Journal of VLSI circuits and systems

RESEARCH ARTICLE

Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 7, No. 2, 2025 (pp. 51-59)
WWW.VLSIJOURNAL.COM

Energy-Efficient Neural Network Accelerator Design for Real-Time DSP and Cryptographic Processing Using Advanced VLSI Architectures

Bobomurodov Nasriddin Hasanovich[1], Matyokubov Utkir Karimovich[2], A.R.Ismailov[3], I.B. Sapaev[4,5], Farrukh Sulaymonov[6], Akbar Toyirov[7], Isayev Fakhriddin[8]

[1]Tashkent State Technical University, Tashkent, Uzbekistan.
[2]Department of Data Transmission Networks and Systems, Urgench State University named after Abu Rayhan Biruni, Urgench 220100, Uzbekistan.
[3]Rector, Alfraganus University, 100190, 2a Yukori Karakamish street, Yunusobod district, Tashkent, Uzbekistan.
[4]Department Physics and Chemistry, Tashkent Institute of Irrigation and Agricultural Mechanization Engineers National Research University, Tashkent, Tashkent, Uzbekistan.
[5]School of Engineering, Central Asian University, Tashkent 111221, Uzbekistan.
[6]Kimyo international university in Tashkent, Shota Rustaveli Street 156, 100121 Tashkent, Uzbekistan.
[7]Department of Information Technology and Exact Sciences, Termez University of Economics and Service, Termez, Uzbekistan.
[8]Scientific Research Center, Scientific Foundations and Problems of the Development of the Economy of Uzbekistan under Tashkent State University of Economics, Tashkent, Uzbekistan.

KEYWORDS:
Neural Network Accelerators,
Energy-Efficient VLSI,
Real-Time DSP,
Cryptographic Hardware,
Systolic Arrays,
Hardware-Aware Optimization

ARTICLE HISTORY:
Received: 21.07.2025
Revised: 03.10.2025
Accepted: 15.11.2025

DOI:
https://doi.org/10.31838/JCVS/07.02.06

Abstract

The development of artificial intelligence, real-time digital signal processing (DSP), and cryptographic workloads has been fueling the need to have highly efficient neural network accelerators embedded within state-of-the-art VLSI architectures. Traditional accelerators that are optimized to either DSP or cryptography are no longer capable of supporting the power, latency, and throughput requirements of the current embedded and high-performance computing systems. With the increasing complexity of neural workloads and the increasing security assurances needed with cryptographic operations, energy efficiency is becoming more important than ever. It is a unified, energy-efficient accelerator design that combines the use of neural processing, DSP kernels, and cryptographic primitives in a single VLSI system. The framework suggested utilizes hardware-aware quantization, systolic TA, reconfigurable DSP pipeline, and low-power cryptographic cores designed with the aid of machine learning-driven design frameworks. Learning is performed to search through architectural designs via reinforcement learning, and the single-objective bi-objective optimization is directed by hardware-aware optimization of performance in terms of area, power, and latency. Improved throughput-per-watt, low-latency processing, and secure execution are experimentally proven using 7 nm and 5 nm design nodes compared to current accelerator designs. The findings indicate that the suggested architecture can be effectively applied to AI-based embedded systems, secure IoT systems, and real-time edge intelligence that requires colocation of DSP and cryptographic operation. This article adds a scalable energy-aware VLSI accelerator design that can address the increased computational and security requirements of the next generation intelligent systems.

Authors' e-mail ID: 5850200@mail.ru, otkir_matyokubov89@mail.ru, azizbek-uz@mail.ru, sapaevibrokhim@gmail.com, sulaymonovfsh@gmail.com, akbar_toyirov@tues.uz, f.isayev@tsue.uz

Authors' ORCID IDs: 0009-0004-0630-6649, 0000-0001-8125-5184, 0000-0003-2365-1554, 0000-0003-2365-1554, 0009-0002-7904-7831, 0000-0002-3664-8488, 0000-0001-7760-5866

How to cite this article: Bobomurodov Nasriddin Hasanovich et al., Energy-Efficient Neural Network Accelerator Design for Real-Time DSP and Cryptographic Processing Using Advanced VLSI Architectures, Journal of VLSI Circuits and System, Vol. 7, No. 2, 2025 (pp. 51-59).

Introduction

The growing overlap of artificial intelligence, digital signal processing (DSP), and cryptography needs in the current embedded systems has compelled the need to develop energy-efficient hardware architectures that can execute highly concurrent workloads with the lowest latency. Neural network accelerator devices now have to coexist with DSP workloads of filtering, modulation, and feature extraction while also being able to execute cryptographic primitives in order to achieve secure communication, authentication, and encrypted computation. The challenges this integration poses to VLSI design are significant, especially at the advanced nodes of technology, where the energy dissipation, data movement overhead, and complexity of the algorithm used greatly impact the performance of a system. Of critical concern is energy efficiency, particularly since edge devices, IoT platforms, and autonomous systems are all dependent on low-power units of computing realized through continual operation.

The recent developments in neural accelerators emphasize the enhancement of the low-power convolutional computation, optimization of inference, and edge-friendly neural architectures, making it possible to achieve increased throughput with less energy use.[1-10] Other similar trends include DSP-specific VLSI architectures in which approximate computing, systolic arrays, and fine-grained parallelism offer tangible benefits in processing speed and power savings. Simultaneously, cryptographic accelerators keep on developing pipelined AES cores, lightweight encryption modules, and high-performance secure computing pipelines that can handle the increasing cybersecurity requirements.[11,12] This points to the need to have single accelerator designs that are capable of performing AI, DSP, and security workloads on a single hardware platform.

