Image Processing for On Screen Display

Gagan D[1], Dr. Jeeru Dinesh Reddy[2]

[1]*Department of Electronics and Communication Engineering, BMS College of Engineering, Bangalore, India*
[2]*Department of Electronics and Communication Engineering, BMS College of Engineering, Bangalore, India*

Abstract - *Understanding and optimizing JPEG decoding is imperative for several reasons. JPEG ubiquity in digital media from online content delivery to medical imaging underscores its significance. Balancing quality and speed are essential, impacting user experience in applications like online streaming and professional photography. The study of JPEG decoding also contributes to storage efficiency, minimizing requirements and optimizing data transfer times. This article presents a "JPEG Decoder" which can deal with different quality of images that can be displayed on the screen with less loss of information and providing best storage capabilities and thereby increasing the overall performance of the Decoder. Overall, the articles achieved significant improvements in resource utilization. The optimized utilization of Slice LUTs and Slice Registers reflects a more judicious use of logic elements Slice LUTs (26.65%) and Slice Registers (8.06%), contributing to a streamlined and powerful logic implementation compared with [2] higher Slice LUT utilization (33.26%) and slightly increased Slice Register usage (8.44%). The efficient management of DSP resources underscores the design's computational capabilities, particularly in digital signal processing applications.*

Key Words: FPGA, VGA Controller, Image Processing, Xilinx Vivado, Verilog.

1.INTRODUCTION

The evolution of JPEG decoding is intricately woven into the fabric of digital image processing, showcasing a remarkable journey through technological advancements and innovative methodologies. As we trace the historical trajectory, we witness the steady refinement of JPEG decoding algorithms, driven by the increasing demand for efficient image reconstruction. The inception of the JPEG standard can be traced back to the late 1980s when the Joint Photographic Experts Group (JPEG) was formed, comprising experts from various fields aiming to establish a standardized image compression method. The result was the creation of the JPEG standard in 1992, which quickly gained widespread adoption due to its ability to strike a balance between compression ratios and acceptable image quality.

The initial JPEG decoding algorithms were primarily implemented on general-purpose processors, leveraging the computing power available at the time. However, as digital imagery proliferated across diverse applications, the demand for faster and more efficient decoding processes intensified. The integration of Field-Programmable Gate Arrays (FPGAs) into JPEG decoding marked a significant turning point in the pursuit of enhanced performance. FPGAs, with their reconfigurable nature and parallel processing capabilities, offered a compelling alternative to traditional processors. This adaptability allowed developers to tailor hardware architectures specifically for JPEG decoding, unlocking the potential for accelerated and optimized processing. The synergy between JPEG decoding and FPGA technology gained prominence in various applications, from real-time video processing to embedded systems. The adaptability of FPGAs allowed for tailored solutions that addressed the unique challenges posed by JPEG decoding, such as computational intensity and real-time processing requirements.

In the contemporary landscape, the integration of FPGA technology into JPEG decoding processes continues to be an area of active research and development. This historical journey sets the stage for our exploration into the intricacies of JPEG decoding, with a specific focus on leveraging FPGA acceleration to enhance efficiency and meet the demands of modern applications. Understanding and optimizing JPEG decoding is imperative for several reasons. JPEG's ubiquity in digital media from online content delivery to medical imaging underscores its significance. Balancing quality and speed are essential, impacting user experience in applications like online streaming and professional photography. As emerging technologies (virtual reality, augmented reality) demand efficient image processing, optimized JPEG decoding becomes pivotal. User satisfaction in contexts such as web browsing and mobile applications hinges on decoding speed. Integrating hardware accelerators like FPGAs offers opportunities for significant performance enhancements.

