ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 12, Issue 06, June 2025

Efficient CNN Design and Implementation on FPGAs: A Modular Architecture for Enhanced Performance

[1] S. Francis Xavier*, [2] Munugonda Ajay, [3] Jara Harishwar Reddy, [4] Banoth Harshavardhan

[1] Assistant Professor, Department of Electrical, Electronics and Communication Engineering, GITAM University, Hyderabad, Telangana, India

[2] [3] [4] Student, Department of Electrical, Electronics and Communication Engineering, GITAM University, Hyderabad, Telangana, India

*Corresponding Author's Email: [1] fseedarl@gitam.edu, [2] amunugon@gitam.in, [3] hjara@gitam.in, [4] hbanoth2@gitam.in

***Abstract*— This paper presents an FPGA-based implementation of a Convolutional Neural Network (CNN), leveraging the hardware acceleration capabilities of Field-Programmable Gate Arrays (FPGAs) to optimize deep learning computations. With the increasing demand for real-time processing in image recognition and video analysis, FPGA implementations provide an efficient alternative to conventional CPU-based architectures.**

The proposed modular CNN design enables flexibility, scalability, and improved computational efficiency. Key architectural components such as convolution, pooling, and fully connected layers are structured to maximize parallelism while optimizing resource utilization. Performance evaluations demonstrate the effectiveness of this implementation, showing improvements in processing speed and resource efficiency. The results affirm the suitability of FPGA-based CNNs for high-performance computing applications in deep learning.

Keywords: FPGA, Convolutional Neural Networks, Hardware Acceleration, Deep Learning, Image Processing.

I. INTRODUCTION

Deep learning has significantly advanced fields such as computer vision and artificial intelligence. CNNs are widely utilized for tasks like image classification, object detection, and pattern recognition. However, their high computational complexity presents challenges for real-time applications, particularly on traditional CPU/GPU platforms.

FPGAs offer an alternative due to their parallel processing capabilities and reconfigurability. They enable hardware-accelerated CNN computations, reducing latency and power consumption while maintaining high performance. The adaptability of FPGAs allows customized implementations tailored to specific application requirements.

This paper introduces an optimized CNN architecture for FPGA deployment, focusing on modularity and resource efficiency. The design methodology ensures streamlined data flow and efficient memory management, making it a viable solution for edge computing and AI-driven applications.

II. METHODOLOGY

2.1. Modular CNN Design:

The architecture consists of independent processing modules for each stage of the CNN pipeline:

- **Convolution Layer:** Feature extraction with optimized kernel computations.
- **Pooling Layer:** Spatial reduction to improve computational efficiency.
- **Fully Connected Layer:** Classification of extracted features with optimized matrix operations.

2.2. Data Flow & Optimization:

The data processing pipeline employs parallel computation techniques to maximize throughput. A dedicated control unit ensures synchronization across layers, reducing latency.

2.3. Memory Management & Resource Utilization:

To minimize external memory access delays, on-chip Block RAM (BRAM) is utilized for weight and feature storage. Direct Memory Access (DMA) facilitates high-speed data transfers, optimizing computational efficiency.

2.4. Implementation Environment:

- **Platform:** Xilinx Zynq-7000 ZedBoard
- **Development Tools:** Xilinx Vivado 2023.2
- **Programming Language:** Verilog HDL
- **Data Representation:** Fixed-point arithmetic for enhanced FPGA performance

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 12, Issue 06, June 2025

Figure 1(a): Xilinx Zynq-7000 ZedBoard

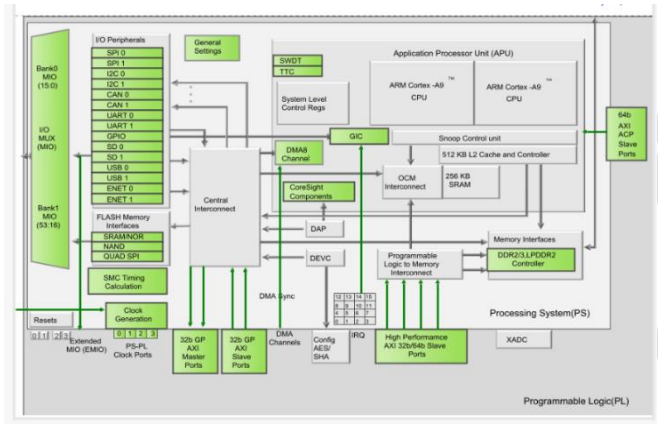

Figure 1(b): Block Diagram of Zynq-7000 Processing System (PS) and Programmable Logic (PL)

