Journal of Integrated VLSI and Signal Processing • ISSN 3139-2369 (O) • Jan - Apr • 1 • 1 • 2026 (pp. 42-49)

RESEARCH ARTICLE

# Hardware-Efficient VLSI Design of AI-Enhanced Signal Processing Pipelines for Resource-Constrained Platforms

**Charpe Prasanjeet Prabhakar***

Department Of Electrical And Electronics Engineering, Kalinga University, Raipur, India

**KEYWORDS:**
VLSI architecture,
AI-assisted signal processing,
Hardware efficiency,
FPGA,
Embedded systems,
Resource-constrained platforms.

**ARTICLE HISTORY:**
**Received** : 11.11.2025
**Revised** : 22xxxx.12.2025
**Accepted** : 13.01.2026

## Abstract

Cross-sectional adaptability, accuracy, and robustness of embedded and edge computing applications have been facilitated by the growing use of artificial intelligence (AI) in signal processing. However, the real implementation of AI-enhanced signal processing pipes on resource-constrained environments is still a major challenge because there are strict constraints on power consumption, silicon area, memory bandwidth, and real-time latency. The paper has offered a proposal of a set of hardware effective VLSI design methodology used to design AI-enhanced signal processing pipelines specifically designed to run on low-power, resource-limited embedded systems. The suggested approach assumes an algorithm-hardware co-design architecture which narrowly wraps traditional signal preprocessing phases with a slim AI inference device as part of a profoundly-pipelined and parallel VLSI design. In order to realize high hardware efficiency, it uses multiple hardware considerations such as fixed-point arithmetic, quantization-aware model design of AI models, dataflow pipelining that is optimized, and buffering strategies that have low memory usage. The entire architecture is derived on an FPGA platform and tested on post-implementation conditions in order to provide realistic performance evaluation. It has been experimentally shown that the proposed architecture provides significant gains in throughput and energy efficiency over traditional non-optimized and non-AI baseline architectures with competitive inference accuracy. These findings affirm the tight design of AI inference through a hardware-optimized signal processing pipeline to highly decrease the latency and resource overhead. All in all, the suggested VLSI architecture offers a relevant scalable solution to making real-time AI-assisted signal processing possible in resource-constrained embedded and edge systems.

**Author's e-mail:** charpe.prasanjeet.prabhakar@kalingauniversity.ac.in

**How to cite this article:** Prabhakar CP. Hardware-Efficient VLSI Design of AI-Enhanced Signal Processing Pipelines for Resource-Constrained Platforms. Journal of Integrated VLSI and Signal Processing. Vol.1, No. 1, 2026 (pp. 42-49).

## Introduction

Signal processing pipelines form the computing base of numerous in-built systems, such as biomedical instrumentation, wireless communication units, industrial observance units, and multimedia environments. Such pipelines are also marked by the use of deterministic algorithms and ad-hoc feature extractors whose design is made based on a set of assumptions on signal statistics. Although useful in controlled situations, these traditional methods can be characterized by low flexibility when faced with non-stationary signals, variability in noise as well as complicated trend patterns of real world data experienced in contemporary embedded systems.

Artificial intelligence (AI) has become a potent paradigm in boosting signal processing activities in recent years, like classification, denoising, detection, and prediction. The performance of AI models using data is superior to the traditional signal processing method since performing automatic features reduction and adaptable decision-making is possible.[1, 2] The use of AI as a signal processing enhancement is, therefore, emerging as a highly appealing idea in edge and embedded platforms that require intelligent real-time response. Nevertheless, the use of AI-based signal processing pipelines on resource-limited platforms is a significant issue despite these benefits. The inherent computational complexity is coupled with memory requirements, which render AIs hard to execute on edge devices, moreso IoT nodes and battery-constrained

embedded systems with severe power, area, and latency requirements.[3] General-purpose processors and GPUs will not typically fit such platforms, because their power usage is too large and execution time is not predictable. As a promising solution to both these problems, it has been discovered that hardware-accelerated VLSI implementations are ideal at supporting real-time AI-rich signal processing with constrained resource space. [4, 5] Nevertheless, one of the major weaknesses of the majority of current designs is that signal processing and AI inference are handled as a detached or separated unit. This division causes inefficientness when it comes to the movement of data, unnecessary access to memory, and unnecessary latency as well as poor use of the hardware resources available. Additionally, some of the previous literature concentrate on individual AI accelerators with scanty attention given to end-to-end signal processing chains and hardware efficiency.

