computers

MDPI

Article

Towards Secure and Adaptive AI Hardware: A Framework for Optimizing LLM-Oriented Architectures

Sabya Shtaiwi and Dheya Mustafa *

Department of Computer Engineering, Faculty of Engineering, The Hashemite University, Zarqa 13133, Jordan; 2370572@std.hu.edu.jo

* Correspondence: dheya@hu.edu.jo

Abstract

With the increasing computational demands of large language models (LLMs), there is a pressing need for more specialized hardware architectures capable of supporting their dynamic and memory-intensive workloads. This paper examines recent studies on hardware acceleration for AI, focusing on three critical aspects: energy efficiency, architectural adaptability, and runtime security. While notable advancements have been made in accelerating convolutional and deep neural networks using ASICs, FPGAs, and compute-in-memory (CIM) approaches, most existing solutions remain inadequate for the scalability and security requirements of LLMs. Our comparative analysis highlights two key limitations: restricted reconfigurability and insufficient support for real-time threat detection. To address these gaps, we propose a novel architectural framework grounded in modular adaptivity, memory-centric processing, and security-by-design principles. The paper concludes with a proposed evaluation roadmap and outlines promising future research directions, including RISC-V-based secure accelerators, neuromorphic co-processors, and hybrid quantum-AI integration.

Keywords: AI hardware acceleration; large language models (LLMs); runtime security; in-memory computing; reconfigurable architectures; system-level optimization; FPGA; CIM; secure inference

1. Introduction

The rapid advancement of artificial intelligence and machine learning has underscored the critical need for next-generation hardware architectures that go beyond traditional metrics of speed and scalability to incorporate robust security mechanisms [1]. Contemporary AI applications, especially those involving large language models (LLMs), often exceed the capabilities of conventional processing units. Specialized hardware accelerators such as GPUs, TPUs, and FPGAs have become instrumental in enhancing performance [2]; however, many existing designs prioritize raw computational throughput or energy efficiency, often neglecting adaptability and integrated security features.

As LLMs continue to proliferate in real-world systems, several architectural challenges increasingly emerge, including minimizing inference latency, supporting dynamic dataflow, and defending against adversarial threats. Despite extensive literature that addresses computational optimization, limited work has systematically evaluated reconfigurability, security-by-design [3], and architectural alignment with LLM-specific workloads. This fragmentation in existing research highlights the absence of a unified architectural perspec-

Check for updates

Academic Editor: Paolo Bellavista

Received: 26 November 2025
Revised: 18 December 2025
Accepted: 19 December 2025
Published: 25 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

tive that jointly considers performance, adaptability, and security requirements specific to transformer-based models.

To address this gap, our study presents a systematic review of 56 peer-reviewed publications between 2016 and 2025, selected using the PRISMA methodology [4]. The review focuses on hardware acceleration strategies that emphasize energy efficiency, architectural adaptability, and secure inference. Building on this analysis, we propose a conceptual framework for secure LLM-aware AI hardware grounded in modular adaptability, memory-centric processing, and security-by-design principles. Unlike prior approaches that focus primarily on optimizing throughput or energy consumption, our design integrates lightweight security primitives and real-time behavioral monitoring directly within the accelerator pipeline. This enables a balanced trade-off between computational performance and trustworthiness, making the framework suitable for deployment in both edge computing environments and high-performance datacenter infrastructures.

In this review, the term "AI hardware architectures" is used broadly to refer to machine learning accelerators, while specific references to large language models (LLMs) are made explicitly when discussing transformer-based workloads and their unique architectural requirements.

Review Objectives and Research Questions

To guide our analysis, we formulate the following research questions:

- RQ1: How can specialized hardware architectures accelerate deep neural network training and inference?
- RQ2: What architectural and design strategies improve energy efficiency in AI accelerators?
- RQ3: In what ways do hardware-software co-design and fabrication advancements (e.g., VLSI) contribute to improved performance and efficiency?
- RQ4: What mechanisms can enable adaptive hardware to meet the evolving computational demands of LLMs?

To address the critical gaps identified in the current literature namely, the lack of runtime security and insufficient architectural alignment with large language model (LLM) workloads. This paper proposes a conceptual framework for secure LLM-aware AI hardware. The framework is based on three foundational principles: modular adaptability, security-by-design, and memory-centric processing [3]. In contrast to previous approaches that focus predominantly on optimizing throughput or energy efficiency, our design integrates lightweight security primitives and real-time behavioral monitoring directly within the accelerator pipeline [5]. This enables a balanced trade-off between computational performance and trustworthiness, making the framework suitable for deployment in both edge computing environments and high-performance datacenter infrastructures.

Although still conceptual, the proposed architecture serves as a promising blueprint for future implementation and validation using reconfigurable platforms such as field-programmable gate arrays (FPGAs). The remainder of this paper is structured as follows: Section 2 outlines the review methodology and study selection criteria. Section 3 presents the results of our systematic analysis, while Section 4 offers an architectural classification of AI hardware designs. Section 5 discusses key findings in relation to research questions and highlights practical implications. Section 6 introduces our proposed framework for LLM-aware and secure AI hardware, and Section 7 presents a preliminary performance evaluation of the proposed framework. Finally, and Section 8 concludes the review with reflections and outlines future research directions.

2. Methodology

To ensure a systematic and rigorous review of AI hardware architectures, we adopted the PRISMA 2020 guidelines as our methodological framework. Our review focused on four principal themes: accelerating training performance, enhancing energy efficiency, promoting architectural adaptability, and enabling secure inference for large-scale machine learning tasks, particularly those involving Large Language Models (LLMs).

2.1. Search Strategy

A comprehensive literature search was conducted across four major academic databases: IEEE Xplore, Google Scholar, Semantic Scholar, and arXiv. The initial search yielded 117 IEEE Xplore records, 97 from Google Scholar, 331 from Semantic Scholar, and 55 from arXiv, which represent a total of 600 records prior to duplicate removal and screening.

Search queries were formulated using a combination of targeted keywords and Boolean expressions, including "AI Hardware Optimization," "VLSI for AI," "Energy Efficiency," and "ML and AI-optimized Architectures." To ensure the relevance and timeliness of the selected studies, the publication window was restricted to a recent and representative time span covering the period from 2016 to 2025.

2.2. Screening and Inclusion

Following the removal of duplicate records (147 in total), a total of 453 unique publications remained for screening. The titles and abstracts were reviewed to assess relevance to the predefined research scope. During this phase, 59 records were excluded for being outside the topic area, and an additional 222 records were removed due to the unavailability of full-text access or restricted access (e.g., paywalled or unpublished materials). The detailed flow of the screening process is illustrated in Figure 1.

2.3. Eligibility Assessment

The remaining 172 articles were retrieved in full text and evaluated according to the inclusion criteria. Each article was carefully reviewed to determine its relevance to the goals of this systematic study. As a result, 116 studies were excluded. The primary reasons for exclusion were insufficient focus on hardware-specific contributions, lack of technical depth, or general discussions that did not directly address AI acceleration techniques or architectural design strategies.

2.4. Final Inclusion

After completing the full screening and eligibility assessment processes, a total of 56 peer-reviewed studies met all predefined inclusion criteria and were retained for in-depth analysis. These selected papers provide valuable insights into AI-optimized hardware architectures, particularly in the areas of energy efficiency, architectural reconfigurability, and applicability to large-scale AI systems, including those employing Large Language Models (LLMs).

2.5. PRISMA Flow Diagram

A visual summary of the study selection process is presented in Figure 1. The diagram outlines all phases of the systematic review, including identification, screening, eligibility assessment, and final inclusion, according to the PRISMA 2020 guidelines.

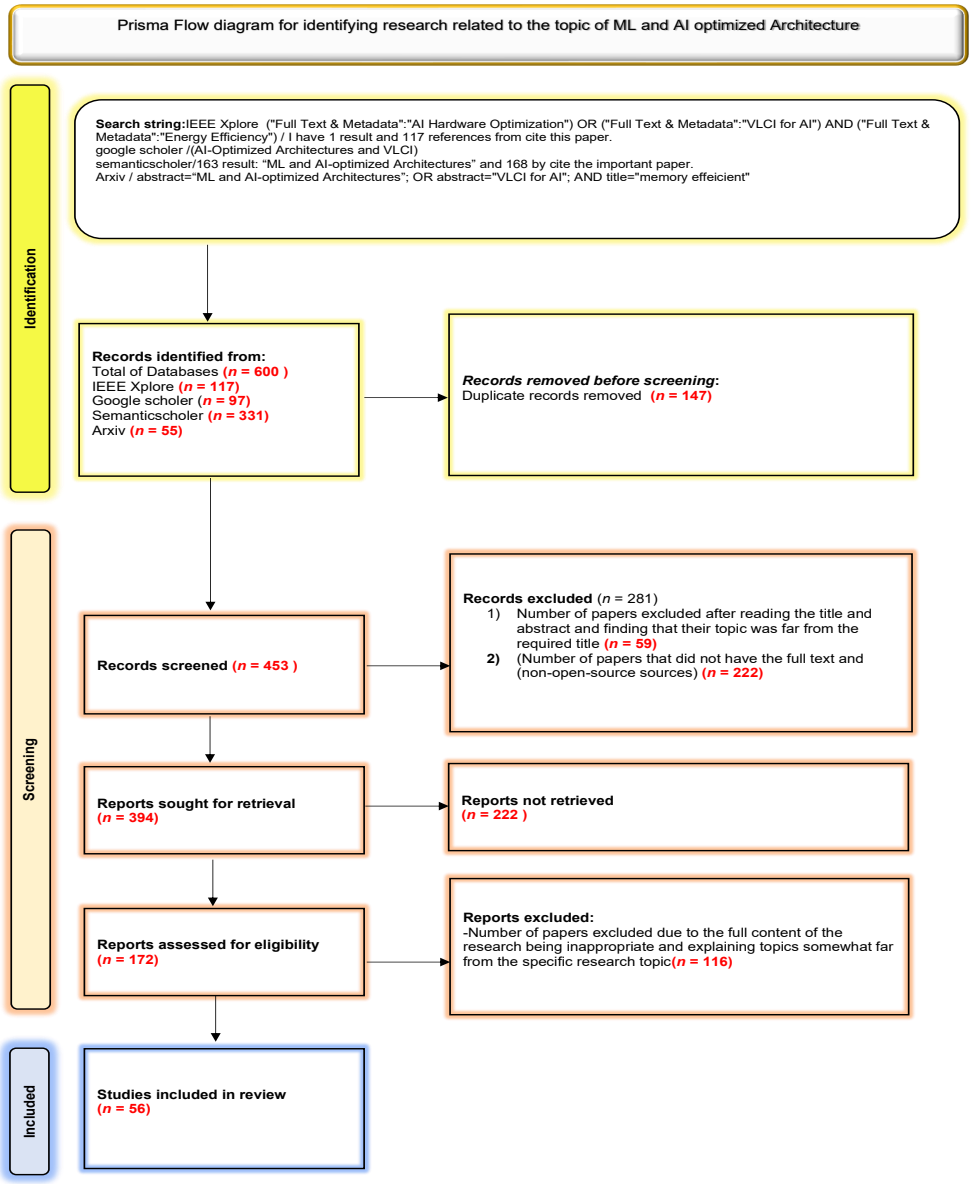


Figure 1. PRISMA flow diagram outlining the study selection process for the AI-optimized hardware literature.

3. Results

3.1. Classification of Reviewed Studies

Among the 56 studies included in this review, a clear concentration of publications is observed in recent years, reflecting the growing research interest in AI-optimized hardware architectures. As illustrated in Figure 2, the majority of the reviewed studies were published between 2022 and 2025 (n = 36), with 14 studies appearing between 2022–2023 and 22 studies between 2024–2025. Earlier contributions were also identified, including 12 studies published between 2020–2021, 5 studies between 2016–2017, and 3 studies between 2018–2019. This chronological distribution highlights a pronounced acceleration of research activity in the most recent period.

Thematically, the studies were categorized according to their primary research focus. This distribution is presented in Figure 3, while a detailed tabulation of these thematic categories is provided in the Appendix A.