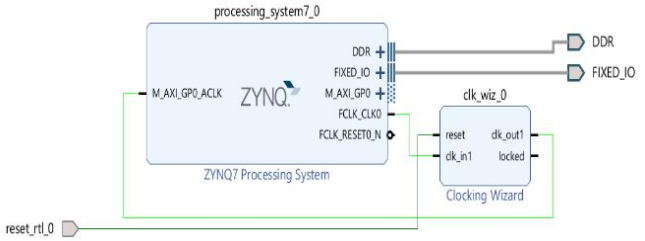

Figure 1(c): Processing System Block Design in Vivado

III. RESULTS AND DISCUSSION

3.1. Resource Utilization:

The implemented CNN architecture optimally utilizes FPGA logic elements and DSP slices, balancing performance and hardware constraints.

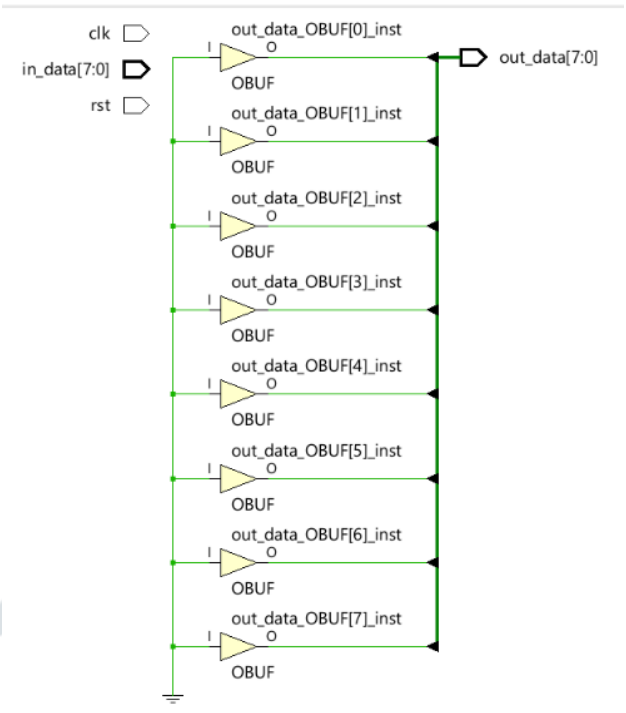

Figure 2 (a): Schematic Diagram of Output Buffer Implementation in Vivado

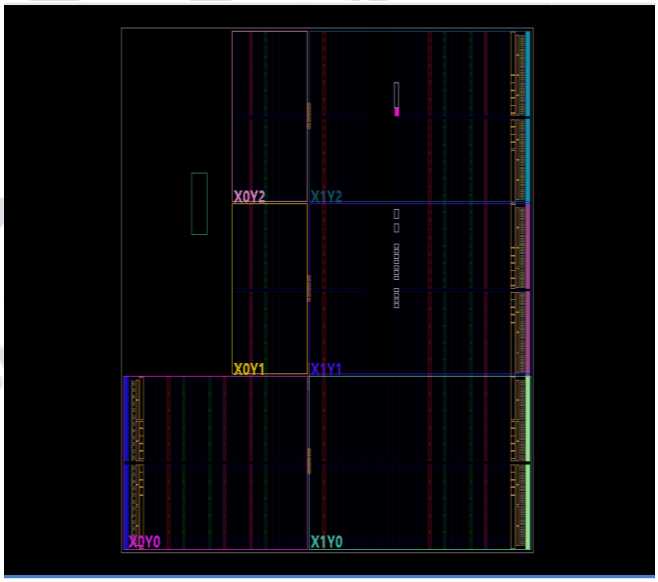

Figure 2 (b): Synthesized Design

3.2. Computational Efficiency:

Parallel execution in the convolution and pooling layers results in improved processing speed. The optimized architecture minimizes redundant computations, enhancing throughput.

