Highly Efficient Hand-Written Digit Recognition using Multilayer Perceptrons through ZyNet and its Hardware Implementation

Guided by:
Mr. Aditya Kumar Hota
Assistant Professor

Submitted by:
Sampad Mohanty
2002070059

Department of Electronics and Telecommunication Engineering
Veer Surendra Sai University of Technology, Burla, Odisha

TABLE OF CONTENTS

Introduction

- Artificial Intelligence (AI) revolutionizes sectors like healthcare and finance by simulating human intelligence in machines, driven by neural networks inspired by the human brain's structure.
- Neural networks, composed of interconnected artificial neurons, learn from data, recognize patterns, and make decisions without explicit programming, making them ideal for image processing tasks.
- Image processing benefits immensely from AI and neural networks, enabling accurate and efficient analysis, interpretation, and manipulation of visual data across various domains.
- Applications of AI in image processing include medical imaging for disease diagnosis, surveillance systems for security, and object detection in autonomous vehicles.
- Multilayer Perceptrons (MLPs) remain fundamental, especially in scenarios requiring interpretability and simplicity.
- MLPs, a type of Artificial Neural Network (ANN), automatically extract features and recognize patterns in image processing applications such as computer vision and pattern recognition.
- Understanding MLPs is crucial for unlocking their potential in various image processing tasks, including image classification, object detection, and image segmentation.

Problem Formulation

- Resource constraints on FPGA platforms pose challenges for implementing neural networks due to limited logic elements, memory resources, and computational capabilities.
- Neural networks, particularly deep learning models like Multilayer Perceptrons (MLPs), require substantial computational resources and memory storage for weight parameters and activations.
- FPGAs offer finite resources, necessitating optimization of neural network implementations to operate efficiently within these constraints.
- Accommodating complex computations and memory requirements of neural networks while maintaining performance and accuracy is crucial for FPGA-based implementations.
- Optimizing hardware designs involves balancing the trade-offs between resource utilization, computational efficiency, and neural network performance.
- Addressing these challenges requires innovative methodologies and techniques to maximize the utilization of FPGA resources while ensuring the effective deployment of neural network models for various applications.

Overview of Proposed Methodology

Development of a novel methodology for optimizing Multilayer Perceptrons (MLPs) to enhance their efficiency and performance on FPGA platforms.

Exploration of innovative techniques for reducing resource utilization while maintaining accuracy and computational efficiency of MLP models.

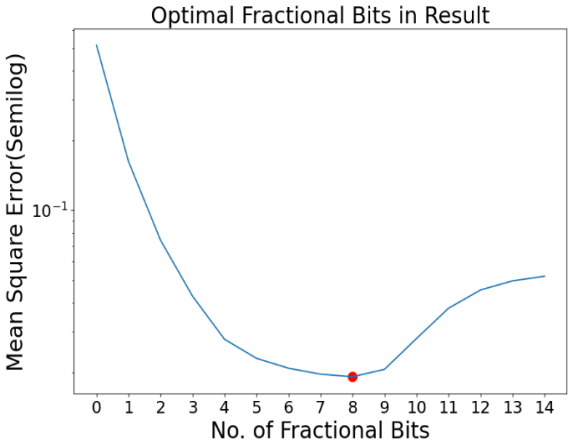
Investigation of architectural optimizations and algorithmic improvements tailored specifically for FPGA-based implementations of MLPs.

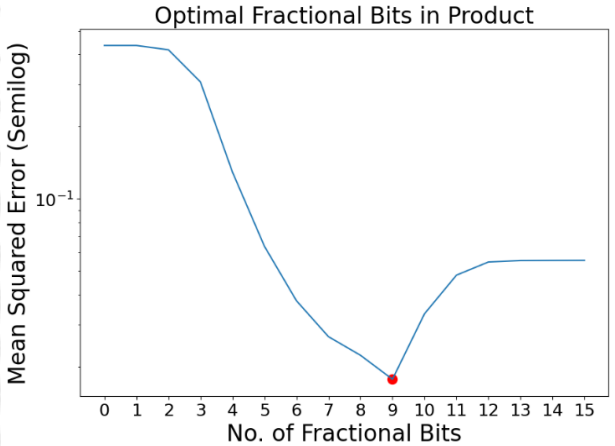
Integration of the proposed optimizations into the ZyNet framework, facilitating seamless deployment and synthesis of optimized MLP designs on FPGA devices.

