Proceedings of CONF-FMCE 2025 Symposium: Semantic Communication for Media Compression and Transmission
DOI: 10.54254/2755-2721/2025.GL39453

Applications and Trends of FPGA–SoC Integration for Low-Power Embedded Systems: A Literature Review

Zhoutong Chen

Department of EEE, University of Sheffield, Sheffield, UK
zchen264@sheffield.ac.uk

Abstract. This literature review examines the current stage of Field Programmable Gate Array-System-on-Chip (FPGA-SoC) integration technology applications for low-power embedded computing. Modern platforms such as the Xilinx Zynq, AMD Versal ACAP, and Microchip's PolarFire SoC demonstrate the power-to-performance advantages of tightly coupling programmable logic with processing subsystems. The advantages of tightly coupling programmable logic with processing subsystems provide a better performance in terms of energy efficiency than traditional discrete systems. In critical low-power application scenarios such as edge AI, smart sensing, and IoT gateways, such systems are capable of real-time computation while meeting energy consumption constraints. Energy-efficient design techniques discussed in the paper include hardware-software co-design, dynamic voltage and frequency regulation (DVFS), clock and power gating, and AI-assisted automated design. The review further explores future directions such as advanced heterogeneous integration (e.g., core-grain technology with 3D ICs), embedded FPGA IP, and AI-driven Design Space Exploration (DSE), highlighting the key position of FPGA-SoC platforms in the next generation of low-power embedded systems.

Keywords: FPGA–SoC integration, low-power embedded systems, hardware/software co-design, dynamic voltage scaling

1. Introduction

Energy efficiency has become a core constraint in modern embedded system design, especially in the context of the increasing popularity of edge computing, low-power devices, and artificial intelligence applications. Traditional architectures rely on discrete CPUs and FPGAs, which often face communication bottlenecks, high latency, and power inefficiencies. Integrating field-programmable gate arrays (FPGAs) with system-on-chips (SoCs) provides a new solution for realizing low-power and high-performance computing. By integrating reconfigurable logic and embedded processors on the same chip, FPGA-SoC platforms effectively eliminate off-chip communication latency and allow for more granular control of computing resources and power consumption.

In recent years, the technology is becoming more mature and diverse with the development of commercial platforms including AMD Versal ACAP, Xilinx Zynq UltraScale+ and Microchip PolarFire SoCs. These platforms support heterogeneous computing, power-aware task scheduling,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.GL39453

and hardware acceleration for energy-constrained but performance-demanding applications, such as smart sensing, industrial automation, real-time signal processing, and edge AI.

Empirical studies have shown that in complex vision processing pipelines built on platforms such as the ZCU102, FPGA implementations can achieve efficiency gains ranging from 1.2× to 22.3× in terms of energy consumption per frame compared to GPU or CPU systems [1]. In addition, FPGA gas pedals built through hardware-level customization can achieve up to 8.3× throughput improvement and up to 75× energy efficiency advantage over high-end GPUs when performing binary convolutional neural network (BNN) inference in low-volume or real-time applications [1].

This review will focus on the development direction of FPGA-SoC integration technology in energy-efficiency optimized embedded systems, systematically sort out its core architectural features, low-power design strategies and typical application cases, and summarize the current research trends and cutting-edge technologies, to further reveal the key roles of FPGA-SoCs in the next-generation embedded systems. The key role of FPGA-SoCs in next-generation embedded systems will be further revealed.

2. Background

FPGA–SoC devices integrate a hard processing system (PS)—typically CPU cores, memory controllers, and peripherals—with programmable logic (PL) on a single die, enabling high-bandwidth, low-latency coupling through on-chip interconnects such as AXI or a network-on-chip (NoC). In terms of power consumption, total power consumption is typically considered to be the sum of dynamic power consumption and static or leakage current power consumption: dynamic power consumption can be approximated as

$$P_{dyn} \approx \alpha C V^2 f$$

Where α is the switching activity factor, C is the effective capacitance, V is the supply voltage, and f is the clock frequency and the static power consumption is denoted as P_{leak}. Therefore, energy efficiency improvements depend on architectural locality (minimizing data movement), runtime power management (DVFS, clock/power gating, sleep states, and PS/PL independent power/clock domains), and hardware/software co-design (mapping computationally intensive, structurally regular cores to PL, and assigning control and irregular tasks to PS). Partial Reconstruction (PR) further improves energy efficiency by reusing accelerators in the time dimension and offloading unused logic to reduce static power consumption. Evaluation often uses composite metrics such as performance per unit power consumption, energy per operation, and EDP/ED^{2}P, and comprehensively applies constraints such as latency, determinism, thermal limits, and battery life. The above foundation provides a unified standard for subsequent platform comparisons, application mapping, and comprehensive analysis of low-power design technologies.

