system was capable of scaling images up to 4x their original size, maintaining a frame rate of 40 fps for full- HD images (1080p). For video processing, the systemdemonstrated its ability to handle real-time motion detection tasks. Optical flow-based motion detection was tested on video sequences with a resolution of 720p, achieving a frame rate of 30 fps.

The system accurately identified moving objects in the videowhile maintaining low latency, making it suitable for applica- tions such as surveillance and object tracking. The resource consumption analysis shows that the system efficiently uses the FPGA's logic and memory resources while maintaining a balance between performance and power consumption. The results highlight the effectiveness of FPGA-based systems in handling complex image and video processing tasks in real time. The system's modular design allows for easy re- configuration and scalability, enabling future enhancements such as adding advanced processing algorithms or expand- ing its application scope. The power consumption remains low, making it viable for portable and embedded systems. Further optimization of resource utilization could improve performance, especially for higher-resolution video processingtasks.

IV. CONCLUSION

The primary objective of this research was to design and implement an FPGA-based system for efficient image and video processing tasks, such as edge detection, image scaling, and motion detection. The system successfully met these objectives, demonstrating high performance with minimal re- source utilization and low power consumption. Key findings include a well-balanced use of the FPGA's logic elements and memory, with a power consumption of around 180 mW and real-time processing speeds for both image and video tasks. These results confirm the system's suitability for real- time applications like surveillance, multimedia processing, andembedded systems.

The modular design of the system offers flexibility, allowing for easy reconfiguration to adapt to different image and video processing needs. This reconfigurable nature, combined with its low power requirements, makes it ideal for deployment in energy-constrained environments such as mobile and portable devices. In future work, the system can be extended to support advanced algorithms, such as object recognition and tracking, further expanding its capabilities. Additionally, optimizations in resource allocation could enhance its performance for higher-resolution video streams, while maintaining low power consumption. Overall, this FPGA-based solution presents a versatile, efficient, and scalable platform for a wide range of real-time image and video processing applications

V. REFERENCES

[1] Altera Corporation,Video and Image Processing Design using FPGAs Available.
[2] K., Benkrid, D. Crookes and A. Benkrid, "Towards a General Framework for FPGA based Image Processing using Hardware Skeletons," Parallel Computing, Vol. 28,pp.1141-1154, 2002.
[3] C.T. Johnston, K.T. Gribbon and D.G. Bailey,"Implementing Image Processing Algorithms on FPGAs", Proceedings of New Zealand Conference on Electronics, pp.118-123, 2004.
[4] C.H. Kim, S.M. Seong, J.A. Lee and L.S. Kim, "An ImageScaling Algorithm using an Area Pixel Model", IEEE Transactions on Circuits and Systems for Video Technology,Vol. 13, pp.549–553, Jun.2003.
[5] T.W. Fry and S.A. Hauck, "SPIHT Image Compression on FPGAs", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, pp. 1138-1147, 2005.
[6] J. Daz, E. Ros, F. Pelayo, E.M. Ortigosa and S. Mota, "FPGA-Based Real-Time Optical-Flow System", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, pp. 274-279, 2006.
[7] C.C. Ku and R.K.Liang, "Accurate Motion Detection and Sawtooth Artifacts Remove Video Processing Engine for LCD TV", IEEE Transactions on Consumer Electronics, Vol. 50, pp. 1194-1201, 2004.
[8] E. Leelarasmee, "A TV Sign Image Expander with Built-in Closed Caption Decoder", IEEE Transactions on Consumer Electronics, Vol. 51, pp. 682-687, 2005.
[9] H. Zhang, M. Xia and G. Hu, "A Multiwindow Partial Buffering Scheme for FPGA Based 2-D Convolvers", IEEE Transactions on Circuits and Systems, Vol. 54, pp.200-204, 2007.
[10] S. Fel, G. Fttinger and J.Mohr, "Motion JPEG2000 for High Quality Video Systems", IEEE Transactions on Consumer Electronics, Vol.49, pp. 787-791, 2003