Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 2, No. 3, 2025 (pp. 47-55)

RESEARCH ARTICLE ECEJOURNALS.IN

AI-Integrated System-on-Chip (SoC) Architectures for High-Performance Edge Computing: Design Trends and Optimization Challenges

K. Maidanov[1]*, Jeon Sungho[2]

[1]*Department of Electrical and Computer Engineering, Ben-Gurion University, Beer Sheva, Israel*

[2]*Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, Korea*

KEYWORDS:
AI-SoC Architectures,
Edge AI Computing,
Neural Processing Units (NPUs),
Heterogeneous System-on-Chip,
Low-Power Inference,
Hardware-Software Co-Design,
Embedded AI Accelerators,
Thermal-Aware SoC Design,
Real-Time AI Processing,
Energy-Efficient AI Hardware

ARTICLE HISTORY:
Submitted : 07.04.2025
Revised : 16.05.2025
Accepted : 13.07.2025

https://doi.org/10.31838/JIVCT/02.03.06

Abstract

This leads to the fact that the development of edge applications in autonomous systems, healthcare, and smart environments requires very efficient and scalable computing frameworks. The paper gives an overview of the state-of-the-art of AI-integrated System-on-Chip (SoC) architectures being tailor-made to satisfy performance, energy, and latency requirements of the modern edge computing. This is aimed at examining how the embedded AI accelerators (e.g. neural processing units (NPUs) and digital signal processors (DSPs)) may be easily implemented in heterogeneous SoC platform. Methodologically, the research analyzes the progress of the last few years in the field of hardware-software co-design, dataflow and memory hierarchies. It also looks into heterogeneous core coupling plans, thermally conscious floor planning, and energy-conscious task scheduling. Comparative lessons based on commercially available SoCs such as Apple ANE, Google Edge TPU, and NVIDIA Jetson are given in an attempt to highlight trade-offs in the real world. Indicators demonstrate that AI-tailored accelerations--including systolic array-based accelerators, near-memory computing and quantization-aware processing--are representative of multi-seemingly magnified inference acceleration and power efficiency. But the problems encountered are scaling the memory bandwidth, real-time workload scheduling, as well as thermal dissipation. It is observed in conclusion of the paper that the future AI-SoC architectures should focus on being modular, reconfigurable, and secure with runtime and power profiles that allow them to be used at the edge. This overview lays out helpful design considerations and outlines every major research area of projected work on the next-generation SoCs with a means to sustain, intelligent computing at the edge.

Author e-mail: maidanov.k@gmail.com, sun.Jeon@snu.ac.k

How to cite this article: Maidanov K, Sungho J. AI-Integrated System-on-Chip (SoC) Architectures for High-Performance Edge Computing: Design Trends and Optimization Challenges. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 2, No. 3, 2025 (pp. 47-55).

Introduction

One of the areas where edge computing is changing the world is autonomous vehicles, smart manufacturing, the industrial IoT and digital healthcare because it upends traditional ways of doing big things with real-time, real-time localized data processing and minimal latency and marginal reliance on cloud infrastructure. Edge computing can relieve the piggyback effect on the bandwidth and increase the privacy and responsiveness of mission-sensitive applications by placing intelligence closer to the data sources themselves. Nevertheless, modern artificial intelligence (AI) workloads, particularly deep learning models, are computationally very greedy and are a very challenging load to the older edge devices, which are limited by power, area, thermal design limits.

In order to satisfy those needs, System-on-Chip (SoC) architectures are in the process of paradigm shift focusing on heterogeneous integration of Accelerators AI, perhaps by incorporating Neural Processing Unit (NPU), Digital Signal Processors (DSP) and reconfigurable logic .[16] These united platforms are predictable to deliver high-performance inference, low-power consumption, and small enclosures to be set up at the edge. In this paper the new design techniques and optimization issues in

integrating AI cores with SoCs are discussed with focus on progress in hardware-software codesign, memory hierarchy control and AI-specific dataflow designs.

Although previous studies have paid attention to the acceleration of AI scenarios in data centers and mobile platforms, current research tends to be narrow in terms of viewpoint specific to edge computing peculiarities, including fluctuating workloads, thermal pressure, and on-demand scheduling of tasks. Moreover, little has been accomplished in harmonising architectural patterns across commercial AI-SoCs and determine design trade-offs in terms applicable to high-performance edge-computing applications .[1, 2]

In this paper the gaps will be filled by:

- Surveying a state of the art in AI-integrated SoC designs;
- Interpreting their performance, power and scalability properties;
- Calling attention to the important issues of AI-core and memory access as well as thermal-aware layout;
- Giving Design ideas and trend of next-generation edge AI hardware.

