

## Jpeg Image Compression Using An Fpga

A.Srivani<sup>1</sup>, K.Santhoshi<sup>2</sup>, P.Anitha<sup>3</sup>, V.Sravya<sup>4</sup>, Dr. S K Cheyadevi<sup>5</sup>

<sup>1,2,3,4</sup> B-Tech Final year students, ECE department, JB Institute of Engineering and Technology, Hyderabad, Telangana

<sup>5</sup> Head of Department, ECE Department, JB Institute of Engineering and Technology, Hyderabad, Telangana.

\*\*\*

**Abstract** - In this design, we provide a co-architectural structure of JPEG image compression utilizing the FPGA platform, where a MicroBlaze-based control unit and an optimized JPEG accelerator have been used. MicroBlaze performs the job of system level controls such as fetching the input image file, image conversion (RGB to YCbCr), triggering the hardware accelerator, generation of JPEG header, etc., while interfacing with the hardware module via AXI-Lite and AXI interconnect interface. Image data stored in DDR memory can be fetched through AXI DMA engine, which can perform MM2S and S2MM transfer schemes. Image data from the memory is streamed to the JPEG accelerator using the AXI-Stream interface and vice versa, and all these processes can be done without requiring intensive processor effort.

The JPEG accelerator handles the time-consuming parts of JPEG compression, such as 2D DCT, quantization, zigzag scanning, and Run-Length Encoding. The use of a hardware-accelerated approach in JPEG compression allows for better speed and efficiency compared to a software-only approach. In conclusion, the overall architecture takes advantage of both the advantages of a soft-core processor and those of a hardware-accelerated design in order to effectively compress images using JPEG algorithms on FPGA.

**Key Words:** FPGA, MicroBlaze Processor, JPEG Hardware Accelerator, AXI-Lite Interface, AXI Interconnect, GB to YCbCr Conversion, 2D DCT, Quantization, Zigzag Scanning, Run-Length Encoding (RLE).

### 1. INTRODUCTION

With the emergence of multimedia applications, it has become important to efficiently store and transmit images. High-quality images take up a lot of storage space and require huge bandwidth for transmission. Image compression techniques have therefore become very important. Among all the compression standards, JPEG (Joint Photographic Experts Group) has emerged as the most popular image compression method that offers good compression and satisfactory perceptual image quality. JPEG compression makes use of human perception by changing the image format from RGB to YCbCr, performing 2D Discrete Cosine Transform on image blocks to decompose low-frequency and high-frequency information from images, then applying quantization and entropy encoding through Run-Length Encoding and Huffman coding methods. Though the above procedure is well-established for JPEG, it is computationally very expensive, and software implementation cannot be used for time-critical applications such as video surveillance and medical imaging applications.

The FPGA offers a good solution by virtue of its reconfigurable parallel hardware characteristics, which allow customized pipelining resulting in fast execution time and low latency. This may be achieved by employing a hardware and software co-design architecture, where the tasks performed as part of JPEG compression are split into two parts – one carried out by the processor, which performs the system-level functions like controlling the system, color conversion, and generation of JPEG header info, and the other by a hardware accelerator, which executes the processes of DCT, Quantization, Zigzag Scan, and RLE concurrently.

This paper presents a hardware design of the JPEG compression algorithm, which uses the co-design architecture, where a Microblaze Soft Core Processor performs the system-level tasks while the customized RTL Hardware Accelerator executes the image compression operations. AXI-Lite and AXI Stream protocols provide the communication means among the components at high speed, whereas the employment of AXI DMA facilitates automatic movement of data from the DDR memory to the hardware accelerator.

## 2. LITERATURE SERVEY

### 2.1 Introduction to Image Compression

Compression of images has become an absolute necessity in contemporary digital systems owing to the fast-growing number of multimedia applications. As more and more emphasis has been laid on storing efficiently and transmitting quickly the visual data, a large amount of research work has gone on in this field. In the words of Iain Richardson [1], compression systems are meant for eliminating redundancy from image and video signals while ensuring acceptable quality in the resulting image or video signal. Redundancy can occur in the spatial, temporal, and perceptual domains. Efficient compression techniques make use of such redundancy. As stated by Peter Symes [2], compression techniques can be grouped into two broad categories - namely, lossless and lossy. The former involves compressing the data without any loss, whereas the latter makes use of some data loss in order to achieve greater compression ratios. Lossy compression is preferred in image and video systems because of its efficiency.

