PAPERS IN ENGLISH

COMPUTER SCIENCE, COMPUTER ENGINEERING AND MANAGEMENT

DOI – 10.32743/UniTech.2024.128.11.18638

IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE ALGORITHMS IN FPGA/ASIC HARDWARE PLATFORMS

Yevgeni Yermolin
Logic design engineer
Israel Haifa
yevgeny.ye@gmail.com

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА АППАРАТНЫХ ПЛАТФОРМАХ FPGA/ASIC

Ермолин Евгений Владимирович
Инженер по проектированию логики,
Израиль, г. Хайфа

ABSTRACT

The introduction of artificial intelligence (AI) algorithms into FPGA and ASIC hardware platforms is an important direction in modern computing systems aimed at improving performance and energy efficiency. The purpose of this work is to consider the features of the integration process of AI algorithms in FPGA/ASIC, to compare their advantages and disadvantages. The methodology used includes an analysis of existing approaches to the implementation of AI algorithms, including deep learning, on hardware platforms. During the study, it was found that FPGAs have high flexibility and adaptability, which makes them ideal for tasks requiring rapid reconfiguration. At the same time, ASICs offer high performance and energy efficiency for highly specialized tasks. However, both approaches face challenges associated with high development complexity and resource constraints. In conclusion, FPGAs and ASICs offer significant opportunities for optimizing AI algorithms, but the choice of platform depends on specific tasks and performance requirements.

АННОТАЦИЯ

Внедрение алгоритмов искусственного интеллекта (ИИ) на аппаратные платформы FPGA и ASIC является важным направлением в современных вычислительных системах, нацеленным на повышение производительности и энергоэффективности. Цель данной работы — рассмотреть особенности процесса интеграции алгоритмов ИИ на платформах FPGA/ASIC, а также сравнить их достоинства и недостатки. Использованная методология включает анализ существующих подходов к реализации алгоритмов ИИ, включая глубокое обучение, на аппаратных платформах. В ходе исследования было выявлено, что FPGA обладают высокой гибкостью и адаптивностью, что делает их идеальными для задач, требующих быстрой перенастройки. В то же время ASIC обеспечивают высокую производительность и энергоэффективность для высокоспециализированных задач. Однако оба подхода сталкиваются с проблемами, связанными с высокой сложностью разработки и ограничениями ресурсов. В заключение, FPGA и ASIC предоставляют значительные возможности для оптимизации алгоритмов ИИ, однако выбор платформы зависит от конкретных задач и требований к производительности.

Keywords: artificial intelligence, FPGA, ASIC, algorithms, deep learning, performance, energy efficiency.
Ключевые слова: искусственный интеллект, FPGA, ASIC, алгоритмы, глубокое обучение, производительность, энергоэффективность.

Introduction

Modern computing systems face an increasing need for enhanced performance and energy efficiency. Artificial intelligence (AI) plays a key role in this area, as its algorithms form the basis for solutions across various sectors, such as healthcare, automotive, telecommunications, and industrial automated systems. Consequently, there is growing interest in hardware platforms like FPGAs (Field-Programmable Gate Arrays) and ASICs (Application-Specific Integrated Circuits), which provide high computational power for executing complex AI algorithms.

Библиографическое описание: Yermolin Y. IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE ALGORITHMS IN FPGA/ASIC HARDWARE PLATFORMS // Universum: технические науки : электрон. научн. журн. 2024. 11(128). URL: *https://7universum.com/ru/tech/archive/item/18638*

FPGAs and ASICs offer unique opportunities for optimizing AI algorithm performance. FPGAs are distinguished by their flexibility and programmability, allowing efficient adaptation of architecture for specific tasks. Meanwhile, ASICs, with their specialized architecture, deliver maximum performance with minimal energy consumption, making them attractive for widespread deployment in commercial applications such as deep learning systems and neural networks.