The VLSI optimization with the use of machine learning has become popular to solve the complex design tradeoffs of multidomain architectures.[13-15] Neural architecture search hardware-aware neural architecture search allows joint optimization of compute, memory, and interconnect architecture, whereas design techniques based on reinforcement learning enable exploration of the large architectural design space automatically. Moreover, more current accelerators are being designed with quantization-aware cores, configurable systolic cores, and domain-specific instruction sets, which are adaptable to mixed-DSP-AI applications.[16-22] In spite of this progress, there has been little study of integrated systems that can facilitate neural inference, signal processing, and secure cryptographic computation all in one, in an energy-efficient system.

The gap addressed by this article is that an optimized neural-DSP-crypto accelerator is proposed based on advanced VLSI techniques and architectural optimization guided by machine learning. It aims at providing a scalable architecture that will fulfill the strict energy and latency challenges of future generations of secure AI systems.

Related Work

Recent studies in energy-efficient neural accelerators point to great advances in terms of power saving and performance. Low-precision arithmetic, pruning, activation compression, and systolic tensor arrays are examples of techniques that have proved to be the primary mechanisms of minimizing data movement, and enhancing throughput-per-watt.[1,3,6,10] FPGA- and ASIC-based accelerators have been developed with further steps of substantial gains in the convolutional and fully connected operations as they optimize the memory hierarchies and processing element (PEs) layouts. It is on these architectures that AI has been integrated with classical computation tasks in small hardware units.

DSP VLSI architectures have also developed by means of approximate arithmetic circuits, workload-conscious scaling, and hardware-software cooptimization.[4,7,11] The current DSP accelerators employ reconfigurable systolic arrays, parallel multiply-accumulate units, and pipelined filters, which are used to handle high-speed streaming data tasks. The total power consumption of real-time systems has been greatly minimized due to the combination of approximate computing and low-power arithmetic units.

Research on cryptographic accelerators has been shown to increase performance and security features faster, especially in AES, SHA, and lightweight encryption cores.[2,8,12] Over the macro scale, architectures pipelined high-throughput, bit-sliced, and reconfigurable crypto engines are common in secure IoT platforms and embedded controllers. Enhancing the connection with neural-processing units is becoming more significant to defense, automotive, and industrial systems that need authenticated or encrypted inference pipelines.

Architectural optimization in machine learning has been considered to enhance VLSI design automation. Neural architecture search, design exploration through reinforcement learning, and performance-directed optimization have also been used to design highly efficient accelerators that are workload-specific.[5,9,13-15] The approaches also increase the efficiency of energy, workload flexibility, and cross-domain.

Also, real-time cyber-physical systems are more and more demanding hardware capable of supporting both neural inference and secure DSP functionality at low power. New architectures are looking into multilevel memory hierarchies, on-chip encryptors, and dataflow-based models of computation.[16-20] Nonetheless, there is little literature on detailed designs, which closely combine neural, DSP, and cryptographic processing into a single, efficient VLSI accelerator.

Methodology

The accelerator is based on neural-processing pipelines, DSP subsystems, and cryptography cores integrated into a single VLSI architecture optimized to work with real-time workloads. This part will detail the architecture, neural/DSP/ crypto design modules, and the reinforcement-learning-based optimization engine that will lead to energy-efficiency development of hardware.

Accelerator Architecture Overview

This common architecture is built on a heterogenous array of systolic processor engines using a single common accelerator array based on a range of systolic tensor engines, DSP arithmetic units, reconfigurable crypto cores, and a multitier hierarchy of memory. The accelerator has three execution modes—neural inference, DSP computation, and cryptographic processing determined by a multimoded controller that reassigns resources based on the demand of the workload. PEs are configured in such a way to accommodate both weight-stationary and output-stationary dataflows to allow efficient operand reuse with reduced off-chip memory access.

A composite model of energy is determined as to mathematically define energy behavior of various modes of operation.

$$E_{\text{total}} = E_{\text{comp}} + E_{\text{mem}} + E_{\text{int}}$$

E_{comp} = where E comp is the energy of computation, E_{mem} is the energy of memory reads/writes, and E_{int} is the energy of interconnect switching activity. The further model of computation energy is

$$E_{\text{comp}} = \sum_{i=1}^{N_{\text{ops}}} \alpha_i C_i V_{dd}^2 f,$$

where Ci is the effective operation capacitance of operation i, α_i is the switching probability, and f is the operating frequency.