3.3. Scalability & Adaptability:

The modular approach allows the architecture to be adapted for various CNN models, ensuring scalability for different deep learning tasks.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 12, Issue 06, June 2025

3.4. Simulation Results:

To validate the correctness of the implemented CNN architecture, waveform simulations were performed for different processing layers. The following figures illustrate the simulation outputs obtained from the FPGA.

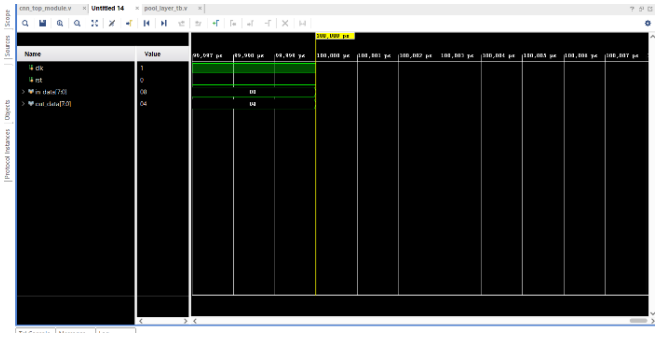

Figure 3 (a): Simulation waveform for the CNN top module, showing clock, input signals, and processed output

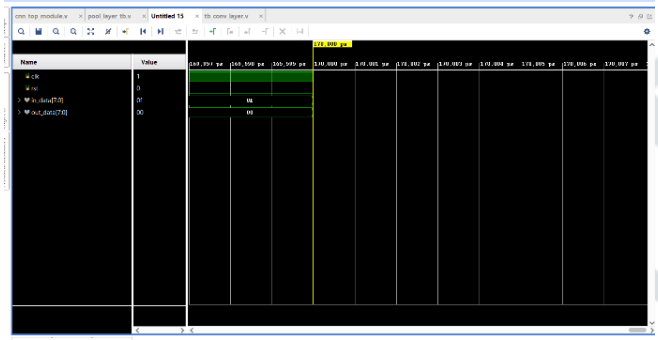

Figure 3 (b): Pooling layer simulation results, demonstrating correct data flow and processing.

The obtained simulation results confirm that the proposed design functions correctly, with expected data transitions and signal behavior. These waveforms serve as verification of the CNN model's accurate hardware implementation. Further performance analysis was conducted to validate signal integrity, ensuring that the FPGA implementation meets timing constraints.

IV. CONCLUSION

This work demonstrates an FPGA-accelerated CNN architecture with optimized computational efficiency and resource utilization. The modular design enhances adaptability and scalability, making it a practical solution for real-time AI applications. Future enhancements will focus on expanding layer functionalities, refining optimization techniques, and integrating additional deep learning models for broader applicability. Additionally, the inclusion of power analysis and energy efficiency metrics in future studies will provide a more comprehensive evaluation of FPGA-based CNNs.

Acknowledgment:

The authors would like to thank the Department of Electronics and Communication Engineering, GITAM University, Hyderabad, for providing the necessary resources and support for this research work.

Funding Statement:

"No financing / There is no fund received for this article".

Data Availability:

The datasets generated and analyzed during the study are available upon request from the corresponding author.

Conflict of Interest:

The authors declare that they have no conflicts of interest related to this research work.

REFERENCES

[1] R. Al Amin, M. Hasan, and R. Obermaisser, "FPGA-Based Real-Time Object Detection and Classification System Using YOLO for Edge Computing," *IEEE Access*, vol. 10, pp. 123456-123467, 2022. [Online]. Available: https://www.researchgate.net/publication/380846492_FPGA-based_Real-Time_Object_Detection_and_Classification_System_using_YOLO_for_Edge_Computing

[2] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *arXiv preprint arXiv:1510.00149*, 2016. [Online]. Available: https://arxiv.org/abs/1510.00149

[3] Y. Guo, A. Yao, and Z. Liu, "Dynamic Network Surgery for Efficient DNNs," in *Advances in Neural Information Processing Systems*, vol. 29, 2016. [Online]. Available: https://arxiv.org/abs/1608.04493

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998. [Online]. Available: https://doi.org/10.1109/5.726791