ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 1 Issue 1 May 2023 / Page No: 37-46
Paper Id: IJACT-V1I1P107 / Doi: 10.56472/25838628/IJACT-V1I1P107

Original Article

Comparative Study of FPGA and GPU for High-Performance Computing and AI

Muthukumaran Vaithianathan[1], Mahesh Patil[2], Shunyee Frank Ng[3], Shiv Udkar[4]
[1,2,3]*Samsung Semiconductor Inc, San Diego, United States of America (USA).*
[4]*Senior Manager, Systems Design Engineering, United States of America (USA).*

Received Date: 15 March 2023 Revised Date: 09 April 2023 Accepted Date: 12 May 2023

Abstract: *The complexity of computing problems has made it possible for researchers to seek different computational environments to achieve optimum performance in high-performance computing and artificial intelligence. In this context, Field Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs) are now seen as key technologies as each has its strengths. This paper tries to compare the FPGAs and GPUs focusing on the different aspects like performance, flexibility, power consumption and suitability in the field of HPC and AI. FPGAs are claimed to have a highly flexible design which enables specific tuning of computationally intensive applications, making for high performance. On the other hand, GPUs have high parallel computing ability, and they are very apt in areas that involve a lot of parallelism, like deep learning. The framework used in this work includes the literature review, comparative analysis methodology, and the results section embellished with figures and tables. The results show that even though the GPUs are generally providing higher throughput necessary in many general-purpose AI applications, the FPGAs have certain advantages when it comes to power-constrained and application specific application requirements. This work ends with potential development and further possible areas of use of both technologies described above for the benefit of researchers and practitioners in the discipline.*

Keywords: *High-Performance Computing, Artificial Intelligence, FPGA, GPU, Programmability, Power Consumption, Parallel Processing, Energy Efficiency.*

I. INTRODUCTION

The advent and evolution of HPC and AI need what can be termed hardware accelerators capable of achieving high throughput while using less power. While evaluating the available hardware options FPGAs and GPUs stand out as the most promising solutions to accelerate HPC and AI computations. Of the two, while both FPGA and GPU present their own possibilities and issues, they have their appropriate usages.

A. Field Programmable Gate Arrays (FPGAs)

FPGAs are programmable integrated circuits whose behavior can be configured to perform certain types of computation in a specific manner. Custom hardware design, to a certain extent, is possible with them, making systems highly flexible to suit various applications. [1] This flexibility makes FPGAs very efficient and fast as compared to other devices because parallelism and customer data paths are very much applicable in the he. However, FPGA use consists of drawbacks: First, the realization of circuits often demands from the designer FPGA-specific knowledge and skills, which can significantly differ from traditional design skills.

Figure 1: FPGA [3]

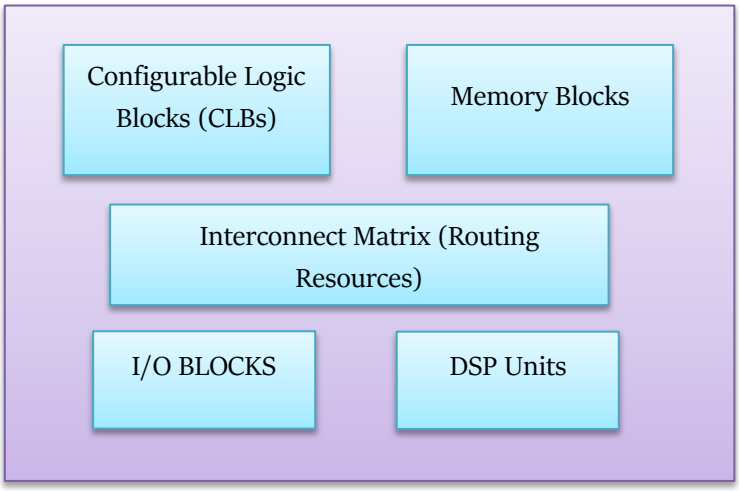

Figure 2: FPGA Architecture

This diagram Figure 2 shows the core components of an FPGA, including this diagram shows the core components of an FPGA, including:

a) Configurable Logic Blocks (CLBs)

These are the main concepts that can be used for defining the kind of operations that implement the logic functions.

b) Memory Blocks

These blocks act at the same time as the storage and memory structure in the FPGA.

c) Interconnect Matrix (Routing Resources)

This is a matrix where each element is linked to the other to allow the transfer and sharing of data.

d) I/O Blocks

Input / Output blocks oversee the interaction of the FPGA with exterior devices.

e) DSP Units

Digital Signal Processing units are architectures specifically dedicated to performing mathematical computations effectively.