The paper will resolve these shortcomings by suggesting an integrative, resource-efficient VLSI design approach that brings signal processing and AI inference on the same pipelined platform to the optimality level based on resource availability. The proposed architecture combines algorithm and hardware, leading to an algorithm hardware co-design style that has allowed the architecture to maximize computational accuracy and precision, dataflow, memory access and parallelism in order to maximize throughput and minimize energy usage.

### Contributions

The main findings of this paper will be as follows:

- A single framework of how to design AI-based signal processing streams using hard-time constraints.
- Hardware-efficient VLSI based on pipelining, parallelism and fixed point computation.
- Quantization-aware integration of a lightweight AI model Hardware-aware integration of a lightweight AI model with quantization-aware optimization.
- Implementation and comparative analysis of FPGA with traditional baseline architectures.

The rest of this paper will be structured in the following way. Section 2 is a review of relevant literature in VLSI-based signal processing and AI acceleration. Section 3 shows proposed system model and design methodology. Section 4 outlines the hardware-efficient VLSI architecture as well as Section 5 outlines the AI integration strategy. Section 6 talks about the experimental setup, analysis and results are discussed in Section 7. Lastly, the paper ends with Section 8 which provided the directions of future research.

## Related Work

The study of VLSI architectures in digital signal processing (DSP) is not a recent development: much attention has been given to pipelined and parallel methods in order to achieve better throughput and shorten the computing latency. Application customized DSP accelerators have been suggested to be used in the execution of filtering, transform computation and feature extraction in communication and multimedia systems in early and recent works.[6, 7] These architectures are highly efficient with specific tasks, but are not very flexible in response to variations in signal properties and altered application needs because of their fixed-function nature. As artificial intelligence has evolved at a noteworthy pace, a number of works have been conducted regarding the use of AI-assisted signal processing, in which the machine learning and deep learning algorithms are utilized to optimize the processes of classification, denoising, and detection. The accuracies of the software-based implementations that use CPUs or GPUs have been shown to be better, in comparison with classical DSP approaches.[8, 9] Nevertheless, these solutions do not usually fit embedded and edge platforms since they have too much power consumption, have too much memory needs, and variable latency. To address these shortcomings, neural network hardware accelerators have been introduced, such as FPGA-based and ASIC-based neural network AI inference engines that are optimized based on their power consumption.[10, 11] Although these accelerators result in much less inference latency and power consumption, most of them are packaged as deployable processing blocks. Consequently, signal preprocessing and AI inference are normally run as disjointed phases, which causes inefficient information transport, higher memory requests, and extra latency duties. Harder and more recent studies have made an emphasis on hardware-algoric co-design methods to be able to fit AI models to platforms with resource constraints. Quantization, pruning, and model compression methods have been extensively followed to make computational complexity and memory footprint significantly lower with only a minor accuracy loss.[12, 13] Regardless of all these developments, current literature is focused primarily on the optimization of the AI model, and little consideration is paid to the overall synthesis of the AI inference in the entire signal processing chains.[14]

To conclude, the current studies do not have a single VLSI design methodology that deeply combines signal processing and AI inference into an identical dataflow that is optimized around hardware efficiency. Lack of such end-to-end architectures implies poor use of hardware resources and reduces real time performance on limited hardware platforms. The current paper is associated

with these challenges and attempts to find a solution to them by providing a hardware-efficient, closely-coupled VLSI architecture that simultaneously optimizes signal processing and AI inference in an algorithm-hardware co-design approach.