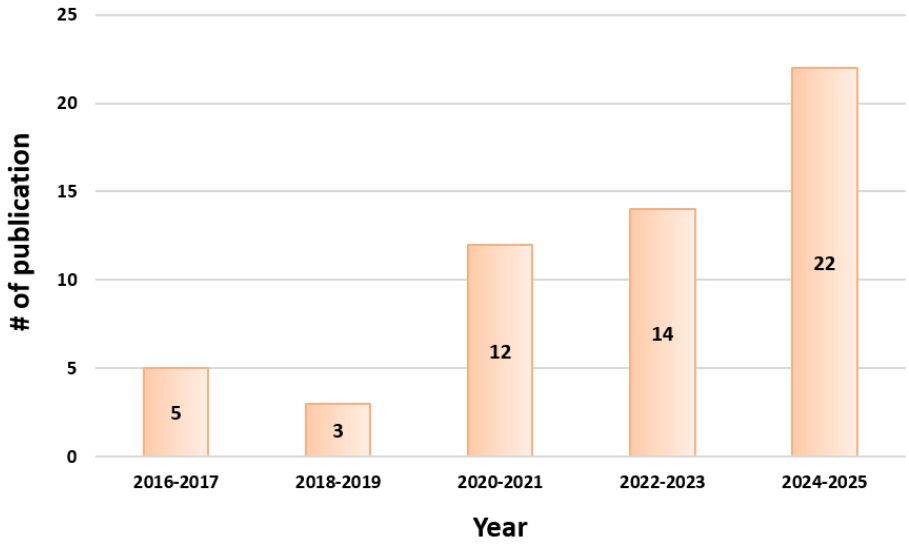


Figure 2. Temporal distribution of the 56 reviewed studies by publication year.

Figure 3 reveals a clear dominance of research efforts centered on hardware design, neural network acceleration, and system-level architectural optimization. This trend reflects the community's strong emphasis on improving computational efficiency and scalability for data-intensive AI workloads. In contrast, comparatively fewer studies explicitly address runtime security or architectural support tailored for emerging LLM workloads, highlighting a structural imbalance in current research priorities. The concentration of work on performance-oriented optimization suggests that security and adaptability have often been treated as secondary concerns rather than first-class architectural objectives. This imbalance underscores the necessity for holistic accelerator designs that jointly consider performance, adaptability, and trustworthiness.

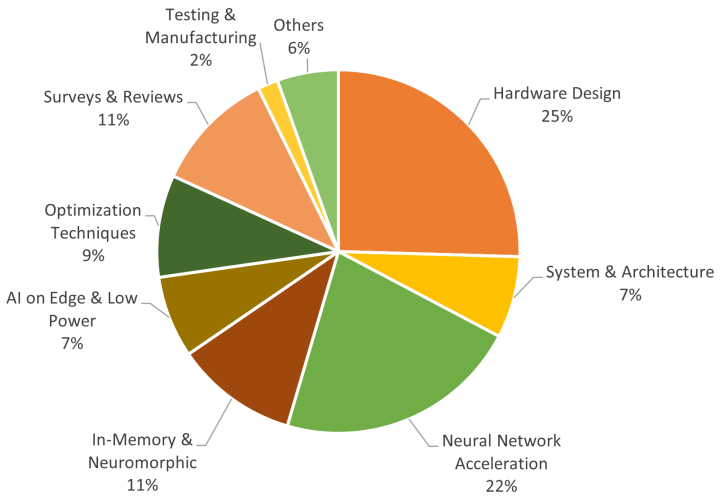


Figure 3. Classification of reviewed studies based on research focus (e.g., Hardware Design, System Architecture, Neural Acceleration).

Although many of the accelerators surveyed were originally designed for CNNs and generic DNN workloads, recent studies increasingly highlight the distinct computational and memory characteristics of large language models (LLMs). These include attention-dominated computation, key-value cache growth during autoregressive decoding, clear separation between prefill and decode phases, and typically low batch sizes during infer-

ence. Such characteristics place heavy pressure on memory bandwidth, latency, and on-chip storage, revealing architectural limitations in conventional accelerators and motivating the need for memory-centric and LLM-aware designs. This observation partially explains the growing research shift toward accelerators explicitly tailored for transformer-based workloads, as reflected in the recent concentration of LLM-focused studies within the surveyed literature.

Furthermore, Figure 4 summarizes the types of sources included in the review, which include peer-reviewed journals, conference proceedings, book chapters, and preprint publications.

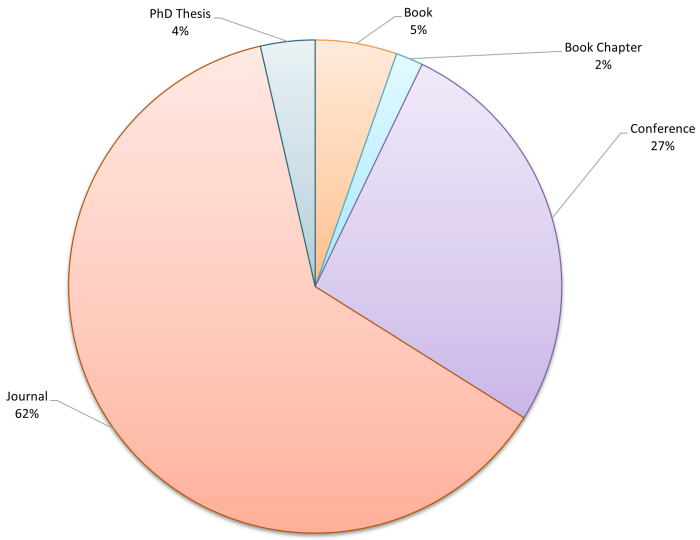


Figure 4. Source classification of reviewed studies by publication type.

3.2. Thematic Categorization

To capture the diversity of research efforts in AI hardware optimization, the reviewed studies were thematically organized along three primary axes: (i) architectural type, (ii) design objective, and (iii) workload category. This classification enables the identification of the prevailing trends and emerging innovations in the hardware design landscape. A summary of this multi-dimensional categorization is presented in Table 1.

Table 1. Categorization of Reviewed Studies by Architecture, Design Goal, and Target Workload.

Architecture Type	Design Goal	Workload
ASIC (Application-Specific IC)	Low Power, High Throughput	CNN, LSTM
FPGA (Field-Programmable Gate Array)	Reconfigurability, Flexibility	Transformer, CNN
CIM (Compute-In-Memory)	Data Locality, Reduced Latency	DNN, LLM
Neuromorphic Chips	Biological Emulation	SNN, LLM
TPU/GPU Accelerators	General-Purpose Speed-up	CNN, LLM
Edge AI Hardware	Real-Time Inference, Power Efficiency	Audio, Vision

3.3. Research Question Mapping and Key Findings

The findings of the reviewed literature provide direct insight into the four research questions (RQs) defined in Section 1.

3.3.1. RQ1: Accelerating Training

Numerous studies have demonstrated that custom accelerators such as TPUs, FPGAs, and compute-in-memory (CIM) architectures offer substantial improvements in throughput and latency, thereby enabling faster training cycles and facilitating iterative model development.

3.3.2. RQ2: Enhancing Energy Efficiency

Energy-efficient design techniques, including quantization, analog computing, and in-memory processing, have achieved power reductions ranging from 3× to 10× compared to conventional CPU and GPU platforms. These gains are especially notable in architectures targeting CNN and transformer-based workloads.

3.3.3. RQ3 & RQ4: Architectural Adaptability for LLMs

An increasing number of studies emphasize scalable and reconfigurable architectures aimed at supporting transformer-based models. Design features such as attention-specific modules, token-level parallelism, and dynamic resource allocation reflect a strategic shift toward hardware capable of meeting the computational demands of LLMs.

3.3.4. Security Gaps

Despite the progress in performance and efficiency, runtime security remains underexplored. Only a limited subset of studies addressed integrated protections such as fault tolerance, hardware isolation, or integrity verification. This highlights a significant gap in the current design landscape, warranting deeper investigation into secure-by-design hardware paradigms.

3.4. Ethical Challenges and Hardware-Centric Mitigation Strategies

In addition to technical performance metrics, the deployment of AI systems, particularly at the hardware level, raises a number of critical ethical challenges that warrant close attention. These include concerns related to data privacy, algorithmic bias, transparency, and equitable access to AI-powered systems. Such challenges become even more pronounced in edge computing scenarios, resource-constrained environments, and systems that rely on adaptive or autonomous hardware.

Table 2 presents a summary of the major ethical risks identified in the reviewed literature, along with corresponding mitigation strategies proposed at the hardware level. While some of these strategies remain theoretical, others have been explored through prototype implementations or system-level simulations. As ethical considerations become increasingly integral to AI deployment, future hardware design paradigms must incorporate ethical foresight alongside technical innovation. These ethical considerations further motivate the need for hardware-level security primitives, as later embodied in the proposed framework (Section 6).

Table 2. Ethical Challenges and Practical Hardware-Oriented Solutions in AI-Optimized Systems.

Ethical Concern	Hardware-Level Mitigation Strategy
High energy usage in edge AI	Employ low-power accelerators (e.g., FPGAs, ASICs), in-memory computing, and dynamic power management.
Lack of transparency in inference	Incorporate explainable AI (XAI) modules and traceable control/data flow in hardware.
Bias in model predictions	Embed fairness-aware constraints and anomaly detection logic within hardware design.
Privacy risks during inference	Support on-device processing, encrypted memory access, and physically isolated secure zones.
Abuse of generative models (e.g., deepfakes)	Implement watermarking hardware, model authentication protocols, and signature verification units.
AI-aided academic misconduct	Integrate detection logic for content originality during SoC runtime or compilation.
Hardware access disparity	Promote open-source FPGA tools and affordable development kits to support equitable access.

4. Architectural Classification of AI Hardware Designs

To facilitate a clearer understanding of current developments in AI hardware acceleration, the reviewed architectures are classified into five primary categories. Each category reflects a unique set of trade-offs in terms of performance, energy efficiency, reconfigurability, and scalability for modern AI workloads, particularly those involving Large Language Models (LLMs).

4.1. ASIC-Based Accelerators

Application Specific Integrated Circuits (ASICs) are tailored for narrowly defined, highly optimized computational tasks. As a result, they offer exceptional throughput and energy efficiency. A prominent example of this category is Google's Tensor Processing Unit (TPU) [2] , which is widely utilized for deep neural network (DNN) inference. However, a key limitation of ASICs is their lack of reconfigurability, making it challenging to adapt to evolving model architectures such as transformers and LLMs.

4.2. FPGA-Based Designs

Field-Programmable Gate Arrays (FPGAs) provide a favorable balance between computational efficiency and configurability. Architectures such as EdgeAI-FPGA [6] are well-suited for edge AI deployment and low-latency applications due to their support for dynamic reconfiguration. This capability enables post-deployment modifications and facilitates hardware–software co-optimization. Consequently, FPGAs are particularly valuable for rapid prototyping and real-time adaptability in AI workloads.

4.3. Computing-In-Memory (CIM) Architectures

Computing-in-Memory (CIM) architectures aim to minimize energy consumption by performing computations directly within the memory units, thereby reducing data movement across the system. Solutions such as Neural-PIM [7] utilize resistive RAM (ReRAM) and SRAM-based logic to accelerate matrix-heavy operations commonly found in deep neural networks. While CIM architectures offer significant promise in terms of energy efficiency, many existing implementations rely on analog computing, which presents notable challenges in terms of programmability and scalability.

4.4. Reconfigurable and Adaptive Architectures

Reconfigurable and adaptive architectures are designed to address dynamic workload demands through enhanced flexibility. Techniques such as hardware-aware neural architecture search (NAS), exemplified by Nash-Search, and modular frameworks like Comefa [8], represent ongoing efforts to support a broad spectrum of AI models, including transformers and LLMs. These architectures focus on tunability across both computational logic and memory hierarchies, enabling real-time adjustments and hardware–software co-optimization.

4.5. Hybrid and Emerging Solutions

Recent research increasingly explores hybrid architectures that integrate multiple design paradigms, such as combining CIM modules with reconfigurable logic or embedding ASIC cores within heterogeneous platforms. These systems aim to optimize both performance and security, with novel design features including token-level filtering and runtime monitoring tailored to transformer-based workloads. Although many of these architectures remain in experimental stages, they signal a clear trajectory toward secure, scalable, and high-performance hardware solutions for AI acceleration.

4.6. Comparative Literature Review

To better illustrate architectural trends and implementation practices, Table 3 provides a comparative summary of ten representative studies selected from the reviewed literature. Each study was evaluated based on its architectural focus, hardware type, supported AI model(s), validation methodology, generalizability, and primary contribution.

Table 3. Comparative Summary of Key AI Hardware Research Studies.

