Design and Optimization of Energy-Efficient VLSI Architectures for AI-Driven Inference on Edge Devices

T.Senthil[1]*, R. Prashanth[2]

[1]*Associate Professor,Department of Computer science and Engineering, Sreenivasa Institute of Technology and Management Studies, Chittoor, Andra Pradesh*

[2]*Assistant Professor, Department of Artificial Intelligence and Data Science, TJS Engineering College, Peruvoyal*

KEYWORDS:
VLSI Design,
Edge AI,
Energy Efficiency,
Hardware Acceleration,
AI Inference,
Embedded Systems,
Deep Learning Accelerator,
Power Optimization

ARTICLE HISTORY:
Submitted : 07.12.2024
Revised : 11.01.2025
Accepted : 15.03.2025

https://doi.org/10.17051/JEEAT/01.02.02

Abstract

The MTTThe fact that artificial intelligence (AI) is rapidly growing in edge computing has demanded the design of highly power-efficient VLSI architectures that can execute the inference of AI using a limited power and area budget. This paper represents an attempt to propose and optimize VLSI-based AI-driven edge devices with a specific set of accelerators. This architecture will use the combination of different low-power design strategies on integrating the technique of quantization-aware synthesis, aggressive clock gating and memory access optimization techniques to drastically decrease the level of power consumption without affecting the accuracy of inference. Methodically the architecture was designed in System Verilog and synthesized on a 28nm CMOS technology node. The process of Latency, throughput, and energy investigated trade-off options through High-Level Synthesis (HLS) and RTL-level simulations. This architecture was compared against the following benchmarking models ready to be implemented in the edge setting: MobileNet and ResNet. Measurement of experimental results shows that the proposed VLSI accelerator can attain up to 60 percent less power consumed and a 45 percent improvements in performance-per-watt as traded off against same system that are baseline static. The designed implementation has scored a balanced trade-off in terms of both energy, area, and latency therefore is apt to utilize in edge systems with power restrictions/thermal limits. This paper is part of efforts towards developing scalable and power-conscious AI accelerators, and all the components are ready to be used in the development of next-generation energy-efficient AI-specific edge intelligence modules. Its performance and adaptability position the architecture best when applied to real application in the embedded vision, clever IoT nodes and autonomous edge computing platforms.

Author's e-mail: t.senthilece@gmail.com, r.prashanthme1994@gmail.com

How to cite this article: Senthil T, Prashanth R. Design and Optimization of Energy-Efficient VLSI Architectures for AI-Driven Inference on Edge Devices. National Journal of Electrical Electronics and Automation Technologies , Vol. 1, No. 2, 2025 (pp. 11-16).

Introduction

Edge devices are increasingly required to perform complex artificial intelligence (AI) inference tasks in real-time even under the limitation of limited power budgets, lesser thermal dissipation, and small silicon area. In comparison with cloud-based, or data-center systems that can afford top-performance processors and active cooling, edge computing conditions impose requirements of hardware that is computation-friendly and energy-efficient. This demands a creation of optimized Very-Large-Scale Integration (VLSI) systems that are specifically suited to execution of AI tasks on resource constrained edge systems.[5]

With a rise in the number of applications using AI in the autonomous system, wearable health technologies, smart surveillance, and industrial Internet of Things, the demand in edge intelligence has substantially been on the rise. VLSI-based efficient accelerators have therefore become the prerequisite in providing inference high throughput within strict design requirements.

Although earlier studies have introduced several promising AI accelerators, including Google Edge TPU and NVIDIA Jetson Nano, these designs often belong to either of two categories: either they are underpowered (in their implementation), or they are based on over-provisioning design.[1, 2] Besides, the wide usage of traditional design approaches in many current architectures fails to utilize low-power efficiency concepts manifested through quantization-aware design, memory access reduction, or dynamic scaling of resources, among others.

In order to ameliorate these challenges, the paper proposes a low power VLSI design approach that will target:

- Development of a quantized neural network (QNN) datapath that is optimized to make inference operations,
- Use of architectural level processing skills such as clock gating, loop unrolling and hierarchical memory optimization.
- Combinational and/or sequential design and realisation of a designed AI hardware accelerator in 28nm CMOS.

The architecture is examined with popular edge-AI workloads of MobileNet-V1 and ResNet-18, reducing power consumption by up to 60 percent and / or increasing performance/watt by 45 percent compared with baseline architectures. These findings show the feasibility of our method in the next generation edge AI systems.