Background and Related Work

This rapid increase in the artificial intelligence (AI) apps at the edge, including object detection, speech recognition, and anomaly classification, has put extraordinary pressure on on-device computation [17]. The Deep Neural Networks (DNNs) such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based networks are considered to be high compute-intensive and memory bandwidth-demanding, which do not get easily captured by the traditional SoC architectures initially.

In a bid to overcome these constraints, more recent generations of System-on-Chip (SoC) there, alternatively perform a heterogeneous architecture which integrates dedicated AI accelerators in their systems. Typical ways to go about integration are:

- Neural Processing Units (NPUs): NPU-style: Designed to do significant amounts of matrix operations typical of DNN inference workloads (eg. Apple ANE, Huawei Ascend);
- Digital Signal Processors (DSPs): Best used in fixed-point, low latency signal processing real-time applications;
- GPUs: Very parallel processing cores that can be used in training and inference of deep networks;
- Reconfigurable FPGA fabric: Linkages that allow flexibility of application specific acceleration and hardware control of low-level.

They have relatively high performance by using dataflow-related architecture (e.g., systolic arrays), hardware-adapted quantization, and tiling to minimize the DRAM access and enhance on-board usage of memory.[3, 4]

In spite of these developments, a number of challenges are still to Deploy AI workloads efficiently at the edge:

- Memory Bottlenecks: Large DNN models refer to the case when the capacity of on-chip memory becomes inadequate, which in turn gives rise to extensive memory access happening on off-chip and the energy penalties.[5]
- Thermal Limits: Computational density in small edge devices means that there is a local hot spot, which impacts reliability and performance limiting.[6]
- Heterogeneous Resource Management: Computation near different CPUs, NPUs and GPUs requires run time task scheduling and data coherence protocols which are in their infancy.[7]
- Disunity of Toolchains: Cross-compiler support and other software that are AI specific (e.g., TensorFlow Lite, ONNX Runtime) have to be specific to each SoC, making them more difficult to develop and less portable.

Additionally, most of the existing research is usually concerned with the benchmarking of performance and this has neglected some important design trade-offs which include reconfigurability, power gating, modular AI core engineering, secure AI execution. These factors make the image in the edge of AI scalable and secured.

This paper overcomes these gaps by discussing the cross-layer optimization strategies, by looking at their commercially deployed AI-SoCs, and show future research directions to bridge these gaps.

AI-Integrated SoC Architecture Overview

The design of AI-enabled System-on-Chip (SoC) systems is locally adapting quickly to the requirements of low delay, precise time, and power-based edge processing. The standard AI-SoC architecture involved very closely coupled components which combined to support control logic, deep learning inference, memory management, and communication. The section explains the basic building blocks and how they are integrated to achieve performance of AI processing on-chip.

Core Components

AI-SoCs are heterogeneous platforms, which means that they consist of several specialized processing units, optimized to different computational tasks (Figure 1: Core Components of AI-Integrated System-on-Chip (SoC) Architectures). These usually involve control logic and system coordination consisting of general-purpose CPUs; high-throughput deep learning processing (AI accelerators) comprising NPUs or TPUs; memory hierarchy to support low-latency access to data; and high-bandwidth interconnects, like Network-on-Chip (NoC) to support efficient movement of data among components.[15]