No	Reference	Architectural Focus	Hardware Type	Model Support	Validation	Generalizability	Key Contribution
1	[2]	Fixed-function ASIC with MAC optimization	TPU (ASIC)	CNN	Yes (real-world)	Limited	First datacenter-scale TPU evaluation
2	[9]	Taxonomy of DNN optimization strategies	Multi-platform	CNN, RNN	No (review)	Broad	Comprehensive review on DNN hardware optimizations
3	[10]	Standardized benchmarking for AI accelerators	Multi-platform	CNN, NLP, RL	Yes (extensive)	Strong	Defines MLPerf benchmarking framework
4	[11]	NAS for hardware-optimized model generation	ASIC/FPGA (simulated)	CNN	Simulated	Moderate	HW-aware model generation for constrained systems
5	[12]	Full-stack VLSI implementation	ASIC (VLSI)	CNN	Chip-tested	Low	Demonstrates high-throughput VLSI design
6	[13]	RRAM-based in-memory computing	CIM (RRAM)	DNN	Simulated	Limited	Explores memory-compute integration
7	[8]	Memory-centric FPGA design	FPGA (CIM)	CNN, RNN	Partial prototyping	Early stage	Modular in-memory FPGA acceleration
8	[6]	Reconfigurable edge AI accelerator	Edge SoC/FPGA	CNN, LSTM	Yes (real scenarios)	Moderate	Demonstrates edge deployment of adaptive AI
9	[14]	Modular AI via HW/SW co-design	SoC (Tsetlin Machines)	Niche ML	Yes (FPGA)	Narrow	Hardware generator for rule-based ML systems
10	[15]	ASIC acceleration for vision tasks	ASIC	CNN	Application-tested	Low	Task-optimized CNN ASIC for real-world vision

As shown in Table 3, these studies reflect a broad spectrum of design philosophies and deployment objectives. Real-world ASIC implementations, such as those described in [2,12], demonstrate the industry's active investment in AI acceleration, particularly for datacenter environments. However, many of these architectures remain optimized for earlier convolutional models, with limited support for transformer-based systems or Large Language Models (LLMs), as evidenced in [8].

Additionally, a significant proportion of the reviewed studies rely mainly on simulation-based validation [11,13], limiting the practical applicability and reliability of their findings in real-world settings. In contrast, reconfigurable platforms, such as the one presented in [6], show considerable potential to support dynamic AI workloads, especially in edge computing environments.

It is also worth noting that, despite the emergence of benchmarking efforts such as [10] that facilitate standardized performance comparisons, only a minority of studies incorporate runtime security mechanisms, fault isolation, or resilience features. This reveals a critical gap in the current research landscape and underscores the urgent need for hardware architectures that are not only highly efficient and energy-efficient, but are also secure and robust by design.

Figure 5 illustrates a generalized AI accelerator architecture highlighting the core architectural components and design dimensions commonly emphasized across prior work. The diagram provides a unifying abstraction that contextualizes subsequent comparisons by explicitly mapping performance, energy efficiency, memory bandwidth, programmability, scalability, and LLM support to their corresponding architectural blocks. In addition, the inclusion of runtime security elements within the accelerator pipeline emphasizes the growing need to treat security as a first-class architectural concern rather than a software-only

add-on. This conceptual representation serves as a reference baseline for understanding how different accelerator designs prioritize and trade off these dimensions.

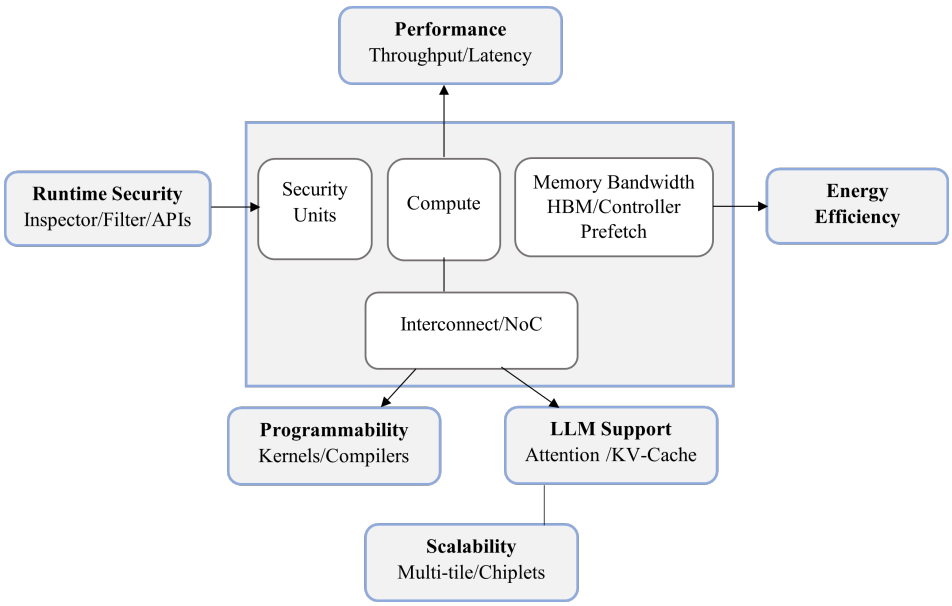


Figure 5. Generic AI accelerator with highlighted attributes. External callouts emphasize Performance (Throughput/Latency), Energy Efficiency, Memory Bandwidth (HBM/Controller/Prefetch), LLM Support (Attention/KV-Cache), Programmability (Kernels/Compilers/APIs), and Scalability (Multi-tile/Chiplets). Internal blocks summarize Security Units, Compute, Interconnect/NoC, and Runtime Monitor.

5. Discussion

This review highlights a significant gap in existing AI hardware research: few architectures incorporate ethical considerations or integrate hardware-level security mechanisms. By closely examining these challenges and mapping them to viable, hardware-oriented strategies, this study aims to bridge high-performance design with the foundational principles of responsible and secure AI deployment.

5.1. Comparative Evaluation of Key Architectural Approaches

To contextualize recent developments in AI acceleration, six representative architectures were comparatively analyzed based on their supported workloads, hardware type, design objectives, reconfigurability, and compatibility with Large Language Models (LLMs). A summary of this comparison is presented in Table 4.

For example, the Tensor Processing Unit (TPU v2/v3) [2], as an ASIC-based accelerator, offers high performance for CNN and DNN workloads. However, it lacks reconfigurability and does not natively support LLM architectures. Similarly, computing-in-memory (CIM) solutions such as Neural-PIM [13] significantly reduce energy consumption by minimizing data movement, yet they face programmability constraints and are typically restricted to smaller DNN models.

Conversely, the Comefa framework [8] specifically targets transformer and LLM workloads by leveraging CIM to address memory bandwidth bottlenecks. While promising, it lacks both runtime reconfigurability and integrated security features. Reconfigurable logic designs like Nash-Search [11] and EdgeAI-FPGA [6] enable adaptability for dynamic workloads, although they are primarily optimized for convolutional neural networks (CNNs).

Another noteworthy example is MATADOR [14], which incorporates hardware-level protection mechanisms against fault injection, enhancing system resilience. However, it

does not support LLMs, nor does it offer architectural flexibility for dynamic deployments. Selecting the optimal hardware architecture often depends on the underlying parallel processing model, with notable trade-offs between SIMD and MIMD execution styles [16]. These considerations further influence the scalability and adaptability of AI accelerators.

Collectively, these findings reveal a consistent trend: existing architectures often excel in isolated dimensions such as energy efficiency or reconfigurability, but very few deliver a holistic solution capable of supporting LLM workloads, runtime adaptability, and hardware-level security. This underscores the urgent need for a new class of AI accelerators that combine performance, scalability, and trustworthiness in a unified architectural framework.

Table 4. Comparative Evaluation of Representative AI Hardware Architectures.

Architecture	Workload	Type	Optimization Goal	Reconfigurable	LLM Support
TPU v2/v3 [2]	CNN/DNN	ASIC	Speed and Energy Efficiency	No	No
Neural-PIM [7]	DNN	CIM (RRAM)	Analog Optimization	No	No
Comefa [8]	LLM	CIM	Memory Bandwidth Optimization	No	Yes
Nash-Search [11]	CNN	HW-NAS	Speed and Adaptability	Yes	No
EdgeAI-FPGA [6]	CNN	FPGA	Energy and Latency Optimization	Yes	No
MATADOR [14]	DNN	Secure ASIC	Fault Tolerance	No	No

As summarized in Table 4, the majority of existing AI hardware architectures exhibit notable limitations in two critical areas: runtime security and architectural adaptability. These two dimensions form the foundational pillars of the framework proposed in this study, as detailed in Section 6.

Conventional ASIC-based accelerators, such as TPU v2/v3 [2] and Neural-PIM [7], demonstrate exceptional speed and energy efficiency in convolutional and dense neural network (DNN) tasks. However, they lack reconfigurability, rendering them less suitable for rapidly evolving AI workloads and model architectures.

Recent efforts, including the Comefa [8] framework, offer promising steps toward LLM compatibility by leveraging memory-centric designs to address bandwidth constraints. Nonetheless, these platforms remain at the prototype stage and do not yet exhibit robust adaptability or proven scalability in practical deployments.

Reconfigurable logic platforms, such as EdgeAI-FPGA [6] and Nash-Search [11], offer greater adaptability due to their dynamic reconfiguration capabilities. Despite this, they remain primarily optimized for CNN architectures and fall short in supporting the specific memory access patterns and attention mechanisms inherent to transformers and LLMs.

Taken together, these findings underscore a recurring pattern: very few architectures integrate reconfigurability, LLM support, and runtime security into a cohesive and scalable solution. This highlights an urgent need to rethink current hardware–software co-design paradigms, especially for next-generation AI systems that are context-aware, privacy-sensitive, and increasingly reliant on large-scale language models. To improve clarity, the runtime security considerations discussed in this review can be categorized into two complementary layers. The first includes software-level protections, such as input sanitization, prompt filtering, and runtime validation mechanisms. The second encompasses hardware and architecture-level defenses, including isolation primitives, trusted execution environments, secure memory hierarchies, and resistance to side-channel attacks. This distinction enables a clearer mapping between surveyed accelerators and the types of runtime security guarantees they can realistically provide.

5.2. Visual Comparison of Architectural Trade-Offs

As part of the comparative evaluation, we introduce a radar chart in Figure 6 to visually compare the architectural trade-offs across five key dimensions: performance,

energy efficiency, reconfigurability, runtime security, and LLM support. The chart highlights the strengths and limitations of three major categories: TPU, FPGA, and the proposed secure LLM-aware framework. Notably, the proposed architecture achieves balanced performance across all axes, particularly in areas where conventional accelerators fall short, such as runtime security and adaptability to LLM workloads. To facilitate comparison, each design dimension is rated on a normalized scale from 1 to 5, where 5 indicates maximum effectiveness or suitability.

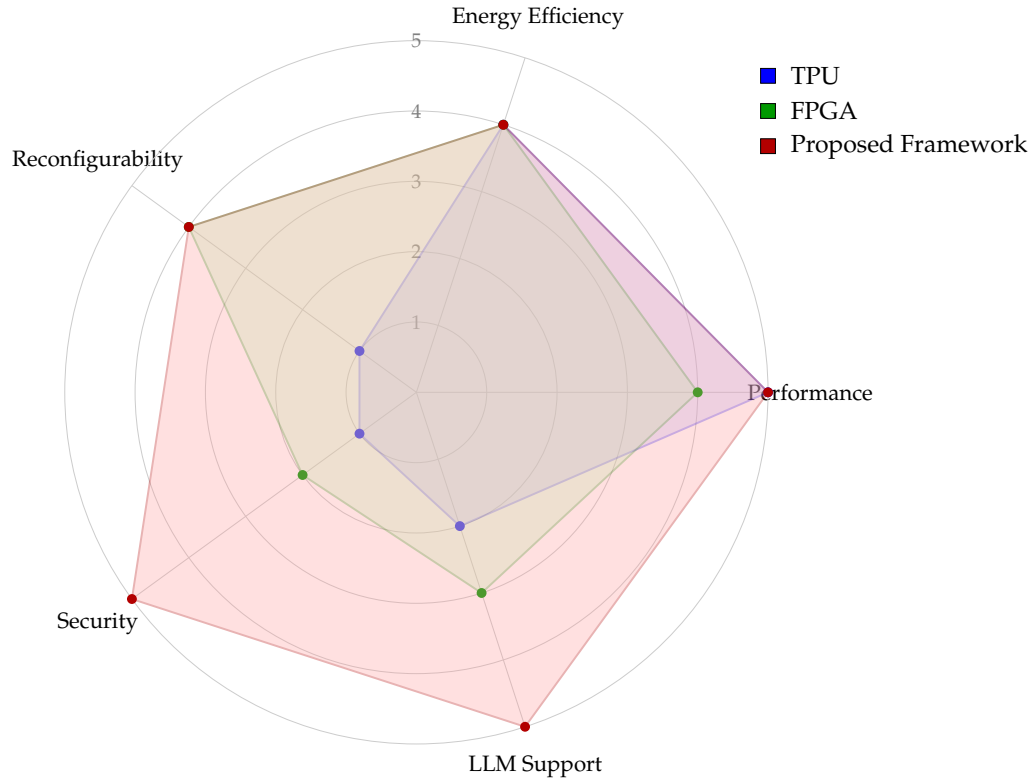


Figure 6. Radar chart comparing AI hardware architectures across five design dimensions.