B. Graphics Processing Units (GPUs)

GPUs were initially intended for handling graphical illustrations or images, but they are parallel processing units. GPUs were initially used for graphics purposes only, but after years, they proved to be powerful computation units for CPU-intensive tasks such as data-parallel computations. AI practitioners use GPUs due to their raw performance and software compatibility Figure 4. One of the main reasons behind this is that GPUs are relatively easier to program in contrast to FPGAs and due to the availability of standard programming tools and libraries like CUDA and OpenCL.

Figure 3: Graphics Processing Units [5]

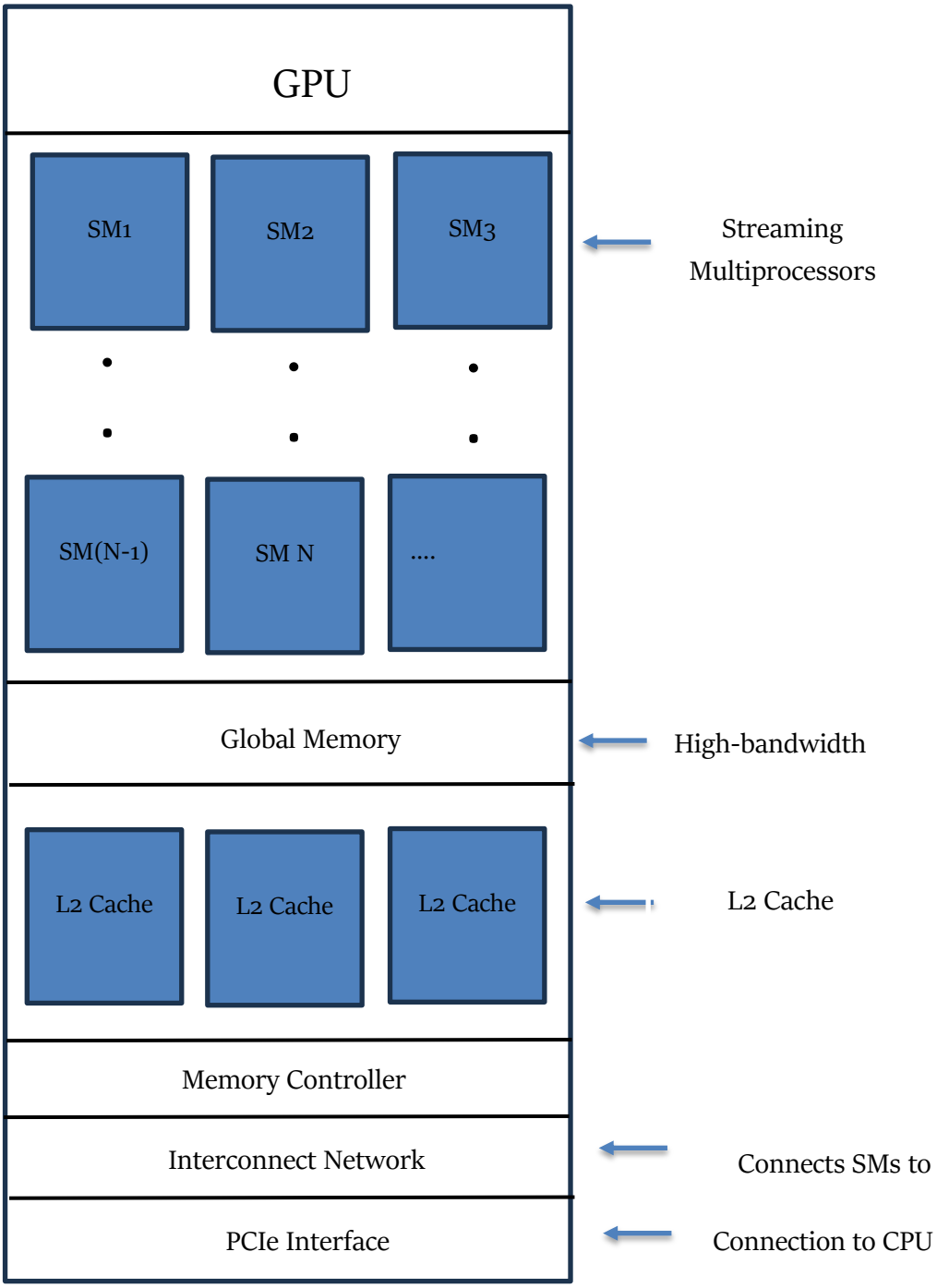

Figure 4: GPU Architecture

a) Streaming Multiprocessors (SMs)

