International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimized Vlsi Architectures For AI-Enabled IoT Systems

Bushra Fatima Masih[1], Mohammed Nomaan[2]

[1,2]BE ECE MJCET OU Hyderabad, Telangana, India

Abstract

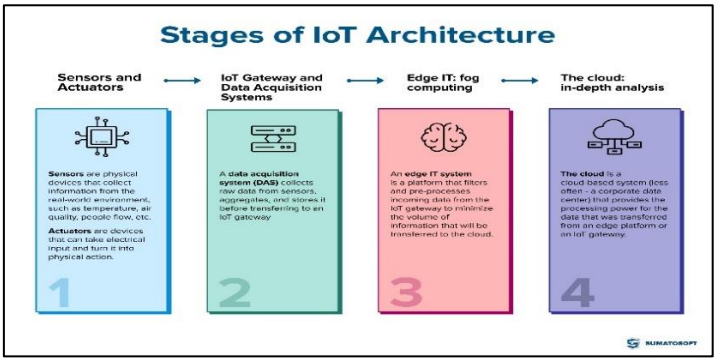
The integration of AI in IoT has revolutionized industries through the development of intelligent decision-making and automation. However, the deployment of AI is limited by the power, computational capacity, and cost of IoT devices. VLSI architectures help to meet these challenges through compact, low power consumption, and high-performance hardware solutions. This paper focuses on the advancements in VLSI architectures for AI in IoT, low-power design techniques, hardware accelerators, neuromorphic computing, and approximate computing. The challenges, design considerations, and future trends are discussed to present a comprehensive understanding of how VLSI enables efficient and intelligent IoT systems.

Keywords: VLSI Architectures, Artificial Intelligence (AI), Internet of Things (IoT), Low-Power Design, Hardware Accelerators, Application-Specific Integrated Circuits (ASICs), Field-Programmable Gate Arrays (FPGAs), Neuromorphic Computing, Approximate Computing, Edge AI, Low Power Design Techniques.

1. Introduction

1.1 Background and Motivation

It is growing so fast that there has been a proliferation of such devices generating humongous data, thereby making traditional cloud-based processing models inadequate because of issues in latency, bandwidth, and energy consumption. AI, integrated into IoT systems, offers real-time analytics and autonomous decision-making capabilities, but it requires computational efficiency and scalability. These demands require specialized hardware solutions that could operate within the constraints of IoT devices.



1.2 Role of VLSI Architectures

VLSI architectures are emerging as the critical enablers for AI in IoT. VLSI enables computational efficiency and energy savings to be packaged into millions of transisto rs within compact chips that support real-time AI processing. Such architectures enable the autonomous operation of IoT devices, which operate with reduced cloud resources, meeting latency and power requirements.

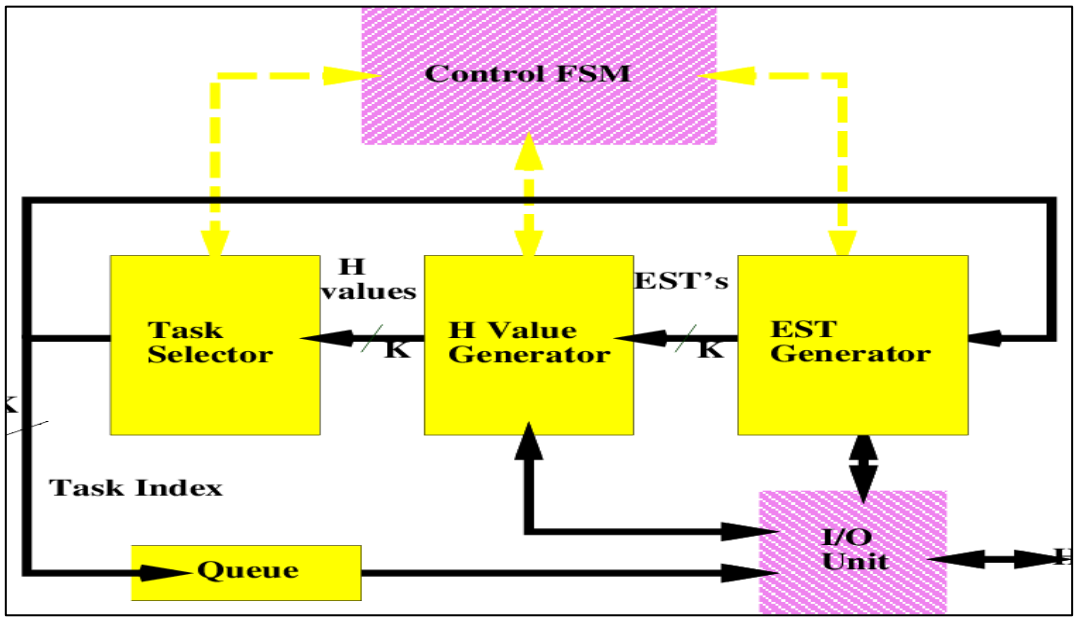


Fig 1: VLSI architecture

1.3 Objectives

This paper delineates the critical role that VLSI plays in optimising AI for IoT, which is underlined by technological advancement along with inherent challenges and emergent opportunities in this direction. It explains how a VLSI architecture might solve the specific constraints arising in IoT devices, power, computational capacity, as well as cost efficiency so that complex AI workloads may be executed. The paper, focusing on innovative solutions like low-power design techniques, hardware accelerators, neuromorphic computing, and approximate computing, makes available a comprehensive understanding about how these architectures support real-time analytics, scalability, and intelligent decision-making. The article also examines the challenges involved in the design of such architectures concerning thermal management, reliability, and security concerns and provides an overview of future trends that promise to shape the evolution of AI-enabled IoT systems. The paper presents this analysis to underpin the fact that VLSI technologies could be the enabler in the development of efficient, scalable, and intelligent IoT ecosystems paving the way for transformative applications in industries.

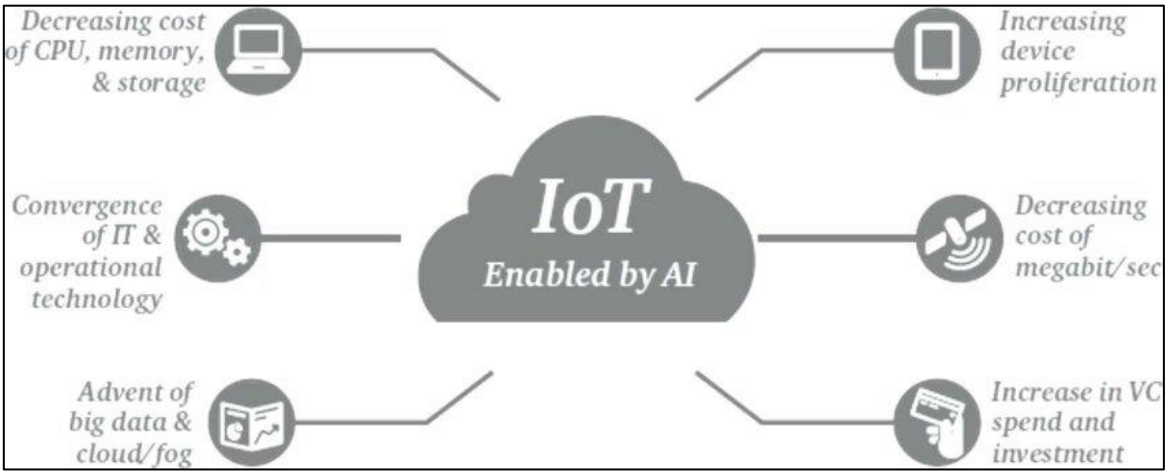
2. Literature Review

The literature on VLSI architectures for AI-enabled IoT systems shows vast improvement in addressing the demands of computation and energy needed in IoT devices. Techniques in low power design, such as dynamic voltage scaling and subthreshold operations, increase the efficiency of energy usage. ASICs, FPGAs, and systolic arrays support the high-performance processing of AI by hardware accelerators.

Neuromorphic computing and approximate computing offer innovative ways to provide energy-efficient and real-time IoT applications, though the scalability and precision issues persist. Co-design of AI algorithms and VLSI architectures improves edge AI systems, reducing latency and improving privacy. Emerging trends, such as 3D integration, AI-driven design, and quantum principles, hold promise for the future, emphasizing VLSI's pivotal role in advancing scalable and intelligent IoT systems.

2.1 AI in IoT Systems

According to consistent research findings, AI algorithms in IoT applications consume very high computational resources due to the necessity of processing and making decisions in real time. Automation also calls for processing data in real-time and thereby cannot be effectively managed using traditional cloud-based models where data is transmitted to a centralized server for processing. These models are no longer viable because they inherently involve high latency and even open doors to privacy vulnerabilities. These constraints are particularly critical for applications requiring instantaneous responses, such as autonomous systems, healthcare monitoring, and industrial automation. The increasing need for localized processing has thrust edge computing and VLSI-based architectures into the forefront as a viable solution. These approaches allow IoT devices to process data closer to the source, reduce latency, minimize energy consumption, and enhance data privacy by limiting external transmissions. VLSI architectures play a fundamental role in enabling these progressions through providing compact energy-efficient hardware of high-performance for AI-enabled IoT systems meeting stringent requirements.