3. FPGA–SoC integration platforms

The FPGA–SoC integrated platform integrates a field-programmable gate array (FPGA) with a hard processor system (PS) on a single chip, tightly coupling software and hardware functions in a single device. This integration eliminates board-level interconnections between the processor and FPGA, reducing I/O power consumption and latency while increasing data exchange bandwidth.

DOI: 10.54254/2755-2721/2025.GL39453

3.1. Zynq

In devices such as Xilinx Zynq, the on-chip processing system is connected to programmable logic via multiple high-bandwidth AXI ports, enabling efficient, low-latency coupling between the CPU and FPGA domains [2]. The SoC FPGA (such as Zynq-7000) pioneered the placement of PS and PL in independent power domains, allowing the programmable logic portion to be powered down independently, reducing power consumption when hardware acceleration is not required. With advances in process technology, next-generation platforms add more heterogeneous compute engines and complex on-chip interconnects. These improvements (discussed in subsequent chapters) include multi-core processors, AI/ML acceleration units, and on-chip network (NoC) structures, which are designed to increase computing throughput per watt.

3.2. Versal ACAP

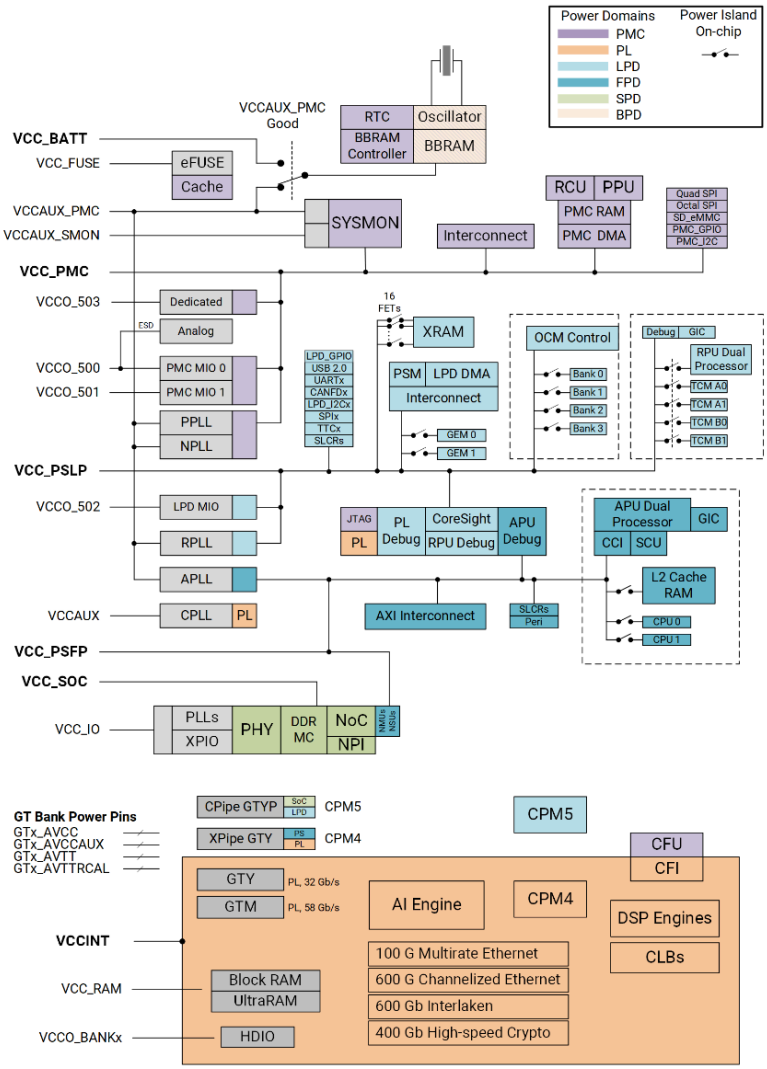

Figure 1. Various power domains and major interconnect structures in the Versal ACAP architecture [3]