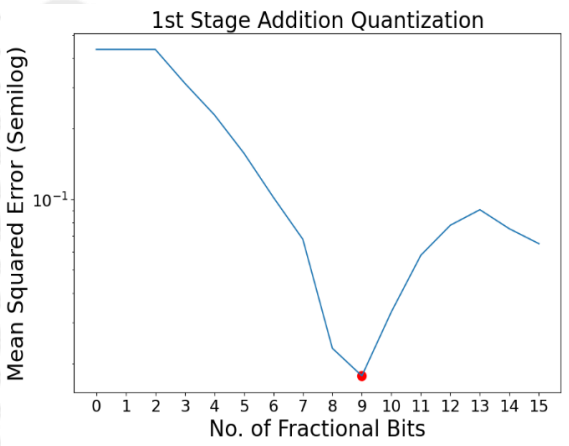
Evaluation of the proposed methodology through extensive experimentation, assessing its impact on resource utilization, computational speed, and overall performance metrics.

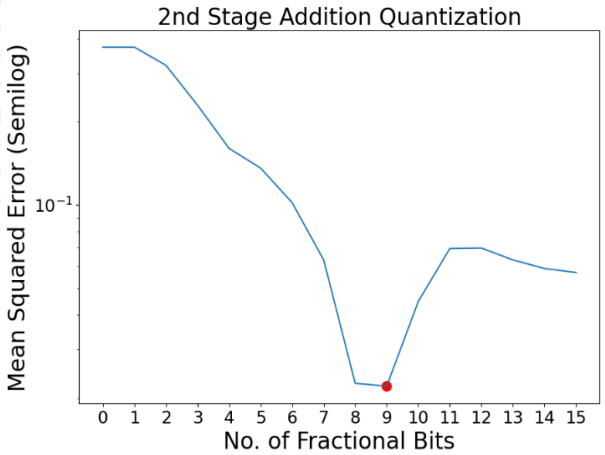
Validation of the effectiveness of the proposed approach through comparative studies with existing methodologies, demonstrating its advantages in terms of resource efficiency and deployment feasibility for real-world applications.