2.2 VLSI for AI in IoT

There is vast research to develop low-power techniques and hardware solutions that are specialized in addressing the constraints of such unique systems to optimize AI inference tasks in IoT environments. Dynamic voltage scaling perhaps is one of the most promising strategies that can be used to minimize the power consumed by the chip through dynamic adjustments of the supply voltage and the clock frequency in relation to workload demands so that the desired performance does not conflict with energy efficiency. Moreover, the Application-Specific Integrated Circuits (ASICs) and Field-Programmable Gate Arrays (FPGAs) have helped improve computational efficiency and processing speed for AI tasks. These accelerators are dedicated to handling AI workloads, such as neural network inference, with minimal energy usage. Apart from the above methods, newer technologies such as neuromorphic computing and approximate computing are receiving attention as promising solutions in energy-efficient processing. Inspired by the architecture and functionality of the human brain, neuromorphic computing provides an

event-driven approach to computing especially well-suited for the real-time applications of the IoT. Additionally, the approximate approach allows for reduced precision in calculation toward significant savings in terms of energy but still achieves acceptable accuracy for various analyses and recognitions of sensor data. Together, these developments reflect a growing trend toward convergence of innovative hardware and algorithmic innovations to address problems thrown up by AI-enabled systems of IoT.

2.3 Challenges and Emerging Trends

These pose significant challenges to balancing performance, power consumption, and cost efficiency with scalability in AI-enabled IoT systems. The constraints of IoT devices, such as limited energy resources, computational capacity, and affordability, demand novel solutions that do not compromise on efficiency or functionality. A multidisciplinary approach that integrates advanced VLSI design principles with novel system architectures would help address these challenges. Emerging trends such as 3D integration are showing promise in overcoming these limitations by enabling the stacking of multiple chip layers, which reduces interconnect delays and improves data throughput while providing overall performance in a compact form factor. Hybrid architectures combining VLSI-based designs with edge computing paradigms are also gaining traction, as they distribute computational workloads between local devices and nearby edge servers effectively. This approach not only reduces latency and energy consumption but also provides greater flexibility in scaling AI capabilities across diverse IoT applications. Advances like these demonstrate the evolving innovation in hardware and system-level design, which bodes well for more efficient, scalable, and cost-effective solutions in the near future of IoT.

3. Key Advancements in VLSI for AI in IoT

Advancements in Very Large Scale Integration (VLSI) have significantly enhanced the capabilities of Artificial Intelligence (AI) in Internet of Things (IoT) applications. Key developments include:

1. **Integration of AI into VLSI Designs for IoT Devices:** Implementing AI algorithms directly into VLSI designs has led to more efficient hardware architectures, supporting machine learning-based applications within IoT devices. This integration optimizes performance and reduces power consumption, which is crucial for IoT devices operating in diverse environments.
2. **Development of AI-Enabled System-on-Chip (SoC) Solutions:** Innovations like the Marsellus SoC demonstrate the potential of heterogeneous RISC-V AI-IoT end-node systems. These systems combine general-purpose clusters of RISC-V DSP cores with reconfigurable binary engines to accelerate deep neural network (DNN) inference, achieving high performance within a limited power envelope.
3. **Energy-Efficient Deep Learning Methodologies for AIoT Applications:** Projects such as VEDLIoT focus on developing energy-efficient deep learning methodologies tailored for distributed AIoT systems. By optimizing algorithms and addressing safety and security challenges, these initiatives aim to enhance the performance and reliability of AIoT applications across various domains.
4. **Fully-Parallel Convolutional Neural Network Hardware Implementations:** Advancements in fully-parallel convolutional neural network hardware have enabled the embedding of complex AI models, like LENET-5, into single FPGA devices. This progress facilitates the deployment of sophisticated AI capabilities in resource-constrained IoT devices, enhancing their functionality and efficiency.

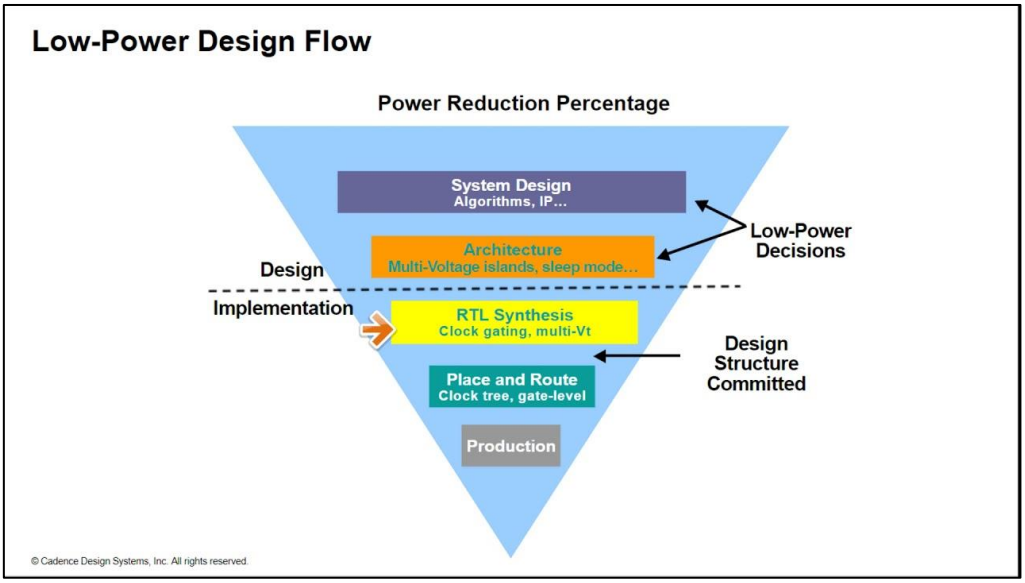
These advancements underscore the pivotal role of VLSI technology in enhancing AI functionalities within IoT ecosystems, paving the way for more intelligent, efficient, and responsive IoT devices.

3.1 Low-Power Design Techniques

A relatively new area of research and development that has gained significant interest in recent years is VLSI Design of Low- Power Edge AI Processors for IoT Devices. It is obvious that an effective and power-conscious approach for integrating AI capabilities into low-power IoT devices is essential given the rising demand for AI-enabled IoT devices. VLSI Design of Low-Power Edge AI Processors for IoT Devices is filling this gap. The latest research in this area will be reviewed, and the prospects and difficulties for pushing forward the creation of VLSI-level AI processors for embedded applications will be discussed [1].Novel approaches for the construction of low-power edge AI processors for IoT devices have emerged at the forefront in recent studies trying to meet these needs. Several papers have been given in this regard that focus on the novel VLSI-based processors for IoT[2]. On-chip neural networks and other special logic circuits are incorporated into a low-power processor that is said to allow effective inference operations. In order to reduce energy Consumption, the processor will utilize dynamic power Management techniques, as well as future memory. Technologies such as resistive RAM and 3D memory hierarchy [3].To support low-latency execution of AI applications on resource a specialized digital signal processor core has been designed for constrained IoT devices. The architecture uses an application-specific instruction set for executing common AI efficiently algorithms and based on a streaming model for dataflow [4].Suggests a low power modular VLSI processor architecture for vision-based IoT applications. The architecture is made up of a general purpose CPU combined with an ultra-low power CNN tower on its hierarchical structure. In its accelerated data processing, programmable computational and memory components can be found in the composition of the CNN tower; by contrast, the control over the device and management over its data have been handled concurrently through its general-purpose processor [5].The development of a processor architecture for edge AI applications. A power-efficient drone management unit and a heterogeneous processing region are included in the architecture in order to facilitate the effective execution of deep learning operations. The implementation of an on-device compiler enables the optimisation of deep learning algorithms. In addition, a custom instruction set is created to facilitate embedded heterogeneous processing strategies and low-level optimisation [6].Finally, an energy-efficient VLSI processor architecture is presented for edge AI applications. To reduce the power consumption, the design incorporates a special-purpose memory accelerator, an adaptive voltage scaling approach, and 3D memory hierarchy. Further, a common AI accelerator fabric is incorporated to efficiently perform deep-learning operations. Overall, the myriad works described here reflect a vibrant field of study with an ever-increasing desire to deliver more energy-efficient VLSI design for low-power edge AI processor in IoT devices [7].A Low-Power Edge AI Processor has emerged as the key study area in both VLSI Design areas and the advancement of IoT Devices due to the rise in the need for smart AI-IoT devices. An edge AI processor properly designed can greatly enhance the accuracy, latency, and power economy. In the essay, many studies related to VLSI design for low-power edge AI processors for Internet of Things devices will be reviewed. First, in the earlier studies, efforts were mainly put on cutting down energy delay consumption by optimized VLSI architectures. A balanced energy-delay-accuracy architecture for AI accelerators that can be based on in-situ energy-over-accuracy (EoA) optimization. To optimize energy, the proposed methodology involves weight sharing,low-precision data instructions, and sparse data flow mapping. As experimental results show, the new technique can attain energy-delay trade-off up to 30 times better compared to the conventional methods.