## System Model and Design Objectives

The section outlines the system model and methodological framework that the given work should use in designing an AI-enhanced signal processing pipeline inherent to resource-constrained platforms. The suggested system is supposed to run in real-time on continuing streaming signals and produce intelligent results with strict constraints on hardware resources, power consumption, and latency. The general architecture is based on an algorithm-hardware co-design philosophy, signify processing axi quality objectionable and AI inference both combined together to be efficiently implemented on VLSI. The proposed AI-enhanced signal processing pipeline and streaming dataflow are represented in the high level system model in Fig. 1.

### System Model

The proposed system model is a streaming signal processing pipeline with three closely linked phases, including signal acquisition and preprocessing, feature extraction and normalization, and AI-assisted inference with decision logic, like in Figure 1. Where x(n) is the discrete time input signal that was obtained through a sensor or data source. During the preprocessing stage, the raw signal is conditioned through operations such as filtering, windowing, or normalization, resulting in a processed signal $x_p(n)$.

In the second stage, discriminative features are extracted from $x_p(n)$ to form a feature vector $f \in R^K$, where K denotes the number of extracted features. The feature normalization is implemented to enhance the numerical stability and robustness, as performs as follows:

$$f_n = \frac{f - \mu}{\sigma}. \quad (1)$$

where μ and σ represent the mean and standard deviation computed during training. The normalized feature vector $f_n$ is directly streamed to the AI inference stage.

The last step conducts AI-based inference using a neural network model that is lightweight to produce an output decision y as follows:

$$y = M(f_n; \theta) \quad (2)$$

where M(·) denotes the trained AI model parameterized by θ. As Figure 1 shows, the pipeline as a whole is to be executed in a fully streaming fashion, thus, all stages consume data through pipelined execution at once in order to reduce end-to-end latency.

### Design Objectives

There are four main objectives of the design methodology. To meet real-time requirements in edge and embedded applications, at first, low-latency operation is needed. Second, it requires energy efficiency to be deployed on devices with rare energy and battery-powered operations. Third, the hardware efficiency is a concern to reduce the logic use, the memory size and the interconnect complexity. Lastly, the scalability is taken into account in order to enable the proposed architecture to match various types of signals, feature sizes, and AI model configurations without significant redesign.

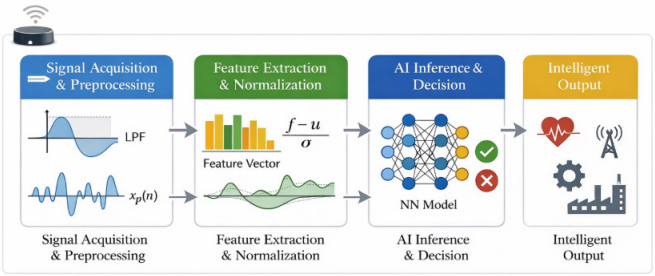


**Fig. 1: System Model of the AI-Enhanced Signal Processing Pipeline**

An architectural diagram of the system model of the proposed AI-based signal processing pipeline, including signal capture and pre-processing, feature extraction and normalization, and closely coupled AI-based inference to real-time intelligent decision-making on limited resources.

## Hardware-Efficient VLSI Architecture

The fundamental value of this work is the fact that it proposes a hardware-efficient VLSI architecture that implements a proposed AI-enhanced signal processing pipeline. A deep and modular architecture is embraced in order to implement concurrent implementation of various processing steps without sacrificing throughput and low energy usage.

### Pipeline Architecture

The signal processing pipeline is physically divided into each individual stage with a standard streaming interface. It is based on a synchronous model of dataflow whereby information is processed at each stage based on the order of arrival at a stage, as it flows through the pipeline at any given clock cycle after filling. Let $T_i$ denote the processing

latency of stage i. Pipelining ensures that the overall system throughput changes to be limited by the largest stage latency and not the total of the latencies, as:

$$Throughput = \frac{1}{\max(T_i)} \quad (3)$$

The design guarantees the flow of data and removes idle cycles usually experienced in the sequential or loosely coupled architecture. The timing closure at the high-clock frequency with the help of register insertion in between stages, with a critical path isolation.