As digital imagery continues to proliferate across diverse domains, from multimedia streaming to medical imaging, the ability to decode compressed images swiftly while maintaining optimal quality has emerged as a critical challenge. At the heart of this challenge lies the JPEG compression standard, a cornerstone in the realm of image compression. This thesis centers its focus on the intricate process of JPEG decoding, unraveling the layers that constitute the post-compression reconstruction of images. A key motivation is to delve into the intricacies of JPEG decoding, with a dual objective: to enhance the efficiency of the decoding process and to explore the integration of Field-Programmable Gate Arrays (FPGA) as a catalyst for accelerated decoding. JPEG

decoding, while foundational, can be computationally intensive, especially when dealing with high-resolution images or real-time applications. To address this, we turn our attention to FPGA technology—a versatile hardware platform known for its parallel processing capabilities. By harnessing the inherent parallelism of FPGA architectures, we aim to design and implement a high-performance JPEG decoding accelerator. The integration of FPGA into the decoding pipeline promises not only to expedite the process but also to provide a scalable solution. Adaptable to varying computational requirements. Moreover, as the volume and resolution of digital images surge, the role of storage in facilitating seamless decoding becomes increasingly crucial.

This thesis recognizes the symbiotic relationship between decoding speed and storage access times, emphasizing the need for a holistic approach. Consequently, a significant portion of this research is dedicated to exploring storage optimization techniques, including efficient data structures, and caching mechanisms, to further augment the overall performance of JPEG decoding. Hence image processing is one of the first step to obtain good quality images with less storage required. In our proposed work we have developed a "Image Processing for On Screen Display" which can deal with different quality of images that can be displayed on the screen with less loos of information and providing best storage capabilities and thereby increasing the overall performance of the Decoder.

2. PROPOSED METHODOLOGY

Need for Lossy Decompression technology: In the context of real-time image and video processing using FPGA (Field-Programmable Gate Array), the choice of lossy decompression remains pertinent due to the FPGA's parallel processing capabilities and the specific demands of real-time applications. FPGA-based systems often prioritize computational efficiency, and lossy compression methods align well with these requirements. JPEG, for instance, is a widely used lossy compression standard that can be implemented efficiently on FPGA platforms. Lossy decompression on FPGAs allows for rapid processing of compressed image or video data, making it suitable for applications such as real-time video streaming, surveillance, or medical imaging, where quick analysis and response are critical.

The parallel processing nature of FPGAs allows for the simultaneous decompression of multiple frames or regions, contributing to reduced latency and improved throughput. While FPGA-based systems are known for their hardware acceleration capabilities, they are still subject to resource constraints, including limited on-chip memory and bandwidth. Lossy compression methods strike a balance between achieving compression efficiency and minimizing resource utilization, making them well-suited for FPGA implementations in real time image and video processing scenarios.

Fig -1: Proposed Block Diagram

2.1 JPEG Decoder:

We observed there is a need of processing unit where it can process the compressed image efficiently with improved performance, clarity of the image and occupies less storage Fig -2 shows the proposed block diagram for JPEG Decoder.

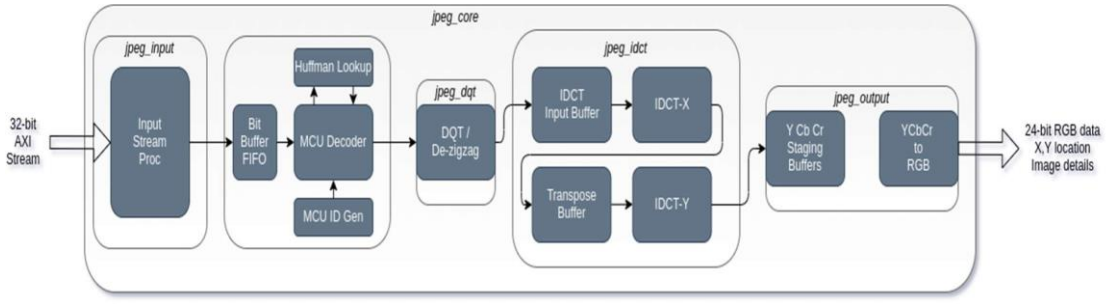

Fig -2: Block Diagram of JPEG Decoder

Description of Processing in each step:

JPEG Input:
This is where the process begins. It represents an input stream of JPEG images.

Input Stream Proc:
This step handles the initial processing of the input stream. It prepares the data for further stages.

MCU Decoder:
MCU decode the compressed image data, extract MCUs, perform Huffman decoding, and generate identification information for proper arrangement during the image reconstruction process in a JPEG decoder.

JPEG_DQT Block:
The JPEG_DQT block plays a pivotal role in the JPEG decoding process by de-quantizing the coefficients and reversing the zigzag ordering applied during compression. This prepares the data for the subsequent stages, such as IDCT, where the original image data is reconstructed.