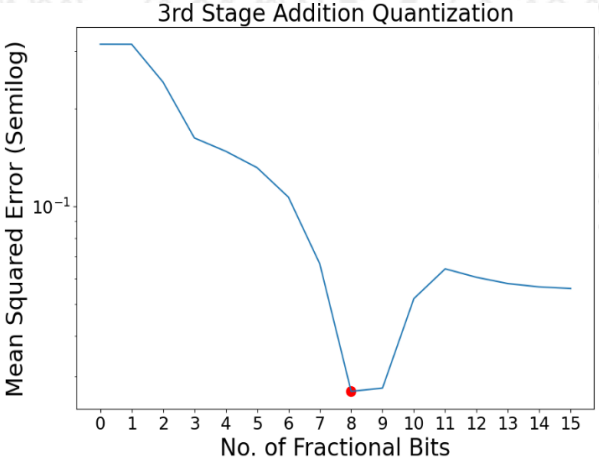
Quantization Analysis



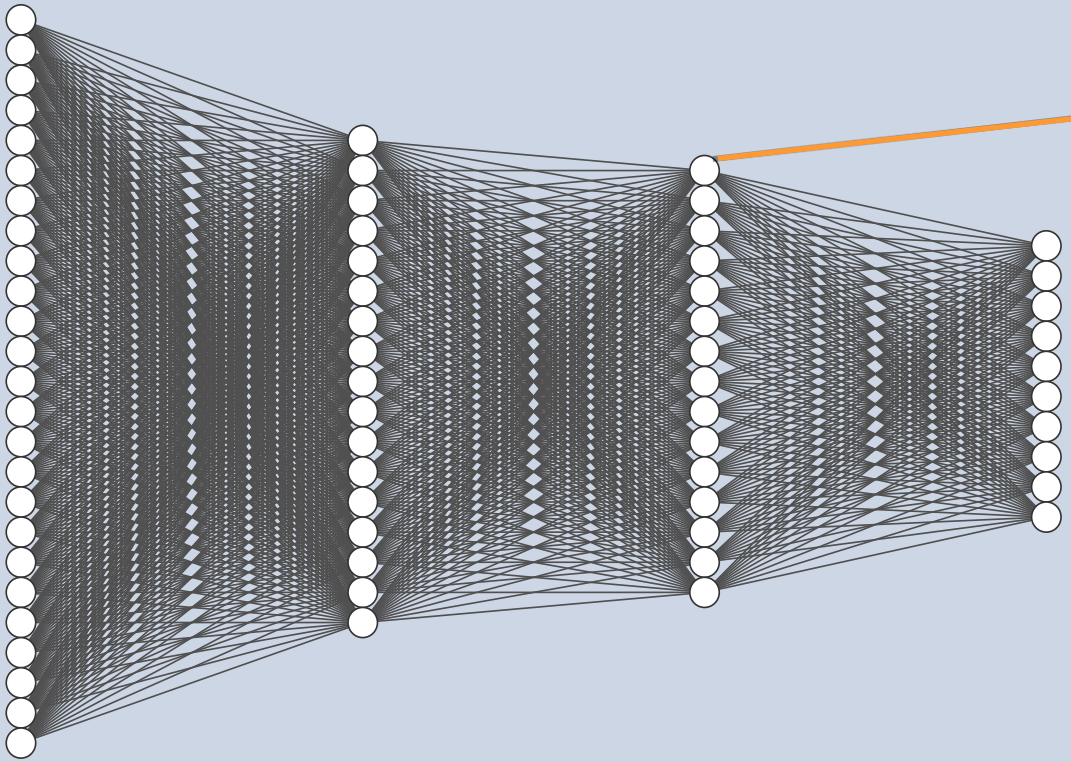


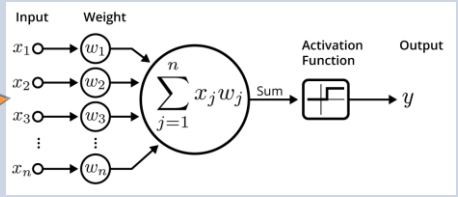






Multi-Layer Perceptron Model Architecture





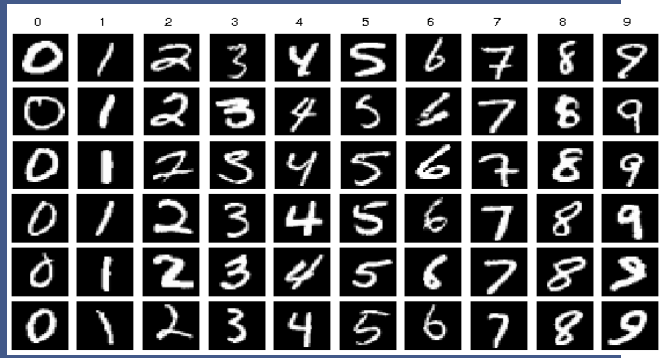
Input Layer
784

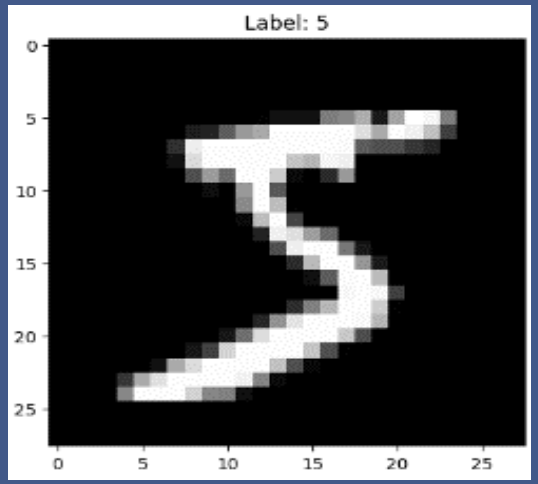
Hidden Layer
30

Hidden Layer
20

Hidden Layer
10

The MNIST Dataset





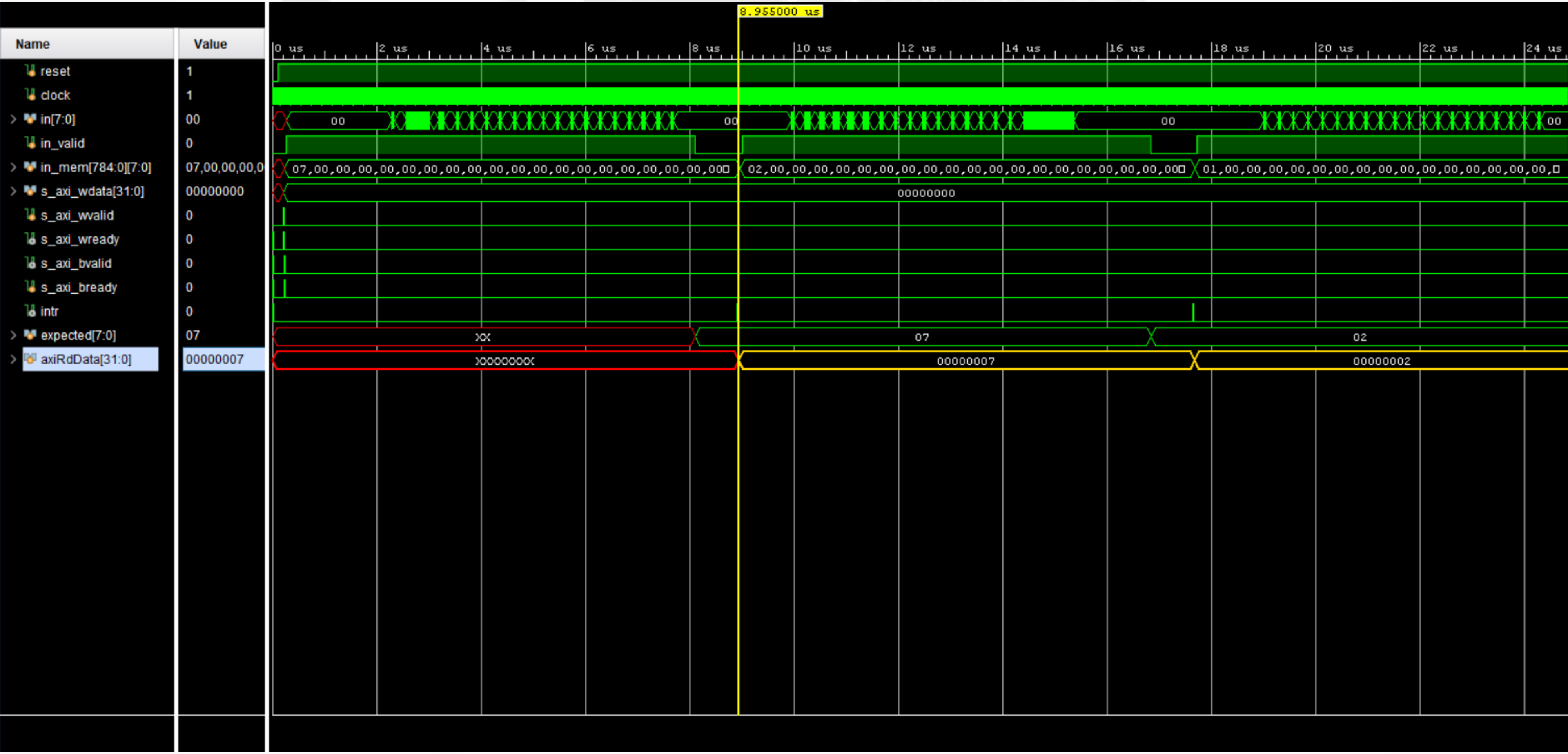
- MNIST dataset, established by Yann LeCun, Corinna Cortes, and Christopher J.C. Burges in 1998, is a cornerstone in machine learning and computer vision.
- Comprising 70,000 hand-written digits ranging from zero to nine, MNIST serves as a standard for digit recognition tasks.
- The dataset includes 60,000 training images and 10,000 test images, each sized at 28x28 pixels, enabling comprehensive model evaluation.
- MNIST's balanced distribution ensures fair assessment of model generalization, crucial for benchmarking and comparing algorithms.

