A COMPARATIVE REVIEW OF FPGA AND GPU ACCELERATORS FOR AI

Maksudjon Usmonov
Lobar Asretdinova
Nurilla Mahamatov
Department of Automatic Control and Computer Engineering
Turin Polytechnic University in Tashkent
maksudjon.usmonov@polito.uz;
L.asretdinova@polito.uz;
n.mahamatov@polito.uz
https://doi.org/10.5281/zenodo.15105579

Abstract

This investigation presents a brief yet thorough comparative assessment of FPGA (SoC)–based and GPU-based AI accelerators across applications that encompass edge devices to data center training environments. Primary performance parameters—latency, throughput, energy efficiency, programmability, and scalability—are thoroughly evaluated with a specific concentration on deep neural network inference and training. The research further highlights the significance of hardware/software co-design and high-level synthesis (HLS) in augmenting FPGA performance. Representative platforms, such as the one from Xilinx and Nvidia, are referenced to illustrate prevailing trends. Findings suggest that while GPUs excel in throughput and development simplicity, FPGAs exhibit reduced latency and enhanced energy efficiency in power-sensitive or real-time applications.

Keywords: FPGA; GPU; AI accelerator; edge computing; deep neural networks; latency; energy efficiency; hardware/software co-design; high-level synthesis.

1. Introduction

The rapid advancement in artificial intelligence has driven a transformative evolution in computing architectures, resulting in the emergence of specialized hardware accelerators tailored to diverse application domains. Traditional central processing units (CPUs) are increasingly inadequate for the computational and energy demands of deep neural network (DNN) training and inference, leading to the rise of accelerators such as Graphics Processing Units (GPUs) and Field-Programmable Gate Arrays (FPGAs). GPUs have become the standard for large-scale AI applications due to their ability to perform massively parallel computations, supported by robust programming ecosystems and libraries like CUDA and cuDNN that have been refined over decades [1]. Their capability to process thousands of threads concurrently makes them ideally suited for training deep neural networks, where high throughput and large-scale data processing are paramount.

Alternatively, Field-Programmable Gate Arrays (FPGAs) present a valuable benefit attributed to their innate versatility and ability to be reconfigured, which permits creators to construct personalized hardware systems that can be thoroughly refined for unique duties. This level of customization engenders deterministic processing with ultra-low latency and enhanced energy efficiency—attributes that are especially critical in environments constrained by power and necessitating real-time performance, such as edge computing devices, autonomous systems, and Internet of Things (IoT) applications [2]. In contrast to Graphics Processing Units (GPUs), whose architectures remain static and are optimized for a wide array of operations,

FPGAs facilitate hardware/software co-design wherein both the algorithm and its corresponding hardware implementation are concurrently optimized. This paradigm fosters applications capable of functioning with minimal power expenditure while still adhering to stringent performance criteria.

A notable difference arises between GPUs and FPGAs, especially as we evaluate key platforms like the NVIDIA A100 GPU and the Xilinx Zynq UltraScale+ System on Chip (SoC) FPGA. The NVIDIA A100, equipped with its sophisticated tensor cores and elevated memory bandwidth, is meticulously engineered to achieve exceptional throughput for data center training tasks. It accommodates mixed-precision computing and scales effectively across multi-GPU clusters, rendering it a preferred option for cloud-based deep learning applications. Conversely, the Xilinx Zynq UltraScale+ platform epitomizes the FPGA paradigm by amalgamating both programmable logic and processing system cores within a singular chip. This synthesis facilitates the creation of application-specific accelerators that attain low latency and high energy efficiency—characteristics that are fundamental for edge inference applications where real-time responsiveness is paramount [1], [3].

Recent scholarly literature has delved into these distinctions with considerable depth. Investigations have indicated that while GPUs are proficient in contexts requiring high-throughput processing and expedited development cycles, they often exhibit elevated latency and energy consumption as a result of intrinsic architectural overheads and dependency on batch processing. In contrast, FPGAs, despite traditionally presenting greater programming challenges, have exhibited substantial advantages in energy efficiency and latency mitigation when optimized through high-level synthesis (HLS) methodologies [3], [4]. This emerging trend of utilizing high-level synthesis (HLS) to bridge the programmability gap is revolutionizing the integration of FPGAs within artificial intelligence systems, thus making them more attainable for developers who may lack advanced hardware design expertise.