Second, VLSI designs have also been proposed for power-efficient AI processor through techniques like neuron and workload pruning. A dynamic neuron and workload pruning-based AI processor network

designed particularly for edge AI applications [8].This method exploits fewer neurons that analyze fewer data samples to maintain all the functions at the high level, which can decrease the energy-delay by 35% by optimizing and updating the neuron shape online during task demands. Thirdly, there have also been extensive studies on protocols developing for low-power communication inside AI computers. A ML architecture for edge AI developed based on federated learning. This architecture takes advantage of distributed job distribution among many nodes does not require large datasets, and it is particularly designed for low-power communication in fog node networks. This approach reduces the risks involved with data privacy and ownership and also enables better device processing. The experimental results on some representative applications are promising. Lastly, the integration of a few functions into low-power devices is a significant VLSI design issue. AI-IoT device implementation using an AI chip and a multi-function integration method. This technology implements cutting-edge techniques for AI-IoT applications by integrating hardware, software, and various sensing capabilities into one device. The proposed device gains 9.8% increase in processing speed with 6.9% improved power efficiency compared to traditional approaches. This paper has analyzed some of the recent studies on VLSI design for low-power edge AI processors for the Internet of Things [9].These above described design methodologies comprise energy-delay optimisation, neuron/workload pruning, low power communication protocol design, multi-function integration. Hence in the near future, design methodologies are likely to become even more detailed and sophisticated in creating such design methods such that low-power edge AI processors for IoT-based applications may be able to provide the maximum output.



3.2 Hardware Accelerators

Hardware acceleration strategies have been the effective solution to the large-scale acceleration of deep neural networks. For the large number of computational resources demanded, the traditional CPU-based systems often cannot meet it efficiently; thus, some specialized hardware accelerators have been developed and utilized in a large number of ways, such as Graphics Processor Units (GPUs), Neural Processing Units (NPUs), among others. These accelerators are designed to address the parallelizability of neural network computation, greatly improving the computational efficiency. Hardware acceleration has been a significant development that transformed deep learning, and hardware acceleration has

supported a variety of computational models such as image recognition and speech recognition. Models like Convolutional Neural Networks (CNNs) and advanced architectures such as TResNet have gained significant performance gains through hardware acceleration [10].Advantages of hardware-accelerated deep learning models are many in computer vision. From self-driving cars with real-time image processing up to medical imaging systems needing a high-precision diagnosis, enhancements to hardware-accelerated models help get more complex and accurate work done. This, on the other hand, paves the way for new innovations and allows researchers and engineers to develop more sophisticated algorithms and applications.

This gives an overview of some of the outstanding hardware components that have been designed or adapted to accelerate DNN-based computer vision algorithms, including FPGAs, GPUs, and other advanced mobile hardware platforms. This includes a discussion of how this hardware performs in practical applications, as well as its architectural features and optimization techniques. DNN methods based on which tasks such as image segmentation, object recognition, and image classification have been widely reviewed in the literature [11].The original goal of developing a GPU was to speed up graphics processing. A GPU is designed specifically for jobs like integrated transform, lighting, and rendering. In addition to being a potent graphics engine, a modern GPU is a highly parallelized compute engine that can perform computations in enormous parallel with tremendous memory bandwidth and throughput. GPGPU is general-purpose graphics processing unit. The technology expands the usage of GPUs in applications beyond their usual traditional graphics rendering workloads. It has the utility in speeding up data-intensive applications like machine learning and scientific computing. The components of GPGPUs are thousands of in-order cores, running at reduced frequencies and rely on small-size caches, whereas the multicore CPUs are typically out-of-order, multi -instructional processors. Using various development platforms, including but not limited to Compute Unified Device Architecture (CUDA), high-performance GPU-accelerated programs can be developed based on parallel programming. Some typical aspectations of applications in such systems are GPU-accelerated embedded devices, data centers, and high-performance servers, among others. There are various hardware suppliers who have made GPUs, with Nvidia, Intel, and AMD/ATI as the big names in the market currently. These GPUs are suitable for a large number of special applications.In the last few years, the neural network technology has expanded at an incredible speed, and most of these improvements have directly or indirectly benefited from the advancement of GPU computing. Some of the very significant improvements include the following: For example, CNNs have multiple ways to exploit algorithmic parallelism [12]: (i) Parallelize matrix convolution operations using kernels; (ii) Independently execute different subsampling and pooling procedures; (iii) To simultaneously activate every neuron in the fully connected layer, employ binary tree multipliers. Because of their strong floating-point computing capabilities and effective parallel processing architecture, GPGPUs are seen to be the ideal choice for speeding up deep learning. Being able to handle copious volumes of data concurrently enhances the Training and reasoning speed for deep neural networks, especially with intricate models and large datasets. Probably due to this reason, actual GPU usage as a network quality metric in recent developments of neural network design paradigms have been neglected by researchers [10].GPU training speed and the maximum batch size are often overlooked, although sometimes inference speed is measured. However, speed and largest batch size are all relevant factors in real world applications. Low GPU utilization may result in wasted resources and increase both the training time and the cost. Thus, enhancing GPU utilization not only makes training faster but also minimizes energy consumption and the operation cost. A model called TResNet improves on those areas by carefully

designing some of those factors [10]. TResNet makes use of several architectures specific optimisations and engineering techniques that make the most out of the GPU utilization and throughput.

GPUs are quite popular for DNN hardware acceleration and provide very fast throughput. However, they are not so suitable for energy-constrained applications like Internet of Things devices because to their enormous power consumption. DNN acceleration is therefore shifting to different approaches based on FPGAs that consume less energy. FPGAs provide flexible parallel processing capabilities, adaptable data types, and specific to the application hardware designs, which can easily adapt to the most recent DNN models, with greater sparsity and compact network architecture. FPGAs have been increasingly popular in deep learning applications over the last few years, given that their benefits in high-performance, low-power computing are now more apparent. FPGAs' programmability allows them to be set up for specific DNN models, thereby optimally utilizing hardware and using up minimal power. Due to this adaptability, FPGAs find applications in embedded systems and IoTs, where low-latency responsiveness and energy-efficiency are of utmost importance. In addition to all these, FPGAs offer dynamic reconfigurability, which lets reprogramming at runtime: the hardware to meet varying workload and optimization requirements. This gives FPGAs a marked advantage in handling different DNN jobs. An FPGA consists of many types of programmable logic blocks that are connected through a hierarchy of adjustable interconnects. Block RAMs, LUTs, DSPs and other key elements like flip-flops are common components of modern FPGAs. Fig. shows an average FPGA architecture for DNN implementation [13].

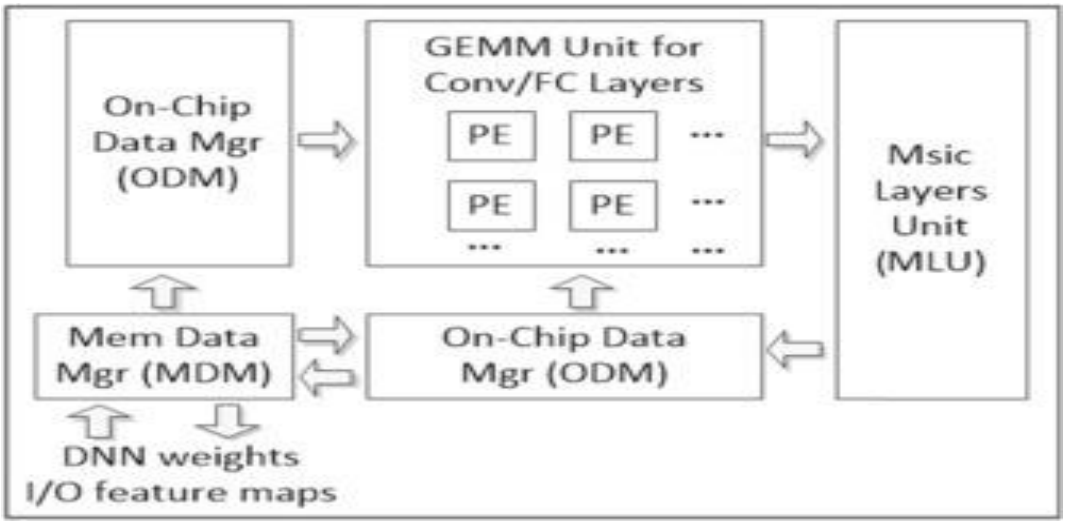


Fig 2: An actual example of a FGPA architecture that is utilized to accomplish DNNs

In real-world applications, DNN models are usually trained or optimized on high-performance computing platforms such as GPUs, and DNN inference is primarily conducted on input data using FPGA accelerators based on pre-trained DNN models. The DNN training process may be significantly sped up due to the high bandwidth and powerful parallel processing capabilities of GPUs, especially when working with large-scale datasets and sophisticated models. FPGAs are particularly well-suited for edge computing and real-time applications because they have low power consumption, great energy efficiency, and minimal inference latency when the DNN model is deployed to an FPGA for inference after training is finished.