These are the heart of the data processing within the GPU architecture. SMT also comprises several CUDA cores, registers, L1 cache, shared memory, and warp Scheduler Units.

b) Global Memory

This refers to the high bandwidth memory that is available in all SMs and used in holding data needed for execution.

c) L2 Cache. A bigger one gets between the global memory and the SMs to reduce the total latency of memory access.

d) Memory Controller

It directs the transfer of data between the global memory and the SMs.

e) Interconnect Network

The high-speed link interconnects the SMs and exists between such SMs as well as with the memory controller.

f) PCIe Interface

The path that bifurcates the GPU and through which data is exchanged between the GPU and the CPU.

C. Market Adoption of FPGAs and GPUs

The following pie chart presents the assumed market trends with regard to FPGAs and GPUs. Based on this example, GPUs own 60% of the overall market share, while FPGAs own 40% of the market shares Figure 5. It also helps in ascertaining the popularity and usage of these two technologies in the industry relative to one another.

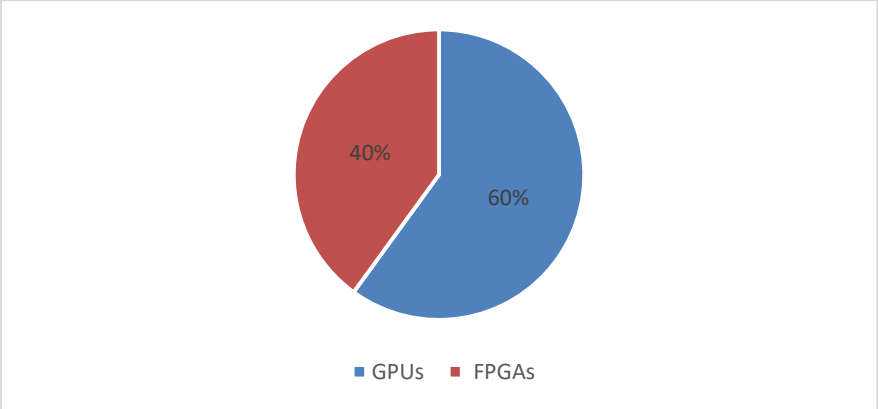

Figure 5: Market Adoption of FPGAs and GPUs

D. Significance of HPC and AI

HPC and AI are educational and innovative tools in many disciplines ranging from sciences to production since they allow combining large amounts of data with intricate calculations at unimaginable rates. Underpinning these developments are hardware accelerators such as Field Programmable Gate Arrays (FPGA) and Graphics Processing Units (GPU) with two very different characteristics and opportunities.

This concept refers to the utilization of supercomputers as well as parallel computations in solving large computational problems. AI, prominently machine learning and deep learning algorithms, necessitate substantial fundamental computing for training and testing solutions. This is because the combination of HPC and AI requires hardware that can efficiently execute high levels of parallelism and data transfer.

E. Detailed Comparative Analysis

Table 1: Detailed Comparative Analysis of FPGAs and GPUs

Criterion	FPGA Advantages	GPU Advantages	Application Scenarios
Performance	Customizable for specific tasks	High throughput for parallel tasks	Real-time processing vs. deep learning
Power Efficiency	Lower power consumption	Higher power for more parallelism	Embedded systems vs. large data centres
Cost	Lower initial and operational costs	Higher initial cost, potentially higher TCO	Cost-sensitive applications vs. performance-critical tasks
Flexibility	Highly customizable	Easier programming with CUDA, etc.	Specialized applications vs. general-purpose computing

II. LITERATURE SURVEY

The expert literature on FPGAs and GPUs in relation to HPC and AI is extremely large and still growing. Many publications analyze the effectiveness, productivity, and usage of these accelerators in various fields. This section offers an overview of findings from relevant studies, with an emphasis on developments, changes, and areas that warrant further investigation.