The purpose of this investigation is to deliver an extensive review of these two accelerator forms by assessing primary metrics like latency, throughput, energy efficiency, programmability, and scalability. In executing this analysis, it not only reviews contemporary research findings but also emphasizes the significance of hardware/software co-design in attaining optimal performance. The discourse is anchored in empirical data and technical specifications, thereby ensuring that the insights remain pertinent to both academic researchers and practicing engineers. By carefully analyzing the trade-offs between the GPU's user-oriented characteristics and substantial computational throughput in relation to the FPGA's low latency and energy-efficient framework, this comparative inquiry intends to contribute to the decision-making process pertaining to the identification of the most suitable hardware accelerator for differing artificial intelligence applications [1], [2], [3], [4]. A range of studies have explored the inherent trade-offs between FPGAs and GPUs. Research by Vaithianathan *et al.* [1] and Goz *et al.* [2] has shown that while GPUs offer superior raw throughput for data-parallel computations and deep learning training, FPGAs achieve markedly lower latency and higher energy efficiency in real-time processing. Nurvitadhi *et al.* [3] provided evidence that, with proper optimization (such as reduced precision and network sparsity), FPGAs can closely match or even exceed GPU performance in inference tasks. Additional studies [4], [5], and [6] have evaluated the power and performance metrics in edge

computing and vision applications, indicating that FPGAs can lead to significant energy savings—up to 22× improvements in some cases. Meanwhile, surveys [7] and deployments like Microsoft's Project Catapult [8] emphasize that while GPUs benefit from a rich software ecosystem and easier programmability, FPGAs are gaining ground through advancements in high-level synthesis and hardware/software co-design approaches. Overall, the literature consistently highlights that the optimal choice depends on application-specific needs such as latency requirements and power budgets.

2. Evaluation Criteria

The comparative examination in this investigation is predicated on a synthesis of information from recent scholarly publications and technical documentation. The assessment concentrates on five principal criteria, namely *performance*, *energy efficiency*, *programmability*, *scalability*, and *domain-specific* appropriateness.

Performance is quantified by latency (duration required per task) and throughput (operations per unit of time). Field Programmable Gate Arrays (FPGAs) are recognized for their deterministic latency, whereas Graphics Processing Units (GPUs) excel in managing extensive batch operations. Energy efficiency is scrutinized in terms of performance per watt, with FPGAs generally surpassing GPUs in energy-critical contexts. The programmability of the device refers to the simplicity of development, contrasting the advantages that GPUs derive from well-established high-level programming languages (e.g., CUDA, OpenCL) and comprehensive libraries, in contrast to FPGA development, which has conventionally necessitated proficiency in hardware description languages. Nonetheless, advancements in High-Level Synthesis (HLS) are diminishing this disparity. Conversely, scalability considerations are directed towards both vertical scaling (within a single device) and horizontal scaling (across multiple devices). GPUs exhibit effective scaling with multi-device configurations, while FPGA scalability frequently necessitates bespoke interconnects. When evaluating the accelerator for domain-specific appropriateness, particular application domains are scrutinized, encompassing real-time inference, edge computing, and data center training.

Representative instances, such as the Xilinx Zynq UltraScale+ and the NVIDIA A100, are employed as benchmarks to exemplify overarching trends without fixating exclusively on any singular model. The analysis also incorporates the concept of hardware/software co-design, especially for FPGA implementations, to emphasize how design optimizations can enhance overall efficiency [3], [4].

3. Results & Comparative Analysis

The Xilinx Zynq UltraScale+ and NVIDIA A100 exemplify the contrasting strengths of FPGA and GPU accelerators in AI applications. Studies indicate that the Xilinx Zynq UltraScale+ FPGA SoC offers remarkable energy efficiency and ultra-low latency, making it highly effective for edge inference and real-time applications; its ability to implement custom hardware pipelines can reduce inference latency by an order of magnitude compared to conventional solutions [1], [3]. In contrast, the NVIDIA A100 GPU is engineered for high-throughput data center training, utilizing advanced tensor cores and mixed-precision computing to accelerate deep neural network training at scale [1]. While the A100 delivers superior raw computational power and scalability for large-batch processing, its higher power consumption highlights the

trade-off between energy efficiency and throughput—a contrast that underlines the decision-making process when choosing the optimal accelerator for specific AI workloads [2], [3].