### 2.2 JPEG Compression Standard

JPEG is one of the widespread still-image compression standards, developed by the Joint Photographic Experts Group (JPEG). The JPEG standard using transform and entropy coding techniques has been described in detail by Pennebaker & Mitchell [9]. This standard was aimed at the effective compression of continuous-tone images, e.g., photographs. The official definition of the JPEG standard can be found in ISO/IEC 10918-1 [10] by the International Telecommunication Union and ISO/IEC. There are three modes of operation in this standard, namely, baseline (sequential), progressive, and hierarchical. However, the baseline mode is more popular due to its effectiveness and ease. JPEG compression, based on Watkinson [4], includes color space conversion, image block segmentation, block transformation, quantization, and entropy coding. Besides, every stage contributes to minimizing the data amount and maintaining the image quality.

### 2.3 Transform Coding and Discrete Cosine Transform (DCT)

Transform coding is one of the most important techniques for achieving image compression. As per Weidong Kou [6], transform coding helps achieve energy compaction where the majority of energy of the signal gets compacted in a small number of low-frequency coefficients. One of the most frequently used transforms is the Discrete Cosine Transform, which is utilized in JPEG. According to Netravali & Haskell [7], the Discrete Cosine Transform is an efficient approach for representing image data using cosine basis functions. By applying the transform, we can effectively split the information about the picture into two types - low-frequency and high-frequency components. As noted by Gonzalez & Woods [3], low-frequency components contain the majority of visible information, whereas high-frequency components capture the remaining details and noise. Therefore, by focusing on low-frequency components, the transform coding becomes more efficient and can preserve the most visible information.

### 2.4 Quantization Techniques in Compression

Quantization is a vital part of JPEG compression and represents the main reason for data loss. In the words of Watkinson [4], quantization lowers the accuracy of DCT coefficients by dividing them by certain pre-defined values of quantization and rounding down these results. As a result, less crucial data is eliminated from DCT coefficients, especially the high-frequency ones. Quantization leads to intentional image quality loss, and most likely, a person cannot notice such damage with their own eyes. According to Kou [6], the selection of a quantization matrix largely influences compression efficiency and image quality itself. The greater quantization value, the better compression and the worse quality, while low quantization leads to higher quality images with larger amounts of data. The basis of quantization matrices depends on the peculiarities of a human vision system.

### 2.5 Entropy Coding and Data Reduction Techniques

Following quantization, compression continues via the application of various entropy coding methods. According to Pennebaker & Mitchell [9], entropy coding is described as a lossless compression method, taking advantage of redundancies within the data. Runs of zero-valued coefficients resulting from quantization are compressed using Run-Length Encoding (RLE). This is then followed by Huffman coding where the symbols occurring more often are encoded with shorter bits while less frequent symbols are assigned longer codes. Such coding helps to greatly reduce the size of the data. Having stated the significance of entropy coding in enhancing the compression process, Effelsberg & Steinmetz [5] stress that entropy coding contributes significantly towards efficient compression without loss.

## 2.6 Research Gap

Significant advancements have been made regarding research on image compression and JPEG implementation using FPGA to increase the efficiency of compression and improve the speed of processing. Most of the researchers' efforts have concentrated on implementing the basics, such as Discrete Cosine Transformation, quantization, entropy encoding, and color space transformation to enable effective compression. The above principles have been adopted since there exists an ideal balance between compression ratio and image quality making the JPEG compression technique quite popular. However, despite the above-mentioned developments, a few weaknesses have emerged. One of the significant constraints that have been discovered in the current literature concerns the software implementation adopted by previous systems. Although software implementation gives room for flexibility in implementation, it cannot satisfy real-time constraints since the process of computing is quite complicated.

## 3. PROPOSED SYSTEM

The paper discusses the design of an FPGA-based system for compressing JPEG images using hardware-software co - design. The key principle involves merging the programmable capabilities of a MicroBlaze soft-core processor with parallelizable hardware accelerators. These components are integrated on a single programmable logic chip of the Xilinx Artix-7 series (Digilent Nexys 4 DDR board). From the perspective of software solutions, the role of MicroBlaze is responsible for managing control processes of the entire system. This processor loads the raw image data from the memory into the DDR module. Then, in software mode, the image is converted from the RGB to YCbCr color space because the JPEG compression process is optimized when applied individually to luminance and chrominance data. Finally, the MicroBlaze processor configures the hardware accelerator using parameters of source and destination addresses and transfer length by means of the AXI-Lite bus protocol. Additionally, it creates the mandatory JPEG file header containing SOI, DQT, and DHT markers and inserts it at the beginning of the bitstream file.