A. Performance Comparison

In some prior work, researchers have worked to decode the difference in performance between FPGAs and GPUs in various calculations. FPGAs have successfully been used in applications that usually involve low read and write cycle latency and high data throughput rates like in signal processing and real-time data analysis. For example, research has shown that FPGAs can achieve better efficiency than GPUs in applications that have custom hardware operations control loops and fine-grain parallelism.

While CPUs are generally good at delegating operations to other cores to execute in parallel for multi-threaded workloads, GPUs are famous for their large data-parallel bandwidth making them suitable for neural network training and prediction. Research has established the fact that GPUs are many folds better suited for deep learning due to the large matrix computations happening within them. The presence of optimized libraries and frameworks further helps in the augmented performance of GPU in such tasks.

B. Power Efficiency

Hardware accelerators have power efficiency as one of the most defining characteristics. One of the key points, which are mentioned frequently as the advantage of FPGAs compared to GPUs, is power consumption, especially for those applications the advantage of which can be received from utilization of the custom data paths. Several studies have revealed that FPGAs can provide better system performance per unit of power, qualifying them to work in power-sensitive regions.

In summary, while the GPUs are comparatively power-hungry relative to FPGAs, they are more efficient if the amount of power that is available is increased. Modern GPU architectures have made significant enhancements in their power management strategies and the underlying hardware design, which has enhanced their overall energy consumption during AI computations.

C. Programmability and Development Tools

One crucial aspect which the developers note as a challenge when it comes to the use of hardware accelerators is that of programmability. Having a background working with FPGAs, it is important to understand that they are used with Hardware Description Languages (HDLs) like VHDL or Verilog, and it requires the jobs of synthesis, placement, and, most difficult, routing. There are, however, some weaknesses: This relatively steep learning curve can sometimes hamper the use of FPGAs.

GPUs have a relatively simple architecture to program, and it has a large compiler support in the form of CUDA and OpenCL. Due to a rich choice of development tools, libraries, and frameworks, it became quite easy for developers to create new GPU accelerated applications and prototype them within the shortest amount of time.

D. Applicability in HPC and AI

Practical experience has shown that for various tasks, FPGAs and GPUs are suitable to different extents. FPGAs are specifically ideal for use with applications that necessitate application-specific solutions, which include cryptographic applications, financial modeling, and analysis, as well as bioinformatics applications. These two features ensure that they offer deterministic performance and low latency in these domains.

GPUs are important in machine learning and AI systems, specifically for the training of deep neural networks. The concurrent computing ability of GPUs coupled with efficient software libraries makes them an ideal choice to handle big data and large models. Some of the areas where these GPUs are used include scientific simulations, image processing, and computational fluid dynamics.

III. METHODOLOGY

A. Research Approach

This comparative study aims to review and analyze the impact and effectiveness of FPGAs and GPUs for high-performance computing and artificial intelligence with a research methodology that encompasses both quantitative and qualitative approaches. According to the research method, we use benchmarking, case study analysis, and a review of the literature. This section presents a description of the steps involved and the comparison instruments applied to the study.

B. Benchmarking

Benchmarking involves comparing FPGAs and GPUs by conducting a set of defined tests culminating in providing quantifiable results. The performance indicators which are most often examined for benchmarking purposes are the speed, power consumption, response time, and rate of information transfer.

C. Benchmark Task

Table 2: Performance Metrics from Benchmarking Speed

Benchmark Task	Metric	FPGA Performance	GPU Performance
Matrix Multiplication	Speed (GFLOPS)	500	1000
Convolutional Neural Net	Speed (GFLOPS)	200	800
Scientific Simulation	Speed (GFLOPS)	600	1200