Versal ACAP (Adaptive Compute Acceleration Platform) is Xilinx's next-generation heterogeneous SoC–FPGA platform, featuring an integrated AI Engine, programmable logic (PL),

DOI: 10.54254/2755-2721/2025.GL39453

multi-core processing system (PS), and on-chip network (NoC), designed to deliver exceptional energy efficiency and performance for edge AI and low-power embedded applications. Versal devices are organized into separate sections: Full Power Domain (FPD), Low Power Domain (LPD), PL Domain, System Power Domain (SPD), PMC and Battery Domain. Each domain is controlled by a separate power rail, enabling fine-grained power management and dynamic power switching. Versal ACAP (Adaptive Computing Acceleration Platform) is Xilinx's next-generation heterogeneous FPGA-SoC platform featuring an integrated AI engine, programmable logic (PL), multicore processing system (PS), and network-on-a-chip (NoC) to provide better energy efficiency for edge AI and low-power embedded applications. (NoC) on-chip to deliver improved power efficiency and performance for edge AI and low-power embedded applications [3]. As shown in Figure 1, this design enables precise power management and ensures efficient data transfer between the programmable logic, processor, and storage subsystems.

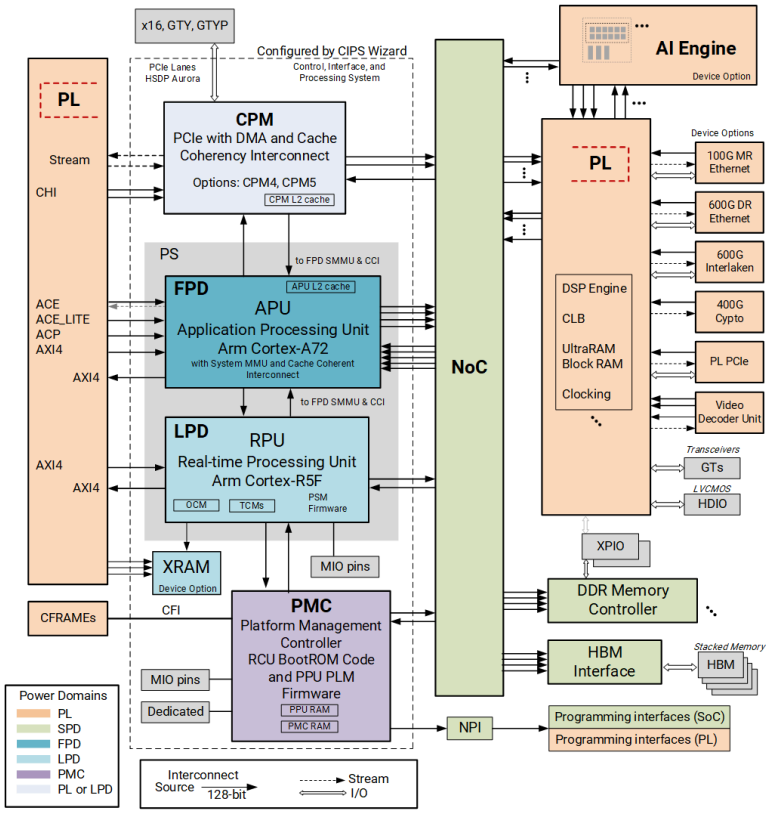

Figure 2. This diagram shows the integrated architecture of programming logic (PL), AI engine, general purpose and real-time processing (APU/RPU), and scalable network-on-chip (NoC) in AMD Xilinx Versal ACAP, supporting flexible tasking, low-power operation, and high-performance embedded computing [5]

The AI Engine is the core heterogeneous acceleration unit of the Versal platform, consisting of a large number of SIMD-VLIW vector processors, typically 128–400, equipped with on-chip local SRAM, DMA controllers, and a interconnect network. It performs high-throughput operations through a pipelined structure, making it highly suitable for high-performance AI inference and signal processing acceleration. The AI Engine is tightly coupled with the PL and NoC, supporting high-speed streaming and memory-mapped AXI4 interface communication, thereby improving memory access efficiency and reducing off-chip DRAM power consumption [4]. Figure 2 shows

DOI: 10.54254/2755-2721/2025.GL39453

how Versal ACAP combines programmable logic (PL), AI engine arrays, general-purpose processors (APUs), and real-time processors (RPUs) with a scalable on-chip network (NoC). This architecture enables flexible task allocation while supporting low-power operation and meeting the demands of high-performance embedded computing.

