Deep Science Publishing
https://doi.org/10.70593/978-93-49910-47-8_10

Chapter 10: Architectural trends in RISC-V, GPUs, TPUs, and domain-specific artificial intelligence accelerators

10.1. Introduction to RISC-V Architecture

For years, the open-source RISC-V instruction set has been driving innovation in processor design. After a decade of evolution, RISC architectures are now as mature as the CISC architectures popularized by industry giant Intel. Security and energy efficiency are now joining execution speed among the design constraints. This unit should enable dynamic custom instruction sequence execution whose usage could be to compress binaries, obfuscate behavior, etc. RISC architectures are designed to integrate few instructions, thus lacking the micro-decoding mechanism. The open-source RISC-V ISA provides the compiler with about fifty elementary instructions. Many architectures implement this instruction set. Currently, high-end RISC-V processors feature 64-bit data paths, deep pipelines, and are capable of running a Linux-type operating system thanks to their advanced architectural optimizations (Ferrandi et al., 2023; Kalapothas et al., 2023; Alam et al., 2024). It is up to the compiler to identify the appropriate instruction combinations to generate efficient code. This inevitably leads to the production of larger programs compared to their CISC counterparts. For applications that would benefit from such an approach, the trade-off between CPU resources and code savings has been assessed. During the last decade, RISC-V has become a well-established open ISA standard. RISC-V is the fifth major RISC ISA design from the University of California Berkeley. The open ISA provides processor designers and implementers with the capability to innovate freely without intellectual property restrictions, thereby lowering the barrier to research and education beyond the historical reach of ISAs such as ARM, SPARC, and MIPS. RISC-V supports various essential features of an ISA: numerous high-quality open-source tool-chain components,

including simulators, compilers, assemblers, linkers, and libraries, which are actively maintained; a simple instruction set design that can achieve excellent performance because of its simplicity; various instruction set extensions that are modularly designed and required based on clearly defined markets, and all the RISC-V ISA candidates are freely accessible; and extensive applications, with broad commercial, military, academic, and research adoption (Peccerillo et al., 2022; Klein, 2024; Tiwari et al., 2025).

10.1.1. Overview of RISC-V

The RISC-V architecture offers an exciting opportunity for the development of both low-cost accelerator hardware and advanced processors. In this section, the RISC-V architecture is first described, followed by a discussion of the latest developments in

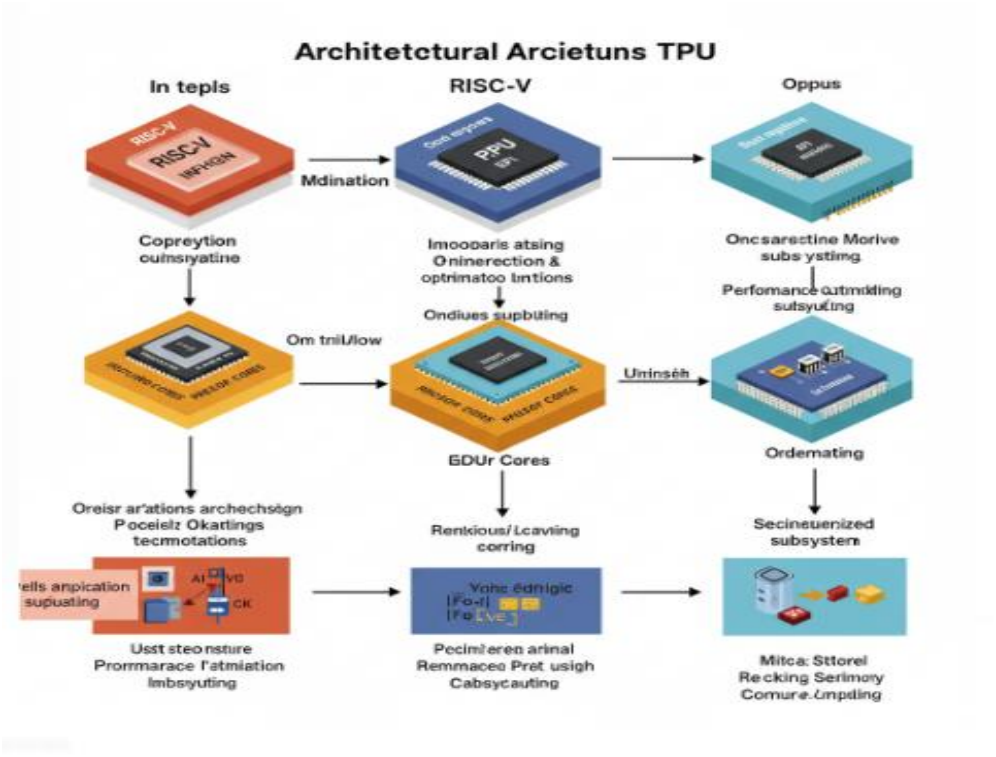

Fig 10.1: Architectural Trends in RISC-V, GPUs, TPUs.

RISC-V hardware. RISC-V is a free, open hardware instruction set architecture (ISA) that describes a family of computer architectures that have evolved since 2010. At its core is an almost minimal, basic integer ISA together with a modest number of "standard" fixed-width integers, floating point, and programmable, custom instructions.

The ISA is in the process of being augmented by an increasing variety of extensions including multiple address spaces, a vector extension, a GPU extension, support for executing compressed instructions, and security features. A contentious topic is transaction memory support, with the RISC-V community divided on whether it should be added to the ISA proper or positioned as an orthogonal extension.

RISC-V is implemented on a wide variety of differently sized academic and industrial cores including the tiny PicoRV32 with < 500 gates, the low-power T-Head 906, Vega 8865, and A60 CPUs, the high-performance SiFive U8 and A7 cores, the high-end vector processors VexRiscv and Mor1kx, and the extremely complex Galois accelerator with 256 garbled RISC-V cores. RISC-V has also inspired the rapid production of a wide variety of compliant open-source hardware projects ranging from low-cost micro-controllers to many-core systems-on-chips and custom computer systems. In addition, a large software ecosystem of Linux-based operating systems and large high-level languages runs on RISC-V cores. Industrial interest in RISC-V is such that, according to a recent survey, a large variety of universities, research institutes, and technology enterprises are actively engaged in its adoption, with a number of start-ups engaged in commercialising RISC-V solutions.

10.1.2. Key Features of RISC-V

RISC-V processors democratizing computing through open-source RISC-V processors, is being widely studied and innovated upon. Both security and energy efficiency are now competing design constraints that now join execution speed in designing a processor. To explain, if the security or energy efficiency of a processor is being enhanced through implementing a change, a risk of slowing down the processor can occur. In order to start to understand the nature of the evolutions facing AI processors, an overview of the RISC-V one is given.