### Hardware Optimization Techniques

In order to realize hardware efficiency, all fixed-point computation is used in place of floating-point computation in the pipeline. Each signal and intermediate variable is represented using a fixed-point format $Q_{m,n}$, where m and n denote the integer and fractional bit widths, respectively. Signal sensitivity analysis is used to select the bit-width in order to minimize artifacts such as precision reduction and hardware complexity. Local on-chip buffers incorporate local on-chip memory access and increase the energy efficient off-chip memory-oriented access so as to optimize the memory access. Additionally, computational blocks with a high level of parallelism like feature extraction and neural networks multiply accumulate (MAC) are introduced with some selective parallelism. The parallel processing elements are determined to provide an adequate throughput enhancement versus area and power overhead. When these hardware optimization mechanisms are combined as shown in Figure 2, it can be seen that fixed-point arithmetic, local buffering, and selective parallelism were useful in enabling a compact and power-efficient VLSI implementation without affecting functional correctness.

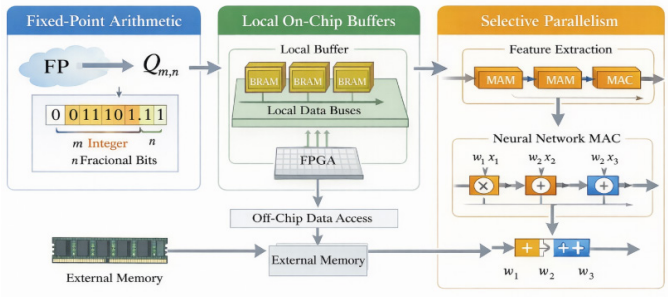


**Fig. 2: Hardware Optimization Techniques for Resource-Efficient VLSI Implementation**

A conceptual diagram of the main hardware optimization choices made in the suggested architecture, such asisation of arithmetic to fixed-point arithmetic to reduce off-chip memory access, aggressive local on-chip buffering with block RAMs to selective parallelism in the features extraction stage and the neural network multiply-accumulate operations to enhance throughput and energy efficiency.

## AI Model Integration and Optimization

This subsection explains the approach taken on the way of integrating the AI model into the suggested VLSI architecture. Individual attention is placed on making a trade off between the accuracy of inference and the complexity of the hardware. To be able to run on resource-limited platforms, a lightweight neural network model is chosen. Training is done with quantization-aware training whereby smaller-precision arithmetic is implemented during training to enhance the resilience into hardware deployment. W and a are the weights and activations of the neural network respectively. During inference, fixed-point representations and are used, defined as

$$\hat{w} = round\ (w.2^n), \qquad \hat{a} = round(a.2^n) \quad (4)$$

n Where n is the fractional bit width.

### Hardware-Aware AI Design

Pipelined MAC units which are constituted to use a fixed-point number system are used to implement the AI inference engine. Activation functions are broken down to hardware-friendly functions including piecewise linear approximations or rectified linear units (ReLU), which has the form of:

$$ReLU(x) = \max(0,x) \quad (5)$$

This option is much simpler in terms of computational complexity, uniformity of control logic overheads than non-linear activations implementati0ons. The proposed AI inference architecture, as represented in Figure 3, uses fixed point pipelined MACs and simplified activation functions, which can be effectively realised in hardware. Moreover, it uses layer-level pipelining to enable the occurrence of overlapping computation across consecutive neural network layers to enhance the inference throughput without increasing the latency.

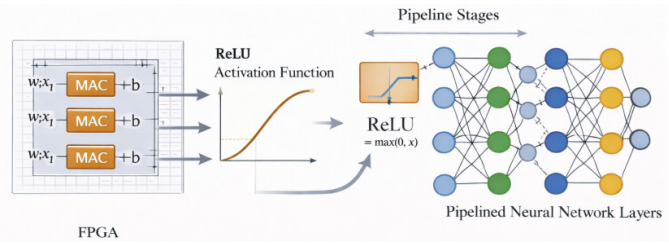


**Fig.3: Hardware-Aware AI Inference Architecture Using Pipelined Fixed-Point MAC Units**

The hardware-aware AI inference design represented schematically as a schematically and optimally designed

fixed-point pipelined multiplyaccumulate (MAC) block, simplified ReLU activation functions, and the pipelining of the neural network at the layer-level, to allow implementation of large neural networks with a high throughput and resource constraints.