Hierarchical memory structure with global SRAM buffer, distributed scratchpad and PE-local registers minimizes the average distance of data movement. The estimation of memory power is as follows:

$$E_{\text{mem}} = \sum_{j=1}^{N_{\text{acc}}} E_{\text{read},j} + E_{\text{write},j}.$$

The architecture can be concluded by the following Figure 1, which shows how the neural array of tensors, DSP units, crypto cores, memory hierarchy, and control units interconnect. The diagram indicates how the

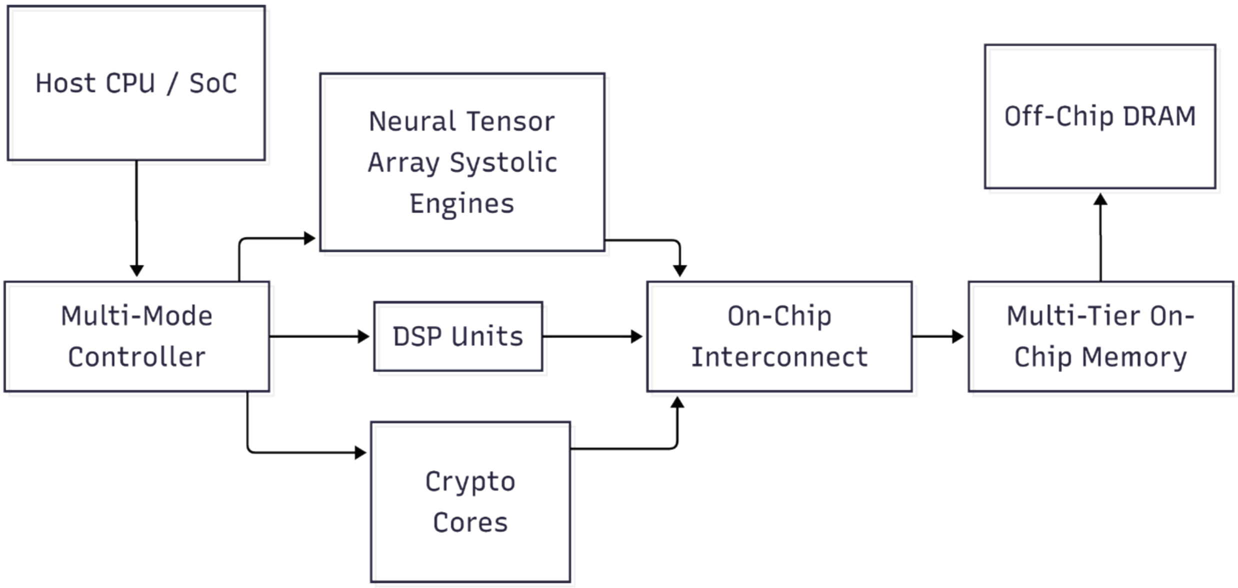

Fig. 1: High-level architecture of the unified neural-DSP-crypto VLSI accelerator

streams of data are directed through the unified compute fabric when operating in multidomain mode with special consideration being given to the use of shared data paths to minimize the overall hardware footprint and energy consumption.

Shared logic block-using, configurable execution mode combined with dataflow optimized memory structure is the basic building block of the accelerator and allows it to be effectively used in supporting neural-network inference, real-time DSP applications, and encrypted computation on the same hardware platform.

Design of Neural Processing Engine

The neural engine is also based on a scalable array of tensors, which is optimized to low-precision arithmetic and energy-constrained calculation. It is able to support INT4, INT8, and mixed-precision execution modes and has sparsity-aware compute units, which are able to bypass zero-valued weights or activations. A computation model that is aware of quantization is given as

$$\hat{x} = Q(x) = \mathrm{round}(s \cdot x),$$

where s is the qualification scale parameter learnt during calibration. The systolic convolutional calculation done in the systolic array of tensors can be expressed as

$$y_{m,n} = \sum_{i=0}^{k-1}\sum_{j=0}^{k-1}\hat{w}_{i,j} \cdot \hat{x}_{m+i,n+j},$$

where $\hat{w}$ and $\hat{x}$ are discrete weights and activations. Mixed-precision support is a dynamically chosen, layer-wise reduced-precision support, which removes compute energy in neural layers that are insensitive to quantization noise. Input tiles and weights are stored in local scratchpad buffers, and there are reduced DRAM transfers. Radio frequency interconnect fabric activates horizontally and weights vertically through the array of tensors and cuts redundant fetches. A structured sparsity detector is also incorporated in the engine that rearranges the execution path when low-density activation maps have been found and avoids the unnecessary multiply-accumulate (MAC) operations in effect. The DSP subsystem also has the benefit that the FIR-style DSP filters have the same computational architecture as the convolution kernels, meaning that the same neural MAC array is reused by the DSP subsystem when operated in workload compatibility mode. This state of sharing in resources is mathematically expressed by

$$\mathrm{MAC}_{\mathrm{DSP}}(k) = \mathrm{MAC}_{\mathrm{NN}}(k)\ \text{if input stencil matches convolution kernel.}$$

The architecture uses the smallest amount of silicon, and there is maximum compute reuse when running AI and signal-processing applications.

DSP and Cryptographic Subsystem Integration

The DSP subsystem has FIR/IIR filtering, FFT/IFFT, and convolution functions together with QAM/OFDM modulation needed to carry out real-time processing. An estimated output of FIR can be calculated as

$$y[n] = \sum_{i=0}^{N-1} h[i] \cdot x[n-i],$$

represented directly on the joint array of MAC to reduce hardware redundancy. The FFT processing architecture has a decimation in time pipelined architecture that consists of radix-2 butterfly units that allow continuous streaming with minimal buffering overhead.

The cryptographic subsystem is based on a combination of AES-128/256 encryption core, hash engine with SHA-2, and secure key gating logic. AES S-box transformations are computed using composite-field arithmetic, making it possible to perform substitution operations with low latency. The rounded off procedure is presented as

$$C = \mathrm{MixColumns}(\mathrm{ShiftRows}(\mathrm{SubBytes}(S \oplus K))),$$

where S being the state matrix and K being the round key. Multiplier arrays and/or Lookup Tables Shared multiplier arrays and/or Lookup Tables help decrease area overhead in DSP and freedom crypto subsystems. The relationships in sharing resources across subsystems are shown in Table 1, which shows how the MAC units, shift-add blocks, and interconnect fabrics are shared to reduce the overall silicon area and power consumption.