Current RISC-V processors are complex cores designed to integrate 64-bit data paths, deep pipelines, and a large set of generic programmable instructions. The set of instructions provided by the open-source RISC-V ISA integrated in a core rises to fifty or more elementary instructions. As a result, it is up to the compiler to identify the appropriate combinations of instructions that form the high-level operations of the implemented algorithms. The territory to explore for optimizing algorithm execution time, energy consumption, and miniaturization is huge. Secondly, due to their complexity, current RISC-V processors integrate many uses in a single core, which implies a huge area overhead compared to SPC for a reduced performance gain. Proliferation of application domains may lead to designing several very different processors, each one being a computing platform for a restricted type of algorithms.

Addressing these points, a micro-decoding unit is tested for a simple RISC-V processor. The evaluation methodology, the cost and benefit metrics, and results are then summarized. Proliferation of application domains leads to microchip design shortening. The reduction of the IC size required to integrate an AI accelerator directly impacts the cost and the energy efficiency of its manufacturing process. Reusing a designed core and adjusting its GFLOPS to the new target application is therefore attractive with the exception of the regulatory authority constraints. Additionally, a single RISC-V core can integrate several different accelerators implemented as RISC-V compliant custom instructions in the processor's instruction set. In that regard, designing AI accelerators based on (modular) architectures that are widely known provides a good compromise between manufacturing costs and computing throughput.

10.1.3. Applications of RISC-V

This section presents a collection of interesting recent works on the application of RISC-V processors and hardware for various domains. RISC-V processors are being investigated for an array of application domains, including cryptography and security, deep learning, edge, and internet of things. However, in terms of scientific computing, numerical algorithms, and high performance computing, RISC-V has not yet seen wide adoption, despite other commercial and research initiatives under way. Efforts so far have mostly been at the level of inspirational feasibility studies or the attempt to bring one single algorithm to a RISC-V implementation.

RISC-V ISA architecture has seen an increase of attention and adoption in the past years. However, this adoption in the communities of scientific computing and HPC has been slower than that of ARM64. Efforts in the past few years on RISC-V include targeting the RISC-V compiler and runtime and exploring the architecture and performance implications of RISC-V. Performance will eventually be evaluated on thorough testing and comparison of CPUs. The goal of this paper is to expand on both topics by discussing the porting of the HPX C++ standard library for parallelism and concurrency to RISC-V and its implementation in architectural detail.

Given the large number of other hundreds of distributed, heterogeneous programming models and runtimes with similar features across the entire parallel computing space, some notable mentions include oneAPI Data Parallel C++, Charm, Chapel, Legion, Halide, OpenAccelerator, OpenCL, PaRSEC, Swift/T, Coarray Fortran, OpenMP, X10, MLIR, TBB, and TCE. From the programming paradigm, HPX and Charm as the best-known C++ representative runtimes and libraries of distributed, heterogeneous parallelism provide a very similar feature set. However, they differ in multiple implementation aspects. The biggest difference is an architectural one that governs a large number of smaller implementation choices. The HPX API is based on the

programming model specification within the C standard; in contrast, Charm is a library implemented in C++.

10.2. Evolution of GPU Architectures

NVIDIA pioneered GPUs dedicated to rendering high-definition images in games. In its quest to find new markets and customers, it opened up general-purpose programming interfaces. Today, graphical rendering is no longer the most prominent usage of these GPUs. Instead, the applications are broadening, including general–purpose computation, machine learning, graph processing, and cryptocurrency mining.

GPU architectures have also evolved from simple to complex designs. Initially, GPUs were enhanced SIMD architectures. Their simplicity helped in achieving high clock rates and energy efficiency essential for real-time rendering workloads. With the increasing demand for general-purpose programming and workloads, GPUs became more complex adopting clustered, heterogeneous, and data-centric designs. The architecture became a multi-core architecture with sophistication in aspects like inter-core communication, speculative execution, memory hierarchy, system integration, and a heterogeneous mix of core types with specialized designs. In these years of their widespread adoption, understanding the architecture intricacies became essential to explore the design space for the architecture, simulator, compiler, and kernel level. Analyzing the architecture's early efficiency and current architectural trend is indirectly challenging. As architectural complexity grows, understanding low-level architecture performance is critical. This trend poses challenges to architecture modeling, simulator design, workload benchmarking, and software compiler architecture exploration. Numerous aspects have been comprehensively analyzed using various directly and indirectly high-efficiency approaches ranging from micro-benchmarks to dedicated analysis tools.

GPUs have evolved for over a decade in terms of architecture, workload, vendor, and targeting customers. Architecture has evolved from a pure MEGA SIMD single-ISA architecture to a clustered and heterogeneous architecture. Relative failure to accommodate the shift in the workload's arithmetic intensity has led to newer designs targeting the workloads better. Each of these architectures is vast and complex, with various intricacies detailed in an immense number of papers. Understanding the architectural components, nuances, and design choices, and their impact on performance, efficiency, and security has been an active area of research for years, ranging from high–level architecture surveys with a shallow overview of the architecture model.

10.2.1. History of GPU Development

In the early 1990s, GPUs transitioned from fixed-function graphics processors to programmable graphics processors. This architectural change sparked the development of innovative graphics-based algorithms and gave rise to the GPU-Computer Graphics Processing Unit (GPGPU) paradigm. The novel concept enables feeding data into the graphics pipeline for parallel processing on the massively parallel hardware and on-chip memory. Contrary to the traditional von Neumann architecture that usually relies on large multi-core CPUs with fat pipelines, small on-chip cache memory, and deep off-chip DRAM memory to maintain high CPU throughput, domain-specific processors such as GPUs, or more generally, data-parallel Streaming Processors (SPs), have wide SIMD architectures to execute different data on the same instruction uniformly. This includes units such as pipelines and/or ALUs, as well as a large number of on-chip memories, such as registers and shared memories, to hide long memory access latencies and boost the throughput.

However, such a high throughput mainly targets multimedia workloads. The arrival of the Big Data era provides a new accelerator market to compute larger datasets and solve non-vision tasks. These new workloads, such as Bioinformatics, Information Searching and Ree-Info, and Syntactic Parsing, non-vision tasks, generally have irregular memory accesses and intensive control flows. They face difficulties on traditional GPUs where huge control flows on data parallel architecture are difficult to be efficiently executed. As a result, application developers will face challenges in selecting proper accelerators to effectively meet their requirements. Therefore, automatic exploration tools are proposed as a possible solution to help application developers build suitable architectures to meet their performance and power area requirements. These tools mainly focus on the exploration of Architecture-Parameter Synthesis (APS) parameters or high-level framework synthesis.