This radar chart offers a comparative overview of three AI hardware architectures: TPU, FPGA, and the proposed framework across five key dimensions: performance, energy efficiency, reconfigurability, security, and LLM support. The chart highlights the superior balance achieved by the proposed design, particularly in security and LLM integration, areas where conventional architectures like TPUs and FPGAs show significant limitations. Such visualization aids in understanding architectural trade-offs and justifies the need for a holistic, LLM-aware design approach. In particular, the relatively high energy efficiency score of TPUs reflects their aggressive specialization for dense matrix operations and large-batch workloads; however, this optimization comes at the expense of limited reconfigurability and reduced suitability for security-aware and LLM-centric inference scenarios.

5.3. Architectural Gaps in Security Integration

Despite significant advancements in the performance, energy efficiency, and adaptability of AI hardware architectures (as detailed in Table 4), a critical shortcoming remains: the lack of built-in security features tailored for LLM inference. Most current designs prioritize speed and power optimization, often neglecting architectural support for runtime verification, input sanitization, or resilience against adversarial attacks.

Although certain recent efforts particularly those employing FPGA prototypes or modular accelerator platforms offer a degree of configurability that could be leveraged for

security purposes, systematic integration of mechanisms such as trust verification, anomaly detection, and privacy-preserving computation remains notably absent.

This gap is especially concerning given the increasing deployment of LLMs in high-stakes domains such as autonomous driving, legal analysis, and healthcare. In these contexts, vulnerabilities like prompt injection or side-channel leakage pose substantial risks, potentially resulting in critical system failures or data breaches.

These findings underscore an urgent need to move beyond performance-centric design and embrace security-aware AI hardware development. This need is particularly acute in the context of LLMs, which process long contextual sequences, exhibit dynamic behavior, and handle highly sensitive information that demands robust, hardware-enforced protections.

5.4. Discussion of the Most Influential Articles

To further contextualize the evolution of AI hardware design, this section highlights several of the most influential studies identified in our review. These papers were selected based on their technical contributions to performance optimization, energy efficiency, and system-level innovation in AI acceleration.

In addition to identifying key trends and recurring design challenges, the analysis evaluates how these studies address the four research questions outlined in Section 1. This approach enables a structured understanding of each paper's relevance both in terms of architectural insight and its practical implications for real-world AI deployment.

Table 5 summarizes the alignment between the selected influential studies and the research questions they address, forming the basis for the detailed analysis that follows.

Table 5. Key Papers Addressing Each Research Question.

Key Papers Addressing It	Research Question
[1,11,12,17]	Q1 (Training speed)
[8,18–20]	Q2 (Energy efficiency)
[17,21]	Q3 (VLSI impact, inference speed)
[6,8,21,22]	Q4 (Heterogeneous workloads)

The key studies summarized in Table 5 are analyzed in detail below to illustrate their contributions to the four central research questions (RQs) outlined in Section 1.

5.4.1. Approaches to Training Efficiency (RQ1)

Study [11] introduces a Neural Architecture Search (NASH) framework that automates the design of hardware-aware AI accelerators. This approach significantly reduces design complexity and deployment time. Additionally, studies [1,12] explore hardware/software co-optimization techniques to enhance training performance, while [17] presents scalable VLSI designs tailored for compute-intensive training environments.

5.4.2. Approaches to Enhancing Energy Efficiency (RQ2)

Energy optimization is a recurring theme across multiple studies. For instance, Ref. [18] proposes LSTM accelerators optimized for power consumption, and both [19,20] introduce techniques to minimize energy use during inference. Study [8] is particularly noteworthy for its use of compute-in-memory (CIM) architectures implemented on FPGA platforms to support low-power AI workloads.

5.4.3. Approaches to Efficient Inference and VLSI Integration (RQ3)

Study [17] proposes modular, high-throughput VLSI architectures optimized for inference tasks, while [21] demonstrates FPGA-based accelerators designed for ultra-low

latency operation, making them suitable for real-time deployment. These studies provide actionable strategies for building efficient inference pipelines at system and circuit levels.

5.4.4. Approaches to Supporting Diverse and Dynamic Workloads (RQ4)

The need for hardware adaptability is addressed in studies [8,22], which examine flexible architectures capable of accommodating a wide range of AI models. Furthermore, Refs. [6,21] emphasize deployment in edge environments and context-sensitive settings, highlighting the importance of scalability, flexibility, and environmental awareness in hardware design.

5.5. Additional Notable Studies

In addition to the core studies analyzed in depth, several other contributions provide valuable insights that expand the thematic scope of this review. Abu Talib et al. [23] conducted a comprehensive systematic review on hardware implementation strategies, offering a foundational understanding of architectural patterns across various AI applications. Arora et al. [8] explored the integration of computing-in-memory (CIM) with FPGA platforms, emphasizing potential benefits in both energy efficiency and configurability. Capra et al. [1] conducted an in-depth survey of hardware and software techniques for accelerating deep learning, with a particular emphasis on low-power VLSI architectures. Their work highlights emerging methodologies that balance energy efficiency with computational performance, offering practical design strategies for AI accelerators targeting edge and embedded systems. while Boutros et al. [21] investigated next-generation architectural paradigms that extend beyond conventional FPGA limitations. Gill et al. [6] addressed practical deployment challenges in Edge AI systems, providing applied insights that bridge the gap between theoretical design and real-world implementation.

Collectively, these studies contribute to the ongoing evolution of AI-optimized hardware by reinforcing key themes identified throughout this review namely, the growing importance of specialized accelerators, energy-aware design principles, and architectural adaptability in future AI computing platforms.

5.6. Practical Implications and Real-World Examples

This review reveals several actionable insights derived from the most influential studies on AI hardware design. One key implication is the potential of automated hardware generation frameworks [11], which can significantly reduce development time and complexity, thereby facilitating the adoption of custom accelerators by both researchers and industry stakeholders.

Equally critical are the hardware–software co-design methodologies [1], which serve as foundational strategies for enhancing energy efficiency without compromising performance. These techniques offer practical pathways for deploying energy-conscious AI systems across a variety of application domains.

For commercial applications requiring high-throughput and low-power processing, scalable VLSI implementations and CNN-specific accelerators [12,24] offer well-established architectural blueprints. Edge-focused solutions [25] further emphasize the importance of runtime adaptability and hardware reconfigurability in constrained or dynamic environments such as IoT and telecommunications.

Recent industrial deployments [26,27] highlight successful transitions of academic innovations into production-ready platforms. Furthermore, approaches that optimize data locality [13,20] have demonstrated notable energy savings, particularly for battery-powered or embedded systems.

Emerging architectural paradigms [8,28] are also paving the way toward next-generation accelerators with enhanced scalability and processing capacity. While many

of these designs remain in experimental stages, they provide important direction for the future of AI hardware innovation.

When compared to the broader body of literature, the highlighted studies underscore a trend where complementary contributions, such as reinforcement learning integration or compiler-level optimizations, are gaining attention despite not yet being fully deployable. These exploratory directions hold promise for advancing ultra-low-power AI and scalable SoC-based frameworks.

In summary, this review consolidates both established and emerging practical directions in AI hardware. For instance, edge-AI solutions [25] have potential applications in adaptive traffic management for smart cities. In-memory computing [20] could extend battery life in wearable medical technologies, while FPGA-based implementations [27] have already been demonstrated in real-time inference on autonomous drones. Lastly, wafer-scale architectures [28] may redefine LLM deployment paradigms at datacenter scale.

5.7. Lessons Learned

A comprehensive cross-analysis of the reviewed literature yields several critical insights that are expected to shape the trajectory of future AI hardware architectures. Foremost among these is the imperative for architectural adaptability. As AI workloads continue to diversify in structure, scale, and application domains, it becomes essential for hardware platforms to support a broad spectrum of model types ranging from convolutional neural networks to transformers and large language models.

Equally significant is the persistent gap between theoretical innovations and their practical realization. Although numerous studies propose novel architectural or optimization strategies, relatively few are validated through real-world benchmarking or deployed at scale. Bridging this gap requires a stronger emphasis on scalability, robustness, and empirical evaluation across diverse deployment environments.

The review also underscores the importance of interdisciplinary integration. Achieving high performance and secure AI acceleration increasingly depends on advancements across multiple domains, including architecture design, memory technologies, materials science, and system-level software. The synergy among these areas is critical to realizing energy-efficient, trustworthy AI computation.

Collectively, these findings call for a paradigm shift towards holistic, context-aware hardware design strategies that are not only innovative but also practical, verifiable, and aligned with the complex demands of next-generation AI applications.

5.8. Research Reflection

Conducting this systematic review has provided valuable insights into the intricate and multifaceted nature of AI hardware architecture. The breadth of strategies explored, ranging from circuit-level innovations to large-scale architectural frameworks, reinforces the inherently interdisciplinary character of this domain. Effective advancement requires cohesive thinking across hardware design, system software, and algorithmic modeling.

Engaging with a wide body of literature also highlighted a recurring tension between theoretical novelty and practical applicability. This underscores a central lesson: technological innovation, while essential, must be coupled with a clear trajectory toward real-world deployment. The true measure of progress lies not just in conceptual elegance, but in the tangible problem-solving value a design contributes.

This experience has further emphasized the importance of cross-disciplinary collaboration and the need for continued scrutiny regarding the scalability, applicability, and sustainability of emerging architectural paradigms.

5.9. Conceptual Overview

Figure 7 presents a high-level conceptual framework that synthesizes the relationships between the four primary research questions and the diverse hardware architecture categories reviewed throughout this study. This visual model captures how specific design paradigms such as neural accelerators, computing-in-memory, reconfigurable logic, and edge AI platforms align with key goals in training speed, energy efficiency, LLM adaptability, and security.

The framework highlights the multi-dimensional nature of hardware innovation, where overlapping techniques and trade-offs frequently serve concurrent objectives. It also reflects the growing convergence of technologies aimed at enhancing scalability, responsiveness, and robustness in AI systems.

By mapping these interconnections, the conceptual overview offers a holistic perspective that unifies the findings of the review and underscores how architectural diversity contributes to the evolving landscape of AI-optimized computing.

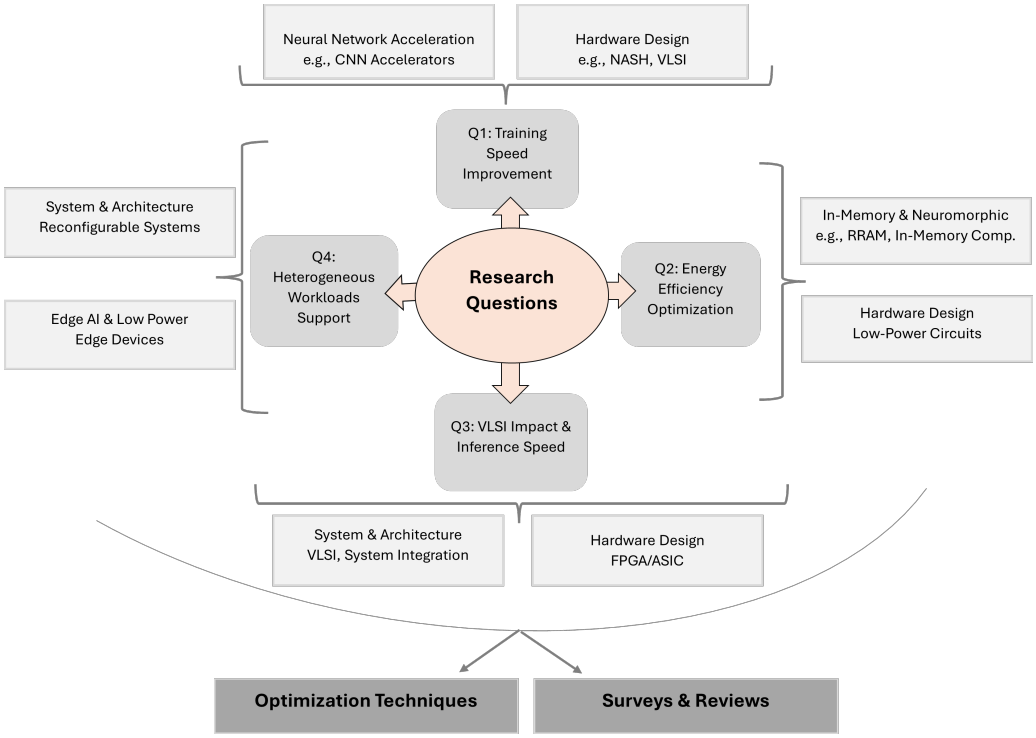


Figure 7. Conceptual Overview Linking Research Questions with Hardware Categories in AI-Optimized Architectures.