3.3. PolarFire SoC

PolarFire SoC is a RISC-V architecture SoC-FPGA device launched by Microchip based on the 28nm SONOS process, specifically designed for low-power, reliability-sensitive embedded applications. The platform integrates a set of deterministic multi-core RISC-V CPU subsystems (MSS), including the U54 application core and monitoring core supporting Linux and real-time operating systems, paired with a processing subsystem featuring deterministic interconnect and L2 cache, enabling support for real-time applications and simplified scheduling architecture [6]. As shown in Figure 3, the interrupt mechanism of Microchip PolarFire SoC FPGA transmits interrupt signals generated in the FPGA logic area through the PLIC and F2M interfaces to the MSS subsystem, which is managed by the AXI interconnect. This architecture enables more efficient collaboration between user-defined logic and the RISC-V processor core.

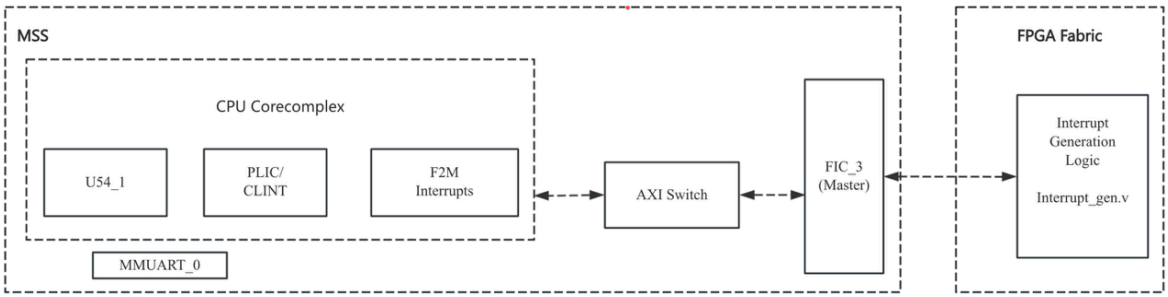

Figure 3. Illustrates the interrupt routing mechanism in the Microchip PolarFire SoC FPGA. Interrupts are generated in the FPGA logic area and passed through the PLIC and F2M interfaces to the MSS (microprocessor subsystem), where they are coordinated and managed by the AXI interconnect bus. The architecture supports efficient collaboration between custom logic and enables efficient collaboration between RISC-V processor cores

In terms of power performance, PolarFire SoCs have very good static and dynamic power management. According to Microchip's official white paper [7], its power consumption is approximately 50% lower than that of traditional FPGAs, and it supports fast power-up without requiring external configuration loading. Compared to SRAM-based FPGAs, it has a shorter startup time and higher configuration stability, making it suitable for use in situations where frequent power outages will be encountered or for systems that require real-time response. With 12.7 Gbps high-speed transceiver (SerDes) and logic resources ranging from 25k to 460k logic units, this model is typically used in embedded deployments ranging from low to medium density to industrial grade [8]. A study successfully quantifies the single event effect (SEE) and total ionizing dose (TID) on PolarFire SoCs by performing high-energy proton tests on bare metal and FreeRTOS applications. An observation-based fault tolerance assessment methodology was proposed, demonstrating that its built-in monitoring mechanism can ensure reliability to a considerable extent [9].

DOI: 10.54254/2755-2721/2025.GL39453

4. Low-power embedded applications

4.1. Edge AI & computer vision

Edge AI and vision applications increasingly require real-time inference capabilities, while at the same time needing to constantly consider the power consumption of implementing these capabilities, making FPGA-SoC platforms very promising for these applications. Research shows that in quantized neural network inference and embedded vision pipelines, FPGAs typically offer 3x–4x better energy efficiency than GPUs. For example, the Versal platform in the Edge Series Gen 2 demonstrates AI performance per watt that is approximately 4 times higher than GPUs in multiple tests, And it exhibits 10x the compute density of UltraScale+ MPSoCs [10]. Another study aimed at comparing the performance of the vision cores of different devices found that on the ZCU102 FPGA platform, FPGAs can reduce the required energy consumption per unit of a complex production line by a factor of 1.2x-22.3x compared to ARM CPUs and GPUs [11].