- General-purpose CPUs: These cores (usually of the ARM Cortex-A/R families or, RISC-V) handle the control flow, system coordination, lightweight preprocessing and task scheduling. Although they are not yet optimized in regard to large-scale matrix operations, flexibility and support of instruction make them an excellent choice in terms of running operating system kernels, input/output drivers, and non-parallel workloads.
- AI Accelerators (e.g., NPUs, TPUs): the key components to run AI workloads are: Neural Processing Units (NPUs), or Tensor Processing Units (TPUs). They are also designed to support inequality gates, high throughput matrix multiply (e.g. systolic arrays), low precision arithmetic (e.g. INT8, bfloat16), and parallel convolution layers. Their instruction set can typically be AI focused and enable direct execution of deep neural network (DNN) functions like convolutions, activations and normalization.
- Memory Hierarchies: High memory bandwidth and low latency are required to support the efficient processing done by AI. AI-SoCs have multiple layers of memory: on-chip SRAM as fast cache, L1/L2 caches to exploit temporal/spatial locality and occasionally shared, unified memory between the CPU and accelerators. The burst-access DMA, compression and intelligent prefetching improves the memory bandwidth.
- Interconnects: Network-on-Chip (NoC) Architectures enable communication among the cores and the memory and give a degree of scalability, high bandwidth, and low latency. NoCs vary in topology and can be based on bus, crossbar, or mesh architecture, but they all aim at connecting the CPUs, NPUs, the memory, and peripherals in a very efficient manner. AI workloads receive priority by means of quality of service (QoS) mechanisms.

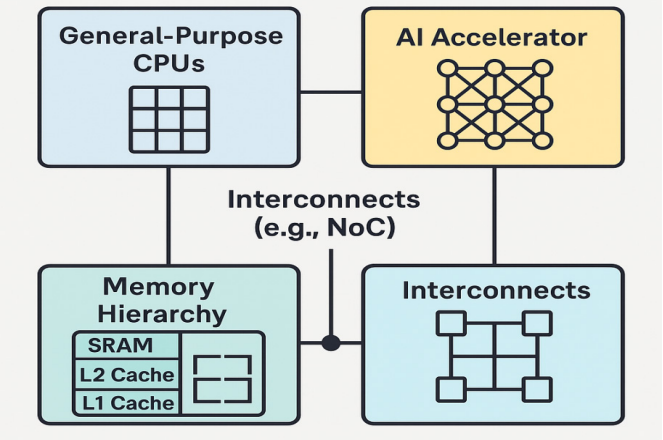

Fig. 1: Core Components of AI-Integrated System-on-Chip (SoC) Architectures

The figure shows the four key building blocks of AI-integrated SoC platforms: General-Purpose CPUs are used to provide control logic as well as system-level coordination, AI Accelerators (e.g. NPUs, TPUs) enable parallel deep learning inference, hierarchical memory subsystems enable low-latency access (including SRAM and cache layers), and interconnection systems such as Network-on-Chip (NoC) are required to provide high-bandwidth, low-latency communications between cores.

Heterogeneous Integration Strategies

Modern AI-SoC systems utilize a variety of integration tactics to control energy, scalability and system intricacy as displayed in Figure 2: Heterogeneous Integration Strategies for AI-Integrated SoC Architectures. The featured strategies are a tightly coupled or laxly affiliated arrangement of accelerators and the advanced mixture of chiplets and co-packaged-memory as of further advancement, in addition to run-time response plans such as dynamic scaling of voltages and frequencies (DVFS).

- Tightly and Loosely Coupled Accelerators: In tightly coupled AI accelerators the accelerator is integrated into the same cache and memory hierarchy with the CPU (e.g. Apple Neural Engine), with very low-latency between the AA and the CPU. More loosely coupled designs correspond to embedded FPGA systems where AI cores are at least physically isolated by dedicated buses or DMA channels and may be more modular, but at the expense of a larger communication overhead.[14]

- Co-Packaged Memory vs. Chiplet-Based Modularity: High-performance SoCs can also consist of co-packaged HBM (High Bandwidth Memory) based on 2.5D interposer or dimensionally modular chiplets (computing and memory tiles are integrated via high-performance advanced packaging (e.g., Intel Foveros, AMD Infinity Fabric). This will allow scaling and yield increase and accommodate large AI models.
- Dynamic Voltage and Frequency Scaling (DVFS): Dynamic scaling at run time is essential in edge settings in which workloads and thermal requirements vary. DVFS controllers are included in AI-SoC clusters and switch between voltage-frequency pairs of CPUs, AI sores and interconnects to balance power efficiency with application power requirements. Power gating/ clock gating is also deployed within more advanced SoCs to turn off the inactive modules.

By working together these architectural features allow AI-integrated SoCs to provide intelligent edge applications with responsiveness in real-time and energy efficiency levels. The nature of edge applications is becoming more varied as users seek to identify new uses; with SoC architecture continuing to shift towards being more reconfigurable, domain specific optimized and secure designed structures.