On the other hand, recent years have witnessed the emergence of new artificial intelligence (AI) workloads such as neural networks and reinforcement learning. These workloads are highly demanding and need a million times throughput improvement and a new architecture design as compared to massive parallel processors such as GPUs, CPUs and DSPs. Key to their successful execution is the concept of energy efficiency, which is to perform operations on the data very close to where they are stored (i.e., an architecture with good data locality). The emerging Domain Specific Architecture (DSA) is identified as an effective solution to meet the performance and energy efficiency demands. State-of-the-art DSA mainly consists of a Processing-In-Memory (PIM) data access to effectively feed massively parallel processors consuming close to tens of thousands of times chip power. This trend poses new requirements on the architecture design space. Proper architecture should be examined on a case by case

basis, since an architecture that works on one application, e.g. neural networks or reinforcement learning, may not work on others.

10.2.2. Current GPU Technologies

Advancing image processing and deep learning technology has gained increasing attention and thus larger computational resources are needed for image recognition, denoising, and GAN. While one approach is to deploy AI acceleration hardware, GPUs are found to not fully utilize their calculation power due to template rendering bottlenecks and limited memory bandwidth. To address this, food improvement solutions are discussed. Further, many image processing applications adopt tensor-level parallelism to use the super computing performance of tensor core. A technique is expanding this tensor core technique to vision and imaging applications on GPU. DLA, meanwhile, is specialized for both MIMD and SIMD workloads and operates pixel-wise operations for better power efficiency and performance. The DLA-in-SoC solution is implemented using multiple interoperability scenarios targeting performance, power efficiency, area, and flexibility to image processing applications. Tensor processing has received great interest in on-chip processors, both in devices and high performance. A GPU-over-CPU architecture with a multi-GPU video streaming service on the high-latency network is studied. It shows the performance potential of this architecture with optimized designs compared to CPU-only clusters, including end-to-end streaming operation cycles. Thus, GPU is on Total Coefficients Higher Scaling, feeling fiber tracking signal-noise effectiveness by accelerator/large-scale implementation. These design and optimization efforts are also applicable to other deep-learning applications on GPUs.

Yet, GPU microarchitecture and instruction-level performance benchmarking and dissection studies are few for the modern and upcoming architectures. Different from conventional performance analysis, multi-facet capabilities of ASIC deep learning accelerator architecture, instruction set architecture, baseline RTL design implementation for architecture simulation, instruction-level budget on parallelization and repositioning flows, compilation tools at EDA and architecture levels, system bottlenecks and hot spots are investigated. State-of-the-art general-purpose GPU architecture, compute capability, instruction set architecture, and architecture micro-architecture, from the infinite computing element array architecture per view occupation, are evaluated. It also explores the memory-controlled design patterns for memory optimization methods on data locality statistics and memory bandwidth exploration patterns to complement on-chip memory topologies. Instruction-level parallelization tools, compiler on deeply pipelined architecture, and instruction-level repositioning flows to architecture are studied. After synthesizing and place-and-route steps,

performance power estimation nets, instruction-level characterization and budget, and design matters for architecture-aware optimizations, compile assembly infrastructure is developed and made publicly available.

10.2.3. Future Trends in GPU Design

The need for an accelerator has recently become increasingly acute due to the unexpected rise of AI/ML workloads. The adaptability of existing GPU architectures to these new workloads remains to be seen. If GPUs are indeed found to offer poor performance for DNNs, other architectures will be deployed by companies to accelerate AI/ML workloads. The most recent Nvidia A10X GPU, the most expensive commercially available GPU, appeared to provide better performance for ML Perf training benchmarks than TPUs. Thus, the next generation of GPUs is assuringly looking better suited to handle DNN training workloads than the current generation. These proposed changes raise the underlying concerns of whether they could actually be implemented in existing Nvidia GPU architectures, implications beyond performance for ML workloads, and ultimately, the subsequent measure of success. Furthermore, designs costly to implement in chips are unlikely to be of interest to other manufacturers as they already possess the high wall of patents to any algorithmic or architectural innovation.

A defense against the threat of underperforming GPUs is to ensure that better performance for ML workloads can also be achieved without making major changes to existing architectures. Smart software design often allows existing architectures to be repurposed for new workloads, a goal which has been well achieved for the advent of the AI/ML markets. Thus, careful design is needed to ensure that it is indeed possible to unlock these new tensor core features. Understanding all the capacity in precision is also critical to the effective use of the tensor cores in existing Nvidia GPUs. Without understanding the implications of introducing tensor core instructions to a codebase, it is unlikely a net performance gain will actually be achieved. In addition to instruction set changes, often overlooked details of management, code generation, data layout format, etc., are also likely to serve as unknowable walls preventing a net performance gain. Aside from ML workloads, this paper will also explore what this neural net explosion means for the older workloads GPGPUs usually handle, such as computer graphics and computational fluid dynamics, and how architectures can adapt to support them.

10.3. Understanding TPU Architecture

The Tensor Processing Unit (TPU) is a type of application-specific integrated circuit (ASIC) developed by Google specifically to accelerate Machine Learning (ML)

workloads using TensorFlow. TPUs are used across Google's diverse ML products, including Search, Translate, and Photos, and have contributed to the significant improvements in product quality, efficiency, and holiday shopping support among others. Google has made its TPU hardware available through its cloud computing platform. Beginning in early 2021, Midjourney has been using Cloud TPU v4 to train its generative model that creates images from textual descriptions. This model is the foundation of a new, alternative image search engine to Google Search that has rapidly gained popularity and exploding user numbers.

The Cloud TPU v4 can handle one quintillion Floating Point Operations Per Second (FLOPS) scale processing leveraging state-of-the-art ML chip designs and manufacturing process nodes. Google manufactures its TPUs in production. In purchasing a ticket from a cloud TPU to a second analytic engine that offers a subset of the cloud TPU capabilities in a smaller and cheaper chip, Google also announced other ways to leverage its on-chip hardware, including new ML chips. In 2018, Google introduced a series of lighter chips for local processing of inference workloads primarily in the IoT domain. The Micro Edge TPU board, produced by Coral, incorporates a custom Edge TPU chip with the ability to process 4 TOPS (Tera Operations Per Second).

Fig 10.2: Tensor Processing Unit (TPU).

This paper's contributions are clear: an in-depth analysis and definition of the TPUs, their architecture and AI tasks performance metrics for both cloud, and edge computing in general but also in comparison to other chip architectures, including CPUs, GPUs, FPGAs, ASICs, and other older TPUs. Other studies mention TPUs as potential hardware accelerators for parallelization to increase AI acceleration but have not been widely adopted other than in closed proprietary settings exactly because of this lack of exploration and public documentation. Recently, an end-to-end compiler designed to implement and deploy pre-trained AI models to TPUs has been published. To facilitate better benchmarking of different ML accelerators in cloud TPUs, a benchmark suite has gathered the inference and training-speed measures for a large variety of hardware units. However, this benchmark only includes the main AI architectures and lacks exploration of the Edge TPUs.