In terms of *performance*, FPGAs are capable of achieving ultra-low, deterministic *latency* by implementing dedicated hardware pipelines. This characteristic is crucial for real-time applications like autonomous systems and high-frequency trading, where even microsecond-level delays are unacceptable [1], [3]. GPUs, in contrast, incur higher latency due to factors such as task batching and data transfer overheads, which can result in non-deterministic response times.

However, GPUs are optimized for *high-throughput* computing, leveraging thousands of cores and high memory bandwidth to handle large-scale, data-parallel workloads. This makes them especially effective for training extensive neural networks and processing large batches of inference tasks [1]. While FPGAs can sustain high throughput through deep pipelining, their performance is often limited by available logic resources and lower clock frequencies.

Moreover, *energy efficiency* is a critical metric for both edge devices and data centers, where power consumption directly affects operational cost and thermal management. FPGAs typically offer superior performance-per-watt because they allow the creation of custom data paths that perform only the necessary operations, thereby minimizing redundant logic and energy wastage [2], [6]. By tailoring the hardware to the specific requirements of an application, FPGAs can achieve a high degree of efficiency; for instance, in complex vision processing, optimized FPGA designs have demonstrated energy savings of up to *22×* compared to GPU implementations. This efficiency arises not only from reduced dynamic power—due to lower operating frequencies and minimized switching activity—but also from the ability to power-gate unused sections of the logic fabric.

In contrast, GPUs are built as general-purpose processors that must cater to a wide range of applications, resulting in architectural overheads that lead to higher overall power consumption. Although modern GPUs incorporate advanced power management features and improved architectures that lower their energy footprint during peak performance, the inherent design for massive parallelism often leads to energy inefficiencies when executing tasks that are not perfectly parallelizable. The energy trade-off becomes particularly evident when comparing deep learning inference tasks on edge devices, where the high power draw of GPUs can be a significant disadvantage.

According the analysis done by [2] and [6], the efficiency of FPGAs is enhanced by their ability to operate at lower clock frequencies while still achieving high throughput through deep pipelining. This contrasts with GPUs, which rely on high clock speeds and large numbers of cores to deliver performance. In many scenarios, especially in battery-powered or thermally constrained environments, the lower power envelope of FPGAs makes them a more suitable choice. Detailed studies have quantified this advantage, showing that for specific tasks, the energy per inference (often measured in inferences per joule) can be an order of magnitude better on FPGAs compared to GPUs.

Another important aspect is the difference in *static* versus *dynamic* power consumption. FPGAs, while highly efficient during active operation, do have a static power component due to leakage in the reconfigurable logic; however, this static consumption can be mitigated through design optimizations and power gating. In contrast, GPUs maintain higher baseline power

consumption even when underutilized, as many of their resources remain powered to maintain readiness for high throughput operations. Such differences highlight the suitability of FPGAs for continuous, energy-sensitive tasks, particularly in remote or mobile deployments where battery life and heat dissipation are critical concerns.