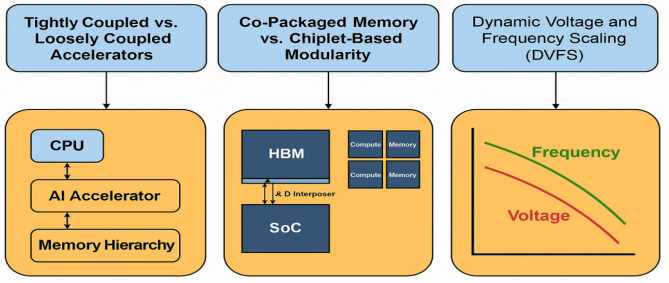

Fig. 2: Heterogeneous Integration Strategies for AI-Integrated SoC Architectures

Design Trends in AI-SoCs

Evolution of AI-incorporated SoC architectures is defined by a number of revolutionary design fads, which are intended to overcome the shortcomings of classic edge computing systems (Table 1). The trends are all centered towards easing compute efficiency, scalability, energy performance and model compatibility with resource constraints on compact and power-sensitive environments.

Optimization Challenges

In spite of the imposing competencies of AI-adapted SoC platforms, a series of core optimization problems interfere with their capabilities in edge settings. These issues cut across thermal management, structure of

Table 1: Emerging Design Trends in AI-Integrated System-on-Chip (SoC) Architectures

Trend	Description
Heterogeneous Computing	Modern AI-SoCs are designed with a diverse mix of processing units, including general-purpose CPUs, AI-specific NPUs, GPUs for parallel processing, and DSPs for signal-intensive tasks. This heterogeneous integration enables task-specific acceleration, improves overall throughput, and allows for concurrent processing of control, signal, and inference workloads within the same chip. It also supports modular system design for flexible deployment in varied edge applications [8].
Dataflow Optimization	With DNNs demanding high arithmetic intensity, AI-SoCs leverage systolic array architectures and SIMD (Single Instruction, Multiple Data) pipelines to minimize data movement and maximize throughput. These architectures exploit spatial data reuse and temporal parallelism, significantly improving inference efficiency in matrix-heavy operations such as convolutions and attention mechanisms [9].
Edge-AI Framework Compatibility	To support a wide range of real-time edge applications, AI-SoCs now offer native compatibility with popular inference frameworks like TensorFlow Lite, PyTorch Mobile, and ONNX Runtime. Moreover, these frameworks are often tightly integrated with real-time operating systems (RTOS) and vendor-specific SDKs, enabling seamless deployment of optimized models on constrained hardware [10].
Hardware-Aware Neural Architecture Search (NAS)	Neural architecture search algorithms are increasingly being customized to account for SoC-level constraints such as memory hierarchy, compute bandwidth, and latency budgets. Hardware-aware NAS tools co-optimize model structure and deployment footprint, enabling the development of highly efficient, platform-specific AI models suitable for edge inference [11].
In-Memory and Near-Memory Computing	A major bottleneck in AI inference is the energy cost associated with frequent memory access. Emerging SoC designs incorporate processing-in-memory (PIM) and near-memory compute engines that bring computation closer to data storage units. This trend reduces off-chip communication, lowers energy consumption, and supports higher model density within a compact silicon footprint.[12]

memory hierarchy, software infrastructure, and system security with the issue of the overall design that presents reliable, real-time, and energy-efficient AI computation (Figure 3: Optimization Challenges in AI-Integrated System-on-Chip (SoC) Architectures).

Thermal and Power Constraints

AI tasks are almost always compute-intensive and they tend to produce high thermal loads, particularly when running at the edge on small devices, such as wearables, drones, and self-driving robots. The edge devices do not have active cooling of server-grade systems and thermal management should be a priority. The reliability may be degraded after exposure to high temperatures, performance throttle may occur, and it hastens device aging.[13]

To handle this, contemporary AI-SoCs employ power gating and clock gating where the idle functional blocks are selectively disabled. As well, performance adaptation to thermal conditions is achieved by low-leakage transistor technologies, dynamic voltage and frequency scaling (DVFS). Thermal-aware floorplanning and thermal sensors added during run-time also facilitate dealing with local hotspots.

Memory Bandwidth Bottlenecks

Large-scale CNNs and transformers based on AI models need extremely huge data transfer between compute units and memory. Conventional memory hierarchies built in SoC devices may not be able to support this bandwidth effectively leading to deteriorated performance and high power usage associated with a high number of accesses to off-chip memory.