Regarding the hardware part of JPEG, the JPEG Accelerator will implement the time-consuming processes of compression using pipeline operations. Firstly, a two-dimensional discrete cosine transform is performed on each pixel block with dimensions 8x8 to convert the signal from the spatial domain to the frequency domain, allowing almost all the energy of the image to be stored in a few low frequencies coefficients. Then quantization, through which each obtained DCT coefficient is divided by a value from the quantization table, rounded off and converted into an integer, resulting in the main data compression process. Following that is zigzag scanning, in which the values of coefficients are rearranged from low frequency to high frequencies so that all non-zero values occupy the leading positions, and a large number of zeroes occur at the tail of the sequence. Finally, run-length encoding compresses these zero sequences into (run\_length, value) tuples.

Communication with the processor and hardware is coordinated through AXI interconnect technology. The AXI-Lite is employed in sending control signals and register read/write operations. The image data is transferred using AXI-Stream interface. The AXI Interconnect connects the masters and slaves through all transactions with address decoding and arbitration. The AXI DMA controller automatically transfers the data where the raw image data is transmitted from DDR memory to the accelerator through the MM2S path while sending the output image data to the DDR memory via the S2MM interface without overloading the MicroBlaze processor. The DDR memory acts as the buffer to store images throughout the process. Overall, there is real-time image compression in which the system achieves high throughput and low latency along with FPGA resources optimization. Additionally, the compression process yields good quality of the compressed images.

## 4. METHODOLOGY / SYSTEM ARCHITECTURE

### 4.1 SYSTEM ARCHITECTURE

Design Architecture of the System Under Consideration Consists of Three Components: the MicroBlaze Soft-Core Processor, the Hardware Encoder/Decoder for JPEG Image Format, and the AXI Interconnect Fabric. Such an arrangement makes it possible to design a co-processor system consisting of hardware and software components, implemented in the Xilinx FPGA device.

#### 4.1.1 MicroBlaze Soft-Core Processor

MicroBlaze processor is a soft-core 32-bit RISC processor provided by Xilinx Company and synthesized directly in the FPGA fabric. The processor serves as the control and management processor for the whole system and completes the following functions: reads the input image data from external DDR memory into the data stream of the system; transforms the RGB input

data to the YCbCr format by software function; configures hardware encoder/decoder through AXI-Lite register writes; issues DMAs.

### 4.1.2 JPEG Hardware Accelerator

JPEG hardware accelerator is user-designed RTL (Register Transfer Level) IP core. It runs on FPGA's programmable logic. The hardware accelerator accepts stream of 8x8 pixels blocks using AXI-Stream bus interface, and passes them through the pipeline, which includes DCT transformation, quantization, zigzag scanning and RLE encoding. For optimal speed, pipeline is optimised so that RLE compression of one block, quantization of second block, and transformation with DCT of third block can be done simultaneously, with no time wasted. Memory-mapped register with AXI-Lite bus is provided for configuration of JPEG hardware accelerator.

### 4.1.3 AXI Interconnect and DMA

In the implementation, the ARM's AMBA (Advanced Microcontroller Bus Architecture) based AXI (Advanced eXtensible Interface) protocol suite is used. AXI Interconnect IP is responsible for connecting transactions from the master (MicroBlaze) and slaves connected to bus network of design. AXI DMA handles the efficient transfer of large amounts of image data between DDR memory and streaming port of the hardware accelerator, without processor intervention. MM2S (Memory-to-Stream) channel accounts for data transfer from DDR memory to JPEG hardware accelerator and S2MM (Stream-to-Memory) channel for compressed data transfer from hardware accelerator to DDR memory.