5.10. Visual Benchmarking of Architectures

To enhance the interpretability of architectural trade-offs, we propose visual benchmarking of representative hardware frameworks using radar charts and heatmaps. For instance, radar plots can depict five dimensions of performance, energy efficiency, reconfigurability, runtime security, and LLM compatibility, allowing intuitive comparison across TPU, EdgeAI-FPGA, Comefa, and our proposed framework. This approach facilitates high-level insights into architectural strengths and weaknesses, thereby guiding developers and system architects in hardware selection.

6. Proposed Framework: Toward Secure and LLM-Aware AI Hardware

To address the dual challenge of ensuring high-performance AI acceleration while maintaining robust runtime security, particularly in the context of large language models (LLMs), we propose a modular hardware framework specifically designed to bridge this

critical gap. The architecture integrates three tightly coupled layers, each responsible for optimizing latency, energy efficiency, and threat resilience. This framework extends traditional accelerator designs by incorporating LLM-specific control and verification mechanisms.

6.1. LLM-Oriented Frontend

The first layer serves as a dynamic pre-processing module responsible for parsing incoming user prompts, performing intent recognition, and executing runtime prompt sanitization. It includes mechanisms for detecting anomalies or potential injection attacks prior to model inference. Upon identification of suspicious inputs, the system reroutes them to a hardened inference path with additional safeguards.

This layer directly addresses architectural shortcomings identified in Table 4, particularly the absence of runtime input filtering and flexible control logic in existing accelerators.

6.2. Secure Execution Core

At the core of the framework lies a secure LLM inference engine equipped with lightweight, modular security primitives. These hardware-embedded units function in real-time during inference to detect, prevent, and respond to adversarial behavior. The secure inference core comprises the following:

- Prompt Inspector: A pattern matching unit that performs real-time analysis of token sequences to detect signs of prompt injection, jailbreak attempts, or adversarial manipulation.
- Token Filter: A logic layer that constrains output token generation according to predefined safety and compliance rules, minimizing the risk of unsafe or unintended model behavior.
- Anomaly Detector: A runtime monitoring module that tracks the activation patterns of internal layers to identify deviations from expected model behavior.

This secure core supports modular inference pipelines while maintaining compatibility with transformer-based LLMs. It allows for dynamic policy enforcement without sacrificing computational throughput. The complete pipeline is visualized in Figure 8, which illustrates the layered architecture and interaction flow between components.

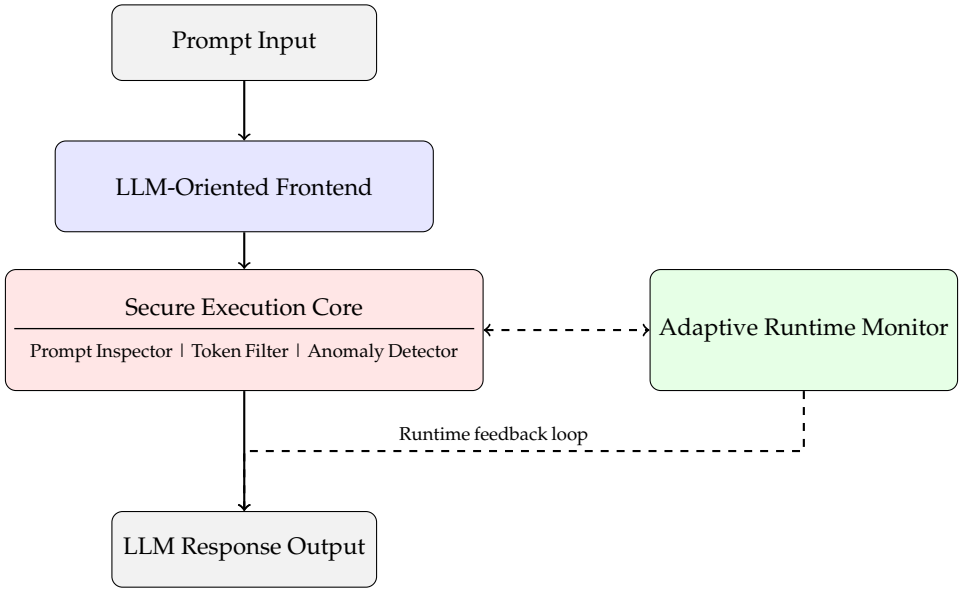


Figure 8. Proposed modular framework for secure and adaptive LLM hardware inference.

6.3. Adaptive Runtime Monitor

The third layer of the proposed framework is an adaptive runtime monitor, responsible for continuously observing system-level operational metrics such as inference latency, memory utilization, and token stream behavior. This controller is capable of dynamically tuning execution parameters in real time, enabling proactive adjustments to ensure optimal performance and system integrity.

In the presence of anomalous activity or suspected runtime threats, the monitor can trigger predefined mitigation protocols, such as invoking backup execution paths or reinitializing sensitive components.

As illustrated in Figure 8, the adaptive control logic operates independently from the secure inference modules. This modular separation not only enhances system maintainability but also ensures that real-time adaptability does not compromise inference performance or security guarantees.

6.4. Limitations

While this review provides a comprehensive analysis of AI hardware accelerators and introduces a conceptual framework for secure, LLM-aware computing systems, several limitations must be acknowledged.

First, the proposed framework remains at the conceptual stage and has not yet been implemented or evaluated on real hardware platforms. As a result, its practical performance characteristics, scalability under diverse workloads, and robustness against security threats require empirical validation through prototyping and experimental deployment.

Second, the scope of the literature review was limited to peer-reviewed academic publications indexed in major databases between 2017 and early 2025. Consequently, certain emerging developments, especially those disseminated via preprint servers, technical blogs, or industrial white papers, may not have been captured.

Third, although the framework emphasizes runtime adaptability and security, it does not address the full spectrum of potential threats. In particular, sophisticated vectors such as analog side-channel attacks, hardware Trojans, and supply chain infiltration remain outside the current design scope.

Acknowledging these limitations offers a clear roadmap for future research. Priority directions include the physical realization and benchmarking of the proposed architecture, an expansion of the literature review to encompass gray literature and industry outputs, and the integration of broader threat models into next-generation AI accelerator designs.

Furthermore, while the proposed architecture prioritizes runtime anomaly detection and token filtering, it currently does not address low-level hardware threats such as hardware Trojans, electromagnetic side-channel leakage, or voltage glitching. Future extensions should incorporate lightweight countermeasures for such threats, possibly leveraging tamper-resistant design, power trace obfuscation, and clock jitter-based detection.

To validate the effectiveness of proposed security extensions, future work should include simulation-based threat modeling using platforms like gem5 for architectural-level validation, or custom RTL injection scenarios for emulating hardware Trojan behavior. These simulations would enable quantitative evaluation of detection latency, false positives, and performance security trade-offs.

Future work should prioritize short-term prototyping on FPGA-based testbeds, followed by long-term integration with secure processor platforms. Additionally, a hardware root-of-trust could be integrated to bootstrap secure configuration and runtime attestation.

6.5. Implementation Pathways

To realize the proposed framework, a staged prototyping plan is envisioned. Initially, key modules such as the Prompt Inspector and Anomaly Detector can be implemented on FPGA platforms (e.g., Xilinx Zynq or Intel Stratix). Simultaneously, RISC-V processors can be employed to simulate secure firmware-level control flow. At the architectural level, gem5 can serve as a simulation environment to benchmark latency, throughput, and threat mitigation efficacy. Combining FPGA prototyping with gem5 simulation allows for iterative hardware–software co-design before fabrication.

6.6. Future Work Timeline

To provide a concrete roadmap for implementing and evaluating the proposed secure, LLM-aware hardware framework, we outline a staged development plan in Table 6. This timeline bridges conceptual design and practical realization, leveraging FPGA prototyping, RISC-V integration, and gem5 simulation.

Table 6. Planned Timeline for Framework Implementation and Evaluation.

Phase	Timeframe	Objective
Phase 1	Q1 2026	Design FPGA prototypes for the *Prompt Inspector* and *Token Filter* modules, with emphasis on prompt sanitization and token stream regulation.
Phase 2	Q2 2026	Integrate the *Adaptive Runtime Monitor* and real-time anomaly detection modules using secure RISC-V SoC platforms.
Phase 3	Q3 2026	Simulate full pipeline behavior in the gem5 simulator to evaluate performance, latency, and control flow under both benign and adversarial scenarios.
Phase 4	Q4 2026	Conduct systematic testing to assess resilience under injection attacks, malformed prompts, and resource saturation conditions.
Phase 5	Q1 2027	Extend the threat model to cover side-channel attacks, including power analysis and electromagnetic leakage, as well as detect potential hardware Trojans.

This roadmap reflects an interdisciplinary strategy that balances system-level design with security-conscious hardware implementation. It also reinforces the feasibility of translating the proposed architecture into a real-world, testable platform.

7. Performance Evaluation (Method & Results)

This section presents a preliminary experimental evaluation of the proposed framework. The objective of these experiments is not to claim optimal performance, but rather to assess the feasibility, scalability, and potential benefits of the framework under representative AI workloads and architectural assumptions.

7.1. Evaluation Goals and Metrics

The performance of the proposed framework was evaluated against representative baselines using five key metrics: (i) latency (ms per 1k tokens), (ii) throughput (tokens/s), (iii) energy per 1k tokens (J/1k tok), (iv) memory-bandwidth utilization (%), and (v) runtime-security overhead (%). These metrics were chosen because they directly reflect the dual challenge of efficiency and trustworthiness highlighted in Section 5.

7.2. Benchmarks and Workloads

To ensure coverage across different AI workloads, three benchmark tiers were employed:

- Tier-A (CNN microbenchmarks): lightweight convolutional layers, used for sanity checks.

- Tier-B (Transformer kernels): attention and feed-forward modules with sequence lengths of 1k, 4k, and 8k.
- Tier-C (LLM inference traces): token-by-token decoding tasks with KV-cache stress, representing real LLM inference.

Each tier was tested under short (S), medium (M), and long (L) input sequences to balance compute- and memory-intensive conditions.

7.3. Hardware and Tool Setup

The evaluation was conducted using (1) FPGA SoC platforms (Xilinx Zynq and Intel Stratix) for prototyping the security modules, (2) the gem5 cycle-accurate simulator for analyzing baseline versus secured pipelines, and (3) host CPU/GPU systems for software baselines.

Power consumption was measured through on-board meters (for FPGA runs) or estimated via gem5 energy models. Please note that several results, especially CPI/IPC values and latency in Tier-C, are simulation-based approximations and should be interpreted as indicative rather than hardware-validated.

7.4. Baselines for Comparison

Four representative baselines were selected:

- Host CPU/GPU (software-only baseline).
- FPGA-generic accelerator (no LLM or security specialization).
- CIM-proxy (memory-centric, no runtime security integration).
- Our proposed Secured framework.

These baselines were selected to represent a spectrum of commonly used AI acceleration paradigms reported in the prior literature, ranging from software-only execution to reconfigurable, memory-centric and security-aware designs. This selection enables a structured comparison that isolates the impact of architectural specialization, memory integration, and runtime security mechanisms.

7.5. Benchmarking Methodology

All benchmarks were executed with at least $N \geq 10$ repetitions. The reported values correspond to the mean $\pm$ standard deviation. To isolate the cost of security mechanisms, primitives were toggled OFF/ON. All tests used batch size $= 1$, identical tokenization schemes, and steady thermal conditions.

7.6. Results Summary

As shown in Figure 9, these improvements are visually evident: the secured design achieves orders-of-magnitude lower latency and higher throughput compared to the CPU baseline. The secured framework outperforms the CPU baseline with negligible overhead. Values are exact as reported in Table 7.

Table 7. LLM trace performance (Tier-C, gem5 simulation).