The ability to use high-level synthesis (HLS) tools with automatic DNN algorithmic compilation capability described in a programming language into a Hardware Description Language (HDL) and then generates low-level hardware configurations for use on FPGAs, has proven to make the task of porting

DNN models from GPUs to FPGAs. This approach drastically reduces design complexity, enabling more developers to profit from FPGAs. Some strategies for accelerating DNNs with FPGAs include the use of highly parallel architectures to process multiple neurons and layers simultaneously, as well as the optimization of specific computations such as convolution operations and matrix multiplication through customized hardware designs to improve computational efficiency and reduce latency. The flexibility of FPGAs allows for specialized optimization of various models to achieve optimal performance. A hardware accelerator proposed in the literature effectively reduces the complexity of the design by merging and reusing certain types of data flow and control logic [14].Google developed the Tensor Processing Units (TPUs) [15,16], an ASIC with specificity made specifically for the TensorFlow framework and neural networks. Large amounts of low-precision targeted to 8-bit data etc. Many Google apps, such as AlphaGo and the search engine, already powered by it [16].The Nervana neural network processor, a product of Intel [17] is to maximize the efficiency of its fundamental hardware components while also offering the necessary flexibility for deep learning primitives.

Moreover, graph core's IPUs also lead in computer vision tasks. The architecture of IPUs enables massively parallel computation on-chip, which is the basis for efficient execution of CNNs and other models. the IPUs can dramatically increase inference speed and power efficiency through high-bandwidth communication between processing elements.

Aside from raising efficiency and performance, this utilization of ASIC hardware accelerators in computer vision neural networks stimulates the creation of new models and algorithms to facilitate the execution of more complicated and precise vision tasks. These developments imply that ASICs will remain necessary in the development of deep learning and artificial intelligence.

3.3 Neuromorphic Computing

Building a machine that can do more in less time than human beings was an objective for computer design for years and, now, the von Neumann architecture has clearly become the standard architecture for this kind of machine. Nonetheless, inevitable comparisons to the human brain point out radical differences in the structural organization, power consumption, and even processing powers of both. This in turn prompts a natural question: whether a different architecture might be developed based on the neurological models, that compare quite to a biological brain.

A neuro-inspired computing chip emulates the structure and operation of the biological brain and represents an innovative and assured approach to the development of intelligent computing[18]. Artificial Intelligence workloads, these neuro-inspired computing chips are expected to provide advantages in power efficiency and computing power over traditional systems [19].Over the past few years, a spread of neuro-inspired computing chips have been developed[20]. Several neuro-inspired innovations have been incorporated into these chips at several levels, from the hardware to the circuitry to the architecture [21]. This is still at a beginning stage in the development of neuro-inspired computing chips, so exploring the hurdles and opportunities for the field is important [22].Neuromorphic computing has, in recent years, emerged as a complementary architecture to von Neumann systems. The term neuromorphic computing was coined in 1990 by Carver Mead [23]. At the time, Mead referred to very large scale integration (VLSI) with analog components that mimicked biological neural systems as "neuromorphic" systems. More recently, the term has come to encompass implementations that are based on biologically-inspired or artificial neural networks in or using non-von Neumann architectures.

These neuromorphic architectures are distinguished by their high connectivity and parallelism, low power,

and co- location of memory and processing. Although interesting in their own right, neuromorphic architectures have gained increased attention recently due to the approaching end of Moore's law, the increasing power demands associated with Dennard scaling, and the low bandwidth between CPU and memory known as the von Neumann bottleneck [24].

Neuromorphic computing systems exhibit the capability to execute intricate computational tasks with enhanced velocity, superior energy efficiency, and a reduced spatial footprint in comparison to conventional von Neumann architectures. Such attributes present persuasive justifications for the advancement of hardware utilizing neuromorphic design principles.

Deep neural networks (DNNs) and associated hardware have demonstrated significant advancements in accuracy across a diverse array of large-scale classification challenges, with some instances even exceeding the performance capabilities of human operators [25]. However, to attain enhanced training efficiency, the parameters of the DNN models increase exponentially, culminating in the necessity of managing hundreds of millions of parameters alongside extensive training datasets within memory systems. In the conventional von Neumann architecture, data is required to be transferred repeatedly between memory and the processor, thereby imposing constraints on the energy efficiency of hardware when executing these machine learning tasks [26,22,27].

The notion of adaptive parallel processing within biological neural networks (BioNNs), facilitated by neuro-inspired computing, is posited as a solution to mitigate the energy-consuming and ineffective transmission characteristic of von Neumann-based architectures.

Digital-bit-encoded artificial neural networks (ANNs) and their exponential possibilities are based on two major paradigms of neuro-inspired computing and spiking neural networks (SNNs) with spike-timing-encoded inputs. Deep neural networks (DNNs) and convolutional neural networks (CNNs), also known in the context of time-series logic as recurrent neural network (RNN). The data are then used to develop various commercial applications, including intel computing, machine vision, intelligent search, and automatic driving .[28,19]

Algorithms of the brain. The human brain is a dynamic, reconfigurable system of interconnected neurons connected by synaptic connections [29]. For physicists, it allows for interesting phenomena like energy minimization (or entropy minimization), phase transitions, criticality, self-oscillation chaos, synchronization, stochastic resonance, etc.[18].Theoretical efforts to clarify the algorithms the brain uses have involved physicists since the beginning, as many of these ideas are drawn from the two pillars of computer science and computational neuroscience. Statistical physics, nonlinear dynamics and complex systems theory shed light on neural mechanisms that enable learning. Hopfield networks and Boltzmann machines, derived from Ising spin systems, are the best-known ones, but many others have been suggested, making use, in particular, of nonlinear dynamics for computing [19,27,22].

Neuromorphic computing influences the brain in that it creates energy-efficient hardware for information processing capable of highly sophisticated tasks. Systems built with normal electronics gain on pace and energy by mimicking the dispersed topology of the brain. Such arrangements should be revolutionized in hardware into a new scale to improve energy consumption, performance, and speed by several orders of magnitude. Neuromorphic computing can be greatly enhanced with the addition of even more physics into algorithms and nanoscale materials. The performance conditions for NVM devices to neuro-inspired computing chips depend mostly on specific systems and applications. Thus, a number of analog states gives further the weight tuning resolution. Reports indicate that for training with relatively large neural networks45, such as ResNet46, at least eight equivalent bits accuracy would be needed. Some of the

conditions of performance for NVM devices dedicated to neuro-inspired computing chips are very specific to types of systems or applications. The number of analog states determines the weight tuning precision because it has been reported that the precision of at least eight equivalent bits has been required for training a rather large neural network45, like ResNet46. Current electronics simply cannot match that. In the brain neurons, which roughly can be seen as doing processing, have direct access to memory through synapses. Current electronics, on the other hand, cannot help structuring memory and computation as entirely separate physical units, for which data must be exchanged across. This "von Neumann bottleneck" is an issue for artificial intelligence algorithms that read a lot of data at every step, perform complex operations on that data and write results to memory. It not only slows down computation, but also sharply increases energy losses during learning and inference.