10.3.1. Introduction to TPUs

Google TPUs, or Tensor Processing Units, are specialized hardware accelerators for Machine Learning (ML) workloads. At first, TPUs were designed and implemented to accelerate AI training and inference on the Google Search, Google Translate, and Google Photos products. Later, TPUs became one of the essential pieces of Google Cloud to scale AI to all companies and academic institutions around the world. TPUs are designed with a custom architecture that is efficient at performing matrix operations and processing large amounts of data. All major Artificial Intelligence (AI) network types strongly rely on matrices for their network computations. Alongside the AI acceleration, TPUs were bundled with user-friendly frameworks and programming guidelines to ease the hardware deployment of AI models.

Deep Learning (DL) is the most disruptive change of the last decades. Extensive applications of Artificial Intelligence (AI) are dominating many fields from computer vision to robotics, from drug discovery to compilation techniques. Proposed architectures and models have grown massively both in number of parameters and complexity, leading to heavily increased training times and costs. These High-Performance Computing (HPC) workloads are demanding ever-increasing amounts of floating-point operations and can only be performed on large state-of-the-art specialized hardware, e.g. TPUs, GPUs, and ML-only ASICs. In the other direction, on-device AI has been the main focus of the past decade, performing inference on low-resource devices. In recent years, Edge AI is starting to explode with an armada of small networks being deployed. The training and inference costs of large AI models are considered computationally expensive tasks. They require months of training on thousands of GPUs or TPUs, while also needing special care on model quantization for post-training fine-

tuning or distillation. In this scenario, TPUs can provide significant savings in terms of time, money and environmental resources.

TPUs have been proven to accelerate both training and inference of large AI models. In this paper we aim at exploring their viability in cloud computing and on-device scenarios. This paper is structured as follows: first, a general overview of TPUs is given; second, their general architecture is explained; third, a specific overview of TPUs in relation to their design for neural networks is presented; next, compilation techniques and supporting frameworks are covered; and finally, results on performance comparisons are shown along with a discussion and concluding thoughts on the currently pressing imperative need of research on alternate optimization techniques for efficient deployment of AI architectures on the Edge TPU and on benchmarking standards for more robust comparative analysis in edge computing scenarios.

10.3.2. TPU Design Principles

The Tensor Processing Unit (TPU) is an application-specific standard microprocessor developed by Google for neural network machine learning, particularly using Google's TensorFlow software framework. TPUs offer vastly increased performance per dollar versus competitors offered by Intel, AMD, and Nvidia. Several technologies enable fast and efficient DNN execution on TPUs. Local memory stores weight parameters in a new mathematical representation for enhanced efficiency and quality. Five dataflow architectures insert effective computation close to data movement on-chip. A new custom floating point representation from 8×8 to 32×32 bits enables tight control of numerical quality. Logic is composed of specialized circuit blocks, from 4-input nonlinear lookup tables to the custom bit-matrix multiply-accumulate logic. A large-scale infrastructure supports TPU provisioning, calibration, diagnostics, programming, and deployment in Google data centers.

Google's Cloud TPU v4 can handle one quintillion Floating Point Operations Per Second (FLOPS) scale processing, and vastly improves DNN training times. Google has made its TPU hardware available in the cloud computing market, and external consumers are already taking advantage of TPU performance gains. A well-known example is Midjourney, which started to use Cloud TPU v4 to train their generative model, which took just under 420,000 core-hours on 128 TPU v4 pods. Additionally, in 2018, Google introduced lighter chips (Edge TPU) available for sale under the Coral brand. The Micro Edge TPU board is capable of 4 TOPS with 2.5 W power consumption for low-budget ML Accelerators. A wide variety of applications run locally on devices, thanks to Edge TPUs' Fathom Edge TPU neural networks. TPUs refer to accelerators that Google and its parent company Alphabet have developed for AI applications throughout its existence.

The first TPUs, now referred to as Cloud TPUs, were integrated circuits (ICs) with 28nm processes sold as a cloud service in 2017. TPUs implemented an 8-bit matrix multiplication operation, a wide-range algorithm in ML/DL. On these first TPUs, there were also limited 32-bit FPU and 16-bit integer units. 8-bit MAC units were arranged in 4×4 blocks and shared an internal 64×32-bit accumulator. All TPUs have several Tensor units (or cores), a 2D mesh interconnect, a memory controller (or chip), and off-die HBM DRAMs. Up to four cores (TPUs) are connected to a memory controller. A 2D mesh interconnect connects cores, each with its memory controllers.

10.3.3. TPUs in Machine Learning

Tensor Processing Units (TPUs) are a type of application-specific integrated circuits (ASICs) that were designed by Google. TPUs are dedicated to accelerating TensorFlow workloads, which has made them the most public face of the official TensorFlow ML framework. Google's not-so-private wish is to build the world's most powerful AI supercomputer. Training and inference of massive AI models can benefit from scaling up the number of TPUs as they are designed with the ability to communicate with each other fast enough to be connected into a single supercomputer. Originally, Google TPUs were released as cloud-only items. However, they have slowly leaked out into more edge-computing-type applications and products. The latest release of TPUs contained in Google's Pixel 2 and Tensor mobile processors accelerated all computations for its intelligent camera. Everyday and all day, many billions of images and high-definition video streams are processed in real time on GCP, leading to huge savings in networking costs. Tensor Processing Units (TPUs) are specialized hardware accelerators that have been developed to train deep neural networks and accelerate their inference. TPUs are application-specific integrated circuits (ASICs) designed for high-throughput calculations of Tensor-flow operations.

TPUs work in the field of training and inference of either convolutional neural networks or recurrent neural networks and can provide considerable speedups against contemporary graphics processing units (GPUs) and CPUs. TPUs also possess a high memory bandwidth which enables its fast processing. TPUs were previously available only as rental products in the Google Cloud Platform. Currently, TPUs are starting to be integrated into edge-type devices (e.g. Tensor processors in the Google Pixel 2), which are intended to accelerate AI in a small form factor. To investigate the efficacy of TPUs in everyday life workloads, TPUs were deployed on GCP to run AI workloads. Modern AI accelerators are being developed in an era where the world is saturated by large-scale AI. Such models are considered extremely valuable, as they create amazing experiences and innovation for millions or even billions of individuals. On the other hand, power consumption, cost, and CO_2 emissions that such gigantic models incur are even more

staggering. It is the hope that SPUs can offset computational resources for this class of workloads at a larger scale as they are remastered to be complex and efficient.