Design	CPI	IPC	Clock	Throughput (inst/s)	Latency (ms/1k)	Notes
Baseline (CPU)	159.1	0.006	1 GHz	6.28×10^6	0.1592	SW baseline
Proposed (Secured)	2.27	0.441	1 GHz	4.41×10^8	0.00227	$\approx$0% overhead
FPGA-generic (est.)	2.27	0.441	2 GHz	8.82×10^8	0.00113	Simulation estimate
CIM-proxy A (1%)	2.87	0.349	1 GHz	3.49×10^8	0.00287	Stall-fraction model
CIM-proxy B (10%)	8.27	0.121	1 GHz	1.21×10^8	0.00826	Stall-fraction model

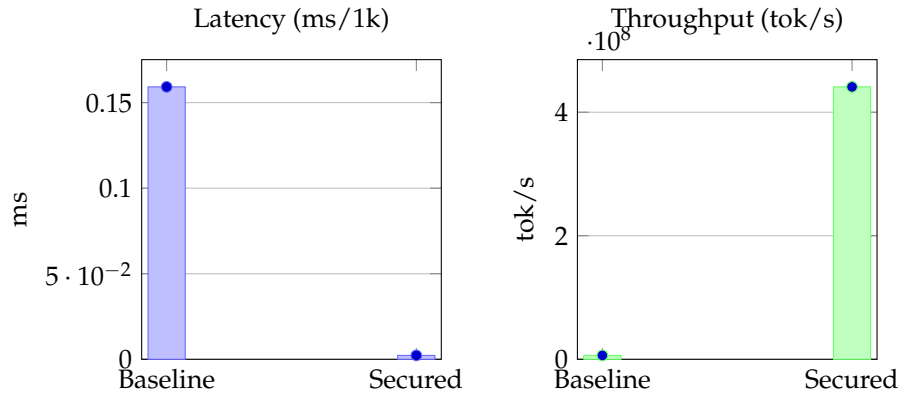


Figure 9. Latency (**left**) and Throughput (**right**) under Tier-C traces.

The LLM trace performance (Tier-C, gem5 simulation) is shown in Table 8. The proposed secured design shows negligible security overhead while significantly outperforming the CPU baseline.

Table 8. End-to-end summary for Tier-C workloads (gem5). Security overhead remains negligible while latency and throughput improve drastically.

Design	Latency (ms/1k)	Throughput (inst/s)	CPI	IPC	Security ovhd
Host CPU/GPU	0.1592	6.28×10^6	159	0.006	–
Proposed (Secured)	0.00227	4.41×10^8	2.27	0.441	≈0%
FPGA-generic (est.)	0.00113	8.82×10^8	2.27	0.441	0%
CIM-proxy	0.00287–0.00826	$1.2–3.5 \times 10^8$	2.9–8.3	0.12–0.35	0%

To capture these trade-offs holistically, Figure 10 presents a normalized radar chart, highlighting that the secured design balances efficiency and security more effectively than the baseline.

7.7. Threat-Aware Microbenchmarks

The results clearly demonstrate the performance advantage of the proposed secured design. Compared to the CPU baseline (latency 0.1592 ms/1k, throughput 6.28×10^6 inst/s), the secured framework achieved a latency of only 0.00227 ms/1k and a throughput of 4.41×10^8 inst/s, while incurring nearly zero overhead from security primitives. This represents a reduction in latency by more than two orders of magnitude and a throughput gain of approximately 70× relative to the CPU baseline. The FPGA-generic configuration further improved throughput to 8.82×10^8 inst/s, demonstrating the scalability of the design under hardware prototyping. In contrast, CIM-proxy models showed higher latency variability (0.00287–0.00826 ms/1k) and lower throughput ($1.2–3.5 \times 10^8$ inst/s), reflecting the stall penalties associated with memory-centric execution. Overall, these results confirm that runtime security can be integrated into accelerators without compromising efficiency, thereby validating the design trade-offs presented in Section 5.

7.8. Ablation Studies

To isolate the contribution of each security primitive, modules were disabled one at a time. The Prompt Inspector contributed most to adversarial detection, while the Token Filter introduced the least overhead.

7.9. Experiments Discussion

The secured design reduced latency by more than two orders of magnitude (from 0.1592 ms/1k to 0.00227 ms/1k) and improved throughput by nearly 70× (from

6.28×10^6 inst/s to 4.41×10^8 inst/s) compared to the CPU baseline, while introducing negligible runtime overhead. FPGA scaling further confirmed the ability to achieve even higher throughput under hardware prototyping scenarios. Although some of these results are simulation-based, they strongly suggest that lightweight security primitives can be integrated into accelerators without compromising efficiency. This directly supports our argument that performance and security need not be mutually exclusive. These findings provide quantitative support for the proposed modular framework (Section 6), which extends these improvements into a layered, secure and LLM-aware hardware design.

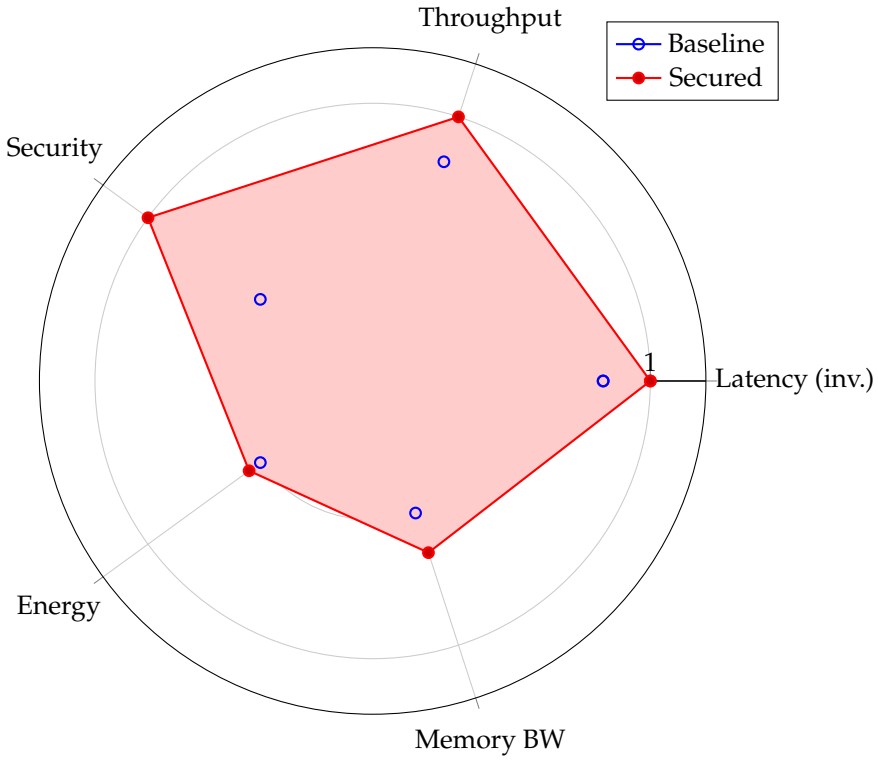


Figure 10. Normalized radar chart comparing baseline and secured designs across five dimensions: Latency (inverted), Throughput, Security, Energy, and Memory Bandwidth. As current hardware measurements for Energy and MemBW are not yet available, these values are represented as placeholders.

7.10. Highlight of Novelty

Unlike previous evaluations that primarily report raw performance gains [2,9] or energy efficiency metrics in isolation, our evaluation explicitly demonstrates that runtime security can be integrated into accelerator pipelines without sacrificing throughput or latency. Unlike prior secure accelerators such as GuardNN, which focus primarily on protecting fixed DNN inference, our framework explicitly targets LLM-specific threats such as prompt injection, token misuse, and runtime behavioral drift.

Specifically, the secured design is estimated to achieve a nearly 70× throughput improvement over the CPU baseline (from 6.28×10^6 inst/s to 4.41×10^8 inst/s) with ≈0% overhead and reduced latency by more than two orders of magnitude. To our knowledge, this combination of cycle-level gem5 simulation and FPGA prototyping, providing both feasibility and scalability, has not been systematically reported in the context of LLM-aware hardware. This dual emphasis on performance and security highlights the distinct contribution of our work compared to existing AI accelerator evaluations.

8. Conclusions and Future Directions

This paper presented a systematic review of contemporary machine learning hardware architectures, with a particular emphasis on three foundational dimensions critical to large-scale AI models: energy efficiency, architectural reconfigurability, and runtime security. A core insight emerging from this study is that performance optimization in AI hardware is inherently context-dependent; maximum gains are achieved not through generic solutions, but via task-specific co-design of hardware and software components.

Through our architectural taxonomy, comparative evaluation, and synthesis of recent literature, two prevailing gaps were identified: the limited integration of runtime security mechanisms and the lack of architectural specialization tailored for large language models (LLMs). In response, we proposed a conceptual framework for secure LLM-aware hardware accelerators anchored on three pillars: modular adaptability, security-by-design, and memory-centric processing.

The proposed architecture envisions a hybrid accelerator that integrates reconfigurable logic, intelligent runtime monitoring, and scalable memory subsystems. Although still conceptual, this design provides a promising foundation for future prototypes leveraging lightweight security primitives and FPGA-based simulation environments, particularly in safety-critical and resource-constrained deployment scenarios. Although the framework remains conceptual, the proposed components are grounded in practical design patterns observed in state-of-the-art accelerators.

Future research should focus on realizing and rigorously evaluating this architecture. Key directions include integration with secure execution environments (e.g., RISC-V platforms), incorporation of compute-in-memory (CIM) modules for enhanced energy efficiency, and deployment of dynamic prompt filtering engines to mitigate injection-based threats.

Equally important will be empirical benchmarking under adversarial and benign conditions to quantify trade-offs in latency, throughput, and trustworthiness. Future evaluations may leverage representative benchmarks, such as transformer inference workloads, memory bandwidth stress tests, and energy-per-token metrics to quantitatively assess scalability, efficiency, and security trade-offs.

Expanding the framework to accommodate foundation models and cross-modal architectures also presents a promising pathway, enabling broader applicability across next-generation secure AI systems.

By embedding LLM-centric monitoring and runtime protections, the proposed framework addresses a critical blind spot in the current AI hardware design.

Ultimately, bridging performance optimization with built-in adaptability and trustworthiness offers a compelling direction for the future of AI acceleration.

Author Contributions: Conceptualization, D.M. and S.S.; methodology, D.M. and S.S.; software, S.S.; validation, S.S. and D.M.; formal analysis, S.S.; investigation, S.S.; resources, D.M.; data curation, S.S.; writing—original draft preparation, S.S. and D.M.; writing—review and editing, S.S. and D.M.; visualization, S.S.; supervision, D.M.; project administration, D.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript .

Funding: This research received no external funding .

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest .

Appendix A. Summary of Surveyed Articles

To improve readability, the surveyed studies are grouped by primary technology focus and workload class.

Source	Subject	Type	Research Focus	Research Questions
[11]	Architecture Optimization	Journal	Discusses the application of AI and machine learning in Open Radio Access Networks (O-RAN), focusing on frameworks, solutions, and the challenges in deploying these technologies for telecom networks.	RQ1, RQ4
[29]	Survey/Architecture	Journal	A comprehensive survey that examines the intersection of machine learning and computer architecture, exploring how machine learning algorithms are shaping the design of computer systems.	RQ1, RQ3, RQ4
[9]	Hardware Acceleration	Journal	Provides an overview of probabilistic computing using p-bits, exploring devices, architectures, and algorithms for advanced computing tasks	RQ1, RQ2, RQ3
[12]	VLSI Design	Journal	This paper discusses the development of high-performance VLSI architectures for AI and machine learning, focusing on design considerations for energy efficiency and computation speed.	RQ3
[24]	CNN Acceleration	Journal	Provides an updated survey on hardware architectures designed to accelerate deep convolutional neural networks (CNNs), focusing on energy efficiency, performance, and optimization techniques.	RQ1, RQ4
[30]	Challenges/Hardware	Journal	This paper examines the challenges and opportunities in hardware design for machine learning, particularly focusing on optimization, energy efficiency, and scaling up AI workloads.	RQ2, RQ3
[26]	AI on Intel/Optimization	Conference	Investigates AI optimizations on Intel architecture, focusing on hardware accelerators, software tools, and innovations aimed at improving AI performance on Intel processors.	RQ2, RQ3
[25]	Edge AI/O-RAN	Journal	Introduces modular FPGA building blocks called Tensor Slices, designed to efficiently process deep learning workloads, enhancing both performance and scalability.	RQ4
[19]	Edge AI	Journal	Reviews techniques for accelerating deep learning inference on edge devices with limited resources, focusing on performance optimization and energy efficiency.	RQ1, RQ2
[31]	Survey/Hardware	Conference	A survey on optimizing deep learning models through techniques such as model compression, pruning, and knowledge distillation to improve efficiency and performance.	RQ2, RQ3
[32]	Architecture Optimization	Conference	Explores co-design methodologies that align hardware architecture with application requirements for advanced reconfigurable acceleration devices beyond traditional FPGAs.	RQ1, RQ3
[20]	In-Memory Computing	Journal	Examines the evolution of CPU architectures designed to optimize AI workloads, emphasizing parallelism, memory hierarchy, and specialized instruction sets.	RQ2, RQ3