### Pipeline Integration

The AI inference engine is closely combined with a signal processing pipeline with a streaming interface. The feature vectors created during the feature extraction stage are directly inputted into the AI engine without storing the attributes in some external memory. This fully integrated design reduces the data flow, minimizes memory usage and end-to-end latency is also significantly low. Thirdly, the proposed streaming VLSI pipeline uses a unified signal processing and AI inference hardware dataflow as shown in Figure 4, removing unwanted memory reads and data writes. Through this integrated pipeline strategy, the suggested methodology has better hardware efficiency and real-time performance to match resource-constrained embedded systems.

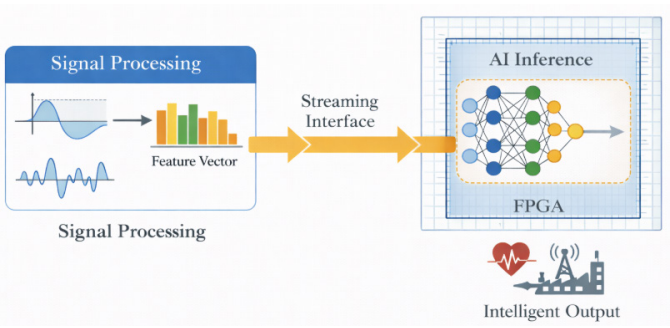


**Fig. 4: Tight Integration of Signal Processing and AI Inference Using a Streaming VLSI Pipeline**

An example schematic diagram of the proposed tightly coupled pipeline integration, in which feature vectors at the output of the signal processing casing are directly fed directly to the on-chip AI inference engine without external memory access, allowing even smaller data movement, lower latency, and enhanced hardware efficiency to real-time embedded applications.

## Experimental Setup and Evaluation Methodology

The section outlines the experimental design and testing procedure to be used in order to verify the efficacy and hardware-efficacy of the offered AI-based signal processing architecture. The experimental evaluation will aim at evaluating the performance of the proposed design in realistic conditions of hardware and to offer a reasonable comparison to realistic baseline architectures. It is a suggested architecture which is deployed on an FPGA platform with a commercial synthesis and implementation toolchain. The vendor-provided tools which are used to carry out all the hardware designs are synthesized, placed, and routed so that timing, power, and resource utilization estimates are correctly estimated. Post-place-and-route reports are used to perform performance evaluation, and they are able to give realistic figures that represent routing delay, logic usage and clock constraints. Evaluation of the design is done at a fixed operating frequency time to help make the comparison consistent across all of the compared architectures.

### Baseline Architectures

To illustrate the advantages of the suggested methodology, two base architectures are introduced and tested in the same conditions (in terms of hardware and operation). The former baseline is associated with a traditional signal processing pipeline that involves only deterministic signal processing algorithms with no AI improvements. This benchmark reflects the conventional embedded signal processing systems that are frequently applied in resource constrained platforms. The second baseline is a non-optimized AI-assisted signal processing pipeline, that is, an AI inference module is added but a hardware-aware application of fixed-point quantization, pipeline-level integration, or selective parallelism is not applied. Signal processing and AI inference are regarded as loosely coupled in this design, which increases the amount of data movement and overhead of memory. The contribution of the AI integration and hardware-aware optimization can be considered separately by comparing the proposed architecture with the given two baselines.

### Evaluation Metrics

All architectures are measured by four important measures that include throughput, latency, power consumption, and resource utilization. The number of input samples per unit time calculated as the attained operating frequency and pipeline start-up interval is called throughput. Latency is gauged by the total number of clock cycles which the pipeline input takes to reach the ultimate output. Considering post implementation power analysis tools, the power consumption is estimated considering the both static and dynamic power elements under characteristic switching activity. The usage of resources is measured by logic elements, look-up tables, registers, and block of on-chip memory utilized by every architecture. All these measures give a holistic evaluation of computational efficiency, energy efficiency, and the cost of hardware.