The relevance lies in the fact that with the growing volume of data and complexity of AI algorithms, selecting an optimal hardware platform for their implementation has become essential. FPGAs and ASICs are among the most promising solutions; however, their choice requires careful analysis based on specific tasks and operating conditions.

The objective of this work is to examine the features and compare the advantages and disadvantages of implementing artificial intelligence algorithms on FPGA and ASIC hardware platforms, as well as to identify the areas where each platform is most effective.

Materials and Methods

The development and training of artificial intelligence technologies involve the use of four different types of silicon solutions. These include central processing units (CPU), graphics processing units (GPU), field-programmable gate arrays (FPGA), and application-specific integrated circuits (ASIC). CPUs are highly programmable but offer lower performance compared to specialized hardware chips designed for specific tasks. FPGAs provide significant flexibility and high computational power, making them particularly valuable in scenarios requiring a limited volume of reprogrammable chips [1]. Field-programmable gate arrays (FPGA) are a distinct type of semiconductor device that allows for multiple configuration changes post-manufacturing to perform specific tasks. Unlike standard chips with fixed architecture, FPGAs offer a high degree of adaptability, making them suitable for a wide range of computational tasks, including artificial intelligence (AI).

The core of an FPGA consists of a configurable logic block (CLB) array connected through programmable interconnects and supplemented by input/output blocks (IOB). This allows the reconfiguration of the circuit logic to execute complex computational algorithms, significantly facilitating task implementation. Device configuration is achieved through hardware description languages (HDL), such as VHDL or Verilog, enabling developers to define chip behavior at a higher level of abstraction.

FPGAs are not only in demand for AI but also in industries like telecommunications, automotive engineering, and aerospace. Their adaptability and ability to be updated in line with technological changes make them indispensable for developing new solutions. FPGA's parallel processing capabilities also significantly speed up task execution, positioning them as strong competitors to traditional hardware solutions.

As for application-specific integrated circuits (ASIC), they are specialized chips designed to perform a specific set of tasks. Unlike more general-purpose processors, ASICs are developed to carry out one specific function, allowing for high performance and energy efficiency. This makes such chips ideal for applications requiring significant computational resources, such as deep neural networks.

ASICs are non-reconfigurable post-manufacturing, limiting their application to dynamically changing tasks that require configuration flexibility. However, they are more efficient for well-defined tasks where performance and energy efficiency are paramount. For example, in AI tasks requiring high computational power and minimal energy consumption, ASICs are often the preferred choice [2].

According to market research firm Tractica, which specializes in market analysis, by 2025, revenues from AI-based software solutions are projected to reach $105.8 billion. However, the growing demand for these technologies presents several challenges for developers. These include the need to improve data processing and transmission speeds as well as enhance the overall performance of AI-based applications [3].

Tables 1 and 2 below present the advantages and disadvantages of using FPGAs and ASICs in the context of artificial intelligence.

Table 1.

Advantages and disadvantages of using FPGA in the context of artificial intelligence [4]

Advantages of FPGA for AI	Disadvantages of FPGA for AI
Flexibility: FPGAs can be reprogrammed to perform specific AI tasks, allowing architecture adjustments for particular algorithms and models.	Development complexity: FPGA programming requires high skill levels, increasing development time and necessitating specialized expertise.
Low power consumption: FPGAs typically consume less power than traditional processors and GPUs.	Limited performance: In some AI tasks, such as large-scale data processing or neural network training, FPGAs may underperform compared to GPUs and ASICs.
High parallelism: FPGAs enable parallel computations, making them efficient for neural networks and deep learning processing.	Long development time: FPGA development takes longer compared to ready-made solutions based on GPUs or CPUs.