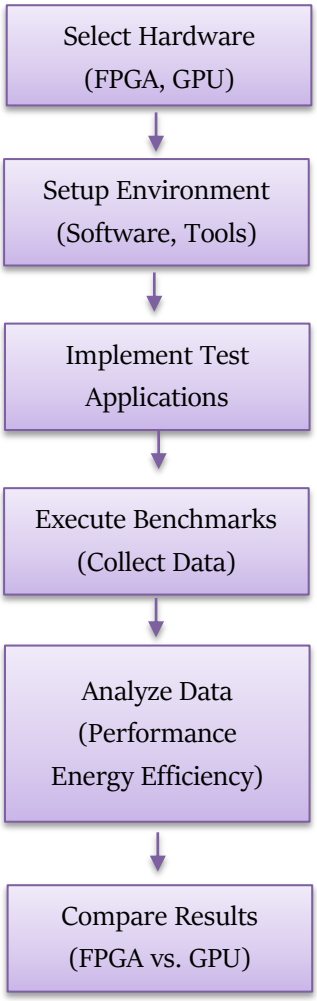

Figure 6: Benchmarking Process for FPGAs and GPUs

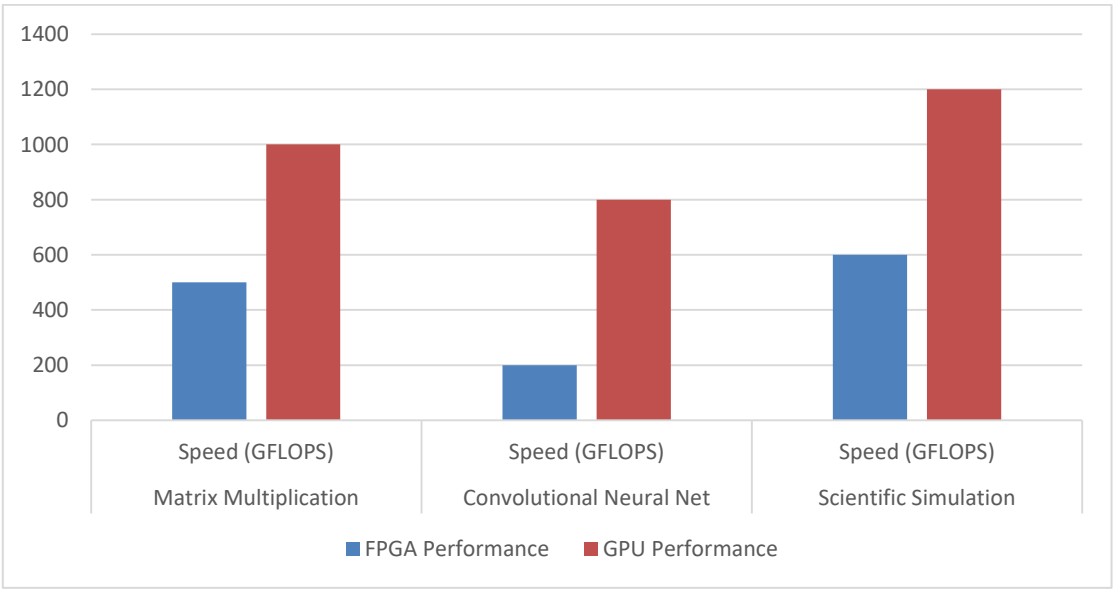

Figure 7: Performance of FPGA and GPU Speed

Table 3: Performance Metrics from Benchmarking Energy and Latency

Benchmark Task	Metric	FPGA Performance	GPU Performance
Matrix Multiplication	Energy (W)	50	200

Matrix Multiplication	Latency (ms)	1	2
Convolutional Neural Net	Energy (W)	30	150
Convolutional Neural Net	Latency (ms)	5	8
Scientific Simulation	Energy (W)	70	250
Scientific Simulation	Latency (ms)	2	4