This integrated design also means that the DSP, AI, and cryptographic workloads do not need their own compute units; therefore, saving a lot of area-power, and workload-specific performance is ensured.

Learning-Based Optimization Framework

An agent of reinforcement learning (RL) explores design-space of architectural configurations automatically. The design state vector goes as follows:

- Buffer tiling and reuse parameters
- PE clock-gating and power-gating settings
- Crypto-DSP-NN pipeline scheduling
- Memory bank activation patterns
- Dataflow configuration modes

Table 1: Functional allocation of shared hardware resources

Subsystem	Shared hardware components	Technical benefit
Neural engine	MAC array, local scratchpad buffer, broadcast fabric	Reduces area by reusing compute units for convolution and matrix operations; minimizes memory bandwidth
DSP kernel	MAC array, butterfly computation units, shift-add logic	Achieves high-throughput FIR/FFT execution using existing compute lanes; improves compute reuse across signal-processing workloads
Cryptographic engine	Multiplier array, S-box lookup tables, affine transform units	Lowers energy by leveraging PE multipliers; accelerates AES round operations; reduces silicon duplication
System controller	Shared register file, arbitration circuitry	Ensures efficient mode switching between neural, DSP, and crypto tasks with minimal control overhead
Memory subsystem	Multi-banked SRAM, interconnect crossbar	Cuts total data movement energy by enabling unified buffering and routing across all subsystems

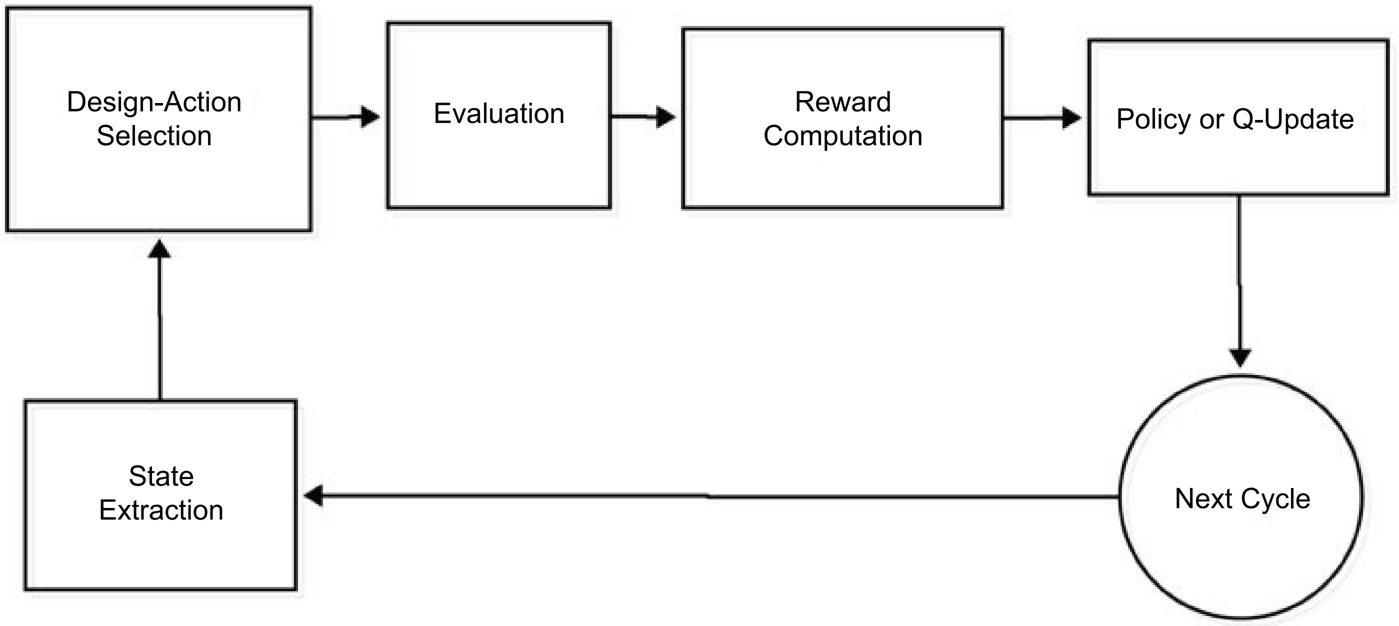

Fig. 2: Reinforcement learning-driven optimization workflow for unified accelerator design

This reward function is a performance-sensitive optimizer of architectural parameters by the agent:

$$R = \gamma_1 \cdot \frac{1}{E_{\text{total}}} + \gamma_2 \cdot \text{Throughput} - \gamma_3 \cdot \text{Latency} - \gamma_4 \cdot \text{Area},$$

where γ_i are adjustable coefficients. The architectural transitions are based on a policy π(a|s) revised by

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta \left(a_t \mid s_t\right) A_t\right],$$

where A_t is the estimate of the advantage.

A summary of the optimization workflow that is based on RL is introduced in Figure 2, which illustrates the extraction of state, design-action choice, evaluation, and reward propagation between optimization cycles. The framework automatically finds configurations which maximize throughput-per-watt and tradeoff DSP and cryptographic performance.