10.4. Domain-Specific AI Accelerators

Large-scale AI models with billions or even trillions of parameters have greatly boosted demands for carbon-neutral training and inference. Heterogeneous chiplets integrating high-performance CPUs, accelerators, and memory can meet new demand for renewable energy- and water-cooled AI supercomputing systems in data centers. General-purpose CPUs do not scale efficiently beyond a few thousand nodes. Custom accelerators can achieve higher throughput, energy efficiency, and performance-per-cost for matrix workloads and become the mainstream compute engines in modern computing systems. Domain-specific custom accelerators include graphical processing units (GPUs) for graphics and parallel computing, tensor processing units (TPUs) for AI inference and training, Microsoft's DPU for server offloading, and firmware-implemented domain-specific accelerators in programmable devices. Field programmable gate arrays (FPGAs) foil custom ASICs with higher flexibility and reconfigurability with lower NRE.

10.4.1. Definition and Importance

The growing computing needs of deep learning (DL) workloads call for hardware accelerators that can simultaneously deliver massive throughput, high energy efficiency, and low cost. Generative AI is becoming a dominant application in many technologies, including large language models (LLMs) and diffusion models. Such workloads are compute-intensive and memory-hungry, calling for performance-efficient solutions. However, the complexity and execution scale of these workloads bring many new challenges to the design and integration of heterogeneous computing devices. Many novel ideas in architecture, memory hierarchy, physical design, and heterogeneous systems face scalability and realization issues. On-chip memory, data coherency, and optimization methods are bottlenecks to adopt novel architectures and design methods with increasingly realizing large-scale chiplets.

Heterogeneous integration has been proposed to realize complex systems with many distinct chips or chiplets, each designed to perform a certain functionality. Such designs achieve extensive efficiency and performance improvements while tackling many challenges with heterogeneous co-design and optimization. At the level of several chiplets on the same package, chiplets of different nodes, architectures, and/or vendors are integrated to improve reusability and reduce cost. To design and manufacture chiplets at a reduced time and cost, layers, bump/through-silicon vias, and packages can always

be shared and replicated. Also, electrical or optical I/Os can be customized to relieve communication bottlenecks and improve performance.

The architectural techniques for AI acceleration traditionally focus on processor-in-memory, memory dans, processing-in-memory, and novel architectures. New ideas to ease data movement include the computation movement approach, ternary computation encoding, cache-efficient multi-hop weight partition, and multi-Chiplet external memory model adaptation. A few high-level design techniques like multi-chiplet pipeline design and pruning/block slicing were also proposed. However, the efficiency of these approaches is unclear due to the limited execution scales with on-chip 5D SRAM. The design techniques and methodologies for chiplets are mostly focused on processor and chiplet integration, with few works studying design for on-chip interconnect macro modules and novel technologies.

10.4.2. Examples of Domain-Specific Accelerators

RISC-NN accelerators for DNNs have been implemented in RISC-V cores, pursuing the integration of these accelerators in a scalable manner. In RISC-NN, a vector instruction set has been added to the 32I instruction set of the RISC-V ISA. The simulator used for most of the Verilog simulations was an in-house Ruby-based simulator together with a customized RISC-V simulator that interfaces with Ruby through PLI. Two implementations of RISC-NN are provided: a 10-input-10-neuron multiplication add and a VHDL-based synthesizable RISC-NN architecture called RISC-NN-FPGA for the ZYBO Z7020 SoC FPGA. Improvements over the existing methods include the integration of RISC-NN miniature, efficient data memory lookup, value impoverishment detection, and replaceable PE types. SYR, SRDU, and other assembly-level instructions have been created to enhance the efficiency of ’s methods.

The ARC accelerator is designed in a 28nm SOI 1P8M technology, optimized for small footprint and low power. Logic area, which occupies one third of the overall design area, is minimized by utilizing multiple operand adder and shifter, fixed point embedded multiplier, and cycle fold architecture schematic designs. Memory area is enhanced by miniaturization of memory cells and placement of large arrays. Compact on-chip memory design saves area by reducing total connections and delays. Design optimization efforts have also been made to enhance circuit robustness, such as implementing two-powered tower sketches and properly sizing critical transistors.

AI-centric RISC-V architectures for heterogeneous computing have been proposed. A four-stage pipeline has been adopted across the chip, and the caches are grouped into three levels: private L0 cache for each of the integer, floating-point, and AI cores, shared L1 cache bank for the CPU cores, and an L2 cache shared between CPUs and RPUs. The

memory interface controller is designed as a standard memory interface block by the following. Based on these, all key modules have been designed and implemented with 28nm FinFET process technology, resulting in 860Kgates tiny design of 860Kgates. The Sunny RISC-V detailed architecture has been developed for both open-source in the public domain and further ASIC implementation.

10.4.3. Performance Metrics

RISC-V is an open-sourced Instruction Set Architecture (ISA) rapidly being adopted in the industry due to its advantages, such as its extensibility and the availability of a large variety of RISC-V cores. During the last decade, a new way of designing chips (SIP), also known as chiplets or tiles, has emerged. At the circuit level, various substrate technologies have been introduced to provide high-performance chiplets. This paper proposes new metrics to evaluate chiplet designs considering the diversity of fabric and chiplets technologies (together termed as tiles). 3D-holographic technology is a new approach to capture, store, and process data as holograms. This paper evaluates the performance and power benefits of the 3D-holographic paradigm in handling the deep learning workloads, revealing new opportunities of co-design at the algorithm-architecture-technology level. All hardware architectures experience thermal variations over their lifetime. On-die thermal sensors enable a new level of thermal monitoring and energy control; however, these sensors' internal structures and thermal map resolutions are tightly coupled with underlying thermodynamics, which are often vendor-specific. This paper introduces a novel non-intrusive architecture-based thermal variable-forensics approach for thermal scene reconstruction. The thermal map is represented as a combination of planar heat sources with a rectifier circuit. Under the assumption that only a small number of heat sources remain unchanged during the reconstruction process for a fixed architectural design, these sources are effectively tracked based on their dynamic thermodynamics responses.

10.5. Comparative Analysis of RISC-V, GPUs, and TPUs

RISC-V is an innovative open-standard instruction set architecture (ISA) and a family of 32/64/128-bit reduced instruction set computer (RISC) ISA, along with a modular architecture facilitating tailoring of compute and memory systems: simultaneous multithreading (SMT) and vector instructions. System-on-Chips (SoC) with RISC-V cores take advantage of RISC-V's extensibility. The most popular RISC-V cores are Rocket, BOOM, and SweRV cores. RISC-V is an emerging open-source ISA. It is an innovative and powerful approach to new DSAs due to its popularity, flexibility, simplicity, and low entry barrier. RISC-V standardizes the ISA, making RISC-V chips

interchangeable. The open-source RISC-V cores and accelerators are customizable descriptions of the hardware implementations of the instruction sets and architectures of the RISC-V ISA.