4.2. Smart sensing & IoT gateways

SamurAI Architecture A typical example of a multifunctional IoT node employs an ultra-low-power event-driven wake-up controller (WuC), which consumes almost micro-watt levels in standby mode and responds to wake-up signals in about 207 nanoseconds. It includes a RISC-V processing core and an ML accelerator, achieving ML performance efficiency of 1.3 TOPS/W. The architecture achieves up to 15,000 peak-to-noise power ratios and saves 3.5x power in intermittent load scenarios. Significantly extends node uptime compared to cloud processing models. The WuC asynchronously processes sensing tasks with extremely low power consumption, while heavy tasks are loaded only on demand, thereby maximizing the idle power consumption budget [12]. This type of intelligent sensing design utilizes the microsecond-level wake-up capability, local PL preprocessing, and selective activation of subsystems of these FPGA-SoCs to significantly reduce end-to-end energy consumption while maintaining responsiveness. This architecture is ideal for battery-powered or energy harvesting IoT nodes and industrial gateways, where minimizing idle power consumption is critical [12].

5. Power-saving design techniques

5.1. HW/SW co-design & accelerator partitioning

Hardware-software co-design in FPGA-SoC systems strives to rationalize the division of computational tasks between the processing system (PS) and programmable logic (PL) to achieve optimal performance and energy efficiency. Representing the approach SECDA uses SystemC simulation in conjunction with hardware prototyping, this flow achieves an average of 3.5× performance acceleration and 2.9× power consumption reduction over CPU-only implementations on the Zynq SoC platform for a wide range of DNN models [13].

Another impact study, HASCO, uses tensor-based intermediate representation (IR) combined with Bayesian optimization and reinforcement learning to efficiently explore the hardware and software design space, achieving a latency improvement of 1.25×-1.44× compared to a separated design flow [14].

In addition to computational partitioning, communication-aware mapping is also critical: tasks with high data locality or streaming behavior should be mapped to PL, thus reducing energy-

DOI: 10.54254/2755-2721/2025.GL39453

intensive PS-PL DMA operations and off-chip access.

A well-established co-design flow typically includes candidate core performance/power analysis, evaluation of partitioning schemes (including compute, storage, timing, area, and power), HLS/SystemC prototype construction, emulation or hardware simulation, and finally hardware synthesis to achieve energy-efficient and responsive solutions in embedded systems.

5.2. DVFS & clock/power gating

Dynamic voltage and frequency scaling (DVFS) is critical in FPGA-SoC systems to reduce dynamic power consumption by dynamically adjusting the voltage and clock frequency at runtime based on load requirements. Reducing the voltage according to the CMOS power equation can significantly reduce power consumption with little impact on performance. Dynamic voltage and frequency scaling (DVFS) is also important in FPGA-SoC systems. It allows the voltage and clock frequency to be dynamically scaled at runtime based on load requirements, which reduces dynamic power consumption. According to the CMOS power equation, lowering the voltage can significantly reduce power consumption with only a slight impact on performance. A study built a DVS platform based on the Leon3 processor core on a Virtex-4 FPGA and achieved significant power optimization for video processing tasks through dynamic clock management and voltage adaption, demonstrating that DVFS applications on FPGAs are feasible [15].

Clock gating can reduce dynamic power consumption by reducing switching activity by turning off clock signals to logic modules that are in an idle state. And power gating can optimize the leakage power consumption problem by directly disconnecting power to unused modules, although this feature requires a trade-off between wake-up latency and the complexity of restoring the state prior to shutdown. These techniques need to be finely managed to trade off energy savings, wake-up latency, and system reliability.

6. Trends and future directions

6.1. Advanced heterogeneous integration

Advanced heterogeneous integration is revolutionizing traditional SoC design through methods such as 3D IC stacking, chiplet-based module packaging, and embedding eFPGA IP into SoCs/ASICs. These approaches offer significant advantages in terms of power consumption, performance, and manufacturing feasibility, overcoming the limitations of single large-scale chips.