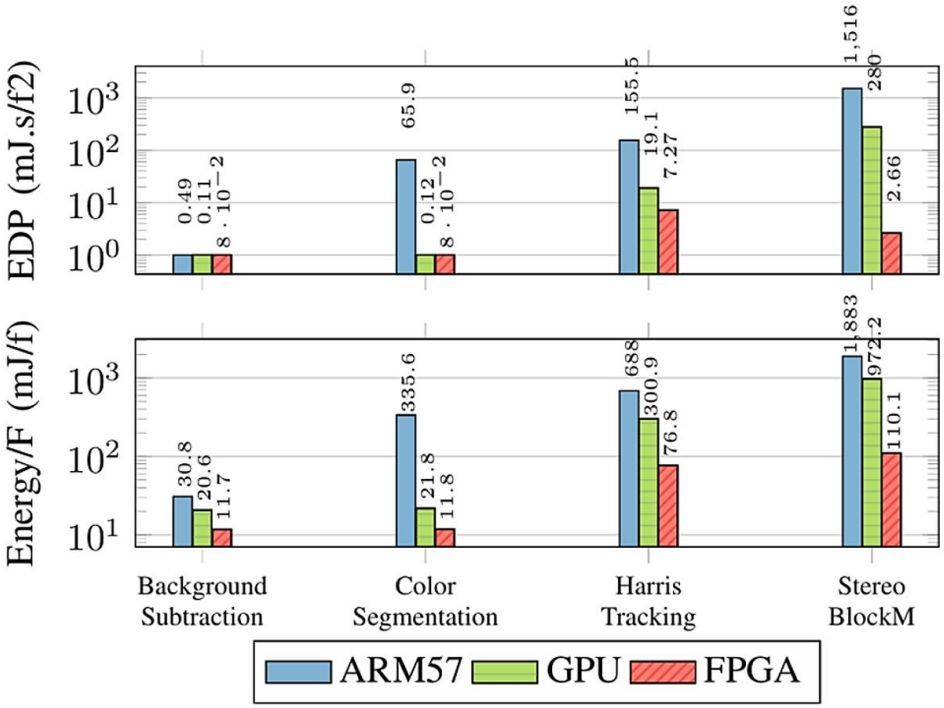

Fig.1. FPGA outperforms GPUand CPU in energy/frame consumption and EDP, based on data from [6]. This graph highlights the substantial energy advantage of FPGAs in specific application scenarios. (EDP – Energy Delay Product = the product of energy/frame and delay time/frame. Lower EDP is better which means that the hardware architecture can finish specific computation tasks using less power in less time) [6]

Overall, the ability of FPGAs to deliver customized, energy-efficient solutions makes them particularly attractive for edge computing and real-time applications, where energy efficiency can be just as important as raw computational throughput. Meanwhile, GPUs continue to improve in energy efficiency with each new generation, yet their design philosophy and architectural overheads inherently limit their performance-per-watt in specialized, low-power applications.

In the context of *programmability*, GPUs benefit from an extensive software ecosystem featuring high-level programming environments, such as CUDA and OpenCL, alongside a wealth of pre-optimized libraries for deep learning and parallel computing. This extensive ecosystem enables rapid prototyping and a straightforward development process, resulting in a shorter time-to-market for GPU-based solutions [1]. Developers can leverage mature frameworks and debugging tools that streamline optimization and performance tuning, which reduces development complexity.

Traditional FPGA development, on the other hand, has typically required expertise in hardware description languages (HDL) like VHDL or Verilog. Such low-level programming approaches demand detailed understanding of hardware architecture, often resulting in longer design cycles and a steeper learning curve [2]. However, the advent of High-Level Synthesis

(HLS) tools has significantly reduced these barriers by allowing developers to write code in high-level languages (such as C/C++ or OpenCL), which is then automatically translated into efficient hardware implementations [3], [4]. HLS not only accelerates the development process but also facilitates iterative design, enabling more rapid exploration of hardware/software co-design strategies. Despite these advances, the maturity and convenience of GPU programming still offer distinct advantages, particularly for teams seeking to deploy solutions quickly without extensive hardware-specific knowledge.

Scalability is another aspect that has to be considered when choosing the right accelerator for a given application. Regarding this aspect, GPUs are highly scalable both within a single device and across multiple devices. Multi-GPU configurations are common in data centers, with established frameworks facilitating distributed training and inference. This scalability is supported by fast interconnects and optimized software libraries, which make it relatively easy to expand performance by adding additional GPU units.

However, scaling FPGA-based solutions can be more complex due to the need for custom designs and interconnect strategies. Vertical scaling—using larger or more capable FPGAs—can increase performance, but horizontal scaling across multiple FPGA units requires careful design of communication protocols. Recent advances in FPGA cloud services and custom interconnect solutions are beginning to address these challenges, though the scalability of FPGAs remains more application-specific compared to GPUs [7], [8]. *Application domains* of both technologies must be decided taking into account their strength and weaknesses. FPGAs excel in real-time inference scenarios due to their low latency and energy-efficient processing capabilities. Their ability to interface directly with sensors and process streaming data makes them well-suited for smart cameras, autonomous vehicles, and IoT devices where immediate responses are critical. Meanwhile, GPUs are the preferred choice for data center environments where high-throughput is essential. Their architecture supports the massive parallelism required for training deep neural networks and processing large batches during inference. The mature software support and scalability of GPUs make them ideal for handling large-scale AI workloads in cloud-based and high-performance computing scenarios.