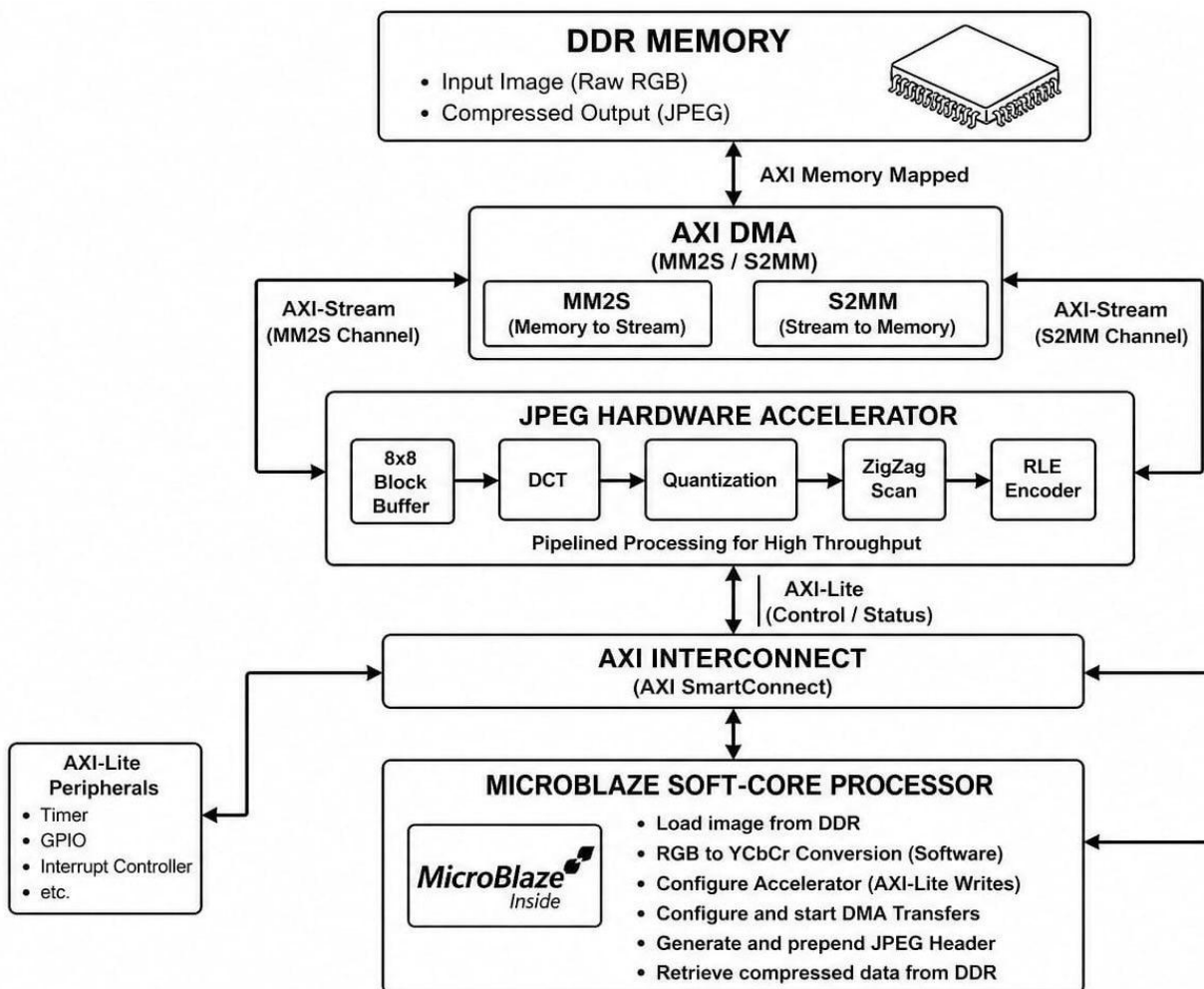


Fig -4.1: System Architecture

## 4.2 BLOCK DIAGRAM

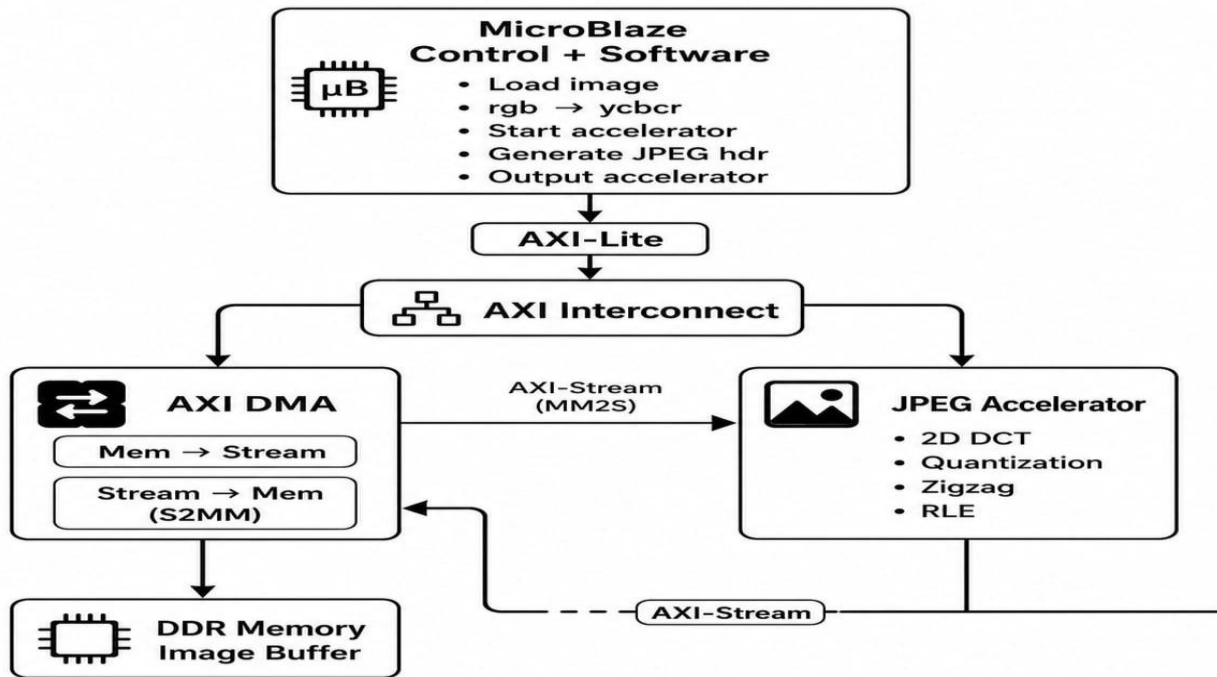


Fig - 4.2: Block Diagram

## 5. RESULT AND DISCUSSIONS

The suggested architecture was assessed for performance, resource use, and compression. JPEG compression was much faster with the hardware accelerator than with software on the MicroBlaze CPU. Testing shows that outsourcing DCT, quantisation, zigzag scanning, and RLE to hardware speeds up processing by several times, allowing near-real-time image compression for moderate-resolution photos. AXI DMA boosts performance by transferring data quickly without CPU constraints. FPGA resources are used according to parallelism, with the accelerator using logic parts, DSP slices, and block RAM. The hardware implementation maintains JPEG standard effectiveness, since the compressed pictures have acceptable visual quality. Based on quantisation settings, compression rates are equivalent to conventional JPEG implementations. The design of an FPGA-based JPEG compression technique based on hardware-software co-design technique has been successfully developed and tested on the Xilinx Artix-7 FPGA platform. In this implementation, the design incorporates the MicroBlaze processor for executing control commands, while a hardware accelerator executes all complex algorithms like the DCT, quantization, zigzag scanning, and Run Length Coding (RLE). The results show a notable improvement in speed and lower latency when compared to software implementations. Since FPGA technology is highly flexible, it provides efficient parallel and pipeline execution of all