In doing that, multi-bank on-chip SRAM, and high-efficiency DMA (Direct Memory Access) engines and data compression solutions are used in reducing the memory latency and traffic congestion. Moreover, AI accelerator attributes such as tiling and buffer reuse strategies improve the benefit of spatial and temporal data locality. Nevertheless, the rise in model complexity still is exerting pressure on the design of the on-chip memory, necessitating new near-memory computing architectures.

Software Stack and Toolchain Support

Heterogeneous design of AI-SoCs, having more than one CPU, GPU, NPU, and DSP, necessitates a complex software stack that can effectively partition tasks, schedule resources, and coordinate cross-core synchronization. The limitations imposed on compilation-level optimizations at the hardware level are availability of instruction set and memory bandwidth as well as parallel execution patterns.

But most toolchains are simply not mature and not platform-independent nowadays. Vendors experience difficulties deploying models that are optimized using TensorFlow Lite, ONNX or vendor-specific compiler tasks; especially in ones oriented around low-precision arithmetic (e.g. INT8, bfloat16). Moreover, AI-SoCs debugging and profiling are not unified, and they hinder model optimization and power-aware scheduling.

Security and Privacy

When the inference process AI shifts towards the edges, it is crucial to guarantee a safe execution process, in particular, in the healthcare, financial, and surveillance spheres. To outsmart malicious hacks, SoCs should include secure boot, hardware root-of-trust, and encrypted memory acess on chip.

Moreover, there should be logical separation of AI accelerators in order to reduce the toll of the side-channel and adversarial attacks. Such methods as trusted execution environments (TEEs), hardware-level watermarking, access control enforcement are also becoming part of SoC design. But such trade-offs between security enforcement and real-time performance has been an ever-present challenge that necessitated co-design of both hardware and firmware layers with great care.

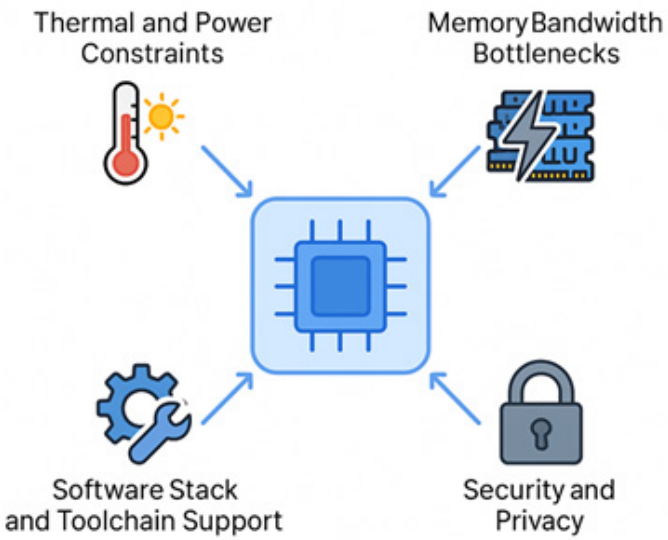

Fig. 3: Optimization Challenges in AI-Integrated System-on-Chip (SoC) Architectures

The four key bottlenecks in current AI-SoC designs are indicated in this diagram: thermal/power and power, memory bandwidth, software stack/toolchain, and security/privacy. Every challenge is graphically described with the focus on a central AI-SoC to show

their significance to performance, scalability and the ability to run in real-time in edge computing applications.

Case Studies

In-the-wild use of AI-integrated SoCs shows off the variety of architectural solutions that have gone into achieving performance-oriented, power-efficient, and application-specific tradeoffs. Three of the typical platforms that have been described in Table 2: Comparison of AI-Integrated System-on- Chip (SoC) Architectures are presented in the section following a concise summary of each of them.

Google Edge TPU

Google Edge TPU is an ultra low power AI accelerator, optimized to run 8-bit quantized inference. On-chip in the Coral Dev Board, it provides high-performance CNNs with low power consumption, which fits in IoT enabled decision-making applications. It has low-latency execution because of its tightly coupled architecture with local memory that fits in power-constrained edge environments.