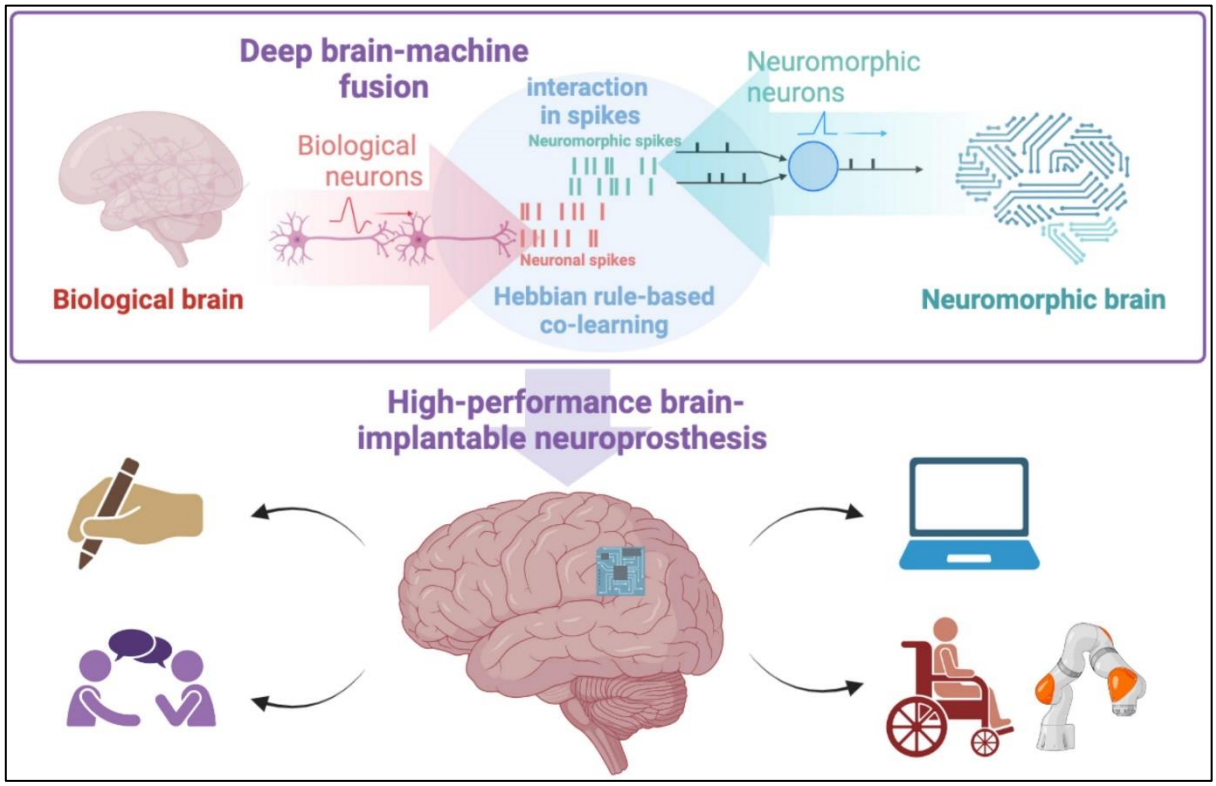
The standard model in neuromorphic computing is to take inspiration from the topology of the brain, thus forming circuits composed of physical neurons that are interconnected by physical synapses that perform memory in-situ, in a non-volatile way, drastically cutting the need to move data around the circuit and providing huge gains in speed and energy efficiency. Unfortunately, this is complicated by using Complementary Metal-Oxide Semiconductor (CMOS) technology alone. For example, dozens of transistors would be needed to imitate each such neuron, and extra outside memories would be needed to execute synapses. Furthermore, this is very inconvenient, as CMOS-based artificial neurons and synapses are typically several micrometers wide. The area in which they can exist is inherently limited. This is problematic because performance of neural networks tends to increase with the number of neurons and synapses: today's image recognition algorithms ideal invocation involves millions of neurons and synapses in average. The number of neurons and synapses needed could be raised by gathering chips. Additionally, the entire system would become unwieldy and would consume significant energy in its interconnects. Nanodevices that are capable of mimicking important features of neurons and synapses at the nanoscale, such as nonlinearity, memory, and learning, are required to build low-power chips comprising several millions of neurons and synapses.

Overall, it is very tough to use CMOS technology to achieve a high degree of interconnection among neurons. The brain features on average about 10,000 synapses per neuron. Such a connectivity level remained far from being reproducible by contemporary electronics. CMOS technology contains mostly two-dimensional (2D) structures; its fanout is also limited, as is the efficient and fair energy supply to components in the circuit. On the other hand, the brain is tri-dimensional (3D). Axons and dendrites of neurons provide a high fan-in/fan-out, and blood efficiently distributes energy to the entire system.

Artificial neurons and synapses have been incorporated into neuromorphic chips, which emulate action potential spikes in the human brain; all the computations handled by the neuromorphic chips are done independently. This translates into a much smarter and energy-efficient computing system. The neuro-inspired computing chips integrate the merits of both neural systems and with this, can provide an energy-efficient alternative to AI computing workloads. Performance metrics for evaluating neuro-inspired computing chips include computer density, energy efficiency, computing accuracy, and on-chip learning ability. Moreover, co-design principles-from the device to the algorithm level-are discussed for neuro-based computing chips using non-volatile memory.

The neuromorphic extended encompasses sensors. Neuromorphic cameras are examples. Notably, researchers are quickly creating different neuromorphic designs exactly as they did in the early days of neural networks. As a result, the line between research tools for neuromorphic computing and industrial adoption is blurry from a practical perspective. Generally speaking, neuromorphic hardware can be

divided into digital, analog, and mixed-signal circuits. Several architectures for simulating and implementing neurons and synapses in hardware have been proposed. Neuromorphic architectures are the most commonly implemented circuitry uses this type. The main advantages of this type of circuitry are the ease of development and low power dissipation, and of course, reusability. The primary reason for the vast popularity of digital over analog neuromorphic architectures is the low development cost that they require. A striking example is the Digital Neural Array, which is basically a large-scale Digital Neural Array; it consists of an array of digital neurons. The DNAs are both for ASICs and FPGAs. The first ones allow, for example, reprogramming and, while the second, higher density and much reliable performance, much less flexibility. But despite analog circuits being less realized than the digital architectures, analog is much better really suited for the movement of neuromorphic systems. Analog implementations have, in many physical properties, capabilities of neuromorphic architectures, and, like SNNs, the architectures that are being used are robust for noise, hence provide an ideal hardware implementation. Such physical characteristics that fall under reliability are asynchronous operation. The best device for analog circuits is the Field-Programmable Analog Array (FPAA). The Field-Programmable Neural Array is an example of a custom-designed approach designed primarily for neuromorphic applications and using programm.



3.4 Approximate Computing

Recently, approximate computing arises as a new design paradigm to explore how computer systems could be improved-more energy efficient, faster, and less complicated-by allowing relaxation with respect to their exact correctness. "Approximate computing exploits the gap created between the requirements of exactness from the applications/users and that provided by the computing system for various optimizations," states Mittal [30].You can look for the system and estimate it that ought to be made better in empowering computers-more energy-efficient, accelerated, and less complicated-by allowing them to

operate with relaxation in terms of their correctness. "Approximate computing exploits the gap between that required by applications/users and that of the computing system in order to achieve various optimizations," stated Mittal.

This newly evolving research effort into approximate computing is mainly spurred by the ongoing advance in two areas

(1) Requirements for electronics with low energy consumption. Certainly, energy efficiency is one of the driving forces of current computer industry with regard to supercomputers on one hand and small portable personal electronics and sensors on the other hand. In this context, approximate computing exploits the fact that applications are inherently error resilient. Errors are not recognizable due to limited human perception capabilities (e.g. in multimedia applications), due to a lack of golden solution for the validation of results (as in data mining applications), or because the user can accept some inaccuracies (e.g., when the battery of a mobile phone is almost depleted, but at least some basic functionality is still requested). Therefore, it can be said that errors or the accuracy of computations can be a design metric and can be traded for performance or power consumption. To accept approximate computing as a normal design paradigm, one needs to have a better understanding regarding the inherent resilience of the application in question across a wider range of applications. One of the studies initiated in this direction has reported that around 83 % of the runtime of applications (that can potentially be made approximate) is spent in computations that could actually be approximated [31]. This result was achieved by a benchmark suite consisting of 12 common recognition, data mining and search applications, paired with representative inputs.

(**2**) **It is apparent that the above-said specific properties are at nanoscale integrated circuits**. Moreover, any recent circuits designed according to the fabrication technologies below 45 nm have been witnessing reliability and uncertainty problems when used at small voltages. The deviations in parameters in the transistors from their average values were no more small because of fabrication variations which occurred at the nanoscale level. Conventional fault-tolerance techniques, for instance the use of redundant components, amount to in efficiencies in commitment to further resources and power for improving the reliability of the circuit. Each component can therefore provide the most efficient trade-off between output quality and energy costs by permitting the computation to have approximation and a graceful degradation at runtime of its performance in the presence of uncertainty or error. Therefore approximate computing is a significant possible solution towards energy efficiency systems which will be inherently unreliable in nature platforms [32], [33]. The approximate computing is found to be at an initial stage of development, but a lot of reports on this subject are there, which indicates very much more into an active research community . All the relevant research communities have been mentioned to include the end-to-end approximation across the entire computer stack below such a proposed term, bringing together microelectronics, circuits, components, architecture, networks, operating systems, compilers, and applications [30],[34]. Approximation will be done for the embedded systems legacy computers, graphics processing units, and field-programmable gate arrays. Expectedly, enormous reductions in energy consumption will be witnessed in data centers and supercomputers [35],[36].

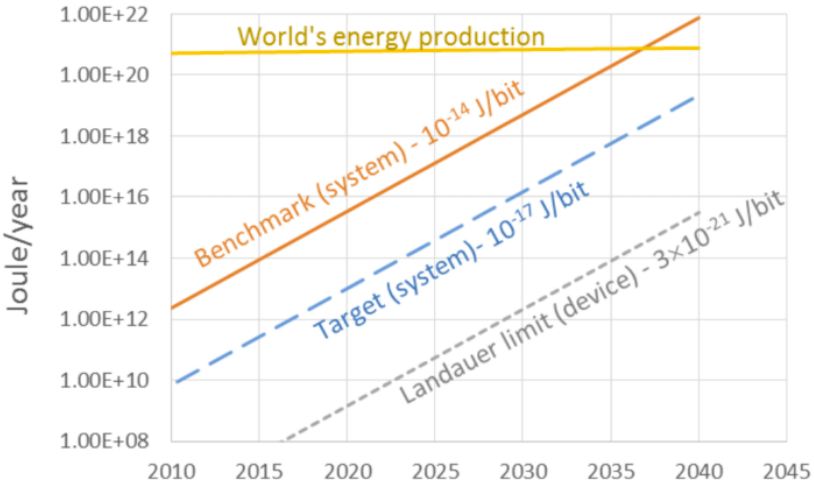


Fig 3 : Energy consumption reduction in (Joules)

The need for functional equivalence between the specification and implementation can certainly be relaxed in approximate computing to achieve power savings along with speed-up, an area reduction, or other realized optimizations in system parameters. Applications amenable to approximate computing can be broadly classified as belonging to four categories [37]:

1. Applications with analog inputs directly inheriting properties from the noisy real-world data.
2. Applications having an output analogue meant for human perception.
3. Applications which do not have a unique answer like those found in web searching and machine learning.
4. Iterative and convergent applications would process huge volumes of data too, and the result will then depend upon the number of iterations being performed.