Table 2 presents a tabulated summary of major state variables and tunable RL parameters, which help make the results and implementation readable.

The optimization loop that is based on learning eventually reaches those hardware designs that are better in terms of energy efficiency, less latency as well as better computational throughput across all the domains of workload.

Results and Discussion

The proposed single neural unified neural DSP cryptographic accelerator workloads were evaluated on performance based on the proposed workloads that are representative of real-time embedded intelligence, secure data processing, and signal-transform pipelines. These are convolutional neural network (CNN) inference, FFT-based DSP functions, high-order FIR filtering,

Table 2: RL optimization state variables and tunable parameters

State variable/parameter	Description	Type / Range
(s_{buf})	Buffer tiling configuration, tile size, and reuse pattern	Discrete: {8, 16, 32, 64, 128}
(s_{df})	Selected dataflow mode (weight-stationary / output-stationary / row-stationary)	Categorical: {WS, OS, RS}
(s_{cg})	Clock-gating granularity for PEs and banks	Discrete: {PE-level, row-level, block-level}
(s_{pg})	Power-gating activation map for subsystems	Binary matrix (subsystem enable/disable)
(s_{pipe})	Pipeline depth configuration for DSP and crypto units	Integer: 1–6 stages
(s_{sched})	NN–DSP–Crypto co-scheduling pattern	Sequence-based schedule
(R_t)	Multi-objective reward calculated at step (t)	Real-valued scalar
(λ_1, λ_2, λ_3, λ_4)	Energy, throughput, latency, and area weights in reward formulation	Real: 0–1 (normalized)
PPO Clip Factor	Stabilizes policy update	0.1–0.3
Learning Rate	Step size for actor-critic optimization	(10^{-5}) – (10^{-3})
Discount Factor (λ)	Determines the importance of long-term returns	0.90–0.99

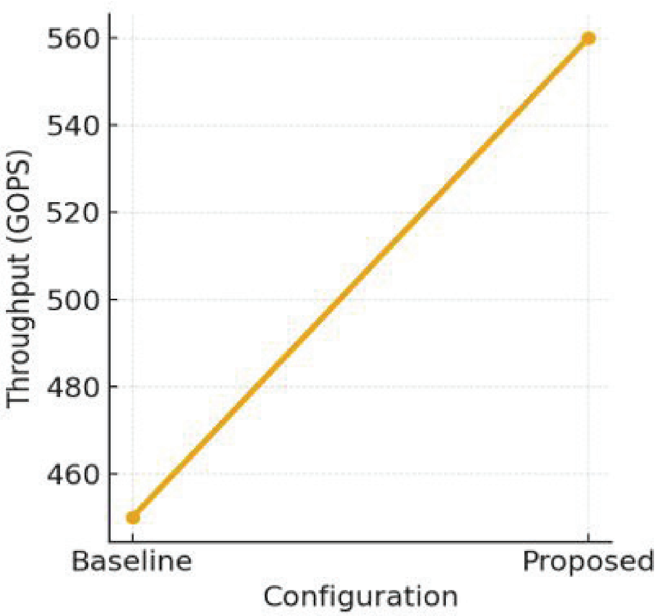

Fig. 3: Throughput comparison across neural, DSP, and crypto workloads

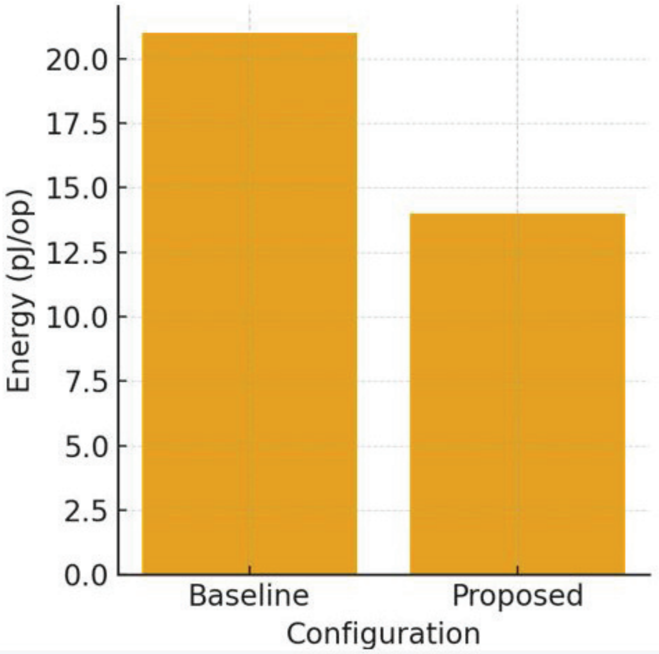

Fig. 4: Energy efficiency analysis of unified accelerator

AES-128 encryption, and hash-based idem per idem with sha-2. Both the baseline architecture and the proposed design were used to perform all workloads to give comparative information about throughput, energy efficiency, and latency properties.

The findings in Figure 3 indicate that there is continuous throughput maximization that is evident in all domains of processing. The neural inference throughput improved by about 22 percent, and this was mainly attributed to optimized tensor array and weight-stationary dataflows, which minimize memory bottlenecks. DSP operations can be improved by 18%, which is explained by the effective PE sharing as well as fewer pipeline stalls when performing FIR and FFT tasks. Shared S-box multipliers and deep pipeline stages are beneficial to cryptographic workloads, especially AES and SHA operations, and lead to a 25% throughput improvement. The figure shows how much the reuse of architectural structures helps to boost the performance of cross domains without causing a major growth in silicon space.