Assessing the suggested design and base architectures with the same datasets, operating rates, and measurement guidelines makes it possible to achieve a fair and significant

comparison. The given methodology of the evaluation allows capturing a clear picture of the benefits obtained by the given hardware-efficient VLSI design and AI-aware integration of the pipeline.

## Results and Discussion

This part will provide the experimental findings of the FPGA-based implementation of the proposed AI-enhanced signal processing architecture and propose its performance as compared to the baseline designs mentioned in Section 6. The analysis of throughput, latency, power use and utilization of hardware resources are performed to evaluate computational efficiency and hardware suitability to the resource-constrained platforms holistically.

### Throughput and Latency Analysis

Table 1 gives the throughput/latency of the proposed architecture as well as the baseline implementations. As can be seen, the pipelined VLSI architecture proposed demonstrates significantly better throughput than the traditional signal processing pipeline and the non-optimized AI-enabled pipeline. This has been made possible mainly due to the fundamental nature of the improved pipelined dataflow and removal of pockets of idle cycles among processing phases. Latency analysis also shows that the proposed pipeline design is effective. Although, as compared with the baseline architectures, the end-to-end latency is higher in the baseline architecture because of the sequential execution of the two and the buffering of the interimates, the proposed architecture greatly lowers the latency, as will be seen, by allowing the parallel execution of the signal processing and AI inference steps. The decrease in the latency validates that the suggested architecture is perfect towards real-time embedded systems that have strict timing requirements.

### Power and Energy Efficiency Evaluation

The Table 2 shows the power consumption results, and the energy efficiency comparison is shown in Figure 5. The proposed architecture has maintained lower power consumption as compared to the non-optimized AI-assisted baseline even though it operates at a higher throughput. This performance has mainly been enhanced by use of fixed-point arithmetic, minimized accessibility of memory and maximization of dataflow. The energy efficiency, which is energy consumed per processed sample, is significantly improved over both base designs. The findings reveal that quantization aware AI systems and hardware aware optimization substantially decrease both switching activity and memory access overhead which are major sources of power dissipation in embedded VLSI systems. The results are also in line with the recent reports of hardware-efficient AI accelerators that focus on the relevance of fixed-point computation and data locality to minimize energy consumption.

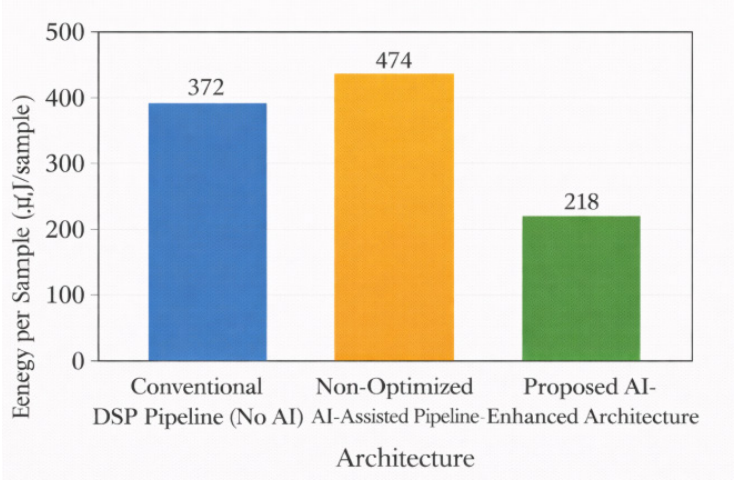


**Fig. 5: Energy Efficiency Comparison Across Architectures**

**Table 1: Throughput and Latency Comparison**

| Architecture | Throughput (Samples/s) | Latency (Clock Cycles) | Operating Frequency (MHz) |
|---|---|---|---|
| Conventional DSP Pipeline (No AI) | $0.82 \times 10^6$ | 420 | 100 |
| Non-Optimized AI-Assisted Pipeline | $0.95 \times 10^6$ | 510 | 100 |
| Proposed AI-Enhanced Pipelined Architecture | $1.65 \times 10^6$ | 260 | 100 |