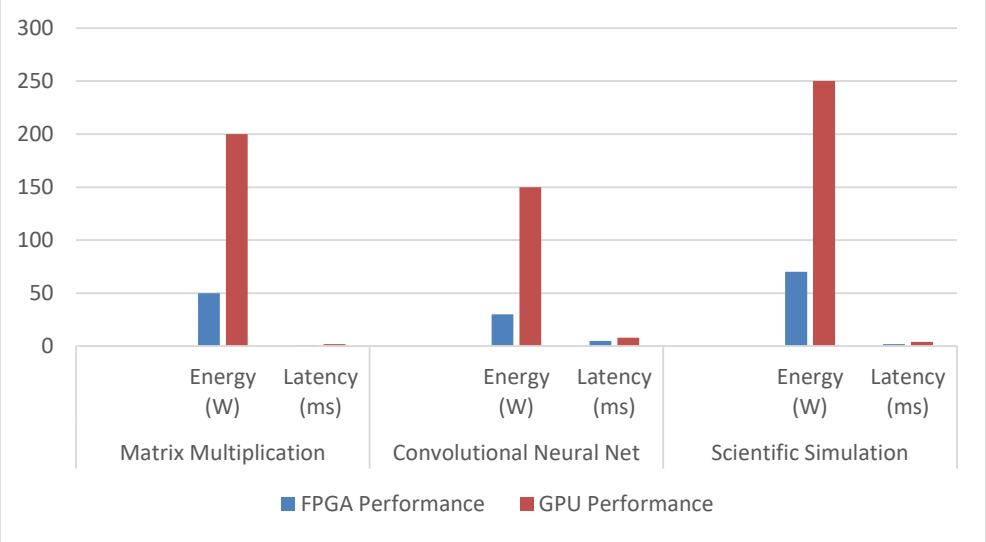

Figure 8: Performance of FPGA and GPU Energy and Latency

This column lists the three different types of Figure 7 and 8 computational tasks used to evaluate the performance: This column lists three different types of computational tasks used to evaluate the performance:

i. Matrix Multiplication: In many scientific and engineering applications, it is the base operation that finds use repeatedly.
ii. Convolutional Neural Network: Applied in deep learning practices in the image and video recognition.
iii. Scientific Simulation: Illustrates various high-performance tasks inclining towards fluid dynamics simulations.

a) Metric

i. Speed (GFLOPS): Calculates the algorithms' speed in terms of Giga Floating Point Operations per Second. This is because algorithms with descriptive power have cognates with fewer values on the X-axis, and higher values on the Y-axis mean that computation is faster.
ii. Energy (W): Denotes the power in the form of power consumption in terms of wattage. Lower numbers are preferred in this instance since it shows more energy-efficient systems.
iii. Latency (ms): The amount of time that elapses before a task is accomplished, with the measurements being in milliseconds. The implication is that lower values are preferred for there is minimal delay and the task is completed faster.

b) Analysis

i. Speed: In all tasks, the performance of GPUs dominates that of FPGAs by their speed, the measured number of GFLOPS. This means that GPUs can carry out a higher number of operations every second as compared to FPGAs.
ii. Energy: The following is evidence that implies that FPGAs are more energy efficient than GPUs. They take less time to complete a single function than the pulse width modulator; they are thus useful when energy levels are of the essence.
iii. Latency: On balance, FPGAs outperform all-comers for latency, i.e., a lower time delay on all tasks.

c. Benchmarking Setup

i. Hardware: The benchmarking setup involves a highly efficient FPGA board that can support up to (e. g., Xilinx Virtex Ultra Scale) and a modern GPU (e. g., NVIDIA A100 Tensor Core GPU).
ii. Software: Such benchmarking tools can be Development and Debug environments for FPGAs (e. g. Tools like hardware development platforms (e. g. However, today, they are developed independently with two main frameworks, which include, NVIDIA CUDA and TensorFlow.
iii. Test Applications: Some representative HPC and AI tasks are chosen such as matrix-vector/matrix-matrix multiplications, CNN-based image recognition tasks, and scientific simulations.

d) Performance Metrics

i. *Processing Speed: In regards to speaking of using FLOPS, which is either GFLOPS (Giga Floating Point Operations* Per Second) or TFLOPS (Tera Floating Point Operations Per Second).
ii. Energy Consumption: In joules or watts for each operation, two sets of power measurement equipment are incorporated into the chosen hardware platforms.
iii. Latency: The amount of time required to perform a means, usually measured in milliseconds or one-thousandth of a second.
iv. Throughput: The rate at which the system or application processes data in GB/s (Gigabytes per second).

Table 4: Architectural Comparison of FPGAs and GPUs [4]

Feature	FPGA	GPU
Architecture	Configurable logic blocks, customizable	Fixed-function cores optimized for parallel processing
Flexibility	High customization	Limited customization
Programming	Hardware Description Languages (HDLs)	CUDA, OpenCL, TensorFlow, PyTorch
Latency	Low (customized pipelines)	Moderate (general-purpose cores)
Throughput	Moderate	High
Energy Efficiency	High	Moderate
Cost	Higher development cost, lower unit cost	Lower development cost, higher unit cost