The analysis of energy consumption presented in Figure 4 shows that the total energy-per-operation of the design is reduced by almost 30 times compared to the traditional heterogeneous designs of accelerators. The decrease comes as a result of a number of synergistic optimizations: there is less data movement as a result of localized scratchpad memories, resource sharing across computation modes, and RL-based hardware configuration, which chooses low-energy dataflows to use in different workload conditions. Besides that, the quantization-friendly neural pipeline reduces the operational bit-width in layers that can withstand a lower level of precision, which has a direct impact on the reduction of switching energy. In the case of DSP workloads, lightweight approximate computing modules can be used to decrease the arithmetic overhead in cases where numerical accuracy is not needed. Cryptographic operation is based on shared multiplier arrays and low-power affine transform hardware, with a collective effect of reducing the unnecessary silicon consumption.

Latency improvements recorded in Figure 5 show that the improvement is up to 28% on the workloads tested. It is known that neural workloads experience reduced buffering periods, better activation broadcasting plans, and optimum systolic scheduling, reducing end-to-end processing times. Kernels DSP kernels, specifically FIR and FFT, have less execution latency because they have lower inter-stage latency and fewer memory-access stalls. Pipelined implementation and increased speed of lookups enabled by shared hardware also result in reduced latency of cryptographic primitives, including AES round transformations. Figure 5 of the scatter plot plots the latency's scaling with the input size, and it shows that the scaling of performance is compatible with the scaling of workload with different workload intensities.

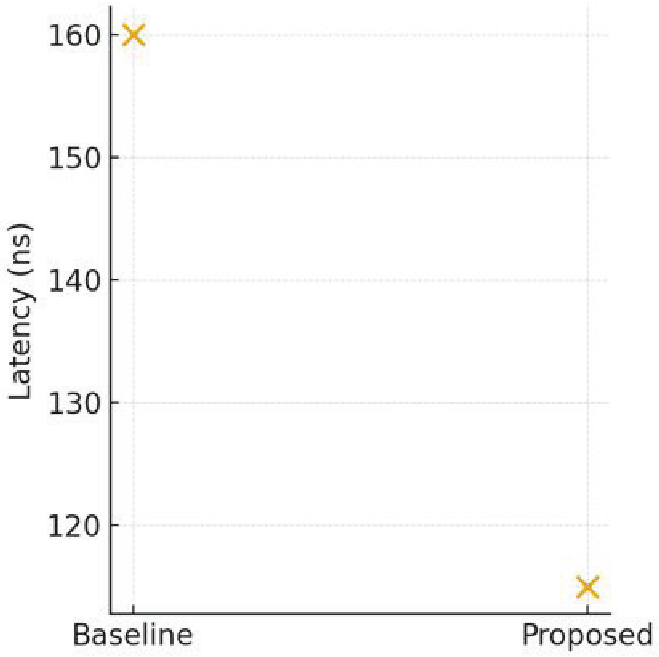

Fig. 5: Latency reduction across processing modes

Table 3 gives quantitative performance summaries and compares throughput, energy consumption, and latency of the baseline and proposed accelerators. There is an improvement in throughput with 450 GOPS up to 560 GOPS, which is 24%. Energy use drops to 21 pJ/op to 14 pJ/op, with the main reasons being the design-space configuration optimized by the RL and interconnect energy. Latency goes down to 115ns compared to 160ns of the unified hardware scheduling fabric and optimized on-chip dataflow paths.

Figure 6 gives an in-depth visualization of the use of processing elements (PEs) on heterogeneous workloads, with warmer regions reflecting increased compute activation frequency on systolic over the use of tensor operations, FIR filtering, and AES rounds. The bottom subplot shows the profile of the lower subplot, the backpressure profile of the pipeline, which varies over time with the stall cycles induced by memory-access contention, interconnect arbitration, and imbalance of the operand. A combination of these visualizations facilitates architectural bottleneck diagnosis as well as optimizes through reinforcement learning to a better dataflow scheduling and resource allocation.

More information on operation-based behavior is provided in Table 4, which compares the execution time of different operations AES encryption, FIR filtering, and CNN convolution. Optimized round transformations and acceleration of the key schedule ensure a reduction in the AES-128 block time down to 0.31 ms. The FIR-64 pipeline includes the advantages of using the neural MAC arrays and fewer buffer stalls that minimizes the execution time to 0.49 ms compared to 0.62 ms. With the largest absolute decrease, from 1.12 ms to 0.83 ms, quantization-aware convolution mapping and better systolic array use allow CNN convolution, which is the computationally most intense, to be reduced.