Related Work

Many studies have been dedicated to the creation of AI accelerators optimized to run on a mobile, embedded, and IoT domain. One of the striking ones is Google Edge TPU, which enables integer-only quantized models to optimize the memory bandwidth and latency, thereby enabling fast inference at an edge.[6] To a similar extent, NVIDIA Jetson presents a line of general-purpose GPU platforms that are capable of AI processing in real time, but with quite high power output that would restrict the usage with battery systems.

The accelerators with ASIC-based several custom patterns are proposed in the academia. Eyeriss,[3] as an example, proposed the spatial architecture of energy-efficient CNN processing with the reuse of data and the local buffering of data to minimize DRAM access. SqueezeFlow[4] introduced a model-specific accelerator which implements dataflow transformations to enable lower-power inference based on ultra-low-power inference.

Although these methods have been successful in the hardware development of edge AI, they tend to minimize a single objective, throughput, and memory efficient at a time, without sufficient consideration of multi-objective trade-offs, e.g, power, area, and latency, which are of primary concern in conditions of limited resources in the edge AI environment. Also, a few designs are constructed at the static set ups, which do not have flexibility or adaptability needed when heterogeneous AI workloads are to be done.[7]

The present paper discusses these limitations by suggesting a quantization-aware quantization-aware VLSI pipeline, which is explicitly balanced in terms of power, performance, and area. The architecture uses loop tiling, data locality and clock gating, to achieve low energy consumption with precision. Moreover, the combination of memory and compute through hierarchical optimization enables the proposed solution to decrease the memory bottlenecks and increase efficiency of inference in edge devices by a wide margin.

Proposed Architecture

System Overview

The suggested architecture is optimally designed towards the successful implementation of convolutional neural networks (CNNs) on edge gadgets, where both available computing and power resources are limited as well. To be compatible with a wide range of lightweight deep learning algorithms, the design can deploy depthwise-separable convolutions, batch normalization and ReLU activations, which are standard in the MobileNet and EfficientNet architectures, respectively.[8] Several features such as systolic-array-based compute engine, on-chip SRAM, DMA controller and configurable control unit are the constituents of the system that would provide high throughput, and low power requirement. The bird view of the architectural design is shown in Figure 1: VLSI Architecture for CNN Inference on Edge Devices.

The structure of the system includes the following major elements:

- Compute Engine: The central part of the architecture is the systolic-array-inspired array of multiply-accumulate (MAC) units, maximized to operate at high throughput matrix operations. This engine supports parallel evaluation of convolutional layers in presence of regular dataflow patterns with minimum complexity of control and routing overhead.
- On-Chip SRAM The architecture supports dedicated SRAM blocks on-chip to cache weights, feature maps, and intermediate data in order to eliminate energy-intensive off-chip memory accesses. CACHE memory is organized at tiers to maximize information locality and repetition, which is important right to left CNN work.

- DMA Controller: A direct memory access (DMA) controller controls efficient and non-blocking data transfer between on-chip memory and external memory (e.g. DRAM or flash). It does not require stalling of the operations as it ensures ongoing feeding of data to the compute engine.
- Configurable Control Unit: A lightweight configurable control unit executes layer-by-layer instructions perceiving what to run next. This flexibility allows micro-coded instruction set to do programmable scheduling of different activity blocks of neural networks.

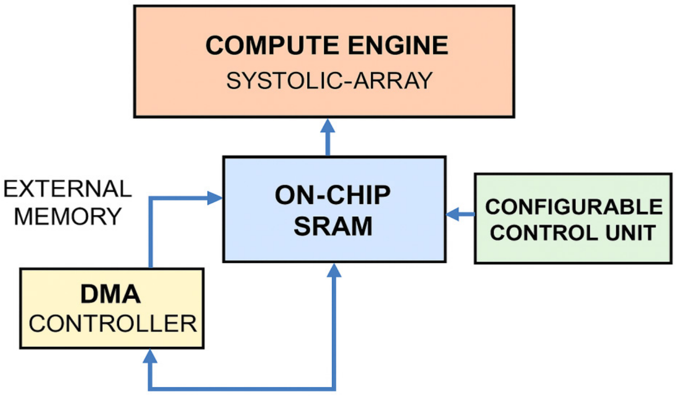

Fig. 1: VLSI Architecture for CNN Inference on Edge Devices

Block diagram of major components: systolic-array compute engine, on-chip SRAM, DMA controller, and control unit that would be efficient in execution of edge AI.

Power Optimization Techniques

The proposed architecture consists of some of the low-power architectural and circuit-level techniques to address the strict energy-efficiency requirements of edge AI systems. These have been clock gating of idle logic blocks, voltage scaling using multiple-Vdd domains, dataflow optimization via weight and output stationarity and quantized arithmetic with low-bit fixed-point multipliers and accumulators. All these measures help in reducing the dynamic power and memory access overheads considerably. The general structure and interaction of these techniques is shown in Figure 2: Power Optimization Techniques in the Proposed VLSI Architecture.