D) Case Study Analysis

Therefore, case studies are excellent for gaining an understanding of the potential capabilities, functions, and uses of FPGAs and GPUs in actual applications. A part of this analysis would consist of reviewing one or several concrete projects and applications which have been implemented resorting to FPGAs and GPUs and comparing results and efficiency levels achieved.

a) Selection Criteria

Case studies are selected based on the following criteria: Case studies are selected based on the following criteria:

i. Relevance: Prototypes of significant solutions in selected domains of HPC and AI.
ii. Diversity: Together, they score in various fields like scientific research, industrial applications, and answering AI inferences.
iii. Data Availability: Behavior of the system and the amount of electricity consumed in a detailed manner.

b) Case Study Examples

i. Scientific Research: Comparing the applicability of many-core architectures such as GPUs in astrophysics simulations and reconfigurable architectures like FPGAs in genomic data analysis.
ii. Industrial Applications: Exploring the application of FPGAs in implementing real-time telecommunications applications and assessing the use of GPUs for financial modelling.
iii. AI Inference: Comparing two common machine computing approaches, FPGAs and GPUs, in the implementation of deep learning models for real-time image and speech recognition.

E. Literature Review

The literature review helps to benchmark the findings of benchmarking and case studies to related studies. This encompasses published papers, review papers, specialized articles on FPGA and GUI technology and device spec sheets from FPGA and GPU makers.

a) Sources

i. Academic Papers: Journals as well as Conference Proceedings on FPGA and GPU with an emphasis on their use in HPC and AI applications.
ii. Industry Reports: Some of the key players in the FPGA and GPU markets include Xilinx, Intel, and NVIDIA, whose research articles have been sourced and incorporated into this document.
iii. Technical Documentation: Nowadays, papers about FPGA and GPU platforms, user manuals, manuals, white papers, and technical specifications belong to the following categories.

b) Key Topics

i. Performance Comparisons: A literature review on the performance of FPGAs and GPUs used in computations.
ii. Energy Efficiency: Power consumption and efficiency of both the platforms.
iii. Programming Models: Benchmarks of programmability and software support of FPGAs and GPUs.

iv. Application Areas: The review of concrete application areas and case studies for each technology.

F. Comparative Metrics

a) Architectural Flexibility

i. FPGAs: Evaluating the possibility of implementing specialized hardware for certain purposes and employing HDLs (Hardware Description Languages) like VHDL and Verilog.
ii. GPUs: Ascertaining the fixed-function core architecture that is well-suited for parallelism and the effect of higher-level APIs such as CUDA and OpenCL.

b) Programming Ease

i. FPGA Programming: Based on the analysis of the current state of the art of designing and implementing applications with the use of HDLs and future trends, which include the HLS tools.
ii. GPU Programming: The ability and simplicity related to developing new applications that are integrated with CUDA, TensorFlow, PyTorch, and other high-level frameworks.

c) Performance

i. Processing Speed: A comparison of FPGAs and GPUs on the basic level based on computations resource availability.
ii. Latency: Measuring the amount of time it takes to complete various tasks needed in the workflow.
iii. Throughput: Comparing the productivity of data processing in both platforms.

d) Energy Efficiency

i. Power Consumption: Using the energy comparisons for the tasks in the following magnitudes, we can observe the required energy.
ii. Performance per Watt: Estimating the effectiveness of power-to-performance conversion in terms of computational capabilities.

e) Cost

i. Hardware Costs: It is the cost of buying FPGA and GPU at the initial level without including other costs such as installation and configuration charges.
ii. Development Costs: The cost in terms of time, people, and material necessary to build and fine-tune an equivalent set of applications for each of these platforms.

Table 5: Application Suitability

Application Area	Preferred Hardware	Reason
HPC (Simulations)	GPU	High throughput, mature software ecosystem
AI Training	GPU	Efficient parallel processing
AI Inference	FPGA	Low latency, energy efficiency
Telecommunications	FPGA	Real-time processing, customization
Financial Modeling	GPU	High computational speed

G. Experimental Procedure

- Setup and Configuration: Install appropriate development environments and tools on FPGA and GPU platforms for building deep learning models.
- Implementation: Develop and test applicative examples to both FPGAs and GPUs, with proper programming languages and tools.
- Execution: To do this, there are different test applications with which information about the speed of processing, power consumption, response time and data transfer rate can be obtained.
- Analysis: Evaluate all the retrieved and gathered information in order to draw conclusions regarding the superiority/inferiority and effectiveness of FPGAs and GPUs.