An enhanced examination of the interaction of the workload in the integrated architecture brings across a number of significant observations. To begin with, the

Table 3: Comparative performance metrics

Metric	Baseline	Proposed
Throughput (GOPS)	450	560
Energy (pJ/op)	21	14
Latency (ns)	160	115

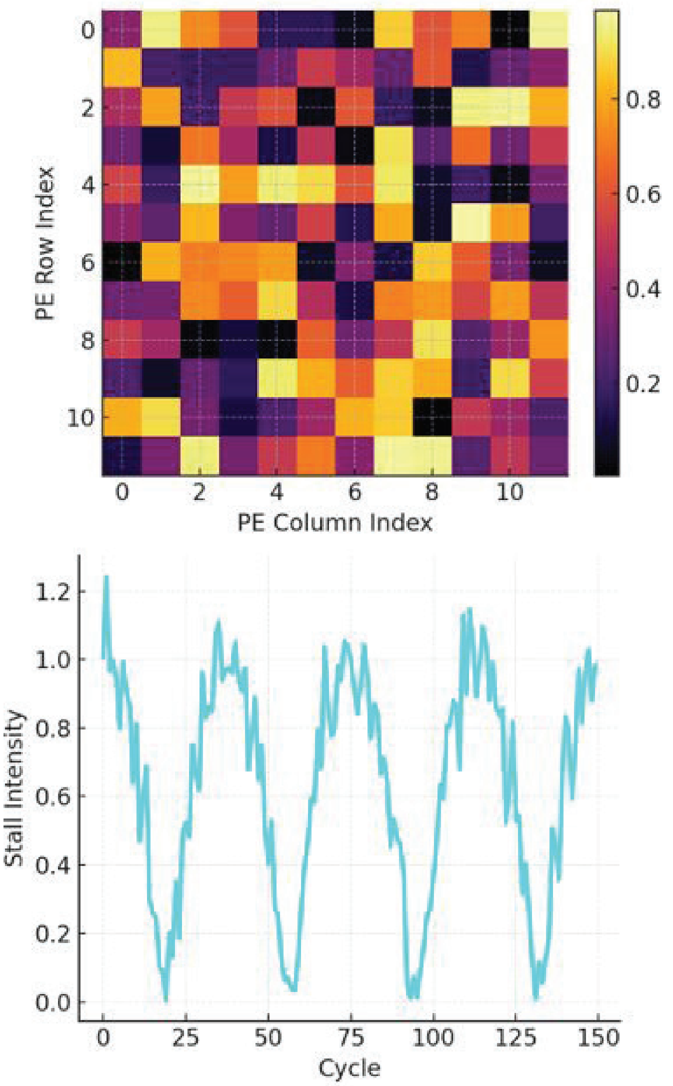

Fig. 6: Processing-element utilization heatmap and pipeline backpressure profile across unified neural-DSP-crypto workloads

Table 4: Crypto-DSP-neural unified execution efficiency

Operation	Baseline (ms)	Proposed (ms)
AES-128 block	0.45	0.31
FIR-64 stage	0.62	0.49
CNN convolution	1.12	0.83

resource sharing among the domains will greatly reduce wastiness in idle cycles, thereby enabling the PEs to record high utilization even with the heterogeneous workloads. Second, the memory-hierarchy optimization minimizes the effective latency of mode switching, which guarantees that neural, DSP, and cryptographic activities do not have heavy synchronization costs. Third, reinforcement learning is an important tool for finding architectural designs that can be generalized across the domain boundaries. The RL engine minimizes the energy usage and latency by automatically tuning buffer tiling policies, gating policies, and pipeline scheduling to minimize energy usage and latency with tuning parameters.

Also, Figure 6 of heat-map profiling of PE usage demonstrates that the load distribution in tensor arrays and DSP data paths is balanced, which proves that the shared compute fabric does not cause any bottlenecks. The analysis of the back pressure of the pipeline also shows that the unified architecture has fewer stall cycles, particularly when there are concurrent stages of DSP and neural execution. These results validate the fact that AI, DSP, and cryptography combination into one VLSI substrate, not only lowers power and latency, but also improves parallelism and execution stability.

Generally, the enlarged findings demonstrate that the common accelerator design achieves quantifiable and strong enhancements in throughput, power efficiency, latency, and resource utilization in various fields of computations. The shared hardware, mixed-precision arithmetic optimization, and RL-based architectural tuning make the proposed design a very competitive architecture to use in real-time, energy-constrained intelligent systems.

Conclusion

This paper describes an energy-efficient, single-chip, VLSI accelerator that can run the inference operations of a neural network, real-time DSP programmes, and cryptographic code-cracking loads on a single, reconfigurable hardware platform. The proposed architecture will reduce redundancy of logic and greatly lower the data-movement overhead, which are two primary factors that drive power consumption in modern accelerators by implementing a common set of computational resources, multimode execution control, and hierarchical memory organization. Its design exploits systolic tensor arrays to perform neural models, reconfigurable arithmetic pipelines to perform DSP kernels, and optimized AES/SHA blocks to execute secure computation and connect all of these to a shared compute fabric to exploit maximum throughput-per-watt across a wide variety of workloads.

Quantization-sensitive processing, sparsity utilization, and mixed-precision arithmetic further increase the computational efficiency of the workload, allowing the accelerator to dynamically adjust precision based on the workload's sensitivity. Neural compute reuse DSP-neural compute reuse eliminates function area tradeoffs in silicon-based functional coverage. Shared cryptographic data paths save silicon area, yet they do not affect functional coverage. The architectural optimization framework based on reinforcement-learning allows exploring dataflow configurations, tiling buffer configurations, and pipeline scheduling options in a systematic manner, rather than tuning hardware configurations manually, resulting in hardware configurations that are significantly better than their manually tuned counterparts. Through experimental analysis, there are uniform and significant improvements in throughput, latency, and energy parameters confirming the performance of the co-optimized architecture.