stages in the compression algorithm. Thus, by offloading the DCT, quantization, zigzag scanning, and RLE from the processor and executing them using hardware, all computationally complex stages have been offloaded from the MicroBlaze processor so that the latter can execute control commands. The hardware-software co-design technique is built on this strength and hence produces improved results as evident from the findings.

From the simulation outputs seen using waveform analysis from tools like ModelSim or Vivado, the functionality of the RTL pipeline was verified. The simulations took place within the time period of 0 to 14 milliseconds, depicting all the events from resetting of the pipeline to the generation of its output. In the first stage, the system remained in the reset state where the majority of the signal outputs were in an uninitialized state. During the second stage from 1 to 8 milliseconds, the data signals for pixels Y, Cb, and Cr toggled rapidly, representing the processing of input data by the pipeline. Finally, during the last stage from 8 to 14 milliseconds, the output signals for pixels Y, Cb, and Cr attained the values of 0xFF, 0x80, and 0x80 respectively, thus depicting pure white color in the YCbCr colour space. This was evident from the output signal of packed 24-bit data being 8080FF. The compressed images had good visual quality, thus indicating a proper trade-off between compression ratio and performance. The AXI-Stream was able to handle data transfers through the pipeline by using comp\_tvalid and comp\_tready handshake signals in order to transfer the data only when both signals are high at once.