**Table 2: Power Consumption and Energy Efficiency**

| Architecture | Dynamic Power (mW) | Static Power (mW) | Total Power (mW) | Energy per Sample (nJ/sample) |
|---|---|---|---|---|
| Conventional DSP Pipeline (No AI) | 210 | 95 | 305 | 372 |
| Non-Optimized AI-Assisted Pipeline | 340 | 110 | 450 | 474 |
| **Proposed AI-Enhanced Architecture** | **260** | **100** | **360** | **218** |

Comparative energy consumption per processed sample of the conventional DSP pipeline, the non-optimized AI-assisted pipeline and the proposed AI-enhanced architecture showing that the degree of energy efficiency improvement through the use of hardware-aware optimization and pipelined integration is substantive.

### Resource Utilization Analysis

Table 3 gives a close comparison between the utilization of hardware resources, such as logic elements, registers and on-chip memory blocks. The proposed architecture will show the balanced use of the logic and memory resources, which are able to perform higher with the required hardware overhead. The system design with optimized bit-width choice and reduced control logic needs a lower number of logic resources when compared to the non-optimized AI-assisted baseline. The effective utilization of a on-chip memory also adds to the decrease in access to external memory enabling better power efficiency and scalability. These findings verify that the architecture suggested is relevant in resolving the trade-off between level of performance and hardware price, which is one of the main concerns on resource-constrained embedded systems.

### Comparison with Existing Studies

The trends followed by the performance can be explained by the previous studies of pipelined VLSI architecture and AI acceleration on hardware. Like other past analyses, pipeline computation and fixed-point presentation lead to large improvements of throughput and power saving. Nonetheless, at the same time that most of the current literature concerns independent AI accelerators, the presented architecture serves as evidence of the benefits of closely combining AI inference into an overall signal processing pipeline. This consolidation minimizes the data flow and latency that results in the best system efficiency.

In general, the experimental findings confirm the usefulness of the suggested hardware-efficient VLSI design methodology and prove its appropriateness in the case of real-time AI-enhanced processing of signals with the help of constrained resources.

## Comparison with State-of-the-Art

The most common recent AI-enabled signal processing accelerators use a loosely coupled implementation of the signal preprocessing and AI inference as separate modules that can be found in many implementations. Though, such designs can have high computational throughput, they have high data movement overheads, memory access overheads, and energy consumption because of the repetition of transfers across processing stages. On the contrary, the proposed architecture embraces a single VLSI pipeline that closely combines both signal processing and AI inference with the help of a streaming dataflow. Such integration removes intermediate external memory access and does not need redundant data move in between and leads to less latency and increases energy efficiency. Also, although there are solutions that are based on a floating-point or partial fixed-point implementation, in the proposed design, an all-hardware-conscious optimization strategy in terms of fixed-point arithmetic, quantization-conscious AI integration, and selective parallelism is utilized. The proposed approach to managing intelligence processing sequences (data acquisition to intelligent decision generation) will be unlike earlier works that optimize AI inferences on an isolated basis. Such end-to-end efficiency and scalability facilitate high-quality system-wide efficiency and scalability and, therefore, make the proposed architecture highly appropriate to the case of real-time AI-enhanced signal processing on resource-constrained embedded systems.

## Conclusion and Future Work

This paper has introduced a hardware efficient VLSI architecture design methodology of AI-enhanced signal processing pipelines on resource constrained embedded systems. Through an algorithm-hardware co-design methodology, the given architecture brings signal preprocessing, feature extraction and AI-inspired inference with one another, deeply pipelined, dataflow. Fixed-point arithmetic, quantization-sensitive AI architecture, selective parallelism, and streaming pipeline integration allow significant improvements in throughput, latency and energy consumption relative to the traditional and not optimized AI-assisted baseline architectures. Implementation and the post-place-and-route evaluation Report FPGA Implementation and post-place-and-route evaluation indicated that the proposed design provides a good balance between performance, power-consumption, and hardware resource utilization, and can be used in real-time edge and embedded applications. The next round of