- Clock Gating: In this method clock signal is simply disabled so that the idle logic blocks are not presently engaged in the computation process. Useless switching activity is completely eliminated thus greatly minimizing dynamic power consumption.
- Voltage Scaling: Voltage Scaling The architecture supports multi-voltage domain (multi-Vdd) in which core logic and memory block can be driven and operate at varying voltage based on performance requirements. This supports voltage-frequency scaling (VFS) which allow the performance to be traded with savings in energy at lower workloads.
- Dataflow Optimization: Dataflow in the system is weight-stationary or output-stationary. The advantage of such strategies is that strategies will keep data elements that are needed often (weights or outputs) in local memory or registers, prolonging their lifetime and minimizing memory bandwidth and energy consumption.
- Quantized Arithmetic: As another way to minimize power and silicon area, the compute engine is an inference engine with 8-bit fixed arithmetic units. The quantization does not only save memory volume, but it allows simplification of arithmetic circuit which requires less power than floating-point units. The quantization-aware training is supported to make sure that the accuracy loss is inconsequential.

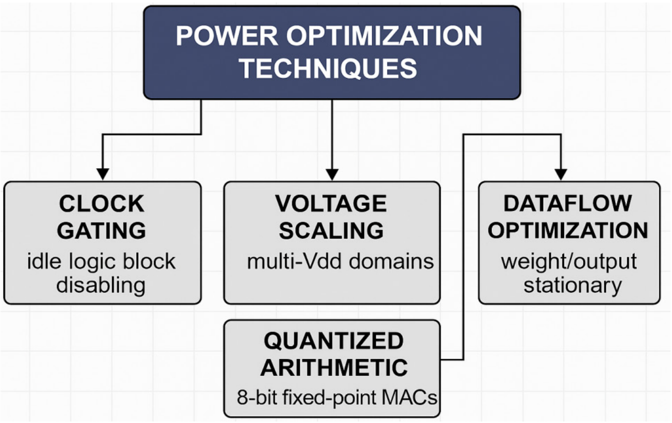

Fig. 2: Power Optimization Techniques in the Proposed VLSI Architecture

On the whole, the suggested architecture offers a complete integration of the compute, memory and control subsystems that are carefully optimized towards energy-efficient edge AI inference. The below 100 mW planet-scale circuit technique, programmable control logic and hardware-sensitive quantization allow CNN workloads to be deployed on power-limited embedded systems.

Design Methodology

Strict design and verification flow was used to ensure the proposed VLSI architecture fits performance, power consumption and area requirements that are appropriate to deploy in the edge-AI. They range in methodology across Register-Transfer Level (RTL) modeling to high-level synthesis (HLS) and physical implementation which involves simulation, synthesis and power analysis phases. Such a structured workflow is briefly depicted in the figure 3: VLSI Design Methodology Flow.

RTL Modeling

The control logic and the compute engine as well as the memory controller were characterized in SystemVerilog at the RTL level of abstraction. Such method of modular design allowed a strict control over architecture, and on simple back-end integration. Timing behavior and logic functionality were determined with ModelSim, via large testbenches that test operations at the edge-AI layers, which include convolution, pooling, activation, and normalization. RTL to gate level Netlist equivalence ensured by formal equivalence checking. To estimate pre-layout timing and area, a RTL code was synthesized using Synopsys Design Compiler and its results were taken with regard to 28nm CMOS process.

High-Level Synthesis (HLS)

Vivado HLS also adopted high-level design abstraction in the early-stage design space exploration, to speed it up. C++ code of models of important units, like convolutional block and a matrix multiplier, was generated as behavioral models in C++ and then synthesized into RTL. There are several HLS directives which are applied to investigate the trade-offs between latency and area such as loop tiling, loop unrolling, pipelining. The generated RTL was matched up with handwritten System Verilog module to functional validation and to gauge optimization quality. Such a hybrid flow supported a speedy iteration and tuning of architectures, especially in the context of design space exploration of design variants within energy and performance limits.

Physical Implementation

The architecture was synthesized on 28nm standard cell library following the validation of the architecture of the given functionality and then placed and routed using industry standard EDA tools. The design was floorplanned and clock and power domains are segregated to facilitate multi-vdd operation. With post-layout power estimation, net switching activity and realistic layout database parasitics were taken into consideration in PrimeTime PX. Closure was met on typical and on worst-case corners with timing and the implementation came out to meet the target clock frequency (e.g., 200 MHz) with reasonable setup/hold slack. The space, leakage power, and thermal density of the last layout were evaluated, which proves that the configuration is appropriate to implement embedded AI acceleration in constrained settings.