Advantages of FPGA for AI	Disadvantages of FPGA for AI
Low latency: Embedded FPGA solutions can perform AI tasks with minimal latency, which is crucial for real-time applications.	Development cost: Initial setup and configuration of FPGA-based systems can be costly due to complex design requirements.
Task-specific optimization: FPGAs can be tailored for narrowly defined AI tasks, enhancing performance for specific applications.	Fewer supported libraries and tools: Unlike platforms such as GPUs, there are fewer available libraries and tools for FPGAs compatible with popular AI frameworks (TensorFlow, PyTorch).
Versatility in hardware integration: FPGAs can integrate with various sensors and peripheral devices, making them suitable for embedded AI systems.	Limited scalability: FPGAs may be less scalable compared to specialized solutions like TPUs or ASICs, especially in large-scale data centers.

Table 2.

Advantages and disadvantages of using ASICs in the context of artificial intelligence [4]

Advantages of ASIC for AI	Disadvantages of ASIC for AI
High performance: ASICs are designed specifically for executing particular AI tasks, providing maximum performance for those tasks.	Lack of flexibility: Unlike FPGAs, ASICs cannot be reprogrammed for other tasks, limiting them to the specific functions they were designed for.
Energy efficiency: Since ASICs are optimized for specific computations, they can perform tasks with significantly lower energy consumption compared to general-purpose processors and GPUs.	High development costs: Designing and manufacturing ASICs requires substantial investment, making them expensive during the initial development phase.
Optimization for specific algorithms: ASICs can be configured to execute specific AI algorithms, making them ideal for specialized tasks (e.g., convolutional neural networks).	Long development time: The design, testing, and production process for ASICs can be time-consuming, a critical drawback in the fast-evolving AI technology landscape.
Compactness: Since ASICs do not require additional software or hardware resources to execute AI tasks, they can be more compact and integrate into devices with limited resources.	High upgrade costs: If AI algorithms change or new tasks emerge, ASICs must be entirely redesigned, increasing the time and cost of developing new versions.
Ideal for mass production: When a specific AI application needs to be implemented on a large scale, ASICs are justified, as mass production reduces unit costs.	Limited applicability: ASICs are effective only for specific tasks, making them impractical for a wide range of AI applications requiring diverse computations and tasks.
Minimal latency: Since ASICs are designed to perform specialized operations without additional abstraction layers, they can ensure minimal latency in data processing, which is critical for real-time tasks.	Risk of obsolescence: ASICs quickly become obsolete as new AI algorithms and technologies emerge, as they cannot be updated for new tasks without a complete redesign.

Thus, the choice between FPGA and ASIC for AI depends on the task's specificity: FPGAs offer adaptability and parallelism, while ASICs provide maximum performance and energy efficiency for narrowly defined tasks.

Results

Advanced Driver Assistance Systems (ADAS) are one of the most rapidly developing areas in the automotive industry, serving as a driver of technological innovations aimed at enhancing road safety, particularly under conditions of heavy traffic. In recent years, ADAS has incorporated various features, such as radar and camera systems, to improve the safety of drivers and passengers. These systems provide real-time information on potential threats, helping to prevent accidents. The complexity of modern automotive electronic systems demands high reliability, as failures may pose risks to passengers' lives and health.

According to Strategy Analytics research, the prevalence of ADAS is expected to increase by 10% in the coming years. The main challenges faced by developers in creating data-processing platforms for ADAS include system performance reduction, video data transmission through high-speed interfaces, parallel and sequential processing, as well as ensuring platform scalability and external memory bandwidth. One solution involves the use of Field Programmable Gate Arrays (FPGA) and System-on-Chip (SoC) architectures, which enable accelerated innovation in the automotive sector.

ADAS relies on two key technologies: vision and sensor data fusion. Cameras integrated into intelligent vehicles perform object recognition, classification, and tracking tasks; however, they cannot accurately measure

the distance to obstacles, which is crucial for collision prevention. For this purpose, sensors such as LIDAR and RADAR are used. Various types of processors, such as CPUs, GPUs, FPGAs, DSPs, ASICs, and microcontrollers, are employed in visual data processing. Yet, the competition between FPGAs and GPUs for performance and energy efficiency leadership remains ongoing. Due to their reconfigurability, FPGAs often surpass ASICs, offering more flexible and cost-effective solutions.