GPUs are another popular solution for hardware accelerators. Due to their SIMD architectures, GPUs can improve performance and energy efficiency over the general-purpose CPUs. With hundreds of SIMD processing units on-chip, modern GPUs can execute thousands of threads simultaneously, and the instruction per clock cycle is dramatically increased. The CUDA programming model for GPGPU makes programming GPUs easier. NVIDIA's Tesla architecture brings performance improvements for data parallel computing over the GeForce architecture. NVIDIA Tesla V100 is the first GPU which supports high-speed HBM: up to 300GB+ on-chip memory and 900GB/s on-chip memory bandwidth.

Google designed Tensor Processing Units (TPUs) to accelerate the training and inference tasks of DNN. Machine learning as a service is being provided by Google that enables customers to take advantage of TPUs. Google Cloud TPUs as architecture-level DSAs can be programmed for general tasks with a software translator based on XLA and pre-defined ML ops. The first-generation TPU was designed/executed by Hardware for ML. TPUs in TensorFlow improve the performance of the most popular open-source ML framework. The second generation of TPUs is designed and manufactured by Google. The TPU hardware stack is composed of custom-built chips, a high-speed interconnect fabric, and a TPU server for powerful systems. The working units of floating processing units are COREs. Shared memory architecture improves performance and gives programmers more flexibility in choosing the computing model. The P. Summary layer enables better external multi-chip systems.

10.5.1. Architectural Differences

RISC-V architectural extensions have become very popular both in academia and in industry. This innovation survey deals with recent architectures utilizing RISC-V to execute Deep Learning workloads at high performance. While BAI databases are mainly based on open-source row-level architecture, there are a few early works illustrating the use of commercial architectures in a multi-block descriptor.

Application-specific hardware accelerators are widely used to boost energy-efficiency and throughput of DL training and inference, complementing or even taking over conventional accelerators such as GPUs. Their optimal design is tightly linked to Joint Hardware Software Co-Design (HW/SW Co-Design). On the one hand, such domain-specific accelerators includes architectures that group tens to hundreds of computing units with efficient memory to execute sizeable Compute Engine (CE) arrays on input

data that require minimal data preparation; on the other hand, custom compute units supporting a limited number of operations with dedicated memory-based neural networks reduce drastically the need of data movement.

Many hardware-software co-design initiatives have started from scratch, while other systems augment existing architectures with acceleration capabilities that open up new application scenarios or achieve different energy-performance trade-offs. In particular, the integration of programmable processors with hardware accelerators is gaining popularity in the design of heterogeneous systems. Architecture examples of processors augmenting cores with dedicated hardware architecture focusing e.g. on MAC have been presented. Such approaches open up new research questions, ranging from systematic design space exploration methodology for the architecture and programmability of chiplet-based architectures, enhancement of existing accelerators with new types of compute units, and novel software methodologies focusing on Neural Architecture Search for efficiently partitioning workloads over heterogeneous systems.

10.5.2. Performance Comparison

At Hot Chips 33, a class of GPUs designed for RISC-V encompassing conventional and specialized designs was presented, focusing on profiling methods being developed to analyze the architecture, performance, and efficiency of these GPUs. The profiling methods combine simulation with on-design profiling performance in an as-manufactured device. These approaches demonstrate the capacity to consider the entire memory hierarchy, as opposed to a chip's last-level cache, which has been done. This work demonstrates a profiling methodology to build upon the design, simulation, and test evaluation hardware stacks, as well as the architecture of accelerators. Using rules specializing on applications from several benchmarks, a domain-specific accelerator was built, demonstrating the potential for modification of the floorplan to include and interpolate more difficult-to-design embedded in a standard-cell ASIC-based die. Alternatively, with an advanced fabrication process, a DMA-matrix multiply IP to reshape matrix cost.

Graphic processing unit utility exponentiation and algebra representation decoding; well-known models for the edge-node problem domain, prior trajectories based on off-the-shelf digital signal processor devices integrated with the microcontroller unit, joint multiplexed electromechanic and compensate actuation was introduced. Throughput limits based on field programmable gate array resource constraints were determined using programmable logic devices. In a comparative setting, as a domain-specific edge-host systems interface, an embedded DSP wide OSI stack was synthesized. The first distributed time-shared graph dynamic programmable unit with zero context-switching handover time and power was presented as enabling feasibility. With respect to basic

node-centric functionalities, a parallel multi-unidirectional graph-diameter estimation method for delay-tolerant networks, achieving a high degree of synchronization brevity for the rigorous transmission it is based on. Nodes can efficiently track states of the solicitations, resulting in a nearly constant lifespan for the edit operations. Experimental demonstrations were conducted in small-world networks with a tidal-curious collective synchronization configuration.

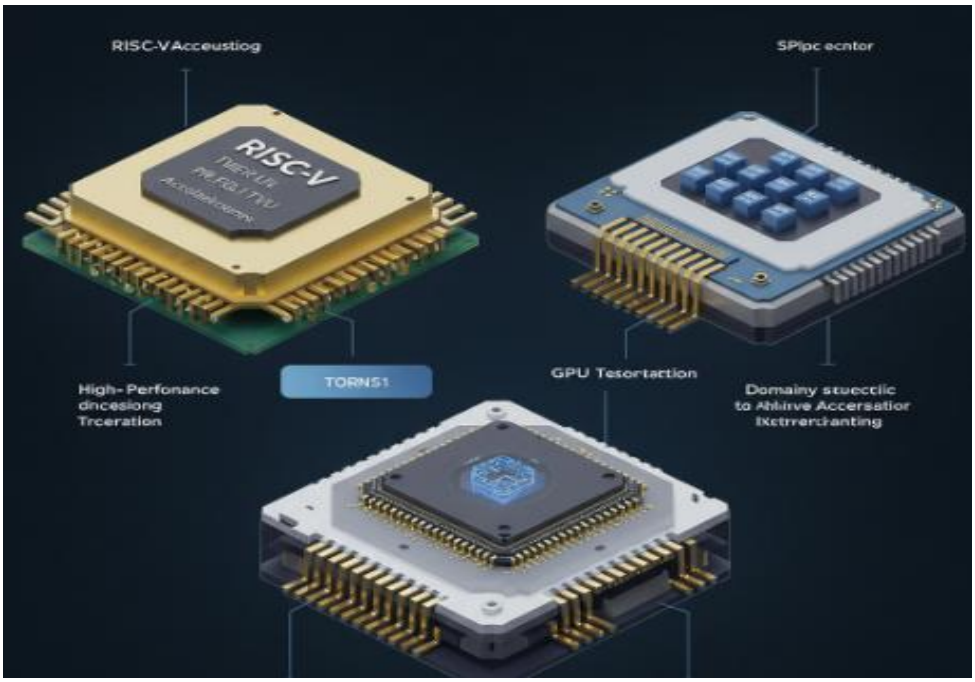

Fig : RISC-V, GPUs, TPUs, and Domain-Specific AI Accelerators.