Fig - 5: (a) Image before compression



Fig-5: (a) Image after compression

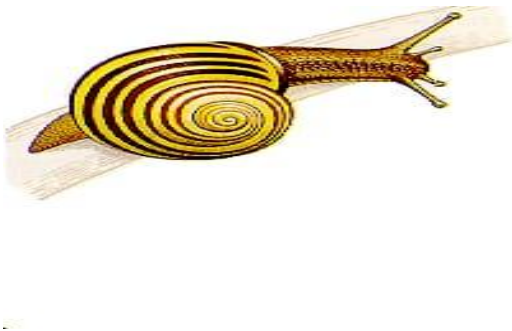


Fig -5: (b) Image before compression

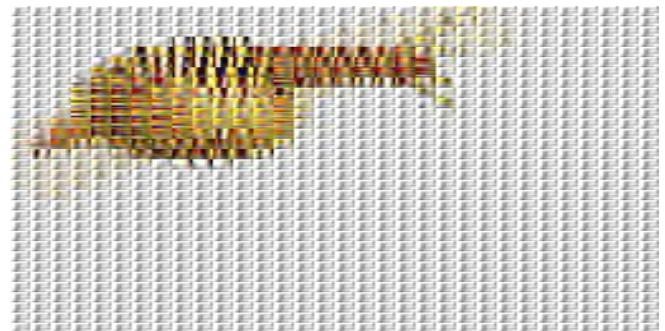


Fig-5: (b) Image after compression

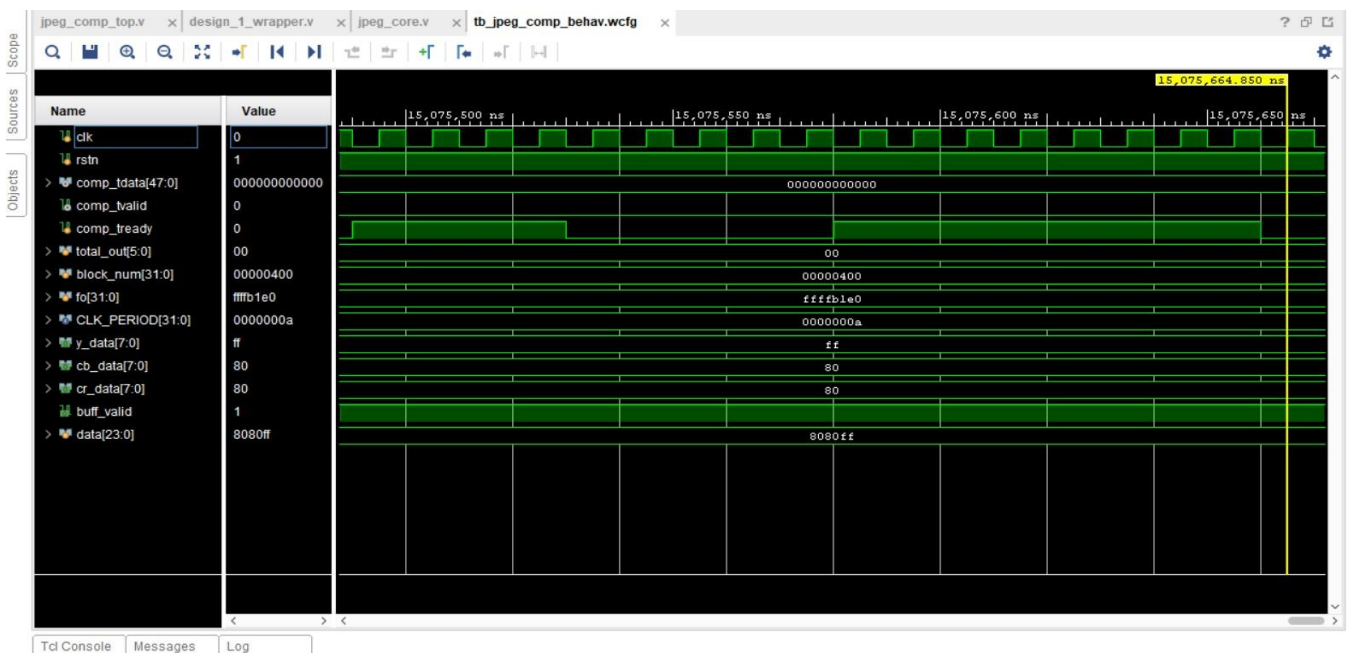
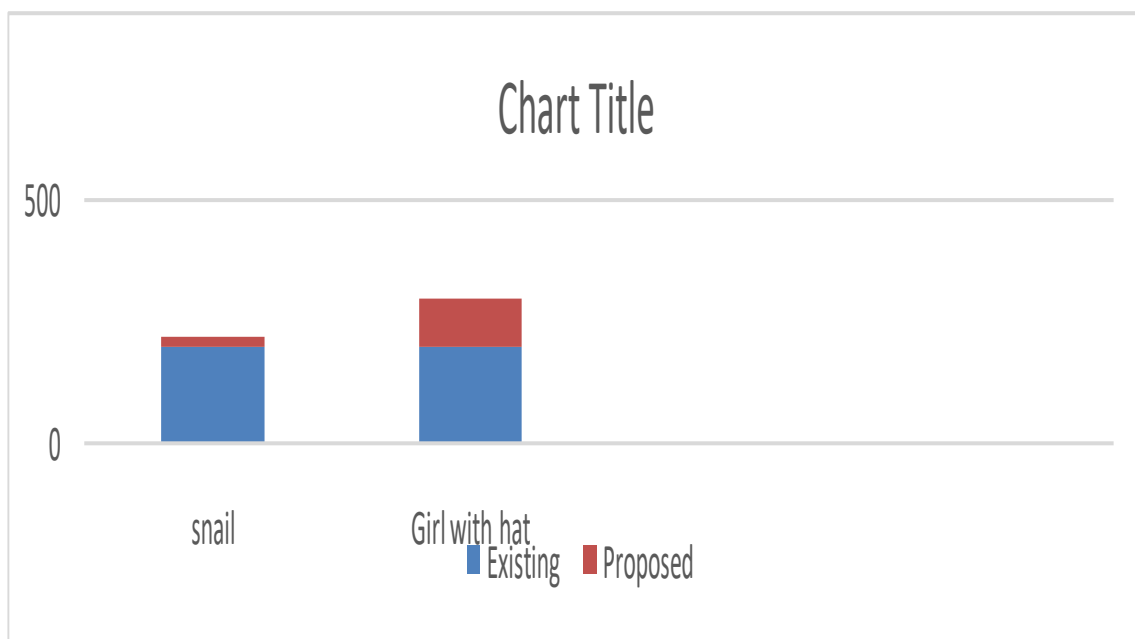