The fusion of data from various sensors and algorithms enables ADAS to create a comprehensive view of the road situation, allowing drivers to receive timely warnings of potential hazards. Requirements for computing platforms for ADAS include architectural flexibility, scalability, high external memory bandwidth, and operational safety. Due to its characteristics, FPGA technology is a suitable platform for implementing such systems and represents a competitive alternative to traditional ASSP and ASIC-based solutions [5].

DeePhi, founded in March 2016, emerged from the collaborative efforts of teams from Stanford and Tsinghua University. According to CEO and co-founder Song Yao, DeePhi's core concept focuses on developing solutions to accelerate deep learning. Challenges such as cost-effectiveness, market timing constraints, and the rapid evolution of deep learning frameworks have made traditional approaches based on central processing units (CPU), graphics processing units (GPU), and application-specific integrated circuits (ASIC) less appealing. According to Yao, CPUs lack sufficient energy efficiency, GPUs are suitable for training but lack adequate inference performance, and DSPs face cache miss issues, failing to deliver adequate performance. Although ASICs offer high performance for specialized tasks, they require significant development time and often fail to be cost-effective due to a limited market reach.

In this context, field-programmable gate arrays (FPGA) have become an attractive alternative due to their combination of high performance, energy efficiency, and architectural flexibility. Yao emphasizes that FPGAs meet the key requirements for deep learning, providing reliability, configurable flexibility, and high memory bandwidth. Additionally, FPGAs, unlike ASICs, are available in a ready-made form, which reduces the time to market. However, programming these devices to support rapidly evolving deep learning frameworks remains a major challenge, making them less accessible for widespread use.

To address this issue, DeePhi developed a fully automated optimization process that includes compression, compilation, and acceleration algorithms, allowing a seamless integration of software and hardware development efforts. This led to the creation of a deep learning processing unit (DPU), capable of competing with GPUs in energy efficiency, especially for tasks such as image and speech recognition. DeePhi actively collaborates with companies involved in drone development, video surveillance, and cloud services to integrate its solutions into their workflows.

The company introduced two FPGA architectures: Aristotle, aimed at accelerating convolutional neural networks (CNN), and Descartes, designed for long short-term memory (LSTM) networks. Instead of using OpenCL, DeePhi employed its compiler, significantly speeding up the compilation process. The application of deep compression methods also reduced computational resources and memory requirements.

DeePhi notes that its FPGA-based solutions demonstrate energy efficiency exceeding that of mobile and desktop GPUs. However, comparisons with server-level workloads remain a matter of discussion. In the future, DeePhi plans to continue advancing its technologies with a focus on integrating software and hardware solutions [6].

Frameworks used in deep neural networks (DNN) are specialized software solutions designed to simplify the process of developing and training models. These tools enable developers to create and configure complex architectures without requiring an in-depth understanding of algorithm mechanics. Among the most popular libraries in this field are Caffe, TensorFlow, PyTorch, and Keras, each offering extensive customization options for various tasks.

Caffe is one of the leading platforms focused on image-processing tasks using convolutional neural networks (CNN). Developed at the Berkeley AI Research Lab, this framework supports various architectures, including RNN and LSTM. The system is optimized for use on both central and graphics processors through support for libraries like NVIDIA cuDNN and Intel MKL. Caffe is compatible with C, C++, Python, and MATLAB, making it convenient for integration into diverse research projects.

TensorFlow, created by Google, is a powerful platform for building and training deep learning models. This open-source framework supports various languages, including R, C++, and Python, and is adapted for use on both CPUs and GPUs. TensorFlow's flexible architecture enables models to run on various hardware configurations, making it a popular choice for tasks involving neural networks. TensorFlow's second version includes numerous updates that enhance GPU performance.