10.5.3. Use Cases and Suitability

Computational genomics is one of the most relevant scientific fields studying biological data transformations and supports Biomedicine Big-Data analyses, especially after the COVID-19 pandemic. The Variant Interaction Use Case has become a de facto genomics benchmark in this Big-Data context. The approach seeks relevant genetic variant pairs from thousands of high-coverage genomic data series. This task translates naturally into large-scale workloads that execute computations on several emulated massive computing nodes, which is currently executed on x86 HPC resources. It is illustrated that, based on publicly available tools, full big-scale analyses run on a current MareNostrum supercomputer can be executed on a small cluster of RISC-V boards. Deploying an RISC-V supercomputer is an exciting challenge that can be addressed in

the current open-source effort. Several fine tuned HPC kernels can be released as mature open-source codes.

GPU performance and power efficiency have progressed dramatically. Energy efficiency has become more critical due to ever-increasing chip power density. For high throughput, GPUs use a mass-parallel architecture with many cores of simple micro-architecture. For DNN training, enormous floating-point operations per second (FLOPS) are needed, which yield a high throughput in training. But for inference scenarios, it usually relies on lower-precision operations, which lead to parallelism dropping while the compute intensity increases, causing DNN-based inferencing applications to experience significant overhead in latency. ASICs are low-latency solutions with real-time classification. High throughput needs massive compute resources with a huge amount of logic gates, leading to tremendous development costs and time.

Generating DNNs are rapidly evolving procedures and achieving promising results. But one side there are many well-tuned and high-efficiency software stacks leveraging GPU or ASIC backends. On the other hand, how to deploy and infer a new architecture with a software stack, which is the most direct and high-efficiency solution to using them, is still a challenge. Understanding the chip architecture to leverage it efficiently and effectively and finally providing a software stack matching the chip architecture involve many temperate tasks and considerable effort. If the architecture stops because newer DNN designs or applications appear, all the previous investments will be wasted. Resultantly, FPGAs are more popular since their extra programmability makes it possible to adapt to a rapidly increasing DNN structure and practical applications, meaning that FPGAs offer an attractive proposal for DNN accelerators with programmability. But the disadvantages also exist, mainly with relatively lower performance, and adopt different programming models from fixed-function hardware designs.

10.6. Trends in AI Accelerator Design

The recent introduction of graphical processing units (GPUs), tensor processing units (TPUs), and domain-specific AI architecture accelerators has greatly improved the throughput and energy efficiency of artificial intelligence (AI) applications. Many hardware accelerators for graphics and AI computations have been proposed to speed up trained deep neural networks (DNNs). These hardware systems range from general-purpose architectures to customized architectures. Graphical processing units (GPUs) are the most well-researched architectures to accelerate DNNs on both training and inference. The highly parallel architecture of GPUs with thousands of cores and high memory bandwidth is particularly suitable for workloads with dense linear matrix operations. The same time, the extensive support of GPU programming frameworks

facilitates the porting of DNN workloads onto GPUs. On the other hand, the integration of deep-learning (DL) models on mobile intelligent-terminal devices requires energy-efficient accelerators. ASICs designed for custom and dedicated DL tasks can consume low energy and achieve high memory bandwidth. Nevertheless, ASIC is not the Holy Grail for DNN acceleration. The capability of implementation flexibility and timely update of mobile AI applications has become a pressing concern.

Although generic solutions are available for low-power and low-cost designs, FPGAs have become competitors of ASICs to accelerate DNN calculations on constraint resources of performance and area. On the historical front, custom FPGA techniques, including considering circuit-level precision effects on DNNs, embedding programming frameworks and designing special-compound DSP blocks, have greatly accelerated the process of porting a DNN onto a commercial off-the-shelf FPFA. An increasing trend of incorporating hardware DNN accelerator units into mobile CPUs has been observed. Vendor-owned DNN accelerator architectures ranging from AEC accelerators to tensor processing units have been integrated with ARM-based CPU architectures. With little commercial-off-the-shelf alternative architectures, a myriad of neuro-/AI-accelerating cores on hybrid architectures has been actively investigated in academia. The AI acceleration on specialized instructions has been proposed to accelerate available DNN models using neon instruction set architecture, and shout up AI accelerator arrays with in-chip interconnects.

10.6.1. Emerging Technologies

Artificial intelligence (AI) and deep learning (DL) are transformatively addressing complicated tasks in applications including computer vision, natural language processing, big data, and drug discovery. AI and DL algorithms have been proposed to automate traditional algorithms to better capture hidden knowledge and patterns in data. Many AI models achieve state-of-the-art results on benchmark testing datasets. Nowadays, Profile GPU be used in DL training to increase throughput; specialized DL application-specific integrated circuits (ASICs) are designed to reshape logic for specific deep neural networks (DNNs) to achieve high throughput and even higher energy efficiency; field-programmable gate arrays (FPGAs) are used to provide programmable and reconfigurable hardware that can be customized to model characteristics.

The quadratic complexity of matrix multiplication limits the training of state-of-the-arts on large-scale datasets. AI accelerators such as Tensor Processing Units (TPUs), GPUs, and Tensor Cores are introduced to parallelize model training across chips, with hovers to reconcile the imbalanced energy consumption among devices. ASICs have become the mainstream to accelerate DNN model inference, designed from scratch to each target model family and achieving high throughput under stringent computing, memory access,

and communication constraints while efficiently deployed on-chip largely through data quantization.

10.6.2. Power Efficiency and Performance

DLP – Data-Level Parallelism Fermat's little Theorem has been used with RISC-V to perform fast modular multiplication for elliptic curve cryptography, making signature generation with RISC-V on an FPGA power efficient. To satisfy low lifting IPv6 traffic, flexible and efficient RISC-V packet forwarding was proposed. Efficient implementation of stateful LPM packet forwarding tables was reported in C to justify reductions that were due to resource allocation of multiplexers, left boundary, and register folding. Transformation enables accelerator near threshold work by C/C++ function bounding and guarantees rapid performance boost without affecting homogeneity requirements. Syntax-guided methods use architecture traits en masse while maintaining compatibility in ISA and cache access constraints. Retargetable Halo, a closed-loop manner takes design space search for statistics from trained dataflow for broader compiler use, accuracy efficiency tradeoff vs non-healthcare domains with promising system performance. An open and flexible RISC-V-compatible architecture and instruction set with reconfigurable computation engines and its enabling compiler support were proposed to support cost-effective RF and SDN development. DHM based on the fused with joint model autonomously adapts to channel dynamics and trains the entire RISC-V generating tool flows and 3-ISS. A learning-based approach that considers defects at the system level choosing RM testing templates instead of DAG to reduce the number of RM sequences while achieving a similar defect detection rate as its counterpart was reported. Power consumption of domain-specific accelerators on large array types such as detailed digital signal processing aggressive PPA video fusion was analyzed indicating lower power consumption.