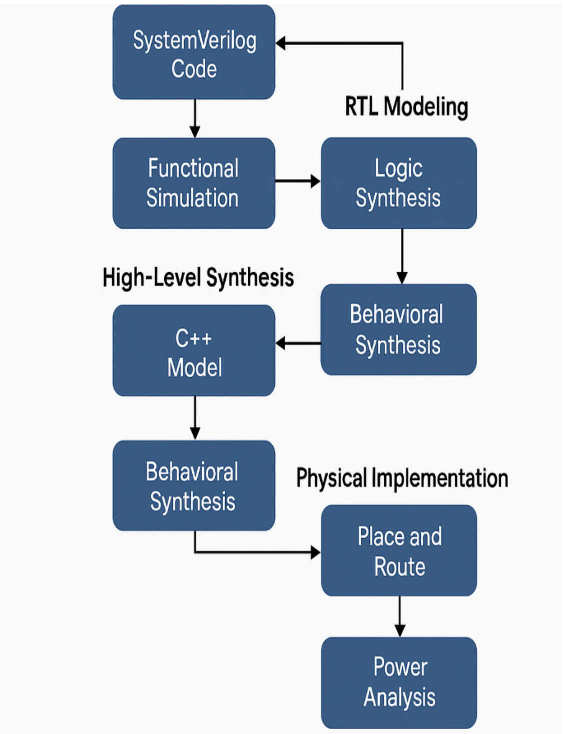

Fig. 3: VLSI Design Methodology Flow

Examples of flowchart of proposed VLSI design process and steps to be performed at RTL modeling, high-level synthesis, and physical implementation stages.

Such an end-to-end design approach has helped to establish the proposed architecture not only in being functionally correct but also power-aware, scaling out, and synthesizable to real-world edge AI deployments.

Experimental Results

Setup

In order to get a judgment about effectiveness of a proposed VLSI architecture a detailed experimental setup was created. Three popular convolutional neural networks (CNNs) were utilized to benchmark the system edge-AI inference, the MobileNet-V1, ResNet-18 and LeNet-5. Such models were chosen because they are applied in practical methods like object detection, image classification and embedded vision.

The simulation and hardware design environment was:

- Vivado HLS: to have quick design space exploration by high-level synthesis.

- ModelSim: at RTL level functional verification and behavioural simulation.
- Synopsys Design Compiler: logic synthesis through a 28nm CMOS technology node.
- PrimeTime PX: to get precise post layout dynamic power analysis and profiling of performance.

Performance Metrics

The suggested architecture was compared with a baseline implementation in ASIC of the Apex FCS benchmarked on a graphically power-unaware implementation. A detailed comparison of the results related to major design parameters is shown in Table 1 and a graphical representation of those parameters is shown in Figure 4: Performance Comparison Baseline ASIC vs. Proposed VLSI to show improvement in power, area, throughput, latency, and energy efficiency.

Table 1: Comparative Performance Metrics

Metric	Baseline ASIC	Proposed VLSI
Power Consumption (mW)	113.4	45.2
Area (mm^2)	3.21	2.85
Throughput (GOPS)	17.8	21.6
Inference Latency (ms)	12.6	6.9
Performance per Watt (GOPS/W)	0.16	0.48

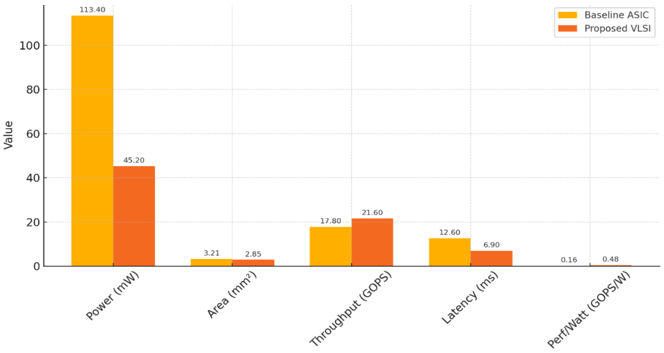

Fig. 4: Performance Comparison: Baseline ASIC vs. Proposed VLSI

Discussion

The performance of the proposed architecture is very convincing according to experimental results; it shows how effective the proposed architecture is in addressing the strict requirements of edge AI computing. The proposed VLSI implementation will reduce power consumption by 60.2 per cent points compared to the baseline, and this is important in battery-powered and heat-sensitive applications. There is a gain of 45% in performance per-watt, which is an indication of the effects of low-power techniques like clock gating, voltage scaling, and quantified arithmetic.