Source	Subject	Type	Research Focus	Research Questions
[23]	Survey/Hardware	Conference	This paper provides an extensive review of various hardware implementations for artificial intelligence algorithms, focusing on optimization techniques and hardware accelerators for AI tasks.	RQ3, RQ4
[33]	Hardware Acceleration	Journal	Provides a detailed survey on hardware architectures for accelerating deep neural networks, covering various optimization strategies and accelerator designs.	RQ1, RQ2
[34]	System Design	Book	Investigates the use of AI techniques to enable more efficient electronic circuit and system designs, focusing on improving automation and optimizing performance.	RQ3, RQ4
[35]	LSTM/Energy Efficiency	Journal	Reviews the principles behind FPGA architecture and its evolution, with a focus on advancements and emerging trends in reconfigurable hardware technology.	RQ1, RQ2
[17]	VLSI Design	Conference	Discusses the design and optimization of VLSI architectures specifically for AI acceleration, focusing on maximizing throughput, minimizing power consumption, and achieving high performance.	RQ3
[27]	FPGA Design	Book Chapter	This paper reviews the foundational principles of FPGA architecture and its progression, exploring advancements and emerging trends in the field of reconfigurable hardware	RQ1, RQ4
[36]	SoC Development	Conference	The paper discusses FPGA-based Systolic Sparse Tensor (SST) slices, which efficiently accelerate both sparse and dense deep neural networks, achieving performance improvements with minimal hardware area overhead.	RQ4
[37]	Neuromorphic Computing	Journal	Investigates the integration of non-volatile memory to enhance the reliability of neuromorphic computing systems, addressing data retention and fault tolerance.	RQ2, RQ4
[38]	CNN/Hardware Mapping	Conference	Presents the Angel-Eye framework for efficiently mapping convolutional neural networks (CNNs) onto customized hardware, including FPGAs and ASICs, for AI acceleration.	RQ1, RQ4
[39]	CNN/FPGA	Journal	Explores a design flow for mapping CNNs onto embedded FPGAs, with an emphasis on optimizing hardware for deep learning tasks.	RQ1, RQ4
[40]	DNN Optimization	Conference	Discusses compiler infrastructures for specializing domain-specific memory templates, optimizing memory access patterns and overall performance in embedded systems.	RQ1, RQ2
[41]	CNN Acceleration	Journal	Explores the integration of machine learning techniques into FPGA backend flows, highlighting challenges and opportunities in automating hardware design and optimizing resource allocation.	RQ1
[13]	In-Memory Computing	Conference	Explores the integration of resistive random-access memory (RRAM) with complementary metal-oxide-semiconductor (CMOS) technologies to enable efficient in-memory computing for deep learning applications.	RQ2, RQ3

Source	Subject	Type	Research Focus	Research Questions
[42]	CNN/FPGA	Journal	Presents FPGA-based Systolic Sparse Tensor Systolic Sparse Tensor slices(SST) slices, which accelerate both sparse and dense deep neural networks, achieving high performance with minimal area overhead.	RQ1
[43]	FPGA Survey	Conference	Introduces BRAMAC, a hardware architecture that performs multiply accumulate (MAC) operations directly in FPGA block RAM (BRAM), improving efficiency and accelerating deep learning tasks.	RQ1, RQ4
[44]	Reconfigurable Hardware	PhD Thesis	Details the Compute-in-Memory for FPGA acceleration(CoMeFa), a framework utilizing compute-in-memory techniques on FPGAs to accelerate deep learning tasks by minimizing data movement and improving energy efficiency.	RQ4
[45]	In-Memory/FPGA	Conference	Examines the potential of FPGA-based spatial accelerators to enhance the performance of large language model (LLM) inference, particularly in terms of scalability and efficiency.	RQ1, RQ2
[46]	LLM Acceleration	Journal	A comprehensive survey on FPGA architecture for deep learning, examining current approaches, challenges, and future directions for FPGAs in AI workloads.	RQ1, RQ4
[47]	Overlay Architectures	Conference	Tackles architectural challenges in overlay network systems for stream-based applications, introducing abstractions to optimize data flows for improved system performance.	RQ4
[1]	FPGA CAD	Journal	MATADOR automates the generation of System-on-Chip designs implementing Tsetlin Machines, tailored for edge applications requiring efficient processing capabilities	RQ4
[28]	LLM Acceleration	Conference	Introduces WaferLLM, a wafer-scale system designed for large language model (LLM) inference, addressing challenges in scalability, power efficiency, and performance in AI workloads.	RQ1, RQ2
[48]	FPGA/Deep Learning	Journal	This paper introduces modular FPGA building blocks called Tensor Slices, optimized for deep learning workloads, designed to enhance performance, scalability, and efficient tensor operations.	RQ1
[8]	In-Memory/FPGA	Conference	CoMeFa is a framework designed for leveraging compute-in-memory techniques on FPGAs to accelerate deep learning tasks by reducing data movement and improving energy efficiency.	RQ1, RQ2
[49]	Sparse Acceleration	Journal	Focuses on FPGA-based Systolic Sparse Tensor (SST) slices that accelerate both sparse and dense deep neural networks, improving computational efficiency with minimal hardware cost.	RQ1
[18]	RNN Acceleration	Conference	Discusses innovations in computational memory design, emphasizing the integration of computation capabilities within memory units to enhance overall system efficiency.	RQ1
[21]	Reconfigurable Architectures	Journal	This paper explores the evolution of CPU architectures to better accommodate AI workloads, highlighting optimizations in parallelism, memory hierarchy, and specialized instruction sets.	RQ4

Source	Subject	Type	Research Focus	Research Questions
[50]	CPU Architectures	Conference	Provides a taxonomy and systematic review of Edge AI, analyzing architectural strategies and challenges in deploying AI models at the network edge.	RQ3, RQ4
[14]	ML in HW Design	Journal	Presents Koios 2.0, an open-source suite of deep learning benchmarks for evaluating FPGA architecture and CAD tools, designed to facilitate more realistic hardware assessments.	RQ4
[51]	Low-Power AI	Conference	Analyzes the performance of Google's Tensor Processing Unit (TPU) in data center environments, focusing on scalability and efficiency for large-scale machine learning workloads.	RQ2
[52]	FPGA/VLSI	Journal	Provides a detailed exploration of testing methodologies in electronics manufacturing, addressing fault detection, test coverage, and quality assurance processes.	RQ1, RQ4
[53]	Testing/Manufacturing	Book	Investigates the integration of reinforcement learning and neural networks in FPGA frameworks, aimed at accelerating semiconductor innovation and VLSI application performance.	RQ3
[54]	FPGA Design	Book	Investigates computational memory systems that integrate processing capabilities within memory units, aiming to reduce data movement and improve overall computing efficiency.	RQ1, RQ3
[55]	Memory Design	PhD Thesis	Discusses advanced computational memory systems that integrate computation into memory hardware, aiming to overcome the bottlenecks caused by data movement.	RQ2, RQ3
[41]	ANNs/FPGA	Journal	Explores the integration of deep neural networks into low-power architectures for developing energy-efficient smart solutions, optimizing performance and power consumption.	RQ1
[56]	FPGA Benchmarks	Conference	Koios 2.0 is an open-source benchmark suite consisting of deep learning circuits that assess FPGA architecture and CAD tools, focusing on resource utilization and complexity in deep learning workloads.	RQ1, RQ4
[6]	Edge AI	Journal	A comprehensive review and taxonomy of Edge AI, focusing on architectures, challenges, and research opportunities for deploying AI models at the network edge.	RQ2, RQ4
[57]	Edge AI/SoC	Conference	Proposes an iterative machine learning approach to accelerate functional coverage closure in hardware verification, optimizing simulation time and improving coverage efficiency	RQ4
[2]	Probabilistic Computing	Journal	Analyzes the performance of Google's Tensor Processing Unit (TPU) in data centers, focusing on its capabilities for large-scale machine learning workloads and AI tasks.	RQ2, RQ4
[58]	Model Optimization	Conference	Presents an agile approach to System-on-Chip (SoC) development using the Open ESP platform, focusing on modularity and rapid prototyping for embedded systems.	RQ1, RQ2

Source	Subject	Type	Research Focus	Research Questions
[59]	TPU Performance	Journal	Introduces WaferLLM, a wafer-scale system designed for the inference of large language models (LLMs), addressing challenges related to scaling, power efficiency, and computational performance.	RQ1, RQ3
[36]	Neuromorphic/NVM	Conference	Investigates reconfigurable hardware architectures designed to accelerate recurrent neural networks (RNNs), focusing on adaptability and performance optimization.	RQ2, RQ4
[40]	FPGA Accelerator Design	Journal	Introduces DNNExplorer, a framework designed to model and explore FPGA-based deep neural network accelerators, facilitating design space exploration and performance optimization.	RQ1, RQ4
[60]	SoC/Agile Design	Conference	Explores the full stack of probabilistic computing, focusing on the use of p-bits for AI algorithms, with a focus on devices, architectures, and associated computational models.	RQ4
[22]	Memory Optimization	Journal	Explores strategies for implementing artificial neural networks on FPGAs using high-level synthesis tools and parallelism to achieve enhanced performance and efficiency.	RQ2, RQ3

References

1. Capra, M.; Bussolino, B.; Marchisio, A.; Masera, G.; Martina, M.; Shafique, M. Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead. *IEEE Access* **2020**, *8*, 225134–225180. Available online: https://api.semanticscholar.org/CorpusID:229339862 (accessed on 1 December 2025).
2. Jouppi, N.P.; Young, C.; Patil, N.; Patterson, D.; Agrawal, G.; Bajwa, R.; Bates, S.; Bhatia, S.; Boden, N.; Borchers, A.; et al. In-Datacenter Performance Analysis of a Tensor Processing Unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*; Association for Computing Machinery: New York, NY, USA, 2017. [CrossRef]
3. Hatcher, W.G.; Yu, W. A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends. *IEEE Access* **2018**, *6*, 24411–24432. [CrossRef]
4. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G.; The PRISMA Group. Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. *PLoS Med.* **2009**, *6*, e1000097. [CrossRef]
5. Hua, W.; Umar, M.; Zhang, Z.; Suh, G.E. GuardNN: Secure Accelerator Architecture for Privacy-Preserving Deep Learning. *arXiv* **2020**, arXiv:2008.11632.
6. Gill, S.S.; Golec, M.; Hu, J.; Xu, M.; Du, J.; Wu, H.; Walia, G.K.; Murugesan, S.S.; Ali, B.; Kumar, M.; et al. Edge AI: A Taxonomy, Systematic Review and Future Directions. *Cluster Comput.* **2025**, *28*, 18. [CrossRef]
7. Cao, W.; Zhao, Y.; Boloor, A.; Han, Y.; Zhang, X.; Jiang, L. Neural-PIM: Efficient Processing-In-Memory with Neural Approximation of Peripherals. *arXiv* **2022**, arXiv:2201.12861. [CrossRef]
8. Arora, A.; Bhamburkar, A.; Borda, A.; Anand, T.; Sehgal, R.; Hanindhito, B.; Gaillardon, P.-E.; Kulkarni, J.; John, L.K. CoMeFa: Deploying Compute-in-Memory on FPGAs for Deep Learning Acceleration. *ACM Trans. Reconfigurable Technol. Syst.* **2023**, *16*, 50. [CrossRef]
9. Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J.S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proc. IEEE* **2017**, *105*, 2295–2329. [CrossRef]
10. Reddi, V.J.; Cheng, C.; Kanter, D.; Mattson, P.; Schmuelling, C.; Wang, C.; Anderson, G.; Breughe, M.; Brooks, D.; Charlebois, M.; et al. MLPerf Inference Benchmark. *Proc. IEEE* **2021**, *109*, 1368–1383.
11. Ji, M.; Chang, Y.; Zhang, B.; Al-Ars, Z. NASH: Neural Architecture Search for Hardware-Optimized Machine Learning Models. *arXiv* **2024**, arXiv:2403.01845. [CrossRef]
12. Ande, J.R.P.K.R.; Khair, M.A. High-Performance VLSI Architectures for Artificial Intelligence and Machine Learning Applications. *Int. J. Recipr. Symmetry Theor. Phys.* **2019**, *6*, 20–30. Available online: https://hal.science/hal-04525631v1 (accessed on 1 December 2025).
13. Yin, S.; Kim, Y.; Han, X.; Barnaby, H.; Yu, S.; Luo, Y.; He, W.; Sun, X.; Kim, J.-J.; Seo, J.-S. Monolithically Integrated RRAM and CMOS Based In-Memory Computing for Efficient Deep Learning. *IEEE Micro* **2019**, *39*, 54–63.