Three-dimensional integrated circuits (3D ICs) use TSV technology to vertically stack bare dies, effectively shortening interconnect lengths and reducing parasitic capacitance. Research shows that average line lengths can be reduced by approximately 36%, power consumption can be lowered by 28–36%, and latency and bandwidth performance can be improved, offering significant advantages over conventional two-dimensional ICs [16]. Additionally, chip let architecture integrates multiple functional bare dies (such as compute, memory, and I/O) within 2.5D (silicon interposer packaging) or 3D packaging, allowing each module to be manufactured using its optimal process technology. This modular approach enhances manufacturing yield, reduces costs, and facilitates the development of large-scale heterogeneous system integration. Particularly in high-performance computing and AI acceleration domains, the chiplet technology market is projected to experience explosive growth.

An embedded FPGA (eFPGA) is an IP logic block integrated into an ASIC or SoC that provides reconfigurability without the area and power overhead of a full FPGA. It supports flexible hardware customization (e.g., ML gas pedals or sensing preprocessing) to maintain configurability while

DOI: 10.54254/2755-2721/2025.GL39453

reducing idle power and silicon area [17]. The review points out that eFPGA integration reduces resource waste, improves latency/power balance, and is increasingly used in energy-sensitive SoCs.

6.2. Toolchains & AI-driven DSE

Modern FPGA-SoC designs are increasingly relying on advanced toolchains with AI-driven Design Space Exploration (DSE) to more efficiently trade-off performance, resources and power (PPA). High-level synthesis (HLS) tools and machine learning predictive models are two key drivers.

HLS tools allow developers to describe hardware functionality using high-level languages (e.g., C / C++ / SystemC), improving design readability and accelerating the iterative process. A comparison of the performance, resource utilization and power consumption of commercial and academic HLS tools has been made, and the study concluded that it is important to reach a balance between the selection and configuration of the tool and the metrics of each parameter of the HLS tool [18].

The DSE framework also includes SpikeExplorer and AutoHLS, which use Bayesian optimization and reinforcement learning in conjunction with predictive models for design acceleration, optimizing for tradeoffs between power, area, latency and accuracy when building FPGA gas pedals for CNNs or SNNs.

7. Conclusion

The integration of FPGA and SoC platforms is an effective approach to enable the development of low-power, high-performance embedded systems. This paper reviews and synthesizes current advances in the integration of this FPGA and SoC platform, as well as future directions, including architectural innovations, power-optimized designs, and intelligent design toolchains.

Platforms such as the Xilinx Zynq and AMD Versal ACAP combine programmable logic with heterogeneous processor subsystems to perform dynamic tasks under increasingly stringent power constraints. Optimization techniques such as hardware-software co-partitioning, optimized gas pedal design, and Dynamic Voltage and Frequency Scaling (DVFS) can go a long way in helping to reduce system-level power consumption. In addition, advanced integration methods such as eFPGA IP and 3D IC packaging provide significant improvements in system flexibility, modularity and scalability.

In applications such as smart sensing, edge AI, real-time control systems, and domain-specific acceleration, the FPGA-SoC platform facilitates a new generation of embedded workloads through its flexibility and extensibility. Approaches such as hardware-software co-design, dynamic voltage frequency scaling (DVFS), clock and power gating, and AI-driven design space exploration also provide better ideas for system optimization

In summary, the convergence of three-dimensional integrated circuits (3D ICs), embedded FPGAs (eFPGAs), and automation will further expand the application boundaries of low-power computing. As these technologies mature, FPGA-SoC integration will provide a richer technology path for energy-efficient embedded computing and can serve as a key platform for cutting-edge research and innovation in low-power system design.