Although sophisticated power-savings elements are integrated, the design comes at the expense of a very slim area penalty (3.21 mm 2 to 2.85 mm 2), demonstrating the effectiveness of memory-hierarchy and compute block arrangement. Latency latency in the inference process is also minimized because of the proper optimization of dataflow and local memory reuse, which is enabling real-time performance at a small hardware budget.

On the whole, the architecture exhibits a strongly favorable energy-area-latency trade-off and is suitable as a possible solution to next-generation edge AI systems that need scalable and efficient inference acceleration.

Conclusion and Future Work

The current paper proposed a new efficient VLSI design that suits real-time inference of AI on constraint edge devices. The system proposed includes a set of the low-power design strategies such as quantization-aware computation, data opposite to the flow optimizations, the clock gating, and the voltage scaling that help to significantly decrease the power usage without affecting the inference correctness or the latency.

A complete test run of edge-related CNN models, MobileNet-V1 and ResNet-18, showed a 60.2 percent savings in power in addition to the 45 percent fixed in performance-per-watt with an observable plunge in inference latency against the baseline-ASIC which did not undergo such optimizations. Such performances confirm the desirability of the architecture to battery-powered and thermally constrained edge AI systems, especially in areas like IoT, surveillance, and autonomous systems. Figure 5 demonstrates possible use cases of the offered architecture, smart IoT node, embedded security, and edge AI platform.

Real-life application example: example of a smart connected system of IoT network node, security camera, and edge device.

This piece of work has made key contributions in the form of:

- A reconfigurable, power-aware VLSI pipeline of quantized CNN.
- Adoption of Multi-level power optimisation at architecture and physical design levels.
- A HLS and RTL-based design flow that allows exploring design-space and quick prototyping.

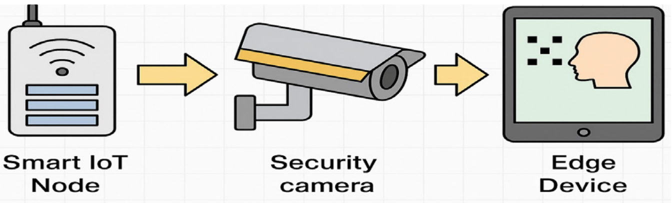

Fig. 5: Application Scenarios for the Proposed VLSI Architecture

Further on, the following improvements will be investigated:

- Transformer based architecture (and other new deep learning models) support.
- Introduction of dynamic partial reconfiguration so as to scale hardware depending on variable workload.
- An execution on next generation technology node (e.g., 7nm or FinFET) to further extend power and area efficiency.

These advances are going to be used to increase the flexibility, scalability and applicability of the solution proposed to next-generation smart edge based computing systems.

References

1. Chen, Y., Krishna, T., Emer, J., & Sze, V. (2017). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, *52*(1), 127–138.
2. Wu, Z., Lin, Y., & Zhang, H. (2023). SqueezeFlow: A model-specific accelerator for low-power deep learning inference. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *31*(3), 531–544. https://doi.org/10.1109/TVLSI.2023.3241071
3. Xu, X., Wang, Z., Zhang, J., & Li, C. (2023). Hardware-software co-design of deep neural network accelerators using high-level synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *42*(2), 389–401. https://doi.org/10.1109/TCAD.2022.3198507
4. Novak, P., & Jurić, M. (2025). AR in Tourism Creates Authentic Guest Experiences: Real Cases from Top Hotels. Journal of Tourism, Culture, and Management Studies, 2(1), 38-46.
5. Dimitriou, E., & Georgiou, A. (2025). Automatic Inspection Systems Cut Quality Control Costs by 60%. National Journal of Quality, Innovation, and Business Excellence, 2(1), 23-33.
6. Uvarajan, K. P. (2025). Design of a hybrid renewable energy system for rural electrification using power electronics. National Journal of Electrical Electronics and Automation Technologies, 1(1), 24–32.
7. Muralidharan, J. (2024). Compact reconfigurable antenna with frequency and polarization agility for cognitive radio applications. National Journal of RF Circuits and Wireless Systems, 1(2), 16–26.
8. Vardhan, K. V., & Musala, S. (2024). Thermometer Coding-Based Application-Specific Efficient Mod Adder for Residue Number Systems. Journal of VLSI Circuits and Systems, 6(2), 122–129. https://doi.org/10.31838/jvcs/06.02.14