ZyNet

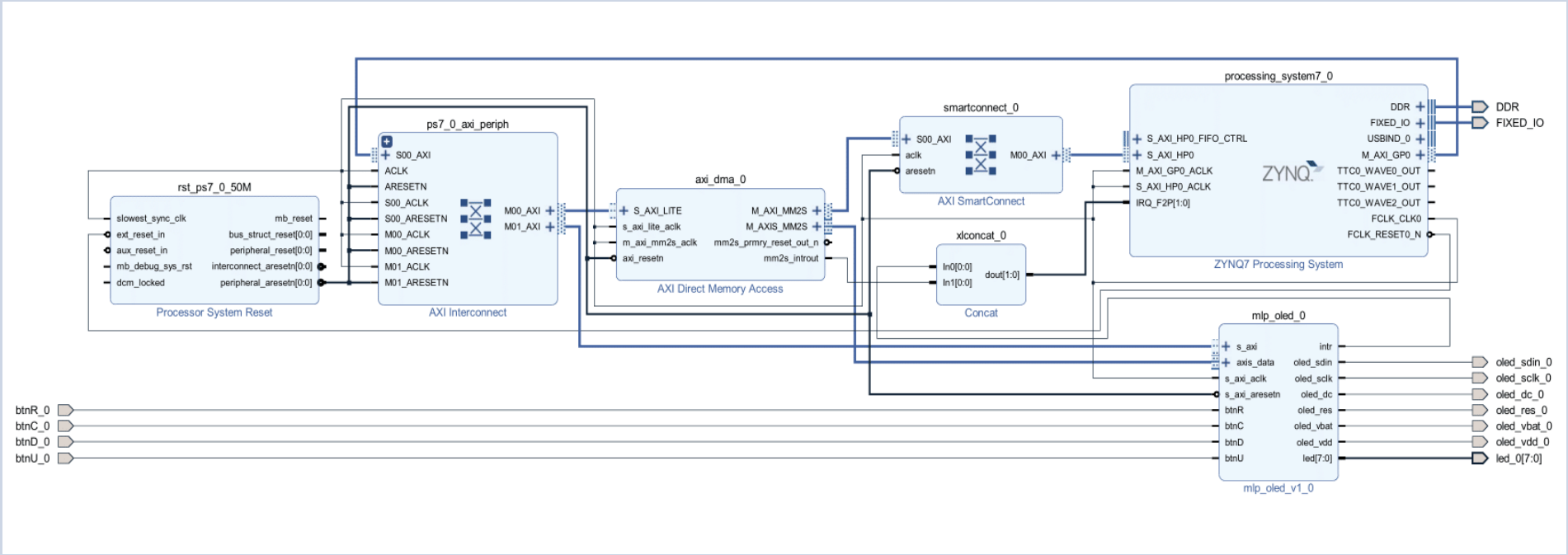
Framework for Python to HDL conversion

- ZyNet offers advanced synthesis techniques optimized for FPGA architectures, enabling the generation of highly efficient hardware implementations of neural networks.
- The framework incorporates innovative resource allocation and optimization strategies to minimize hardware resource utilization while maximizing performance, making it suitable for resource-constrained FPGA platforms.
- ZyNet provides users with the flexibility to customize neural network architectures and synthesis parameters to meet specific application requirements, ensuring optimal performance for diverse use cases.
- With its intuitive design environment and extensive toolset, ZyNet tops the competition with MyHDL, Nngen, and other frameworks.
- It also accelerates the development cycle of FPGA-based neural network applications, allowing researchers and developers to focus on algorithm design and optimization.

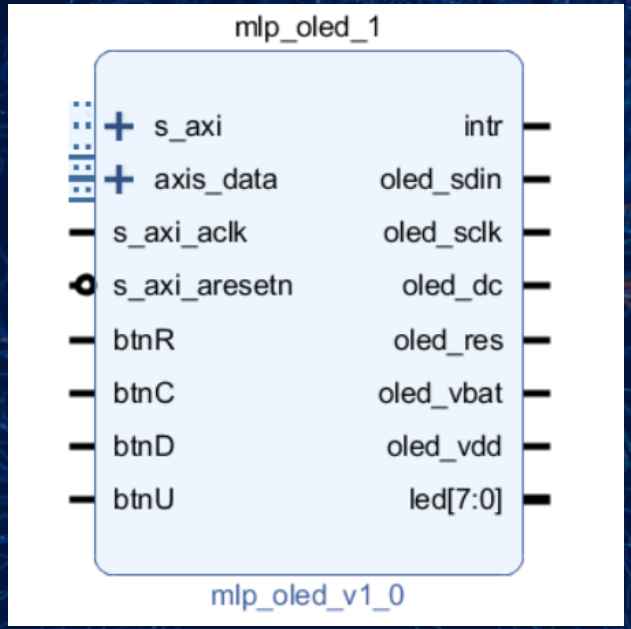
Simulation Results



Generated Block Design

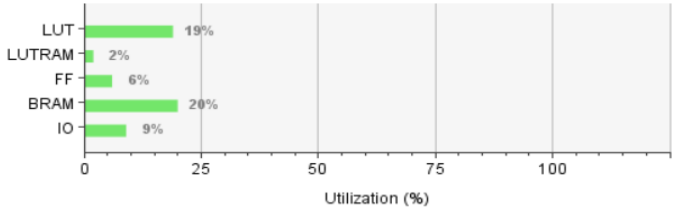


Generated IP



Reports related to synthesized design

Resource	Utilization	Available	Utilization %
LUT	10301	53200	19.36
LUTRAM	312	17400	1.79
FF	6660	106400	6.26
BRAM	28.50	140	20.36
IO	18	200	9.00