Fig-5: Simulation Result

**Table-1:** Parameter comparison of image compression1

Image	Existing	Proposed
Girl with a hat	258KB	38KB
Snail	193KB	19KB



## 6. CONCLUSION

This paper shows an effective FPGA-based hardware–software co-design method for JPEG picture compression. With the flexibility of a MicroBlaze processor and the high-performance of a specialised hardware accelerator, the system improves processing speed while using resources efficiently. AXI-based communication and DMA-driven data transfer allow software-hardware integration, decreasing processor overhead and enhancing system performance. The suggested approach is ideal for high-performance real-time image processing. Future work may optimise the accelerator architecture, enable greater picture resolutions, and include compression capabilities like Huffman encoding or advanced image formats. To summarize, the project provides a high-performance system for compressing images in JPEG format, designed based on FPGA technology by means of hardware–software co-design. Integration of the processor with a dedicated hardware accelerator provides fast, low-latency, and high-throughput operations impossible to perform in case of software-only solution. The pipelining of the hardware accelerator enables simultaneous running several stages of the compression process and efficient usage of resources. Usage of AXI interfaces and Direct Memory Access significantly increases system efficiency by providing reliable and processor-independent data communication. The developed system efficiently decreases processor load and provides good-quality compressed images, therefore being suitable for real-time applications including multimedia processing, surveillance systems, embedded systems, and communication systems. Although it has some drawbacks the design is still very efficient and scalable, and it can be employed for efficient modern image compression. As can be seen, the idea of hardware–software codesign shows itself to be quite effective, enabling the best trade-off between the speed of calculations, flexibility, and efficiency. Moreover, the proposed system is very flexible and can be modified for different applications. Therefore, one could add full Huffman encoding for fully compliant JPEG encoding; create a similar design for supporting video compression upgrade the design by using more

advanced FPGAs for increasing the performance and diminishing the energy consumption; use the design for very high-resolution images; and finally implement adaptive algorithms by using machine learning. This makes our design very attractive.