References

[1] Li, Y., Liu, Z., Xu, K., Yu, H., & Ren, F. (2018). A GPU-outperforming FPGA accelerator architecture for binary convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems, 14*(2), 1–16. doi:10.1145/3154839

DOI: 10.54254/2755-2721/2025.GL39453

[2] Xilinx. (2019, October). *Zynq-7000 SoC technical reference manual, UG585 (v1.12.2)*. https://www.xilinx.com/publications/prod_mktg/zynq-7000-generation-ahead-backgrounder.pdf
[3] Xilinx. (2021). *Versal ACAP power domains, UG1304 – Versal Adaptive SoC power management*. https://docs.amd.com/r/2021.2-English/ug1304-versal-acap-ssdg/Versal-ACAP-Platform-Management
[4] Xilinx. (2020, July 16). *AI engine architecture manual, AM009 v1.0*. https://0x04.net/~mwk/xidocs/am/am009-versal-ai-engine.pdf?utm_source=chatgpt.com
[5] Xilinx. (2025, May 29). *Versal Adaptive SoC design guide, UG1273 – AI engine*. https://docs.amd.com/r/en-US/ug1273-versal-acap-design/AI-Engine
[6] Microchip Technology. (2025, May 21). *PolarFire SoC FPGAs – product overview*. https://www.microchip.com/en-us/products/fpgas-and-plds/system-on-chip-fpgas/polarfire-soc-fpgas
[7] Microchip Technology. (2022, March 30). PolarFire SoC FPGA: Interrupt latency and data transfer throughput measurements white paper. *Microchip Technology*. https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/SupportingCollateral/whitepapers/Microchip_PolarFire_SoC_FPGA_Interrupt_Latency_Data_Transfer_Throughput_Measurements_White_Paper.pdf
[8] Microchip Technology. (2022, March 30). PolarFire SoC CPU performance benchmarking white paper. *Microchip Technology*. https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/SupportingCollateral/microchip_polarfire_soc_cpu_performance_benchmarking_white_paper.pdf
[9] Mattos, A. M. P., dos Santos, D. A., & Dilillo, L. (2024, December). Reliability analysis of baremetal and FreeRTOS applications on Microchip PolarFire SoC RISC-V multiprocessors using high-energy protons. *IEEE Access*, 19922–19936. https://doi.org/10.1109/ACCESS.2024.3524726
[10] AMD. (n.d.). Versal AI edge series [Product brief]. *AMD*. https://www.amd.com/en/products/adaptive-socs-and-fpgas/versal/ai-edge-series.html?utm_source=chatgpt.com
[11] Qasaimeh, M., Denolf, K., Lo, J., et al. (2019, May). Comparing energy efficiency of CPU, GPU and FPGA implementations for vision kernels. *arXiv preprint*. https://arxiv.org/abs/1906.11879
[12] Miro-Panades, I., et al. (2023, June). SamurAI: A versatile IoT node with event-driven wake-up and embedded ML acceleration. *IEEE Journal of Solid-State Circuits, 58*(6), 1782–1797. https://doi.org/10.1109/JSSC.2023.3274650
[13] Haris, J., Gibson, P., Cano, J., Agostini, N. B., & Kaeli, D. (2021). SECDA: Efficient hardware/software co-design of FPGA-based DNN accelerators for edge inference. In *Proceedings of the 2021 IEEE 33rd Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. doi:10.1109/SBAC-PAD53543.2021.00015
[14] Xiao, Q., Zheng, S., Wu, B., Xu, P., Qian, X., & Liang, Y. (2021). HASCO: Towards agile hardware and software co-design for tensor computation. In *Proceedings of the 2021 ACM/IEEE International Symposium on Computer Architecture (ISCA)* (pp. 1055–1068).
[15] Nunez-Yanez, J. L., Chouliaras, V. A., & Gaisler, J. (2007). Dynamic voltage scaling in a FPGA-based system-on-chip. In *Proceedings of the 2007 International Conference on Field-Programmable Logic and Applications (FPL)*. doi:10.1109/FPL.2007.4380688
[16] Khan, N. H., Alam, S. M., & Hassoun, S. (2011). Power delivery design for 3-D ICs using different through-silicon vias. *IEEE Transactions on VLSI Systems, 19*(4), 647–657. doi:10.1109/TVLSI.2009.2038165
[17] A survey and comparative study of embedded FPGA architectures. (2020). *IJCSNS*. https://journals.tubitak.gov.tr/cgi/viewcontent.cgi?article=1412&context=elektrik
[18] Nane, R., et al. (2016). A survey and evaluation of FPGA high-level synthesis tools. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35*(10), 1591–1604. doi:10.1109/TCAD.2015.2513673