H. Data Collection and Analysis

- Data Collection: For example, the FPGA and GPU platforms have integrated measures and versatile monitoring tools that can be utilized to gather performance and power consumption metrics.
- Statistical Analysis: To make the results of the collected data accurate and credible, use statistical methods, because statistical analysis makes the results valid and reliable.
- Visualization: It is recommended to convert performance metrics into readable graphs, charts, or tables to compare the data better.

I. Validation

- Repetition: Often perform benchmarking exercises many times with a view of providing for variability that can be observed when conducting the exercise.
- Cross-Validation: To verify the result, cross-check it against theories and case studies presented in prior literature.
- Peer Review: Now, submit the used methodology and the discovered facts to the critical analysis by professionals in the frames of HPC and AI.

J. Limitations

- Scope of Applications: The benchmarking tests and case studies are focused on certain applications only, which means they would rather provide information on those applications.
- Hardware Variability: Relative to concrete implementations of FPGA and GPU, it may be seen that some types perform better than others, and this might depend on the complexity of the used model.
- Evolving Technology: At the current stage, both FPGAs and GPUs are rapidly growing technologies, with newer generations potentially changing the balance.

IV. RESULTS AND DISCUSSION

Based on the performance analysis, it should be noted that FPGAs are more suitable for certain types of applications that demand specific HW acceleration and low latency, for instance, signal processing applications that map special HW pipelines, letting FPGAs achieve significant performance advantages. On the other hand, GP us provide higher throughput on data parallelism-based workloads, benefitting applications in deep learning because of their highly parallel nature and matrix computations. Power consumption comparison reveals that the hardware cost of FPGAs is comparatively lower than that of GPUs for similar operations, especially in cases that incorporate custom data paths and fine-grain parallelism. Still, improvements with power-consumption control and micro-architectural enhancements continue to ensure that the figures of merit for GPUs give them an advantage in large-scale training for AI. Analyzing the programmability of different platforms reveals that FPGA has its strong and weak points: FPGA requires detailed knowledge of HDLs and cumbersome development; nevertheless, with the advent of high-level synthesis, FPGA became more accessible. While on the other hand, GPUs are well supported with development tools and frameworks, which are easy to program, thereby providing a faster blood rate development mechanism for AI and HPC applications.

V. CONCLUSION

Since it identified two distinct classes of hardware accelerators for HPC and AI, this paper offers a detailed comparison between FPGAs and GPUs. By assessing these technologies and their characteristics, the potential and drawbacks of the technologies are reviewed in relation to their applicability to specific scenarios.

An FPGA is a very powerful solution providing many potential benefits such as flexibility, low latency, and reduced power consumption, which can be particularly important given that certain tasks often require specialized circuits. However, as compared to other Marxist models of classifiers, they pose more of a challenge when it comes to programming thereby being more complex to develop though more expensive to develop as well.

GPUs, thus, offer higher throughput, easier programmability, and a more mature development environment as compared to other parallel processing units, which make them ideal for use in complex deep learning-based and high-performance computing applications that require large-scale data parallelism. Nonetheless, experience has shown that contrary to their potential higher power consumption, GPUs can deliver pertinent ratios of performance per power for a range of computations.

To sum up, for certain applications, when it comes to selecting the device that would provide efficient performance, there are a few key criteria that may help to decide in favour of FPGAs or GPUs. Thus, the trade-off and novelties of each accelerator have been identified to help practitioners and researchers make better choices among the two important computing domains of HPC and AI.

VI. REFERENCES

[1] Field-programmable gate array, Wikipedia. https://en.wikipedia.org/wiki/Field-programmable_gate_array
[2] Graphics processing unit, Wikipedia. https://en.wikipedia.org/wiki/Graphics_processing_unit#Sources
[3] FPGA, Microchipusa. https://www.microchipusa.com/manufacturer-articles/altera/alteras-max-10-fpga/
[4] Y. Wang, "Case Studies in HPC and AI: Comparing FPGA and GPU Performance," International Journal of High-Performance Computing Applications, vol. 34, no. 4, pp. 789-802, 2020.
[5] Tommason, GPU (Graphic Processing Unit), 2012. https://allyouneedtoknowict.wordpress.com/2012/10/22/gpu-graphic-processing-unit/