Keras is a high-level API designed for rapid prototyping of neural networks and experimentation. This tool supports both convolutional and recurrent neural networks and was created to integrate with TensorFlow. The simplicity and efficiency of Keras, written in Python, attract many researchers, especially for experiments with neural networks.

PyTorch, developed by Facebook's research lab, is widely used for tasks related to computer vision and natural language processing. This library supports interfaces in Python and C++, making it a flexible tool for various machine-learning applications.

Hardware support for deep neural networks is also an important aspect. Using FPGA as hardware accelerators for DNN provides certain advantages but requires extensive knowledge of hardware description languages, such as VHDL or Verilog. To simplify this process, High-Level Synthesis (HLS) tools have been developed, which allow high-level code to be converted into hardware descriptions. However, even with these tools,

implementing models on FPGA requires a deep understanding of both architecture and neural networks. There are also specialized platforms such as OpenCL, Intel's OpenVINO, and Xilinx Vitis AI, which significantly ease the process of integrating DNN at the hardware level [7].

Below, Table 3 describes the possibilities of applying deep learning algorithms on various hardware platforms.

Table 3.

The possibilities of using deep learning algorithms on various hardware platforms [7]

Hardware Platform	Deep Learning Capabilities	Advantages	Disadvantages
Central Processing Units (CPU)	Can be used for deep learning computations, though less efficient than GPU.	Readily accessible, no need for special equipment.	Slow computation speed, high memory load.
Graphics Processing Units (GPU)	Provide accelerated computations for parallel data processing, making them effective for DL.	High performance, effective with large data volumes, parallel computations.	High cost, additional power requirements.
Tensor Processing Units (TPU)	Specialized processors for accelerating matrix multiplications and other DL tasks.	Optimized for deep neural networks, high energy efficiency.	Limited availability, primarily used in Google cloud platforms.
Field-Programmable Gate Arrays (FPGA)	Used to customize hardware for specific deep-learning tasks.	High flexibility, low power consumption, customizable for specific architectures.	Development complexity requires expert programming and lower performance than GPU and TPU.
Application-Specific Integrated Circuits (ASIC)	Specially designed chips for executing specific DL tasks.	High performance, optimized for specific deep learning models.	Expensive production, limited versatility, and high costs for design and development.
Neuromorphic Processors	Mimic the structure of the human brain for computations, especially effective for neural networks.	Energy-efficient, high speed for complex neural network tasks.	Low availability, complex integration, and limited support for standard DL libraries.

Therefore, selecting a hardware platform depends on the task's specifics, with speed, scalability, and energy efficiency as key factors.

Discussion

The challenges and prospects of implementing artificial intelligence (AI) algorithms in field-programmable gate arrays (FPGA) and application-specific integrated circuits (ASIC) represent a relevant topic, especially in light of increasing computational power demands and task complexity. One primary challenge is the high complexity of developing and optimizing AI algorithms for hardware solutions. Unlike standard software implementations, FPGA and ASIC require detailed configuration and optimization for specific tasks, which can be a labor-intensive process. Furthermore, designing such solutions demands a deep understanding of hardware component architecture and specialized modeling tools, which limits the pool of developers and increases development costs. Another issue is the limited flexibility of ASIC. Despite their high performance, they cannot be reconfigured for new algorithms, making their use challenging in rapidly evolving AI fields.

On the other hand, the prospects for using FPGA and ASIC for AI algorithms are significant. FPGA offers greater flexibility than ASIC, allowing architecture adaptation for specific tasks while maintaining high energy efficiency and performance. This makes FPGA attractive for resource-constrained systems such as unmanned devices or Internet of Things (IoT) systems. In contrast, ASIC provides maximum performance with minimal power consumption, crucial for deep learning tasks requiring real-time processing of large data volumes. In the future, further improvements in development tools and design technologies could simplify the process of integrating AI algorithms into FPGA and ASIC, making these solutions more accessible for widespread application across various industries.