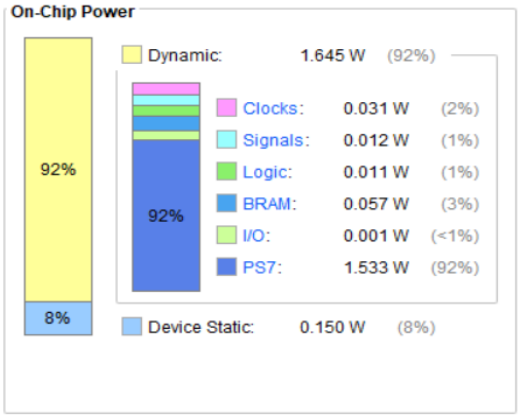


Pad	Max Delay	Max Edge	Max Process Corner	Min Delay	Min Edge	Min Process Corner	Edge Skew
led_0[0]	7.467	Rise	SLOW	5.728	Rise	SLOW	0.000
led_0[0]	3.504	Rise	FAST	2.372	Rise	FAST	0.000
led_0[1]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[1]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
led_0[2]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[2]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
led_0[3]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[3]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
led_0[4]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[4]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
led_0[5]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[5]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
led_0[6]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[6]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
led_0[7]	-∞	Rise/Fall	SLOW	∞	Rise/Fall	SLOW	-
led_0[7]	-∞	Rise/Fall	FAST	∞	Rise/Fall	FAST	-
Worst Case Summary	7.467	Rise	SLOW	5.728	Rise	SLOW	0.000
Worst Case Summary	3.504	Rise	FAST	2.372	Rise	FAST	0.000

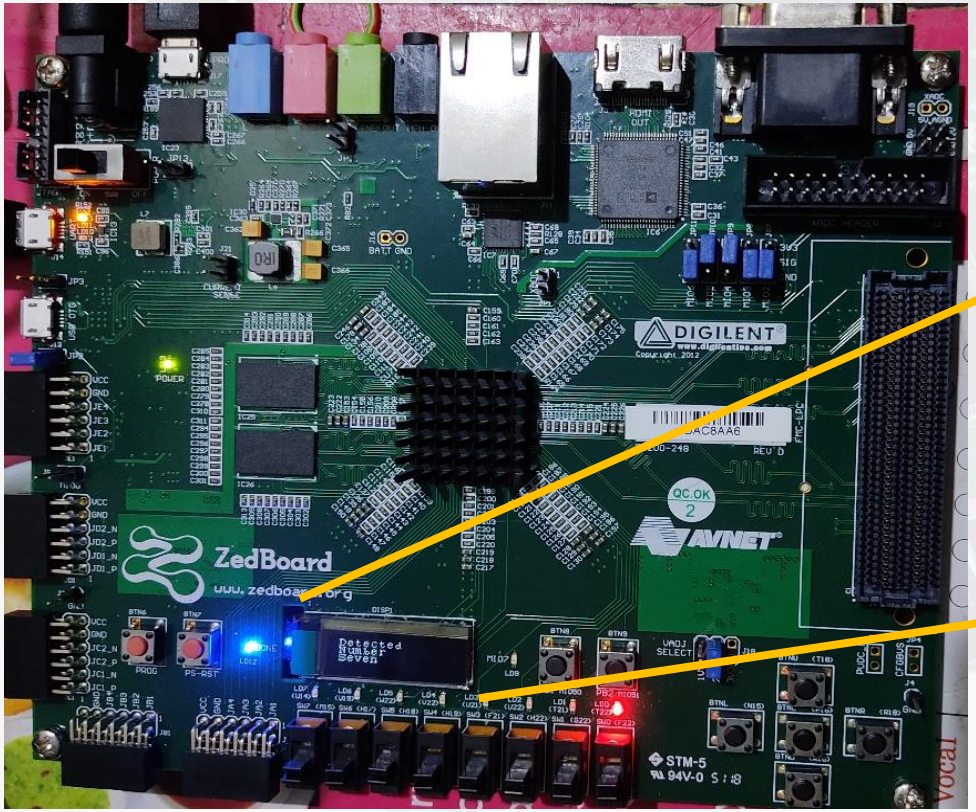
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**1.795 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**45.7°C**
Thermal Margin:	39.3°C (3.3 W)
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