## 7. REFERENCES

- [1] Chen, Q., Li, Y., Tao, J., Chen, S., Li, L., & Liu, D. (2025). "An FPGA Implementation of Displacement Vector Search for Intra Pattern Copy in JPEG XS." arXiv preprint, arXiv:2603.10671. University of Science and Technology of China. DOI/URL: <https://arxiv.org/abs/2603.1067>
- [2] IEEE Xplore Publication (2025). "FPGA-Based Visually Lossless JPEG XS Image Encoder Architecture Design and Hardware Verification." IEEE Conference Publication (IEEE Xplore, DOI: 10.1109/11120545). DOI/URL: <https://ieeexplore.ieee.org/document/11120>
- [3] Fang, J. (2024). "A Multi-Stage Optimization Framework for Deploying Learned Image Compression on FPGAs." arXiv preprint, arXiv:2511.17135. Shanghai Jiao Tong University. DOI/URL: <https://arxiv.org/abs/2511.17135>
- [4] Mazouz, A.E., Chaudhuri, S., Cagnazzo, M., Mitrea, M., Tartaglione, E., & Fiandrotti, A. (2025). "RD Efficient FPGA Deployment of Learned Image Compression: Knowledge Distillation and Hybrid Quantization." arXiv preprint, arXiv:2503.04832. LTCI, Telecom Paris, Institut Polytechnique de Paris. DOI/URL: <https://arxiv.org/abs/2503.04832>
- [5] Mazouz, A.E. et al. (2025). "Security and Real-time FPGA Integration for Learned Image Compression." arXiv preprint, arXiv:2503.04867. LTCI, Telecom Paris. DOI/URL: <https://arxiv.org/abs/2503.04867>
- [6] Tian, S., Song, Q., He, J. et al. (2024). "Implementation of JPEG XS Entropy Encoding and Decoding on FPGA." *Journal of Real-Time Image Processing*, 21, 34. Springer. DOI: 10.1007/s11554-023-01410-8. DOI/URL: <https://doi.org/10.1007/s11554-023-01410-8>
- [7] Ghodhbani, R., Saidani, T., Horrigue, L., Algarni, A.M., & Alshammari, M. (2024). "An FPGA Accelerator for Real Time Hyperspectral Images Compression based on JPEG2000 Standard." *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13118-13123. DOI/URL: <https://www.etasr.com/index.php/ETASR/article/view/6853>
- [8] Applications & Science Journal (MDPI) (2025). "A Tile-Based Multi-Core Hardware Architecture for Lossless Image Compression and Decompression." *Applied Sciences*, 15(11), 6017. MDPI. DOI: 10.3390/app15116017. DOI/URL: <https://doi.org/10.3390/app15116017>
- [9] Deshmukh, M.S. & Khandekar, P.D. (2023). "Time Efficient FPGA Overlay for Compute Intensive JPEG Compression Blocks for Real Time Applications." *International Journal of Intelligent Systems and Applications in Engineering*, 12(5s), 258-265. DOI/URL: <https://www.ijisae.org/index.php/IJISAE/article/view/3883>
- [10] Lam, D., Quang, T.V., & Luan, P.H. (2024). "A Real-Time JPEG Image Compression Hardware Design Architecture." Conference Paper, November 2024. (Cited in ResearchGate: DOI via ResearchGate). DOI/URL: <https://www.researchgate.net/publication/224171583>
- [11] Kulkarni, P.D. & Dixit, M.M. (2025). "FPGA Implementation of Pipelined 2D-DCT and Quantization Architecture for JPEG Image Compression." Conference Paper, October 2025. (Cited in ResearchGate, original IEEE conference submission). DOI/URL: [https://www.researchgate.net/publication/224171583\\_FPGA\\_implementation\\_of\\_pipelined\\_2D-DCT](https://www.researchgate.net/publication/224171583_FPGA_implementation_of_pipelined_2D-DCT)
- [12] Yang, D. & Chen, L. (2023). "FPGA-Based Hardware Implementation of JPEG XS Encoder." In: *Digital Multimedia Communications. IFTC 2022. Communications in Computer and Information Science*, vol. 1766. Springer, Singapore. DOI: 10.1007/978-981-99-0856-1\_14. DOI/URL: [https://doi.org/10.1007/978-981-99-0856-1\\_14](https://doi.org/10.1007/978-981-99-0856-1_14)
- [13] Dang, F. (2023). "Image Lossless Compression Algorithm Optimization and FPGA Implementation." *Frontiers in Computing and Intelligent Systems*, 3(2), 51-57. DOI: 10.54097/fcis.v3i2.7194. DOI/URL: <https://doi.org/10.54097/fcis.v3i2.7194>
- [14] Agostini, L.V., Silva, I.S., & Bampi, S. (2024). "Real-Time Lossless Image Compression by Dynamic Huffman Coding Hardware Implementation." *Journal of Real-Time Image Processing*, Springer. DOI: 10.1007/s11554-024-01467-z. DOI/URL: <https://doi.org/10.1007/s11554-024-01467-z>
- [15] Sensors Journal (MDPI) (2024). "Parallel Lossless Compression of Raw Bayer Images on FPGA-Based High-Speed Camera." *Sensors*, 24(20), 6632. MDPI. DOI: 10.3390/s24206632. DOI/URL: <https://doi.org/10.3390/s24206632>