JPEG-IDCT block:
The JPEG_IDCT block is responsible for reversing the effect of the Discrete Cosine Transform applied during JPEG compression.

JPEG Output:
The JPEG_Output block takes the YCbCr image data obtained from the JPEG_IDCT block, stages it in separate buffers, and then performs the YCbCr to RGB color space conversion.

2.2 Implementation of VGA Controller:

The VGA controller design follows a modular approach, with two main modules: `vga_test` and `vga_controller`. The `vga_test` module serves as the top-level module and instantiates the `vga_controller` module, while also providing the RGB color values for the VGA output. The `vga_controller` module implements the VGA timing and generates the necessary synchronization signals. Block diagram is shown in Fig -3.

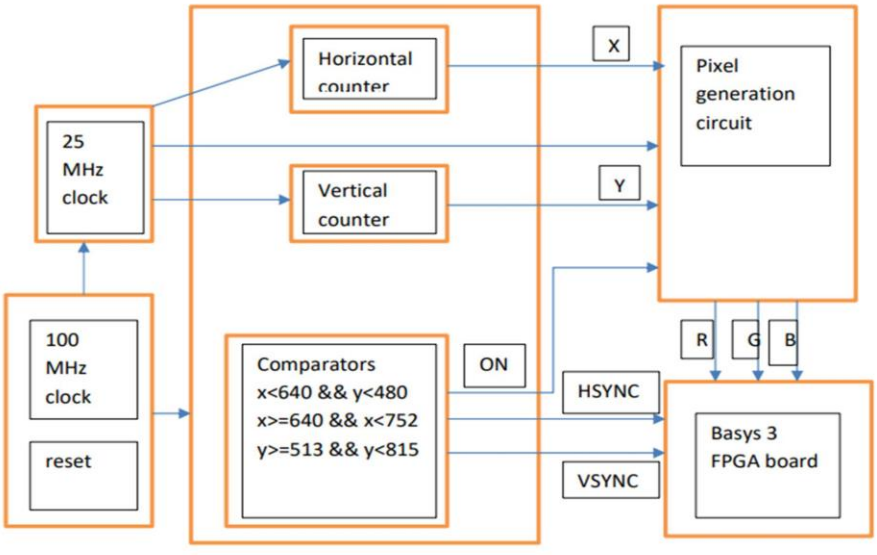

Fig -3: Block Diagram of VGA Controller

3. Results and Discussions

3.1 RTL Schematic of the Proposed Block Diagram

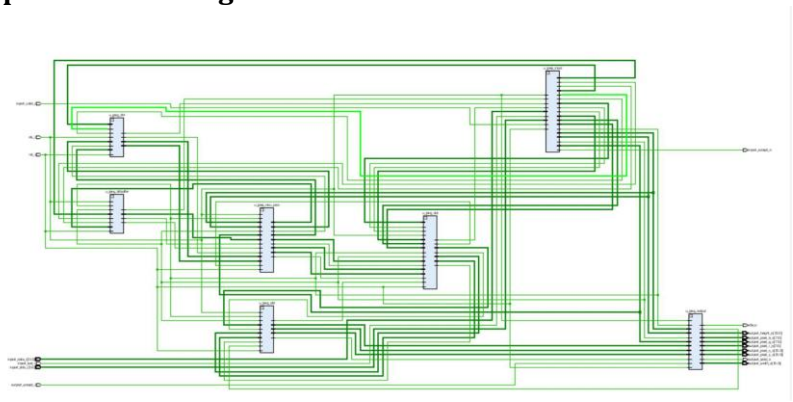

Fig -4: RTL schematic of the proposed block diagram

3.2 Synthesis Result of Decoder Block

In this section Synthesis output of Proposed Decoder Block is shown in Fig 5. In the next section details regarding utilization summary is explained