Mittal [30] explains the following approximation strategies: precision scaling, loop perforation, load value approximation, memorization, task dropping/skipping, memory access skipping, data sampling, using different program versions, using inexact or faulty hardware, voltage scaling, refresh rate reducing, inexact read/write, reducing divergence in GPUs, and also lossily compressing and using neural networks. Out of all above approximations, this paper mainly deals with the approximation of digital circuits. We will hence briefly introduce the principles of over-scaling and functional approximation, which are very popular methods existing in various literatures.

Circuits are designed to work in a supposedly normal environment, with normal power supply and temperature. The consumption of energy can be reduced using voltage over-scaling, that is applying lower energy voltage, by which the circuits are known to sometimes give wrong outputs. To cope with the errors arising from a considerable number of paths near critical points, cell sizing [38], logic restructuring [39], and retiming [40] for improving the path delay distribution of overscaled circuits has been proposed. On the other hand, in over-clocked circuits, performance enhancement occurs. Timing-induced errors occur when a number of paths do not meet delay conditions. The combination of scaling supply voltage and clock frequency is called dynamic voltage scaling.

Functional approximation is taking a not-so-good function for the original (that accurate) one as long as a certain limit of error is acceptable and it optimizes the power consumption or other parameters of the system well enough. Functional approximation has many approaches which will be discussed in later sections. Most often functional approximation goes along with voltage over-scaling [41].

"ad hoc approximations" we refer to several approaches introduced to approximate a certain component, assuming that this method is not a "general" approximation method. Much critical information about a particular system, its typical utilizations, and a quality measurement methodology will add into approximation techniques to yield the best trade-offs for the most important parameters of the circuit.

Emphasis on a specific component approximation process with the stipulation that the method is not one of general approximation. Many specific data on a particular system, its usual uses, and the quality measurement procedure can be incorporated in the respective approach to have the best trade-off for the key parameters of any circuit.

Ad hoc approximations can be applied in different levels of a system as shown in the following examples.

- Transistor level: adders [42], median circuit [43]
- Gate-level: multipliers [44], [45], fault tolerant logic [46]
- RT-level: image filters [44]
- Microarchitecture: pipeline circuits [47]
- Processor: approximate vector processor [48]
- Memory: Multi-Level Cell [49], SRAM [32], memory hierarchy [50], [35]

A heuristic procedure that iteratively modifies the original, accurate (hardware or software) implementation typically develops a functional approximation. Many design automation methods are proposed to approximate digital circuits. Examples of this are: SALSA [51], ASLAN [52], SASIMI [41], ABACUS [53], ABM [54], and genetic programming-based methods [55], [56]. Approximate high-level synthesis (in particular, allocation and scheduling) enabled composing complex circuit systems using approximate components [57].

While ensuring a dynamic approximation (and hence dynamic reconfiguration) at the circuit level in response to varying specifications on the quality of result (for example in image compression), quality configurable circuits can be used by demonstrating an adder where its operands are divided into multiple regions, then each region adds independently before the different region's partial sums conditionally compose into the entire partial addition. Adaptive quality control can be achieved through various design approaches like, SASIMI metod [41] or subcircuits isolation and their approximation in accordance with the requested quality of results [58].

There are many ways to approximate software, as demonstrated in a recent survey [30]. Among these avenues of research, one focuses on extending commonly used programming languages so that they can take part in approximate computing. This encompasses the disciplined approximate programming paradigm that allows programmers to specify that certain portions of the algorithm can be executed approximately. EnerJ [59] is Java-based exten-sions that introduce approximate data types and approximate operations. It assumes the presence in the hardware platform of data storages of different reliabilities and approximate arithmetic units. Approximate data can be processed more cheaply but less acceptably. The system can guarantee statically that the precise program component is isolated from the approximate component.

Another alternative-Axilog developed graph language annotations which provides the necessary syntax and semantics for approximate hardware design and reuse in Verilog [60]. Axilog allows the designer to free some areas of design requirements while keeping other areas strictly zero inaccuracies.

The implementation of these computing kernels has been optimized for reliability and accuracy with respect to programming through the Chisel initiative using integer linear programming approaches targeting approximate hardware platforms [61].

In order to speed up program execution while reducing power consumption, trained ANNs are intended to be substituted for original code written in the imperative language as a general complex purpose code [62]. Moreover, chips directly implementing neural networks might be a good target for these approximation [63].

In comparison with the exact counterparts, approximate solutions are always assessed.

Techniques developed for actual approximation also need to be compared. In the case of software approximations, AxBench-an ensemble of representative applications from different domains-was introduced to explore different aspects of approximate computing and to compare approximation strategies [62]. Unfortunately, benchmark circuits to evaluate the circuit approximation techniques are not available with known error and other parameters.

4. Challenges and Design Considerations

Designing VLSI architectures for AI-enabled IoT requires dealing with a very complex set of challenges in order to meet the requirements of today's applications. The issues arise from the peculiarities of IoT devices, often operating in resource-constrained environments and executing increasingly complex AI computations. We continue below with a detailed discussion of the major challenges and associated design considerations:

1. Power Efficiency

Most IoT devices rely on batteries or harvested energy sources. This means power efficiency is the primary issue. AI computations, and deep learning algorithms in particular, are energy-intensive due to their high computational demand. Hence, the key objective of VLSI architecture design is to optimize power consumption while still maintaining adequate computational performance.

Power Optimization Techniques : Most commonly, low-power design techniques such as dynamic voltage and frequency scaling (DVFS), clock gating, and power gating are utilized to reduce energy consumption. In addition, hardware accelerators specifically designed for specific AI tasks (such as matrix multiplications in neural networks) can enhance energy efficiency by avoiding the overhead of general-purpose processing.

Energy-Aware Algorithms : Another critical aspect is optimizing AI algorithms for hardware efficiency. Techniques such as model quantization, pruning, and knowledge distillation can reduce the computational requirements, thus making it possible to execute efficiently on resource-constrained VLSI platforms.

2. Scalability

IoT applications vary widely, ranging from simple sensor nodes to edge devices running complex AI models. Therefore, scalability is an important consideration in VLSI designs. Architectures need to be able to support diverse workloads and accommodate the increasingly complex AI algorithms without a need for frequent redesigns.

Heterogeneous Computing Architectures: VLSI architecture mainly focuses on heterogeneous computing units such as CPUs, GPUs, and AI accelerators, e.g., TPUs, NPUs for scalability purposes. This heterogeneity can execute general- and AI-specific workloads efficiently.

Reconfigurability and Modularity: Reconfigurable hardware, such as Field Programmable Gate Arrays (FPGAs), can be programmed to handle different AI workloads, providing flexibility for a range of IoT applications. Modular design approaches enable easy integration and scalability across devices with varying performance needs.

3. Cost Constraints

IoT devices are often deployed at scale, making cost a critical factor. Balancing the trade-off between performance and cost is a key challenge in VLSI design for AI-enabled IoT.

Economies of Scale: Leveraging standard semiconductor manufacturing processes and open hardware designs can reduce the cost of production.

Optimized Resource Utilization: The key is to minimize expensive hardware resources, such as memory and processing cores, to ensure adequate performance for AI tasks. Techniques like approximate computing, which allows slight reductions in precision for gains in cost and power efficiency, are gaining traction.

Other Considerations

Security and Privacy: Ensuring secure data processing and transmission within IoT devices adds another layer of complexity. Incorporating hardware-level security features such as encryption engines and secure boot mechanisms is critical.

Thermal Management: As AI workloads grow more demanding, managing heat dissipation in compact IoT devices becomes increasingly challenging. Thermal-aware VLSI designs are critical to maintaining system reliability.

Integration with Emerging Technologies: VLSI designs must incorporate emerging advancements, such as neuromorphic computing, 3D ICs, and emerging memory technologies, to stay future-proof.

To solve these challenges, there will be a need for multi-disciplinary innovation involving innovations in circuit design, architecture, and AI algorithm optimization in order to develop efficient, scalable, and cost-effective VLSI solutions for AI-enabled IoT.