Apple Neural Engine (ANE)

The Apple ANE included in A-series and M-series SoCs provides high TOPS throughput on-device deep learning applications, such as Face ID, ARKit and camera AI. It allows smoothing the AI workloads as it shares memory with the CPU and GPU cores. Apple full-stack tool chain (e.g. Core ML) can be used to deploy and convert models that are optimized to the ANE.

NVIDIA Jetson Orin

Jetson Orin is based on an ARM Cortex-A based NVDLA accelerator and Ampere GPU to provide high-performance edge AI in the range of up to 200 TOPS. It enables robotics and autonomous systems with support of mixed-precision inference (INT8 to FP32). Full support of software (JetPack, TensorRT) allows the implementation of AI applications in real-time multi-sensor scenarios to be efficient.

Discussion

Adding AI accelerators to the System-on-Chip (SoC) layout has transformed the edge compute computation paradigm by permitting reaction processing, energy-efficient inference, and smaller scale deployment. This section is a synthesis of the insight in the architecture as well as the trade offs of current design and the deployment bottlenecks and emerging research problems that are important in the design of the next generation of edge AI platform.

Design Trade-offs and System Bottlenecks

0CPU and memory on the same silicon die plus AI acceleration core co-location delivers even higher performance/watt characteristics. Nonetheless, such degree of integration implies architectural trade-offs that should be properly balanced:

- Power vs. Performance: High-throughput AI cores, like NPUs and DSPs, provide a tremendous boost to inference latency and many tend to cause thermal hotspots and higher dynamic power in localized areas. To reduce the thermal effect, methods, such as dynamic voltage and frequency scaling (DVFS) and adaptive task scheduling, have been developed that reduce the thermal effect but enhance the complexity of system control and software stack reliance.
- Memory Bandwidth in Area Constrained Environments: AI loads mostly CNNs and transformers, can require sustained high bandwidth data transport. Off-chip SRAM doubling of on-chip SRAM or adding high-bandwidth memory (HBM) ameliorates the bottleneck, and causes extra chip area, leakage power, and routing overhead. To deal with this, current SoCs have incorporated systems of systolic arrays, loop tiling and on-chip data compression in order to minimize off-chip memory tasking.
- Usability: Flexibility vs. Specialization Flexible domain-general accelerators such as TPUs are designed to achieve the best possible performance but may be limited to the ability

Table 2: Comparison of AI-Integrated System-on-Chip (SoC) Architectures

SoC Platform	AI Performance	AI Precision	Memory	Primary Use Case
Google Edge TPU	4 TOPS	INT8	8 GB LPDDR4	Embedded Vision, IoT
Apple Neural Engine	11 TOPS	INT8 / FP16	Unified (Shared)	On-Device AI (Face ID, AR)
NVIDIA Jetson Orin	Up to 200 TOPS	INT8 / FP16 / FP32	32 GB LPDDR5	Robotics, Autonomous Systems

to support changing AI model topologies. Conversely, reconfigurable-logic (e.g., FPGA overlays) based or chiplet modularity designs provide flexibility, but at an increased design burden and an increased resource cost.

AI Workload Diversity and Heterogeneity Management

Multitasking is also becoming more and more critical: Edge devices are expected to multimodal: vision, audio, control, sensor fusion, and often with fine latency constraints. This pushes the demand of a heterogenous computing building block in the SoCs, like a tightly integrated CPU+GPU+NPU complex. To distribute efficient workload, it is necessary:

- Task offloading that is aware of the workload, where the workload of high latency is allocated to NPUs, and low latency logic to this side is on CPUs.
- Homogeneous stacks such as TensorFlow Lite, ONNX Runtime and vendor SDKs (e.g. NVIDIA TensorRT, ARM Ethos-U) that allow models to be converted and scheduled, and parallel kernels to be optimized in the same way.

Inter-core data coherency, context switching latency, and symmetric memory access lag behind, although problems with latency in AI models are less threatening as they become more deep and branched.

Thermal and Reliability Considerations

Edge applications AI-SoCs are more likely to be applied in tight housing with no active thermal management, including wearable, drone and automotive applications. These are thermally constrained environments and they demand:

- Thermal-trained floorplanning to allocate blocks efficiently that are heat-sensitive.
- Connection of microfluidic channels or thermal through-silicon vias (TSVs) in 2.5D/3D SoC stacks in order to improve the vertical heat removal.

It is also major when it comes to long-term reliability. Thus, redundancy-aware accelerators, real-time fault detection and Error-Correcting Code (ECC) memory must be incorporated into the SoC to address environmental stress, wear-out, and soft errors.