The table below summarizes the key differences between FPGA and GPU accelerators:

Effective Criterias	**FPGA based accelerators**	**GPU based accelerators**
Latency	Ultra-low, deterministic latency; ideal for real-time tasks [1], [3].	Higher latency due to batching and data transfers.
Throughput	High for pipelined tasks; constrained by clock speed and resource limits.	Extremely high throughput for parallel operations [1].
Energy Efficiency	Superior performance-per-watt with custom-tailored data paths [2], [6].	Generally lower energy efficiency despite high compute power.
Programmability	Requires HDL/HLS; development is more complex, though HLS reduces effort [3], [4].	Mature programming models and extensive libraries simplify development.

Scalability	Flexible scaling options, but multi-unit coordination is challenging [7], [8].	Easily scalable in multi-GPU systems with standardized frameworks.

Table 1: Summary of comparative strengths and weaknesses.

4. Conclusion

The study suggests that Field-Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs) confer unique benefits for the enhancement of Artificial Intelligence (AI), with both technologies displaying marked effectiveness in designated contexts. GPUs demonstrate remarkable proficiency for activities that demand substantial throughput and expedited development phases, positioning them as the favored alternative for the training of large-scale neural networks and the execution of batch inference within data center frameworks. Their well-established software ecosystems, comprehensive libraries, and strong support for distributed processing facilitate efficient scaling for intricate, data-intensive operations [1]. In another regard, FPGAs stand out due to their potential to achieve ultra-low latency while maintaining high energy efficiency, which are indispensable attributes for real-time applications and edge computing environments that enforce strict power and timing restrictions. The capacity of FPGAs to implement customized data pathways and utilize hardware/software co-design via High-Level Synthesis (HLS) enables them to be meticulously tailored to the specific task, achieving performance levels that may be unattainable with the more generalized architecture of GPUs [2], [3].

Moreover, the decision between utilizing FPGA and GPU accelerators ought to be informed by particular application specifications, including latency sensitivity, limitations on power consumption, and the intricacy of the workload. For instance, in situations where accurate timing and minimized power consumption are imperative—such as in autonomous systems, Internet of Things (IoT) devices, and real-time inference—FPGAs offer a significant advantage. In contrast, for applications that can accommodate greater latency in exchange for increased throughput, such as extensive deep learning training and high-volume data processing, GPUs continue to represent the most effective solution. Anticipating future developments, the advancing landscape of accelerator architecture indicates a prospective scenario wherein both FPGAs and GPUs might be utilized in a complementary manner to capitalize on the advantages inherent to each architecture. Ongoing progress in High-Level Synthesis (HLS) and other development tools is anticipated to further diminish the programmability divide associated with FPGAs, potentially facilitating broader adoption in domains that have historically been dominated by GPUs [1]–[8]. This synergistic strategy may catalyze a new epoch of heterogeneous computing systems, in which the meticulous orchestration of various accelerator types will foster optimal performance, energy efficiency, and scalability across a wide array of artificial intelligence applications.

Foydalanilgan adabiyotlar/Используемая литература/References:

1. M. Vaithianathan *et al.*, “Comparative Study of FPGA and GPU for High-Performance Computing and AI,” *ESP Int. J. Adv. Comput. Technol.*, vol. 1, no. 1, pp. 37–46, 2024.
2. D. Goz *et al.*, “Performance and Energy Footprint Assessment of FPGAs and GPUs on HPC Systems Using Astrophysics Application,” *Computation*, vol. 8, no. 2, Art. 34, 2020.

3. E. Nurvitadhi *et al.*, “Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks?” in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA)*, 2017, pp. 271–274.
4. C. Vasile, A. Ulmămei, and C. Bîră, “Image Processing Hardware Acceleration—A Review of Operations Involved and Current Hardware Approaches,” *J. Imaging*, vol. 10, no. 12, p. 298, 2024.
5. T. P. D. Gamage, M. Z. K. Siddiqui, and H. T. Mouftah, “Novel Case Study and Benchmarking of AlexNet for Edge AI: From CPU and GPU to FPGA,” in *Proc. IEEE CCECE*, 2020, pp. 1–6.
6. M. Qasaimeh *et al.*, “Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels,” in *Proc. IEEE Int. Conf. Embedded Software and Systems (ICESS)*, 2019, pp. 1–8.
7. S. Biookaghazadeh and M. Zhao, “Are FPGAs Suitable for Edge Computing?” in *Proc. 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge)*, 2018, pp. 1–6.
8. A. Putnam *et al.*, “A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services,” in *Proc. 41st Int. Symp. Computer Architecture (ISCA)*, 2014, pp. 13–24.