5. Applications of VLSI in AI-Enabled IoT

VLSI technology is a key enabler of AI-enabled IoT systems because it allows for the design and integration of complex hardware components. These architectures enhance the computational capabilities of IoT devices, enabling them to process data intelligently and perform tasks autonomously. Some of the key applications of VLSI in AI-enabled IoT are as follows:

5.1 Healthcare

Wearable devices, in healthcare, rely on very large scale integration architectures that allow for the real-time processing of complex data. The system is embedded with high performance sensors that capture various health-related parameters, including heart rates, blood pressure, levels of oxygen, and temperature of the body. Since these devices are VLSI-based, they can process a large amount of data; analyze them, and store data in compact, low power chips, which makes these devices suitable for continuous and non-invasive patient monitoring.

With the help of VLSI architectures, wearable devices can process the patient data in real-time and give immediate feedback to the users or healthcare providers. Such a feature enables the early detection of possible health issues, such as arrhythmias, hypertension, or respiratory distress, thereby allowing medical intervention in due time. Moreover, these systems usually use wireless communication technologies to send data to cloud platforms or healthcare databases for further analysis and long-term monitoring.

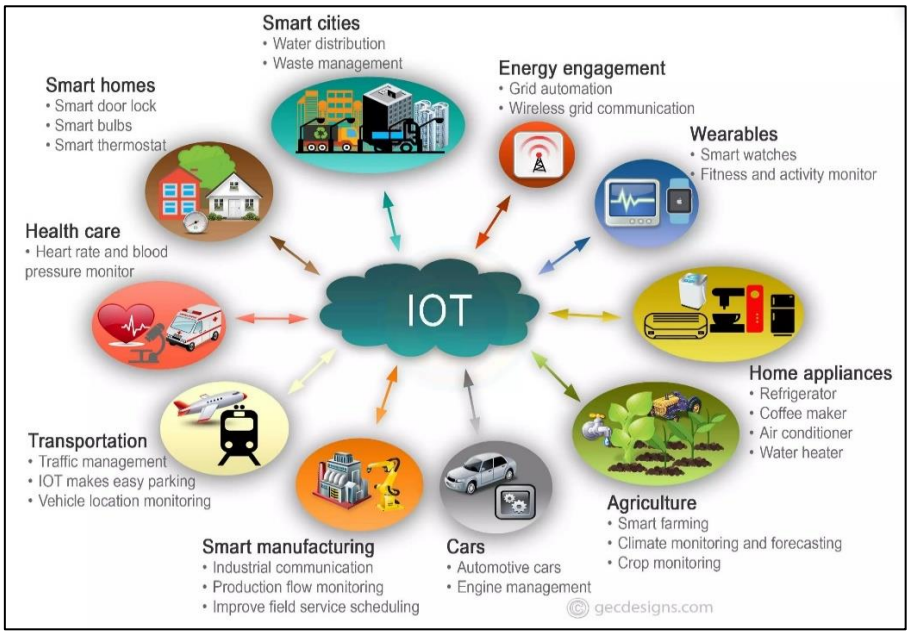
VLSI architectures ensure high performance, energy efficiency, and reliability are crucial in healthcare applications where accuracy and uninterrupted operation are very important. Wearable devices, therefore, become very valuable in modern healthcare and support preventive medicine, remote patient care, and personalized treatment plans.

5.2 Industrial Automation

IoT-enabled factories are revolutionizing industrial automation by connecting devices and sensors in a quest to optimize operation and increase productivity. In the case of smart factories, real-time and efficient processing of massive amounts of data from the IoT system is possible because of VLSI-based accelerators. One of the major applications of the same is predictive maintenance wherein sensors monitor machine parameters like vibration, temperature, and pressure. The collected data is analyzed by VLSI-based accelerators through machine learning algorithms to detect patterns and predict potential failures, which can be proactively maintained and minimize downtime. For process optimization, IoT devices collect data on production workflows, resource usage, and product quality. Accelerators process this information to identify inefficiencies and recommend adjustments for maximum resource utilization and better output quality. By combining IoT's connectivity with the high-speed, energy-efficient computation of VLSI accelerators, these factories achieve smarter, more responsive, and cost-effective industrial operations, paving the way for future innovation.

5.3 Smart Homes and Cities

Smart homes and cities are an enormous quantum leap forward in connected, intelligent environments and VLSI technology stands at the linchpin of these developments. It is by virtue of VLSI that compact energy-efficient chips could be produced for powering whole host devices and systems constituting smart environments. Energy Management-In this, VLSI-based solutions drive smart grids and meters, thus making available real-time monitoring and optimizing of energy usage in homes and urban areas, waste minimization, and sustainability improvement. In security, VLSI makes possible more complex systems like facial recognition, motion detection, and biometric access to offer robust efficient protection for the individual and the public. In infrastructure monitoring, VLSI-powered sensors track parameters like traffic flow, air quality, and structural integrity, enabling predictive maintenance and improved urban planning. VLSI technology underlines the efficient functioning of smart homes and cities, helping to enhance convenience, safety, and sustainability for their inhabitants by supporting seamless integration of IoT devices and intelligent analytics.



6. Future Trends and Opportunities

The future of VLSI technology is to witness radical transformations, with new trends like 3D integration, advanced packaging, and hybrid architectures that include edge computing and VLSI capabilities. 3D integration is the technology where circuits can be stacked in multiple layers vertically, hence offering enhanced performance, efficiency in power, and compactness as compared to planar designs. This innovation is more critical for AI-enabled IoT systems, where high processing power and miniaturization are a must. Advanced packaging techniques, such as system-in-package (SiP) and chiplet designs, further improve functionality by allowing heterogeneous integration of different technologies, including sensors, processors, and memory, into a unified solution optimized for specific IoT applications.

Hybrid architectures blend VLSI with edge computing, enabling real-time, low-latency processing at the device level, reducing their dependence on centralized cloud systems. This approach enhances security, decreases latency, and conserves bandwidth, hence suitable for IoT systems, especially in critical applications such as healthcare, autonomous vehicles, and industrial automation.

At the same time, advances in material science and fabrication technology—such as graphene-based transistors, new semiconductor materials, and advanced lithography processes—promise to take VLSI to the next level of efficiency and scalability. These will redefine what is possible for AI-enabled IoT systems, which will make technological ecosystems smarter, more efficient, and highly integrated.

7. Conclusion

Integration of VLSI architectures in AI-enabled IoT systems has shown to be instrumental in the surmounting of significant challenges that have emerged related to power efficiency, scalability, and performance. VLSI technology enables the production of compact, energy-efficient, and high-performance chips with the capacity to handle localized data processing. This capability decreases latency, improves security, and minimizes bandwidth requirements since data is processed closer to its source, which is fundamental for intelligent and autonomous IoT ecosystems. VLSI empowers applications such as smart cities, industrial automation, and healthcare systems by supporting real-time analytics and decision-making. The future will be further enhanced by continued advancements in VLSI design, materials, and fabrication techniques. Interdisciplinary collaboration across engineering, material science, and computer science will drive innovation in areas like 3D integration, advanced packaging, and hybrid architectures. These developments will ensure that VLSI remains at the center of AI-enabled IoT; they will enable smarter, more efficient, and highly interconnected systems that redefine living and working.

8. References

1. A. Roohi, “IRC Cross-Layer Design Exploration of Intermittent Robust Computation Units for IoTs”, Proceedings of IEEE Computer Society Annual Symposium on VLSI, pp. 354-359, 2023.
2. J.H. Kim, S. Yoo and J.Y. Kim, “South Korea’s Nationwide Effort for AI Semiconductor Industry”, Communications of the ACM, Vol. 66, No. 7, pp. 46-51, 2023.
3. Paul R. Gray,“Analysis and Design of Analog Integrated Circuits”,Wiley, 1993.
4. Kyung Ki Kim and Yong-Bin Kim, “A Novel Adaptive Design Methodology for Minimum Leakage Power Considering PVT Variations on Nanoscale VLSI Systems”, IEEE Transactions on Very Large Scale Integrated Systems, Vol. 17, No. 4, pp. 517-528, 2009.
5. S. Gupta and S. Vyas, “Contemporary Role of Edge-AI in IoT and IoE in Healthcare and Digital Marketing”, CRC Press, 2022.