Real-World Deployment Insights

The design of commercial AI-SoCs embodies design philosophies that are custom to the application they are built to suit:

- Apple Neural Engine (ANE) focuses on close work with the operating system and low-latency AI inference in on-device activities, such as Face ID and augmented reality programs.
- NVIDIA Jetson Orin is high-performance yet scalable compute, with mixed-precision and heterogeneous cores, optimized with robotics, edge server, and autonomous navigation in mind.
- Google Edge TPU is an example of ultra-low-power-AI inference with stringent INT8 model support of machine vision use cases and always-on AI.

These deployments emphasize the value of model-architecture co-designs, in which hardware is deliberately tailored to model properties (e.g. convolution-heavy networks vs. transformer-based networks).

Emerging Paradigms and Research Opportunities

AI-SoC development in the future will depend on how the new constraints will be discussed and how the adaptability will be widened. Potentially there might be developments in the following directions:

- In-Memory Computing (IMC) and Processingin Memory (PIM) to minimize latencies and power expenses of data flow between logic and storage.
- Hardware-Aware Neural Architecture Search (NAS) in order to automatically design models that take into account the constraints of the SoC (e.g. cache size, quantization boundaries).
- Real-time AI-Driven Reconfiguration, supporting the real-time optimisation of logic blocks in accordance with different tasks, work load, or temperature constraints.
- SoC architectures that put security first and incorporate the Trusted Execution Environment (TEE), hardware-based AI model watermarking and secure boot capabilities, to ensure inference integrity and IP ownership.

The examined literature on case studies and architecture trends show that AI-augmented SoCs are key to successful edge intelligence development, however, realizing scalable, safe, efficient architectures needs a top-down system-level approach. The combination of AI model complexity, flexibility in edge deployment scenarios and performance under real-time responsive use cases requires an integrated system that spans hardware architecture, software optimisation, and collection system control. Overcoming the described challenges and adopting the new paradigms of AI-SoC development will prove to be instrumental in determining the future generation of systems.

Conclusion and Future Directions

The paper provided a detailed overview of AI-accommodating System-on-Chip (SoC) systems and focus on integration methodologies, design trends, optimization issues, and field adoption of edge computing systems. It described how co-integration of AI accelerators, like NPUs, GPUs and DSPs with general purpose CPUs, and the application of optimized memory hierarchies will deliver energy-efficient, inference performance in real time for a set of diverse edge workloads.

Among the most notable contributions of this paper, one should mention:

- Organised description of fundamental components of an AI-SoC, compute, memory, and connect components, and their position in context relevant to modern implementations of edge applications.
- An overview of design trends, e.g. dataflow optimization, in-memory computing and hardware-aware NAS that represent direction in the industry towards scalable, adaptive architectures.
- Complex study and discussion on optimization issues with regard to thermal constraints, bandwidth, the fragmentation of tools in the toolchain, and the security issues.
- Informative feedback of commercial platforms such as Google Edge TPU, Apple Neural Engine, and NVIDIA Jetson Orin, they are prominently featured in the case studies with distinct approaches in actual AI-SoC deployment.

Even as there are constant improvements, latest AI-SoC platforms suffers restriction in memory bandwidth, power efficiency, and heterogeneous software stack heterogeneity. Closing these gaps would need cross-layer innovation, spanning architecture, compilers, system software, and model optimization pipeline.

Future Directions

In order to make the most out of intelligent edge systems, the following areas will be addressed in the future research and development:

- Chiplet-Based Modular SoCs: Suggesting scalability and flexibility of design due to the integration of chiplets that allow upgrades and specialization of AI workloads to cost effectively be upgraded in stages.
- Hardware Reliability with AI Enhancements: Dynamic machine learning models infused into failure analysis, dynamic thermal throttling and intelligent aging compensation to manage long term performance.
- Quantum-Inspired SoC Accelerators: Investigtarion of quantum-inspired algorithms and annealing-based hardware to accelerate the execution of complex optimization problems in the edge with minimal energy costs.
- Unified AI-SoC Frameworks: Creation of end-to-end application software ecosystems that are contiguously interconnected with compilers, tool chains, firmware and runtime schedulers to achieve maximum performance on heterogeneous compute engines.