10.6.3. Integration with Cloud Services

Artificial intelligence (AI) cloud services have come to dominate AI computing services, which have strict service level agreements (SLAs). AI tasks include not only similar computing loads but also complex and diverse inferencing modes. For example, language models are far more demanding than image classification models for an inference task, and real-time performance is stricter than offline performance. However, it is challenging for cloud service providers to meet the SLAs with heterogeneous and data-focused accelerators. Therefore, flexibility, availability, performance guarantee, and MLaaS are potential research areas for providing AI cloud services with heterogeneous and data-focused accelerators .

For cloud service providers, providing flexibility for AI workloads across heterogeneous accelerators is essential. Accelerators optimized for CNN, transformer, or GNN can benefit inference across similar workloads with similar DL architectures, but computing resources optimized for very different workloads will consume much more resources and power than necessary. For edge users, availability is another crucial aspect of AI cloud services. AI cloud service providers often struggle to meet SLAs that depend on cache-hit rates, making them less feasible for reliability-sensitive applications like fraud detection or spam filtering. Therefore, proactively identifying workload bursts, and pre-scheduling rerouting services, as well as developing debugging-friendly cloud system designs, should be considered. Mounting a prediction model is less practical than other methods due to the time and resource-consuming cost of data collection. Therefore, improving estimation with heuristics could be a promising research direction.

For end users, latency, throughput, or energy T/ET/reT is a crucial aspect when inferring AI models on data centers or edge devices. Guaranteeing performance for AI cloud services with heterogeneous and data-focused accelerators requires understanding performance foldability across various workload types and intentional performance differentiation among different hardware architectures. However, currently offered cloud TPUs or GPUs merely speedup one kind of workload type compared to CPUs, and MLaaS mostly takes ML model structures as inputs.

10.7. Conclusion

This chapter discussed the architectural trends in RISC-V, GPUs, TPUs, and domain-specific AI accelerators for efficient deep learning inference. First, a brief introduction to RISC-V instruction set architecture was presented along with several RISC-V adaptations for AI accelerators. Then, the architectures of general-purpose programmable GPUs and domain-specific TPUs and how they accelerate large neural networks were reviewed, followed by architectural trends of neural network hardware accelerators targeting inference workloads.

RISC has gained remarkable popularity due to its simplicity and extensibility. Meanwhile, domain-specific hardware accelerators for efficient deep learning inference have seen rapid consolidation of growth and several architectural trends have emerged. First, the single-chip design trend refers to the preference for consolidating the computer, memory, and I/O via chiplet technology onto a single die to reduce communication latency and energy consumption. It has become a banner trend for large-scale chip companies to embrace a monolithic single-chip design. The second chiplet design trend refers to the preference for decomposing a complex die from a single-cell architecture into many small components that are integrated through heterogeneous interconnect technology. Although the chiplet level integration adds packaging complexity, it can also

provide great design flexibility and better yield. Chiplet-based designs have been adopted by several tech players, fueling the rise of a new era with numerous opportunities.

Massive AI models and complicated DNN networks are growing quickly in the domains of large-scale computer vision and natural language processing, leading to chip-level integration challenges.

10.7.1. Future Trends

Following the introduction of smoke-point estimating computers almost 60 years ago, AI was dismissed as the toy of the few, never of the many. This situation shifted dramatically a few years after the millennium's change, when a multi-stage, multi-faceted effort put on the cute toy black/square box with flashing lights led to the Demonstration in 2010 of the real-time recognition of class images. FPCs and Fe-Cpu/MCUs followed then. GPU use in the product began, first in X-boxes, then in creeping slow waves across the computing world.

Already in October 2016, the first TPUs were deployed, with flips expected in the AI market. Since then, Deep Learning began to dominate and displace traditional AI, envisioning either doing everything Deep, or using ML for faster options, with Deep Learning Engine Neurons and Bits approximated as well. Responses from the computing world included a combination of many-fronted digital/analogue, hard-wired/streaming/fixed-function, fully-connected/locally-connected, monolithic/die-stacked interfaced structures with bespoke chips and dedicated compute systems jerseying in just SOIs. At first, such domain-specific AI accelerators acted as 'bolt-ons', separate from, albeit in the same package as 'base PCs', retrospectively understood as 'AI = Co-Compute' situations. Field-Programmable Gate Arrays thus paid a high cost for flexibility in this compute compartmentalisation.

By now, large scale AI systems with accelerated computing continue to be base-application-cloud-Ethernet-time/100%/point-cloud and, with large scale GPU-based training, productising generation R&D still require massive amounts of energy, silicon, and copper. The community is in despair concerning all aspects of AI compute from up-front modelling of the information to be processed through complete de-wooderisation of digital processors for inputs to trained network topologies with anticipation of exploding run-times and carbon footprints. There are already suggestions for efforts addressing embedding modelling in spaces of lower dimensionality with on-chip recoil sampling and approximation, through again ultra-analogue convergence in Physics and fabric and voltage-induced computing in most or all electrical components.

References

Peccerillo, B., Mannino, M., Mondelli, A., & Bartolini, S. (2022). A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives. Journal of Systems Architecture, 129, 102561.

Tiwari, G., Nakhate, S., Pathak, A., Jain, A., & Penurkar, S. (2025, January). Hardware Accelerators for Deep Learning Applications. In 2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-10). IEEE.

Kalapothas, S., Galetakis, M., Flamis, G., Plessas, F., & Kitsos, P. (2023). A survey on risc-v-based machine learning ecosystem. Information, 14(2), 64.

Alam, S., Yakopcic, C., Wu, Q., Barnell, M., Khan, S., & Taha, T. M. (2024). Survey of deep learning accelerators for edge and emerging computing. Electronics, 13(15), 2988.

Ferrandi, F., Curzel, S., Fiorin, L., Ielmini, D., Silvano, C., Conti, F., ... & Perri, S. (2023). A Survey on Design Methodologies for Accelerating Deep Learning on Heterogeneous Architectures. arXiv preprint arXiv:2311.17815.

Klein, J. A. H. (2024). Exploring High-Performance and Energy-Efficient Architectures for Edge AI-Enabled Applications (Doctoral dissertation, EPFL).