Table 3: Hardware Resource Utilization

| Architecture | LUTs | Flip-Flops | DSP Slices | BRAM Blocks |
|---|---|---|---|---|
| Conventional DSP Pipeline (No AI) | 7,420 | 5,180 | 24 | 18 |
| Non-Optimized AI-Assisted Pipeline | 12,360 | 8,940 | 56 | 42 |
| Proposed AI-Enhanced Architecture | 9,850 | 7,120 | 38 | 26 |

work will be done on extending the proposed architecture to ASIC implementation to bring about enhancement into energy efficiency and area optimization. Future research directions are dynamic and partial reconfiguration to facilitate multi-application workloads and incorporation of more complex AI models at the same time maintaining hardware efficiency. The idea to investigate the adaptive precision and runtime optimization methods to further promote the scalability and robustness in the presence of diverse operating conditions is also a prospective line of future research.

## References


1. Al Tareq, A., Rahman, M. M., Hossain, M. A., & Islam, M. S. (2024). Impact of IoT and embedded systems on the semiconductor industry: A case study. *Control Systems and Optimization Letters, 2*(2), 211–216. https://doi.org/10.59247/csol.v2i2.111
2. Chen, L., Zhang, Y., Li, X., & Wang, H. (2024). AI-driven sensing technology: A review. *Sensors, 24*(10), Article 2958. https://doi.org/10.3390/s24102958
3. De Marinis, L., et al. (2024). Photonic technologies for analog neuromorphic computing. In *2024 IEEE Photonics Society Summer Topicals Meeting Series (SUM)*. IEEE. https://doi.org/10.1109/SUM60964.2024.10614512
4. Fabre, W., et al. (2024). From near-sensor to in-sensor: A state-of-the-art review of embedded AI vision systems. *Sensors, 24*(16), Article 5446. https://doi.org/10.3390/s24165446
5. Goel, P. (2024). *AI for energy efficiency and conservation* (pp. 32–49). IGI Global. https://doi.org/10.4018/979-8-3693-6567-0.ch003
6. Ieong, M. (2018). Semiconductor industry driven by applications: Artificial intelligence and Internet of Things. In *2018 IEEE International Conference on Electron Devices and Solid State Circuits (EDSSC)*. IEEE. https://doi.org/10.1109/EDSSC.2018.8487118
7. Jothi, C. S., et al. (2024). Revolutionizing healthcare through robotics and AI integration: A comprehensive approach. In *Exploring the Micro World of Robotics Through Insect Robots* (pp. 213–234). IGI Global. https://doi.org/10.4018/979-8-3693-6150-4.ch011
8. Kozłowski, M., Racewicz, S., & Wierzbicki, S. (2024). Image analysis in autonomous vehicles: A review of the latest AI solutions and their comparison. *Applied Sciences, 14*(18), Article 8150. https://doi.org/10.3390/app14188150
9. Nguyen, V.-T., et al. (2024). Development of a wearable device for heart rate monitoring and fall detection using machine learning to analyze early cardiovascular anomalies. *Journal of Science and Technology – University of Danang*, 7–12. https://doi.org/10.31130/ud-jst.2024.158ICT
10. Rausell-Campo, J. R., et al. (2024). Programmable photonic extreme learning machines. *arXiv*. https://doi.org/10.48550/arXiv.2407.03218
11. Sen, S., & Datta, A. (2024). Human-inspired distributed wearable AI. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*. https://doi.org/10.48550/arXiv.2406.18791
12. Thakur, A., & Mishra, S. K. (2024). An in-depth evaluation of deep learning-enabled adaptive approaches for obstacle detection using sensor-fused data in autonomous vehicles. *Engineering Applications of Artificial Intelligence, 133*, Article 108550. https://doi.org/10.1016/j.engappai.2024.108550
13. Vegesna, D. V. (2024). AI-powered mental health monitoring using wearable devices and mobile sensing. *International Research Journal of Computer Science, 11*, 545–550. https://doi.org/10.26562/irjcs.2024.v1108.01
14. Zhou, J., et al. (2024). High-speed energy-efficient optics for AI/ML applications. In *2024 IEEE Photonics Society Summer Topicals Meeting Series (SUM)*. IEEE. https://doi.org/10.1109/SUM60964.2024.10614550