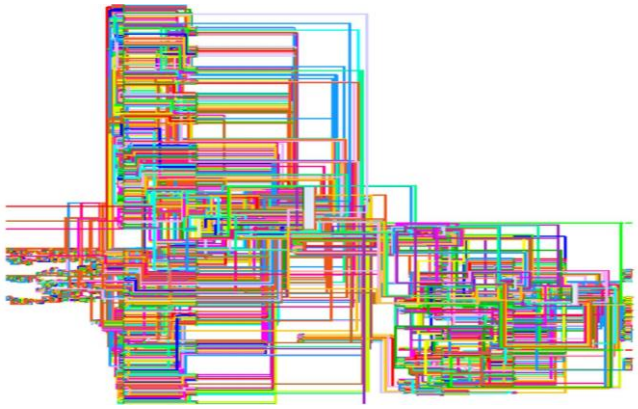

Fig -5: Synthesis Output of proposed Decoder Block

3.3 Utilization summary of Decoder Block

The Utilization Summary offers a concise overview of how FPGA resources are allocated within the Decoder design. This section provides key insights into the efficiency and distribution of resources across various components. This is represented in Table -1.

Table -1: Resource Utilization Table of Decoder Block

Resource	Utilization	Available	Utilization%
Slice LUTs	5544	20800	26.65
Slice Registers	3351	41600	8.06
Memory	6	50	12.00
DSP	31	90	34.44
IO	132	106	124.53
Clocking	1	32	3.12

3.4 Utilization Summary of VGA Controller Block

The Utilization Summary offers a concise overview of how FPGA resources are allocated within the VGA Controller design. This section provides key insights into the efficiency and distribution of resources across various components. This is represented in Table -2.

Table -2: Resource Utilization Table of VGA Controller

Resource	Utilization	Available	Utilization%
Slice LUTs	41	20800	0.20
Slice Registers	56	41600	0.13
IO	28	106	26.42
Clocking	1	32	3.12

4. CONCLUSIONS

In conclusion, the FPGA design has achieved significant improvements in resource utilization, showcasing enhanced efficiency across various parameters as shown in Table -1. The optimized utilization of Slice LUTs and Slice Registers reflects a more judicious use of logic elements Slice LUTs (26.65%) and Slice Registers (8.06%), contributing to a streamlined and powerful logic implementation compared with [2] higher Slice LUT utilization (33.26%) and slightly increased Slice Register usage (8.44%). The efficient management of DSP resources underscores the design's computational capabilities, particularly in digital signal processing applications.

The IO subsystem, although exhibiting potential for optimization, requires continued refinement for seamless interfacing and reduced constraints. The clocking architecture, with its current stability, may benefit from ongoing research into adaptive strategies for enhanced, precision and synchronization.

Overall, in comparison to results presented in [2], the current design developed is more optimized and efficient demonstrating effective management of on-chip memory resources.

REFERENCES

[1] Junming Shan, Duyao Wang, Eryan Yang, "High Performance JPEG Decoder Based on FPGA," *IEEE,* 2011.
[2] David J. Lucking, Eric J. Balster Kerry, L. Hill Frank, A. Scarpino, "FPGA Implementation of the JPEG 2000 binary architecture (MQ) decoder," *Springer* 2011.
[3] Rushikesh Tade, Saniya Ansari, "Acceleration of JPEG Decoding Process using CUDA," International Journal of Computer Applications, Volume 120 No. 9, June 2015.
[4] Tiago Rodrigues, Mario Vestias, "Using Dynamic Reconfiguration to Reduce the Area of a JPEG Decoder on FPGA," *IEEE*, 2015.
[5] G. Kyrtsakas, R. Muscedere "An FPGA Implementation of a Custom JPEG Image Decoder SoC Module," *IEEE*, 2017.
[6] Dr. Kamlesh Kumar Singh, Dr. Deependra Pandey, "Implementation of DCT and IDCT Based Image Compression and Decompression on FPGA," International Conference on Inventive Systems and Control *ICISC,* 2017.
[7] Yousef Baroud, Jose Manue, Marin os Velarde, Zhe Wang, Steffen Kie, Seyyed Mahdi Najmabadi, Jajnabalkya Guhathakurta, Sven Simon, "Architecture for parallel marker-free variable length streams decoding," *Springer*, 2017.
[8] S. D. Jayavathi, A. Shenbagavalli, "FPGA-based Auxiliary Minutest MQ-Coder Architecture of JPEG2000," *Springer,* 2017.
[9] An efficient FPGA-based dynamic partial reconfiguration design flow and environment for image and signal processing IP cores.
[10] https://digilent.com/reference/programmable-logic/basys-3/reference-manual