14. Rahman, T.; Mao, G.; Maheshwari, S.; Shafik, R.; Yakovlev, A. MATADOR: Automated System-on-Chip Tsetlin Machine Design Generation for Edge Applications. In Proceedings of the Design, Automation and Test in Europe Conference, Valencia, Spain, 25–27 March 2024.
15. Moolchandani, D.; Kumar, A.; Sarangi, S.R.; Accelerating CNN Inference on ASICs: A Survey. *J. Syst. Archit.* **2021**, *113*, 101887. [CrossRef]
16. Mustafa, D.; Alkhasawneh, R.; Obeidat, F.; Shatnawi, A.S. MIMD Programs Execution Support on SIMD Machines: A Holistic Survey. *IEEE Access* **2024**, *12*, 34354–34377. [CrossRef]
17. Chauhan, A.; Patle, D. Design and Optimization of VLSI Architectures for AI Acceleration. *Int. J. VLSI Des.* **2024**, *10*, 15–22.
18. Qian, C.; Ling, T.; Schiele, G. Exploring Energy Efficiency of LSTM Accelerators: A Parameterized Architecture Design for Embedded FPGAs. *J. Syst. Archit.* **2024**, *152*, 103181. [CrossRef]
19. Shuvo, M.H.; Islam, S.K.; Cheng, J.; Morshed, B.I. Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review. *Proc. IEEE* **2022**, *111*, 42–91. [CrossRef]
20. Meng, J.; Yang, L.; Peng, X.; Yu, S.; Fan, D.; Seo, J.-S. Structured Pruning of RRAM Crossbars for Efficient In-Memory Computing Acceleration of Deep Neural Networks. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 1576–1580.
21. Boutros, A.; Nurvitadhi, E.; Betz, V. Architecture and Application Co-Design for Beyond-FPGA Reconfigurable Acceleration Devices. *IEEE Access* **2022**, *10*, 95067–95082.
22. Que, Z.; Nakahara, H.; Nurvitadhi, E.; Fan, H.; Zeng, C.; Meng, J. Optimizing Reconfigurable Recurrent Neural Networks. In Proceedings of the 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Fayetteville, AR, USA, 3–6 May 2020. [CrossRef]
23. Abu Talib, M.; Majzoub, S.; Nasir, Q.; Jamal, D. A Systematic Literature Review on Hardware Implementation of Artificial Intelligence Algorithms. *J. Supercomput.* **2021**, *77*, 1897–1938. [CrossRef]
24. Capra, M.; Bussolino, B.; Marchisio, A.; Shafique, M.; Masera, G.; Martina, M. An Updated Survey of Efficient Hardware Architectures for Accelerating Deep Convolutional Neural Networks. *Future Internet* **2020**, *12*, 113. [CrossRef]
25. Habibi, M.A.; Han, B.; Saimler, M.; Pavón, I.L.; Schotten, H.D. Towards an AI/ML-driven SMO Framework in O-RAN: Scenarios, Solutions, and Challenges. In Proceedings of the IEEE Future Networks World Forum (FNWF), Dubai, United Arab Emirates, 15–17 October 2024.
26. Gungor, M. AI Optimizations on Intel Architecture (IA); Intel Research Group: 2022. Available online: https://www.cisl.ucar.edu/sites/default/files/2022-02/AI (accessed on 1 December 2025).
27. Badhoutiya, A.; Jaffer, Z.; Hussein, H.M.; Juyal, A.; Mittal, M.; Anand, R. Field Programmable Gate Array: An Extensive Review, Recent Trends, Challenges and Applications. In Proceedings of the 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 28 February–1 March 2024; pp. 1084–1090.
28. He, C.; Huang, Y.; Mu, P.; Miao, Z.; Xue, J.; Ma, L.; Yang, F.; Mai, L. WaferLLM: A Wafer-Scale LLM Inference System. In Proceedings of the 2024 Conference on Machine Learning and Systems (MLSys), Santa Clara, CA, USA, 13–16 May 2024.
29. Wu, N.; Xie, Y. A Survey of Machine Learning for Computer Architecture and Systems. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 54. [CrossRef]
30. Sze, V.; Chen, Y.-H.; Einer, J.S.; Suleiman, A.; Zhang, Z. Hardware for Machine Learning: Challenges and Opportunities. In Proceedings of the IEEE Custom Integrated Circuits Conference, Austin, TX, USA, 30 April–3 May 2017. Available online: https://api.semanticscholar.org/CorpusID:4665260 (accessed on 1 December 2025).
31. Reddy, S.T.A. Evolving CPU Architectures for AI. *Int. J. Sci. Res. (IJSR)* **2022**, *12*, 1238–1242. [CrossRef]
32. Meda, S.; Domazet, E. Advanced Computer Architecture Optimization for Machine Learning/Deep Learning. *CRJ* **2024**, 28–41. Available online: https://crj.cit.edu.al/wp-content/uploads/2014/10/4.pdf (accessed on 1 December 2025).
33. Dhilleswararao, P.; Boppu, S.; Sabarimalai Manikandan, M.; Cenkeramaddi, L.R. Efficient Hardware Architectures for Accelerating Deep Neural Networks: Survey. *IEEE Access* **2022**, *10*, 131788–131828. [CrossRef]
34. Iranmanesh, A.; Sayadi, H. (Eds.) *AI-Enabled Electronic Circuit and System Design: From Ideation to Utilization*; Springer: Cham, Switzerland, 2025. [CrossRef]
35. Singh, R.P. Integration of Reinforcement Learning and Neural Architectures in FPGA Frameworks for Accelerating Semiconductor Innovation and High-Performance VLSI Applications. *Int. J. Comput. Sci. Inf. Technol. Res.* **2025**, *6*, 14–27.
36. Mantovani, P.; Giri, D.; Di Guglielmo, G.; Piccolboni, L.; Zuckerman, J.; Cota, E.G.; Petracca, M.; Pilato, C.; Carloni, L.P. Agile SoC Development with Open ESP. In Proceedings of the 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Virtual Event, USA, 2–5 November 2020.
37. Sangeetha, M.; Ponkanagavalli, M.; Selvi, S.A.M.; Mohammed, M.A. Optimizing Low-Power Device Architectures with Deep Neural Networks for Smart Energy Solutions. *ICTACT J. Microelectron.* **2025**, *10*, 1962–1967. [CrossRef]
38. Guo, K.; Sui, L.; Qiu, J.; Yao, S.; Han, S.; Wang, Y.; Yang, H. Angel-Eye: A Complete Design Flow for Mapping CNN onto Customized Hardware. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, USA, 11–13 July 2016; pp. 24–29. [CrossRef]

39. Guo, K.; Sui, L.; Qiu, J.; Yu, J.; Wang, J.; Yao, S.; Han, S.; Wang, Y.; Yang, H. Angel-Eye: A Complete Design Flow for Mapping CNN onto Embedded FPGA. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *37*, 35–47. [CrossRef]
40. Song, S.; Das, A.; Kandasamy, N. Improving Dependability of Neuromorphic Computing With Non-Volatile Memory. In Proceedings of the European Dependable Computing Conference (EDCC), Munich, Germany, 7–10 September 2020.
41. Wang, C.; Zhang, X.; Cong, J.; Hoe, J.C. Addressing Architectural Obstacles for Overlay with Stream Network Abstraction. In *Proceedings of the 2024 IEEE/ACM International Symposium on Field-Programmable Gate Arrays (FPGA), Monterey, CA, USA, 3–5 March 2024*; IEEE: Piscataway, NJ, USA, 2024.
42. Liu, X.; Chen, Y.; Hao, C.; Dhar, A.; Chen, D. WinoCNN: Kernel Sharing Winograd Systolic Array for Efficient Convolutional Neural Network Acceleration on FPGAs. *arXiv* **2021**, arXiv:2107.04244. [CrossRef]
43. Chen, Y.; Abdelfattah, M.S. BRAMAC: Compute-in-BRAM Architectures for Multiply Accumulate on FPGAs. In *Proceedings of the 2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Marina Del Rey, CA, USA, 8–11 May 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–9. [CrossRef]
44. Li, Z. Investigation of Reconfigurable Hardware Acceleration for Low-Power Embedded Neural Networks. Ph.D. Thesis, Université Côte d'Azur, Nice, France, 2024.
45. Chen, H.; Zhang, J.; Du, Y.; Xiang, S.; Yue, Z.; Zhang, N.; Cai, Y.; Zhang, Z. Understanding the Potential of FPGA-Based Spatial Acceleration for Large Language Model Inference. *arXiv* **2023**, arXiv:2312.15159. [CrossRef]
46. Boutros, A.; Arora, A.; Betz, V. Field-Programmable Gate Array Architecture for Deep Learning: Survey and Future Directions. *arXiv* **2024**, arXiv:2404.10076. [CrossRef]
47. Taka, E.; Huang, N.-C.; Chang, C.-C.; Wu, K.-C.; Arora, A.; Marculescu, D. Systolic Sparse Tensor Slices: FPGA Building Blocks for Sparse and Dense AI Acceleration. In Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, USA, 27 February–1 March 2025.
48. Arora, A.; Ghosh, M.; Mehta, S.; Betz, V.; John, L.K. Tensor Slices: FPGA Building Blocks for the Deep Learning Era. *ACM Trans. Reconfigurable Technol. Syst. (TRETS)* **2022**, *15*, 1–34. [CrossRef]
49. Taj, I.; Farooq, U. Towards Machine Learning-Based FPGA Backend Flow: Challenges and Opportunities. *Electronics* **2023**, *12*, 935. [CrossRef]
50. Reddy, K.V. Accelerating Functional Coverage Closure Through Iterative Machine Learning. *Int. J. Comput. Appl. Inf. Technol.* **2024**, *7*, 1401–1411.
51. Camacho, P.J. A Comprehensive Literature Review of Artificial Intelligence (AI) and Machine Learning (ML) Hardware Implementation and Architectures. Ph.D. Thesis, San Francisco State University, San Francisco, CA, USA, 2024.
52. Nayak, G.N. Electronics Manufacturing Testing: A Technical Deep Dive. *Int. J. Comput. Eng. Technol.* **2024**, *16*, 253–262. [CrossRef]
53. Namboothiripad, M.K.; Vadhyan, G. Efficient Implementation of Artificial Neural Networks on FPGAs Using High-Level Synthesis and Parallelism. *Int. J. Adv. Technol. Eng. Explor.* **2024**, *11*, 1497. [CrossRef]
54. Boutros, A.; Betz, V. FPGA Architecture: Principles and Progression. *IEEE Circuits Syst. Mag.* **2021**, *21*, 4–29. [CrossRef]
55. Khaddam-Aljameh, M.R. Computational Memory Design. Ph.D. Thesis, ETH Zurich, Zurich, Switzerland, 2024.
56. Arora, A.; Boutros, A.; Damghani, S.A.; Mathur, K.; Mohanty, V.; Anand, T.; Elgammal, M.A.; Kent, K.B.; Betz, V.; John, L.K. Koios 2.0: Open-Source Deep Learning Benchmarks for FPGA Architecture and CAD Research. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 3895–3909. [CrossRef]
57. Menghani, G. Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. *ACM Comput. Surv.* **2023**, *55*, 1–37. [CrossRef]
58. Soldavini, S.; Pilato, C. Compiler Infrastructure for Specializing Domain-Specific Memory Templates. *arXiv* **2021**, arXiv:2104.01448. [CrossRef]
59. Xiao, Q.; Liang, Y.; Lu, L.; Yan, S.; Tai, Y.W. Exploring Heterogeneous Algorithms for Accelerating Deep Convolutional Neural Networks on FPGAs. In *Proceedings of the 54th Annual Design Automation Conference (DAC), Austin, TX, USA, 18–22 June 2017*; ACM: New York, NY, USA, 2017; pp. 1–6.
60. Chowdhury, S.; Grimaldi, A.; Aadit, N.A.; Niazi, S.; Mohseni, M.; Kanai, S.; Ohno, H.; Fukami, S.; Theogarajan, L.; Finocchio, G.; et al. A Full-Stack View of Probabilistic Computing with p-bits: Devices, Architectures and Algorithms. *IEEE J. Explor.-Solid-State Comput. Devices Circuits* **2023**, *9*, 1–11. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.