An example is the Xilinx platform, which has introduced the ML Suite tool package to optimize and deploy machine learning tasks on FPGA-based systems. Xilinx ML Suite supports popular frameworks such as Caffe, MxNet, and TensorFlow, offering convenient interfaces for working in Python and through REST API. Notably, the xDNN inference processor, designed for high-performance, energy-efficient neural network operations, stands out as one of the best alternatives to traditional CPU and GPU for real-time task execution [8].

The development toolkit based on Intel solutions offers new possibilities for optimizing neural networks and deploying them on various Intel computing platforms. Specifically, Intel's Open Visual Inference & Neural Network Optimization (OpenVINO™) package provides

tools for converting and configuring models developed with frameworks like TensorFlow™, MXNet, and Caffe, allowing efficient use of standard Intel devices and accelerators. This approach enables the deployment of models on a variety of target platforms, including central processing units (CPU), graphics processors with integrated graphics, and specialized devices such as Movidius and FPGA. The toolkit allows for model testing and adaptation to achieve an optimal balance between cost and performance across different Intel hardware types, significantly enhancing development flexibility and efficiency [9].

Compared to ASIC, FPGA offers faster reprogramming and adaptability to changing conditions, which is particularly relevant in the rapidly evolving AI field. The prospects for FPGA use are associated with their capabilities for integration into systems with irregular parallel computing and unique data types, making them competitive with traditional processors and graphic accelerators in deep learning and computer vision tasks.

Conclusion

In conclusion, this study identified key aspects of implementing artificial intelligence algorithms on hardware platforms such as FPGA and ASIC. FPGA demonstrated high flexibility and the capability for rapid reconfiguration, making them preferable for tasks requiring adaptation to changing conditions. At the same time, ASIC stands out for its high performance and energy efficiency, particularly in specialized applications, making it more effective for well-defined tasks such as deep learning. However, both types of platforms face challenges, including development complexity and resource constraints, necessitating highly specialized approaches to design and optimization. Thus, the choice between FPGA and ASIC should be based on the specific requirements of the project, balancing flexibility, performance, and energy efficiency.

References:

1. Gupta, N. (2021). Introduction to hardware accelerator systems for artificial intelligence and machine learning. In Advances in Computers (Vol. 122, pp. 1-21). Elsevier.
2. Boutros, A., Arora, A., & Betz, V. (2024). Field-Programmable Gate Array Architecture for Deep Learning: Survey & Future Directions. arXiv preprint arXiv:2404.10076.
3. Seng, K.P., Lee, P.J., & Ang, L.M. (2021). Embedded intelligence on FPGA: Survey, applications and challenges. Electronics, 10(8), 895.
4. Perepelitsyn, A. (2023). Method of creation of FPGA based implementation of Artificial Intelligence as a Service. Radioelectronic and Computer Systems, (3), 27-36.
5. Mishra, A., Yadav, P., & Kim, S. (2023). Artificial intelligence accelerators. In Artificial Intelligence and Hardware Accelerators (pp. 1-52). Cham: Springer International Publishing.
6. Zhang, Z., & Li, J. (2023). A review of artificial intelligence in embedded systems. Micromachines, 14(5), 897.
7. Machupalli, R., Hossain, M., & Mandal, M. (2022). Review of ASIC accelerators for deep neural network. Microprocessors and Microsystems, 89, 104441.
8. Amuru, D., Zahra, A., Vudumula, H.V., Cherupally, P.K., Gurram, S.R., Ahmad, A., & Abbas, Z. (2023). AI/ML algorithms and applications in VLSI design and technology. Integration, 93, 102048.
9. Badkhutia A. et al. (2024). Programmable gate array in the field: an extensive overview, recent trends, problems and applications //11th International Conference 2024 on Computing for Sustainable Global Development (INDIACom). – IEEE, 1084-1090.