Launching the application on Zedboard Zynq xc7z020clg484-1



Comparison with earlier works

Model	FPGA	Accuracy	Hardware Costs (LUTs used)	Latency
LeNet-5 CNN	Xilinx Zynq 7Z020-100	90-96%	18,426	3.2 ms
DNN	Xilinx Zynq 7Z020-100	94.67%	38,899	637 us
CNN	Intel Cyclone10-150	97.57%	12,588	17.6 us
SS-CNN	Intel Cyclone IVE-30	98.8%	98,000	220 us
LeNet-5 CNN HLS	Xilinx Zynq 7Z020-100	98.64%	33,585	3.58 ms
Proposed	Zynq xc7z020clg484-1	99%	10,301	8.96 us

Conclusion

- **Performance Advancements**: The proposed method showcases significant improvements in accuracy and resource utilization compared to existing solutions, achieving a remarkable 99% accuracy while consuming only 10301 LUTs on the Zedboard Zynq7000 platform.
- **Comparative Efficiency**: Comparative analysis with early works demonstrates the superior efficiency of the proposed method, outperforming various neural network models in terms of both accuracy and resource utilization across different FPGA platforms.
- **Power Efficiency**: Additionally, power consumption analysis reveals exceptional efficiency, with the power report indicating a minimal usage of only 1.795W, highlighting the energy-efficient nature of the implemented design.

Future Scope

- Multilayer Perceptrons (MLP) show promise in efficiently utilizing hardware resources, but Convolutional Neural Networks (CNN) maintain dominance in tasks like image processing and computer vision due to their superior feature extraction abilities.’

- The ZyNet framework, tailored for MLP implementation, lacks support for efficiently accommodating CNN architectures, highlighting a need for a more comprehensive framework.

- Future efforts aim to develop a versatile framework capable of converting diverse neural network architectures, including CNNs and RNNs, into Hardware Description Language (HDL) representations.

- By enabling seamless conversion of neural network models into hardware-accelerated implementations, the future framework will empower developers and researchers to leverage FPGA-based acceleration across various machine learning tasks and applications.

References

[1] L. Deng, “The MNIST database of handwritten digit images for machine learning research,” *IEEE Signal Process Mag*, vol. 29, no. 6, pp. 141–142, 2012, doi: 10.1109/MSP.2012.2211477.

[2] K. Vipin, “ZyNet: Automating Deep Neural Network Implementation on Low-Cost Reconfigurable Edge Computing Platforms,” in *Proceedings - 2019 International Conference on Field-Programmable Technology, ICFPT 2019*, Institute of Electrical and Electronics Engineers Inc., Dec. 2019, pp. 323–326. doi: 10.1109/ICFPT47387.2019.00058.

[3] M. Cho and Y. Kim, “Implementation of Data-optimized FPGA-based Accelerator for Convolutional Neural Network,” in *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, 2020, pp. 1–2. doi: 10.1109/ICEIC49074.2020.9050993.

[4] T.-H. Tsai, Y.-C. Ho, and M.-H. Sheu, “Implementation of FPGA-based Accelerator for Deep Neural Networks,” in *2019 IEEE 22nd International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, 2019, pp. 1–4. doi: 10.1109/DDECS.2019.8724665.

[5] H. Madadum and Y. Becerikli, “FPGA-Based Optimized Convolutional Neural Network Framework for Handwritten Digit Recognition,” in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, 2019, pp. 1–6. doi: 10.1109/UBMYK48245.2019.8965628.

[6] R. Xiao, J. Shi, and C. Zhang, “FPGA Implementation of CNN for Handwritten Digit Recognition,” in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020, pp. 1128–1133. doi: 10.1109/ITNEC48623.2020.9085002.

[7] J. Si, E. Yfantis, and S. L. Harris, “A SS-CNN on an FPGA for Handwritten Digit Recognition,” in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019, pp. 88–93. doi: 10.1109/UEMCON47517.2019.8992928.

THANK YOU