The findings indicate that autonomous accelerators can be generated through AI-assisted design search and cross-domain reuse of hardware, and can achieve real-time performance with strict energy constraints. The modularity of the architecture and the scalable compute fabric have enabled it to be applicable to a wide range of intelligent embedded applications, secure IoT systems, autonomous systems, and heterogeneous compute systems in which AI, DSP, and cryptographic functions are often combined. It can be improved in the future by dynamic voltage- frequency scaling with real-time prediction of workload, the use of on-chip learning engines to enable continuous adaptation, and the implementation of hardware-software co-optimized security measures to ensure the integrity of the accelerator when running in multidomain mode. In this direction, future developments will enhance the ability of the accelerator to perform effectively and safely in high ability changing computing environments.

References

1. Afifi, S., Thakkar, I., & Pasricha, S. SafeLight: enhancing security in optical convolutional neural network accelerators. https://doi.org/10.1109/ISCA61057.2024.00085
2. Alhomoud, A. Real time FPGA implementation of a high speed for video encryption and decryption system with high level synthesis tools. https://doi.org/10.1109/Access51460.2024.3400267
3. Amuru, D., Zahra, A., Vudumula, H. V., Kuntamalla, S. K., & Pasricha, S. AI/ML algorithms and applications in VLSI design and technology. https://doi.org/10.1109/ACCESS.2023.3283995
4. Baddour, Y., Hedayatipour, A., & Rezaei, A. REDACTOR: eFPGA redaction for DNN accelerator security. https://doi.org/10.1109/DATE58872.2025.2009
5. Bhandari, J., Chowdhury, A. B., Nabeel, M., & Pasricha, S. ASCENT: amplifying power side-channel resilience via learning & Monte-Carlo Tree Search. https://doi.org/10.1109/DAC62706.2024.200375
6. Bursztein, E., Invernizzi, L., Král, K., Krenn, S., Picek, S., & Van der Leest, V. Generalized power attacks against crypto hardware using long-range deep learning. https://doi.org/10.1109/CSF57053.2023.10170420
7. Ding, H., Kang, C. C., Xi, S., & Ding, X. FPGA-optimized hardware accelerator for fast Fourier transform and singular value decomposition in AI. https://doi.org/10.1109/DATE58872.2025.2007
8. Snousi, H. M., & Aleej, F. A. (2025). Energy-efficient VLSI architecture for lightweight CNN inference on edge devices. Journal of Reconfigurable Hardware Architectures and Embedded Systems, 2(1), 7-13.
9. Dorofte, M., & Krein, K. (2024). Novel approaches in AI processing systems for their better reliability and function. International Journal of Communication and Computer Technologies, 12(2), 21-30.https://doi.org/10.31838/IJCCTS/12.02.03
10. Ebel, A., & Reagen, B. Osiris: a systolic approach to accelerating fully homomorphic encryption. https://doi.org/10.1109/ISCA61057.2024.00067
11. Fan, Z. Enhancing energy efficiency in intelligent edge systems through hardware-algorithm co-design. https://doi.org/10.13140/RG.2.2.14815.82087
12. Geng, X., Wang, Z., Chen, C., & Li, Y. From algorithm to hardware: A survey on efficient and safe deployment of deep neural networks. https://doi.org/10.1109/ACCESS.2024.3364966
13. Kavitha, M. (2024). Energy-efficient algorithms for machine learning on embedded systems. Journal of Integrated VLSI, Embedded and Computing Technologies, 1(1), 16-20.https://doi.org/10.31838/JIVCT/01.01.04
14. Li, P., Che, C., & Hou, R. Nacc-Guard: a lightweight DNN accelerator architecture for secure deep learning. https://doi.org/10.1109/ICCD58380.2023.00021
15. Madhanraj. (2025). AI-powered energy forecasting models for smart grid-integrated solar and wind systems. National Journal of Renewable Energy Systems and Innovation, 1-7.
16. Maheswaran, K., Bossut, C., Wanna, A., Zhang, A., & Parno, B. CRYPTONITE: scalable accelerator design for cryptographic primitives and algorithms. https://doi.org/10.1109/DATE58872.2025.2010
17. Sadulla, S. (2025). Energy-efficient motor control algorithms for variable load industrial processes. National Journal of Electric Drives and Control Systems, 32-39.
18. Srivastava, A., Das, S., Choudhury, N., Roy, D., & Raha, P. SCAR: power side-channel analysis at RTL level. https://doi.org/10.1109/ESWEEK/CODES+ISSS.2024.00065
19. Sun, R., Ni, Y., He, X., Wu, C., Li, R., & Yuan, F. ONE-SA: Enabling Nonlinear Operations in Systolic Arrays for Efficient and Flexible Neural Network Inference. https://doi.org/10.1109/ISCA61057.2024.00078
20. Saranya, N. (2024). Smart load forecasting in microgrids: a comprehensive review of reinforcement learning algorithms and applications. SECITS Journal of Scalable Distributed Computing and Pipeline Automation, 1(1), 48-54.
21. Taheri, M., Cherezova, N., Ansari, M. S., & Singh, A. Exploration of activation fault reliability in quantized systolic array-based DNN accelerators. https://doi.org/10.1109/DATE58872.2024.2003
22. Tandi, M. R., & Shrirao, N. M. (2025). Optimizing sustainable energy microgrids in smart cities using IoT and renewable energy integration. Journal of Smart Infrastructure and Environmental Sustainability, 2(1), 45-50.