6. X. Wang, P. Zhou and D. Yu, "Hyperchaotic Circuit based on Memristor Feedback with Multistability and Symmetries", Complexity, Vol. 56, pp. 1-12, 2020.
7. Xiaoyuan Wang, Chenxi Jin, Jason K. Eshraghian, Herbert Ho-Chinglu and Congying Ha, "A Behavioral SPICE Model of a Binarized Memristor for Digital Logic Implementation", Circuits, Systems, and Signal Processing, Vol. 40, pp. 682-2693, 2021.
8. P. Haribabu and D. Das, "RISC-V Core for Ethical Intelligent IoT Edge: Analysis and Design Choice",Proceedings of International Symposium on Cluster, Cloud and Internet Computing, pp. 110-117, 2023.
9. C. Delacour, M. Abernot and A. Todri Sanial, "Energy- Performance Assessment of Oscillatory Neural Networks Based on VO $ _2 $ Devices for Future Edge AI Computing", IEEE Transactions on Neural Networks and Learning Systems, Vol. 23, pp. 1-14, 2023..
10. Ridnik T, Lawen H, Noy A, et al. Tresnet: High performance gpu-dedicated architecture [C]//proceedings of the IEEE/CVF winter conference on applications of computer vision. 2021: 1400 - 1409.
11. Feng X, Jiang Y, Yang X, et al. Computer vision algorithms and hardware implementations: A survey[J]. Integration, 2019, 69: 309 - 320.
12. Magnus Halvorsen, Hardware Acceleration of Convolutional Neural Networks, MS thesis, Norwegian University of Science Technology, 2015.
13. Eriko Nurvitadhi, et al., Can FPGAs beat GPUs in accelerating next-generation deep neural networks? in: International Symposium on Field-Programmable Gate Arrays,2017.
14. Wu X, Ma Y, Wang M, et al. A flexible and efficient FPGA accelerator for various large-scale and lightweight CNNs [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 69 (3): 1185 – 1198
15. Joe Osborne, Google's Tensor Processing Unit Explained: This Is what the Future Computing Looks like, TechRadar, 2016.
16. Norm Jouppi, Google Supercharges Machine Learning Tasks with TPU Custom Chip, Google Could, 2017.
17. Naveen Rao, Intel Nervana Neural Network Processor (NNP) Redefine AI Silicon, Intel Website, 2017.
18. Zhang, W., Gao, B., Tang, J. et al. Neuro-inspired computing chips. Nat Electron 3, 371–382 (2020). https://doi.org/10.1038/s41928-020-0435-7
19. Wang, P., Yu, S. Ferroelectric devices and circuits for neuro-inspired computing. MRS Communications 10, 538–548 (2020). https://doi.org/10.1557/mrc.2020.71
20. S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," in Proceedings of the IEEE, vol.106, no. 2, pp. 260-285, Feb. 2018, doi: 10.1109/ JPROC.2018.2790840.
21. https://www.researchgate.net/publication/339840879_Physics_for_Neuromorphic_Computing
22. Zhang, Y.; Qu, P.; Ji, Y.; Zhang, W.; Gao, G.; Wang, G.; Song, S.; Li, G.; Chen, W.; Zheng, W.; et al. A system hierarchy forbrain-inspired computing.Nature2020,586, 378–384
23. C. Mead, "Neuromorphic electronic systems," Proceedings of the IEEE, vol. 78, no. 10, pp. 1629–1636, Oct 1990.
24. D. Monroe, "Neuromorphic computing gets ready for the (really) big time," Communications of the ACM, vol. 57, no. 6, pp. 13–15, 2014.
25. Danijela Markovic,Alice Mizrahi,Damien Querlioz,Julie Grollier:

26. Buesing et al.(2011) Neural Dynamics_as_Sampling_A_Model_for_Stochastic Computation_in_Recurrent Networks_of Spiking Neurons
27. Ao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, Yang J J and Qian H 2020 Fully hardware-implemented memristor convolutional neural network Nature
28. Wang, Q.; Niu, G.; Ren, W.; Wang, R.; Chen, X.; Li, X.; Ye, Z.; Xie, Y.; Song, S.; Song, Z. Phase change random access memory forneuro-inspired computing.Adv. Electron. Mater.2021, 2001241.
29. Z. Yu, A. M. Abdulghani, A. Zahid, H. Heidari, M. A. Imran and Q. H. Abbasi, "An Overview of Neuromorphic Computing for Artificial Intelligence Enabled Hardware-Based Hopfield Neural Network," in IEEE Access, vol. 8, pp. 67085-67099, 2020, doi: 10.1109/ACCESS.2020.2985839.
30. H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for general-purpose approximate programs," Commun. ACM,vol. 58, no. 1, pp. 105–115, 2015.
31. A. B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in 15th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2010, pp. 825–831.
32. L. Wan and D. Chen, "Ccp: Common case promotion for improved timing error resilience with energy efficiency," in Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design, ser. ISLPED '12. ACM, 2012, pp. 135–140.
33. S. G. Ramasubramanian, S. Venkataramani, A. Parandhaman, and A. Raghunathan, "Relax-and-retime: A methodology for energy-efficient recovery based design," in 50th ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE, 2013, pp. 1–6.
34. S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-and-simplify: a unified design paradigm for approximate and quality configurable circuits," in Design, Automation and Test in Europe, DATE'13. EDA Consortium San Jose, CA, USA, 2013, pp.1367–1372.
35. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. on CAD of Integr. Circuits and Systems, vol.32, no. 1, pp. 124–137, 2013.
36. M. Monajati, S. M. Fakhraie, and E. Kabir, "Approximate arithmetic for low-power image median filtering," Circuits, Systems, and Signal Processing, vol. 34, no. 10, pp. 3191–3219, 2015.
37. P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electronics, vol. 7, no. 4, pp. 490–501, 2011.
38. C.-H. Lin and I.-C. Lin, "High accuracy approximate multiplier with error correction," in Computer Design (ICCD), 2013 IEEE 31st Inter-national Conference on. IEEE, 2013, pp. 33–38.
39. M. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 32, no. 8 , pp. 1163–1176, 2013.
40. S.-L. Lu, "Speeding up processing with approximation circuits," IEEE Computer, vol. 37, no. 3, pp. 67–73, 2004.
41. S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Quality programmable vector processors for approximate computing," in Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture. ACM, 2013, pp. 1–12.
42. A. Sampson, J. Nelson, K. Strauss, and L. Ceze, "Approximate storage in solid-state memories," in Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture. ACM, 2013, pp. 25–36.

43. M. Shoushtari, A. BanaiyanMofrad, and N. Dutt, "Exploiting partially forgetful memories for approximate computing," IEEE Embedded Systems Letters, vol. 7, no. 1, pp. 19–22, 2015.
44. S. Venkataramani, A. Sabne, V. J. Kozhikkottu, K. Roy, and A. Raghunathan, "SALSA: systematic logic synthesis of approximate circuits," in The 49th Annual Design Automation Conference 2012, DAC'12. ACM, 2012, pp. 796–801.
45. A. Ranjan, A. Raha, S. Venkataramani, K. Roy, and A. Raghunathan, "ASLAN: Synthesis of approximate sequential circuits," in Proceedings of the Conference on Design, Automation and Test in Europe, ser. DATE'14. EDA Consortium, 2014, pp. 1–6.
46. K. Nepal, Y. Li, R. I. Bahar, and S. Reda, "ABACUS: A technique for automated behavioral synthesis of approximate computing circuits," in Proceedings of the Conference on Design, Automation and Test in Europe, ser. DATE'14. EDA Consortium, 2014, pp. 1–6.
47. M. Soeken, D. Grosse, A. Chandrasekharan, and R. Drechsler, "BDD minimization for approximate computing," in 21st Asia and South Pacific Design Automation Conference ASP-DAC 2016. in press, 2016.
48. Z. Vasicek and L. Sekanina, "Evolutionary approach to approximate digital circuits design," IEEE Transactions on Evolutionary Computation, vol. 19, no. 3, pp. 432–444, 2015.
49. Z. Vasicek and L. Sekanina, "Evolutionary design of complex approximate combinational circuits," Genetic Programming and Evolvable Machines, vol. 17, no. 2, pp. 1–24, 2016
50. C. Li, W. Luo, S. S. Sapatnekar, and J. Hu, "Joint precision optimization and high level synthesis for approximate computing," in Proc. of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC). ACM, 2015.
51. S. V. Shubham Jaina and A. Raghunathanc, "Approximation through logic isolation for the design of quality configurable circuits," in Proc. of the 2016 Design, Automation & Test in Europe Conference and Exhibition (DATE). EDA Consortium, 2016, pp. 1–6.
52. A. Sampson, W. Dietl, E. Fortuna, D. Gnanapragasam, L. Ceze, and D. Grossman, "EnerJ: Approximate data types for safe and general low-power computation," in Proc. of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation. ACM, 2011, pp. 164–174.
53. A. Yazdanbakhsh, D. Mahajan, B. Thwaites, J. Park, A. Nagendrakumar, S. Sethuraman, K. Ramkrishnan, N. Ravindran, R. Jariwala, A. Rahimi, H. Esmailzadeh, and K. Bazargan, "Axilog: Language support for approximate hardware design," in Design, Automation and Test in Europe, DATE'15. EDA Consortium, 2015, pp. 1–6.
54. S. Misailovic, M. Carbin, S. Achour, Z. Qi, and M. C. Rinard,"Chisel: Reliability- and accuracy-aware optimization of approximate computational kernels," in Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications. ACM, 2014, pp. 309–328
55. H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for general-purpose approximate programs," in Proc. of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture.IEEE Computer Society, 2012, pp. 449–460.
56. S. S. Sarwar, S. Venkataramani, A. Raghunathan, and K. Roy, "Multiplier-less artificial neurons exploiting error resiliency for energy-efficient neural computing," in Proc. of the 2016 Design, Automation & Test in Europe Conference and Exhibition (DATE). EDA Consortium, 2016, pp. 16.