With the ongoing increase in complexity and usages of AI workloads, the co-design of intelligent, secure and thermally-aware SoCs will be the key driver in defining the future of autonomous systems, healthcare diagnostics, industrial automation and so on. The combination of the architecture level and AI powered flexibility offer a great potential horizon to the next generation edge AI processor.

References

1. Jain, A. K., Sinha, S., & Kumar, R. (2023). Design considerations for AI-enabled SoCs in edge computing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42*(2), 345–358. https://doi.org/10.1109/TCAD.2022.3208124
2. Lee, Y., Park, K., & Pedram, M. (2023). Energy-efficient hardware architectures for deep learning on edge devices: A survey. *IEEE Transactions on Circuits and Systems I: Regular Papers, 70*(5), 1842–1856. https://doi.org/10.1109/TCSI.2023.3241217
3. Jouppi, N. P., et al. (2017). In-datacenter performance analysis of a tensor processing unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*, 1–12. https://doi.org/10.1145/3079856.3080246
4. Yazdanbakhsh, A., et al. (2021). Edge AI hardware: Benchmarking and co-design. *IEEE Micro, 41*(5), 48–56. https://doi.org/10.1109/MM.2021.3095932
5. Li, H., Wang, K., & Pedram, M. (2022). DRAM power-aware DNN acceleration for edge devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41*(4), 1033–1044. https://doi.org/10.1109/TCAD.2021.3107609
6. Yu, L., Lin, Y., & Zhang, Z. (2021). Thermal-aware design and management for AI edge SoCs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29*(11), 2047–2060. https://doi.org/10.1109/TVLSI.2021.3094847
7. Chen, X., Venkataramani, S., & Esmaeilzadeh, H. (2020). Eyeriss v2: A flexible accelerator for emerging deep neural

networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 10*(4), 417–430. https://doi.org/10.1109/JETCAS.2020.3035155

8. Devgan, A. S., Kumar, R., & Pedram, M. (2023). Heterogeneous integration in edge-AI SoCs: Trends and performance trade-offs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42*(7), 1552–1565. https://doi.org/10.1109/TCAD.2023.3249217
9. Chen, H., Liu, Y., & Zhou, J. (2023). Design of systolic arrays for low-power CNN inference in edge devices. *IEEE Transactions on Circuits and Systems I: Regular Papers, 70*(3), 892–905. https://doi.org/10.1109/TCSI.2023.3251902
10. Patel, M., Wang, L., & Dey, S. (2023). Optimizing edge AI inference with framework-aware SoC design. *IEEE Embedded Systems Letters, 15*(2), 101–104. https://doi.org/10.1109/LES.2023.3258887
11. Zhang, B., Tan, A., & Reddi, V. (2023). Efficient neural architecture search for edge AI: Challenges and solutions. *IEEE Transactions on Neural Networks and Learning Systems, 34*(1), 123–136. https://doi.org/10.1109/TNNLS.2022.3189731
12. Kim, Y., Lee, K., & Park, H. (2023). Near-memory computing strategies for energy-efficient deep learning on edge SoCs. *IEEE Journal of Solid-State Circuits, 58*(4), 1087–1099. https://doi.org/10.1109/JSSC.2023.3244309
13. Cheng, L. W., & Wei, B. L. (2024). Transforming smart devices and networks using blockchain for IoT. Progress in Electronics and Communication Engineering, 2(1), 60–67. https://doi.org/10.31838/PECE/02.01.06
14. Choset, K., & Bindal, J. (2025). Using FPGA-based embedded systems for accelerated data processing analysis. SCCTS Journal of Embedded Systems Design and Applications, 2(1), 79–85.
15. Christian, J., Paul, M., & Alexander, F. (2025). Smart traffic management using IoT and wireless sensor networks: A case study approach. Journal of Wireless Sensor Networks and IoT, 2(2), 45-57.
16. Alwetaishi, N., &Alzaed, A. (2025). Smart construction materials for sustainable and resilient infrastructure innovations. Innovative Reviews in Engineering and Science, 3(2), 60–72. https://doi.org/10.31838/INES/03.02.07
17. Schmidt, J., Fischer, C., & Weber, S. (2025). Autonomous systems and robotics using reconfigurable computing. SCCTS Transactions on Reconfigurable Computing, 2(2), 25–30. https://doi.org/10.31838/RCC/02.02.04